



**VICTORIA UNIVERSITY**  
MELBOURNE AUSTRALIA

*A multi-angle hierarchical differential evolution approach for multimodal optimization problems*

This is the Published version of the following publication

Hong, Zhao, Chen, Zong-Gan, Liu, Dong, Zhan, Zhi-Hui and Zhang, Jun  
(2020) A multi-angle hierarchical differential evolution approach for multimodal optimization problems. IEEE Access, 8. pp. 178322-178335. ISSN 2169-3536

The publisher's official version can be found at  
<https://ieeexplore.ieee.org/document/9208664>  
Note that access to this version may require subscription.

Downloaded from VU Research Repository <https://vuir.vu.edu.au/46348/>

Received September 15, 2020, accepted September 25, 2020, date of publication September 29, 2020, date of current version October 9, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3027559

# A Multi-Angle Hierarchical Differential Evolution Approach for Multimodal Optimization Problems

ZHAO HONG<sup>1</sup>, (Student Member, IEEE), ZONG-GAN CHEN<sup>1</sup>, DONG LIU<sup>2</sup>, ZHI-HUI ZHAN<sup>1</sup>, (Senior Member, IEEE), AND JUN ZHANG<sup>3</sup>, (Fellow, IEEE)

<sup>1</sup>School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China

<sup>2</sup>School of Computer and Information Engineering, Henan Normal University, Xinxiang 453007, China

<sup>3</sup>Hanyang University, Ansan 15588, South Korea

Corresponding author: Zhi-Hui Zhan (zhanapollo@163.com)

This work was supported in part by the Key-Area Research and Development of Guangdong Province under Grant 2020B010166002, in part by the National Key Research and Development Program of China under Grant 2019YFB2102102, in part by the Outstanding Youth Science Foundation under Grant 61822602, in part by the National Natural Science Foundation of China under Grant 61772207 and Grant 61873097, in part by the Guangdong Natural Science Foundation Research Team under Grant 2018B030312003, and in part by the Ministry of Science and ICT through the National Research Foundation of Korea under Grant NRF-2019H1D3A2A01101977.

**ABSTRACT** Multimodal optimization problem (MMOP) is one of the most common problems in engineering practices that requires multiple optimal solutions to be located simultaneously. An efficient algorithm for solving MMOPs should balance the diversity and convergence of the population, so that the global optimal solutions can be located as many as possible. However, most of existing algorithms are easy to be trapped into local peaks and cannot provide high-quality solutions. To better deal with MMOPs, considerations on the solution quality angle and the evolution stage angle are both taken into account in this paper and a multi-angle hierarchical differential evolution (MaHDE) algorithm is proposed. Firstly, a fitness hierarchical mutation (FHM) strategy is designed to balance the exploration and exploitation ability of different individuals. In the FHM strategy, the individuals are divided into two levels (i.e., low/high-level) according to their solution quality in the current niche. Then, the low/high-level individuals are applied to different guiding strategies. Secondly, a directed global search (DGS) strategy is introduced for the low-level individuals in the late evolution stage, which can improve the population diversity and provide these low-level individuals with the opportunity to re-search the global peaks. Thirdly, an elite local search (ELS) strategy is designed for the high-level individuals in the late evolution stage to refine their solution accuracy. Extensive experiments are developed to verify the performance of MaHDE on the widely used MMOPs test functions i.e., CEC'2013. Experimental results show that MaHDE generally outperforms the compared state-of-the-art multimodal algorithms.

**INDEX TERMS** Multi-angle hierarchical, two-stage search, differential evolution, multimodal optimization problems.

## I. INTRODUCTION

Multimodal optimization problems (MMOPs), as one kind of challenging and interesting optimization problems, have attracted increasing attentions in recent years [1]–[4]. MMOPs are an important problem area as it widely exists in many real-world applications [5], such as virtual camera composition problems [6], metabolic network modeling problems [7], laser pulse shaping problems [8], job scheduling problems [9], [10], and neural network problems [11]. These

The associate editor coordinating the review of this manuscript and approving it for publication was Pavlos I. Lazaridis<sup>1</sup>.

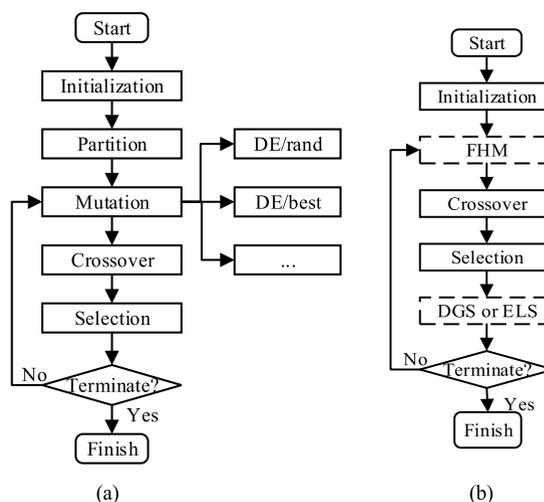
MMOPs are required the algorithms to locate the global peaks as many as possible and refine the accuracy of the found solutions as high as possible, so that the high-quality decisions can be finally made. In detail, MMOP is a kind of complex optimization problem that requires the algorithm to not only locate multiple global peaks simultaneously, but also achieve certain accuracy of solutions on the global peaks. In fact, the algorithm for MMOPs faces the problem of how to improve the population diversity to locate more peaks and accelerate the convergence speed on each of the found peaks.

Different from the single optimization problems (SOPs), MMOPs have many local peaks and multiple global peaks.

Most existing evolutionary algorithms (EAs) are successful used in solving SOPs [12]–[18]. Drawing on the success of EAs in SOPs, many advanced EAs are designed for solving MMOPs, such as differential evolution (DE) [19]–[28], particle swarm optimization (PSO) [29], [30], genetic algorithm (GA) [31]–[34], and ant colony optimization [35]. These algorithms all obtain a great success when dealing with MMOPs. Among these algorithms, DE variants have shown their effectiveness and superiority [19]–[28]. However, existing DE-based multimodal optimization algorithms still have some limitations. Firstly, the DE/rand or DE/best strategy are widely used in many algorithms, but there are still some problems. The DE/rand strategy decreases the convergence speed due to the random searching directions, while the DE/best strategy provides homogeneous guidance, which makes the population easily get trapped into local peaks. Therefore, different strategies should be used for different individuals according to their fitness quality. Secondly, traditional DEs may suffer from the dilemma of the balance between exploration and exploitation because they usually use the fixed evolutionary operators during the whole process. This is not efficient to satisfy the search requirements of different evolution stages.

Therefore, to solve above difficulties, this paper enhances the DE algorithm from two angles. The one angle is to consider the different fitness quality of the individuals so as to configure them with different mutation strategies satisfy their search requirements. The other angle is to consider the different evolution stages so as to carry out different evolutionary operators to satisfy the search requirements. Firstly, a fitness hierarchical mutation (FHM) strategy is proposed that divide the individuals into different levels based on individuals' fitness quality, so that they can use different mutation strategies to avoid the weakness of single mutation strategy. Secondly, by considering the angle of evolution stage, a directed global search (DGS) strategy and an elite local search (ELS) strategy are proposed in the late evolution stage to help enhance the population diversity for locating more peaks and to help refine the accuracy of the good solutions found, respectively. This way, a multi-angle hierarchical DE (MaHDE) is proposed, whose main differences with the common method in solving MMOPs is shown in Fig. 1. The frameworks of the common method and the MaHDE to solve MMOPs are shown as Fig. 1(a) and Fig. 1(b), respectively. The novelties and advantages of the proposed MaHDE algorithm are described as the following three aspects.

- 1) The FHM strategy is proposed to balance the exploration and exploitation ability of different individuals. To achieve these targets, FHM divides the individuals into two levels (i.e., low/high-level) according to their fitness quality in the current niche. The low-level individuals are guided by the best individual of a niche to move towards its nearest peak. In addition, the high-level individuals are guided by themselves to maintain their superiority. Moreover, the neighbors' perturbation



**FIGURE 1. The difference between the common method and MaHDE in solving MMOPs. (a) The framework of the common method to solve MMOPs. (b) The framework of MaHDE to solve MMOPs.**

and the global perturbation are added to the evolution of low-level and high-level individuals, respectively, thus helping the algorithm to avoid local peaks.

- 2) The DGS strategy is designed for the low-level individuals in the late evolution stage to increase the population diversity and provide the new chance to re-search for the new global peaks. By expanding the search range of the low-level individuals, the DGS strategy helps them jump out of the local peaks and further to improve the population diversity.
- 3) The ELS strategy is proposed to refine the solution accuracy and save the fitness evaluations. Concretely, the ELS strategy is only performed on the high-level individuals in the current niche, which works via Gauss perturbation for its narrow sampling space. In this way, the ELS not only can refine the solution accuracy, but also can save the number of fitness evaluations.

The rest of this paper is organized as follows. Section II reviews the DE algorithm and the related works of MMOPs. Section III presents the proposed MaHDE algorithm. The experimental study is shown in Section IV. Finally, the conclusions are given in Section V.

## II. RELATED WORKS

### A. DE

DE was first proposed by Storn and Price [36], which works through mutation, crossover, and selection strategies to imitate the process of biological evolution. Due to its simple and efficient performance, DE is adopted by many researches to solve some complex problems [25], [37]–[42].

#### 1) MUTATION

DE is to start from a randomly generated initial population, use the difference vector of two randomly selected individuals (i.e.,  $\mathbf{x}_{g,r2}$ ,  $\mathbf{x}_{g,r3}$ ) as the source of a third individual

(i.e.,  $\mathbf{x}_{g_{r1}}$ ), then a mutation vector ( $\mathbf{v}_i$ ) is produced by weighting the difference vector and  $\mathbf{x}_{g_{r1}}$ . This operation is called mutation. The common used mutation strategies as (1) and (2), which are named DE/rand and DE/best, respectively.

$$\mathbf{v}_i = \mathbf{x}_{r1} + F \times (\mathbf{x}_{r2} - \mathbf{x}_{r3}) \quad (1)$$

$$\mathbf{v}_i = \mathbf{x}_{best} + F \times (\mathbf{x}_{r1} - \mathbf{x}_{r2}) \quad (2)$$

where  $i = 1, 2, \dots, NP$ ,  $NP$  is the population size.  $\mathbf{x}_{g_{r1}}, \mathbf{x}_{g_{r2}}, \mathbf{x}_{g_{r3}}$  are randomly selected from the population, and  $\mathbf{x}_{g_{r1}} \neq \mathbf{x}_{g_{r2}} \neq \mathbf{x}_{g_{r3}} \neq i$ .  $\mathbf{x}_{best}$  is the best individual in the current generation.  $F$  is often referred to as the scaling factor, which is used to control the weights of the difference vectors.

## 2) CROSSOVER

The new individual  $\mathbf{u}_i$  is generated by using a binomial crossover operation on the individuals  $\mathbf{x}_i$  and  $\mathbf{v}_i$  as

$$u_{ij} = \begin{cases} v_{ij} & \text{if } (\text{rand}_j(0, 1) \leq CR) \text{ or } (j = j_{rand}) \\ x_{ij} & \text{otherwise} \end{cases} \quad (3)$$

where  $j = 1, 2, \dots, D$ ,  $D$  is the dimension of search space.  $j_{rand}$  is a randomly selected number from  $\{1, 2, \dots, D\}$ .  $\text{rand}_j(0, 1)$  is a random number between 0 and 1, which is different in each dimension.  $CR$  is the crossover probability,  $j = j_{rand}$  can ensure that at least one dimension comes from  $v_{ij}$ .

## 3) SELECTION

If the fitness value of  $\mathbf{u}_i$  (i.e.,  $f(\mathbf{u}_i)$ ) is better than that of  $\mathbf{x}_i$ , the  $\mathbf{u}_i$  is replaced by  $\mathbf{x}_i$  in the next generation, otherwise  $\mathbf{x}_i$  remains, and the process is called selection as (4) for a maximization problem.

$$\mathbf{x}_i = \begin{cases} \mathbf{u}_i, & \text{if } f(\mathbf{u}_i) \geq f(\mathbf{x}_i) \\ \mathbf{x}_i, & \text{otherwise} \end{cases} \quad (4)$$

Recently, another efficient selection method has been widely used in many researches, where  $\mathbf{u}_i$  is compared to its nearest individual  $\mathbf{x}_p$  in the parental generation. If  $f(\mathbf{u}_i)$  is better than that of  $\mathbf{x}_p$ , then  $\mathbf{x}_p$  is replaced by  $\mathbf{u}_i$  and enter into the next generation. This selection process is shown as (5) and it is adopted in this paper.

$$\mathbf{x}_i = \begin{cases} \mathbf{u}_i, & \text{if } f(\mathbf{u}_i) \geq f(\mathbf{x}_p) \\ \mathbf{x}_p, & \text{otherwise} \end{cases} \quad (5)$$

## B. RELATED WORKS ON MMOPs

There are many researches on MMOPs. To better review these researches, we divide them into three categories according to the different base algorithms. They are MMOPs based on DE, MMOPs based on PSO, and MMOPs based on GA, respectively.

### 1) DE FOR MMOPs

The crowding DE (CDE) [19] and speciation DE (SDE) [20] are two typical methods used to solve MMOPs. CDE works by comparing the fitness of  $\mathbf{u}_i$  (generated by crossover)

with the nearest individual in parental generation. Then the better individual is selected to enter the next generation. SDE locates multiple peaks by dividing the population into multiple independent sub-populations. However, these two methods all need to employ extra parameters (i.e., the crowding size in CDE and the speciation radius in SDE), which are sensitive to the performance of algorithm. To reduce the influence of parameters on the algorithm, Qu *et al.* [21] designed a neighborhood mutation strategy by borrowing the neighborhoods information, resulting in a neighborhood CDE (NCDE) and a neighborhood SDE (NSDE). Subsequently, Gao *et al.* [22] utilized the clustering technique to divide the population into several subpopulations, and combing the self-adaptive parameter control technique to deal with MMOPs, resulting in self-CCDE and Self-CSDE. By designing the new mutation strategies, Biswas *et al.* [23] proposed an information sharing mechanism based on CDE and SDE, termed as LoICDE and LoISDE. Meanwhile, a parent-centric normalized mutation strategy also was designed by Biswas *et al.* [24], resulting in PNPcDE.

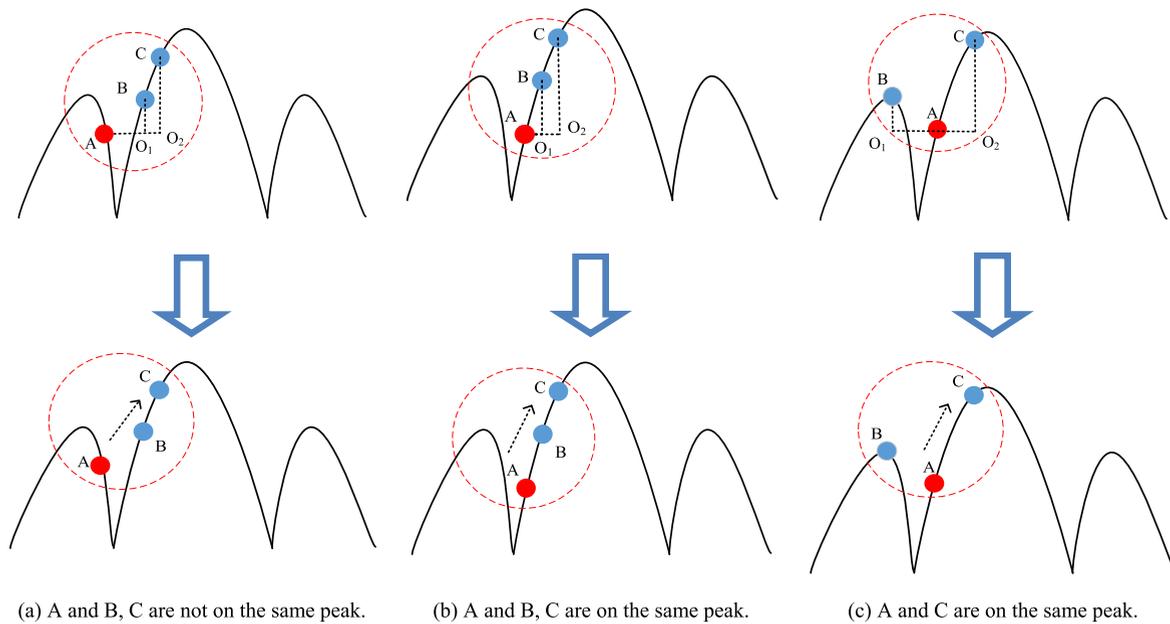
### 2) PSO FOR MMOPs

To solve the poor local search ability of the PSO-based niching algorithms, Li [29] designed a ring neighborhood topology based on PSO, which termed as r2pso and r3pso. Subsequently, a distance-based locally informed particle swarm (LIPS) optimizer was proposed by Qu *et al.* [30], which formed a stable niching by using the neighborhood information to solve the niching parameter sensitivity problem algorithm in MMOPs. Parsopoulos *et al.* [43] proposed a sequential PSO niching technique by using the objective function stretching to solve MMOPs. Brits *et al.* [44] adopted an initial swarm to produce the multiple sub-swarms by monitoring the fitness of particles. The number of particles in a sub-swarm can change by absorbing particles from the main swarm. Ren *et al.* [45] proposed a scatter learning PSO algorithm for MMOPs, which contracted an exemplar pool by collecting the high-quality solutions in the solution space.

### 3) GA FOR MMOPs

Many researches used GA to solve MMOPs as reviewed in [46]. Li *et al.* [31] divided the population into several species according to their similarity, and formed a species conservation technique for evolving parallel subpopulations to solve MMOPs. Gan and Warwick [32] proposed a dynamic niche clustering technique-based fitness sharing to solve MMOPs. Petrowski [33] proposed a new method that sharing the resources within subpopulations of individuals characterized by some similarities, and combined the best individuals of each subpopulation with GA to solve MMOPs. Bandaru and Deb [47] combined the power of dominance with traditional variable-space niching to solve MMOPs, which is implemented within the NSGA-II framework, and termed as ANSGAII.

Recently, Lin *et al.* [48] designed a novel algorithm that focus on the formulation, balance, and keypoint of species



**FIGURE 2.** The different distributions of individuals in a niche.

to balance exploration and exploitation in generating offspring. Wang *et al.* [49] proposed an automatic niching DE, in which the affinity propagation clustering that did not need sensitive clustering parameter was used to divide the population. Zhao *et al.* [50] borrowed the local binary operator idea in image processing to help efficiently form the niches. Chen *et al.* [51] designed a distributed individual DE that treated each individual as a distributed niche to track peaks. These algorithms all use DE as the base algorithm to solve MMOPs and achieve a great success. Therefore, this paper proposes a new algorithm based on DE, which designs different strategies to accommodate individuals of different fitness levels and search requirement in different evolution stages for efficiently solving MMOPs.

### III. MaHDE

In this section, the motivations of MaHDE are first introduced. Then, the FHM strategy is proposed to balance the exploration and exploitation ability of different individuals. Furthermore, the DGS strategy is proposed in the late evolution stage to increase the population diversity and provide the new opportunity for the low-level individuals to re-search for the global peaks. In addition, the ELS strategy is introduced to refine the solution accuracy of the high-level individuals. Last but not least, the completed MaHDE algorithm is given. The complexity analysis of MaHDE is given finally.

#### A. MOTIVATIONS

Since MMOPs have multiple peaks, it is most efficient method to partition the population into several overlapping or independent niches (subpopulations), and each niche targets to locate a peak. In this paper, the overlapping niches are

adopted to locate more peaks. Here, how to uses the individual's information of a niche (e.g., the fitness of individuals, the distance between individuals) has a key effective to enhance the performance of algorithm. From this perspective, there are several interesting observations.

Fig. 2 shows some different situations of individuals' distribution in a niche. For ease of description, we define the dotted circle with red represents a niche, the solid circles (e.g. A, B, and C) represent the individuals in a niche, and the solid circle with red (e.g., A) represents the current individual. Fig. 2(a) shows the diagram of whether to guide A by B or C. Here, by calculating the vertical distance from B, C to A (i.e.,  $BO_1$ ,  $CO_2$ ), we can obtain that  $CO_2$  is larger than  $BO_1$ . Besides, we can also find that  $f(C)$  (i.e., the fitness of C) is better than  $f(B)$  by calculating the fitness of B, C. This shows that C is the better individual within the niche. Therefore, A is guided by C to pursue the global peaks quickly. Similarly, when A, B, and C are on the same peak as Fig. 2(b), the vertical distance from B, C to A (i.e.,  $BO_1$ ,  $CO_2$ ) can be calculated and we can obtain that  $CO_2$  is larger than  $BO_1$ . Moreover,  $f(C)$  is better than  $f(B)$  by calculating the fitness of them. Therefore, A is also guided by C in this situation to improve the search ability. Fig. 2(c) shows that A and C are on the same peak, but A and B are on different peaks. From Fig. 2(c) we can find that  $CO_2$  is larger than  $BO_1$  and  $f(C)$  is better than  $f(B)$ . Therefore, A is also guided by C in this situation to avoid local peaks.

From the above three situations, we can find that when  $f(A)$  is lower than the mean fitness ( $\bar{f}$ ) of the current niche, using the better individuals (i.e., the larger vertical distance) to guide A can help to locate the global peaks quickly and avoid local peaks. Motivated by these ideas, the individuals

of a niche can be divided into two groups according to their fitness. Here,  $\bar{f}$  is used as a boundary, and we can obtain the individuals worse than  $\bar{f}$  and marked as the low-level individuals. Meanwhile, the individuals better than  $\bar{f}$  are marked as the high-level individuals. The best individual of the niche is used as a leader to guide the low-level individuals, this way can make these individuals move towards the promising direction and further to locate the nearest peak. On the contrary, the high-level individuals are guided by themselves to maintain their superiority in the niche. Besides, the neighbors or global perturbation can be added to the above evolution to avoid local peaks, respectively. Therefore, an FHM strategy is proposed to make the low-level individuals move towards the nearest peak and make the high-level individuals maintain the global peaks.

Besides, with the population evolving, most of the individuals converge to the local or global peaks, leading to the loss of population diversity. Moreover, in the late evolution stage, the low-level individuals may trap into local peaks due to its poor fitness, and the high-level individuals may close to the global peaks. Therefore, we consider performing the DGL strategy on the low-level individuals to improve the population diversity and help these low-level individuals to obtain the chance of re-search for the global peaks. Besides, the ELS strategy is performed on the high-level individuals to refine their accuracy.

To sum up, the FHM strategy is proposed based on the fitness hierarchical technique, the DGL strategy and ELS strategy are performed on the low-level individuals and high-level individuals in the late evolution stage, respectively. Therefore, a multi-angle (i.e., fitness quality and evolutionary stage) hierarchical differential evolution (MaHDE) is proposed for MMOPs in this paper.

## B. FHM STRATEGY

Generally speaking, the different individuals of a niche adopt the single mutation strategy (e.g. DE/rand, DE/best). However, the DE/rand strategy decreases the convergence speed due to the random searching directions, and the DE/best strategy is easy to trap into local peaks for the homogeneous guidance direction. To avoid this situation, the FHM strategy can provide adaptive guide individual evolution, which divides the individuals of a niche into two levels (i.e., low/high level) according to their fitness quality. On one hand, the mean fitness ( $f_1$ ) of the niche that formed by the current individual  $x_i$  is calculated to measure the individuals' level. If the fitness of  $x_i$  (i.e.,  $f(x_i)$ ) is worse than or equal to  $\bar{f}_1$  (i.e.,  $f(x_i) \leq \bar{f}_1$ ), the  $x_i$  is marked as a low-level individual. To accelerate the convergence speed, the best individual in a niche ( $x_{nbest}$ ) is used to guide  $x_i$  to quickly locate to the global optimum that nearest to  $x_i$ . Besides, the neighbors' perturbation is added to the evolutionary process to avoid local peaks. On the other hand, if  $f(x_i)$  is better than  $\bar{f}_1$  (i.e.,  $f(x_i) > \bar{f}_1$ ), it means that  $x_i$  is a promising solution in this niche and is marked as a high-level individual. To maintain the superiority of the high-level individual in the niche, we can use itself as the

base for the mutation. Besides, the global perturbation is added to evolution to avoid local peaks. By combing these two-level mutation strategies, the FHM strategy not only can accelerate the convergence speed but also can maintain the found solutions. The detailed process of the FHM strategy is as follows.

- i) Find the nearest  $K$  individuals to the current individual  $x_i$  to form the niche of  $x_i$  and stores them in  $S$ .
- ii) Calculate the mean fitness ( $\bar{f}_1$ ) of these  $K$  individuals (not include  $x_i$  itself) as

$$\bar{f}_1 = \frac{\sum_{x_j \in S} f(x_j)}{|S|} \quad (6)$$

- iii) Select the corresponding mutation strategy according to  $f(x_i)$  as

$$v_i = \begin{cases} x_{nbest} + F \times (x_{n_{r1}} - x_{n_{r2}}), & \text{if } f(x_i) \leq \bar{f}_1 \\ x_i + F \times (x_{g_{r1}} - x_{g_{r2}}), & \text{otherwise} \end{cases} \quad (7)$$

where  $x_{nbest}$  is the best individual in the current niche.  $x_{n_{r1}}$  and  $x_{n_{r2}}$  are two individuals randomly selected from  $S$ , and  $x_{n_{r1}} \neq x_{n_{r2}} \in S$ . Besides,  $x_{g_{r1}}$  and  $x_{g_{r2}}$  are two individuals randomly selected from the population excluding  $S$ , and  $x_{g_{r1}} \neq x_{g_{r2}} \neq i \notin S$ .

## C. DGS STRATEGY

With population evolving, most of the individuals gradually converge to local or global peaks, leading to a loss of population diversity in the late evolution stage. This situation will cause some peaks not to be located due to the poor population diversity. It is necessary to increase the population diversity in the late evolution stage. Therefore, in the late evolution stage of MaHDE, we use two strategies as DGS and ELS for those low-level individuals and high-level individuals, respectively. Similar to the FHM strategy, we also use the mean fitness of all the individuals of the niche to divide the individuals. It should be noted that as the DGS/ELS is carried out after the FHM, crossover, and selection, as Fig. 1(b), the fitness values of the individuals in the niche have changed and we should re-calculate the mean fitness and denote it as  $\bar{f}_2$ .

Herein, the DGS is described for the individual  $x_i$  whose fitness is worse than  $f_2$  (i.e.,  $f(x_i) < \bar{f}_2$ ). Such an individual is regarded as low-level individual, which is not located the global peaks due to its poor fitness quality in the late evolution stage. Moreover, it may be trapped into local peaks due to the stagnation in late evolution stage. It will be meaningless to continue to evaluate the low-level individuals. Therefore, expanding the search range for these low-level individuals can help them jump out of the local peaks. Here, the DGS strategy is proposed as (8) to expand the search space of the low-level individual  $x_i$  in the late evolution stage. Here, we define when the population satisfies  $fe > \eta \times MaxFEs$  as the late evolution stage.

$$p\_x_i = x_i + \frac{f(x_{g_{r1}}) - f(x_{g_{r2}})}{f(x_{g_{best}}) - f(x_{g_{worst}})} \times (x_{g_{r1}} - x_{g_{r2}}) \quad (8)$$

where  $fe$  is the current number of fitness evaluations,  $MaxFEs$  denotes the maximum number of fitness evaluations.  $\eta$  is a parameter that controls the evolutionary stages of population. The impact of the value of  $\eta$  on the algorithm will be investigated in Section IV-E. Besides,  $\mathbf{x}_{g_{best}}$  and  $\mathbf{x}_{g_{worst}}$  are the best individual and the worst individual (measured by fitness) of the whole population in current generation.  $f(\mathbf{x}_{g_{best}})$  and  $f(\mathbf{x}_{g_{worst}})$  are the fitness of  $\mathbf{x}_{g_{best}}$  and  $\mathbf{x}_{g_{worst}}$ .  $\mathbf{x}_{g_{r1}}$  and  $\mathbf{x}_{g_{r2}}$  are two individuals selected from the population randomly excluding  $S$ , and  $\mathbf{x}_{g_{r1}} \neq \mathbf{x}_{g_{r2}} \neq i \notin S$ .

In (8), the second part controls the search direction. That is, if  $f(\mathbf{x}_{g_{r1}}) > f(\mathbf{x}_{g_{r2}})$ ,  $(\mathbf{x}_{g_{r1}} - \mathbf{x}_{g_{r2}})$  is the promising direction to exploit more optimal regions. Conversely, if  $f(\mathbf{x}_{g_{r1}}) < f(\mathbf{x}_{g_{r2}})$ ,  $(\mathbf{x}_{g_{r1}} + \mathbf{x}_{g_{r2}})$  is the promising direction to explore more global peaks. Therefore, this global search strategy can be considered as the directed. Note that only the new individual  $p\_x_i$  will enter the next generation when its fitness is better than the original individual  $\mathbf{x}_i$  (i.e.,  $f(p\_x_i) > f(\mathbf{x}_i)$ ). By this way, the DGS strategy can improve the population diversity. Moreover, the DGS strategy provides the low-level individuals with the opportunity to re-search for the global peaks.

#### D. ELS STRATEGY

To refine the solution accuracy, the local search strategy is a widely used method. However, if a local search is performed on each solution, it will result in a waste of fitness evaluations. Conversely, if a local search is performed only for the best individual, it may be useful to single optimization problems (i.e., only one global peaks solution can be located) and it will not works to MMOPs. Since the global peaks generally survive in around the better individuals in the late evolution stage, we consider that the local search strategy is executed only for the high-level individuals in the late evolution stage. Note that by comparing the fitness of current individual  $f(\mathbf{x}_i)$  with the mean fitness ( $\bar{f}_2$ ) of the niche formed by the current individual  $\mathbf{x}_i$ , we can obtain the high-level individuals (i.e.,  $f(\mathbf{x}_i) > \bar{f}_2$ ). Here, the ELS strategy is proposed as (9) to refine the accuracy of the high-level individuals in the late evolution stage.

$$q\_x_i = \mathbf{x}_i + (U_j - L_j) \times N(\mu, \sigma) \quad \text{if } f(\mathbf{x}_i) > \bar{f}_2 \quad (9)$$

where  $U_j$  and  $L_j$  are the upper boundary and lower boundary of the search space. Besides,  $N(\mu, \sigma)$  is the Gaussian distribution, here  $\mu = 0$ ,  $\sigma$  is set to  $1.0E-04$ . Here, it is note that the better individual is selected into the next generation by comparing  $f(\mathbf{x}_i)$  and  $f(q\_x_i)$ .

In (9), there are two advantages by borrowing the narrow sampling space of Gaussian distribution. On one hand, the search range can be controlled around the high-level individuals, avoiding the blind and ineffective search. On the other hand, selecting the better individual from  $\mathbf{x}_i$  and  $q\_x_i$  enter into the next generation ensure that the found solutions can be maintained in the whole evolutionary process. By this

way, the ELS strategy not only ensures the accuracy of the solutions to be refined but also avoids a waste of fitness evaluations in evolution.

#### E. COMPLETE MaHDE ALGORITHM

The MaHDE includes the FHM strategy, the DGS strategy, and the ELS strategy for dealing with MMOPs. Here, the advantages of FHM strategy are shown in the following three aspects: i) It can accelerate the convergence speed by guiding the low-level individuals move towards the nearest peak. ii) It can maintain the superiority of the high-level individuals by the autonomous guidance strategy. The DGS strategy is designed for the low-level individuals in the late evolution stage, which can improve the population diversity by expanding the search range. Besides, the ELS strategy is performed on the high-level individuals in the late evolution stage, which can help our MaHDE refine the accuracy of the solutions by the Gaussian distribution. The detailed process of MaHDE is shown in **Algorithm 1**.

The termination condition of the MaHDE algorithm is when achieves the maximum fitness evaluations ( $MaxFEs$ ). Moreover, if the algorithm has located all the global peaks, it also stops.

#### F. COMPLEXITY ANALYSIS

In the niching method for MaHDE, the computation complexity is  $O(NP^2 \times D)$  because of the calculations of pairwise distance between individuals, as obtained by line 4 in **Algorithm 1**. With the population evolution, the FHM, crossover and selection strategies are conducted generation by generation, and the MaHDE stops until achieves the  $MaxFEs$ . The complexity of this process is  $O(NP \times D)$ , as obtained by lines 5-8 in **Algorithm 1**. For the next steps, the DGS strategy (i.e., lines 13-18) and the ELS strategy (i.e., lines 19-25) are conducted on the low-level individuals and high-level individuals, respectively. The computation complexity of them is  $O(NP \times D)$  as obtained by lines 11-25 in **Algorithm 1**. Summarizing the above analysis, the computational complexity of MaHDE is  $O(NP^2 \times D)$ . It should be noted that the computational complexity of most state-of-the-art multimodal algorithms is  $O(NP^2 \times D)$  due to the sharing distance between each pair of individuals, which is same to our proposed MaHDE.

#### IV. EXPERIMENTAL STUDIES

In this section, a comprehensive experimental analysis and comparison verify the advantages of our MaHDE algorithm. Firstly, the test functions and experimental settings are introduced. Secondly, the results of MaHDE with some state-of-the-art algorithms are listed and analyzed. Thirdly, we compare MaHDE with two winners of CEC competitions in different accuracies. Then, the landscapes of the typical problems are listed to visualize the evolution process. Finally, the impacts of parameter settings are analyzed.

**Algorithm 1:** MaHDE

---

```

Begin
1: Randomly initialize the population with size  $NP$  and set
    $fe = 0$ ;
2: While  $fe < MaxFEs$ 
3:   For  $i = 1$  to  $NP$ 
4:     Find the nearest (measured by Euclidean distance)
      $K$  individuals of the current individual  $x_i$  and store
     them in  $S$ .
5:     Calculate the mean fitness of the individuals in  $S$  by
     (6);
6:     Perform the FHM strategy by (7);
7:     Perform the crossover strategy by (3);
8:     Perform the selection strategy by (5);
9:   End For
10:  If  $fe > \eta \times MaxFEs$ 
11:    For  $i = 1$  to  $NP$ 
12:      Calculate the mean fitness ( $\bar{f}_2$ ) of the current
      niche formed by  $x_i$ ;
13:      If  $f(x_i) < \bar{f}_2$  /*The DGS strategy*/
14:        Perform the DGS strategy by (8) and
        produce the new individual  $p\_x_i$ ;
15:        Evaluate the fitness of  $p\_x_i$ ;
16:         $fe++$ ;
17:        Compare the fitness of  $x_i$  and  $p\_x_i$  and select
        the better one to enter the next generation;
18:      Else
19:        /*The ELS strategy*/
20:        Calculate  $f(x_{gbest})$  and  $f(x_{gworst})$  in current
        generation
21:        Perform the ELS strategy by (9) and produce
         $q\_x_i$ ;
22:        Evaluate the fitness of  $q\_x_i$ ;
23:         $fe++$ ;
24:        Select the better one from  $x_i$  and  $q\_x_i$  to
        enter the next generation;
25:      End If
26:    End For
27:  End If
28:   $fe = fe + NP$ ;
29: End While

```

---

**End**

---

**A. TEST FUNCTIONS AND EXPERIMENTAL SETTINGS**

The 20 widely used multimodal functions from CEC'2013 are utilized to test the performance of MaHDE. The detailed features of these test functions can refer to [52]. In addition, two commonly used measures of MMOPs are adopted to measure the performance of MaHDE and different compared algorithms, which are peak ratio ( $PR$ ) and success rate ( $SR$ ), respectively. The definition of these measures also can refer to [52].

In this paper, four accuracy levels ( $\varepsilon$ ) that  $\varepsilon = 1E-01$ ,  $\varepsilon = 1E-02$ ,  $\varepsilon = 1E-03$ , and  $\varepsilon = 1E-04$  are adopted in the

experiments. The results of  $\varepsilon = 1E-04$  are mainly reported as [48]–[51]. Besides, the  $NP$ ,  $MaxFEs$ , and  $K$  of MaHDE adopt the same settings as [50].

The compared algorithms are divided into three groups according to different algorithms as reviewed in Section II-B. The first group uses DE as a base algorithm to deal with MMOPs (i.e., CDE, Self\_CCDE, PNPCCDE, LoICDE, NCDE, SDE, and NSDE). The second group uses PSO as a base to deal with MMOPs (i.e., r2pso, r3pso, and LIPS). The third group uses GA as a base to deal with MMOPs (i.e., ANSGAII). Here, the relevant parameters in the comparison algorithms are set the same as the settings in their original papers. Moreover, we ensure a fair comparison between the proposed method and the comparative methods by the following three settings. Firstly, the proposed algorithm and the compared algorithms are all tested on the same multimodal functions in CEC'2013 benchmark set. Secondly, the algorithm parameters in the compared algorithms are set as their original settings that have been fine turned by their authors. Thirdly, the  $NP$  and the  $MaxFEs$  in solving each function are the same for the proposed algorithm and the compared algorithms.

Note that all results in this paper are the mean values obtained from 51 independent runs of the algorithm. The experimental are conducted on a PC with 8 Intel Core i5 CPUs, 8 Gb memory and Windows10 with 64-bit system.

**B. COMPARISON WITH STATE-OF-THE-ART ALGORITHMS**

In order to comprehensively analyze the performance of MaHDE, we compare the  $PR$  and  $SR$  results of MaHDE with the compared state-of-the-art algorithms. The results of  $PR$  and  $SR$  on F1–F20 with the accuracy  $\varepsilon = 1.0E-04$  are shown in Table 1. The best  $PR$  values are **bolded** in all the algorithms. In addition, the Wilcoxon rank-sum test [53] is used to statistically evaluate the  $PR$  results of MaHDE and the compared algorithms in 51 runs, and the significance level is set as 0.05. The symbols “+”, “–”, and “ $\approx$ ” represent the MaHDE is significantly better, worse, and similar than the compared algorithms, respectively.

From Table 1, we can find that our MaHDE can locate all the global peaks on F1–F6 and F10 in each run. For F1–F5, the results of MaHDE are similar with CDE, Self\_CCDE, PNPCCDE, LoICDE, and NCDE. These algorithms can locate all the global peaks in each run except for CDE and LoICDE in F4. We can find that these algorithms are the improvements of CDE, which indicates that the principle of CDE has a good advantage for F1–F5. The r2pso and r3pso also perform better on F1–F5 except for F4. Because F4 is a nonlinear function, the global peaks are difficult to identify by the simple search strategy. For F6, only our MaHDE and LoICDE can locate all the global peaks in each run with all the compared algorithms.

MaHDE performs slightly worse than CDE, Self\_CCDE, PNPCCDE, and NCDE on F7. But our MaHDE can locate more than half of the global peaks on F7 (i.e.,  $PR$  is 0.804). Moreover, the  $PR$  result of MaHDE on F7 is better than

**TABLE 1.** The *PR* and *SR* results of MaHDE with the state-of-the-art algorithms on accuracy  $\epsilon = 1.0E-04$ .

Func	MaHDE		CDE		Self CCDE		PNPCDE		LoICDE		NCDE	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
F1	<b>1.000</b>	1.000	<b>1.000(≈)</b>	1.000								
F2	<b>1.000</b>	1.000	<b>1.000(≈)</b>	1.000								
F3	<b>1.000</b>	1.000	<b>1.000(≈)</b>	1.000								
F4	<b>1.000</b>	1.000	0.985(≈)	1.000	<b>1.000(≈)</b>	1.000	<b>1.000(≈)</b>	1.000	0.975(+)	0.902	<b>1.000(≈)</b>	1.000
F5	<b>1.000</b>	1.000	<b>1.000(≈)</b>	1.000								
F6	<b>1.000</b>	1.000	0.640(+)	1.000	0.942(+)	0.490	0.537(+)	0.000	<b>1.000(≈)</b>	1.000	0.305(+)	0.000
F7	0.804	0.000	0.861(-)	0.000	<b>0.884(-)</b>	0.020	0.874(-)	0.000	0.705(+)	0.020	0.873(-)	0.000
F8	0.983	0.291	0.000(+)	0.000	<b>0.994(-)</b>	0.882	0.000(+)	0.000	0.000(+)	0.000	0.002(+)	0.000
F9	0.351	0.000	0.474(-)	0.000	0.459(-)	0.000	<b>0.474(-)</b>	0.000	0.187(+)	0.000	0.461(-)	0.000
F10	<b>1.000</b>	1.000	<b>1.000(≈)</b>	1.000	<b>1.000(≈)</b>	1.000	<b>1.000(≈)</b>	1.000	<b>1.000(≈)</b>	1.000	0.988(+)	0.863
F11	0.725	0.078	0.330(+)	0.000	<b>0.778(-)</b>	0.137	0.667(+)	0.000	0.660(+)	0.000	<b>0.727(≈)</b>	0.059
F12	<b>0.650</b>	0.000	0.002(+)	0.000	0.422(+)	0.000	0.002(+)	0.000	0.495(+)	0.000	0.253(+)	0.000
F13	<b>0.667</b>	0.000	0.140(+)	0.000	0.653(+)	0.000	0.461(+)	0.000	0.510(+)	0.000	<b>0.667(≈)</b>	0.000
F14	<b>0.667</b>	0.000	0.024(+)	0.000	0.520(+)	0.000	0.258(+)	0.000	0.657(+)	0.000	<b>0.667(≈)</b>	0.000
F15	<b>0.648</b>	0.000	0.005(+)	0.000	0.343(+)	0.000	0.015(+)	0.000	0.299(+)	0.000	0.319(+)	0.000
F16	<b>0.667</b>	0.000	0.000(+)	0.000	0.655(+)	0.000	0.000(+)	0.000	0.556(+)	0.000	<b>0.667(≈)</b>	0.000
F17	<b>0.352</b>	0.000	0.000(+)	0.000	0.246(+)	0.000	0.000(+)	0.000	0.222(+)	0.000	0.250(+)	0.000
F18	<b>0.663</b>	0.000	0.167(+)	0.000	0.337(+)	0.000	0.150(+)	0.000	0.219(+)	0.000	0.500(+)	0.000
F19	<b>0.455</b>	0.000	0.000(+)	0.000	0.113(+)	0.000	0.000(+)	0.000	0.032(+)	0.000	0.148(+)	0.000
F20	<b>0.250</b>	0.000	0.000(+)	0.000	0.024(+)	0.000	0.000(+)	0.000	0.126(+)	0.000	0.150(+)	0.000
+ (MaHDE is better)			12		10		12		14		9	
- (MaHDE is worse)			2		4		2		0		2	
≈			6		6		6		6		9	

Func	SDE		NSDE		r2pso		r3pso		LIPS		ANSGAIL		
	PR	SR											
F1	0.657(+)	0.373	<b>1.000(≈)</b>	1.000	<b>1.000(≈)</b>	1.000	<b>1.000(≈)</b>	1.000	0.833(+)	0.686	0.930(+)	0.860	
F2	0.737(+)	0.529	0.776(+)	0.667	<b>1.000(≈)</b>	1.000	<b>1.000(≈)</b>	1.000	<b>1.000(≈)</b>	1.000	<b>1.000(≈)</b>	1.000	
F3	<b>1.000(≈)</b>	1.000	<b>1.000(≈)</b>	1.000	<b>1.000(≈)</b>	1.000	<b>1.000(≈)</b>	1.000	0.961(+)	0.961	<b>1.000(≈)</b>	1.000	
F4	0.284(+)	0.000	0.240(+)	0.000	0.670(+)	0.784	0.966(+)	0.863	0.990(+)	0.961	0.320(+)	0.000	
F5	0.922(+)	0.843	0.745(+)	0.490	<b>1.000(≈)</b>	1.000	<b>1.000(≈)</b>	1.000	<b>1.000(≈)</b>	1.000	0.900(+)	0.800	
F6	0.056(+)	0.000	0.056(+)	0.000	0.388(+)	0.000	0.687(+)	0.000	0.246(+)	0.000	0.001(+)	0.000	
F7	0.053(+)	0.000	0.053(+)	0.000	0.509(+)	0.000	0.434(+)	0.000	0.400(+)	0.000	0.509(+)	0.000	
F8	0.013(+)	0.000	0.013(+)	0.000	0.000(+)	0.000	0.421(+)	0.000	0.086(+)	0.000	0.000(+)	0.000	
F9	0.013(+)	0.000	0.006(+)	0.000	0.091(+)	0.000	0.127(+)	0.000	0.108(+)	0.000	0.140(+)	0.000	
F10	0.147(+)	0.000	0.098(+)	0.000	0.788(+)	0.000	0.850(+)	0.157	0.748(+)	0.000	0.953(+)	0.580	
F11	0.314(+)	0.000	0.248(+)	0.000	0.667(+)	0.000	0.650(+)	0.000	<b>0.974(-)</b>	0.843	0.033(+)	0.000	
F12	0.208(+)	0.000	0.135(+)	0.000	0.448(+)	0.000	0.537(+)	0.000	0.574(+)	0.000	0.010(+)	0.000	
F13	0.297(+)	0.000	0.225(+)	0.000	0.660(+)	0.000	0.647(+)	0.000	<b>0.794(-)</b>	0.176	0.000(+)	0.000	
F14	0.216(+)	0.000	0.190(+)	0.000	0.403(+)	0.000	0.637(+)	0.000	0.644(+)	0.000	0.000(+)	0.000	
F15	0.108(+)	0.000	0.125(+)	0.000	0.103(+)	0.000	0.213(+)	0.000	0.336(+)	0.000	0.000(+)	0.000	
F16	0.108(+)	0.000	0.170(+)	0.000	0.000(+)	0.000	0.431(+)	0.000	0.307(+)	0.000	0.000(+)	0.000	
F17	0.076(+)	0.000	0.108(+)	0.000	0.000(+)	0.000	0.096(+)	0.000	0.168(+)	0.000	0.000(+)	0.000	
F18	0.026(+)	0.000	0.163(+)	0.000	0.000(+)	0.000	0.100(+)	0.000	0.098(+)	0.000	0.000(+)	0.000	
F19	0.105(+)	0.000	0.098(+)	0.000	0.000(+)	0.000	0.032(+)	0.000	0.000(+)	0.000	0.000(+)	0.000	
F20	0.000(+)	0.000	0.123(+)	0.000	0.000(+)	0.000	0.078(+)	0.000	0.000(+)	0.000	0.000(+)	0.000	
+		19		18		16		16		16		18	
-		0		0		0		0		2		0	
≈		1		2		4		4		2		2	

‘+’, ‘-’, and ‘≈’ indicate that the results of the MaHDE are significantly better than, worse than, and similar to the compared algorithms by Wilcoxon’s rank sum test with  $\alpha=0.05$ .

SDE, NSDE, r2pso, r3pso, LIPS, and ANSGAIL. For F8, our MaHDE almost can locate all the global peaks due to its strong search ability. In addition, the *PR* result of MaHDE on F8 is better than all the compared algorithms except for Self\_CCDE. MaHDE performs slightly worse on F9, which may be because that F9 has many global peaks (i.e., 216) and local peaks. However, the *PR* result of F9 is still better than SDE, NSDE, r2pso, r3pso, LIPS, and ANSGAIL.

F11-F20 are the complex and composition functions, which are difficult to locate all the global peaks for the algorithms. The *PR* result of MaHDE on F11 is better than all the compared algorithms except for Self\_CCDE and NCDE. Here, our MaHDE performs competitive with Self\_CCDE and NCDE on F11, and they all locate most of the global

peaks (i.e., the *PR* results of these algorithms all achieve above 0.5). For F12, F15, and F17–F20, our MaHDE performs best on all the compared algorithms. In particular, F18–F20 are the problems of more than 10 dimensions, and the search spaces of them are complex that the other algorithms have difficult to locate the peaks. However, the MaHDE algorithm can still locate at least 2 global peaks in each run, which shows the superiority of our MaHDE algorithm on high-dimensional problems.

Based on the experimental results, the following analysis is obtained. Firstly, from the fitness quality perspective, the FHM strategy considers that individuals of different levels should play a different role to balance the exploration and exploitation ability of different individuals. By designing

**TABLE 2.** The PR and SR results of MaHDE, NEA2, and NMMSO on F1-F20 with accuracies of 1E-01, 1E-02, and 1E-03.

Func	1E-01						1E-02						1E-03					
	MaHDE		NEA2		NMMSO		MaHDE		NEA2		NMMSO		MaHDE		NEA2		NMMSO	
	PR	SR																
F1	<b>1.000</b>	1.000																
F2	<b>1.000</b>	1.000																
F3	<b>1.000</b>	1.000																
F4	<b>1.000</b>	1.000																
F5	<b>1.000</b>	1.000																
F6	<b>1.000</b>	1.000	0.963	0.480	0.998	0.960	<b>1.000</b>	1.000	0.963	0.480	0.994	0.900	<b>1.000</b>	1.000	0.958	0.440	0.992	0.880
F7	<b>1.000</b>	1.000	0.946	0.160	<b>1.000</b>	1.000	0.990	0.000	0.925	0.080	<b>1.000</b>	1.000	0.935	0.000	0.918	0.060	<b>1.000</b>	1.000
F8	<b>1.000</b>	1.000	0.241	0.000	0.954	0.060	<b>0.944</b>	0.000	0.240	0.000	0.939	0.040	<b>0.944</b>	0.000	0.240	0.000	0.922	0.020
F9	0.680	0.000	0.622	0.000	<b>0.978</b>	0.120	0.554	0.000	0.595	0.000	<b>0.978</b>	0.120	0.441	0.000	0.584	0.000	<b>0.978</b>	0.120
F10	<b>1.000</b>	1.000																
F11	<b>1.000</b>	1.000	<b>0.980</b>	0.880	0.990	0.940	<b>1.000</b>	1.000	0.967	0.800	0.990	0.940	<b>1.000</b>	1.000	0.967	0.800	0.990	0.940
F12	0.989	0.909	0.853	0.180	<b>0.995</b>	0.960	0.818	0.182	0.850	0.180	<b>0.995</b>	0.960	0.716	0.091	0.843	0.180	<b>0.995</b>	0.960
F13	0.939	0.818	0.977	0.860	<b>0.990</b>	0.940	0.766	0.000	0.970	0.820	<b>0.987</b>	0.920	0.667	0.000	0.960	0.760	<b>0.983</b>	0.900
F14	<b>1.000</b>	1.000	<b>0.830</b>	0.160	0.770	0.020	0.755	0.000	<b>0.817</b>	0.100	0.740	0.020	0.667	0.000	<b>0.810</b>	0.000	<b>0.723</b>	0.020
F15	<b>0.955</b>	0.818	0.743	0.020	0.650	0.000	<b>0.825</b>	0.000	0.723	0.000	0.674	0.000	0.594	0.000	0.720	0.000	0.642	0.000
F16	<b>1.000</b>	1.000	0.673	0.000	0.660	0.000	<b>0.816</b>	0.000	0.673	0.000	0.660	0.000	<b>0.677</b>	0.000	0.673	0.000	0.660	0.000
F17	<b>1.000</b>	1.000	0.695	0.000	0.480	0.000	<b>0.855</b>	0.000	0.695	0.000	0.477	0.000	<b>0.757</b>	0.000	0.695	0.000	0.470	0.000
F18	<b>1.000</b>	1.000	0.667	0.000	0.650	0.000	<b>0.836</b>	0.000	0.667	0.000	0.650	0.000	<b>0.706</b>	0.000	0.667	0.000	0.650	0.000
F19	<b>0.977</b>	0.909	0.667	0.000	0.460	0.000	<b>0.677</b>	0.000	<b>0.667</b>	0.000	0.460	0.000	<b>0.455</b>	0.000	<b>0.667</b>	0.000	0.457	0.000
F20	<b>0.511</b>	0.272	0.363	0.000	0.180	0.000	<b>0.450</b>	0.000	0.360	0.000	0.175	0.000	0.325	0.000	0.360	0.000	0.172	0.000
#Best	<b>17</b>		8		10		<b>15</b>		8		10		<b>13</b>		8		11	

different mutation strategies for the low/high-level individuals, FHM strategy helps the low-level individuals move towards the nearest peak by the guidance of the best individual in the current niche, and help the high-level individuals maintain their superiority by the autonomous guidance strategy. Secondly, from the evolution stage perspective, the DGS strategy is designed for the low-level individuals in the late evolution stage, which can improve the population diversity and provide the new opportunity for exploring more global peaks. Thirdly, the ELS strategy is designed for the high-level individuals, which borrows a narrow sampling space of Gaussian distribution to refine the accuracy of solutions in the late evolution stage. Therefore, our MaHDE obtains the promising results in dealing with MMOPs.

**C. COMPARISON WITH WINNERS OF CEC COMPETITIONS**

To further investigate the performance of MaHDE, we also compare MaHDE with the winners of CEC’2013 and CEC’2015, which are the niching CMA-ES via nearest-better clustering (NEA2) [54] and niching migratory multi-swarm optimizer (NMMSO) [55], respectively. Table 2 presents the PR and SR results of MaHDE, NEA2, and NMMSO on F1–F20 with accuracies of 1E–01, 1E–02, and 1E–03, respectively. The best results in Table 2 are marked as **bold**, and the last line counts the number of best results for each algorithm on F1-F20, which are marked as “#Best”.

From Table 2, we can find that our MaHDE performs best on most of the test functions compared with NEA2 and NMMSO with  $\epsilon = 1E-01, 1E-02,$  and  $1E-03$ . The “#Best” of MaHDE are 17, 15, and 13 when the accuracies are 1E–01, 1E–02, and 1E–03, respectively. It indicates the

better performance of our MaHDE on F1-F20 compared with NEA2 and NMMSO in different accuracies.

When  $\epsilon = 1E-01$ , the “#Best” of MaHDE, NEA2, and NMMSO are 17, 8, and 10, respectively. We can find that our MaHDE can locate all the global peaks in each run on F1–F8, F10, F11, F14, and F16–F18, that is the PR and SR results of MaHDE all are 1.000 on these problems (i.e.,  $PR = SR = 1.000$ ). The results of NMMSO are second to MaHDE, and NMMSO performs slightly poor on the complex problems (i.e., F14–F20). For  $\epsilon = 1E-02$  and  $\epsilon = 1E-03$ , our MaHDE still is a winner. This indicates that MaHDE has strong search ability for MMOPs in different accuracies.

**D. THE LANDSCAPES OF MaHDE ON THE SELECTED FUNCTIONS IN DIFFERENT GENERATIONS**

To better visualize the distribution of individuals in evolution, Fig. 3– Fig. 5 show the landscapes of MaHDE with different typical problems in different generations. Notes that “Gen” denotes the current generation, and Gen= 0 means the initial distribution of the population. It can be divided into three groups based on the characteristics of problems. The first group is the simple functions (i.e., F1 and F2), and the global peaks of them can be located in early stage of evolution as shown in Fig. 3. The second group is the problems that have many peaks (i.e., F6 and F7), locating to all peaks of them may exhaust a lot of fitness evaluations as shown in Fig. 4. The third group are more complex functions (i.e., F11 and F12), and the global peaks of them are difficult to locate with high accuracy as shown in Fig. 5.

From Fig. 3, we can find that our MaHDE can locate all the global peaks in the early evolution stage and until all individuals converges to the peaks. Fig. 4 shows the problem with

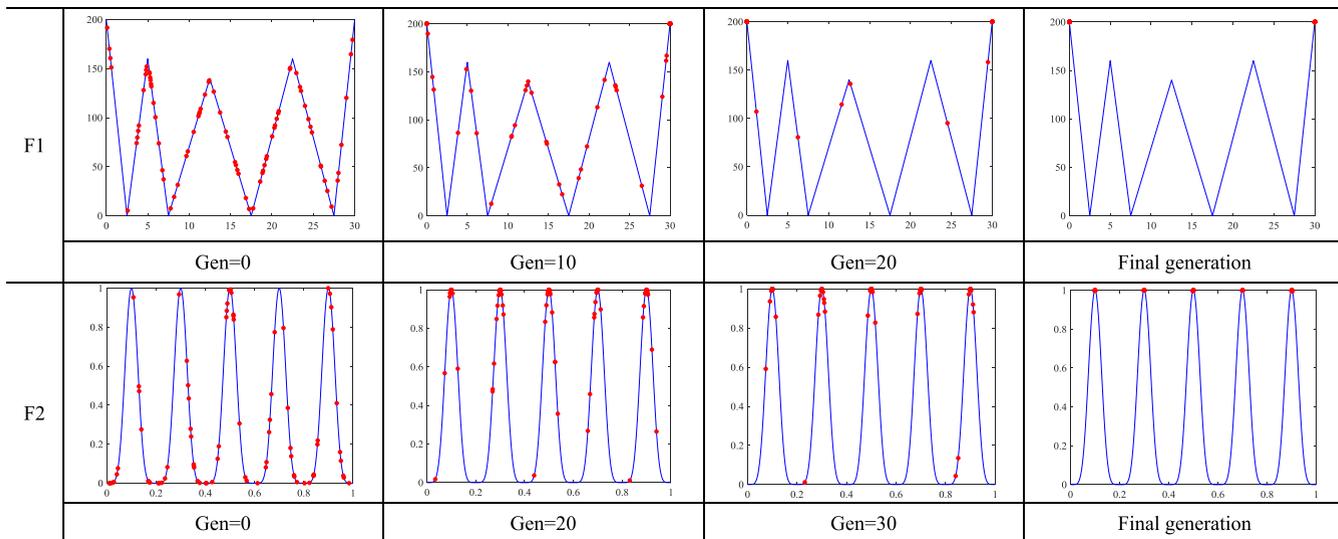


FIGURE 3. The landscapes of MaHDE on F1 and F2 in different generations.

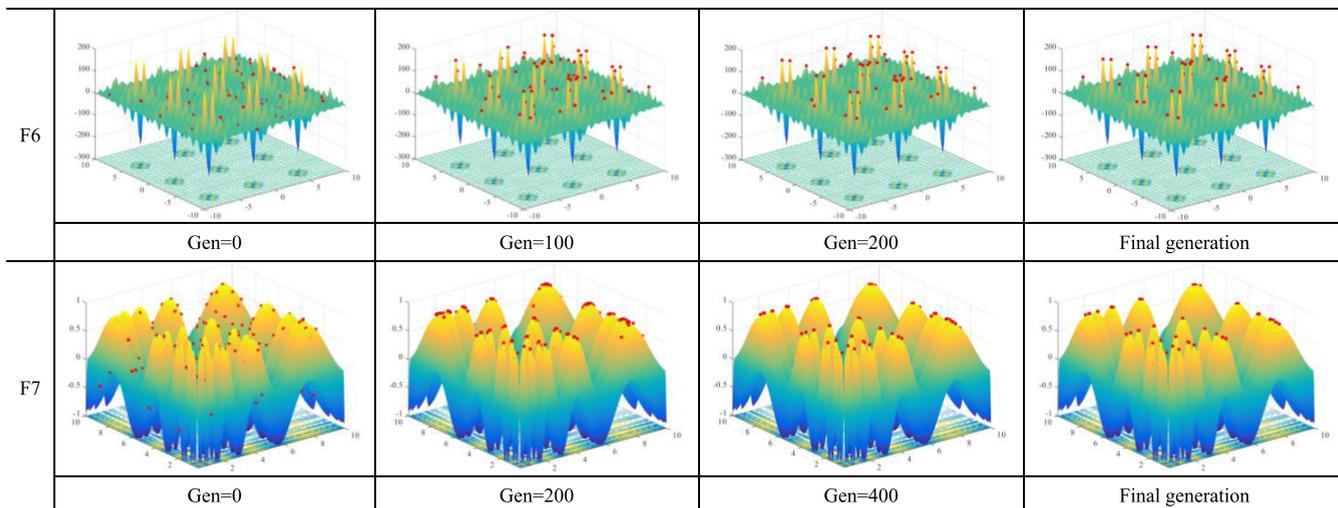


FIGURE 4. The landscapes of MaHDE on F6 and F7 in different generations.

more global peaks, which requires the algorithm to maintain the found solutions during the evolution. From Fig. 4, we can find that our MaHDE not only can locate all the peak regions but also can maintain the found solutions until the final generation. From Fig. 5, we can find that our MaHDE still can locate all the peaks regions on these complex problems. It indicates that our MaHDE not only has strong global search ability but also can avoid local peaks.

From Fig. 3–Fig. 5, we also can find that the different individuals can converge around different peaks with the increase of generation, which indicates that our MaHDE has a better convergence speed in the whole evolution.

E. IMPACTS OF PARAMETER SETTINGS

The parameter  $\eta$  controls the evolution stage of the population. In other words,  $\eta$  balances the algorithm ability of

exploration and exploitation. The small  $\eta$  will reduce the exploration ability of MaHDE, which causes some individuals to trap into local peaks. Conversely, the large  $\eta$  will reduce the exploitation ability of MaHDE, so some peaks cannot be located. Therefore, the value of  $\eta$  is important for the performance of MaHDE. Table 3 shows the PR results of MaHDE on F1–F20 with different  $\eta$ .

As we can see from Table 3, on the first 6 functions, different  $\eta$  values make nearly no difference in MaHDE on F1-F6, which can locate all the global peaks in each run except for  $\eta$  are 0.7 and 0.9. For the problems F7–F9, our MaHDE obtains the best results when  $\eta$  is 0.4. For F10, F13, F14, and F16, different  $\eta$  values also make nearly no difference in MaHDE. However, with the dimension of problem increase, the advantages of MaHDE become obvious when  $\eta$  is 0.4. Therefore, we set  $\eta$  is 0.4 in this paper.

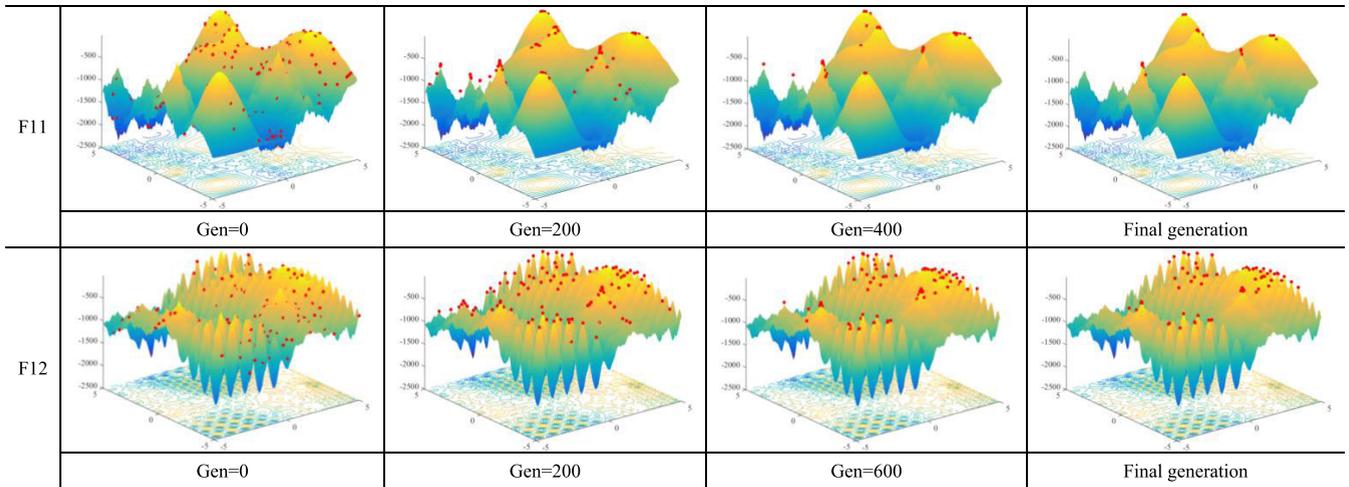


FIGURE 5. The landscapes of MaHDE on F11 and F12 in different generations.

TABLE 3. The PR results of MaHDE with different  $\eta$ .

Func	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
F1	1.000(≈)	1.000(≈)	1.000(≈)	1.000(≈)	1.000	1.000(≈)	1.000(≈)	1.000(≈)	1.000(≈)	1.000(≈)	1.000(≈)
F2	1.000(≈)	1.000(≈)	1.000(≈)	1.000(≈)	1.000	1.000(≈)	1.000(≈)	1.000(≈)	1.000(≈)	1.000(≈)	1.000(≈)
F3	1.000(≈)	1.000(≈)	1.000(≈)	1.000(≈)	1.000	1.000(≈)	1.000(≈)	1.000(≈)	1.000(≈)	1.000(≈)	1.000(≈)
F4	1.000(≈)	1.000(≈)	1.000(≈)	1.000(≈)	1.000	1.000(≈)	1.000(≈)	1.000(≈)	1.000(≈)	1.000(≈)	1.000(≈)
F5	1.000(≈)	1.000(≈)	1.000(≈)	1.000(≈)	1.000	1.000(≈)	1.000(≈)	1.000(≈)	1.000(≈)	1.000(≈)	1.000(≈)
F6	1.000(≈)	1.000(≈)	1.000(≈)	1.000(≈)	1.000	1.000(≈)	1.000(≈)	0.995(+)	1.000(≈)	0.995(+)	1.000(≈)
F7	0.755(+)	0.780(+)	0.753(+)	0.795(≈)	0.804	0.740(+)	0.755(+)	0.763(+)	0.758(+)	0.778(+)	0.763(+)
F8	0.961(+)	0.958(+)	0.952(+)	0.975(+)	0.983	0.975(+)	0.978(+)	0.972(+)	0.965(+)	0.961(+)	0.949(+)
F9	0.322(+)	0.325(+)	0.320(+)	0.322(+)	0.351	0.316(+)	0.330(+)	0.327(+)	0.334(+)	0.319(+)	0.316(+)
F10	1.000(≈)	1.000(≈)	1.000(≈)	1.000(≈)	1.000	1.000(≈)	1.000(≈)	1.000(≈)	1.000(≈)	1.000(≈)	1.000(≈)
F11	0.667(+)	0.682(+)	0.667(+)	0.667(+)	0.725	0.667(+)	0.682(+)	0.667(+)	0.667(+)	0.667(+)	0.667(+)
F12	0.635(+)	0.750(-)	0.727(-)	0.614(+)	0.650	0.670(-)	0.636(+)	0.545(+)	0.443(+)	0.318(+)	0.284(+)
F13	0.667(≈)	0.667(≈)	0.667(≈)	0.667(≈)	0.667	0.667(≈)	0.667(≈)	0.667(≈)	0.667(≈)	0.667(≈)	0.667(≈)
F14	0.667(≈)	0.667(≈)	0.667(≈)	0.667(≈)	0.667	0.667(≈)	0.667(≈)	0.667(≈)	0.667(≈)	0.667(≈)	0.667(≈)
F15	0.500(+)	0.636(+)	0.523(+)	0.557(+)	0.648	0.511(+)	0.545(+)	0.500(+)	0.432(+)	0.409(+)	0.398(+)
F16	0.606(+)	0.667(≈)	0.667(≈)	0.667(≈)	0.667	0.667(≈)	0.667(≈)	0.667(≈)	0.667(≈)	0.667(≈)	0.667(≈)
F17	0.159(+)	0.273(+)	0.295(+)	0.295(+)	0.352	0.273(+)	0.295(+)	0.295(+)	0.307(+)	0.295(+)	0.330(+)
F18	0.500(+)	0.652(+)	0.652(+)	0.652(+)	0.663	0.606(+)	0.621(+)	0.621(+)	0.621(+)	0.606(+)	0.636(+)
F19	0.148(+)	0.330(+)	0.386(+)	0.432(+)	0.455	0.443(+)	0.420(+)	0.420(+)	0.432(+)	0.420(+)	0.432(+)
F20	0.125(+)	0.125(+)	0.114(+)	0.125(+)	0.250	0.205(+)	0.239(+)	0.227(+)	0.227(+)	0.250(≈)	0.216(+)
+ ( $\eta=0.4$ is better)	11	9	9	10	NA	9	10	11	10	10	10
- ( $\eta=0.4$ is worse)	0	1	1	0	NA	1	0	0	0	0	0
≈	9	10	10	10	NA	10	10	9	10	10	10

‘+’, ‘-’, and ‘≈’ indicate that the PR results of MaHDE when  $\eta=0.4$  are significantly better than, worse than, and similar to the PR results of other  $\eta$  values on MaHDE by Wilcoxon’s rank sum test with  $\alpha=0.05$ .

To further investigate the advantages of the MaHDE when  $\eta = 0.4$ , we use the results of the MaHDE with  $\eta = 0.4$  as a base and perform the Wilcoxon rank-sum test with  $\alpha = 0.05$  on the different PR results of the other  $\eta$  values. Last three rows of Table 3 (i.e., ‘+’, ‘-’, and ‘≈’) show the PR results of  $\eta = 0.4$  are significantly better than, worse than, and similar to the PR results of other  $\eta$  values on MaHDE, respectively. From Table 3, we can find that the PR results of  $\eta = 0.4$  are

significantly better than all the results of the other  $\eta$  values, which shows the overall better  $\eta$  performance of MaHDE with  $\eta = 0.4$ .

### V. CONCLUSION

This paper proposes the MaHDE algorithm to better deal with MMOPs. Considering both fitness quality angle and evolution stage angle, our MaHDE not only can accelerate the

convergence speed and preserve the found solutions around peaks, but also can improve population diversity for locating more peaks and to help refine the accuracy of the found good solutions. Concretely, three novel strategies are designed to ensure the performance of the algorithm. Firstly, the FHM strategy considers that the different levels individual should play a different role to balance the exploration and exploitation ability of different individuals. By designing the different mutation strategies for the low/high-level individuals, it makes the low-level individuals move towards the nearest peak by the guidance of the best individual in the current niche. Besides, it makes the high-level individuals maintain their superiority by the autonomous guidance strategy. Secondly, the DGS strategy is designed for the low-level individuals in the late evolution stage, which can improve the population diversity and provide the new opportunity of re-search the global peaks for the low-level individuals. Thirdly, the ELS strategy is designed for the high-level individuals, which borrows a narrow sampling space of Gaussian distribution to refine the accuracy of solutions in the late evolution stage. Based on these novel strategies, MaHDE can achieve a promising performance when comparing with some state-of-the-art multimodal algorithms and two winners of the CEC competitions in different accuracy levels. The experimental results can show the superiority of the proposed MaHDE when dealing with MMOPs.

Although MaHDE shows promising performance in dealing with MMOPs, it still has some limitations. For example, its performance on some high-dimensional problems is still not good enough. In the future, we will extend the MaHDE to solve some MMOPs in some potential real-world applications, including resource-constrained project scheduling [56], electricity markets [57], energy resource management [58], and optical networks [59]. For example, how to detect multiple equilibriums simultaneously is a key challenging economic game problem in electricity markets, and our MaHDE algorithm can be used to solve these problems. Meanwhile, we can also further improve the performance of MaHDE in the high dimensional problems.

## REFERENCES

- [1] S. Das, S. Maity, B.-Y. Qu, and P. N. Suganthan, "Real-parameter evolutionary multimodal optimization—A survey of the state-of-the-art," *Swarm Evol. Comput.*, vol. 1, no. 2, pp. 71–88, Jun. 2011.
- [2] Y. Cao, H. Zhang, W. Li, M. Zhou, Y. Zhang, and W. A. Chaovaitwongse, "Comprehensive learning particle swarm optimization algorithm with local search for multimodal functions," *IEEE Trans. Evol. Comput.*, vol. 23, no. 4, pp. 718–731, Aug. 2019.
- [3] Z.-J. Wang, Z.-H. Zhan, Y. Lin, W.-J. Yu, H.-Q. Yuan, T.-L. Gu, S. Kwong, and J. Zhang, "Dual-strategy differential evolution with affinity propagation clustering for multimodal optimization problems," *IEEE Trans. Evol. Comput.*, vol. 22, no. 6, pp. 894–908, Dec. 2018.
- [4] Y.-H. Zhang, Y.-J. Gong, Y. Gao, H. Wang, and J. Zhang, "Parameter-free Voronoi neighborhood for evolutionary multimodal optimization," *IEEE Trans. Evol. Comput.*, vol. 24, no. 2, pp. 335–349, Apr. 2020.
- [5] X. Li, M. G. Epitropakis, K. Deb, and A. Engelbrecht, "Seeking multiple solutions: An updated survey on niching methods and their applications," *IEEE Trans. Evol. Comput.*, vol. 21, no. 4, pp. 518–538, Aug. 2017.
- [6] M. Preuss, P. Burelli, and G. N. Yannakakis, "Diversified virtual camera composition," in *Proc. Eur. Conf. Appl. Evol. Comput.*, 2012, pp. 265–274.
- [7] M. Kronfeld, A. Dräger, M. Aschoff, and A. Zell, "On the benefits of multimodal optimization for metabolic network modeling," in *Proc. GCB*, 2009, pp. 191–200.
- [8] O. M. Shir, C. Siedschlag, T. Bäck, and M. J. J. Vrakking, "Niching in evolution strategies and its application to laser pulse shaping," in *Proc. Int. Conf. Artif. Evol.*, 2005, pp. 85–96.
- [9] E. Pérez, M. Posada, and F. Herrera, "Analysis of new niching genetic algorithms for finding multiple solutions in the job shop scheduling," *J. Intell. Manuf.*, vol. 23, no. 3, pp. 341–356, Jun. 2012.
- [10] E. Pérez, M. Posada, and A. Lorenzana, "Taking advantage of solving the resource constrained multi-project scheduling problems using multimodal genetic algorithms," *Soft Comput.*, vol. 20, no. 5, pp. 1879–1896, May 2016.
- [11] C. Castillo, G. Nitschke, and A. Engelbrecht, "Niche particle swarm optimization for neural network ensembles," in *Proc. Eur. Conf. Artif. Life*, 2009, pp. 399–407.
- [12] X.-Y. Zhang, J. Zhang, Y.-J. Gong, Z.-H. Zhan, W.-N. Chen, and Y. Li, "Kuhn–Munkres parallel genetic algorithm for the set cover problem and its application to large-scale wireless sensor networks," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 695–710, Oct. 2016.
- [13] T. Friedrich, T. Kötzing, M. S. Krejca, and A. M. Sutton, "The compact genetic algorithm is efficient under extreme Gaussian noise," *IEEE Trans. Evol. Comput.*, vol. 21, no. 3, pp. 477–490, Jun. 2017.
- [14] Z. Wang, H. Xing, T. Li, Y. Yang, R. Qu, and Y. Pan, "A modified ant colony optimization algorithm for network coding resource minimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 3, pp. 325–342, Jun. 2016.
- [15] X.-F. Liu, Z.-H. Zhan, J. D. Deng, Y. Li, T. Gu, and J. Zhang, "An energy efficient ant colony system for virtual machine placement in cloud computing," *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 113–128, Feb. 2018.
- [16] X. Liang, H. Chen, and J. A. Lozano, "A Boltzmann-based estimation of distribution algorithm for a general resource scheduling model," *IEEE Trans. Evol. Comput.*, vol. 19, no. 6, pp. 793–806, Dec. 2015.
- [17] Z.-H. Zhan, J. Li, J. Cao, J. Zhang, H. S.-H. Chung, and Y.-H. Shi, "Multiple populations for multiple objectives: A coevolutionary technique for solving multiobjective optimization problems," *IEEE Trans. Cybern.*, vol. 43, no. 2, pp. 445–463, Apr. 2013.
- [18] N. R. Sabar, J. Abawajy, and J. Yearwood, "Heterogeneous cooperative co-evolution memetic differential evolution algorithm for big data optimization problems," *IEEE Trans. Evol. Comput.*, vol. 21, no. 2, pp. 315–327, Apr. 2017.
- [19] R. Thomsen, "Multimodal optimization using crowding-based differential evolution," in *Proc. Congr. Evol. Comput.*, Jun. 2004, pp. 1382–1389.
- [20] X. Li, "Efficient differential evolution using speciation for multimodal function optimization," in *Proc. Conf. Genetic Evol. Comput. (GECCO)*, 2005, pp. 873–880.
- [21] B. Y. Qu, P. N. Suganthan, and J. J. Liang, "Differential evolution with neighborhood mutation for multimodal optimization," *IEEE Trans. Evol. Comput.*, vol. 16, no. 5, pp. 601–614, Oct. 2012.
- [22] W. Gao, G. G. Yen, and S. Liu, "A cluster-based differential evolution with self-adaptive strategy for multimodal optimization," *IEEE Trans. Cybern.*, vol. 44, no. 8, pp. 1314–1327, Aug. 2014.
- [23] S. Biswas, S. Kundu, and S. Das, "Inducing niching behavior in differential evolution through local information sharing," *IEEE Trans. Evol. Comput.*, vol. 19, no. 2, pp. 246–263, Apr. 2015.
- [24] S. Biswas, S. Kundu, and S. Das, "An improved parent-centric mutation with normalized neighborhoods for inducing niching behavior in differential evolution," *IEEE Trans. Cybern.*, vol. 44, no. 10, pp. 1726–1737, Oct. 2014.
- [25] S. Hui and P. N. Suganthan, "Ensemble and arithmetic recombination-based speciation differential evolution for multimodal optimization," *IEEE Trans. Cybern.*, vol. 46, no. 1, pp. 64–74, Jan. 2016.
- [26] M. G. Epitropakis, X. Li, and E. K. Burke, "A dynamic archive niching differential evolution algorithm for multimodal optimization," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2013, pp. 79–86.
- [27] A. Basak, S. Das, and K. C. Tan, "Multimodal optimization using a biobjective differential evolution algorithm enhanced with mean distance-based selection," *IEEE Trans. Evol. Comput.*, vol. 17, no. 5, pp. 666–685, Oct. 2013.
- [28] Y. Wang, H.-X. Li, G. G. Yen, and W. Song, "MOMMOP: Multiobjective optimization for locating multiple optimal solutions of multimodal optimization problems," *IEEE Trans. Cybern.*, vol. 45, no. 4, pp. 830–843, Apr. 2015.

- [29] X. Li, "Niching without niching parameters: Particle swarm optimization using a ring topology," *IEEE Trans. Evol. Comput.*, vol. 14, no. 1, pp. 150–169, Feb. 2010.
- [30] B. Y. Qu, P. N. Suganthan, and S. Das, "A distance-based locally informed particle swarm model for multimodal optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 3, pp. 387–402, Jun. 2013.
- [31] J.-P. Li, M. E. Balazs, G. T. Parks, and P. J. Clarkson, "A species conserving genetic algorithm for multimodal function optimization," *Evol. Comput.*, vol. 10, no. 3, pp. 207–234, Sep. 2002.
- [32] J. Gan and K. Warwick, "Dynamic niche clustering: A fuzzy variable radius niching technique for multimodal optimisation in GAs," in *Proc. Congr. Evol. Comput.*, May 2001, pp. 215–222.
- [33] A. Petrowski, "A clearing procedure as a niching method for genetic algorithms," in *Proc. IEEE Int. Conf. Evol. Comput. (ICEC)*, May 1996, pp. 798–803.
- [34] B. L. Miller and M. J. Shaw, "Genetic algorithms with dynamic niche sharing for multimodal function optimization," in *Proc. IEEE Int. Conf. Evol. Comput.*, May 1987, pp. 41–49.
- [35] Q. Yang, W.-N. Chen, Z. Yu, T. Gu, Y. Li, H. Zhang, and J. Zhang, "Adaptive multimodal continuous ant colony optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 2, pp. 191–205, Apr. 2017.
- [36] R. Storn and K. Price, "Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.
- [37] Y.-L. Li, Z.-H. Zhan, Y.-J. Gong, W.-N. Chen, J. Zhang, and Y. Li, "Differential evolution with an evolution path: A DEEP evolutionary algorithm," *IEEE Trans. Cybern.*, vol. 45, no. 9, pp. 1798–1810, Sep. 2015.
- [38] Q. Fan and X. Yan, "Self-adaptive differential evolution algorithm with zoning evolution of control parameters and adaptive mutation strategies," *IEEE Trans. Cybern.*, vol. 46, no. 1, pp. 219–232, Jan. 2016.
- [39] G. W. Greenwood, "Using differential evolution for a subclass of graph theory problems," *IEEE Trans. Evol. Comput.*, vol. 13, no. 5, pp. 1190–1192, Oct. 2009.
- [40] J. Wang, G. Liang, and J. Zhang, "Cooperative differential evolution framework for constrained multiobjective optimization," *IEEE Trans. Cybern.*, vol. 49, no. 6, pp. 2060–2072, Jun. 2019.
- [41] L. Tang, X. Wang, and Z. Dong, "Adaptive multiobjective differential evolution with reference axis vicinity mechanism," *IEEE Trans. Cybern.*, vol. 49, no. 9, pp. 3571–3585, Sep. 2019.
- [42] J. Sun, S. Gao, H. Dai, J. Cheng, M. Zhou, and J. Wang, "Bi-objective elite differential evolution algorithm for multivalued logic networks," *IEEE Trans. Cybern.*, vol. 50, no. 1, pp. 233–246, Jan. 2020.
- [43] K. E. Parsopoulos, V. P. Plagianakos, G. D. Magoulas, and M. N. Vrahitis, "Stretching technique for obtaining global minimizers through particle swarm optimization," in *Proc. Part. Swarm Optimiz. Workshop*, 2001, pp. 22–29.
- [44] R. Brits, A. Engelbrecht, and F. van den Bergh, "A niching particle swarm optimizer," in *Proc. 4th Asia-Pacific Conf. SEAL*, 2002, pp. 692–696.
- [45] Z. Ren, A. Zhang, C. Wen, and Z. Feng, "A scatter learning particle swarm optimization algorithm for multimodal problems," *IEEE Trans. Cybern.*, vol. 44, no. 7, pp. 1127–1140, Jul. 2014.
- [46] N. N. Glibovets and N. M. Gulayeva, "A review of niching genetic algorithms for multimodal function optimization," *Cybern. Syst. Anal.*, vol. 49, no. 6, pp. 815–820, Nov. 2013.
- [47] S. Bandaru and K. Deb, "A parameterless-niching-assisted bi-objective approach to multimodal optimization," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2013, pp. 95–102.
- [48] X. Lin, W. Luo, and P. Xu, "Differential evolution for multimodal optimization with species by nearest-better clustering," *IEEE Trans. Cybern.*, early access, Apr. 22, 2019, doi: 10.1109/TCYB.2019.2907657.
- [49] Z.-J. Wang, Z.-H. Zhan, Y. Lin, W.-J. Yu, H. Wang, S. Kwong, and J. Zhang, "Automatic niching differential evolution with contour prediction approach for multimodal optimization problems," *IEEE Trans. Evol. Comput.*, vol. 24, no. 1, pp. 114–128, Feb. 2020.
- [50] H. Zhao, Z.-H. Zhan, Y. Lin, X. Chen, X.-N. Luo, J. Zhang, S. Kwong, and J. Zhang, "Local binary pattern-based adaptive differential evolution for multimodal optimization problems," *IEEE Trans. Cybern.*, vol. 50, no. 7, pp. 3343–3357, Jul. 2020.
- [51] Z.-G. Chen, Z.-H. Zhan, H. Wang, and J. Zhang, "Distributed individuals for multiple peaks: A novel differential evolution for multimodal optimization problems," *IEEE Trans. Evol. Comput.*, vol. 24, no. 4, pp. 708–719, Aug. 2020.
- [52] X. Li, A. Engelbrecht, and M. G. Epitropakis, "Benchmark functions for CEC'2013 special session and competition on niching methods for multimodal function optimization," *Evol. Comput. Mach. Learn. Group, RMIT Univ., Melbourne, VIC, Australia, Tech. Rep.*, 2013. [Online]. Available: <http://titan.csit.rmit.edu.au/~e46507/cec13-niching/competition/cec2013-niching-benchmark-tech-report.pdf>
- [53] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 3–18, Mar. 2011.
- [54] M. Preuss, "Niching the CMA-ES via nearest-better clustering," in *Proc. 12th Annu. Conf. Companion Genetic Evol. Comput. (GECCO)*, Portland, OR, USA, 2010, pp. 1711–1718.
- [55] J. E. Fieldsend, "Running up those hills: Multi-modal search with the niching migratory multi-swarm optimiser," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Beijing, China, Jul. 2014, pp. 2593–2600.
- [56] S. Elsayed, R. Sarker, T. Ray, and C. C. Coello, "Consolidated optimization algorithm for resource-constrained project scheduling problems," *Inf. Sci.*, vols. 418–419, pp. 346–362, Dec. 2017.
- [57] F. Zaman, S. M. Elsayed, T. Ray, and R. A. Sarker, "Evolutionary algorithms for finding Nash equilibria in electricity markets," *IEEE Trans. Evol. Comput.*, vol. 22, no. 4, pp. 536–549, Aug. 2018.
- [58] F. Lezama, L. E. Sucar, E. M. de Cote, J. Soares, and Z. Vale, "Differential evolution strategies for large-scale energy resource management in smart grids," in *Proc. Genetic Evol. Comput. Conf. Companion*, Berlin, Germany, Jul. 2017, pp. 1279–1286.
- [59] F. Callegati, L. H. Bonani, F. Lezama, W. Cerroni, A. Campi, and G. Castanon, "Trunk reservation for fair utilization in flexible optical networks," *IEEE Commun. Lett.*, vol. 18, no. 5, pp. 889–892, May 2014.



**ZHAO HONG** (Student Member, IEEE) received the bachelor's and master's degrees in computer science from the South China University of Technology, Guangzhou, China, in 2014 and 2017, respectively, where she is currently pursuing the Ph.D. degree.

Her current research interests include artificial intelligence, evolutionary computation, and swarm intelligence and their applications in design and optimization.



**ZONG-GAN CHEN** received the B.S. degree from Sun Yat-sen University, Guangzhou, China, in 2016. He is currently pursuing the Ph.D. degree in computer science and technology with the South China University of Technology, Guangzhou.

His current research interests include differential evolution, colony optimization, and the estimation of distribution algorithm and their applications in real-world optimization problems.



**DONG LIU** received the B.S. and M.S. degrees in computer science from Zhengzhou University, in 2004, and the Ph.D. degree in computer science from Tianjin University, in 2013.

He is currently an Associate Professor with Henan Normal University. His research interests include educational data mining and complex network analysis.



**ZHI-HUI ZHAN** (Senior Member, IEEE) received the bachelor's and Ph.D. degrees in computer science from Sun Yat-sen University, Guangzhou, China, in 2007 and 2013, respectively.

From 2013 to 2015, he was a Lecturer and an Associate Professor with the Department of Computer Science, Sun Yat-sen University. Since 2016, he has been a Professor with the School of Computer Science and Engineering, South China University of Technology, where he is also the

Changjiang Scholar Young Professor and the Pearl River Scholar Young Professor. His current research interests include evolutionary computation, swarm intelligence, and their applications in real-world problems, and in the environments of cloud computing and big data.

Dr. Zhan's doctoral dissertation was awarded the China Computer Federation (CCF) Outstanding Ph.D. Dissertation and the IEEE Computational Intelligence Society (CIS) Outstanding Ph.D. Dissertation. He was a recipient of the Outstanding Youth Science Foundation from the National Natural Science Foundations of China (NSFC), in 2018, and the Wu Wen Jun Artificial Intelligence Excellent Youth from the Chinese Association for Artificial Intelligence in 2017. He is listed as one of the Most Cited Chinese Researchers in Computer Science. He is currently an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, *Neurocomputing*, and the *International Journal of Swarm Intelligence Research*.



**JUN ZHANG** (Fellow, IEEE) received the Ph.D. degree in electrical engineering from the City University of Hong Kong, in 2002.

He is currently a Visiting Scholar with Hanyang University, Ansan, South Korea. His current research interests include computational intelligence, cloud computing, high performance computing, operations research, and power electronic circuits.

Dr. Zhang was a recipient of the Changjiang Chair Professor from the Ministry of Education, China, in 2013, the First-Grade Award in Natural Science Research from the Ministry of Education, China, in 2009, the China National Funds for Distinguished Young Scientists from the National Natural Science Foundation of China in 2011. He is currently an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, the IEEE TRANSACTIONS ON CYBERNETICS, and the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS.

• • •