Received April 15, 2019, accepted April 30, 2019, date of publication May 6, 2019, date of current version May 15, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2914963

Differential Evolution With Adaptive Guiding Mechanism Based on Heuristic Rules

YIQIAO CAI[1], CHI SHAO[1], YING ZHOU[2], SHUNKAI FU[1], HUIZHEN ZHANG[1], AND HUI TIAN[1]

[1]College of Computer Science and Technology, Huaqiao University, Xiamen 361021, China

[2]School of Computer Sciences, Shenzhen Institute of Information Technology, Shenzhen 518172, China

Corresponding author: Yiqiao Cai (yiqiao00@163.com)

This work was supported in part by the Natural Science Foundation of Fujian Province of China under Grant 2018J01091, Grant 2017J01111, and Grant 2015J01258, in part by the Natural Science Foundation of Guangdong Province of China under Grant 2018A0303130055, in part by the Promotion Program for Young and Middle-Aged Teacher in Science and Technology Research of Huaqiao University under Grant ZQN-PY410, and in part by the Opening Project of Guangdong Province Key Laboratory of Computational Science, Sun Yat-sen University.

ABSTRACT This paper proposes to resolve the limitation of differential evolution (DE) that the difference between the individuals in search behavior has not yet been utilized effectively for guiding the evolution of the population. An adaptive guiding mechanism (AGM) based on the heuristic rules is thus suggested to make possible, individual-dependent guidance. The AGM mainly comprises three stages: *construction*, *separation*, and *guidance*. In the *construction* stage, the elite leadership team (*ELT*) is established with an adaptive control scheme by using good information of the population. In the *separation* stage, the *ELT* is divided into distinct elite groups that are allocated to different individuals based on their search behaviors. In the *guidance* stage, the leader that is chosen from the respective elite group, as well as the promising directions extracted from the population, are used together to guide the search of each individual. By incorporating AGM into DE, a novel algorithm framework, named DE with AGM (DE-AGM), is proposed to enhance the performance of DE. As a general framework, DE-AGM can be easily and seamlessly applied to most DE variants. The experimental results on 58 benchmark functions have demonstrated the competitive performance of DE-AGM.

INDEX TERMS Differential evolution, adaptive guiding mechanism, heuristic rule, mutation operator, numerical optimization.

I. INTRODUCTION

Differential evolution (DE), developed by Storn and Price, is a simple and efficient evolutionary algorithm (EA) and swarm intelligent (SI) optimizer for solving global numerical optimization [1], [2]. Because of its attractive characteristics, DE has been widely explored and successfully applied to various real-world optimization problems in the fields of scientific and engineering [3]. In recent decades, DE has been extensively studied, which leads to various advanced DE variants that are proposed to further improve its performance [4]. In these DE variants, various promising mechanisms, such as adaptive or self-adaptive method [5], [6], [6]–[8], ensemble learning method [9]–[11], decentralized and/or parallel population scheme [12]–[15], and hybrid method [16]–[18], are introduced.

The associate editor coordinating the review of this manuscript and approving it for publication was Md Asaduzzaman.

Mutation, as the salient feature of DE, attracts lots of attention from the researchers. According to the earlier studies [9], [19], different mutation strategies show different search characteristics during the evolutionary process. For example, DE/rand/1 and DE/rand/2 are completely random in nature, in which the parent vectors involved in the mutation process are randomly selected from the current population. Due to the high degree of randomness, these strategies will prefer to the global exploration but may result in slow convergence [19], [20]. Conversely, both DE/best/1 and DE/current-to-best/1 involve the best individual of current population to generate the mutant vectors. It makes these strategies be good at local exploration but easily lead to premature convergence. Based on these considerations, various approaches have been proposed to enhance the search ability of the mutation operator for different complex problems, which roughly fall into the following categories: designing new mutation strategies [21], [22], integrating multiple mutation strategies [9], [10], [23], and selecting parent vectors for mutation [13]–[15], [24]. These works related to the mutation operator of DE will be reviewed in Section III.

In most DE variants, however, the difference between individuals in search behavior has not yet been effectively utilized for guiding the evolution of population. As shown in [14], [20], and [24]–[26], different individuals have distinct effects on the evolution of population. The individuals with better fitness values can guide the population towards the more promising regions, while the individuals with worse fitness values can explore new search space for keeping populations diversity [26]. Due to the different search roles, each individual will reveal distinguished search behavior during the process of evolution. To elaborate, the superior individuals in the exploitation behavior prefer to search the area surrounding it, and the inferior individuals in the exploration behavior are likely to explore over a larger region. Therefore, the guidance for different individuals ought to be designed elaborately and individually, which will make each individual be guided in a more effective way that matches with its search behavior.

To address the above issue, an adaptive guiding mechanism (AGM) based on heuristic rules is presented in this study. In AGM, there are three stages, i.e., *construction*, *separation*, and *guidance*, in each of which a heuristic rule is designed. In the first stage, a collection of individuals, named elite leadership team (*ELT*), is constructed by selecting best individuals from the sorted population. Further, an adaptive control scheme is embedded to dynamically adjust the size of *ELT* along with the process of evolution. In the following stage, different individual-dependent elite groups (*IEG*s) are separated from *ELT* and assigned to different individuals based on their distinct search behaviors. To elaborate, the individual with better fitness value will be allocated with an *IEG* of smaller size. In the final stage, the leader that guides the search of each individual is selected from the respective *IEG*. Moreover, to make the individuals evolve more efficiently, the promising direction information is extracted from population and incorporated into the mutation process. With these three stages, AGM can take full advantage of the difference between individuals in search behavior to guide the evolution of population.

To evaluate the performance of DE-AGM, the proposed framework is incorporated into several original and advanced DE algorithms. Extensive experiments are carried out on 58 benchmark functions from the special session on real-parameter optimization of CEC2013 [27] and CEC2017 [28]. Experimental results have demonstrated the advantages of DE-AGM when compared with other DE variants and EAs on the test functions.

In general, the main contributions of this study are as follows:

1) By considering the differences in search behaviors between individuals, an adaptive guiding mechanism (AGM) is proposed to achieve an individual-dependent guidance.
2) AGM, composed of construction, separation and guidance stages, is designed with the heuristic rules to assign different individuals with distinct elite groups based on their search behaviors. In addition, the elite groups are dynamically changed along with the evolutionary process by an adaptive control scheme strategy.
3) By incorporating AGM into DE, the proposed DE-AGM framework can provide an effective way to further improve the performance of DE with the information of population.

The rest of this paper is organized as follows. Sections II and III review the original DE algorithm and the work related to the approaches for enhancing the mutation strategy, respectively. The proposed DE-AGM is presented in detail in Section IV. In Section V, the experimental results are presented and discussed. Finally, the conclusions and future work are given in Section VI.

II. DIFFERENTIAL EVOLUTION (DE)

For an optimization problem (e.g., $Minimize f(X)$ in a D-dimensional space), DE starts with a population of NP individuals, and each individual at the $g$the generation is denoted as $X_{i,g} = [x_{i,g}^1, x_{i,g}^2, \ldots, x_{i,g}^D]$, $i = 1, 2, \ldots, NP$. For each individual $X_{i,g}$, the jth variable can be initialized as follows:

$$x_{i,g}^j = L_j + rndreal(0, 1) \cdot (U_j - L_j) \tag{1}$$

where $rndreal(0, 1)$ represents a uniformly distributed random number in the interval [0,1], and U_j and L_j represent the upper and lower bounds of the jth dimension, respectively. After the initialization, three main operators, i.e., mutation, crossover and selection, are carried out iteratively during the process of evolution.

A. MUTATION

In the mutation operator, each target vector $X_{i,g}$ uses the mutation strategy to generate a mutant vector $V_{i,g}$ by combining a base vector and difference vector(s). Several frequently used mutation strategies are shown as follows:

- DE/rand/1

$$V_{i,g} = X_{r1,g} + F \times (X_{r2,g} - X_{r3,g}) \tag{2}$$

- DE/best/1

$$V_{i,g} = X_{best,g} + F \times (X_{r1,g} - X_{r2,g}) \tag{3}$$

- DE/current-to-best/1 (or, DE/c-t-b/1 for short)

$$\begin{aligned} V_{i,g} = X_{i,g} &+ F \times (X_{best,g} - X_{i,g}) \\ &+ F \times (X_{r1,g} - X_{r2,g}) \end{aligned} \tag{4}$$

- DE/rand-to-best/1 (or, DE/r-t-b/1 for short)

$$\begin{aligned} V_{i,g} = X_{r1,g} &+ F \times (X_{best,g} - X_{r1,g}) \\ &+ F \times (X_{r2,g} - X_{r3,g}) \end{aligned} \tag{5}$$

where the indices $r1$, $r2$, $r3$, $r4$ and $r5 \in \{1, 2, \ldots, NP\} \setminus \{i\}$ are mutually different random indices, $X_{best,g}$ is the best individual at generation g, and F is the scaling factor that typically lies in the interval [0.4, 1]. During the mutation process, $v_{i,g}^{j}$ will be reinitialized with Eq. (1) when it is out of the boundary. More details of them can be found in [1] and [3].

B. CROSSOVER

After mutation, to enhance the diversity of population, a trial vector $U_{i,g}$ is generated by performing the crossover operator on each pair of $X_{i,g}$ and $V_{i,g}$. The classic binomial crossover operator is defined as follows:

$$u_{i,g}^{j} = \begin{cases} v_{i,g}^{j}, & \text{if } rndreal(0, 1) \leq Crorj = j_{rand} \\ x_{i,g}^{j}, & \text{otherwise.} \end{cases} \tag{6}$$

where $Cr \in [0, 1]$ is the crossover rate and $j_{rand} \in [1, D]$ is an integer that is randomly selected from the interval.

C. SELECTION

To decide whether $X_{i,g}$ or $U_{i,g}$ to survive in the next generation, DE uses a one-to-one selection operator based on their fitness values, which is performed as follows:

$$X_{i,g+1} = \begin{cases} U_{i,g}, & \text{if } f(U_{i,g}) \leq f(X_{i,g}) \\ X_{i,g}, & \text{otherwise.} \end{cases} \tag{7}$$

III. RELATED WORKS

As the salient feature of DE, the mutation operator has a great impact on the performance of DE. Therefore, a lot of studies from the worldwide researchers are devoting on improving the search ability of mutation operator for solving the complex problems [2]–[4]. In this paper, we focus on the work related to the enhanced mutation strategies in the following three aspects: developing new mutation strategy, integrating multiple mutation strategies, and selecting parents for mutation.

A. DEVELOPING NEW MUTATION STRATEGY

In view of that original DE mutation strategies are not competent to different complex problems, many new and sophisticated mutation strategies have been developed. In [5], a novel variant, JADE with ''DE/current-to-*p*best'' strategy, is proposed by using the information of multiple best solutions to generate the mutant vector, which can enhance the diversity of the population and thus alleviate the problem of trapping in local optimum. Similar to that, the best one among a pool of randomly selected solutions from the current population is used to guide the mutation process [29]. Due to the excellent and robust performance of JADE, a series of advanced DE variants is presented to further enhance its search ability, i.e., success-history based adaptive DE (SHADE) [6], SHADE with linear population size reduction (L-SHADE) [30], L-SHADE with a weighted variant of mutation strategy (jSO) [8], and parameter adaptive DE (PaDE) [31]. In DE with a trigonometric mutation operator (TDE) [21], the center point of the hypergeometric triangle is set as the base vector, and the mutation operator is carried out by perturbing the base vector with the sum of three weighted difference vectors. In gaussian bare-bones differential evolution (GBDE) [32], the gaussian sampling method is introduced into DE, and the vectors for the next generation are generated through a standard Gaussian distribution. In DE with a neighborhood-based mutation operator (DEGL) [33], the ring topology is employed to define the neighborhood of each vector in DE, and the mutant vector is generated by linearly combining the local and global vectors with the DE/current-to-best/1 strategy. In DE with neighborhood and direction information (NDi-DE) [19], the best and worst near-neighbor vectors of each individual are used to define three types of direction information, and the mutation strategies with different characteristics are equipped with different types of direction information to generate the mutant vectors. In the collective information-powered DE (CIPDE) [34], multiple top best solutions of current population are linearly combined to construct a collective vector, which is used as a terminal point of difference vector to generate the mutant vector. In DE with interactive information scheme (IIN-DE) [35], a new vector is constructed by an interactive information scheme, which is treated as the directional vector for generating the mutant vector.

B. INTEGRATING MULTIPLE MUTATION STRATEGIES

To utilize the advantages of the search characteristics of different mutation strategies, various attempts have been made by integrating multiple mutation strategies together into DE. In DE with strategy adaptation (SaDE) [9], four mutation strategies are chosen as the candidate strategies, and a probabilistic selection mechanism based on the success and failure historical experience is designed to adaptively select the suitable strategy for different target vectors to generate the mutant vectors. In composite DE (CoDE) [23], three mutation strategies are used to generate three trail vectors simultaneously, and then the best one will survive for the next generation if it defeats the target vector. In DE with an individual-dependent mechanism (IDE) [26], four types of mutation operators are designed for different sets of vectors (i.e., the superior and inferior sets) and different conditions in terms of the generation index. In DE with adaptive multiple sub-populations (MPADE) [22], three mutation strategies with different search preferences are employed respectively for the vectors in three sub-populations with different fitness values. In multi-population ensemble DE (MPEDE) [36], three different mutation strategies are used and evaluated with three smaller sub-populations, and then the best performing strategy will be applied to another larger sub-population to generate the mutant vectors. In the historical and heuristic DE (HHDE) [37], three mutation strategies are employed to construct a candidate pool, and then the cumulative probability based on the historical experience of each strategy and the heuristic value based on the fitness value of each vector are combined together to select the suitable strategies for different vectors. Besides, there are other DE variants that are proposed with the ensemble of mutation strategies, such as, DE with ensemble of parameters and mutation strategies (EPSDE) [38], DE with a multi-layer competitive-cooperative (MLCC) [39], Ensemble of DE variants (EDEV) [11], DE with strategy adaptation mechanisms (SaM-DE) [10], DE with self-adaptive control parameters based on zoning evolution (ZEPDE) [40], Self-adaptive DE with multiple strategies (XADE) [41], dual-strategy DE with affinity propagation clustering (DSDE-APC) [42], DE with underestimation-based multimutation strategy (DE-UMS) [43], and so on.

C. SELECTING PARENTS FOR MUTATION

In the mutation operator, the selection of parents for mutation greatly affects the performance of DE [44]. Thus, many researches focus on designing the novel mechanism for selecting parents involved in the mutation operator. In [15], several decentralized population topologies (e.g., cellular, ring, or small-world) are introduced into DE to define the neighborhood for each vector, and the parents for mutation are selected from the neighbors of the target vector. In DE with ranking-based mutation operators (rankDE) [25], each vector of current population is assigned a ranking value based on its fitness value, and the parents are probabilistically selected based on their ranking values. In DE with proximity-based mutation operators (proDE) [45], the Euclidean distance between each pair of vectors is calculated to evaluate their proximities, and then the parents for mutation are selected by a roulette wheel method based on their proximities with the target vector. In DE with multiobjective sorting based mutation operators (MSDE) [20], the multi-objective non-dominated sorting method is introduced into DE to calculate the probability of selection as the parents for all the vectors based on their fitness values and diversity indexes. In neighborhood guided DE (NGDE) [13], the ring topology is used to define the neighborhood of each vector, and the selection probability for each neighbor is calculated based on its fitness value. In neighborhood-adaptive DE (NaDE) [46], two index-based neighborhood topologies are used as the candidate topologies, and an adaptive operator selection method is employed to adaptively select the suitable topology for each vector based on the historical experience. With the chosen topology, the parents are selected from the neighborhood of target vector based on the partition mechanism. Following the similar idea, an individual-dependent adaptive selection strategy is designed in multi-topology-based DE (MTDE) [14] to adaptively select the topology for different target vectors. In addition, the external archive mechanism is used in DE with a successful-parent-selecting framework (SPS-DE) [47] and DE with guiding archive (GAR-DE) [48] to store the promising vectors generated during the evolutionary process, and then some parents for mutation are selected from the archive when the number of iterations that the target vector is not continuously promoted exceeds the default value.

IV. PROPOSED ALGORITHM

A. MOTIVATIONS

As shown in Section III, many researches put forward different approaches to enhance the search ability of the DE mutation strategy for complex problems. We have observed, however, that the guidance for different individuals in most DE variants has not yet been effectively distinguished based on their distinct search behaviors. For example, the fitness information of population has been used in [25] to guide the search of each individual. In this way, the selection probabilities of elites as leaders are identical for all the individuals. That is, all the individuals are allocated the same set of elites to guide the search. However, it is not an efficient way to utilize the difference between individuals in search behavior for guiding the search. Generally, the superior individuals are likely to search the optimal solutions in their surrounding areas. Thus, it is beneficial for them to have a smaller size of candidate leaders for exploitation. On the contrary, the inferior individuals tend to find the better solutions in the new search regions; thus, it is good for them to have a larger size of candidate leaders for exploration. Therefore, a more effective guiding mechanism is eagerly demanded by taking into account the difference in search behavior between individuals to further enhance the performance of DE.

To resolve the limitations of guiding mechanism in DE, an adaptive guiding mechanism (AGM) based on heuristic rules is proposed in this paper. By incorporating AGM into DE, the resultant algorithm, named DE with AGM (DE-AGM), is presented as a framework for further improving the performance of DE. In the following subsections, the details of AGM and DE-AGM will be given. First, the three stages of AGM, i.e., *construction*, *separation*, and *guidance*, together with the heuristic rules, will be elaborately described. Then, the complete framework of DE-AGM will be shown. Finally, some discussions on DE-AGM will be given.

B. ADAPTIVE GUIDING MECHANISM (AGM)

The AGM consists of three stages to guide the search of each individual based on its search behavior. Figure 1 illustrates the basic idea of the AGM. As shown in Figure 1, an elite leadership team (*ELT*) is constructed from the sorted population at the first stage of AGM. Then, AGM partitions the *ELT* into diverse overlapped elite groups for different individuals depending on their distinct search behaviors. Followed that, the individual-dependent elite group (*IEG*), together with the rest of population, are used to guide the search of each individual by selecting parents for mutation. The details of these three stages of AGM are described below.

1) STAGE 1: CONSTRUCTION

To make better use of the evolutionary information of population to guide the search, AGM firstly considers the following problem: which individuals are more beneficial for guiding the evolution of population? Here, the fitness information-based selection method [25] is used due to its

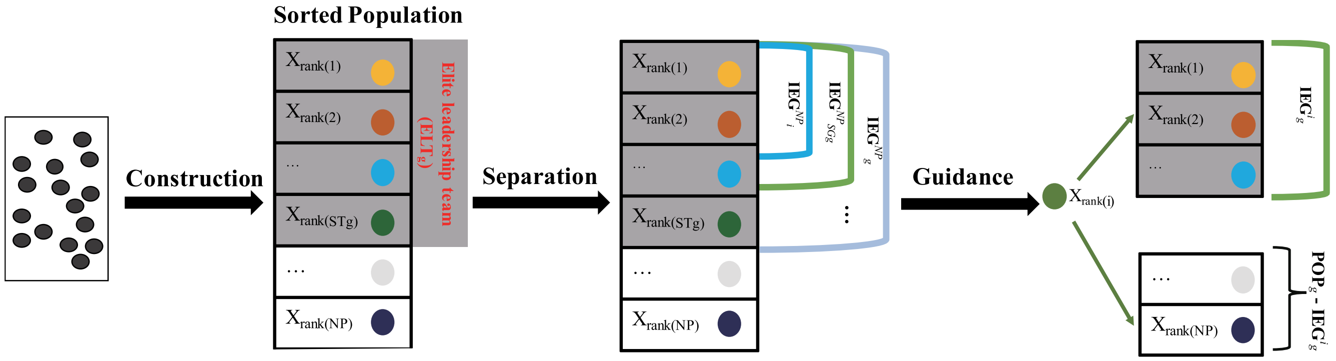

FIGURE 1. The basic idea of the adaptive guiding mechanism (AGM).

effectiveness and simplicity. Further, based on the previous study [49], the whole population is spread in the search space to explore different promising areas at the beginning of evolution, and then they will gradually gather together and converge to the globally or local optimal regions in the final evolutionary stages. Therefore, to utilize the evolution characteristics of population at different stages, an adaptive control scheme is embedded into the *construction* stage to dynamically adjust the size of *ELT* based on the number of iterations.

Overall, to construct *ELT* with an adaptive control scheme for current population, the *heuristic rule 1* (*HR*1) is presented, which is shown as follows:

Heuristic rule 1 (HR 1): Construction

$$ELT_g = \{X_{rank(i)} | i = 1, 2, \ldots, ST_g\} \tag{8}$$

$$ST_g = \lfloor NP \times \frac{1}{\lfloor r \times g/G_{max} \rfloor + 1} \rfloor \tag{9}$$

where $X_{rank(i)}$ is the *i*th individual in the sorted population based on the fitness values of population, g is the current generation, ST_g is the size of ELT_g, and r is a parameter to control the rate of reducing ST_g during the evolutionary process.

As defined in the *HR*1 (Eq.8 and Eq.9), the ELT_g contains the top ST_g best individuals (or elites) as the candidate leaders for guiding the evolution of population at the *g*th generation. Further, the adaptive control scheme is carried out through Eq.9, in which the ST_g gradually decreases as the g increases. To show how the ST_g value changes under the adaptive control scheme, the change curve of ST_g with $G_{max} = 3000$ and $r = 10$ is depicted in Figure 2. As Figure 2 shows, ST_g is equal to *NP* at the beginning of evolution and then gradually reduces along with the iterations.

2) STAGE 2: SEPARATION

During the evolutionary process, the individuals located in various search regions are usually with different search behaviors, resulting in distinct potential in exploring and exploiting the search space. To take effectively advantages of this feature to guide the search, AGM will focus on the following issue: how to separate the elite groups from the ELT_g for different individuals to match their search behaviors? As discussed in Section IV-A, the superior individuals in exploitation behavior attempt to search the areas with the elites. Relatively, the inferior individuals in exploration behavior are tend to perform random search in the large region.

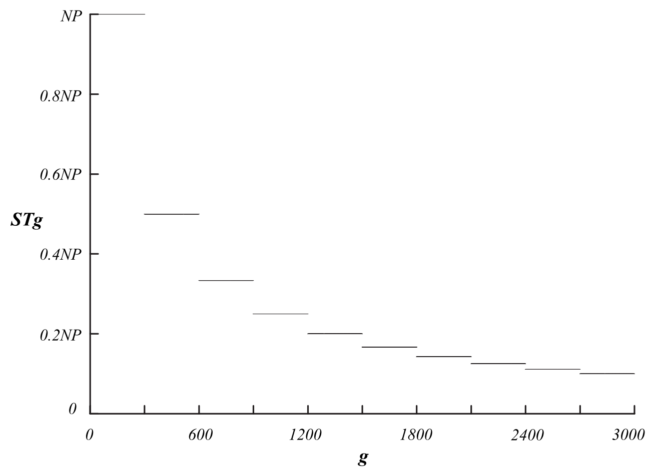

FIGURE 2. The change curve of ST_g with $G_{max} = 3000$ and $r = 10$ with the adaptive control scheme.

Based on this consideration, the *heuristic rule 2* (*HR*2) is designed to allocate the individual-dependent elite group (*IEG*) to each individual with respect to its search behavior.

Heuristic rule 2 (HR 2): Separation

$$IEG_g^i = \{X_{rank(j)} | X_{rank(j)} \in ELT_g \wedge j = 1, \ldots, SS_g^i\} \tag{10}$$

$$SS_g^i = \lfloor \frac{rank(i) - 1}{NP} \times |ELT_g| \rfloor + 1 \tag{11}$$

where IEG_g^i and SS_g^i are the elite group and its size for the *i*th individual in the sorted population, respectively.

From the definition of *HR*2 (Eq.10 and Eq.11), different individuals are assigned with different sizes of IEG_g. To elaborate, the individual with the larger ranking value will be equipped with a bigger size of IEG_g and thus has more elites as leaders for guiding. In addition, as $HR2$ and Fig.1 indicate, the IEG_gs for different individuals are overlapped. That is,

$$IEG_g^1 \subseteq IEG_g^2 \subseteq \cdots \subseteq IEG_g^{NP-1} \subseteq IEG_g^{NP} = ELT_g \quad (12)$$

3) STAGE 3: GUIDANCE

As mentioned in Section III-C, selecting parent vectors involved in the mutation operator plays a very important role for DE. In DE, a mutant vector can be treated as a leading vector to explore the search space, which is constructed by linearly combining the difference vector(s) and a base vector [50]. In the final stage, AGM will deal with the following problem: how to select the leaders as parent vectors from the IEG_g of each individual to guide its search effectively?

For this purpose, the *heuristic rule 3* ($HR3$) is given for selecting parents with the aid of IEG_g and the rest of current population.

Heuristic rule 3 (HR 3): Guidance

- *Selection of base vector*

$$X_{base_{rank(i),g}} = RandSele(IEG_g^i) \quad (13)$$

- *Selection of difference vector*

$$X_{end_{rank(i),g}} = RandSele(IEG_g^i) \quad (14)$$

$$X_{start_{rank(i),g}} = RandSele(POP_g - IEG_g^i) \quad (15)$$

where $RandSele(A)$ means an individual that is randomly selected from the set A, $X_{start_{rank(i),g}}$ and $X_{end_{rank(i),g}}$ are the start and end points of difference vector for $X_{rank(i),g}$, respectively, and POP_g is the population at the gth generation.

According to $HR3$ (Eq.13 - Eq.15), the base vector, as the leader to guide the search, is randomly selected from the IEG_g of the target individual. For difference vector, the end point is randomly selected from the corresponding IEG_g, while the start point is randomly chosen from the rest of population (i.e., $POP_g - IEG_g^i$). If $POP_g - IEG_g^i = \varnothing$, the worst individual of POP_g will be treated as the start point and a randomly selected individual from POP_g will be set as the end point. By this way, each individual can be further guided by the promising directions that are constructed through Eq.14 and Eq.15 in $HR3$. In addition, the randomness is used in the procedure of parents selection to further promote the potential in improving the diversity of population.

4) COMPLETE PROCEDURE OF AGM

By combining the above three stages, the complete procedure of AGM is depicted in **Algorithm 1**. From Algorithm 1, it is clear that AGM can be implemented and incorporated into most DE variants easily.

Based on the heuristic rules of the three stages in Algorithm 1, the original DE mutation strategies (i.e., Eq. (2) - (5)) enhanced with AGM are shown below.

- DE-AGM/*IEG*/1 (DE/rand/1 or DE/best/1 with AGM)

$$V_{rank(i),g} = X_{base_{rank(i),g}} + F \times (X_{end_{rank(i),g}} - X_{start_{rank(i),g}}) \quad (16)$$

Algorithm 1 Adaptive Guiding Mechanism (AGM)

1: Sort the current population in ascending order based on the fitness values;
2: *Construction stage*: use *HR 1* (Eq.8 – Eq.9) to establish an elite leadership team (ELT_g);
3: *Separation stage*: use *HR 2* (Eq.10 – Eq.11) to assign each individual with a distinct individual-dependent elite group (IEG_g) that is separated from ELT_g;
4: *Guidance stage*: use *HR 3* (Eq.13 – Eq.15) to select parents from IEG_g of each individual and current population for mutation.

- DE-AGM/DE/c-t-*IEG*/1 (DE/c-t-b/1 with AGM)

$$V_{rank(i),g} = X_{rank(i),g} + F \times (X_{base_{rank(i),g}} - X_{rank(i),g}) + F \times (X_{end_{rank(i),g}} - X_{start_{rank(i),g}}) \quad (17)$$

- DE-AGM/DE/r-t-*IEG*/1 (DE/r-t-b/1 with AGM)

$$V_{rank(i),g} = X_{r1,g} + F \times (X_{base_{rank(i),g}} - X_{r1,g}) + F \times (X_{end_{rank(i),g}} - X_{start_{rank(i),g}}) \quad (18)$$

In Eq. (16)–(18), $r1$, g, $base_{rank(i),g}$ and $start_{rank(i),g}$ are mutually different integers.

C. THE DE-AGM FRAMEWORK

By incorporating AGM into DE, the proposed DE-AGM framework is shown in **Algorithm 2**, where the differences from the original DE algorithm are highlighted with "$*$". As seen from Algorithm 2, DE-AGM is only different from the original DE algorithm in the selection of parents for the mutation operator, which is caused by introducing the AGM. Clearly, as an algorithm framework, DE-AGM can be seamlessly and handily applied to most DE variants.

In regard to algorithmic complexity, the additional computation of DE-AGM mainly comes from the AGM (see Algorithm 1). For sorting the whole population, it will take $O(NP \times logNP)$. In the three stages of AGM (i.e., construction, separation, and guidance), the complexity of establishing the ELT, separating the IEG, and selecting the parents are $O(1)$, respectively. Overall, the total complexity of DE-AGM is $O(NP \times (logNP + D) \times G_{max})$, which is still computationally efficient when compared with the original DE algorithm that takes $O(NP \times D \times G_{max})$.

D. DISCUSSIONS

1) DIFFERENCES BETWEEN DE-AGM AND THE NICHING-BASED DE VARIANTS

For the multi-modal optimization problems, the niching techniques are widely used in DE for locating multiple optimal solutions [51]. In most niching-based DE variants, the whole population is divided into several subpopulations (or species) so that each individual is restricted to communicate with the ones that belong to the same subpopulation [52]. Similar

Algorithm 2 DE With AGM (DE-AGM)

Initialize the population POP_0 and set $g = 0$;
Evaluate the fitness values of POP_0;
For $g = 0$ to G_{max} **Do**
  $*$ Perform the *adaptive guiding mechanism* (Algorithm 1);
  $*$ Apply the mutation operator with the selected parents to generate a mutant vector for each individual;
  Apply the crossover operator on each pair of target individual and its mutation vector to generate a trial vector;
  Evaluate the fitness values of all the trial vectors;
  Apply the selection operator on each pair of target individual and its trial vector to decide the winners for the next population POP_{g+1};
  $g = g + 1$;
End For
Output the best individual with its fitness value.

to the niching techniques used in DE, the proposed AGM partitions the population into different groups for guiding the evolution of population. However, there are significant differences between them, which are shown as follows:

1) The goal of niching techniques is to maintain the diversity of population. By using the niching techniques, the multiple potential solutions can be found and preserved during the evolutionary process of DE [51]. Oppositely, the AGM is used to guide each individual by a distinct group of elites based on its search behavior. To elaborate, the superior individuals are assigned with a smaller size of IEG_g for exploitation, while the inferior individuals are allocated with a bigger size of IEG_g for exploration. Therefore, other than the niching techniques for seeking multiple optimal solutions, the proposed AGM attempts to guide the DE population towards the global optimal solution via assigning each individual with an individual-dependent elites group based on its search behavior.
2) The partition of population in most niching-based DE variants is achieved by distance-based or topology-based methods [51]–[53]. In both methods, the niching parameters (e.g., niche radius, or crowding factor) are needed to be specified for setting the number of subpopulations. Thus, the size of subpopulations is fixed with the niching-based DE variants. Further, the overlapping subpopulations are always avoided in these DE variants to reduce the redundant searches. On the contrary, the partition of elite groups in DE-AGM is realized based on the fitness information of population by AGM. To elaborate, the elite leadership team (ELT_g) is constructed with the best individuals, followed by that different overlapping IEG_gs are separated from ELT_g for distinct individuals. Clearly, the partition of ELT_g in AGM depends on the search behaviors of different individuals, which is dramatically different from that in the niching techniques.

2) CHARACTERISTICS OF DE-AGM

Overall, the characteristics of DE-AGM can be summarized in the following aspects:

1) The elite leadership team is constructed by the ranking-based selection method and an adaptive control scheme strategy, which is beneficial for the balance between exploration and exploitation. As the previous studies show, DE is good at exploring the search space and thus is characterized by a high exploration [25], [54]. In DE-AGM, on the one hand, the ranking-based selection method utilizes good information of the best individuals to construct the elite team for guiding the evolution of population, which can help DE to enhance its exploitation ability. On the other hand, the adaptive control scheme gradually reduces the size of the elite team with the increasing of iterations. It is beneficial for DE to focus the search in the promising areas along with the process of evolution. With these two operators, AGM can make DE achieve a better balance between exploration and exploitation.
2) Distinct elite groups are assigned to different individuals based on their search behaviors, which can effectively promote the utilization of population information for guiding the search. Other than most DE variants that use the same leaders for different individuals, DE-AGM makes the best of the difference between individuals in search behavior to guide each individual by an individual-dependent elite group. Specifically, the superior individual is allocated with a smaller size of elites group for exploitation, while the inferior individual is assigned with a bigger size of elites group for exploration. In this way, the population information can be utilized more effectively by the AGM.
3) The leader that guides the search of each individual is selected from the respective elite group, meanwhile the promising direction information extracted from the population is incorporated into the mutation operator. On the one hand, each individual of population is simultaneously guided by multiple elites to reduce the chance of trapping in local optimum. On the other hand, the incorporation of direction information can effectively guide the search of individuals towards the promising regions. In addition, the randomness of selecting parents used in the *guidance* stage of AGM can make the mutation process in a more balanced way.

V. EXPERIMENTAL STUDY

In this section, 58 benchmark functions from the CEC2013 and CEC2017 special session on real-parameter optimization are chosen to evaluate the performance of DE-AGM. The 28 benchmark functions from CEC2013 [27] include 5 unimodal functions ($F1$-$F5$), 15 basic multimodal functions

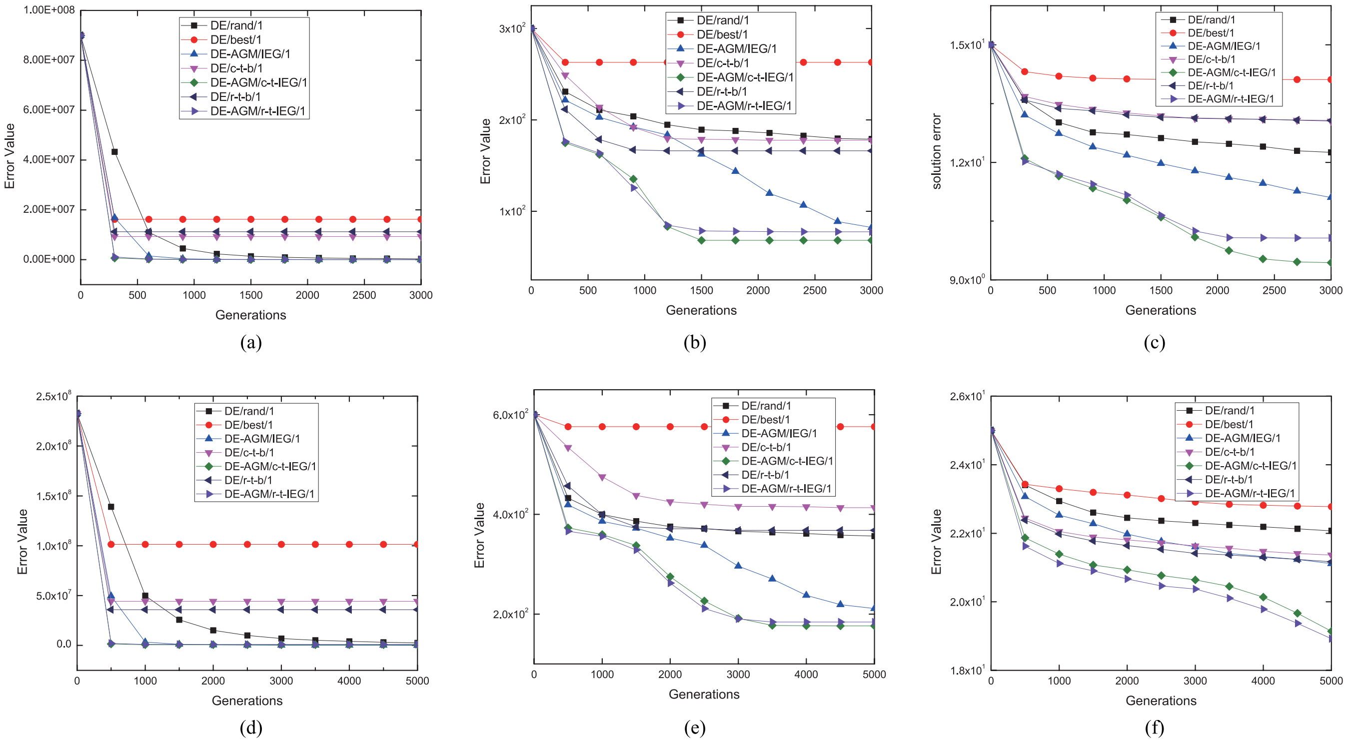

FIGURE 3. Convergence graphs of DE-AGM and the corresponding original DE algorithms for the selected CEC2013 functions. (a) F2, 30D. (b) F13, 30D. (c) F20, 30D. (d) F2, 50D. (e) F13, 50D. (f) F20, 50D.

TABLE 1. Parameter settings for the DE algorithms.

Parameters	Setting
Population size (NP)	100
Scaling factor (F)	0.5
Crossover factor (Cr)	0.9
Rate of reducing ST (r)	10
Independent number of runs ($NumR$)	30
Maximum number of generations (G_{max})	$10^2 \times D$

($F6$-$F20$) and 8 composition functions ($F21$-$F28$), while the 30 benchmark functions from CEC2017 [28] consist of 3 unimodal functions ($f1$-$f3$), 7 simple multimodal functions ($f4$-$f10$), 10 hybrid functions ($f11$-$f20$) and 10 composition functions ($f21$-$f30$). More details of them can be found in [27] and [28].

For a fair comparison, the parameters for the DE algorithms considered in this paper are set as Table 1 if no change is mentioned. To show the significant differences among the algorithms, the non-parametric statistical tests are carried out through the KEEL software.[1] The results of the single-problem analysis by the Wilcoxon test [55]–[57] at $\alpha = 0.05$ are shown in the tables as "$+/=/-$", which means that DE-AGM significantly outperforms, equals to and is significantly outperformed by the compared algorithm on the corresponding number of functions, respectively. For brevity, only the statistical results for the comparisons are given in this paper, and the detailed numerical values of simulations in the supplementary file can be obtained from the first author.

[1]KEEL: a software tool to assess EAs to data mining problems, which can be available from http://www.keel.es/.

A. PERFORMANCE ENHANCEMENT OF ORIGINAL DE ALGORITHMS

To test the effectiveness of the proposed framework on the original DE algorithm, DE-AGM is incorporated into four original mutation strategies, i.e., DE/rand/1, DE/best/1, DE/current-to-best/1 and DE/rand-to-best/1. The statistics summarizing the performance comparisons for the CEC2013 functions at 30D and 50D are shown in Tables 2 - 3. In these tables, based on the multiple-problem analysis by the Wilcoxon test, $R+$ and $R-$ mean the sum of ranks that DE-AGM performs significantly better than and worse than its competitor, respectively, which are used to identify differences between pair of algorithms on all the test functions [55]–[57]. In addition, the convergence graphs of DE-AGM and the corresponding DE algorithms for some selected test functions are shown in Figure 3.

According to Table 2, DE-AGM can enhance the performance of all the considered DE algorithms on the CEC2013 test functions at 30D. For the explorative strategies (i.e., DE/rand/1), DE-AGM is significantly better than it on 13 functions. For the exploitative strategies (i.e., DE/best/1, DE/current-to-best/1, and DE/rand-to-best/1), DE-AGM is significantly better than them on 23, 27 and 24 functions, respectively. Based on the statistical tests

TABLE 2. Results of single- and multiple-problem analysis by the Wilcoxon test between DE-AGM and the corresponding original DE algorithms for the CEC2013 functions at 30D.

Algorithm	$+/=/-$	$R+$	$R-$	p-value	$\alpha = 0.05$	$\alpha = 0.1$
DE-AGM/IEG/1 *vs.* DE/rand/1	13/6/9	269.5	136.5	1.26E-01	YES	YES
DE-AGM/IEG/1 *vs.* DE/best/1	23/2/3	349.0	57.0	8.50E-04	YES	YES
DE-AGM/c-t-IEG/1 *vs.* DE/c-t-b/1	27/1/0	378.0	0	5.00E-06	YES	YES
DE-AGM/r-t-IEG/1 *vs.* DE/r-t-b/1	24/4/0	388.0	18.0	2.40E-05	YES	YES

TABLE 3. Results of single- and multiple-problem analysis by the Wilcoxon test between DE-AGM and the corresponding original DE algorithms for the CEC2013 functions at 50D.

Algorithm	$+/=/-$	$R+$	$R-$	p-value	$\alpha = 0.05$	$\alpha = 0.1$
DE-AGM/IEG/1 *vs.* DE/rand/1	12/7/9	242.0	136.0	1.99E-01	YES	YES
DE-AGM/IEG/1 *vs.* DE/best/1	22/3/3	349.5	56.5	8.16E-04	YES	YES
DE-AGM/c-t-IEG/1 *vs.* DE/c-t-b/1	26/2/0	378.0	0.0	5.00E-06	YES	YES
DE-AGM/r-t-IEG/1 *vs.* DE/r-t-b/1	24/4/0	377.0	1.0	6.00E-06	YES	YES

TABLE 4. Results of single- and multiple-problem analysis by the Wilcoxon test between DE-AGM and the corresponding advanced DE algorithms for the CEC2013 functions at 30D.

Algorithm	$+/=/-$	$R+$	$R-$	p-value	$\alpha = 0.05$	$\alpha = 0.1$
ODE-AGM *vs.* ODE	23/3/2	375.0	3.0	7.00E-06	YES	YES
CoDE-AGM *vs.* CoDE	15/11/2	316.5	61.5	2.04E-03	YES	YES
SaDE-AGM *vs.* SaDE	15/11/2	298.5	79.5	8.22E-03	YES	YES
MDEpBX-AGM *vs.* MDEpBX	18/10/0	396.0	10.0	1.00E-05	YES	YES
SHADE-AGM *vs.* SHADE	11/11/6	196.5	181.5	8.47E-01	NO	NO

TABLE 5. Results of single- and multiple-problem analysis by the Wilcoxon test between DE-AGM and the corresponding advanced DE algorithms for the CEC2013 functions at 50D.

Algorithm	$+/=/-$	$R+$	$R-$	p-value	$\alpha = 0.05$	$\alpha = 0.1$
ODE-AGM *vs.* ODE	13/10/5	286.5	119.5	5.58E-02	NO	YES
CoDE-AGM *vs.* CoDE	15/12/1	315.5	62.5	1.66E-03	YES	YES
SaDE-AGM *vs.* SaDE	16/10/2	311.0	67.0	3.16E-03	YES	YES
MDEpBX-AGM *vs.* MDEpBX	20/6/2	337.0	41.0	3.46E-04	YES	YES
SHADE-AGM *vs.* SHADE	8/13/7	234.5	143.5	2.68E-01	NO	NO

for multiple-problem analysis, DE-AGM obtains the higher $R+$ values than $R-$ values in all the cases. Further, all the p values are less than 0.05 and 0.1, which means that DE-AGM is significantly better than the corresponding original DE algorithms overall.

From Table 3, DE-AGM also consistently outperforms the corresponding DE algorithms on the functions at 50D. Specifically, DE-AGM is significantly better than DE/rand/1, DE/best/1, DE/current-to-best/1 and DE/rand-to-best/1 on 12, 22, 26 and 24 functions, respectively. In addition, the statistical tests show that DE-AGM obtains the higher $R+$ values than $R-$ values and the p values are less than both 0.05 and 0.01, in all the cases.

Furthermore, as shown in Figure 3, DE-AGM is better than the corresponding DE algorithm in terms of convergence rate for most selected functions.

In general, the above results clearly indicate that the proposed DE-AGM framework can greatly enhance the performance of these basic DE algorithms.

B. PERFORMANCE ENHANCEMENT OF ADVANCED DE VARIANTS

To further test the effect on advanced DE variants, DE-AGM is incorporated with five DE variants, ODE [58], CoDE [23], SaDE [9], MDEpBX [29] and SHADE [6]. The parameters of these DE variants are set as their original papers except that *NP* in CoDE and SaDE are set to 100. The statistics summarizing the performance comparisons for the functions at 30D and 50D are shown in Tables 4 - 5. Additionally, Figure 4 depicts the convergence graphs of the competitors for some selected test functions.

As shown in Tables 4 - 5, on the one hand, DE-AGM can significantly outperform most advanced DE variants on the test functions. DE-AGM is significantly better than the corresponding ODE, CoDE, SaDE, MDEpBX and SHADE on 23, 15, 15, 18 and 11 functions at 30D, respectively. For the functions at 50D in Table 5, DE-AGM significantly outperforms the corresponding DE variants on 13, 15, 16, 20 and 8 functions, respectively. On the other hand, in terms

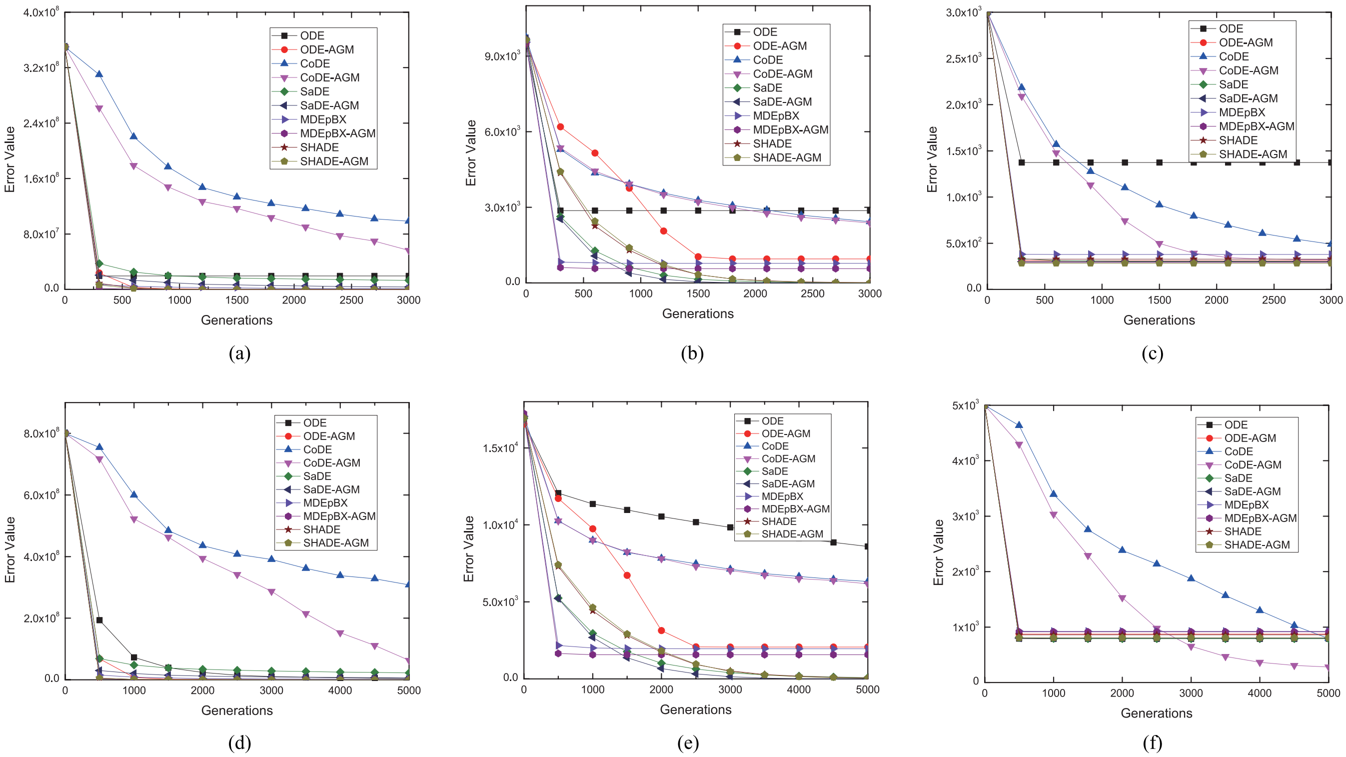

FIGURE 4. Convergence graphs of DE-AGM and the corresponding advanced DE variants for the selected CEC2013 functions. (a) F2, 30D. (b) F14, 30D. (c) F21, 30D. (d) F2, 50D. (e) F14, 50D. (f) F21, 50D.

of convergence speed, DE-AGM also beats the corresponding DE variants on most selected functions according to Figure 4.

Based on the multiple-problem analysis by the Wilcoxon test in Tables 4 - 5, DE-AGM can obtain the higher $R+$ values than $R-$ values in all the cases. Further, according to the p value, four cases obtain the values that are less than 0.05 and 0.1, for the functions at 30D in Table 4. From Table 5, the p values are less than 0.05 and 0.1 in three and four cases, respectively, for the functions at 50D. These results indicate that DE-AGM performs significantly better than most corresponding DE variants on the test functions overall according to the multiple problem statistical analysis.

In summary, these results in Tables 4-5 and Figure 4 clearly show that DE-AGM is an effective framework to further improve the performance of most advanced DE variants.

C. COMPARISON WITH STATE-OF-THE-ART EAS

In this subsection, the proposed DE-AGM framework is further compared with six state-of-the-art EAs that have been reported to have good performance in the CEC2013 competition on real-parameter optimization. They are covariance adaptation matrix evolution strategy with re-sampled inheritance search (CMAES-RIS) [59], frequency-based heterogeneous particle swarm optimizer (fk-HPSO) [60], genetic algorithm with three-parent crossover (TPC-GA) [61], standard particle swarm optimization with artificial bee colony (ABC-SPSO) [62], continuous differential ant-stigmergy algorithm (CDASA) [63], and hybrid mean-variance mapping optimization (MVMO-SH) [64]. The results of these algorithms are obtained directly from their original papers.

To show the advantages of the proposed framework, the best overall performing DE-AGM algorithm will be chosen for this comparison. For this purpose, the Friedman test, conducted by the KEEL [55], is used to obtain the rankings of different DE-AGM versions and the corresponding DE algorithms for all CEC2013 test functions. The results are shown in Table 6, where "ARV" and "Ranking" mean the average ranking value and the final ranking value of the corresponding algorithm, respectively. As shown in Table 6, SHADE-AGM obtains the first rank, followed by SHADE for all the functions both at 30D and 50D. In addition, it is interesting to find that all the DE-AGM versions obtain the better ranking value than the corresponding DE algorithms. Based on the results in Table 6, SHADE-AGM is selected for comparison in this study. The results for the performance comparisons between SHADE-AGM and other EAs are shown in Table 7, where "NBS" means the number of best solutions obtained by the algorithm in terms of the mean error value. The detailed results of comparison can be found in the supplemental file of this paper.

As shown in Table 7, SHADE-AGM gets the second rank overall for the functions both at 30D and 50D. It outperforms CMADE-RIS, ABC-SPSO, fk-PSO, CDASA, and TPC-GA, and is only outperformed by MVMO-SH. In addition, with respect to "NBS", SHADE-AGM achieves the best mean error value on 13 and 11 functions at 30D

TABLE 6. Average ranking of all the competitors by Friedman test for the CEC2013 functions at 30D and 50D.

Algorithm at 30D	AVG	Ranking	Algorithm at 50D	AVG	Ranking
SHADE-AGM	3.804	1	SHADE-AGM	3.946	1
SHADE	4.000	2	SHADE	4.446	2
DE-AGM/c-t-IEG/1	5.286	3	MDEpBX-AGM	6.554	3
MDEpBX-AGM	5.643	4	SaDE-AGM	6.589	4
SaDE-AGM	6.482	5	ODE-AGM	7.036	5
ODE-AGM	6.714	6	DE-AGM/c-t-IEG/1	7.054	6
DE-AGM/r-t-IEG/1	7.536	7	DE-AGM/IEG/1	7.518	7
SaDE	7.661	8	SaDE	7.875	8
DE-AGM/IEG/1	7.821	9	DE-AGM/r-t-IEG/1	8.661	9
DE/rand/1	8.929	10	ODE	8.732	10
MDEpBX	9.054	11	MDEpBX	8.821	11
DE/r-t-b/1	12.196	12	DE/rand/1	8.893	12
DE/c-t-b/1	12.964	13	CoDE-AGM	12.321	13
CoDE-AGM	13.232	14	DE/r-t-b/1	12.643	14
ODE	13.804	15	CoDE	13.786	15
DE/best/1	13.804	16	DE/c-t-b/1	13.982	16
CoDE	14.071	17	DE/best/1	14.143	17

TABLE 7. Average ranking of SHADE-AGM and other EAs by Friedman test for the CEC2013 functions at 30D and 50D.

Algorithm at 30D	AVG	Ranking	NBS	Algorithm at 50D	AVG	Ranking	NBS
MVMO-SH	2.393	1	10	MVMO-SH	2.393	1	9
SHADE-AGM	2.750	2	13	SHADE-AGM	2.839	2	11
CMAES-RIS	3.839	3	9	CMAES-RIS	3.536	3	7
ABC-SPSO	4.179	4	6	ABC-SPSO	4.429	4	4
fk-PSO	4.518	5	4	fk-PSO	4.661	5	3
CDASA	5.143	6	2	CDASA	4.714	6	2
TPC-GA	5.179	7	2	TPC-GA	5.429	7	1

and 50D, respectively, while MVMO-SH obtains the best mean error value on 10 and 9 functions at 30D and 50D, respectively. Note that MVMO-SH is a memetic EA that incorporates local search and multi-parent crossover strategies [64]. Therefore, it is not surprising that MVMO-SH can obtain the better result than SHADE-AGM overall, due to that SHADE-AGM is an pure DE algorithm. In the future, some sophisticated local search operators will be incorporated into the DA-AGM framework to further improve its performance.

D. COMPARISON WITH MULTI-TOPOLOGY-BASED DE VARIANT

Recently, Sun *et al.* [14] proposed a multi-topology-based DE (MTDE) variant that employs four population topologies with different degrees of connectivity to guide the search of DE by an individual-dependent adaptive topology selection scheme [14]. Similar to MTDE, DE-AGM also designs a guiding mechanism that depends on the search behaviors of different individuals to improve the performance of DE. Hence, to test the effectiveness of AGM, the comparison between DE-AGM and MTDE is carried out. The experimental results on the CEC2013 functions at 30D and 50D are shown in Tables 8 and 9, respectively. In addition, the box plots for some selected functions are shown in Figure 5.

As observed from Table 8, DE-AGM can obtain significantly better results than MTDE overall in most cases. Specifically, in the cases of DE/best/1, DE/current-to-best/1, DE/rand-to-best/1, CoDE, and SaDE, DE-AGM is significantly better than MTDE on 17, 24, 15, 15, and 13 functions at 30D, respectively. For the functions at 50D in Table 9, DE-AGM also significantly outperforms the corresponding MTDE on most functions. Based on the results of multi-problem analysis by the Wilcoxon test, DE-AGM obtains the higher $R+$ values than $R-$ values in all the cases. With regards to p, the values are less than 0.05 and 0.1 in most cases for the functions both at 30D and 50D. These results clearly indicate that significant differences can be observed between DE-AGM and MTDE in most cases according to the multiple problem statistical analysis. Furthermore, Figure 5 shows that DE-AGM performs better than the corresponding MTDE on all the selected functions in terms of robustness.

According to the above analysis, DE-AGM is a more effective framework than MTDE on the test functions in utilizing the difference between individuals in search behavior. Further, the advantages of AGM also be demonstrated when compared with the guiding mechanism in MTDE.

E. EFFECTIVENESS OF AGM

As reviewed in Section III-C, several DE frameworks are proposed for selecting elites as leaders to guide the evolution of population. To further investigate the effectiveness of AGM, rank-based strategy [25], proximity-based strategy [45], and multiobjective sorting-based strategy [20] are selected for

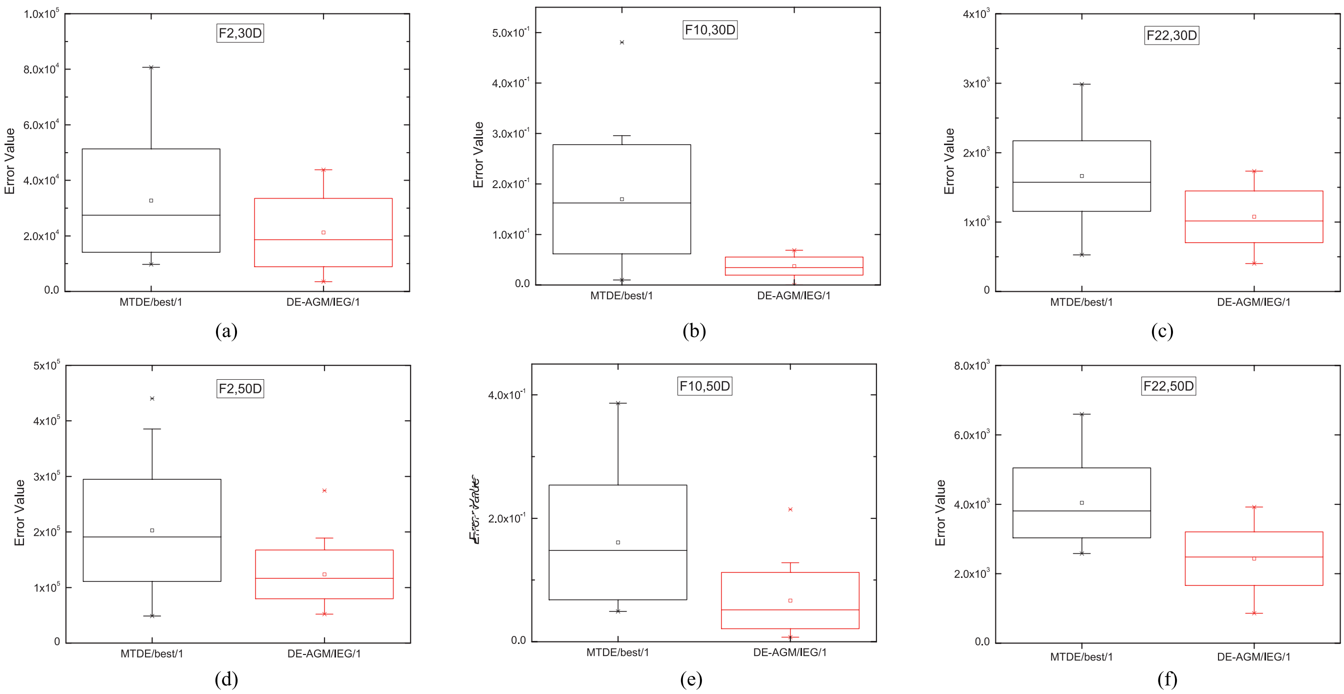

FIGURE 5. Box plots of DE-AGM and MTDE with the "DE/best/1" strategy on the selected functions.

TABLE 8. Results of single-and multiple-problem analysis by the Wilcoxon test between DE-AGM and MTDE for the CEC2013 functions at 30D.

Algorithm	$+/=/-$	$R+$	$R-$	p**-value**	$\alpha = 0.05$	$\alpha = 0.1$
DE-AGM/IEG/1 *vs.* MTDE/rand/1	7/10/11	222.5	183.5	6.49E-01	NO	NO
DE-AGM/IEG/1 *vs.* MTDE/best/1	17/6/5	299.0	107.0	2.75E-02	YES	YES
DE-AGM/c-t-IEG/1 *vs.* MTDE/c-t-b/1	24/4/0	390.0	16.0	2.00E-05	YES	YES
DE-AGM/r-t-IEG/1 *vs.* MTDE/r-t-b/1	15/11/2	310.0	96.0	1.44E-02	YES	YES
CoDE-AGM *vs.* MTCoDE	15/12/1	310.5	67.5	3.01E-03	YES	YES
SaDE-AGM *vs.* MTSaDE	13/11/4	288.5	117.5	5.02E-02	NO	YES

TABLE 9. Results of single-and multiple-problem analysis by the Wilcoxon test between DE-AGM and MTDE for the CEC2013 functions at 50D.

Algorithm	$+/=/-$	$R+$	$R-$	p**-value**	$\alpha = 0.05$	$\alpha = 0.1$
DE-AGM/IEG/1 *vs.* MTDE/rand/1	8/10/10	170.0	208.0	1.00E+00	NO	NO
DE-AGM/IEG/1 *vs.* MTDE/best/1	16/10/2	310.0	68.0	3.41E-03	YES	YES
DE-AGM/c-t-IEG/1 *vs.* MTDE/c-t-b/1	22/3/3	306.0	72.0	4.76E-03	YES	YES
DE-AGM/r-t-IEG/1 *vs.* MTDE/r-t-b/1	15/12/1	253.0	125.0	1.21E-01	NO	NO
CoDE-AGM *vs.* MTCoDE	15/12/1	318.5	59.5	1.52E-03	YES	YES
SaDE-AGM *vs.* MTSaDE	14/10/4	314.5	63.5	2.40E-03	YES	YES

comparison. The DE variants with these three strategies are denoted as rankDE, proDE and MSDE, respectively. Here, six DE algorithms, i.e., DE/rand/1, DE/best/1, DE/current-to-best/1, DE/rand-to-best/1, CoDE, and SaDE, are used in this experimental study. In addition, to show the robustness of the DE variants when equipped with different selection strategies, the box plots for some selected functions are depicted in Figure 6.

1) COMPARISON WITH RANK-BASED STRATEGY

The results of the comparisons between DE-AGM and rankDE are shown in Tables 10 - 11. From Table 10, DE-AGM significantly outperforms rankDE in most cases. In the cases of DE/rand/1, DE/best/1, DE/current-to-best/1, DE/rand-to-best/1, CoDE, and SaDE, DE-AGM is significantly better than corresponding rankDE on 11, 23, 27, 23, 10 and 8 functions at 30D, respectively. For the functions at 50D, DE-AGM also obtains significantly better results than rankDE in most cases. Further, based on the results of multi-problem analysis by the Wilcoxon test, DE-AGM obtains higher $R+$ values than $R-$ values in all the cases. According to the p values, DE-AGM is significantly better than the corresponding rankDE variant overall in most cases. Also, Figure 6 exhibits the advantage of DE-AGM in terms of

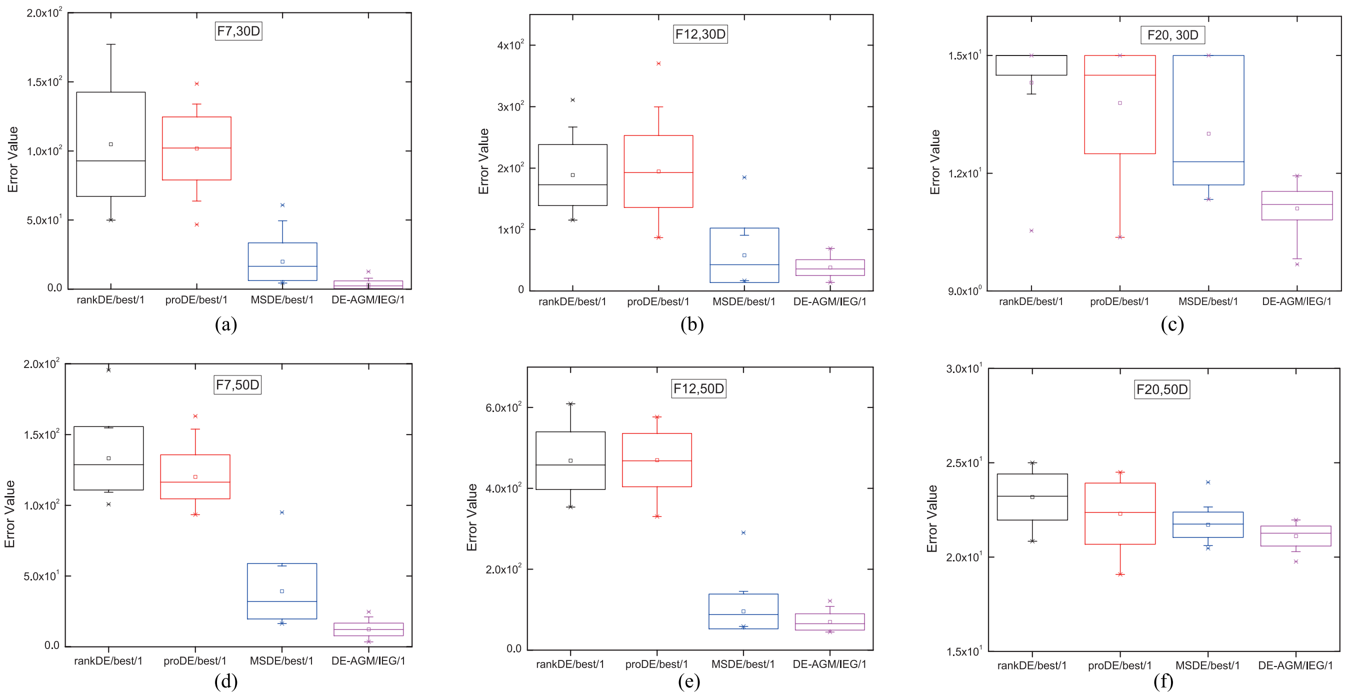

FIGURE 6. Box plots of DE-AGM, rankDE, proDE, and MSDE with the "DE/best/1" strategy on the selected functions.

TABLE 10. Results of single- and multiple-problem analysis by the Wilcoxon test between DE-AGM and rankDE for the CEC2013 test functions at 30D.

Algorithm	$+/=/-$	$R+$	$R-$	p-**value**	$\alpha = 0.05$	$\alpha = 0.1$
DE-AGM/IEG/1 *vs.* rankDE/rand/1	11/7/10	247.5	158.5	3.05E-01	NO	NO
DE-AGM/IEG/1 *vs.* rankDE/best/1	23/2/3	350.0	56.0	7.83E-04	YES	YES
DE-AGM/c-t-IEG/1 *vs.* rankDE/c-t-b/1	27/1/0	406.0	0.0	4.00E-06	YES	YES
DE-AGM/r-t-IEG/1 *vs.* rankDE/r-t-b/1	23/3/2	321.0	57.0	1.46E-03	YES	YES
CoDE-AGM *vs.* rankCoDE	10/15/3	296.5	109.5	3.13E-02	YES	YES
SaDE-AGM *vs.* rankSaDE	8/15/5	270.0	108.0	4.95E-02	YES	YES

TABLE 11. Results of single- and multiple-problem analysis by the Wilcoxon test between DE-AGM and rankDE for the CEC2013 test functions at 50D.

Algorithm	$+/=/-$	$R+$	$R-$	p-**value**	$\alpha = 0.05$	$\alpha = 0.1$
DE-AGM/IEG/1 *vs.* rankDE/rand/1	10/8/10	231.5	174.5	5.09E-01	NO	NO
DE-AGM/IEG/1 *vs.* rankDE/best/1	22/3/3	327.0	51.0	8.76E-04	YES	YES
DE-AGM/c-t-IEG/1 *vs.* rankDE/c-t-b/1	27/1/0	378.0	0.0	5.00E-06	YES	YES
DE-AGM/r-t-IEG/1 *vs.* rankDE/r-t-b/1	24/3/1	343.0	35.0	2.06E-04	YES	YES
CoDE-AGM *vs.* rankCoDE	14/10/4	255.5	122.5	1.03E-01	NO	NO
SaDE-AGM *vs.* rankSaDE	15/9/4	293.5	84.5	1.16E-02	YES	YES

robustness on all the selected functions. Overall, these results in Tables 10 - 11 and Figure 6 indicate that DE-AGM is more effective than rankDE on the test functions.

2) COMPARISON WITH PROXIMITY-BASED STRATEGY

The results for the comparison between DE-AGM and proDE are shown in Tables 12 - 13. As shown in the Tables, DE-AGM obtains the significantly better results than proDE in most cases. Further, based on the results of multi-problem analysis by the Wilcoxon test, DE-AGM obtains higher $R+$ values than $R-$ values in all the cases. The values of p also clearly indicate that DE-AGM is significantly better than proDE overall in five cases for the functions both at 30D and 50D. According to Figure 6, DE-AGM is better than proDE in terms of robustness on all the selected functions. Therefore, we can conclude that AGM is more effective than the proximity-based strategy in utilizing the neighborhood information to guide the search of DE.

3) COMPARISON WITH MULTIOBJECTIVE SORTING-BASED STRATEGY

The results for the comparison between DE-AGM and MSDE are shown in Tables 14 - 15. From the results

TABLE 12. Results of single- and multiple-problem analysis by the Wilcoxon test between DE-AGM and proDE for the CEC2013 functions at 30D.

Algorithm	$+/=/-$	$R+$	$R-$	p-value	$\alpha = 0.05$	$\alpha = 0.1$
DE-AGM/IEG/1 *vs.* proDE/rand/1	14/5/9	269.5	136.5	1.27E-01	NO	NO
DE-AGM/IEG/1 *vs.* proDE/best/1	23/2/3	351.0	55.0	7.21E-04	YES	YES
DE-AGM/c-t-IEG/1 *vs.* proDE/c-t-b/1	27/1/0	406.0	80.0	4.00E-06	YES	YES
DE-AGM/r-t-IEG/1 *vs.* proDE/r-t-b/1	24/4/0	363.0	15.0	2.80E-05	YES	YES
CoDE-AGM *vs.* proCoDE	14/12/2	329.5	48.5	6.30E-04	YES	YES
SaDE-AGM *vs.* proSaDE	15/11/2	300.5	77.5	6.96E-03	YES	YES

TABLE 13. Results of single- and multiple-problem analysis by the Wilcoxon test between DE-AGM and proDE for the CEC2013 functions at 50D.

Algorithm	$+/=/-$	$R+$	$R-$	p-value	$\alpha = 0.05$	$\alpha = 0.1$
DE-AGM/IEG/1 *vs.* proDE/rand/1	12/7/9	226.5	151.5	3.60E-01	NO	NO
DE-AGM/IEG/1 *vs.* proDE/best/1	22/3/3	328.0	50.0	8.04E-04	YES	YES
DE-AGM/c-t-IEG/1 *vs.* proDE/c-t-b/1	26/2/0	378.0	0.0	5.00E-06	YES	YES
DE-AGM/r-t-IEG/1 *vs.* proDE/r-t-b/1	24/4/0	377.0	1.0	6.00E-06	YES	YES
CoDE-AGM *vs.* proCoDE	14/14/0	364.5	41.5	1.90E-04	YES	YES
SaDE-AGM *vs.* proSaDE	16/9/3	341.0	65.0	1.61E-03	YES	YES

TABLE 14. Results of single-and multiple-problem analysis by the Wilcoxon test between DE-AGM and MSDE for the CEC2013 functions at 30D.

Algorithm	$+/=/-$	$R+$	$R-$	p-value	$\alpha = 0.05$	$\alpha = 0.1$
DE-AGM/IEG/1 *vs.* MSDE/rand/1	15/3/10	272.5	133.5	1.11E-01	NO	NO
DE-AGM/IEG/1 *vs.* MSDE/best/1	13/9/6	291.5	114.5	4.27E-02	YES	YES
DE-AGM/c-t-IEG/1 *vs.* MSDE/c-t-b/1	25/3/0	397.0	9.0	9.00E-06	YES	YES
DE-AGM/r-t-IEG/1 *vs.* MSDE/r-t-b/1	22/5/1	360.0	18.0	3.80E-05	YES	YES
CoDE-AGM *vs.* MSCoDE	14/12/2	351.5	54.5	5.78E-04	YES	YES
SaDE-AGM *vs.* MSSaDE	15/12/1	298.0	80.0	8.32E-03	YES	YES

TABLE 15. Results of single-and multiple-problem analysis by the Wilcoxon test between DE-AGM and MSDE for the CEC2013 functions at 50D.

Algorithm	$+/=/-$	$R+$	$R-$	p-value	$\alpha = 0.05$	$\alpha = 0.1$
DE-AGM/IEG/1 *vs.* MSDE/rand/1	12/7/9	228.5	177.5	5.54E-01	NO	NO
DE-AGM/IEG/1 *vs.* MSDE/best/1	13/10/5	257.0	121.0	9.98E-02	NO	YES
DE-AGM/c-t-IEG/1 *vs.* MSDE/c-t-b/1	26/1/1	370.0	8.0	1.30E-05	YES	YES
DE-AGM/r-t-IEG/1 *vs.* MSDE/r-t-b/1	23/5/0	367.5	10.5	1.50E-05	YES	YES
CoDE-AGM *vs.* MSCoDE	14/14/0	364.5	41.5	1.90E-04	YES	YES
SaDE-AGM *vs.* MSSaDE	16/9/3	341.0	65.0	1.61E-03	YES	YES

in Tables 14 - 15, DE-AGM significantly outperforms MSDE in most cases. For DE/best/1, DE/current-to-best/1, DE/rand-to-best/1, CoDE, and SaDE, DE-AGM is significantly better than MSDE on most functions both at 30D and 50D. Based on multi-problem analysis by the Wilcoxon test, the significant differences between DE-AGM and MSDE are also observed in these five cases. In the case of DE/rand/1, DE-AGM can also obtain higher $R+$ values than $R-$ values, although no significant differences can be observed between them. Additionally, Figure 6 indicates the better performance of DE-AGM when compared with MS-DE on the selected functions. In general, the effectiveness of AGM is further demonstrated when compared with the multiobjective sorting-based strategy.

F. PERFORMANCE SENSITIVITY TO r

In DE-AGM, the control parameter r is used to control the rate of reducing ST_g for constructing the ELT_g during the evolutionary process. To evaluate the influence of r on the performance of DE-AGM, the comparisons of the DE-AGM variants with different r values are carried out. Here, five different r values (i.e., 10, 20, 30, 40 and 50) are considered, and two DE-AGM variants (DE-AGM/IEG/1 and ODE-AGM) are used for comparison. The results of multiple-problem Wilcoxon tests on the CEC2013 functions are summarized in Tables 16 - 17.

As Table 16 shows, all the DE-AGM/IEG/1 variants are able to obtain the significantly better results than DE/best/1. When $r = 10$, the performance of DE-AGM/IEG/1 is better than other four variants, although no significant differences can be observed between them. When $r = 20$, DE-AGM/IEG/ is significantly better than the variants with $r = 30$ and $r = 50$. Similarly, the results in Table 17 show that all the ODE-AGM variants are significantly better than ODE. ODE-AGM with $r = 10$ is significantly better than other four variants at both $\alpha = 0.1$ and $\alpha = 0.05$,

TABLE 16. Summary of the multi-problem Wilcoxon test for DE/best/1 versus DE-AGM/IEG/1 with different *r* values on the CEC2013 functions at 30D. DE-AGM/IEG/1(*a*) denotes DE-AGM/IEG/1 with the *r* value that is set to *a*.

Algorithm		$\alpha = 0.05$	$\alpha = 0.1$	Algorithm		$\alpha = 0.05$	$\alpha = 0.1$
DE/best/1				**DE-AGM/IEG/1(10)**			
	vs. DE-AGM/IEG/1(10)	–	–		*vs.* DE-AGM/IEG/1(20)	=	=
	vs. DE-AGM/IEG/1(20)	–	–		*vs.* DE-AGM/IEG/1(30)	=	=
	vs. DE-AGM/IEG/1(30)	–	–		*vs.* DE-AGM/IEG/1(40)	=	=
	vs. DE-AGM/IEG/1(40)	–	–		*vs.* DE-AGM/IEG/1(50)	=	=
	vs. DE-AGM/IEG/1(50)	–	–	**DE-AGM/IEG/1(30)**			
DE-AGM/IEG/1(20)					*vs.* DE-AGM/IEG/1(40)	–	–
	vs. DE-AGM/IEG/1(30)	+	+		*vs.* DE-AGM/IEG/1(50)	=	=
	vs. DE-AGM/IEG/1(40)	=	=	**DE-AGM/IEG/1(40)**			
	vs. DE-AGM/IEG/1(50)	+	+		*vs.* DE-AGM/IEG/1(50)	=	=

TABLE 17. Summary of the multi-problem Wilcoxon test for ODE versus ODE-AGM with different *r* values on the CEC2013 functions at 30D. ODE-AGM(*a*) denotes ODE-AGM with the *r* value that is set to *a*.

Algorithm		$\alpha = 0.05$	$\alpha = 0.1$	Algorithm		$\alpha = 0.05$	$\alpha = 0.1$
ODE				**ODE-AGM(10)**			
	vs. ODE-AGM(10)	–	–		*vs.* ODE-AGM(20)	+	+
	vs. ODE-AGM(20)	–	–		*vs.* ODE-AGM(30)	+	+
	vs. ODE-AGM(30)	–	–		*vs.* ODE-AGM(40)	+	+
	vs. ODE-AGM(40)	–	–		*vs.* ODE-AGM(50)	+	+
	vs. ODE-AGM(50)	–	–	**ODE-AGM(30)**			
ODE-AGM(20)					*vs.* ODE-AGM(40)	=	=
	vs. ODE-AGM(30)	=	=		*vs.* ODE-AGM(50)	=	=
	vs. ODE-AGM(40)	+	+	**ODE-AGM(40)**			
	vs. ODE-AGM(50)	=	+		*vs.* ODE-AGM(50)	=	=

TABLE 18. Results of single-and multiple-problem analysis by the Wilcoxon test between DE-AGM and the corresponding DE algorithm for the CEC2017 functions at 30D.

Algorithm	$+/=/-$	$R+$	$R-$	p-value	$\alpha = 0.05$	$\alpha = 0.1$
DE-AGM/IEG/1 *vs.* DE/rand/1	21/5/4	419.5	45.5	1.15E-04	YES	YES
DE-AGM/IEG/1 *vs.* DE/best/1	29/0/1	446.0	19.0	1.10E-05	YES	YES
DE-AGM/c-t-IEG/1 *vs.* DE/c-t-b/1	29/1/0	465.0	0	2.00E-06	YES	YES
DE-AGM/r-t-IEG/1 *vs.* DE/r-t-b/1	29/1/0	465.0	0	2.00E-06	YES	YES
CoDE-AGM *vs.* CoDE	23/6/1	446.5	18.5	9.00E-06	YES	YES
ODE-AGM *vs.* ODE	9/15/6	294.5	170.5	1.99E-01	NO	NO
SaDE-AGM *vs.* SaDE	18/10/2	403.5	31.5	5.30E-05	YES	YES
MDEpBX-AGM *vs.* MDEpBX	19/11/0	416.0	49.0	1.42E-04	YES	YES
jSO-AGM *vs.* jSO	0/29/1	339.5	125.5	2.66E-02	YES	YES

while ODE-AGM with $r = 20$ significantly outperforms ODE-AGM with both $r = 40$ and $r = 50$. There are no significant differences among the ODE-AGM variants with $r = 30$, $r = 40$ and $r = 50$.

Further, some observations can be obtained from these results in Tables 16 - 17. First, the value of r has a marked impact on the performance of DE-AGM. The DE-AGM variants with different r values show significantly different performances on the test functions. Second, the performance of DE-AGM gets worse as the r value increases. In most cases, the DE-AGM variant with a smaller r value can perform better than those with a larger r value. The reason may lie in that when the value of r is too large, the size of *ELT* will be reduced rapidly. Consequently, the evolution of population will be guided by a smaller set of elites, which will make DE-AGM be over-exploitative and thus lead to premature convergence.

Overall, these results of Tables 16-17 indicate that DE-AGM with different r values can obtain significant improvements for the DE algorithms considered, and a smaller value of r (e.g., 10 or 20) is a good choice for DE-AGM when solving different test functions. In the future, the adaptive or self-adaptive parameter control strategies will be investigated to select the suitable values of r for different stages of evolution or different types of functions.

G. MORE COMPARISON ON THE CEC2017 TEST SUITE

To further test DE-AGM on other optimization functions, the CEC2017 test suite with 30 functions [28] is used. Here, the original DE algorithm, as well as the advanced DE variants, are employed for comparison. In addition, jSO [8], which has shown its highly competitive in the CEC2017 competition, is included. The results for the CEC2017 test

TABLE 19. Results of single-and multiple-problem analysis by the Wilcoxon test between DE-AGM and the corresponding DE algorithm for the CEC2017 functions at 50D.

Algorithm	$+/=/-$	$R+$	$R-$	p-value	$\alpha = 0.05$	$\alpha = 0.1$
DE-AGM/IEG/1 *vs.* DE/rand/1	19/4/7	355.0	110.0	1.14E-02	YES	YES
DE-AGM/IEG/1 *vs.* DE/best/1	26/2/2	432.0	33.0	3.70E-05	YES	YES
DE-AGM/c-t-IEG/1 *vs.* DE/c-t-b/1	28/2/0	460.0	5.0	3.00E-06	YES	YES
DE-AGM/r-t-IEG/1 *vs.* DE/r-t-b/1	27/2/1	460.0	5.0	1.40E-05	YES	YES
CoDE-AGM *vs.* CoDE	22/8/0	435.0	0	2.00E-06	YES	YES
ODE-AGM *vs.* ODE	10/13/7	281.0	184.0	3.14E-01	NO	NO
SaDE-AGM *vs.* SaDE	20/8/2	403.0	62.0	3.80E-04	YES	YES
MDEpBX-AGM *vs.* MDEpBX	22/8/0	432.0	33.0	3.70E-05	YES	YES
jSO-AGM *vs.* jSO	1/28/1	273.5	191.5	3.87E-01	NO	NO

functions at 30D and 50D are shown in Tables 18 and 19, respectively.

As shown in Tables 18 - 19, it is clear that DE-AGM also significantly improves the performance of most corresponding DE algorithms on the CEC2017 functions. Based on multi-problem analysis by the Wilcoxon test, DE-AGM can obtain the higher $R+$ values than $R-$ values in all the cases. In addition, the p values in most cases indicate that DE-AGM significantly outperforms the corresponding DE algorithm overall. In the case of jSO, jSO-AGM also obtains the higher $R+$ values than $R-$ values for all the functions both at 30D and 50D, although no significant differences between them can be observed on most functions. In general, the effectiveness of DE-AGM is further demonstrated on the CEC2017 test functions.

VI. CONCLUSION AND FUTURE RESEARCH

In this study, to take fully advantages of the difference in search behavior between individuals, an adaptive guiding mechanism (AGM) with heuristic rules is proposed and incorporated into DE, resulting in an algorithm framework, DE-AGM. In DE-AGM, an elite leadership team (ELT) is dynamically constructed by selecting the best individuals from the current population. Followed that, distinct individual-dependent elite groups (IEG), separated from ELT, are assigned to different individuals according to their search behaviors. Finally, the search of each individual is guided by selecting elites as leaders from its IEG and building the promising directions as difference vectors from current population. By this way, the utilization of population information can be promoted for guiding the search of DE. Extensive experiments on the CEC2013 and CEC2017 test functions have been carried out to evaluate the effectiveness of DE-AGM. The experimental results have demonstrated the competitive performance of DE-AGM over the original DE algorithms, advanced DE variants and the state-of-the-art EAs considered. In addition, the effectiveness of AGM and the parameter sensitivity of DE-AGM have also been analyzed.

In the future, the adaptive or self-adaptive techniques for setting the size of elites group will be investigated. Moreover, the sophisticated local search operators will be incorporated into the proposed framework to further improve its performance on the complex optimization problems. Finally, the AGM is developed for DE to guide the evolution of population by considering the difference between individuals in search behavior. Thus, whether the proposed AGM are promising for other population-based EAs, is another work for study.

REFERENCES

[1] 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.

[2] R. D. Al-Dabbagh, F. Neri, N. Idris, and M. S. Baba, "Algorithmic design issues in adaptive differential evolution schemes: Review and taxonomy," *Swarm Evol. Comput.*, vol. 43, pp. 284–311, Dec. 2018.

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

[4] S. Das, S. S. Mullick, and P. N. Suganthan, "Recent advances in differential evolution—An updated survey," *Swarm Evol. Comput.*, vol. 27, pp. 1–30, Apr. 2016.

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

[6] R. Tanabe and A. Fukunaga, "Success-history based parameter adaptation for differential evolution," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Cancun, Mexico, Jun. 2013, pp. 71–78.

[7] V. A. Tatsis and K. E. Parsopoulos, "Differential evolution with grid-based parameter adaptation," *Soft Comput.*, vol. 21, no. 8, pp. 2105–2127, 2017.

[8] J. Brest, M. S. Maučec, and B. Bošković, "Single objective real-parameter optimization: Algorithm JSO," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, San Sebastián, Spain, Jun. 2017, pp. 1311–1318.

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

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

[11] G. Wu, X. Shen, H. Li, H. Chen, A. Lin, and P. N. Suganthan, "Ensemble of differential evolution variants," *Inf. Sci.*, vol. 423, pp. 172–186, Jan. 2018.

[12] Z.-H. Zhan *et al.*, "Cloudde: A heterogeneous differential evolution algorithm and its distributed cloud version," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 3, pp. 704–716, 2018.

[13] Y. Cai, M. Zhao, J. Liao, T. Wang, H. Tian, and Y. Chen, "Neighborhood guided differential evolution," *Soft Comput.*, vol. 21, no. 16, pp. 4769–4812, Aug. 2017.

[14] G. Sun, Y. Cai, T. Wang, H. Tian, C. Wang, and Y. Chen, "Differential evolution with individual-dependent topology adaptation," *Inf. Sci.*, vol. 450, pp. 1–38, Jun. 2018.

[15] B. Dorronsoro and P. Bouvry, "Improving classical and decentralized differential evolution with new mutation operator and population topologies," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 67–98, Feb. 2011.

[16] Y. Wang, B. Li, and T. Weise, "Estimation of distribution and differential evolution cooperation for large scale economic load dispatch optimization of power systems," *Inf. Sci.*, vol. 180, no. 12, pp. 2405–2420, 2010.

[17] B. Xin, J. Chen, J. Zhang, H. Fang, and Z.-H. Peng, "Hybridizing differential evolution and particle swarm optimization to design powerful optimizers: A review and taxonomy," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 5, pp. 744–767, Sep. 2012.

[18] H. Fang, A. Zhou, and H. Zhang, "Information fusion in offspring generation: A case study in DE and EDA," *Swarm Evol. Comput.*, vol. 2, pp. 99–108, Oct. 2018.

[19] Y. Cai and J. Wang, "Differential evolution with neighborhood and direction information for numerical optimization," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 2202–2215, Dec. 2013.

[20] J. Wang, J. Liao, Y. Zhou, and Y. Cai, "Differential evolution enhanced with multiobjective sorting-based mutation operators," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 2792–2805, Dec. 2014.

[21] H.-Y. Fan and J. Lampinen, "A trigonometric mutation operation to differential evolution," *J. Global Optim.*, vol. 27, no. 1, pp. 105–129, 2003.

[22] L. Cui, G. Li, Q. Lin, J. Chen, and N. Lu, "Adaptive differential evolution algorithm with novel mutation strategies in multiple sub-populations," *Comput. Oper. Res.*, vol. 67, pp. 155–173, Mar. 2016.

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

[24] Y. Cai, C. Shao, H. Zhang, S. Fu, H. Tian, and Y. Chen, "A neighborhood-assisted framework for differential evolution," *IEEE Access*, vol. 7, pp. 44338–44358, 2019.

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

[26] L. Tang, Y. Dong, and J. Liu, "Differential evolution with an individual-dependent mechanism," *IEEE Trans. Evol. Comput.*, vol. 19, no. 4, pp. 560–574, Aug. 2015.

[27] J. Liang, B. Y. Qu, P. N. Suganthan, and A. G. Hernández-Díaz, "Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization," Comput. Intell. Lab., Zhengzhou Univ., Zhengzhou China and Nanyang Technol. Univ., Singapore, Tech. Rep. 201212, Jan. 2013.

[28] N. H. Awad, M. Z. Ali, B. Y. Qu, J. J. Liang, and P. N. Suganthan, "Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective bound constrained real-parameter numerical optimization," Nanyang Technol. Univ., Singapore, Tech. Rep., Nov. 2016.

[29] S. Islam, S. Das, S. Ghosh, S. Roy, and P. Suganthan, "An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization," *IEEE Trans Syst, Man, Cybern. B, Cybern*, vol. 42, no. 2, pp. 482–500, Apr. 2012.

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

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

[32] H. Wang, S. Rahnamayan, H. Sun, and M. G. H. Omran, "Gaussian bare-bones differential evolution," *IEEE Trans. Cybern.*, vol. 43, no. 2, pp. 634–647, Apr. 2013.

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

[34] L. M. Zheng, S. X. Zhang, K. S. Tang, and S. Y. Zheng, "Differential evolution powered by collective information," *Inf. Sci.*, vol. 399, pp. 13–29, Aug. 2017.

[35] L. M. Zheng, L. Liu, S. X. Zhang, and S. Y. Zheng, "Enhancing differential evolution with interactive information," *Soft Comput.*, vol. 22, no. 23, pp. 7919–7938, 2018.

[36] G. Wu, R. Mallipeddi, P. N. Suganthanc, R. Wang, and H. Chen, "Differential evolution with multi-population based ensemble of mutation strategies," *Inf. Sci.*, vol. 329, pp. 329–345, 2016.

[37] X.-F. Liu *et al.*, "Historical and heuristic-based adaptive differential evolution," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published.

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

[39] S. X. Zhang, L. M. Zheng, K. S. Tang, S. Y. Zheng, and W. S. Chan, "Multi-layer competitive-cooperative framework for performance enhancement of differential evolution," *Inf. Sci.*, vol. 482, pp. 86–104, May 2019.

[40] 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.

[41] B. Xu, W. Cheng, F. Qian, and X. Huang, "Self-adaptive differential evolution with multiple strategies for dynamic optimization of chemical processes," *Neural Comput. Appl.*, 2019. doi: 10.1007/s00521-018-03985-x.

[42] Z.-J. Wang *et al.*, "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.

[43] X.-G. Zhou and G.-J. Zhang, "Differential evolution with underestimation-based multimutation strategy," *IEEE Trans. Cybern.*, vol. 49, no. 4, pp. 1353–1364, Apr. 2019.

[44] Y. Wang, Z.-Z. Liu, J. Li, H.-X. Li, and J. Wang, "On the selection of solutions for mutation in differential evolution," *Frontiers Comput. Sci.*, vol. 12, no. 2, pp. 297–315, 2018.

[45] M. Epitropakis, D. Tasoulis, N. Pavlidis, V. Plagianakos, and M. Vrahatis, "Enhancing differential evolution utilizing proximity-based mutation operators," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 99–119, Feb. 2011.

[46] Y. Cai, G. Sun, T. Wang, H. Tian, Y. Chen, and J. Wang, "Neighborhood-adaptive differential evolution for global numerical optimization," *Appl. Soft Comput.*, vol. 59, pp. 659–706, Sep. 2017.

[47] S.-M. Guo, C.-C. Yang, P.-H. Hsu, and J. S. H. Tsai, "Improving differential evolution with a successful-parent-selecting framework," *IEEE Trans. Evol. Comput.*, vol. 19, no. 5, pp. 717–730, Oct. 2015.

[48] Y. Zhou, J. Wang, Y. Zhou, Z. Qiu, Z. Bi, and Y. Cai, "Differential evolution with guiding archive for global numerical optimization," *Appl. Soft Comput.*, vol. 43, pp. 424–440, Jun. 2016.

[49] K.-C. Wong, K.-S. Leung, and M.-H. Wong, "Effect of spatial locality on an evolutionary algorithm for multimodal optimization," in *Proc. Eur. Conf. Appl. Evol. Comput.*, Berlin, Germany: Springer, 2010, pp. 481–490.

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

[51] 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.

[52] 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.

[53] L. Li and K. Tang, "History-based topological speciation for multimodal optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 1, pp. 136–150, Feb. 2015.

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

[55] S. Garc "ia, A. Fernández, J. Luengo, and F. Herrera, "A study of statistical techniques and performance measures for genetics-based machine learning: Accuracy and interpretability," *Soft Comput*, vol. 13, no. 10, pp. 959–977, 2009.

[56] 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.

[57] S. García, D. Molina, M. Lozano, and F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: A case study on the CEC'2005 special session on real parameter optimization," *J. Heuristics*, vol. 15, no. 6, pp. 617–644, 2009.

[58] S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, "Opposition-based differential evolution," *IEEE Trans. Evol. Comput.*, vol. 12, no. 1, pp. 64–79, Feb. 2008.

[59] F. Caraffini, G. Iacca, F. Neri, L. Picinali, and E. Mininno, "A CMA-ES super-fit scheme for the re-sampled inheritance search," in *Proc. IEEE Congr. Evol. Comput.*, Cancun, Mexico, Jun. 2013, pp. 1123–1130.

[60] F. V. Nepomuceno and A. P. Engelbrecht, "A self-adaptive heterogeneous PSO for real-parameter optimization," in *Proc. IEEE Congr. Evol. Comput.*, Cancun, Mexico, Jun. 2013, pp. 361–368.

[61] S. M. Elsayed, R. A. Sarker, and D. L. Essam, "A genetic algorithm for solving the CEC'2013 competition problems on real-parameter optimization," in *Proc. IEEE Congr. Evol. Comput.*, Cancun, Mexico, Jun. 2013, pp. 356–360.

[62] M. El-Abd, "Testing a particle swarm optimization and artificial bee colony hybrid algorithm on the CEC13 benchmarks," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Cancun, Mexico, Jun. 2013, pp. 2215–2220.
[63] P. Korošec and J. Šilc, "The continuous differential ant-stigmergy algorithm applied on real-parameter single objective optimization problems," in *Proc. IEEE Congr. Evol. Comput.*, Cancun, Mexico, Jun. 2013, pp. 1658–1663.
[64] J. L. Rueda and I. Erlich, "Hybrid mean-variance mapping optimization for solving the IEEE-CEC 2013 competition problems," in *Proc. IEEE Congr. Evol. Comput.*, Cancun, Mexico, Jun. 2013, pp. 1664–1671.

YIQIAO CAI received the B.S. degree from Hunan University, Changsha, China, in 2007, and the Ph.D. degree from Sun Yat-sen University, Guangzhou, China, in 2012. He is currently an Associate Professor with the College of Computer Science and Technology, Huaqiao University. His research interests include differential evolution, multiobjective optimization, and other evolutionary computation techniques.

CHI SHAO received the B.S. degree from the Tianjin University of Commerce, Tianjin, China, in 2015. She is currently pursuing the M.Sc. degree with the College of Computer Science and Technology, Huaqiao University, Xiamen, China. Her research interests include differential evolution, evolutionary computation, and machine learning.

YING ZHOU received the B.S. and Ph.D. degrees from Sun Yat-sen University, Guangzhou, China, in 2009 and 2014, respectively. She is currently a Lecturer with the School of Computer Sciences, Shenzhen Institute of Information Technology. Her research interests include local search algorithms and their applications, multiobjective optimization, and other evolutionary computation techniques.

SHUNKAI FU received the B.S. degree from Chongqing University, Chongqing, China, in 2001, and the B.Sc. and Ph.D. degrees from the University of Montreal, Canada, in 2005 and 2010, respectively. He is currently a Lecturer with the College of Computer Science and Technology, Huaqiao University. His research interests include data mining, intelligent recommendation algorithm and its application, and the mobile Internet application.

HUIZHEN ZHANG received the B.S. and Ph.D. degrees from the University of Science and Technology of China, Hefei, China, in 2005 and 2010, respectively. He is currently an Associate Professor with the College of Computer Science and Technology, Huaqiao University. His research interests include computer architecture, hardware/software optimization, and reconfigurable accelerator.

HUI TIAN received the B.Sc. and M.Sc. degrees from the Wuhan Institute of Technology, Wuhan, China, in 2004 and 2007, respectively, and the Ph.D. degree from the Huazhong University of Science and Technology, Wuhan, China, in 2010. He is currently a Professor with the College of Computer Science and Technology, Huaqiao University. His current research interests include network and multimedia information security, digital forensics, and information hiding.