

Competition-Driven Dandelion Algorithms With Historical Information Feedback

Shoufei Han^{1b}, Kun Zhu^{1b}, *Member, IEEE*, and MengChu Zhou^{1b}, *Fellow, IEEE*

Abstract—A Dandelion algorithm (DA) inspired by the seed dispersal process of dandelions has been proposed as a newly intelligent optimization algorithm. For improving its exploration ability as well as reducing the probability of its falling into a local optimum, this work proposes to add a novel competition mechanism with historical information feedback to current DA. Specifically, the fitness value of each dandelion in the next generation, which is calculated by linear prediction, is compared with the current best dandelion, and the loser is replaced by a new offspring. Current DA generates new offsprings without considering historical information. This work improves its offspring generation process by exploiting historical information with an estimation-of-distribution algorithm. Three historical information models are designed. They are best, worst, and hybrid historical information feedback models. The experimental results show that the proposed algorithms outperform DA and its variants, and the proposed algorithms are superior or competitive to nine participating algorithms benchmarked on 28 functions from CEC2013. Finally, the proposed algorithms demonstrate the effectiveness on four real-world problems, and the results indicate that the proposed algorithms have better performance than its peers.

Index Terms—Applications, competition mechanism, dandelion algorithm (DA), estimation-of-distribution algorithm (EDA), historical information, intelligent optimization, linear prediction, machine learning.

I. INTRODUCTION

NOWADAYS, optimization problems are becoming more complex and difficult. Their aims are to find a set of parameter values that satisfy the constraints while optimizing their single or multiple performance indicators. Their solution

methods can be roughly divided into two categories: 1) exact methods and 2) intelligent optimization algorithms [1]. The former can get the optimal solution and always generates the same results for different runs under the same conditions. The latter often provides a near-optimal solution and tends to generate different solutions under the same conditions. For large-scale problems, the former is time-consuming and often impossible, while the latter can provide a satisfactory solution in a reasonable time. Therefore, many intelligent optimization algorithms have been proposed including genetic algorithm [2], differential evolution [3], cultural algorithm [4], particle swarm optimization (PSO) [5], ant colony optimization [6], artificial bee colony (ABC) [7], bat algorithm [8], grey wolf algorithm [9], harmony search [10], brain storm optimization [11], fireworks algorithm [12], and teaching-learning-based optimization (TLBO) [13]. These algorithms have been applied in many fields [14]–[17].

A dandelion algorithm (DA) [18] is recently proposed, which has been applied to optimize an extreme learning machine for biomedical classification problems. In DA, there are two types of dandelions, they are the best dandelion (BD) and assistant ones (ADs). The dandelions in different types have different seed dispersal ways. Due to the advantages, such as simplicity and efficiency, DA has attracted attentions of researchers, and some variants have been proposed [19], [20]. In [19], an improved DA is proposed (abbreviated as MDA), in which the convergence and parameters of DA are studied, and the mean position of all seeds is used to replace the Levy mutation. In [20], a variant of DA, i.e., a DA with probability-based mutation (DAPM), aiming to improve its mutation ability is proposed, in which three probability models are presented to switch among different mutations, and DAPM with three probability models is abbreviated as DAPML (linear), DAPMB (binomial), and DAPME (exponential), respectively.

Note that exploitation and exploration significantly impact the performance of intelligent optimization algorithms [21]. Excessive exploitation can easily lead them to local optima, while excessive exploration causes their difficulty to find the optimal solution. Cooperation could enhance their exploitation ability, and competition can improve their exploration ability. Thus, the phenomenon of cooperation ([5], [12], [22]), competition ([2], [9]) and their combination ([3], [23], [24]) often appear when designing such algorithms [25].

In this article, we design a novel competition mechanism for DA to enhance its exploration ability and reduce the probability of falling into local optima. This mechanism is based

Manuscript received January 5, 2020; revised April 8, 2020; accepted July 13, 2020. Date of publication August 5, 2020; date of current version January 17, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 61701230; in part by the Natural Science Foundation of Jiangsu Province under Grant BK20170805; and in part by the Fundamental Research Funds for the Central Universities under Grant NE2018107. This article was recommended by Associate Editor M. K. Tiwari. (Corresponding authors: Kun Zhu; MengChu Zhou.)

Shoufei Han and Kun Zhu are with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China, and also with the Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing 211106, China (e-mail: hanshoufei@gmail.com; zhukun@nuaa.edu.cn).

MengChu Zhou is with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA, and also with the Institute of Systems Engineering and Collaborative Laboratory for Intelligent Science and Systems, Macau University of Science and Technology, Macau 999078, China (e-mail: zhou@njit.edu).

This article has supplementary material provided by the authors and color versions of one or more figures available at <https://doi.org/10.1109/TSMC.2020.3010052>.

Digital Object Identifier 10.1109/TSMC.2020.3010052

on the linear prediction of fitness values. The predicted fitness value of a dandelion is compared with the best one, and a losing dandelion (loser) should be abandoned and replaced with a new offspring to avoid wasting resources on searching unpromising areas. Thus, how to generate its new offspring is an another important issue.

The generation of new offspring in a current DA ignores the historical information including historically best and worst dandelions, which contain rich information. In this article, we reuse such information to generate new offspring and propose three historical information feedback model-based DAs, which are best-historical-information-feedback-model-based DA (BDA), worst-historical-information-feedback-model-based DA (WDA), and hybrid-historical-information-feedback-model-based DA (HDA). In the proposed algorithms, we use the estimation-of-distribution algorithm (EDA) [26], instead of using traditional historical feedback methods, which tend to consume large memory, to estimate the distribution information of historical information and generate new offsprings.

The main contributions of this article are as follows.

- 1) A competition mechanism is proposed to improve the exploration ability of DA and to reduce the probability of DA to fall into its local optima.
- 2) Three historical information feedback models based on EDA are proposed to generate new offsprings, which are used to replace losers in the competition.
- 3) Our proposed algorithms are compared with DA's variants and other intelligent optimization algorithms on CEC2013 standard functions to show their superiority.
- 4) The proposed algorithms are applied to four real-world problems and the results show their superiority over other state-of-the-art methods.

The remainder of this article is organized as follows. Related studies about competition and information feedback in intelligent optimization algorithms are introduced in Section II. Section III introduces a framework and the operators of DA. Our proposed algorithms are presented and discussed in Section IV. Section V provides the mathematical analyses. Experimental and comparison results are given in Section VI. Section VII concludes this article.

II. RELATED WORK

A. Competition Mechanisms in Intelligent Optimization

Competition mechanisms have been widely applied to design intelligent optimization algorithms. Genetic algorithm is one of the most popular ones [2]. Its selected individuals are directly inherited to the next generation via a competitive mechanism. The greedy selection is adopted in differential evolution [3]. The individual with the better fitness is retained, and the loser is abandoned. The selection strategy of fireworks algorithm always keeps the best fireworks into the next generation [12]. The competition mechanisms in these algorithms are due to a selection mechanism. Some other algorithms without a selection process adopt a competition mechanism. In PSO [5], choosing the *gBest* and *pBest* from particles may be viewed as a competition mechanism. In the grey wolf algorithm [9], the selection of α , β , and δ wolves also involves

a competition mechanism. Different competition mechanisms are applied to different heuristic algorithms, but they have one thing in common: the best or better individual is selected.

In DA, a competition mechanism is triggered due to selection, i.e., BD in the population is always kept into the next generation. In this article, a novel competition mechanism is designed for DA to improve its exploration ability, which is different from the existing one of DA.

B. Information Feedback in Intelligent Optimization

In recent years, team of scholars have exploited feedback information from various sources to improve the performance of intelligent optimization algorithms. Reusing valuable information to guide current and future search shows promising effects for them. In this section, related studies are briefly reviewed.

In the original ABC [7], the previous bee colony history information is not reused. Therefore, a bare bones ABC is proposed [27], which generates candidates by combining the Gaussian distribution and the best bee at the onlooker phase, and a neighborhood mechanism is introduced to enhance the search ability at the employed bee phase.

In order to make full use of the historical information of all particles in PSO, Li *et al.* [28] proposed a new composite PSO. Therein, EDA is first used to estimate and preserve the historical promising distribution information of particles to construct a historical information pool, which can generate new individuals to guide the evolution of PSO. A comprehensive learning PSO is proposed by using all other particles' historical best information [5], [29], which is the most popular algorithm among PSO variants.

TLBO is a promising intelligent optimization algorithm. In order to improve the convergence speed and accuracy, a novel TLBO is proposed by using empirical information to help learners judge learning behavior [30]. In addition, in order to improve the global performance of TLBO, Zou *et al.* [31] proposed an improved TLBO algorithm by using other learners' learning experience.

In order to make full use of the spark information generated by the fireworks algorithm at an explosion process, a novel guiding strategy is designed [32]. Therein, the sparks are sorted according to the fitness value, and then a guide vector is constructed by calculating the difference between the top ones and remaining sparks.

Wang *et al.* [33] proposed a variable neighborhood bat algorithm. Therein, a bat can be guide to search by its neighbors. In addition, Wang and Tan [1] presented a novel method to reuse the valuable information provided by previous individuals to guide future search.

Surrogate-assisted evolutionary algorithms (SAEAs) have been developed mainly for solving the optimization problems that have high cost in computing fitness values [34]. Their main idea is to replace expensive fitness evaluations with computationally cheap surrogate that can be constructed based on historical information [35]–[37].

Different from the previous studies, instead of simply reusing historical information, a competition mechanism is designed to judge whether historical information needs to be

reused, and only the losers in the competition need be guided by historical information. Meanwhile, in this article, three historical information feedback models are constructed and used to improve the performance of DA.

III. OVERVIEW OF DA

For convenience to illustrate DA, we only consider the following minimization problem in this article:

$$y = \min f(\mathbf{x}) \quad (1)$$

where \mathbf{x} is a D -dimensional solution, and the objective is to find a suitable \mathbf{x} such that $f(\mathbf{x})$ reaches its minimum.

Before introducing the rationale of DA, we need to explain some proper nouns, in order to help the reader better understand the technical details. The number of seeds represents the number of solutions; dispersal radius represents step size, i.e., search range; search space is the solution domain; and position stands for a solution.

Like other algorithms, DA can be divided into two phases: 1) initialization and 2) iterative optimization. In initialization, we generate dandelions in search space randomly as the initial population. Iterative optimization consists of normal seed dispersal, mutation seed dispersal, and selection, as introduced next.

A. Normal Seed Dispersal

In normal seed dispersal, dandelions generate a certain number of seeds within dispersal radius. Thus, the number of seeds generated by each dandelion is calculated, and then the dispersal radius is computed. Finally, the process of generating normal seeds is given.

Consider N dandelions in a D -dimensional search space. The i th dandelion at generation t has a current position $\mathbf{x}_i(t) = (x_i^1(t), \dots, x_i^d(t), \dots, x_i^D(t))$. In DA, there are two types of dandelions: BD (the one with the minimum fitness) and ADs (other than the best one).

The number of the i th dandelion seeds at the t -th generation is

$$C_i(t) = \Phi\left(\hat{C} \times \frac{\hat{y}(t) - f(\mathbf{x}_i(t)) + \varepsilon}{\hat{y}(t) - \check{y}(t) + \varepsilon}\right) \quad (2)$$

where C is the number of seeds, and \hat{C} is the maximum number of seeds. The purpose of the function $\Phi()$ is to round a variable to its nearest integer, and $\hat{y}(t)$ and $\check{y}(t)$ are the maximum and minimum fitness values at the t -th generation, respectively. $f(\mathbf{x}_i(t))$ is the fitness value of the i th dandelion at the t -th generation, and ε is given to avoid a denominator of zero. Note that C_i is set to \hat{C} when $C_i < \hat{C}$, where \hat{C} is the minimum number of seeds.

The dispersal radius of BD at the t -th generation is

$$R_B(t) = \begin{cases} U - L, & t = 1 \\ R_B(t-1) \times \gamma, & a = 1 \\ R_B(t-1) \times \nu, & a \neq 1 \end{cases} \quad (3)$$

where U and L are, respectively, the upper and lower bounds of the search space, while γ and ν are two factors to control the enlargement and reduction of the radius. When $t = 1$, the

Algorithm 1 Generating Normal Seeds for \mathbf{x}_i

Input: $\mathbf{x}_i, C_i, R_B, R_i$

Output: all s_{ik}

```

1: for  $k = 1$  to  $C_i$  do
2:   for  $d = 1$  to  $D$  do
3:     Randomly take a value for  $\theta$  in  $(0, 1)$ 
4:     if  $\theta < 0.5$  then
5:       Randomly take a value for  $\varphi$  in  $(-1, 1)$ 
6:       if  $\mathbf{x}_i == \mathbf{x}_B$  then
7:          $s_{ik}^d = \mathbf{x}_i^d + \varphi \cdot R_B$ 
8:       else
9:          $s_{ik}^d = \mathbf{x}_i^d + \varphi \cdot R_i$ 
10:      end if
11:      if  $s_{ik}^d$  out of bounds then
12:        Randomly take a value for  $s_{ik}^d$  in  $[L, U]$ 
13:      end if
14:    else
15:       $s_{ik}^d = \mathbf{x}_i^d$ 
16:    end if
17:  end for
18: end for

```

dispersal radius is initialized to the diameter of the search space. Parameter a is defined as

$$a = \frac{\check{y}(t) + \varepsilon}{\check{y}(t-1) + \varepsilon} \quad (4)$$

where $\check{y}(t)$ and $\check{y}(t-1)$ are the minimum fitness value in generation t and $t-1$, respectively. When $a = 1$, i.e., $\check{y}(t) = \check{y}(t-1)$, which means a better solution is not found in generation t . In contrast, $a \neq 1$ means that $\check{y}(t) \neq \check{y}(t-1)$, i.e., a better solution is found in generation t .

The dispersal radius of the i th dandelion belonging to ADs at the t -th generation is

$$R_i(t) = \begin{cases} U - L, & t = 1 \\ \omega \times R_i(t-1) + (\|\mathbf{x}_B\|_\infty - \|\mathbf{x}_i\|_\infty), & \text{otherwise} \end{cases} \quad (5)$$

where \mathbf{x}_B is the position of BD, and ω is the weight factor, which is designed to control the effect of the dispersal radii between the previous and current generations dynamically. It is calculated as

$$\omega = 1 - \frac{F_E(t)}{F_M} \quad (6)$$

where $F_E(t)$ is the number of function evaluations at generation t , and F_M is the maximum number of function evaluations.

Algorithm 1 shows how the seeds are generated for each dandelion. Therein, some dimensions are chosen to update their positions based on dispersal radius (lines 5–10); while others remain unchanged (line 15). Meanwhile, note that position updates of BD and ADs are different (lines 7 and 9) due to different dispersal radii.

B. Mutation Seed Dispersal

To help jump out of local optima, Levy mutation is selected to update the position of BD, and this process is called

Algorithm 2 Generating Mutation Seeds for *BD***Input:** z, \mathbf{x}_B **Output:** all \mathbf{m}_j

```

1: for  $j = 1$  to  $z$  do
2:   for  $d = 1$  to  $D$  do
3:     Randomly take a value for  $\theta$  in  $(0, 1)$ 
4:     if  $\theta < 0.5$  then
5:        $\mathbf{m}_j^d = \mathbf{x}_B^d \cdot (1 + \text{Levy}())$ 
6:       if  $\mathbf{m}_j^d$  out of bounds then
7:         Randomly take a value for  $\mathbf{m}_j^d$  in  $[L, U]$ 
8:       end if
9:     else
10:       $\mathbf{m}_j^d = \mathbf{x}_B^d$ 
11:    end if
12:  end for
13: end for

```

mutation seed dispersal, i.e.,

$$\mathbf{m} = \mathbf{x}_B \cdot (1 + \text{Levy}()) \quad (7)$$

where \mathbf{m} is the position of a mutation seed, and $\text{Levy}()$ is a random number following Levy distribution.

Algorithm 2 shows the process to generate the mutation seeds for BD, where z is the number of mutation seeds. Therein, same as Algorithm 1, some dimensions are chosen to update their positions based on Levy mutation (line 5); while others remain unchanged (line 10). Note that mutation seed dispersal is only for BD but not for ADs.

C. Selection

In this article, in order to implement a fair competition, we design a new selection strategy based on a greedy strategy. Its core idea is that a dandelion and its own seeds are considered as a single population and greedily select the best individual in the population into the next generation

$$\mathbf{x}_i(t+1) = \begin{cases} \arg \min \{f(\mathbf{x}_i(t)), f(\mathbf{s}_{ik}), f(\mathbf{m}_j)\}, & \mathbf{x}_i = \mathbf{x}_B \\ \arg \min \{f(\mathbf{x}_i(t)), f(\mathbf{s}_{ik})\}, & \text{otherwise} \end{cases} \quad (8)$$

where $f(\mathbf{s}_{ik})$ is the fitness values of normal seeds generated by the i th dandelion, and $f(\mathbf{m}_j)$ is the fitness values of mutation seeds generated by BD.

Algorithm 3 summarizes the framework of DA. Therein, N dandelions are generated randomly in search space as initial population (line 1), and then each dandelion generates its normal seeds by Algorithm 1 (line 4), while BD is selected to generate the mutation seeds by Algorithm 2 (line 5). Finally, selecting N dandelions into the next generation from all dandelions (dandelions, normal seeds and mutation seeds) via a selection strategy. BD is returned when the termination condition is satisfied.

IV. PROPOSED ALGORITHMS

In this section, we introduce the proposed algorithms in detail. First, a novel competition mechanism is designed, and then three historical information feedback models are proposed

Algorithm 3 Framework of DA**Input:** N **Output:** Best dandelion

```

1: Randomly generating  $N$  dandelions as initial population
2: Assess  $N$  dandelions' fitness
3: while a termination condition is not satisfied do
4:   Generating normal seeds according to Algorithm 1
5:   Generating mutation seeds according to Algorithm 2
6:   Assess the fitness of all seeds
7:   Select  $N$  dandelions into the next generation via a selection strategy
8: end while

```

based on EDA. Finally, we give a complete framework of the proposed algorithms.

A. Competition Mechanism

In this article, our competition mechanism is designed based on fitness values, and if the fitness value of a dandelion in the next generation, which can be predicted by evolutionary speed e , is not better than that of the current BD, it is considered as a loser and then is replaced. Define e in generation t as

$$\bar{f}(t-1) = \frac{1}{N} \sum_{i=1}^N f(\mathbf{x}_i(t-1)) \quad (9)$$

$$e(t) = \bar{f}(t-1) - \check{y}(t) \quad (10)$$

where $\bar{f}(t-1)$ is the average fitness value of dandelions in generation $t-1$ and $\check{y}(t)$ is the minimum fitness value of dandelions in generation t . e is the difference between $\bar{f}(t-1)$ and $\check{y}(t)$ reflecting how fast dandelions evolve.

From (10), if e is large, which means that DA evolve quickly, few dandelions are the losers. It becomes smaller when the algorithm falls into a local or global optimum.

However, how to estimate the fitness value of dandelions in generation $t+1$ is a challenge. To address this, we introduce a simple linear model to predict the fitness value based on the fitness values in generation t and e . It is designed as follows:

$$\tilde{f}(\mathbf{x}_i(t+1)) = f(\mathbf{x}_i(t)) - e \times \delta \quad (11)$$

where δ is the learning rate, and its detailed empirical analysis is given in Section VI-B.

For each dandelion \mathbf{x}_i , the approximate fitness value in the next generation could be calculated according to (11). The framework to realize the above competition mechanism is shown in Algorithm 4.

In Algorithm 4, line 2 is a trigger condition to use the proposed competition mechanism, which means that the mechanism is triggered if a better solution is found. On the contrary, if a better solution is not found, the mechanism should not be triggered because all dandelions are losers. Furthermore, in DA evolution, the above situation often occurs when a better solution cannot be found, thus indicating that the losers are often replaced if a trigger condition is deleted, and DA loses its current potential area. Thus, a trigger condition is necessary for starting a competition mechanism.

Algorithm 4 Competition Mechanism**Input:** N (N is the number of dandelion)**Output:** the position and the fitness of losers

```

1: for  $i = 1$  to  $N$  do
2:   if  $\frac{\check{y}(t)}{\check{y}(t-1)} \neq 1$  then
3:      $\bar{f}(t-1) = \frac{1}{N} \sum_{i=1}^N f(\mathbf{x}_i(t-1))$ 
4:      $e = \bar{f}(t-1) - \check{y}(t)$ 
5:     if  $f(\mathbf{x}_i(t)) - e \times \delta > \check{y}(t)$  then
6:        $\mathbf{x}_i(t)$  is the loser
7:     end if
8:   end if
9: end for

```

Lines 3 and 4 calculate the evolutionary speed e . Instead of using average fitness value, if the maximum fitness value is used, e would be larger, and $f(\mathbf{x}_i(t)) - e \times \delta$ would be small, which means that there are few losers to be replaced and it is more difficult to jump out of a local optimum. Similarly, if using the minimum fitness value, e is small, and $f(\mathbf{x}_i(t)) - e \times \delta$ would be large, which means that there are many losers to be replaced and DA may lose its current promising area.

Line 5 is to determine whether a dandelion is a loser. Given e and δ , if the condition of line 5 is satisfied, it is considered a loser. In the following, the effectiveness of the proposed competition mechanism is illustrated by comparing the resulting DA with a single-population DA.

For example, there is a function to be optimized, and assume it has p local optima and one global optimum. For single-population DA, the probability of finding the global optimum with q trials is calculated as follows:

$$P_{sp} = \frac{p-1}{p} \cdot \frac{p-1}{p} \cdot \dots \cdot \frac{p-1}{p} \cdot \frac{1}{p} = \left(\frac{p-1}{p}\right)^{q-1} \cdot \frac{1}{p}. \quad (12)$$

With the proposed competition mechanism in multipopulation DA, the probability of falling into the same local optima is reduced, and the probability of finding the global optimum with q trials increases, i.e.,

$$P_{cs} = \frac{p-1}{p} \cdot \frac{p-2}{p-1} \cdot \dots \cdot \frac{p-q+1}{p-q+2} \cdot \frac{1}{p-q+1} = \frac{1}{p}. \quad (13)$$

From (12) and (13), we conclude that the probability of finding the global optimum $P_{cs} > P_{sp}$ when the number of trials $q > 1$. It indicates that the proposed mechanism is effective for jumping out of a local optimum.

Line 6 is to replace the loser with a new offspring generated by a historical information feedback model. Here, we must consider the following case: if a loser is not replaced, it may be BD in the next generation, and if it is replaced, it is harmful to the search process.

But we find that the following conditional probability is low:

$$P(\mathbf{x}_i(t+1) \text{ is BD} | \mathbf{x}_i(t) \text{ is a loser})$$

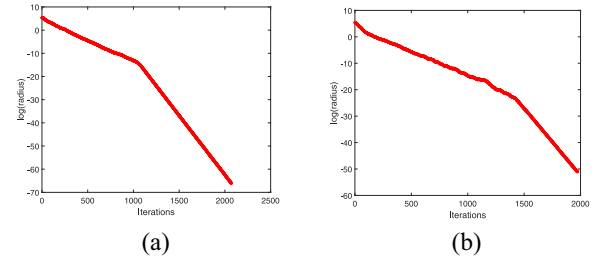


Fig. 1. Trend of dispersal radius R of BD on test functions (a) $f1$ and (b) $f23$ from CEC2013 in Table I.

where $\mathbf{x}_i(t)$ is the loser in generation t . The reasons are as follows: 1) DA does not have an improvement in every generation, so that the condition in line 2 is not always triggered and 2) in DA, the dispersal radius of BD is dynamically changing, whose trend is shown in Fig. 1, where the differences between Fig. 1(a) and (b) are that the trends of dispersal radius are different on different functions, numbers on the horizontal and vertical axes represent iteration count, and logarithm of dispersal radius, respectively. From Fig. 1(a), it can be seen that the overall trend of the radius is decreasing, and the speed of decline is faster in the later stage than that in the early stage. From Fig. 1(b), it can be observed that the radius drops slowly in the early stage, but the speed of decline has accelerated when iteration count is between 1500 and 2000. Overall, the radius has a clear downward trend in a whole process, and the number of drops is significantly more than the number of rises, which means that the current BD is also a BD in the next generation in most cases. Thus, the probability of $\mathbf{x}_i(t+1)$ being BD is low, and thus the above conditional probability is also low.

Unavoidably, losers may also be wrongly replaced, and this case is considered to be acceptable because BD is always retained in every generation. On the other hand, the proposed competition mechanism is effective because the losers are directly replaced without wasting resources, i.e., fitness evaluations. Meanwhile, the replacements can search in (disperse into) an unknown area, which enhances the exploration of DA.

B. Historical Information Feedback Models

In an evolutionary process of DA, historical information may be reused to guide its search process. If all historical information of a particular individual, i.e., BDs, is kept, it needs large memory space. In order to solve this problem, EDA is used to model the distribution of historical information. Here, we only consider the historical information of the best and worst dandelions. In the following, EDA is first introduced to construct a distribution model and generate new offsprings, and then three historical information feedback models are proposed.

1) *Construct Distribution Model and Generate New Offsprings:* We assume that there are H historical records of the best or worst dandelions. We use EDA to estimate the distribution of H records. For each dandelion \mathbf{x}_i , a distribution of x_i^d follows a Gaussian distribution is assumed, the reasons are as follows: 1) Gaussian distribution is used in this article

Algorithm 5 Constructing a Distribution Model and Generating a New Offspring

Input: H_M , BD**Output:** $\tilde{\mathbf{x}}_B = (x^1, x^2, \dots, x^D)$

```

1: Calculate the number of current historical records:  $L$ 
2: if  $L < H_M$  then
3:   Add BD into historical records
4: else
5:   Replace the worst individual with BD
6: end if
7: for  $d = 1$  to  $D$  do
8:   Calculate the mean:  $\mu_d$ 
9:   Calculate the standard deviation:  $\sigma_d$ 
10:  Generate new dimension  $x^d$  by following  $\mathcal{N}(\mu_d, \sigma_d)$ 
11: end for
  
```

following the suggestions in [28] and [38], in which it is used in EDA to generate new individuals and 2) the new individuals generated by the Gaussian distribution is closer to the mean of samples with a large probability, and thus the new individuals can well simulate the characteristics of the samples. In the Gaussian distribution, the mean and standard deviation need be determined as

$$\mu_d = \frac{1}{H} \sum_{i=1}^H x_i^d \quad (14)$$

$$\sigma_d = \sqrt{\frac{1}{H} \sum_{i=1}^H (x_i^d - \mu_d)^2} \quad (15)$$

where μ_d and σ_d are the mean and standard deviation of the d th attribute of historical information of the best or worst dandelions. Once μ_d and σ_d are determined, the d th attribute of a new offspring is generated which follows the Gaussian distribution $\mathcal{N}(\mu_d, \sigma_d)$. Taking the historical information of the BD as an example, the process of constructing a distribution model and generating new offsprings is shown in Algorithm 5, where H_M is the maximum number of historical records. In Algorithm 5, a distribution model is first constructed, and then a new offspring is generated. Note that for the historical information of the worst dandelion, the process of constructing its distribution model and generating a new offspring is same as the construction process for BD. The only changes are that the input is replaced by the worst dandelion in the current generation.

2) *Three Historical Information Feedback Models*: Based on the new offspring generated by EDA, historical information feedback models can be built. In this section, we present best historical information feedback model (BM), worst one (WM), and hybrid one (HM).

a) *Best historical information feedback model*: The one stores the probability model that characterizes the historically BDs, and then the new offspring $\tilde{\mathbf{x}}_B$ is generated. Based on $\tilde{\mathbf{x}}_B$, the replacement of a loser can be defined as follows:

$$\tilde{\mathbf{x}}_{\bar{B}} = L + U - \tilde{\mathbf{x}}_B \quad (16)$$

In (16), we use the opposite solution of $\tilde{\mathbf{x}}_B$ to replace the loser instead of using $\tilde{\mathbf{x}}_B$. There are two reasons: the first is

that $\tilde{\mathbf{x}}_B$ has the same distribution as the selected BD, and if $\tilde{\mathbf{x}}_B$ is used to replace a loser, it may repeat BDs' previous search process. Using its opposite solution can avoid this situation. Second, an opposite solution has 50% chance of finding a better solution, which can help DA find a better search area.

b) *Worst historical information feedback model*: The one stores the probability model that characterizes the historically worst dandelions, and then the new offspring $\tilde{\mathbf{x}}_W$ is generated. Based on $\tilde{\mathbf{x}}_W$, the replacement of a loser can be defined as follows:

$$\tilde{\mathbf{x}}_{\bar{W}} = L + U - \tilde{\mathbf{x}}_W. \quad (17)$$

Similar to the above, this model uses the opposite solution to replace a loser.

c) *Hybrid historical information feedback model*: This one is the combination of the prior two models, we have

$$\tilde{\mathbf{x}}_H = \alpha_B \cdot \tilde{\mathbf{x}}_B + \alpha_W \cdot (L + U - \tilde{\mathbf{x}}_W) \quad (18)$$

where α_B and α_W are the weighting factors, defined as follows:

$$\alpha_B = \frac{|f(\tilde{\mathbf{x}}_B)|}{|f(\tilde{\mathbf{x}}_B)| + |f(\tilde{\mathbf{x}}_W)|} \quad (19)$$

$$\alpha_W = \frac{|f(\tilde{\mathbf{x}}_W)|}{|f(\tilde{\mathbf{x}}_B)| + |f(\tilde{\mathbf{x}}_W)|} \quad (20)$$

where $|f(\tilde{\mathbf{x}}_B)|$ and $|f(\tilde{\mathbf{x}}_W)|$ are the absolute value of fitness of $\tilde{\mathbf{x}}_B$ and $\tilde{\mathbf{x}}_W$, respectively. The following outlines a theoretical analysis to show the effectiveness of replacing a loser with $\tilde{\mathbf{x}}_H$.

Consider function $f = x^2$. Assume that x^d is the value of an attribute of f , and R is the dispersal radius. Then x^d takes a value in $[-R, R]$ randomly. The following theorem presents the error bound of irrelevant directions.

Theorem 1: If x^d follows a uniform distribution $\mathcal{U}(-R, R)$, and \tilde{x}_H is generated by (18). For any $D \neq 1$, with probability at least $1 - p$, the error bound of irrelevant directions can be obtained as

$$|\tilde{x}_H^d| \leq \sqrt{\frac{R^2}{3p}}. \quad (21)$$

Proof: Since $x^d \sim \mathcal{U}(-R, R)$, so $\tilde{x}_H^d \sim \mathcal{U}(-R, R)$. Then, the expectation of \tilde{x}_H^d is

$$E(\tilde{x}_H^d) = 0. \quad (22)$$

The variance is

$$D(\tilde{x}_H^d) = E((\tilde{x}_H^d)^2) + E(\tilde{x}_H^d)^2. \quad (23)$$

Since $E(\tilde{x}_H^d)^2 = 0$, we have

$$\begin{aligned} D(\tilde{x}_H^d) &= E((\tilde{x}_H^d)^2) \\ &= \int_{-R}^R (\tilde{x}_H^d)^2 \frac{1}{2R} d\tilde{x}_H^d \\ &= \frac{1}{2R} \int_{-R}^R (\tilde{x}_H^d)^2 d\tilde{x}_H^d \\ &= \frac{1}{2R} \frac{(\tilde{x}_H^d)^3}{3} \Big|_{-R}^R \\ &= \frac{1}{2R} \frac{R^3 + R^3}{3} = \frac{R^2}{3} \end{aligned} \quad (24)$$

where $(1/2R)$ is the probability density function for \tilde{x}_H^d .

Algorithm 6 Framework of the Proposed Algorithms**Input:** N **Output:** Best dandelion

- 1: Randomly generating N dandelions
- 2: Assess the fitness of dandelions
- 3: **while** a termination condition is not satisfied **do**
- 4: Generating normal seeds by Alg.1
- 5: Generating mutation seeds by Alg.2
- 6: Assess the fitness of all seeds
- 7: Select N dandelions via selection strategy
- 8: Dandelions compete with each other by Alg.4
- 9: Construct a distribution model and generate a new offspring by Alg. 5
- 10: Built BM, WM or HM by the new offspring
- 11: Replace the loser with BM, WM or HM
- 12: **end while**

By the Chebyshev's theorem, for any $\varepsilon > 0$, we have

$$P\left\{\left|\tilde{x}_H^d - 0\right| \geq \varepsilon\right\} \leq \frac{R^2}{3\varepsilon^2}. \quad (25)$$

Thus, we take the probability getting error bound of irrelevant directions $p = (R^2/3\varepsilon^2)$. Then (21) holds. ■

In DA, dispersal radius R is dynamically changing. However, we find that R is becoming smaller in its process of evolution. Thus, the error bound of irrelevant directions and p are becoming smaller. It can be concluded that the proposed HM can get smaller error bound of irrelevant directions with higher probability $(1 - p)$.

3) *Framework of Proposed Algorithms*: The complete framework is shown in Algorithm 6. At initialization, N dandelions are generated randomly in search space. During optimization, normal seeds and mutation ones are generated by Algorithms 1 and 2. N dandelions are then selected from the whole population via our selection strategy. Finally, if there are losers among N dandelions, they are replaced with the new offspring generated by one of three models.

In the framework of the proposed algorithms, lines 1 and 2 are the initialization phase and require N function evaluations. The number of normal and mutation seeds are assumed to be C and z , respectively. Thus, lines 4 and 5 require $C + M$ function evaluations. In Algorithm 4, the replacements of the best, worst, and hybrid historical information feedback models need to be evaluated once, once, and three times, respectively. In summary, if the maximum number of generations is set to T , the computational complexity of BDA, WDA, and HDA are $O(N+T(C+z+1))$, $O(N+T(C+z+1))$, and $O(N+T(C+z+3))$, respectively.

V. MATHEMATICAL ANALYSES

The following outlines a mathematical analysis for the convergence of the proposed algorithms. We first recall a result for convergence of basic DA from [19], and then prove the convergence of the proposed algorithms.

Theorem 2: DA can converge to a local or global optimal solution with enough function evaluations.

Theorem 2 has been proved in [19], which indicates that as long as DA search within a given range, i.e., $[L, U]$, it can obtain a local or global optimal solution with enough function evaluations. Thus, we only need to prove that the replacements generated by BDA, WDA, and HDA are in $[L, U]$.

Theorem 3: The new offsprings generated by BDA, WDA, or HDA are in $[L, U]$.

Proof:

1) *BDA*: Here, \tilde{x}_B is a new offspring generated by BDA. So $L \leq \tilde{x}_B \leq U$. If \tilde{x}_B takes U

$$\begin{aligned} \tilde{x}_B &= L + U - \tilde{x}_B \\ &\geq L + U - U = L. \end{aligned} \quad (26)$$

Meanwhile, if \tilde{x}_B takes L

$$\begin{aligned} \tilde{x}_B &= L + U - \tilde{x}_B \\ &\leq L + U - L = U. \end{aligned} \quad (27)$$

From (26) and (27), we have

$$L \leq \tilde{x}_B = L + U - \tilde{x}_B \leq U. \quad (28)$$

Thus, \tilde{x}_B falls in $[L, U]$.

2) *WDA*: The proof is similar to BDA. We have

$$L \leq \tilde{x}_W = L + U - \tilde{x}_W \leq U. \quad (29)$$

3) *HDA*: Since $L \leq \tilde{x}_B \leq U$ and $L \leq L + U - \tilde{x}_W \leq U$, then we have $\alpha_B \cdot L \leq \alpha_B \cdot \tilde{x}_B \leq \alpha_B \cdot U$ and $\alpha_W \cdot L \leq \alpha_W \cdot (L + U - \tilde{x}_W) \leq \alpha_W \cdot U$. Thus, we can get

$$\begin{aligned} (\alpha_B + \alpha_W) \cdot L &\leq \tilde{x}_H = \alpha_B \cdot \tilde{x}_B + \alpha_W \cdot (L + U - \tilde{x}_W) \\ &\leq (\alpha_B + \alpha_W) \cdot U. \end{aligned} \quad (30)$$

Since $\alpha_B + \alpha_W = 1$, we have

$$L \leq \tilde{x}_H = \alpha_B \cdot \tilde{x}_B + \alpha_W \cdot (L + U - \tilde{x}_W) \leq U. \quad (31)$$

Thus, \tilde{x}_H falls in $[L, U]$.

In sum, the new offsprings of the proposed algorithms fall in $[L, U]$. ■

Theorem 4: The proposed algorithms can converge to a local or global optimal solution with enough function evaluations

Proof: In the proposed algorithms, the normal and mutation seeds must be in the range $[L, U]$ due to cross-border handling, and the new offsprings are also in the range based on Theorem 3.

From the above analysis, we can conclude that the proposed algorithms can obtain a local or global optimal solution with enough function evaluations at all times based on Theorem 2. ■

VI. EXPERIMENTS AND RESULT ANALYSES

A. Benchmark Functions and Parameters Settings

Twenty eight benchmark functions are selected from CEC2013 [39], as shown in Table I, where $f1$ – $f5$ are unimodal, $f6$ – $f20$ are multimodal, and $f21$ – $f28$ are composed functions, to verify the performance of the proposed algorithms. “Accept” represents an acceptable solution, which is found by an algorithm and its mean error falls in the range. For a fair comparison, the number of fitness evaluations of

TABLE I
28 BENCHMARK FUNCTIONS FROM CEC2013

No.	Range	Optimum	Name	Accept
f_1	[-100, 100] ^d	-1400	Sphere	[0, 1.00E-06]
f_2		-1300	Rotated High	[0, 1.00E+06]
f_3		-1200	Conditioned Elliptic	[0, 1.00E+08]
f_4		-1100	Rotated Bent Cigar	[0, 1.00E+00]
f_5		-1000	Rotated Discus	[0, 1.00E-03]
f_6	[-100, 100] ^d	-900	Different Powers	[0, 1.00E-03]
f_7		-800	Rotated Rosenbrocks	[0, 2.00E+01]
f_8		-800	Rotated Schaffers F7	[0, 7.00E+01]
f_9		-700	Rotated Ackleys	[0, 2.10E+01]
f_{10}		-600	Rotated Weierstrass	[0, 2.00E+01]
f_{11}		-500	Rotated Griewanks	[0, 2.00E-02]
f_{12}		-400	Rastrigin Function	[0, 7.00E+01]
f_{13}		-300	Rotated Rastrigin	[0, 1.00E+02]
f_{14}		-200	Non-Continuous	[0, 2.00E+02]
f_{15}		-100	Rotated Rastrigin	[0, 2.30E+03]
f_{16}		100	Schwefels	[0, 2.30E+03]
f_{17}		100	Rotated Schwefels	[0, 3.50E+03]
f_{18}		200	Rotated Katsuura	[0, 3.00E-01]
f_{19}	[-100, 100] ^d	300	Lunacek Bi_Rastrigin	[0, 1.30E+02]
f_{20}		400	Rotated Lunacek	[0, 1.70E+02]
f_{21}		500	Bi_Rastrigin	[0, 1.70E+02]
f_{22}		500	Expanded Griewanks	[0, 6.00E+00]
f_{23}		600	plus Rosenbrocks	[0, 6.00E+00]
f_{24}		700	Expanded Scaffers F6	[0, 1.30E+01]
f_{25}		800	Composition Function 1	[0, 3.00E+02]
f_{26}		900	Composition Function 2	[0, 3.00E+03]
f_{27}		1000	Composition Function 3	[0, 4.00E+03]
f_{28}		1100	Composition Function 4	[0, 2.65E+02]
		1200	Composition Function 5	[0, 2.85E+02]
		1300	Composition Function 6	[0, 2.05E+02]
		1400	Composition Function 7	[0, 8.50E+02]
		1400	Composition Function 8	[0, 3.50E+02]

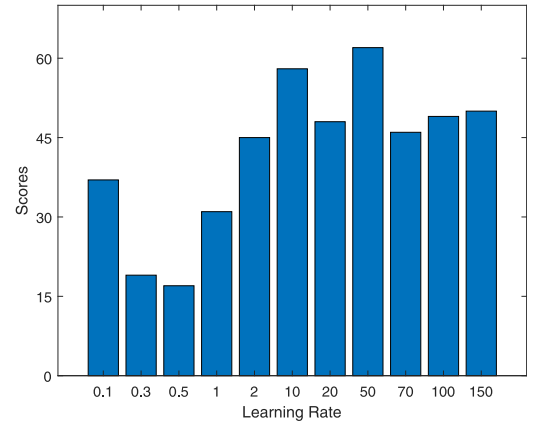
each compared algorithm is set to 3×10^5 , and each comparison is repeated 51 times to reduce statistical errors. The dimension of each test function is set to 30. The parameters of all compared algorithms are set to be same as their corresponding references, and parameters of the improved DAs follow the suggestion in [18] with $H_M = 40$.

B. Sensitivity to Learning Rate

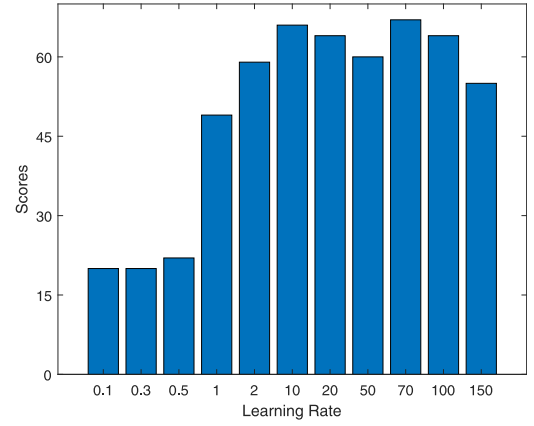
In this section, we study the impact of learning rate δ of Algorithm 4 on the proposed algorithms' performance. Set δ to 0.1, 0.3, 0.5, 1, 2, 10, 20, 50, 70, 100, and 150 for each proposed algorithm. In order to compare the performance of δ taking different values, a set of pairwise Wilcoxon rank-sum tests is used to show if δ taking a value significantly better than taking other values. Meanwhile, we define that if a method performs significantly better than another one on a certain test function, it receives one point. Thus, the maximum total score (the total number of received points) for an algorithm with δ taking a specific value is $28 \times (11 - 1) = 280$. The results with different δ values are shown in Fig. 2. We conclude that when $\delta = 50$, BDA has better performance, $\delta = 70$ is suitable for WDA, and HDA reaches its best with $\delta = 100$. Thus, δ is set to 50, 70, and 100 for them in the following experiments, respectively.

C. Efficiency of Our Competition Mechanism

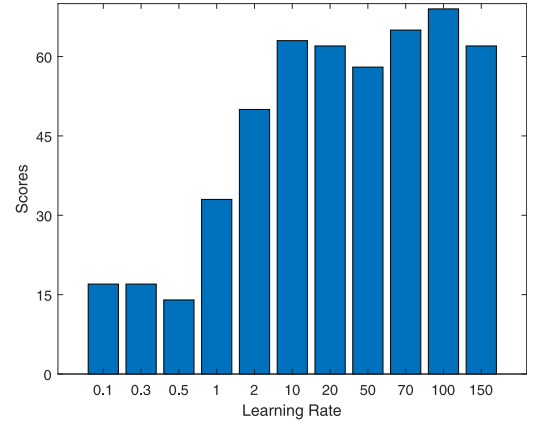
In this section, we need to answer the following question: Why is the average fitness chosen instead of the minimum or maximum fitness in Algorithm 4?



(a)



(b)



(c)

Fig. 2. Total scores of (a) BDA, (b) WDA, and (c) HDA taking different learning rates.

In the following, we take BDA as an example to answer the above question, with the minimum fitness we denote it as BDAm, and with the maximum fitness as BDAM. The detail results of WDA and HDA are given in Table S1 and S2 of the supplementary file.

The mean errors are presented in Table II, and the smallest mean error is marked bold. The Wilcoxon rank-sum tests

TABLE II
COMPARISON AMONG BDAm, BDAM, AND BDA

F.	BDAm	BDAM	BDA	vs. BDAm	vs. BDAM
<i>f</i> 1	0.00E+00	1.75E-07	1.25E-08	0	0
<i>f</i> 2	6.08E+05	1.16E+06	8.25E+05	-1	1
<i>f</i> 3	2.66E+08	1.29E+08	1.95E+07	1	1
<i>f</i> 4	1.74E+03	1.54E+03	4.16E+02	0	1
<i>f</i> 5	1.27E+00	8.77E-04	1.26E-03	1	-1
<i>f</i> 6	2.34E+01	2.54E+01	1.45E+01	0	1
<i>f</i> 7	6.84E+01	5.58E+01	5.58E+01	1	0
<i>f</i> 8	2.09E+01	2.09E+01	2.09E+01	1	1
<i>f</i> 9	2.30E+01	2.05E+01	1.82E+01	1	1
<i>f</i> 10	3.43E-02	3.11E-02	1.54E-02	1	1
<i>f</i> 11	9.27E+01	7.77E+01	6.82E+01	1	0
<i>f</i> 12	1.31E+02	1.18E+02	9.24E+01	1	1
<i>f</i> 13	2.16E+02	1.85E+02	1.85E+02	1	0
<i>f</i> 14	3.04E+03	2.52E+03	2.25E+03	1	1
<i>f</i> 15	3.85E+03	3.66E+03	2.94E+03	1	1
<i>f</i> 16	3.87E-01	3.67E-01	1.83E-01	1	1
<i>f</i> 17	1.27E+02	1.28E+02	1.06E+02	1	1
<i>f</i> 18	1.81E+02	1.73E+02	1.44E+02	1	1
<i>f</i> 19	5.51E+00	6.06E+00	4.40E+00	1	1
<i>f</i> 20	1.28E+01	1.21E+01	1.27E+01	0	-1
<i>f</i> 21	3.05E+02	3.19E+02	2.31E+02	0	1
<i>f</i> 22	3.24E+03	2.99E+03	2.57E+03	1	1
<i>f</i> 23	4.51E+03	4.13E+03	3.62E+03	1	1
<i>f</i> 24	2.59E+02	2.56E+02	2.52E+02	1	0
<i>f</i> 25	2.83E+02	2.81E+02	2.78E+02	1	1
<i>f</i> 26	2.00E+02	2.00E+02	2.00E+02	1	1
<i>f</i> 27	8.38E+02	8.72E+02	8.03E+02	1	1
<i>f</i> 28	3.60E+02	3.51E+02	2.65E+02	1	1
AR.	2.64	2.14	1.21	-	-
1/-1	-	-	-	22/1	21/2

between BDA and BDAm/BDAM are conducted. In Table II, “1” indicates that BDA performs significantly better than a compared algorithm, “-1” indicates otherwise, and “0” means no significant difference between them.

From Table II, BDAm only has the best performance on function *f*1 and *f*2, BDAM performs the best on function *f*5, *f*7, *f*13, and *f*20, and BDA performs best on all the other functions. Judging from the average rankings, BDA has the best average ranking, which means that BDA has better performance than BDAm and BDAM on most test functions.

In the pairwise comparisons, the results achieved by BDA are significantly better than those by BDAm and BDAM, which proves that choosing the average in line 3 of Algorithm 4 is the best.

D. Efficiency of Information Feedback Models

The proposed algorithms use a solution from three information feedback models to replace a loser in the competition. In order to verify the effectiveness of using these models, we need to answer the following two questions.

- 1) Do the proposed algorithms perform better than DA with our proposed selection in Section III-C, denoted as DAS?
- 2) Do the proposed algorithms perform better than DA with a random replacement mechanism, denoted as DAR?

In DAR, if a dandelion is a loser, it is replaced with an individual randomly generated in the feasible domain. For a fair comparison, the learning rate is set to 50, 70, and 100

for DAR (abbreviated as DAR₅₀, DAR₇₀, and DAR₁₀₀). The mean errors and standard deviations are shown in Table III. The smallest mean error of each function and the best average ranking are marked bold. The results achieved by pairwise Wilcoxon rank-sum tests between the proposed algorithms and DAS/DAR are shown in the bottom of Table III, in which “+” indicates that the proposed algorithms perform significantly better than the compared algorithm, and “-” indicates otherwise.

From Table III, DAS, DAR₅₀, DAR₇₀, and DAR₁₀₀ perform well on *f*1, *f*20, *f*5, and *f*7, respectively, and they perform poorly on other functions. The results of the proposed ones are comparable. It can be seen that their average rankings are better than DAS and DARs. Judging from the average rankings, HDA is the best and DAR₇₀ is the worst, and BDA and WDA are the second and third places, respectively. The results of pairwise Wilcoxon rank-sum tests indicate that the proposed algorithms are significantly better than DAS and DARs. We can conclude that the proposed information feedback models are more effective than a random replacement mechanism.

E. Comparison With DA and Its Variants

In this section, the proposed algorithms are compared with DA and its variants [19], [20]. Note that the super parameter *p* of DAPML, DAPMB, and DAPME is, respectively, set to 0.5, 0.5, and 0.1 according to [20]. The mean errors and the ranking on each function are shown in Table IV. The average rankings are presented at its bottom.

From Table IV, it is easy to find that DA performs poorly, and its average ranking is also the worst. Its variants have better results on *f*1–*f*5 except *f*3, but perform poorly on other functions. BDA, WDA, and HDA have the best results on 8, 5, and 10 functions, respectively. It means that the proposed algorithms have better performance than DA and its variants on *f*6–*f*28 except *f*20. Judging from the average rankings, HDA ranks the first on the 28 test functions with the average ranking of 2.57. BDA and WDA take the second and third place. The results of the proposed algorithms are comparable.

The results achieved by pairwise Wilcoxon rank-sum tests between the proposed algorithms and their peers are shown in the bottom of Table IV, in which “+” indicates that the proposed algorithms statistically outperform its peers, “-” indicates otherwise, and “=” indicates no significant difference between them. It can be seen that the proposed algorithms outperform DA and its variants significantly. Moreover, their convergence curves are shown in Fig. S1 of the supplementary file.

The convergence time of these eight algorithms is compared. We define an algorithm that has converged when an acceptable solution (defined in Table I) is found. All compared algorithms are run in MATLAB (R2018a) on a PC with Windows 7 operating system (64 bit) and 4-GB RAM. The results of convergence time are shown in Table V, where the best total and average convergence time are marked in bold. From Table V, it can be seen that the convergence time of DAPML, DAPMB, and DAPME is comparable. MDA is the worst among them. BDA, WDA, and HDA are better than

TABLE III
COMPARISON OF DAS, DAR₅₀, DAR₇₀, DAR₁₀₀, BDA, WDA, AND HDA

	DAS	DAR ₅₀	DAR ₇₀	DAR ₁₀₀	BDA	WDA	HDA
F.	Mean \pm Std	Mean \pm Std	Mean \pm Std	Mean \pm Std	Mean \pm Std	Mean \pm Std	Mean \pm Std
f1	3.00E-10 \pm 0.00E+00	3.50E-07 \pm 2.00E-06	5.19E-08 \pm 0.00E+00	5.80E-09 \pm 0.00E+00	1.25E-08 \pm 0.00E+00	1.44E-08 \pm 0.00E+00	1.25E-08 \pm 0.00E+00
f2	1.26E+06 \pm 7.00E+05	1.03E+06 \pm 5.51E+05	1.11E+06 \pm 6.10E+05	1.01E+06 \pm 5.33E+05	8.25E+05 \pm 2.98E+05	8.00E+05 \pm 2.56E+05	8.67E+05 \pm 2.87E+05
f3	5.21E+07 \pm 6.99E+07	1.39E+08 \pm 2.06E+08	1.31E+08 \pm 2.73E+08	6.61E+07 \pm 6.47E+07	1.95E+07 \pm 2.16E+07	2.43E+07 \pm 2.41E+07	1.55E+07 \pm 1.75E+07
f4	1.12E+03 \pm 1.44E+03	2.13E+03 \pm 3.68E+03	2.19E+03 \pm 2.26E+03	2.48E+03 \pm 2.94E+03	4.16E+02 \pm 2.75E+02	3.67E+02 \pm 2.92E+02	4.66E+02 \pm 3.30E+02
f5	9.61E-04 \pm 3.67E-04	8.85E-04 \pm 2.04E-04	8.01E-04 \pm 2.84E-04	9.86E-04 \pm 3.83E-04	1.26E-03 \pm 3.05E-04	1.26E-03 \pm 2.63E-04	1.14E-03 \pm 2.58E-04
f6	2.19E+01 \pm 1.66E+01	2.34E+01 \pm 1.62E+01	2.11E+01 \pm 1.37E+01	1.93E+01 \pm 1.23E+01	1.45E+01 \pm 5.96E+00	1.41E+01 \pm 3.60E+00	1.42E+01 \pm 9.02E+00
f7	5.62E+01 \pm 1.70E+01	5.50E+01 \pm 1.68E+01	5.28E+01 \pm 1.55E+01	5.23E+01 \pm 1.53E+01	5.58E+01 \pm 1.85E+01	6.12E+01 \pm 1.70E+01	5.31E+01 \pm 1.52E+01
f8	2.09E+01 \pm 6.75E-02	2.09E+01 \pm 5.12E-02	2.09E+01 \pm 5.32E-02	2.09E+01 \pm 7.86E-02	2.09E+01 \pm 7.47E-02	2.09E+01 \pm 9.22E-02	2.09E+01 \pm 7.75E-02
f9	1.97E+01 \pm 3.14E+00	1.99E+01 \pm 2.93E+00	2.08E+01 \pm 2.18E+00	2.06E+01 \pm 2.56E+00	1.82E+01 \pm 2.79E+00	1.88E+01 \pm 2.63E+00	1.78E+01 \pm 2.85E+00
f10	3.01E-02 \pm 1.95E-02	3.20E-02 \pm 1.63E-02	3.50E-02 \pm 2.11E-02	3.00E-02 \pm 2.10E-02	1.54E-02 \pm 9.74E-03	1.55E-02 \pm 8.40E-03	1.58E-02 \pm 1.18E-02
f11	7.88E+01 \pm 1.84E+01	7.38E+01 \pm 1.44E+01	7.63E+01 \pm 2.00E+01	8.09E+01 \pm 1.92E+01	6.82E+01 \pm 1.34E+01	6.67E+01 \pm 1.26E+01	6.86E+01 \pm 1.26E+01
f12	1.14E+02 \pm 3.07E+01	1.20E+02 \pm 3.37E+01	1.14E+02 \pm 3.38E+01	1.10E+02 \pm 3.07E+01	9.24E+01 \pm 2.23E+01	9.84E+01 \pm 2.11E+01	9.68E+01 \pm 2.08E+01
f13	1.90E+02 \pm 2.97E+01	1.77E+02 \pm 3.46E+01	1.84E+02 \pm 3.01E+01	1.86E+02 \pm 3.14E+01	1.85E+02 \pm 3.16E+01	1.73E+02 \pm 2.99E+01	1.85E+02 \pm 3.38E+01
f14	2.62E+03 \pm 5.04E+02	2.60E+03 \pm 5.67E+02	2.55E+03 \pm 5.57E+02	2.54E+03 \pm 4.76E+02	2.25E+03 \pm 3.68E+02	2.24E+03 \pm 4.06E+02	2.20E+03 \pm 3.01E+02
f15	3.73E+03 \pm 5.96E+02	3.55E+03 \pm 6.69E+02	3.68E+03 \pm 5.44E+02	3.55E+03 \pm 5.13E+02	2.94E+03 \pm 5.19E+02	3.16E+03 \pm 3.39E+02	3.12E+03 \pm 4.12E+02
f16	4.27E-01 \pm 3.16E-01	4.02E-01 \pm 2.37E-01	3.32E-01 \pm 2.10E-01	3.75E-01 \pm 2.85E-01	1.83E-01 \pm 6.83E-02	1.92E-01 \pm 5.95E-02	1.95E-01 \pm 8.95E-02
f17	1.26E+02 \pm 2.09E+01	1.23E+02 \pm 3.01E+01	1.27E+02 \pm 2.42E+01	1.24E+02 \pm 2.33E+01	1.06E+02 \pm 1.54E+01	1.07E+02 \pm 1.66E+01	1.06E+02 \pm 1.27E+01
f18	1.80E+02 \pm 4.92E+01	1.76E+02 \pm 4.04E+01	1.71E+02 \pm 3.77E+01	1.83E+02 \pm 4.67E+01	1.44E+02 \pm 2.21E+01	1.41E+02 \pm 2.62E+01	1.43E+02 \pm 2.55E+01
f19	5.95E+00 \pm 1.82E+00	5.97E+00 \pm 2.02E+00	5.89E+00 \pm 1.79E+00	6.24E+00 \pm 1.75E+00	4.40E+00 \pm 1.15E+00	4.52E+00 \pm 9.28E-01	4.29E+00 \pm 9.19E-01
f20	1.21E+01 \pm 7.48E-01	1.21E+01 \pm 6.78E-01	1.22E+01 \pm 8.76E-01	1.23E+01 \pm 7.86E-01	1.27E+01 \pm 1.04E+00	1.30E+01 \pm 1.07E+00	1.29E+01 \pm 1.09E+00
f21	2.98E+02 \pm 6.16E+01	3.05E+02 \pm 8.89E+01	3.08E+02 \pm 6.63E+01	3.00E+02 \pm 7.94E+01	2.31E+02 \pm 5.83E+01	2.16E+02 \pm 5.05E+01	2.14E+02 \pm 5.30E+01
f22	2.91E+03 \pm 6.08E+02	2.89E+03 \pm 5.48E+02	2.83E+03 \pm 5.24E+02	2.87E+03 \pm 4.96E+02	2.57E+03 \pm 5.16E+02	2.63E+03 \pm 4.43E+02	2.61E+03 \pm 3.84E+02
f23	4.20E+02 \pm 7.02E+01	4.28E+02 \pm 8.48E+02	4.32E+02 \pm 8.42E+02	4.27E+02 \pm 8.04E+02	3.62E+03 \pm 6.10E+02	3.66E+03 \pm 5.38E+02	3.69E+03 \pm 5.93E+02
f24	2.55E+02 \pm 1.04E+01	2.54E+02 \pm 8.55E+00	2.57E+02 \pm 9.44E+00	2.57E+02 \pm 8.45E+00	2.52E+02 \pm 1.03E+01	2.53E+02 \pm 1.09E+01	2.51E+02 \pm 1.04E+01
f25	2.80E+02 \pm 8.00E+00	2.78E+02 \pm 8.09E+00	2.81E+02 \pm 6.98E+00	2.80E+02 \pm 8.39E+00	2.78E+02 \pm 7.46E+00	2.78E+02 \pm 6.94E+00	2.77E+02 \pm 7.49E+00
f26	2.00E+02 \pm 4.19E-01	2.00E+02 \pm 6.40E-01	2.00E+02 \pm 8.07E-01	2.00E+02 \pm 7.26E-01	2.00E+02 \pm 1.74E-02	2.00E+02 \pm 1.57E-02	2.00E+02 \pm 1.65E-02
f27	8.52E+02 \pm 7.39E+01	8.36E+02 \pm 7.65E+01	8.58E+02 \pm 8.58E+01	8.60E+02 \pm 7.78E+01	8.03E+02 \pm 7.59E+01	8.10E+02 \pm 6.87E+01	8.09E+02 \pm 6.90E+01
f28	3.60E+02 \pm 2.29E+02	3.42E+02 \pm 2.09E+02	3.47E+02 \pm 2.27E+02	3.42E+02 \pm 2.05E+02	2.65E+02 \pm 7.70E+01	2.69E+02 \pm 7.35E+01	2.80E+02 \pm 6.01E+01
AR	5.21	4.96	5.28	4.93	2.39	2.75	2.32
	20/2	20/2	23/2	21/1	BDA (+/-)		
	21/3	19/2	23/3	20/3	WDA (+/-)		
	21/2	20/2	23/2	21/2	HDA (+/-)		

TABLE IV
COMPARISON OF THE PROPOSED ALGORITHMS WITH DA AND ITS VARIANTS

F.	DA	MDA	DAPML	DAPMB	DAPME	BDA	WDA	HDA
f1	0.00E+00 / 1	0.00E+00 / 1	0.00E+00 / 1	0.00E+00 / 1	0.00E+00 / 1	1.25E-08 / 2	1.44E-08 / 3	1.25E-08 / 2
f2	3.21E+05 / 4	4.57E+05 / 5	3.15E+05 / 3	3.13E+05 / 2	3.01E+05 / 1	8.25E+05 / 7	8.00E+05 / 6	8.67E+05 / 8
f3	1.28E+08 / 8	1.16E+08 / 7	9.03E+07 / 5	1.13E+08 / 6	7.16E+07 / 4	1.95E+07 / 2	2.43E+07 / 3	1.55E+07 / 1
f4	4.02E-01 / 4	2.13E-01 / 1	4.02E-01 / 5	3.34E-01 / 3	3.00E-01 / 2	4.16E+02 / 7	3.67E+02 / 6	4.66E+02 / 8
f5	2.72E-04 / 3	3.10E-04 / 5	2.82E-04 / 4	2.59E-04 / 1	2.61E-04 / 2	1.26E-03 / 7	1.26E-03 / 8	1.14E-03 / 6
f6	2.66E+01 / 7	2.69E+01 / 8	2.59E+01 / 5	2.59E+01 / 6	1.96E+01 / 4	1.45E+01 / 3	1.41E+01 / 1	1.42E+01 / 2
f7	8.41E+01 / 5	8.77E+01 / 7	7.97E+01 / 4	8.50E+01 / 6	9.06E+01 / 8	5.58E+01 / 2	6.12E+01 / 3	5.31E+01 / 1
f8	2.09E+01 / 8	2.09E+01 / 5	2.09E+01 / 4	2.09E+01 / 7	2.09E+01 / 6	2.09E+01 / 3	2.09E+01 / 1	2.09E+01 / 2
f9	2.17E+01 / 6	2.27E+01 / 8	2.11E+01 / 4	2.18E+01 / 7	2.16E+01 / 5	1.82E+01 / 2	1.88E+01 / 3	1.78E+01 / 1
f10	3.34E-02 / 7	4.36E-02 / 8	2.82E-02 / 4	3.10E-02 / 5	3.23E-02 / 6	1.54E-02 / 1	1.55E-02 / 2	1.58E-02 / 3
f11	8.76E+01 / 5	8.34E+01 / 4	9.40E+01 / 8	8.87E+01 / 6	9.18E+01 / 7	6.82E+01 / 2	6.67E+01 / 1	6.86E+01 / 3
f12	1.35E+02 / 6	1.38E+02 / 8	1.25E+02 / 5	1.24E+02 / 4	1.37E+02 / 7	9.24E+01 / 1	9.84E+01 / 3	9.68E+01 / 2
f13	2.30E+02 / 7	2.36E+02 / 8	2.25E+02 / 5	2.24E+02 / 4	2.26E+02 / 6	1.85E+02 / 3	1.73E+02 / 1	1.85E+02 / 2
f14	2.79E+03 / 7	2.83E+03 / 8	2.76E+03 / 6	2.72E+03 / 4	2.73E+03 / 5	2.25E+03 / 3	2.24E+03 / 2	2.20E+03 / 1
f15	3.77E+03 / 8	3.62E+03 / 5	3.46E+03 / 4	3.68E+03 / 7	3.67E+03 / 6	2.94E+03 / 1	3.16E+03 / 3	3.12E+03 / 2
f16	3.41E-01 / 6	5.21E-01 / 8	3.16E-01 / 4	3.41E-01 / 7	3.34E-01 / 5	1.83E-01 / 1	1.92E-01 / 2	1.95E-01 / 3
f17	1.35E+02 / 7	1.33E+02 / 6	1.31E+02 / 5	1.30E+02 / 4	1.37E+02 / 8	1.06E+02 / 2	1.07E+02 / 3	1.06E+02 / 1
f18	1.89E+02 / 8	1.88E+02 / 7	1.76E+02 / 4	1.80E+02 / 5	1.85E+02 / 6	1.44E+02 / 3	1.41E+02 / 1	1.43E+02 / 2
f19	6.62E+00 / 8	6.17E+00 / 7	6.06E+00 / 5	5.93E+00 / 4	6.11E+00 / 6	4.40E+00 / 2	4.52E+00 / 3	4.29E+00 / 1
f20	1.22E+01 / 5	1.20E+01 / 4	1.20E+01 / 3	1.20E+01 / 2	1.20E+01 / 1	1.27E+01 / 6	1.30E+01 / 8	1.29E+01 / 7
f21	3.57E+02 / 8	3.30E+02 / 5	3.20E+02 / 4	3.45E+02 / 6	3.47E+02 / 7	2.31E+02 / 3	2.16E+02 / 2	2.14E+02 / 1
f22	3.24E+03 / 8	3.20E+03 / 7	3.15E+03 / 5	3.18E+03 / 6	3.10E+03 / 4	2.57E+03 / 1	2.63E+03 / 3	2.61E+03 / 2
f23	4.08E+03 / 5	4.07E+03 / 4	4.14E+03 / 6	4.30E+03 / 8	4.17E+03 / 7	3.62E+03 / 1	3.66E+03 / 2	3.69E+03 / 3
f24	2.64E+02 / 4	2.68E+02 / 8	2.65E+02 / 6	2.65E+02 / 5	2.65E+02 / 7	2.52E+02 / 2	2.53E+02 / 3	2.51E+02 / 1
f25	2.88E+02 / 6	2.90E+02 / 8	2.86E+02 / 4	2.89E+02 / 7	2.87E+02 / 5	2.78E+02 / 2	2.78E+02 / 3	2.77E+02 / 1
f26	2.12E+02 / 8	2.10E+02 / 7	2.03E+02 / 4	2.06E+02 / 5	2.06E+02 / 6	2.00E+02 / 2	2.00E+02 / 3	2.00E+02 / 1
f27	9.04E+02 / 7	9.15E+02 / 8	8.95E+02 / 5	8.76E+02 / 4	8.99E+02 / 6	8.03E+02 / 1	8.10E+02 / 3	8.09E+02 / 2
f28	4.05E+02 / 8	3.19E+02 / 4	3.68E+02 / 6	3.41E+02 / 5	3.95E+02 / 7	2.65E+02 / 1	2.69E+02 / 2	2.80E+02 / 3
AR	6.21	6.11	4.57	4.89	5.18	2.61	3.00	2.57
	21 / 3 / 4	21 / 3 / 4	21 / 3 / 4	22 / 2 / 4	22 / 2 / 4	BDA (+ / = / -)		
	21 / 3 / 4	21 / 3 / 4	21 / 3 / 4	22 / 2 / 4	21 / 3 / 4	WDA (+ / = / -)		
	21 / 3 / 4	21 / 3 / 4	21 / 3 / 4	21 / 3 / 4	22 / 2 / 4	HDA (+ / = / -)		

others, and HDA is the best. Finally, our proposed algorithms have the least convergence time on these functions among all algorithms.

F. Comparison With Competitors

To further study their performance, the proposed algorithms are compared with eight optimizers participating

in the CEC2013 competitions. The results of optimizers used in the comparison are taken from the original articles. The competitors are ABC [7], Standard PSO 2011 (SPSO) [40], A hybrid PSO and ABC (PSOABC) [41], genetic algorithm with three-parent crossover (GATC) [42], population's variance-based adaptive differential evolution (PVADE) [43], differential evolution with automatic parameter configuration (DEAPC) [44], continuous differential

TABLE V

COMPARISON OF CONVERGENCE TIME, WHERE “SUM” AND “AVG” INDICATE THAT TOTAL AND AVERAGE CONVERGENCE TIME FOR ALL FUNCTIONS

F.	DA	MDA	DAPML	DAPMB	DAPME	BDA	WDA	HDA
<i>f</i> 1	2.36	0.87	1.18	0.69	0.78	1.48	1.70	2.19
<i>f</i> 2	1.60	1.73	1.70	1.02	0.61	1.02	2.71	2.29
<i>f</i> 3	0.83	0.82	0.58	0.82	0.77	1.04	0.89	0.91
<i>f</i> 4	1.47	1.28	1.47	1.40	1.55	2.39	2.14	2.05
<i>f</i> 5	0.64	0.78	0.70	0.62	0.63	1.72	1.64	1.57
<i>f</i> 6	1.68	0.31	1.79	1.78	1.65	0.50	0.44	0.52
<i>f</i> 7	5.06	4.88	0.66	1.06	1.04	0.68	0.66	1.05
<i>f</i> 8	1.35	1.26	1.27	1.51	1.52	2.03	1.85	0.52
<i>f</i> 9	3.25	26.47	2.68	3.97	3.05	3.41	4.40	6.53
<i>f</i> 10	0.77	2.49	0.77	0.87	0.80	2.75	2.10	2.92
<i>f</i> 11	2.92	2.74	2.82	2.28	2.90	3.37	0.60	3.57
<i>f</i> 12	3.72	3.49	3.77	3.61	3.40	0.78	0.88	1.16
<i>f</i> 13	0.41	0.28	3.65	3.31	3.74	0.48	0.49	0.43
<i>f</i> 14	3.31	3.03	0.33	0.36	0.24	0.85	0.69	0.62
<i>f</i> 15	0.45	3.26	0.41	0.38	0.51	0.74	4.40	0.46
<i>f</i> 16	7.09	6.86	0.79	0.95	0.85	1.79	1.23	1.48
<i>f</i> 17	2.28	2.10	0.39	0.29	0.36	0.60	0.66	0.63
<i>f</i> 18	2.79	0.42	0.31	0.35	0.32	0.77	0.67	0.70
<i>f</i> 19	2.07	1.79	2.03	2.01	2.07	0.67	0.64	0.57
<i>f</i> 20	0.32	0.24	0.34	0.42	0.28	0.16	0.58	0.27
<i>f</i> 21	6.73	6.48	6.60	6.70	6.76	6.99	7.21	1.59
<i>f</i> 22	0.92	8.33	0.78	0.85	0.87	1.27	1.17	1.09
<i>f</i> 23	1.09	9.06	1.17	1.32	1.28	1.30	1.55	1.17
<i>f</i> 24	1.80	32.45	3.29	3.31	2.84	6.80	6.51	5.00
<i>f</i> 25	33.41	4.16	33.00	33.05	33.44	7.80	6.28	5.15
<i>f</i> 26	1.29	35.76	1.80	2.25	1.36	1.77	1.56	1.69
<i>f</i> 27	36.01	35.07	35.11	34.83	37.63	6.49	8.40	4.03
<i>f</i> 28	3.00	3.74	3.30	3.31	2.80	4.37	4.35	3.49
Sum	128.66	200.15	112.69	113.32	114.02	63.98	66.43	53.65
Avg	4.59	7.15	4.02	4.05	4.07	2.28	2.37	1.92

TABLE VI
MEAN VALUES OF 12 ALGORITHMS ON TEST FUNCTIONS

F.	ABC	SPSO	PSOABC	GATC	PVADE	DEAPC	CDAA	RCES	PLES	BDA	WDA	HDA
<i>f</i> 1	-1.40E+03	-1.40E+03	-1.40E+03	-1.40E+03	-1.40E+03	-1.40E+03	-1.40E+03	-1.40E+03	-1.40E+03	-1.40E+03	-1.40E+03	-1.40E+03
<i>f</i> 2	6.20E+06	3.37E+05	8.77E+05	2.43E+05	2.12E+06	1.74E+05	9.51E+05	2.99E+03	1.34E+07	8.24E+05	7.99E+05	8.66E+05
<i>f</i> 3	5.74E+08	2.88E+08	5.16E+07	3.80E+07	4.50E+02	3.21E+06	4.54E+07	-1.20E+03	1.94E+09	1.95E+07	2.43E+07	1.55E+07
<i>f</i> 4	8.64E+04	3.75E+04	4.92E+03	-1.09E+03	1.59E+04	-1.10E+03	-1.10E+03	-1.10E+03	4.36E+04	-6.84E+02	-7.33E+02	-6.34E+02
<i>f</i> 5	-1.00E+03	-1.00E+03	-1.00E+03	-1.00E+03	-1.00E+03	-1.00E+03	-1.00E+03	-1.00E+03	-1.00E+03	-1.00E+03	-1.00E+03	-1.00E+03
<i>f</i> 6	-8.85E+02	-8.62E+02	-8.89E+02	-8.76E+02	-8.92E+02	-8.91E+02	-8.65E+02	-9.00E+02	-8.22E+02	-8.86E+02	-8.86E+02	-8.86E+02
<i>f</i> 7	-6.75E+02	-7.12E+02	-7.49E+02	-7.71E+02	-7.99E+02	-7.78E+02	-7.31E+02	-7.83E+02	-6.82E+02	-7.44E+02	-7.39E+02	-7.47E+02
<i>f</i> 8	-6.79E+02	-6.79E+02	-6.79E+02	-6.79E+02	-6.79E+02	-6.79E+02	-6.79E+02	-6.79E+02	-6.79E+02	-6.79E+02	-6.79E+02	-6.79E+02
<i>f</i> 9	-5.70E+02	-5.71E+02	-5.71E+02	-5.64E+02	-5.94E+02	-5.69E+02	-5.77E+02	-5.76E+02	-5.67E+02	-5.82E+02	-5.81E+02	-5.82E+02
<i>f</i> 10	-5.00E+02	-5.00E+02	-5.00E+02	-5.00E+02	-5.00E+02	-5.00E+02	-5.00E+02	-5.00E+02	-4.88E+02	-5.00E+02	-5.00E+02	-5.00E+02
<i>f</i> 11	-4.00E+02	-2.95E+02	-4.00E+02	-3.76E+02	-3.42E+02	-3.97E+02	-3.99E+02	-3.98E+02	-2.35E+02	-3.32E+02	-3.33E+02	-3.31E+02
<i>f</i> 12	1.90E+01	-1.96E+02	-2.36E+02	-2.59E+02	-1.85E+02	-2.68E+02	-1.83E+02	-2.98E+02	-8.50E+01	-2.08E+02	-2.02E+02	-2.03E+02
<i>f</i> 13	1.29E+02	-6.00E+00	-8.50E+01	-1.16E+02	-6.90E+01	-1.25E+02	-1.40E+01	-1.98E+02	1.29E+02	-1.50E+01	-2.70E+01	-1.50E+01
<i>f</i> 14	-9.96E+01	3.89E+03	-8.45E+01	8.25E+02	3.10E+03	3.74E+03	5.64E+02	1.99E+02	2.51E+03	2.15E+03	2.14E+03	2.10E+03
<i>f</i> 15	3.98E+03	3.91E+03	3.65E+03	4.07E+03	5.71E+03	4.24E+03	3.97E+03	4.38E+02	4.49E+03	3.04E+03	3.26E+03	3.22E+03
<i>f</i> 16	2.01E+02	2.01E+02	2.01E+02	2.03E+02	2.02E+02	2.02E+02	2.00E+02	2.03E+02	2.01E+02	2.00E+02	2.00E+02	2.00E+02
<i>f</i> 17	3.30E+02	4.16E+02	3.31E+02	3.54E+02	4.02E+02	3.59E+02	3.34E+02	3.34E+02	5.43E+02	4.06E+02	4.07E+02	4.06E+02
<i>f</i> 18	7.04E+02	5.21E+02	4.90E+02	4.70E+02	5.82E+02	4.60E+02	5.96E+02	4.82E+02	6.57E+02	5.44E+02	5.41E+02	5.43E+02
<i>f</i> 19	5.00E+02	5.10E+02	5.02E+02	5.03E+02	5.05E+02	5.02E+02	5.02E+02	5.02E+02	5.24E+02	5.04E+02	5.05E+02	5.04E+02
<i>f</i> 20	6.14E+02	6.14E+02	6.11E+02	6.14E+02	6.11E+02	6.13E+02	6.15E+02	6.15E+02	6.14E+02	6.13E+02	6.13E+02	6.13E+02
AR1	7.40	8.00	4.50	5.90	6.15	4.80	6.05	3.35	9.70	5.40	5.55	5.30
<i>f</i> 21	8.65E+02	1.01E+03	1.02E+03	9.92E+02	1.02E+03	9.67E+02	9.77E+02	9.55E+02	1.03E+03	9.31E+02	9.16E+02	9.14E+02
<i>f</i> 22	8.24E+02	5.10E+03	8.84E+02	2.07E+03	3.30E+03	5.36E+03	1.29E+03	1.30E+03	4.05E+03	3.37E+03	3.43E+03	3.41E+03
<i>f</i> 23	5.85E+03	5.73E+03	5.08E+03	5.23E+03	6.71E+03	5.08E+03	6.31E+03	1.48E+03	5.90E+03	4.52E+03	4.56E+03	4.59E+03
<i>f</i> 24	1.29E+03	1.27E+03	1.25E+03	1.27E+03	1.20E+03	1.29E+03	1.30E+03	1.29E+03	1.30E+03	1.25E+03	1.25E+03	1.25E+03
<i>f</i> 25	1.41E+03	1.40E+03	1.38E+03	1.40E+03	1.33E+03	1.40E+03	1.42E+03	1.39E+03	1.43E+03	1.38E+03	1.38E+03	1.38E+03
<i>f</i> 26	1.40E+03	1.49E+03	1.46E+03	1.53E+03	1.42E+03	1.53E+03	1.49E+03	1.52E+03	1.45E+03	1.40E+03	1.40E+03	1.40E+03
<i>f</i> 27	1.72E+03	2.30E+03	2.21E+03	2.33E+03	1.63E+03	2.49E+03	2.38E+03	2.44E+03	2.45E+03	2.10E+03	2.11E+03	2.11E+03
<i>f</i> 28	1.66E+03	1.80E+03	1.73E+03	1.70E+03	1.70E+03	1.70E+03	1.79E+03	1.70E+03	3.48E+03	1.67E+03	1.67E+03	1.68E+03
AR2	4.63	8.25	5.38	7.25	5.38	8.88	8.75	6.13	10.3	3.50	4.50	3.50
	18/2/8	24/3/1	13/4/11	13/3/12	13/4/11	13/3/12	15/6/7	8/3/17	26/1/1		BDA (+/-/-)	
	18/2/8	25/2/1	12/4/12	14/2/12	12/5/11	13/3/12	14/7/7	8/3/17	26/1/1		WDA (+/-/-)	
	18/1/9	24/3/1	13/4/11	14/2/12	13/4/11	13/3/12	17/4/7	8/3/17	26/1/1		HDA (+/-/-)	

ant-stigmergy algorithm (CDAA) [45], restart covariance-matrix-adaptation evolution strategy with increasing population size (RCES) [46], and parameter-less evolutionary search (PLES) [47]. The mean values of these 12 algorithms with $D = 30$ are shown in Table VI. The best mean value on each function is highlighted. “AR1” and “AR2” indicate the average rankings on $f1-f20$ and $f21-f28$ (composed functions), respectively.

From Table VI, it can be seen that BDA and HDA achieves the better performance on $f21-f28$ (AR2 is the top rank), the

best results on $f1-f20$ are obtained by RCES (AR1 is the top rank). ABC performs the best on nine functions, but it has bad performance on other functions. WDA ranks the second on $f21-f28$ with AR2 of 4.50. Moreover, according to the pairwise Wilcoxon rank-sum tests, BDA, WDA, and HDA are significantly better than ABC, SPSO, PSOABC, GATC, PVADE, DEAPC, CDAA, and PLES. In comparison with RCES, they provide better results in eight cases, comparable results in three cases. In the rest of the test functions, they have worse results than it. It indicates that the proposed algorithms

TABLE VII
COMPARISON OF NINE ALGORITHMS ON REAL-WORLD PROBLEMS

Problems		RCES	DA	MDA	DAPML	DAPMB	DAPME	BDA	WDA	HDA
FMSW	Max	23.54	20.51	24.66	21.36	21.17	17.61	14.81	17.20	12.56
	Min	0	0	0	0	0	0	0	0	0
	Mean	12.34	11.33	12.73	11.13	11.16	9.95	5.08	7.23	4.36
	Std	7.40	6.71	6.32	6.67	7.26	5.41	5.92	5.80	5.52
LJP	Max	-9.31	-12.40	-12.04	-11.11	-12.31	-10.11	-15.60	-15.58	-15.31
	Min	-28.24	-23.10	-24.10	-26.25	-26.11	-27.45	-27.56	-27.48	-27.45
	Mean	-21.35	-17.40	-17.33	-17.61	-17.48	-17.28	-21.54	-21.13	-20.94
	Std	6.39	3.17	3.28	3.85	3.83	4.15	3.11	2.90	3.51
NLSTR	Max	21.83	14.33	14.33	14.33	14.33	14.33	14.33	14.33	14.33
	Min	13.95	13.77	13.77	13.77	13.77	13.77	13.77	13.77	13.77
	Mean	19.72	14.20	14.26	14.20	14.17	14.15	13.95	13.99	14.02
	Std	2.34	0.24	0.19	0.24	0.25	0.26	0.26	0.28	0.28
SSRPCD	Max	1.16	1.33	1.40	1.38	1.32	1.43	0.98	1.04	1.05
	Min	0.54	0.60	0.66	0.67	0.57	0.50	0.53	0.58	0.56
	Mean	0.76	0.91	0.98	0.99	0.99	0.95	0.77	0.85	0.79
	Std	0.14	0.21	0.19	0.17	0.17	0.21	0.11	0.12	0.12
	AR	4.75	6.25	7.75	6.00	6.25	5.75	1.50	3.00	2.75

are superior or competitive to nine participating algorithms benchmarked on 30-D CEC2013 benchmark functions.

G. Comparison on Real-World Problems

Four real-world problems are selected from CEC2011 [48] to validate the performance of our proposed algorithms in applications. They are parameter estimation for frequency-modulated sound waves (FMSW), Lennard-Jones potential (LJP), optimal control of a nonlinear stirred tank reactor (NLSTR), and spread spectrum radar polyphase code design (SSRPCD) problems. RCES (its performance is the best in the above comparison), DA and its variants are selected for comparison. Their parameters are the same as the above, and $F_M = 1.5 \times 10^5$. The results of maximum fitness (Max), minimum fitness (Min), mean fitness (Mean), and standard deviation (Std) after 25 independent runs are shown in Table VII, and the best mean fitness and average ranking values are in a boldface.

RCES has the best performance on SSRPCD due to a restart mechanism, but performs poorly on other three problems. HDA achieves the better performance on FMSW. BDA is the best on LJP and NLSTR. The results of our proposed algorithms are comparable. BDA ranks the first on these problems, HDA takes the second place, and followed by WDA. The outstanding results on these problems prove the superiority of the proposed algorithms over its compared algorithms and indicate that our proposed competition mechanism is a more promising methodology than a restart one.

VII. CONCLUSION

In this article, a competition-driven DA has been proposed, in which dandelions compete with each other, and losers are abandoned and replaced by new offsprings. Three historical information feedback models were innovatively proposed to generate the offsprings based on the EDA for losers. The experimental results showed that the proposed algorithms have better performance than its compared algorithms. Finally, the proposed algorithms are validated on four real-world problems, and the results indicate their superiority over their peers in solving these problems.

In the future, developing other competition mechanisms is important for DA. Furthermore, a cooperative mechanism

in DA is also worth studying. Extending these principles [49]–[56] to DA for further improving its performance is another interesting direction.

REFERENCES

- [1] G. Wang and Y. Tan, "Improving metaheuristic algorithms with information feedback models," *IEEE Trans. Cybern.*, vol. 49, no. 2, pp. 542–555, Feb. 2019.
- [2] G. R. Harik, F. G. Lobo, and D. E. Goldberg, "The compact genetic algorithm," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 287–297, Nov. 1999.
- [3] Y. Yu, S. Gao, Y. Wang, and Y. Todo, "Global optimum-based search differential evolution," *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 2, pp. 379–394, Mar. 2019.
- [4] M. G. H. Omran, "A novel cultural algorithm for real-parameter optimization," *Int. J. Comput. Math.*, vol. 93, no. 9, pp. 1541–1563, 2016.
- [5] Y. Cao, H. Zhang, W. Li, M. Zhou, Y. Zhang, and W. A. Chaovalitwongse, "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.
- [6] M. Dorigo, V. Maniezzo, and A. Colnari, "Ant system: Optimization by a colony of cooperating agents," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 26, no. 1, pp. 29–41, Feb. 1996.
- [7] D. Karaboga, B. Gökemli, C. Ozturk, and N. Karaboga, "A comprehensive survey: Artificial bee colony (ABC) algorithm and applications," *Artif. Intell. Rev.*, vol. 42, no. 1, pp. 21–57, 2018.
- [8] X. Yang, "A new metaheuristic bat-inspired algorithm," in *Proc. Nat. Inspired Cooper. Strategies Optim. (NICSO)*, Granada, Spain, May 2010, pp. 65–74.
- [9] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.
- [10] Z. W. Geem, J. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: Harmony search," *Simulation*, vol. 76, no. 2, pp. 60–68, 2001.
- [11] S. Cheng, Q. Qin, J. Chen, and Y. Shi, "Brain storm optimization algorithm: A review," *Artif. Intell. Rev.*, vol. 46, no. 4, pp. 445–458, 2016.
- [12] Y. Tan and Y. Zhu, "Fireworks algorithm for optimization," in *Proc. 1st Int. Conf. Adv. Swarm Intell. (ICSI)*, Beijing, China, Jun. 2010, pp. 355–364.
- [13] F. Zou, D. Chen, and Q. Xu, "A survey of teaching–learning-based optimization," *Neurocomputing*, vol. 335, pp. 366–383, Mar. 2019.
- [14] Z. Cao, Y. Zhang, J. Guan, S. Zhou, and G. Wen, "A chaotic ant colony optimized link prediction algorithm," *IEEE Trans. Syst., Man, Cybern., Syst.*, early access, Nov. 12, 2019, doi: [10.1109/TSMC.2019.2947516](https://doi.org/10.1109/TSMC.2019.2947516).
- [15] A. N. K. Nasir and M. O. Tokhi, "An improved spiral dynamic optimization algorithm with engineering application," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 6, pp. 943–954, Jun. 2015.
- [16] T.-F. Zhao, W.-N. Chen, A. W.-C. Liew, T. Gu, X.-K. Wu, and J. Zhang, "A binary particle swarm optimizer with priority planning and hierarchical learning for networked epidemic control," *IEEE Trans. Syst., Man, Cybern., Syst.*, early access, Oct. 30, 2019, doi: [10.1109/TSMC.2019.2945055](https://doi.org/10.1109/TSMC.2019.2945055).

- [17] H. Chen *et al.*, "Artificial bee colony optimizer based on bee life-cycle for stationary and dynamic optimization," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 2, pp. 327–346, Feb. 2017.
- [18] X. Li, S. Han, L. Zhao, C. Gong, and X. Liu, "New dandelion algorithm optimizes extreme learning machine for biomedical classification problems," *Comput. Intell. Neurosci.*, vol. 2017, pp. 1–13, Sep. 2017.
- [19] C. Gong, S. Han, X. Li, L. Zhao, and X. Liu, "A new dandelion algorithm and optimization for extreme learning machine," *J. Exp. Theor. Artif. Intell.*, vol. 30, no. 1, pp. 39–52, 2018.
- [20] H. Zhu, G. Liu, M. Zhou, Y. Xie, and Q. Kang, "Dandelion algorithm with probability-based mutation," *IEEE Access*, vol. 7, pp. 97974–97985, 2019.
- [21] M. Crepinsek, S. Liu, and M. Mernik, "Exploration and exploitation in evolutionary algorithms: A survey," *ACM Comput. Surveys*, vol. 45, no. 3, pp. 1–33, 2013.
- [22] M. Dorigo and L. M. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 53–66, Apr. 1997.
- [23] B. A. Whitehead and T. D. Choate, "Cooperative-competitive genetic evolution of radial basis function centers and widths for time series prediction," *IEEE Trans. Neural Netw.*, vol. 7, no. 4, pp. 869–880, Jul. 1996.
- [24] C. K. Goh and K. C. Tan, "A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 13, no. 1, pp. 103–127, Feb. 2009.
- [25] J. Horn, *The Nature of Niching: Genetic Algorithms and the Evolution of Optimal, Cooperative Populations*, vol. 302, Univ. Illinois at Urbana-Champaign, Champaign, IL, USA, 1995.
- [26] H. Mühlenbein and G. Paass, "From recombination of genes to the estimation of distributions I. binary parameters," in *Proc. 4th Int. Conf. Evol. Comput. Parallel Problem Solving Nat. (PPSN)*, Berlin, Germany, Sep. 1996, pp. 178–187.
- [27] W. Gao, F. T. S. Chan, L. Huang, and S. Liu, "Bare bones artificial bee colony algorithm with parameter adaptation and fitness-based neighborhood," *Inf. Sci.*, vol. 316, pp. 180–200, Sep. 2015.
- [28] J. Li, J. Zhang, C. Jiang, and M. Zhou, "Composite particle swarm optimizer with historical memory for function optimization," *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2350–2363, Oct. 2015.
- [29] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Trans. Evol. Comput.*, vol. 10, no. 3, pp. 281–295, Jun. 2006.
- [30] Z. Wang, R. Lu, D. Chen, and F. Zou, "An experience information teaching-learning-based optimization for global optimization," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 9, pp. 1202–1214, Sep. 2016.
- [31] F. Zou, L. Wang, X. Hei, and D. Chen, "Teaching-learning-based optimization with learning experience of other learners and its application," *Appl. Soft Comput.*, vol. 37, pp. 725–736, Dec. 2015.
- [32] J. Li, S. Zheng, and Y. Tan, "The effect of information utilization: Introducing a novel guiding spark in the fireworks algorithm," *IEEE Trans. Evol. Comput.*, vol. 21, no. 1, pp. 153–166, Feb. 2017.
- [33] G. Wang, M. Lu, and X. Zhao, "An improved bat algorithm with variable neighborhood search for global optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Vancouver, BC, Canada, Jul. 2016, pp. 1773–1778.
- [34] L. Pan, C. He, Y. Tian, H. Wang, X. Zhang, and Y. Jin, "A classification-based surrogate-assisted evolutionary algorithm for expensive many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 1, pp. 74–88, Feb. 2019.
- [35] Q. Zhang, W. Liu, E. P. K. Tsang, and B. Virginas, "Expensive multiobjective optimization by MOEA/D with Gaussian process model," *IEEE Trans. Evol. Comput.*, vol. 14, no. 3, pp. 456–474, Jun. 2010.
- [36] Z. Lv, L. Wang, Z. Han, J. Zhao, and W. Wang, "Surrogate-assisted particle swarm optimization algorithm with pareto active learning for expensive multi-objective optimization," *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 3, pp. 838–849, May 2019.
- [37] H. Wang, Y. Jin, and J. Doherty, "Committee-based active learning for surrogate-assisted particle swarm optimization of expensive problems," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2664–2677, Sep. 2017.
- [38] M. El-Abd, "Preventing premature convergence in a PSO and EDA hybrid," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Trondheim, Norway, May 2009, pp. 3060–3066.
- [39] J. Liang, B. Qu, P. Suganthan, and A. G. Hernández-Díaz, "Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization," Dept. Comput. Intell. Lab., Zhengzhou Univ., Zhengzhou, China, Rep. 201212, pp. 281–295, 2013.
- [40] M. Zambrano-Bigiarini, M. Clerc, and R. Rojas-Mujica, "Standard particle swarm optimisation 2011 at CEC-2013: A baseline for future PSO improvements," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Cancun, Mexico, Jun. 2013, pp. 2337–2344.
- [41] 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.
- [42] S. M. 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. (CEC)*, Cancun, Mexico, Jun. 2013, pp. 356–360.
- [43] L. dos Santos Coelho, H. V. H. Ayala, and R. Z. Freire, "Population's variance-based adaptive differential evolution for real parameter optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Cancun, Mexico, Jun. 2013, pp. 1672–1677.
- [44] S. M. M. Elsayed, R. A. Sarker, and T. Ray, "Differential evolution with automatic parameter configuration for solving the CEC2013 competition on real-parameter optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Cancun, Mexico, Jun. 2013, pp. 1932–1937.
- [45] P. Korosec and J. Silc, "The continuous differential ant-stigmergy algorithm applied on real-parameter single objective optimization problems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Cancun, Mexico, Jun. 2013, pp. 1658–1663.
- [46] I. Loshchilov, "CMA-ES with restarts for solving CEC 2013 benchmark problems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Cancun, Mexico, Jun. 2013, pp. 369–376.
- [47] G. Papa and J. Silc, "The parameter-less evolutionary search for real-parameter single objective optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Cancun, Mexico, Jun. 2013, pp. 1131–1137.
- [48] S. Das and P. N. Suganthan, *Problem Definitions and Evaluation Criteria for CEC 2011 Competition on Testing Evolutionary Algorithms on Real World Optimization Problems*, Jadavpur Univ., Kolkata, India, 2010, pp. 341–359.
- [49] Q. Kang, C. Xiong, M. Zhou, and L. Meng, "Opposition-based hybrid strategy for particle swarm optimization in noisy environments," *IEEE Access*, vol. 6, pp. 21888–21900, 2018.
- [50] J. Zhang, X. Zhu, Y. Wang, and M. Zhou, "Dual-environmental particle swarm optimizer in noisy and noise-free environments," *IEEE Trans. Cybern.*, vol. 49, no. 6, pp. 2011–2021, Jun. 2019.
- [51] J. Zhao, S. Liu, M. Zhou, X. Guo, and L. Qi, "Modified cuckoo search algorithm to solve economic power dispatch optimization problems," *IEEE/CAA J. Autom. Sinica*, vol. 5, no. 4, pp. 794–806, Jul. 2018.
- [52] W. Dong and M. Zhou, "A supervised learning and control method to improve particle swarm optimization algorithms," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 7, pp. 1135–1148, Jul. 2017.
- [53] 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.
- [54] Z. Cao, C. Lin, and M. Zhou, "A knowledge-based cuckoo search algorithm to schedule a flexible job shop with sequencing flexibility," *IEEE Trans. Autom. Sci. Eng.*, early access, Nov. 12, 2019, doi: [10.1109/TASE.2019.2945717](https://doi.org/10.1109/TASE.2019.2945717).
- [55] M. Ghahramani, Y. Qiao, M. Zhou, A. O. Hagan, and J. Sweeney, "AI-based modeling and data-driven evaluation for smart manufacturing processes," *IEEE/CAA J. Automatica Sinica*, vol. 7, no. 4, pp. 1026–1037, Jul. 2020.
- [56] J. Wang, Y. Sun, Z. Zhang, and S. Gao, "Solving multitrip pickup and delivery problem with time windows and manpower planning using multiobjective algorithms," *IEEE/CAA J. Automatica Sinica*, vol. 7, no. 4, pp. 1134–1153, Jul. 2020.



Shoufei Han received the B.S. degree in computer science from Hefei University, Hefei, China, in 2012, and the M.S. degree in computer science from Shenyang Aerospace University, Shenyang, China, in 2018. He is currently pursuing the Ph.D. degree in computer science with the Nanjing University of Aeronautics and Astronautics, Nanjing, China.

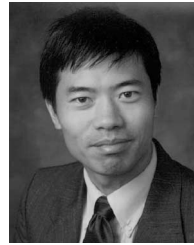
His current research interests include machine learning, intelligent optimization algorithms, extreme learning machine, and evolutionary computation.



Kun Zhu (Member, IEEE) received the Ph.D. degree in computer engineering from Nanyang Technological University, Singapore, in 2012.

He is currently a Professor with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China, where he is also a Jiangsu Specially Appointed Professor. His research interests include intelligent optimization algorithms, 5G/B5G, resource management, applied game theory and optimization, and self-organization networks.

Prof. Zhu has won several research awards, including the IEEE WCNC 2019 “Best Paper Award,” and the ACM Rising Star Chapter Award in 2018. He is an Editor of *Computer and Communications* and has served as the guest editor for several journals. He has been serving as the Program Chair for the 4th MLICOM Conference and a TPC Member for IEEE INFOCOM, GLOBECOM, and ICC. He is a PI of more than ten research projects.



Mengchu Zhou (Fellow, IEEE) received the B.S. degree in control engineering from the Nanjing University of Science and Technology, Nanjing, China, in 1983, the M.S. degree in automatic control from the Beijing Institute of Technology, Beijing, China, in 1986, and the Ph.D. degree in computer and systems engineering from Rensselaer Polytechnic Institute, Troy, NY, USA, in 1990.

In 1990, he joined the New Jersey Institute of Technology (NJIT), Newark, NJ, USA, where he is currently a Distinguished Professor of Electrical and

Computer Engineering. He has over 900 publications, including 12 books, over 600 journal papers (over 460 in IEEE TRANSACTIONS), 26 patents, and 29 book-chapters. His research interests are in Petri nets, intelligent automation, Internet of Things, big data, Web services, and intelligent transportation.

Dr. Zhou was a recipient of the Excellence in Research Prize and Medal from NJIT, the Humboldt Research Award for U.S. Senior Scientists from Alexander von Humboldt Foundation, and the Franklin V. Taylor Memorial Award and the Norbert Wiener Award from IEEE SMC Society, and the Computer-Integrated Manufacturing UNIVERSITY-LEAD Award from Society of Manufacturing Engineers. He is the founding Editor of *IEEE Press Book Series on Systems Science and Engineering* and an Editor-in-Chief of *IEEE/CAA Journal of Automatica Sinica*. He served as an Associate Editor of the IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, and the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, and an Editor of the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING. He served as a Guest-Editor for many journals, including IEEE INTERNET OF THINGS JOURNAL, the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, and the IEEE TRANSACTIONS ON SEMICONDUCTOR MANUFACTURING. He is presently an Associate Editor of the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, the IEEE INTERNET OF THINGS JOURNAL, the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, and *Frontiers of Information Technology and Electronic Engineering*. He was General Chair of IEEE Conference on Automation Science and Engineering, August 2008, a General Co-Chair of 2003 IEEE International Conference on System, Man and Cybernetics (SMC), October 2003 and 2019 IEEE International Conference on SMC, October 2019, the Founding General Co-Chair of 2004 IEEE International Conference on Networking, Sensing and Control, March 2004, and the General Chair of 2006 IEEE International Conference on Networking, Sensing and Control, April 2006. He was a Program Chair of 2010 IEEE International Conference on Mechatronics and Automation, August 2010, and 1998 and 2001 IEEE International Conference on SMC, and 1997 IEEE International Conference on Emerging Technologies and Factory Automation. He has led or participated in over 50 research and education projects with total budget over \$12M, funded by National Science Foundation, Department of Defense, NIST, New Jersey Science and Technology Commission, and industry. He has been among most highly cited scholars since 2012 and ranked top one in the field of engineering worldwide in 2012 by Web of Science. He is a Life Member of Chinese Association for Science and Technology–USA and served as its President in 1999. He is a Fellow of International Federation of Automatic Control, American Association for the Advancement of Science, and Chinese Association of Automation.