Differential Evolution with Ranking-based Mutation Operators

Wenyin Gong and Zhihua Cai

Abstract—**Differential evolution (DE) has been proven to be one of the most powerful global numerical optimization algorithms in the evolutionary algorithm family. The core operator of DE is the differential mutation operator. Generally, the parents in the mutation operator are randomly chosen from the current population. In the nature, good species always contain good information, and hence, they have more chance to be utilized to guide other species. Inspired by this phenomenon, in this paper, we propose the ranking-based mutation operators for the DE algorithm, where some of the parents in the mutation operators are proportionally selected according their rankings in the current population. The higher ranking a parent obtains, the more opportunity it will be selected. In order to evaluate the influence of our proposed ranking-based mutation operators on DE, our approach is compared with the jDE algorithm, which is a highly competitive DE variant with self-adaptive parameters, with different mutation operators. In addition, the proposed ranking-based mutation operators are also integrated into other advanced DE variants to verify the effect on them. Experimental results indicate that our proposed ranking-based mutation operators are able to enhance the performance of the original DE algorithm and the advanced DE algorithms.**

Index Terms—**Differential evolution, ranking, mutation operator, numerical optimization.**

I. Introduction

EVOLUTIONARY algorithms (EAs), including genetic algorithm (GA), evolution strategy (ES), evolutionary programming (EP), and genetic programming (GP), are search algorithms that simulate evolutionary process of natural selection, variation, and genetics [1]. During the last few decades, research in evolutionary computation and the application of EAs to real-world problems have steadily and significantly expanded. Differential evolution (DE), which was firstly proposed by Storn and Price in 1995 [2], [3], is one of the most powerful evolutionary algorithms for global numerical optimization. The advantages of DE are its ease of use, simple structure, speed, efficacy, and robustness. In the last few years, DE has obtained many successful applications in diverse domains, such as engineering optimal design, digital filter design, image processing, data mining, multisensor fusion, and so on [4], [5]. Interested readers can refer to two good surveys of DE in [6] and [7] and the references therein.

This work was partly supported by the National Natural Science Foundation of China under Grant No. 61203307 and 61075063, the Fundamental Research Funds for the Central Universities at China University of Geosciences (Wuhan) under Grant No. CUG130413 and CUG090109, and the Research Fund for the Doctoral Program of Higher Education under Grant No. 20110145120009.

The authors are all with School of Computer Science, China University of Geosciences, Wuhan, 430074, P. R. China. (Email: cug11100304@yahoo.com.cn; zhcai@cug.edu.cn).

Similar to other EAs, in the DE algorithm, it employs the mutation, crossover, and selection operators at each generation to evolve the population to the global optimum. In these three operators, the core operator is the differential mutation operator. Through the mutation operator, the *mutant* vector (also known as *donor* vector) is generated. Generally, the parents in the mutation operator are chosen randomly from the current population. For example, in the classical "DE/rand/1" mutation, three parent vectors $\mathbf{x}_{r_1}$, $\mathbf{x}_{r_2}$, and $\mathbf{x}_{r_3}$ are selected randomly from the current population. The indexes r_1, r_2, and r_3 satisfy $r_1, r_2, r_3 \in \{1, Np\}$ and $r_1 \neq r_2 \neq r_3 \neq i$, where Np is the population size. However, since all parents are chosen randomly, it may lead to the DE algorithm be good at exploring the search space and locating the region of global minimum, but be slow at exploitation of the solutions [8]. Some researchers investigate to hybridize other techniques with DE to accelerate its convergence. Fan and Lampinen [9] proposed a new version of DE which uses an additional mutation operation called trigonometric mutation operation. Sun *et al.* [10] proposed a new hybrid algorithm based on a combination of DE and estimation of distribution algorithm. Kaelo and Ali [11] adopted the attraction-repulsion concept of electromagnetism-like algorithm to boost the mutation operation of the original DE. Yang *et al.*. [12] proposed a neighborhood search based DE. Noman and Iba incorporated local search (LS) into the classical DE algorithm in [8]. They presented an LS technique to solve this problem by adaptively adjusting the length of the search, using a hill-climbing heuristic. Cai *et al.* [13] presented an one-step-K-means based DE algorithm, where the K-means method is used to enhance the exploitation ability of DE.

Combing DE with other search techniques is effective to improve its performance, however, the hybrid approaches are usually more complicated than the original DE algorithm. Generally, in the nature, good species always contain good information, and hence, they are more likely to be utilized to guide other species. Based on these considerations, in this paper, we present ranking-based mutation operators for the DE algorithm. Different from the parent selection in the original DE algorithm, in our approach some of the parents in the mutation operators are proportionally selected according their rankings in the current population. The higher ranking a parent obtains, the more opportunity it will be selected. The major advantages of our approach are as follows: i) since good parents are more likely to be chosen, the ranking-based mutation operators are able to enhance DE's exploitation ability; ii) our approach is still very simple, it does not destroy the simple structure of the original DE algorithm any more;

iii) the ranking-based mutation operators can be easily used in other advanced DE variants; and iv) our approach does not increase the overall complexity of DE. In order to evaluate the influence of our proposed ranking-based mutation operators on DE, our approach is compared with the original DE algorithm with different mutation operators. In addition, the proposed ranking-based mutation operators are also integrated into other advanced DE variants to verify the effect on them. Experimental results indicate that our proposed ranking-based mutation operators are able to enhance the performance of the original DE algorithm and the advanced DE algorithms.

The rest of this paper is organized as follows. Section II briefly describes the original DE algorithm and some related work to the mutation operators in DE. We present our proposed ranking-based mutation operators in Section III in detail. Section IV performs the comprehensive experiments using benchmark functions and real-world application problems. The experimental results are also analyzed in this section. In the last section, Section V draws the conclusions from this work and points out the possible future work.

II. Related Work

For the sake of completeness, in this section, we first describes the original DE algorithm briefly. Then, some related work to the mutation operators in DE are presented.

Without loss of generality, in this work, we consider the following numerical optimization problem:

$$\text{Minimize} \quad f(\mathbf{x}), \qquad \mathbf{x} \in S, \tag{1}$$

where $S \subseteq \mathbb{R}^D$ is a compact set, $\mathbf{x} = [x_1, x_2, \cdots, x_D]^T$, and D is the dimension, *i.e.*, the number of decision variables. Generally, for each variable x_j, it satisfies a boundary constraint, such that:

$$\underline{x}_j \leq x_j \leq \overline{x}_j, \qquad j = 1, 2, \cdots, D. \tag{2}$$

where $\underline{x}_j$ and $\overline{x}_j$ are respectively the lower bound and upper bound of x_j.

A. Differential Evolution

The DE algorithm [3] is a simple evolutionary algorithm (EA) for global numerical optimization. It creates new candidate solutions by combining the parent individual and several other individuals of the same population. A candidate replaces the parent only if it has an equal or better fitness value. The pseudo-code of the original DE algorithm is shown in Algorithm 1, where D is the number of decision variables; Np is the population size; F is the mutation scaling factor; Cr is the crossover rate; $x_{i,j}$ is the j-th variable of the solution $\mathbf{x}_i$; $\mathbf{u}_i$ is the offspring. The function $\text{rndint}(1, D)$ returns a uniformly distributed random integer number between 1 and D, while $\text{rndreal}_j[0, 1)$ gives a uniformly distributed random real number in $[0, 1)$, generated anew for each value of j. Many mutation strategies to create a candidate are available; in Algorithm 1, the use of the classic "DE/rand/1" mutation operator is illustrated (see line 9).

From Algorithm 1, we can see that there are only three control parameters (Np, F and Cr) in DE. As for the terminal conditions, we can either fix the maximum number of fitness function evaluations (*Max_NFFEs*) or define a desired solution value-to-reach (*VTR*).

Algorithm 1 The DE algorithm with "DE/rand/1/bin" strategy

```
1:  Generate the initial population randomly
2:  Evaluate the fitness for each individual in the population
3:  while the stop criterion is not satisfied do
4:     for $i = 1$ to $Np$ do
5:        Select uniform randomly $r_1 \neq r_2 \neq r_3 \neq i$
6:        $j_{rand}$ = rndint$(1, D)$
7:        for $j = 1$ to $D$ do
8:           if $\text{rndreal}_j[0, 1) < Cr$ or $j$ is equal to $j_{rand}$ then
9:              $u_{i,j} = x_{r_1,j} + F \cdot (x_{r_2,j} - x_{r_3,j})$
10:          else
11:             $u_{i,j} = x_{i,j}$
12:          end if
13:       end for
14:    end for
15:    for $i = 1$ to $Np$ do
16:       Evaluate the offspring $\mathbf{u}_i$
17:       if $f(\mathbf{u}_i)$ is better than or equal to $f(\mathbf{x}_i)$ then
18:          Replace $\mathbf{x}_i$ with $\mathbf{u}_i$
19:       end if
20:    end for
21: end while
```

B. Mutation Operators in DE

In the DE algorithm, the core operator is the differential mutation operator. There are many mutation operators that have been proposed [14], [4]. They use different learning strategies in the reproduction stage. In order to distinguish among DE's mutation operators, the notation "DE/*a*/*b*" is used, where "DE" indicates the Differential Evolution; "*a*" denotes the vector to be mutated; and "*b*" is the number of difference vectors used. In DE, some well-known mutation operators are listed as follows.

1) "DE/rand/1":
$$\mathbf{v}_i = \mathbf{x}_{r_1} + F \cdot \big(\mathbf{x}_{r_2} - \mathbf{x}_{r_3}\big) \tag{3}$$

2) "DE/rand/2":
$$\mathbf{v}_i = \mathbf{x}_{r_1} + F \cdot \big(\mathbf{x}_{r_2} - \mathbf{x}_{r_3}\big) + F \cdot \big(\mathbf{x}_{r_4} - \mathbf{x}_{r_5}\big) \tag{4}$$

3) "DE/current-to-best/1"[1]:
$$\mathbf{v}_i = \mathbf{x}_i + F \cdot \big(\mathbf{x}_{best} - \mathbf{x}_i\big) + F \cdot \big(\mathbf{x}_{r_2} - \mathbf{x}_{r_3}\big) \tag{5}$$

4) "DE/current-to-best/2":
$$\mathbf{v}_i = \mathbf{x}_i + F \cdot \big(\mathbf{x}_{best} - \mathbf{x}_i\big) + F \cdot \big(\mathbf{x}_{r_2} - \mathbf{x}_{r_3}\big) + F \cdot \big(\mathbf{x}_{r_4} - \mathbf{x}_{r_5}\big) \tag{6}$$

5) "DE/rand-to-best/1":
$$\mathbf{v}_i = \mathbf{x}_{r_1} + F \cdot \big(\mathbf{x}_{best} - \mathbf{x}_{r_1}\big) + F \cdot \big(\mathbf{x}_{r_2} - \mathbf{x}_{r_3}\big) \tag{7}$$

6) "DE/rand-to-best/2":
$$\mathbf{v}_i = \mathbf{x}_{r_1} + F \cdot \big(\mathbf{x}_{best} - \mathbf{x}_{r_1}\big) + F \cdot \big(\mathbf{x}_{r_2} - \mathbf{x}_{r_3}\big) + F \cdot \big(\mathbf{x}_{r_4} - \mathbf{x}_{r_5}\big) \tag{8}$$

[1]"DE/current-to-best" is also referred to as "DE/target-to-best/" [4], [15].

where $\mathbf{x}_{best}$ represents the best individual in the current generation, $r_1, r_2, r_3, r_4,$ and $r_5 \in \{1, \cdots, Np\}$, and $r_1 \neq r_2 \neq r_3 \neq r_4 \neq r_5 \neq i$. As shown in Equation (3), $\mathbf{x}_i$ is referred to as the *target* vector; $\mathbf{u}_i$ is the *trial* vector; $\mathbf{v}_i$ is the *mutant* vector; $\mathbf{x}_{r_1}$ is the *base* vector; and $\mathbf{x}_{r_2} - \mathbf{x}_{r_3}$ is the *differential* vector.

Generally, different mutation operators have different features and are suitable to different set of problems. However, the choice of the best mutation operators for DE is not easy for a specific problem [16], [17], [18]. Therefore, in order to make the mutation operator selection more easily, some researchers studied new mutation operators. For example, Iorio and Li [19] presented a rotation-invariant operator, namely "DE/current-to-rand/1". Price *et al.* [4, pp. 117] proposed the "DE/rand/1/either-or" algorithm, where the trial vector are either pure mutant or pure recombinant with a given probability. Ensemble of different mutation operators is also an interesting topic for improving the performance of DE, such as SaDE [18], EPSDE [20], SaJADE [21], CoDE [22], DE-SG [23], etc.

Apart from developing new mutation operators, some researchers investigated the selection of vectors in the existing mutation operators. In [4, pp. 61], Price *et al.* studied the vector index selection for DE, where the random selection, stochastic universal sampling selection, one-to-one selection, best-so-far base vector selection, and so on, are presented. Kaelo and Ali [24] proposed *tournament-best* base vector selection for DE, where the best vector among the three random ones is selected as the base vector and the remaining two are contributed to the difference vector in the "DE/rand/1" mutation operator. Inspired by the particle swarm optimization, Das *et al.* [15] proposed a modified "DE/current-to-best/1" mutation operator, namely local version of "DE/current-to-best/1", where all of the vectors are selected in the neighborhood of the target vector. In [25], Zhang and Sanderson presented the "DE/current-to-$p$best/1" mutation operator with optional archive, where the $\mathbf{x}^p_{best}$ is a $p$best solution, which is randomly selected as one of the top $100p\%$ solutions with $p \in (0, 1]$. When the archive $\mathbf{A}$ is used, $\mathbf{x}_{r_3}$ in Equation (5) is randomly chosen from the union, $\mathbf{P} \cup \mathbf{A}$, of the archive and current population $\mathbf{P}$. Epitropakis *et al.* [26] proposed the proximity-based mutation operators, in which the proximity characteristics among the vectors are used to assign the selection probabilities of different vectors. In [27], García-Martínez *et al.* presented the role differentiation and malleable mating for DE, where the vectors in the population are differentiated into four groups, *i.e.*, *receiving* group, *placing* group, *leading* group, and *correcting* group. In the mutation and crossover operations, the vectors are chosen from the corresponding groups, instead of the whole population.

III. Ranking-based Mutation Operators

As mentioned above, the DE algorithm is good at exploring the search space, however, it may be slow at exploitation of the solutions in the current population, especially when the best-so-far vector (*i.e.*, $\mathbf{x}_{best}$) is not used in the mutation operator. Thus, in order to improve the performance of DE, one possible way is to enhance its exploitation ability. Additionally, in the nature good species always contain good information and are more likely to be selected to propagate the offspring. Based on these considerations, in this section, in order to balance the exploration and exploitation abilities of DE we propose the ranking-based mutation operators, where some of the vectors in the mutation operators are proportionally chosen according to their rankings in the current population. The key points of our approach are described in detail as follows.

Algorithm 2 Ranking-based vector selection for "DE/rand/1"

```
1:  **Input**: The target vector index $i$
2:  **Output**: The selected vector indexes $r_1, r_2, r_3$
3:  Randomly select $r_1 \in \{1, Np\}$ {base vector index}
4:  **while** $\text{rndreal}[0,1) > p_{r_1}$ **or** $r_1 == i$ **do**
5:      Randomly select $r_1 \in \{1, Np\}$
6:  **end while**
7:  Randomly select $r_2 \in \{1, Np\}$ {terminal vector index}
8:  **while** $\text{rndreal}[0,1) > p_{r_2}$ **or** $r_2 == r_1$ **or** $r_2 == i$ **do**
9:      Randomly select $r_2 \in \{1, Np\}$
10: **end while**
11: Randomly select $r_3 \in \{1, Np\}$
12: **while** $r_3 == r_2$ **or** $r_3 == r_1$ **or** $r_3 == i$ **do**
13:     Randomly select $r_3 \in \{1, Np\}$
14: **end while**
```

A. Our Approach

1) Rankings Assignment: In order to utilize the information of good vectors in the DE population, in this work, we assign a ranking for each vector according to its fitness. Firstly, the population is sorted in ascent order (*i.e.*, from the best to the worst) based on the fitness of each vector. Then, the ranking of a vector is assigned as follows:

$$R_i = Np - i, \quad i = 1, 2, \cdots, Np \tag{9}$$

where Np is the population size. According to Equation (9), the best vector in the current population will obtain the highest ranking.

2) Selection Probability: After assigning the ranking for each vector, the selection probability p_i of the i-th vector $\mathbf{x}_i$ is calculated as

$$p_i = \frac{R_i}{Np}, \quad i = 1, 2, \cdots, Np \tag{10}$$

Note that the selection probability calculation is similar to the assignment of the emigration rate in biogeography-based optimization (BBO) [28]. In addition, it is worth pointing out that the probability calculation method in Equation (10) is also similar to the linear ranking fitness assignment presented in evolutionary algorithms [1]. Also, other methods can be used to replace the probability calculation in Equation (10) similar to the migration models of BBO presented in [29]. However, in this work, we only use the simplest method as shown in Equation (10). The influence of other probability calculation techniques on the performance of the ranking-based mutation operators will be evaluated in Section IV-D.

3) Vector Selection: After calculating the selection probability of each vector in Equation (10), the other issue is that in the mutation operator which vectors should be selected according to the selection probabilities. In this work, we select the *base* vector[2] and the *terminal* point of the difference vector based on their selection probabilities, while other vectors in the mutation operator are selected randomly as the original DE algorithm. For example, for the "DE/rand/1" mutation the vectors are selected as shown in Algorithm 2. Note that the notation "$a == b$" indicates a is equal to b. From Algorithm 2 we can see that the vectors with higher rankings (or selection probabilities) are more likely to be chosen as the base vector or the terminal point in the mutation operator. We do not select the starting point according to its selection probability, because if the two points in the difference vector are chosen from the better vectors, then the search step-size of the difference vector maybe decrease quickly and lead to premature convergence. The influence of other vector selection methods will be empirically compared in Section IV-E. Note that in Algorithm 2 we only illustrate the vector selection for "DE/rand/1", for other mutation operators the vector selection is similar to Algorithm 2.

B. DE with Ranking-based Mutation Operators

Combing our above-proposed ranking-based mutation operator with DE, the ranking-based DE algorithm (rank-DE for short) are presented. The pseudo-code of rank-DE with "DE/rand/1" mutation is shown in Algorithm 3. The differences between Algorithm 1 and Algorithm 3 are highlighted in "⇐". From Algorithm 3, it is clear that rank-DE maintains the advantages of the original DE algorithm, such as simple structure, ease of use, and so on. In addition, since some vectors in the mutation operator are chosen based on their rankings, better ones are more likely to be chosen. In this way, the exploitation ability of DE can be enhanced. Moreover, the ranking-based mutation operators are also able to integrate into other advanced DE variants.

Unlike proximity-based DE proposed in [26], our proposed rank-DE does not significantly increase the overall complexity of the original DE algorithm any more. The additional complexity of our proposed ranking-based DE is population sorting and probability calculation, as shown in Algorithm 3. The complexity of population sorting is $O(Np \cdot \log(Np))$, and the complexity of probability calculation is $O(Np)$. Since the total complexity of DE is $O(G \cdot Np \cdot D)$, where G is the maximal number of generations, rank-DE has the total complexity of $O(G \cdot Np \cdot (D + \log(Np) + 1))$. In general, the population size Np is set to be proportional to the problem dimension D in the DE literature [30]. Thus, the total complexity of rank-DE is $O(G \cdot D^2)$, which is the same as the original DE algorithm and many other DE variants.

Algorithm 3 DE with ranking-based "DE/rand/1" mutation

```
1: Generate the initial population randomly
2: Evaluate the fitness for each individual in the population
3: while the stop criterion is not satisfied do
4:    Sort the population based on the fitness of each individual    ⇐
5:    Calculate the selection probability for each individual according to Equation (10)    ⇐
6:    for $i = 1$ to $Np$ do
7:       Select $r_1, r_2, r_3$ as shown in Algorithm 2    ⇐
8:       $j_{rand}$ = rndint$(1, D)$
9:       for $j = 1$ to $D$ do
10:         if $\text{rndreal}_j[0,1) < Cr$ or $j$ is equal to $j_{rand}$ then
11:            $u_{i,j} = x_{r_1,j} + F \cdot (x_{r_2,j} - x_{r_3,j})$
12:         else
13:            $u_{i,j} = x_{i,j}$
14:         end if
15:      end for
16:   end for
17:   for $i = 1$ to $Np$ do
18:      Evaluate the offspring $\mathbf{u}_i$
19:      if $f(\mathbf{u}_i)$ is better than or equal to $f(\mathbf{x}_i)$ then
20:         Replace $\mathbf{x}_i$ with $\mathbf{u}_i$
21:      end if
22:   end for
23: end while
```

IV. Experimental Results and Analysis

In this section, we perform comprehensive experiments to verify the performance of the proposed ranking-based DE algorithm. We select 25 benchmark functions presented in the CEC-2005 competition [31] on real-parameter optimization as the test suite. These functions can be categorized into three groups: i) unimodal functions (F01 - F05); ii) basic multimodal functions (F06 - F12); iii) expanded multimodal functions (F13 - F14); and iv) hybrid composition functions (F15 - F25). More details for these functions can be found in [31].

TABLE I
PARAMETER SETTINGS FOR ALL DE VARIANTS.

Algorithm	Parameter settings
jDE, rank-jDE	$Np = 100, \tau_1 = 0.1, \tau_2 = 0.1$ [32]
ODE, rank-ODE	$Np = 100, Cr = 0.9, F = 0.5, J_r = 0.3$ [33]
SaDE, rank-SaDE	$Np = 50, LP = 50$ [18]
JADE, rank-JADE	$Np = 100, p = 0.05, c = 0.1$ [25]
CoDE, rank-CoDE	$Np = 30$ [22]
DEGL, rank-DEGL	$Np = 10 \times D, Cr = 0.9, F = 0.8$ [15]

A. Parameter Settings

In order to compare the results between ranking-based DE and its corresponding original DE, in all experiments, we use the following parameters as shown in Table I unless a change is mentioned. Note that we use jDE [32], a self-adaptive DE algorithm, to test the influence of our approach in different mutation operators, since this algorithm obtains promising results among various mutation operators. The ranking-based jDE algorithm is referred to as rank-jDE. To make a fair comparison, all parameters of DE variants in Table I are kept the same as used in their original literature.

[2]If the base vector is the best-so-far vector or the target vector, we do not need to select it based on its selection probability.

TABLE II
COMPARISON ON THE ERROR VALUES BETWEEN jDE AND ITS CORRESPONDING RANK-jDE WITH DIFFERENT MUTATION OPERATORS FOR FUNCTIONS F01 - F25 AT $D = 30$.

Prob	DE/rand/1/bin			DE/current-to-best/1/bin			DE/rand-to-best/1/bin		
	jDE		rank-jDE	jDE		rank-jDE	jDE		rank-jDE
F01*	7.37E+00 ± 3.02E+00	+	**8.93E-02 ± 4.02E-02**	1.56E-03 ± 9.82E-04	+	**1.31E-04 ± 1.05E-04**	1.06E-05 ± 1.13E-05	+	**2.59E-08 ± 1.80E-08**
F02	1.08E-05 ± 1.54E-05	+	**1.44E-11 ± 2.64E-11**	4.97E-11 ± 1.82E-10	+	**1.15E-12 ± 4.65E-12**	2.77E-17 ± 5.78E-17	+	**3.30E-22 ± 7.43E-22**
F03	1.89E+05 ± 1.04E+05	+	**8.12E+04 ± 3.87E+04**	3.85E+04 ± 2.81E+04	+	**3.08E+04 ± 2.71E+04**	3.90E+04 ± 2.50E+04	+	**2.59E+04 ± 1.77E+04**
F04	2.98E-01 ± 5.78E-01	+	**7.98E-04 ± 1.65E-03**	**1.08E+00 ± 2.95E+00**	=	1.29E+00 ± 5.24E+00	**1.58E-03 ± 4.63E-03**	=	5.04E-02 ± 2.25E-01
F05	**1.10E+03 ± 4.44E+02**	=	1.11E+03 ± 5.67E+02	2.26E+03 ± 6.82E+02	+	**2.07E+03 ± 6.00E+02**	**1.67E+03 ± 4.94E+02**	=	1.82E+03 ± 5.54E+02
F06	2.46E+01 ± 2.57E+01	+	**5.74E-01 ± 1.37E+00**	9.31E+00 ± 1.70E+01	+	**2.93E+00 ± 4.22E+00**	1.52E+00 ± 1.95E+00	+	**1.44E+00 ± 1.93E+00**
F07	1.31E-02 ± 9.30E-03	+	**9.75E-03 ± 8.92E-03**	1.60E-02 ± 1.26E-02	+	**1.44E-02 ± 1.32E-02**	**1.42E-02 ± 1.44E-02**	=	1.53E-02 ± 1.36E-02
F08	2.09E+01 ± 4.94E-02	=	2.09E+01 ± 4.98E-02	2.10E+01 ± 4.20E-02	=	2.10E+01 ± 4.91E-02	2.09E+01 ± 5.44E-02	=	2.09E+01 ± 5.16E-02
F09*	7.64E+01 ± 8.36E+00	+	**6.42E+01 ± 9.08E+00**	8.92E+01 ± 8.62E+00	+	**8.74E+01 ± 9.69E+00**	6.13E+01 ± 7.94E+00	+	**5.01E+01 ± 7.77E+00**
F10	5.86E+01 ± 1.05E+01	+	**4.71E+01 ± 9.42E+00**	4.44E+01 ± 8.41E+00	=	**4.41E+01 ± 9.55E+00**	**3.49E+01 ± 8.04E+00**	=	3.71E+01 ± 9.61E+00
F11	2.80E+01 ± 1.74E+00	=	**2.79E+01 ± 2.29E+00**	2.57E+01 ± 1.54E+00	+	**2.46E+01 ± 1.63E+00**	2.72E+01 ± 1.69E+00	+	**2.48E+01 ± 5.31E+00**
F12	1.16E+04 ± 8.08E+03	+	**1.65E+03 ± 1.80E+03**	**2.05E+03 ± 2.13E+03**	=	2.48E+03 ± 2.93E+03	**1.59E+03 ± 2.32E+03**	=	1.91E+03 ± 2.57E+03
F13	1.70E+00 ± 1.43E-01	+	1.60E+00 ± 1.26E-01	**1.68E+00 ± 2.63E-01**	–	1.80E+00 ± 2.36E-01	**1.55E+00 ± 1.33E-01**	–	1.63E+00 ± 2.34E-01
F14	1.30E+01 ± 2.00E-01	=	1.30E+01 ± 2.05E-01	1.26E+01 ± 2.56E-01	=	1.26E+01 ± 2.68E-01	1.28E+01 ± 2.83E-01	=	1.28E+01 ± 2.66E-01
F15	**3.40E+02 ± 1.09E+02**	=	3.66E+02 ± 5.58E+01	**3.36E+02 ± 1.27E+02**	=	3.50E+02 ± 1.51E+02	**3.28E+02 ± 1.38E+02**	=	3.58E+02 ± 9.00E+01
F16	7.56E+01 ± 8.99E+00	+	**6.12E+01 ± 9.00E+00**	**1.43E+02 ± 1.41E+02**	=	1.48E+02 ± 1.40E+02	1.51E+02 ± 1.57E+02	=	**1.42E+02 ± 1.55E+02**
F17	1.33E+02 ± 1.43E+01	+	**1.06E+02 ± 3.81E+01**	**1.58E+02 ± 1.19E+02**	=	1.92E+02 ± 1.47E+02	1.55E+02 ± 1.22E+02	+	**1.40E+02 ± 1.32E+02**
F18	**9.07E+02 ± 1.45E+00**	=	9.08E+02 ± 2.28E+00	**8.98E+02 ± 4.97E+01**	=	9.09E+02 ± 4.56E+01	**8.92E+02 ± 4.96E+01**	–	9.04E+02 ± 4.27E+01
F19	**9.06E+02 ± 1.72E+00**	–	9.08E+02 ± 1.90E+00	9.10E+02 ± 4.19E+01	=	**9.05E+02 ± 4.36E+01**	**8.90E+02 ± 5.11E+01**	–	8.97E+02 ± 4.95E+01
F20	**9.06E+02 ± 1.68E+00**	–	9.08E+02 ± 1.87E+00	9.12E+02 ± 3.83E+01	=	**9.10E+02 ± 3.79E+01**	**8.83E+02 ± 5.49E+01**	–	8.95E+02 ± 5.14E+01
F21	5.00E+02 ± 0.00E+00	=	5.00E+02 ± 0.00E+00	7.01E+02 ± 2.71E+02	=	**5.85E+02 ± 1.96E+02**	5.25E+02 ± 1.07E+02	=	**5.21E+02 ± 1.14E+02**
F22	9.04E+02 ± 1.03E+01	+	**8.97E+02 ± 1.16E+01**	**9.28E+02 ± 1.62E+01**	=	9.32E+02 ± 1.86E+01	9.19E+02 ± 9.89E+00	=	**9.18E+02 ± 1.52E+01**
F23	5.34E+02 ± 2.19E-04	=	5.34E+02 ± 1.20E-03	6.39E+02 ± 2.20E+02	=	**6.29E+02 ± 2.20E+02**	**5.35E+02 ± 1.91E+00**	–	5.52E+02 ± 7.98E+01
F24	2.00E+02 ± 0.00E+00	=	2.00E+02 ± 0.00E+00	2.00E+02 ± 0.00E+00	=	2.00E+02 ± 0.00E+00	2.00E+02 ± 0.00E+00	=	2.00E+02 ± 0.00E+00
F25	2.10E+02 ± 3.33E-01	+	**2.09E+02 ± 2.76E-01**	**2.30E+02 ± 1.40E+02**	=	2.56E+02 ± 1.99E+02	2.10E+02 ± 7.77E-01	=	2.10E+02 ± 6.86E-01
$w/t/l$	$14/9/2$		–	$8/16/1$		–	$7/13/5$		–

Prob	DE/rand/2/bin			DE/current-to-best/2/bin			DE/rand-to-best/2/bin		
	jDE		rank-jDE	jDE		rank-jDE	jDE		rank-jDE
F01*	4.88E+01 ± 1.82E+01	+	**1.20E+00 ± 5.54E-01**	2.85E-01 ± 1.52E-01	+	**5.28E-02 ± 2.78E-02**	6.81E-03 ± 4.10E-03	+	**9.35E-06 ± 7.55E-06**
F02	2.83E-03 ± 7.26E-03	+	**6.79E-09 ± 2.49E-08**	5.05E-18 ± 1.68E-17	+	**5.89E-22 ± 1.94E-21**	1.64E-13 ± 5.10E-13	+	**7.77E-21 ± 2.50E-20**
F03	2.85E+05 ± 1.76E+05	+	**9.65E+04 ± 5.65E+04**	2.85E+04 ± 2.17E+04	+	**1.91E+04 ± 1.27E+04**	5.34E+04 ± 3.34E+04	+	**3.16E+04 ± 1.77E+04**
F04	6.06E+00 ± 1.32E+01	+	**3.38E-03 ± 6.47E-03**	1.27E-06 ± 3.28E-06	+	**4.46E-08 ± 1.40E-07**	6.78E-06 ± 2.80E-05	+	**5.08E-06 ± 3.02E-05**
F05	7.72E+02 ± 4.44E+02	+	**5.24E+02 ± 3.49E+02**	8.94E+02 ± 4.56E+02	+	**7.59E+02 ± 4.26E+02**	**7.01E+02 ± 4.12E+02**	=	7.45E+02 ± 4.02E+02
F06	1.92E+01 ± 1.77E+01	+	**9.87E-01 ± 1.74E+00**	1.74E+01 ± 2.35E+01	+	**7.92E+00 ± 1.36E+01**	2.34E+00 ± 2.54E+00	+	**6.42E-01 ± 1.47E+00**
F07	6.70E-03 ± 5.90E-03	+	**4.88E-03 ± 5.98E-03**	**8.94E-03 ± 1.12E-02**	=	1.04E-02 ± 1.04E-02	1.24E-02 ± 1.27E-02	=	**9.45E-03 ± 9.97E-03**
F08	2.10E+01 ± 4.45E-02	+	**2.09E+01 ± 4.99E-02**	2.09E+01 ± 4.17E-02	=	2.09E+01 ± 5.85E-02	2.10E+01 ± 4.95E-02	=	**2.09E+01 ± 4.85E-02**
F09*	9.56E+01 ± 1.03E+01	+	**8.65E+01 ± 1.09E+01**	1.06E+02 ± 1.03E+01	=	**1.02E+02 ± 1.08E+01**	7.77E+01 ± 9.04E+00	+	**6.98E+01 ± 9.53E+00**
F10	6.75E+01 ± 7.99E+00	+	**5.65E+01 ± 9.96E+00**	4.73E+01 ± 9.79E+00	+	**4.20E+01 ± 7.41E+00**	3.99E+01 ± 7.44E+00	+	**3.49E+01 ± 6.76E+00**
F11	2.88E+01 ± 1.76E+00	=	**2.87E+01 ± 1.46E+00**	2.55E+01 ± 1.59E+00	=	**2.54E+01 ± 1.53E+00**	**2.78E+01 ± 1.79E+00**	=	2.82E+01 ± 1.54E+00
F12	2.13E+04 ± 5.21E+03	+	**1.97E+04 ± 6.16E+03**	9.74E+03 ± 3.80E+03	+	**8.31E+03 ± 4.25E+03**	1.30E+04 ± 7.07E+03	+	**2.09E+03 ± 3.77E+03**
F13	1.80E+00 ± 1.64E-01	+	**1.67E+00 ± 1.61E-01**	1.72E+00 ± 1.59E-01	=	1.72E+00 ± 1.66E-01	**1.60E+00 ± 1.79E-01**	=	1.66E+00 ± 1.45E-01
F14	1.30E+01 ± 2.58E-01	=	1.30E+01 ± 2.35E-01	1.28E+01 ± 2.11E-01	=	1.28E+01 ± 2.49E-01	**1.29E+01 ± 2.10E-01**	=	1.30E+01 ± 2.10E-01
F15	**1.20E+02 ± 1.60E+02**	–	3.34E+02 ± 1.35E+02	**2.28E+02 ± 1.82E+02**	–	3.33E+02 ± 1.37E+02	3.62E+02 ± 1.18E+02	=	**3.56E+02 ± 1.03E+02**
F16	9.82E+01 ± 1.48E+01	+	**7.85E+01 ± 1.47E+01**	**9.17E+01 ± 3.34E+01**	+	1.10E+02 ± 1.08E+02	**9.12E+01 ± 8.50E+01**	+	9.61E+01 ± 9.71E+01
F17	1.60E+02 ± 2.02E+01	+	**1.36E+02 ± 1.71E+01**	1.32E+02 ± 3.52E+01	+	**1.24E+02 ± 4.07E+01**	**1.29E+02 ± 6.73E+01**	=	1.30E+02 ± 7.88E+01
F18	9.05E+02 ± 1.52E+01	+	**9.04E+02 ± 1.54E+00**	**8.80E+02 ± 5.07E+01**	–	9.05E+02 ± 2.69E+01	**9.07E+02 ± 1.58E+01**	+	9.09E+02 ± 1.60E+01
F19	9.05E+02 ± 1.52E+01	+	**9.04E+02 ± 1.67E+00**	**8.84E+02 ± 4.80E+01**	–	9.05E+02 ± 2.70E+01	**8.98E+02 ± 3.32E+01**	+	9.09E+02 ± 1.59E+01
F20	9.07E+02 ± 1.44E+00	+	**9.06E+02 ± 1.33E+00**	**8.85E+02 ± 4.82E+01**	–	9.01E+02 ± 3.40E+01	**9.00E+02 ± 3.00E+01**	+	9.07E+02 ± 2.22E+01
F21	5.00E+02 ± 0.00E+00	=	5.00E+02 ± 0.00E+00	**5.32E+02 ± 9.78E+01**	=	5.37E+02 ± 1.20E+02	5.00E+02 ± 0.00E+00	=	5.00E+02 ± 0.00E+00
F22	9.23E+02 ± 9.69E+00	+	**9.02E+02 ± 6.68E+00**	9.15E+02 ± 1.18E+01	+	**9.08E+02 ± 1.51E+01**	9.04E+02 ± 8.43E+00	+	**8.98E+02 ± 1.23E+01**
F23	5.34E+02 ± 1.37E-04	+	5.34E+02 ± 2.41E-04	5.71E+02 ± 1.28E+02	=	**5.66E+02 ± 1.09E+02**	**5.42E+02 ± 5.70E+01**	=	5.47E+02 ± 8.84E+01
F24	2.00E+02 ± 0.00E+00	=	2.00E+02 ± 0.00E+00	2.00E+02 ± 0.00E+00	=	2.00E+02 ± 0.00E+00	2.00E+02 ± 0.00E+00	=	2.00E+02 ± 0.00E+00
F25	2.09E+02 ± 2.40E-01	+	2.09E+02 ± 1.89E-01	2.09E+02 ± 2.63E-01	+	2.09E+02 ± 3.09E-01	2.09E+02 ± 2.46E-01	=	2.09E+02 ± 2.66E-01
$w/t/l$	$20/4/1$		–	$12/9/4$		–	$13/12/0$		–

$\star$ indicates that when several algorithms obtain the global optimum, the intermediate results are reported at $\mathrm{NFFEs} = 20,000$.
"+", "–", and "=" indicate our approach is respectively better than, worse than, or similar to its competitor according to the Wilcoxon signed-rank test at $\alpha = 0.05$.

The maximal number of fitness function evaluations (Max_NFFEs) are set to $D \cdot 10,000$ [31]. To compare the results of different algorithms, each function is optimized over 50 independent runs. We use the same set of initial random populations to evaluate different algorithms in a similar way done in [8], *i.e.*, all of the compared algorithms are started from the same initial population in each out of 50 runs. In addition, it is important to point out that the boundary-handling method has significant influence to the performance of DE [34]. Therefore, in order to make a fair comparison, in this work, for all mentioned DE methods we use the *reinitialization* method, *i.e.*, when one of the decision variable is beyond its boundary constraint, it is generated with the uniform distribution within the boundary [34].

B. Influence on jDE with Different Mutation Operators

In this section, we evaluate the effectiveness of our proposed ranking-based mutation operators in jDE. Six mutation operators (see Equations (3) - (8)) are used in the experimental study. Among these six mutation operators, three of them have one difference vectors, while the rest three ones have two difference vectors. Normally, the mutation operators with two difference vectors are more explorative. There are four mutation operators that utilize the best-so-far solution ($\mathbf{x}_{best}$); these operators always converges faster and are more exploitive, especially only with one difference vector.

The results for all functions at $D = 30$ are shown in Table II. The better values compared between jDE and its corresponding rank-jDE are highlighted in **boldface**. In order to compare the significance between two algorithms the paired Wilcoxon signed-rank test is used. In Table II, according to the Wilcoxon's test, the results are summarized as "$w/t/l$", which

TABLE III
COMPARISON ON THE ERROR VALUES BETWEEN ADVANCED DE AND ITS CORRESPONDING RANKING-BASED DE VARIANT FOR FUNCTIONS F01 - F25 AT $D = 30$.

Prob	jDE		rank-jDE	ODE		rank-ODE	SaDE		rank-SaDE
F01⋆	7.37E+00 ± 3.02E+00	+	**8.93E-02 ± 4.02E-02**	2.82E+00 ± 2.17E+00	+	**3.46E-02 ± 2.86E-02**	2.22E-05 ± 1.51E-05	+	**2.80E-08 ± 2.29E-08**
F02	1.08E-05 ± 1.54E-05	+	**1.44E-11 ± 2.64E-11**	3.68E-04 ± 5.56E-04	+	**1.43E-10 ± 2.39E-10**	1.53E-18 ± 9.14E-18	+	**2.13E-27 ± 2.03E-27**
F03	1.89E+05 ± 1.04E+05	+	**8.12E+04 ± 3.87E+04**	5.86E+05 ± 2.80E+05	+	**2.52E+05 ± 1.57E+05**	5.40E+04 ± 4.32E+04	+	**1.85E+04 ± 1.68E+04**
F04	2.98E-01 ± 5.78E-01	+	**7.98E-04 ± 1.65E-03**	1.95E-01 ± 4.65E-01	+	**6.76E-05 ± 2.14E-04**	3.66E-01 ± 1.39E+00	+	**3.81E-02 ± 2.55E-01**
F05	**1.10E+03 ± 4.44E+02**	=	1.11E+03 ± 5.67E+02	1.55E+02 ± 1.30E+02	+	**2.68E+01 ± 3.21E+01**	1.58E+03 ± 5.06E+02	+	**1.13E+03 ± 4.18E+02**
F06	2.46E+01 ± 2.57E+01	+	**5.74E-01 ± 1.37E+00**	4.56E+01 ± 2.82E+01	+	**1.42E+01 ± 8.73E+00**	2.09E+00 ± 2.93E+00	+	**1.20E+00 ± 1.85E+00**
F07	1.31E-02 ± 9.30E-03	+	**9.75E-03 ± 8.92E-03**	**6.26E-03 ± 7.73E-03**	=	6.94E-03 ± 7.67E-03	**1.44E-02 ± 1.15E-02**	=	1.48E-02 ± 1.18E-02
F08	2.09E+01 ± 4.94E-02	=	2.09E+01 ± 4.98E-02	2.10E+01 ± 4.99E-02	=	**2.09E+01 ± 4.75E-02**	2.09E+01 ± 5.80E-02	=	2.09E+01 ± 5.19E-02
F09⋆	7.64E+01 ± 8.36E+00	+	**6.42E+01 ± 9.08E+00**	2.26E+02 ± 1.73E+01	+	**2.05E+02 ± 2.04E+01**	7.16E+01 ± 8.07E+00	+	**6.05E+01 ± 6.66E+00**
F10	5.86E+01 ± 1.05E+01	+	**4.71E+01 ± 9.42E+00**	5.13E+01 ± 4.54E+01	+	**3.78E+01 ± 2.28E+01**	4.81E+01 ± 7.26E+00	+	**4.44E+01 ± 8.96E+00**
F11	2.80E+01 ± 1.74E+00	=	**2.79E+01 ± 2.29E+00**	**7.50E+00 ± 8.10E+00**	–	9.72E+00 ± 6.51E+00	2.83E+01 ± 3.22E+00	=	**2.77E+01 ± 4.18E+00**
F12	1.16E+04 ± 8.08E+03	+	**1.65E+03 ± 1.80E+03**	2.57E+03 ± 2.91E+03	=	**2.16E+03 ± 2.34E+03**	2.44E+03 ± 3.17E+03	+	**1.63E+03 ± 1.91E+03**
F13	1.70E+00 ± 1.43E-01	+	1.60E+00 ± 1.26E-01	7.08E+00 ± 2.43E+00	+	**2.87E+00 ± 7.73E-01**	**2.24E+00 ± 1.79E-01**	=	2.25E+00 ± 2.59E-01
F14	1.30E+01 ± 2.00E-01	=	1.30E+01 ± 2.05E-01	1.31E+01 ± 2.28E-01	+	**1.29E+01 ± 4.39E-01**	1.29E+01 ± 1.81E-01	+	**1.28E+01 ± 1.90E-01**
F15	**3.40E+02 ± 1.09E+02**	=	3.66E+02 ± 5.58E+01	4.18E+02 ± 3.88E+01	=	**4.12E+02 ± 5.58E+01**	3.86E+02 ± 6.71E+01	+	**3.45E+02 ± 9.59E+01**
F16	7.56E+01 ± 8.99E+00	+	**6.12E+01 ± 9.00E+00**	9.79E+01 ± 7.18E+01	+	**6.95E+01 ± 5.36E+01**	7.51E+01 ± 4.89E+01	+	**6.91E+01 ± 4.97E+01**
F17	1.33E+02 ± 1.43E+01	+	**1.06E+02 ± 3.81E+01**	1.48E+02 ± 8.04E+01	+	**1.14E+02 ± 9.11E+01**	1.43E+02 ± 5.05E+01	=	**1.34E+02 ± 7.75E+01**
F18	**9.07E+02 ± 1.45E+00**	=	9.08E+02 ± 2.28E+00	9.01E+02 ± 2.08E+01	–	9.01E+02 ± 2.09E+01	8.76E+02 ± 5.54E+01	=	**8.75E+02 ± 5.49E+01**
F19	**9.06E+02 ± 1.72E+00**	–	9.08E+02 ± 1.90E+00	**8.90E+02 ± 3.66E+01**	–	9.03E+02 ± 1.49E+01	**8.77E+02 ± 5.36E+01**	=	8.86E+02 ± 4.87E+01
F20	**9.06E+02 ± 1.68E+00**	–	9.08E+02 ± 1.87E+00	**8.92E+02 ± 3.42E+01**	–	8.97E+02 ± 2.89E+01	**8.75E+02 ± 5.45E+01**	–	8.95E+02 ± 4.22E+01
F21	5.00E+02 ± 0.00E+00	=	5.00E+02 ± 0.00E+00	5.00E+02 ± 0.00E+00	=	5.00E+02 ± 0.00E+00	5.06E+02 ± 4.24E+01	=	5.06E+02 ± 4.24E+01
F22	9.04E+02 ± 1.03E+01	+	**8.97E+02 ± 1.16E+01**	9.09E+02 ± 9.31E+00	+	**9.04E+02 ± 9.75E+00**	9.29E+02 ± 1.51E+01	+	**9.22E+02 ± 1.47E+01**
F23	5.34E+02 ± 2.19E-04	=	5.34E+02 ± 1.20E-03	5.34E+02 ± 3.08E-04	=	5.34E+02 ± 3.37E-04	5.34E+02 ± 1.20E-03	=	5.34E+02 ± 8.30E-03
F24	2.00E+02 ± 0.00E+00	=	2.00E+02 ± 0.00E+00	2.00E+02 ± 0.00E+00	=	2.00E+02 ± 0.00E+00	2.00E+02 ± 0.00E+00	=	2.00E+02 ± 0.00E+00
F25	2.10E+02 ± 3.33E-01	+	**2.09E+02 ± 2.76E-01**	2.09E+02 ± 2.29E-01	+	2.09E+02 ± 1.26E-01	2.10E+02 ± 3.34E-01	+	**2.09E+02 ± 2.72E-01**
$w/t/l$	$14/9/2$		–	$14/7/4$		–	$14/10/1$		–

Prob	JADE		rank-JADE	CoDE		rank-CoDE	DEGL		rank-DEGL
F01⋆	7.91E-04 ± 4.22E-04	+	**2.60E-04 ± 1.63E-04**	3.29E-02 ± 2.27E-02	+	**2.22E-04 ± 1.52E-04**	9.02E-03 ± 2.23E-03	+	**7.73E-04 ± 2.40E-04**
F02	4.39E-28 ± 1.51E-28	+	**2.99E-28 ± 1.25E-28**	3.57E-14 ± 8.14E-14	+	**4.73E-21 ± 7.52E-21**	4.89E-27 ± 1.26E-26	+	**9.12E-28 ± 2.22E-28**
F03	8.12E+03 ± 5.58E+03	=	**7.67E+03 ± 6.70E+03**	1.41E+05 ± 7.39E+04	+	**6.07E+04 ± 3.84E+04**	4.83E+04 ± 2.97E+04	=	**4.28E+04 ± 2.32E+04**
F04	8.15E-16 ± 2.97E-15	+	**5.61E-16 ± 3.03E-15**	6.79E-02 ± 2.87E-01	+	**1.08E-03 ± 3.74E-03**	6.95E-16 ± 2.86E-15	+	**4.27E-20 ± 1.94E-19**
F05	9.67E-02 ± 2.88E-01	+	**4.77E-02 ± 1.59E-01**	8.27E+02 ± 4.12E+02	+	**7.12E+02 ± 4.32E+02**	4.10E+02 ± 2.54E+02	+	**2.14E+02 ± 1.81E+02**
F06	8.24E+00 ± 2.44E+01	+	**7.74E-01 ± 3.87E+00**	**3.29E-08 ± 1.22E-07**	+	3.99E-01 ± 1.21E+00	6.72E+01 ± 4.99E+01	+	**4.89E+01 ± 2.71E+01**
F07	9.55E-03 ± 8.31E-03	+	**6.06E-03 ± 7.82E-03**	**5.71E-03 ± 6.79E-03**	–	9.65E-03 ± 8.37E-03	2.14E+02 ± 8.41E+01	+	**1.20E+02 ± 4.66E+01**
F08	2.09E+01 ± 1.43E-01	=	2.09E+01 ± 1.43E-01	2.09E+01 ± 4.66E-02	+	**2.08E+01 ± 3.47E-01**	2.09E+01 ± 5.30E-02	=	2.09E+01 ± 6.92E-02
F09⋆	8.12E+01 ± 8.81E+00	+	**7.86E+01 ± 7.01E+00**	8.03E+01 ± 8.04E+00	+	**6.31E+01 ± 9.02E+00**	1.98E+02 ± 1.49E+01	+	**1.93E+02 ± 1.23E+01**
F10	2.66E+01 ± 4.97E+00	+	**2.48E+01 ± 4.66E+00**	4.63E+01 ± 1.03E+01	=	**4.54E+01 ± 1.21E+01**	1.50E+02 ± 4.22E+01	+	**1.17E+02 ± 5.94E+01**
F11	**2.50E+01 ± 1.43E+00**	=	2.55E+01 ± 1.58E+00	**1.10E+01 ± 2.99E+00**	–	1.32E+01 ± 3.26E+00	2.54E+01 ± 1.63E+01	+	**1.62E+01 ± 1.67E+01**
F12	7.26E+03 ± 3.88E+03	+	**3.91E+03 ± 3.88E+03**	1.68E+03 ± 2.21E+03	=	**1.50E+03 ± 2.28E+03**	**5.61E+03 ± 4.73E+03**	=	6.62E+03 ± 5.82E+03
F13	**1.43E+00 ± 1.08E-01**	=	1.47E+00 ± 1.08E-01	3.25E+00 ± 1.16E+00	+	**1.82E+00 ± 4.99E-01**	1.17E+01 ± 1.42E+00	+	**1.06E+01 ± 2.49E+00**
F14	1.23E+01 ± 3.14E-01	=	**1.22E+01 ± 3.29E-01**	1.23E+01 ± 4.73E-01	=	1.23E+01 ± 5.27E-01	1.25E+01 ± 3.18E-01	+	**1.22E+01 ± 3.98E-01**
F15	**3.43E+02 ± 8.67E+01**	=	3.56E+02 ± 9.29E+01	4.04E+02 ± 1.98E+01	+	**3.82E+02 ± 9.84E+01**	3.45E+02 ± 8.77E+01	=	**3.44E+02 ± 8.86E+01**
F16	**7.78E+01 ± 8.76E+01**	=	8.83E+01 ± 1.12E+02	**6.80E+01 ± 1.33E+01**	=	6.92E+01 ± 1.43E+01	1.45E+02 ± 1.27E+02	+	**1.07E+02 ± 1.02E+02**
F17	1.13E+02 ± 9.17E+01	=	**1.05E+02 ± 8.49E+01**	**6.58E+01 ± 1.36E+01**	=	6.91E+01 ± 1.48E+01	2.22E+02 ± 1.01E+02	+	**1.73E+02 ± 1.38E+02**
F18	**8.95E+02 ± 3.90E+01**	=	9.02E+02 ± 2.62E+01	**8.91E+02 ± 4.01E+01**	=	8.98E+02 ± 3.32E+01	**8.70E+02 ± 5.52E+01**	=	8.84E+02 ± 4.77E+01
F19	**8.97E+02 ± 3.61E+01**	=	9.05E+02 ± 2.18E+01	**8.95E+02 ± 3.57E+01**	–	9.01E+02 ± 3.03E+01	**8.68E+02 ± 5.64E+01**	=	8.96E+02 ± 3.93E+01
F20	8.96E+02 ± 3.60E+01	=	**8.95E+02 ± 3.57E+01**	**8.96E+02 ± 3.57E+01**	–	9.02E+02 ± 2.78E+01	**8.73E+02 ± 5.50E+01**	=	8.91E+02 ± 4.32E+01
F21	5.00E+02 ± 0.00E+00	=	5.00E+02 ± 0.00E+00	5.00E+02 ± 0.00E+00	=	5.00E+02 ± 0.00E+00	5.34E+02 ± 9.82E+01	=	**5.18E+02 ± 7.20E+01**
F22	8.95E+02 ± 1.26E+01	+	**8.90E+02 ± 1.37E+01**	9.18E+02 ± 1.23E+01	+	**8.88E+02 ± 2.19E+01**	9.20E+02 ± 1.48E+01	+	**9.12E+02 ± 1.35E+01**
F23	5.34E+02 ± 1.29E-04	+	5.34E+02 ± 2.82E-03	5.34E+02 ± 4.29E-04	=	5.34E+02 ± 4.36E-04	5.88E+02 ± 1.35E+02	+	**5.72E+02 ± 1.22E+02**
F24	2.00E+02 ± 0.00E+00	=	2.00E+02 ± 0.00E+00	2.00E+02 ± 0.00E+00	=	2.00E+02 ± 0.00E+00	2.00E+02 ± 0.00E+00	=	2.00E+02 ± 0.00E+00
F25	2.09E+02 ± 1.22E-01	=	2.09E+02 ± 1.05E-01	2.09E+02 ± 2.47E-01	+	2.09E+02 ± 2.33E-01	2.94E+02 ± 2.39E+02	+	**2.46E+02 ± 1.50E+02**
$w/t/l$	$11/14/0$		–	$12/9/4$		–	$16/9/0$		–

⋆ indicates that when several algorithms obtain the global optimum, the intermediate results are reported at $\mathrm{NFFEs} = 20,000$.
"+", "–", and "=" indicate our approach is respectively better than, worse than, or similar to its competitor according to the Wilcoxon signed-rank test at $\alpha = 0.05$.

denotes that our proposed ranking-jDE wins in w functions, ties in t functions, and loses in l functions, compared with its corresponding jDE method.

With respect to the overall performance, from Table II we can see that in the majority of the test functions at $D = 30$ the ranking-based jDE methods obtain the significantly better errors values compared with their corresponding jDE methods. For example, with "DE/rand/1/bin" strategy, rank-jDE significantly improves the performance of jDE in 14 out of 25 functions, but only loses in 2 functions. With "DE/current-to-best/2/bin" strategy, rank-jDE wins in 12 functions, ties in 9 functions, and only loses in 4 functions according to the Wilcoxon's test results at $\alpha = 0.05$. The only exception is for the "DE/rand-to-best/1/bin" strategy, where rank-jDE improves jDE in 7 functions, but loses in 5 functions. For the rest 13 functions, both rank-jDE and jDE provides similar error values. The reasons might be two-fold: First, in "DE/rand-to-best/1/bin" the best-so-far solution is always used and there is only one difference vector, both of them make this strategy be more exploitative. When the ranking-based vector selection is applied to "DE/rand-to-best/1/bin", it may be over-exploitative, and hence, it leads to deteriorate the performance of rank-jDE. Second, as shown in Equation (7) the base vector $\mathbf{x}_{r_1}$ is also the starting point of $\mathbf{x}_{best} - \mathbf{x}_{r_1}$, in this way, the ranking-based selection of $\mathbf{x}_{r_1}$ may also deteriorate the improved performance of rank-jDE. Oppositely, although "DE/current-to-best/1/bin" is also more exploitative due to the best-so-far solution and one difference vector, rank-jDE gets significantly better results in 8 functions, but only loses in 1 functions. The reason is that the base vector $\mathbf{x}_i$, which is also the starting point of $\mathbf{x}_{best} - \mathbf{x}_i$, is not selected based on its ranking.

As mentioned above, DE mutation operators with two difference vectors are more explorative than those with only one difference vector. This can be verified according to the results shown in Table II. When the ranking-based mutation operators

TABLE IV
COMPARISON ON THE ERROR VALUES BETWEEN ADVANCED DE AND ITS CORRESPONDING RANKING-BASED DE VARIANT FOR FUNCTIONS F01 - F25 AT $D = 50$.

Prob	jDE		rank-jDE	ODE		rank-ODE	SaDE		rank-SaDE
F01*	4.88E-03 ± 2.15E-03	+	**2.29E-06 ± 1.32E-06**	9.65E-02 ± 9.15E-02	+	**6.30E-05 ± 8.72E-05**	7.83E-11 ± 6.33E-11	+	**2.23E-15 ± 2.96E-15**
F02	8.99E-02 ± 8.43E-02	+	**3.46E-05 ± 3.84E-05**	8.37E+00 ± 4.93E+00	+	**1.20E-03 ± 1.19E-03**	2.62E-10 ± 4.61E-10	+	**3.30E-18 ± 1.25E-17**
F03	5.30E+05 ± 3.05E+05	+	**3.25E+05 ± 1.33E+05**	3.84E+06 ± 1.40E+06	+	**6.15E+05 ± 2.38E+05**	1.50E+05 ± 5.63E+04	+	**7.22E+04 ± 3.57E+04**
F04	8.31E+02 ± 7.64E+02	+	**2.87E+02 ± 4.72E+02**	8.77E+02 ± 4.38E+02	+	**3.45E+01 ± 2.81E+01**	1.19E+03 ± 1.05E+03	+	**5.34E+02 ± 6.59E+02**
F05	**3.39E+03 ± 6.32E+02**	–	3.63E+03 ± 5.83E+02	2.30E+03 ± 3.66E+02	+	**2.13E+03 ± 3.75E+02**	4.60E+03 ± 8.83E+02	+	**4.10E+03 ± 7.06E+02**
F06	3.98E+01 ± 2.68E+01	+	**7.65E+00 ± 1.68E+01**	4.20E+04 ± 1.82E+05	+	**6.25E+01 ± 5.14E+01**	5.39E+00 ± 2.07E+01	+	**1.28E+00 ± 1.88E+00**
F07	**4.13E-03 ± 8.97E-03**	+	4.82E-03 ± 9.17E-03	1.38E-02 ± 1.44E-02	+	**6.15E-03 ± 7.95E-03**	7.46E-03 ± 1.32E-02	+	**5.90E-03 ± 1.11E-02**
F08	2.11E+01 ± 3.58E-02	=	2.11E+01 ± 3.81E-02	2.11E+01 ± 4.07E-02	=	2.11E+01 ± 3.94E-02	2.11E+01 ± 3.61E-02	=	2.11E+01 ± 4.26E-02
F09*	7.69E+01 ± 9.24E+00	+	**5.66E+01 ± 6.53E+00**	4.14E+02 ± 3.40E+01	+	**3.76E+02 ± 3.45E+01**	8.09E+01 ± 6.36E+00	+	**6.32E+01 ± 7.57E+00**
F10	1.00E+02 ± 1.31E+01	+	**7.66E+01 ± 1.85E+01**	1.12E+02 ± 1.03E+02	=	**8.22E+01 ± 6.18E+01**	1.27E+02 ± 1.54E+01	+	**1.02E+02 ± 1.91E+01**
F11	5.54E+01 ± 2.31E+00	+	**5.31E+01 ± 5.03E+00**	**1.67E+01 ± 9.27E+00**	–	2.02E+01 ± 5.78E+00	5.73E+01 ± 3.77E+00	=	**5.66E+01 ± 6.14E+00**
F12	3.71E+04 ± 2.27E+04	+	**6.20E+03 ± 6.10E+03**	1.12E+04 ± 9.58E+03	+	**7.61E+03 ± 7.34E+03**	1.10E+04 ± 9.51E+03	+	**6.17E+03 ± 6.14E+03**
F13	2.90E+00 ± 2.23E-01	+	**2.81E+00 ± 3.07E-01**	1.47E+01 ± 4.72E+00	+	**5.87E+00 ± 1.40E+00**	**4.89E+00 ± 6.23E-01**	=	4.94E+00 ± 6.14E-01
F14	2.26E+01 ± 3.03E-01	=	2.26E+01 ± 2.99E-01	2.29E+01 ± 2.55E-01	=	2.29E+01 ± 2.87E-01	2.25E+01 ± 2.31E-01	+	**2.24E+01 ± 1.68E-01**
F15	3.32E+02 ± 9.57E+01	+	**3.16E+02 ± 9.97E+01**	3.96E+02 ± 2.83E+01	+	**3.62E+02 ± 8.30E+01**	3.65E+02 ± 7.97E+01	=	**3.41E+02 ± 8.73E+01**
F16	8.54E+01 ± 8.91E+00	+	**6.93E+01 ± 1.61E+01**	8.70E+01 ± 7.40E+01	+	**6.55E+01 ± 4.34E+01**	8.64E+01 ± 9.91E+00	+	**7.81E+01 ± 9.28E+00**
F17	1.75E+02 ± 1.26E+01	+	**1.32E+02 ± 6.16E+01**	1.55E+02 ± 9.78E+01	+	**1.09E+02 ± 8.99E+01**	1.99E+02 ± 1.54E+01	+	**1.89E+02 ± 5.05E+01**
F18	**9.25E+02 ± 2.93E+00**	–	9.31E+02 ± 4.18E+00	**8.96E+02 ± 4.88E+01**	–	9.05E+02 ± 4.32E+01	**9.42E+02 ± 3.82E+01**	=	9.48E+02 ± 2.37E+01
F19	**9.25E+02 ± 3.10E+00**	–	9.30E+02 ± 4.58E+00	**8.94E+02 ± 5.07E+01**	–	9.03E+02 ± 4.57E+01	**9.46E+02 ± 3.95E+01**	+	9.48E+02 ± 1.49E+01
F20	**9.25E+02 ± 3.26E+00**	–	9.30E+02 ± 4.69E+00	**8.99E+02 ± 4.71E+01**	–	9.07E+02 ± 4.35E+01	**9.43E+02 ± 3.88E+01**	=	9.46E+02 ± 2.42E+01
F21	5.00E+02 ± 0.00E+00	=	5.00E+02 ± 0.00E+00	5.00E+02 ± 0.00E+00	=	5.00E+02 ± 0.00E+00	5.46E+02 ± 1.65E+02	+	**5.20E+02 ± 1.04E+02**
F22	9.43E+02 ± 1.26E+01	+	**9.40E+02 ± 1.11E+01**	9.59E+02 ± 1.11E+01	+	**9.57E+02 ± 1.19E+01**	9.75E+02 ± 8.38E+00	+	**9.67E+02 ± 1.01E+01**
F23	5.39E+02 ± 1.67E-05	–	5.39E+02 ± 7.98E-03	5.39E+02 ± 1.27E-02	=	5.39E+02 ± 2.34E-02	5.91E+02 ± 1.79E+02	=	**5.85E+02 ± 1.63E+02**
F24	2.00E+02 ± 0.00E+00	=	2.00E+02 ± 0.00E+00	2.00E+02 ± 0.00E+00	=	2.00E+02 ± 0.00E+00	2.00E+02 ± 0.00E+00	=	2.00E+02 ± 0.00E+00
F25	2.14E+02 ± 6.08E-01	=	2.14E+02 ± 8.89E-01	2.14E+02 ± 4.62E-01	+	**2.13E+02 ± 3.63E-01**	2.16E+02 ± 8.10E-01	+	**2.14E+02 ± 5.92E-01**
$w/t/l$	$15/5/5$		–	$15/6/4$		–	$17/8/0$		–

Prob	JADE		rank-JADE	CoDE		rank-CoDE	DEGL		rank-DEGL
F01*	1.73E-12 ± 2.24E-12	+	**9.02E-14 ± 1.02E-13**	5.09E+00 ± 2.68E+00	+	**2.56E-02 ± 1.36E-02**	5.36E-01 ± 3.05E-01	+	**5.25E-02 ± 6.47E-02**
F02	1.04E-26 ± 4.22E-27	+	**7.03E-27 ± 3.23E-27**	2.37E-08 ± 3.43E-08	+	**1.81E-12 ± 3.81E-12**	1.06E-11 ± 1.38E-11	+	**4.33E-15 ± 4.98E-15**
F03	1.54E+04 ± 7.61E+03	=	**1.46E+04 ± 5.95E+03**	1.83E+05 ± 7.17E+04	+	**1.08E+05 ± 4.98E+04**	2.43E+05 ± 8.27E+04	+	**1.97E+05 ± 7.32E+04**
F04	1.94E+00 ± 6.06E+00	+	**9.22E-01 ± 2.52E+00**	5.81E+02 ± 4.49E+02	+	**2.34E+02 ± 4.19E+02**	7.56E-03 ± 1.01E-02	+	**6.67E-04 ± 1.84E-03**
F05	**1.86E+03 ± 4.32E+02**	=	1.87E+03 ± 3.61E+02	3.45E+03 ± 5.34E+02	=	**3.38E+03 ± 5.54E+02**	2.61E+03 ± 5.92E+02	+	**2.22E+03 ± 3.90E+02**
F06	2.13E+00 ± 6.66E+00	+	**1.04E+00 ± 1.77E+00**	**1.13E+00 ± 2.07E+00**	=	1.28E+00 ± 1.87E+00	2.33E+03 ± 3.70E+03	+	**3.40E+02 ± 7.46E+02**
F07	**3.55E-03 ± 6.53E-03**	=	4.97E-03 ± 7.23E-03	5.90E-03 ± 1.04E-02	+	**4.43E-03 ± 8.38E-03**	8.65E+02 ± 1.53E+02	+	**6.40E+02 ± 1.19E+02**
F08	2.11E+01 ± 2.23E-01	=	**2.10E+01 ± 3.21E-01**	2.11E+01 ± 4.41E-02	=	2.11E+01 ± 3.59E-02	2.11E+01 ± 3.17E-02	=	2.11E+01 ± 4.37E-02
F09*	7.44E+01 ± 5.36E+00	+	**7.08E+01 ± 5.18E+00**	3.99E+02 ± 6.54E+01	+	**3.53E+02 ± 5.43E+01**	3.87E+02 ± 1.74E+01	=	**3.86E+02 ± 1.83E+01**
F10	6.15E+01 ± 9.23E+00	+	**5.49E+01 ± 8.55E+00**	**1.05E+02 ± 2.12E+01**	–	1.16E+02 ± 2.37E+01	3.33E+02 ± 6.60E+01	+	**2.82E+02 ± 1.13E+02**
F11	**5.16E+01 ± 2.41E+00**	=	5.19E+01 ± 2.50E+00	**2.84E+01 ± 4.95E+00**	–	3.26E+01 ± 4.98E+00	7.21E+01 ± 1.82E+00	+	**7.09E+01 ± 6.93E+00**
F12	1.66E+04 ± 2.04E+04	+	**1.43E+04 ± 1.60E+04**	7.31E+03 ± 6.04E+03	+	**5.21E+03 ± 7.44E+03**	4.16E+04 ± 3.73E+04	+	**3.21E+04 ± 3.05E+04**
F13	**2.70E+00 ± 1.61E-01**	–	2.74E+00 ± 1.40E-01	4.79E+00 ± 1.55E+00	+	**3.60E+00 ± 7.89E-01**	2.28E+01 ± 5.26E+00	=	**2.16E+01 ± 6.20E+00**
F14	**2.16E+01 ± 4.75E-01**	=	2.17E+01 ± 4.03E-01	2.19E+01 ± 6.22E-01	=	2.19E+01 ± 4.80E-01	2.24E+01 ± 2.59E-01	=	**2.23E+01 ± 2.97E-01**
F15	**3.02E+02 ± 9.76E+01**	=	3.24E+02 ± 9.60E+01	3.92E+02 ± 3.96E+01	+	**3.52E+02 ± 8.63E+01**	3.36E+02 ± 8.78E+01	+	**3.29E+02 ± 9.40E+01**
F16	6.35E+01 ± 5.58E+01	=	**6.17E+01 ± 3.46E+01**	**7.55E+01 ± 1.45E+01**	–	8.54E+01 ± 2.29E+01	2.35E+02 ± 9.17E+01	+	**1.55E+02 ± 1.21E+02**
F17	1.16E+02 ± 4.95E+01	+	**1.07E+02 ± 3.25E+01**	**7.27E+01 ± 2.25E+01**	–	7.94E+01 ± 1.48E+01	2.88E+02 ± 4.86E+01	+	**2.74E+02 ± 6.29E+01**
F18	**9.31E+02 ± 3.50E+01**	=	9.33E+02 ± 2.85E+01	9.33E+02 ± 2.85E+01	=	**9.32E+02 ± 2.89E+01**	**9.16E+02 ± 4.59E+01**	=	9.22E+02 ± 2.50E+01
F19	9.39E+02 ± 1.04E+01	+	**9.31E+02 ± 2.78E+01**	**9.31E+02 ± 2.78E+01**	=	9.32E+02 ± 2.93E+01	9.26E+02 ± 3.32E+01	=	9.26E+02 ± 2.97E+01
F20	9.37E+02 ± 1.12E+01	+	**9.32E+02 ± 2.05E+01**	9.32E+02 ± 2.05E+01	=	**9.31E+02 ± 2.87E+01**	9.29E+02 ± 2.75E+01	=	**9.25E+02 ± 3.15E+01**
F21	5.18E+02 ± 7.20E+01	=	5.18E+02 ± 7.20E+01	5.00E+02 ± 0.00E+00	=	5.00E+02 ± 0.00E+00	5.27E+02 ± 9.61E+01	+	**5.24E+02 ± 9.76E+01**
F22	9.44E+02 ± 1.24E+01	+	**9.39E+02 ± 1.16E+01**	9.52E+02 ± 2.48E+01	+	**9.30E+02 ± 1.64E+01**	9.78E+02 ± 1.03E+01	=	**9.74E+02 ± 1.11E+01**
F23	5.53E+02 ± 6.90E+01	+	**5.46E+02 ± 4.94E+01**	5.39E+02 ± 3.53E-03	=	5.39E+02 ± 6.64E-03	6.36E+02 ± 1.76E+02	+	**5.82E+02 ± 1.16E+02**
F24	2.00E+02 ± 0.00E+00	=	2.00E+02 ± 0.00E+00	2.00E+02 ± 0.00E+00	=	2.00E+02 ± 0.00E+00	2.09E+02 ± 1.96E+01	+	**2.02E+02 ± 6.45E+00**
F25	2.14E+02 ± 5.58E-01	=	2.14E+02 ± 5.92E-01	2.14E+02 ± 3.65E-01	=	2.14E+02 ± 5.92E-01	9.98E+02 ± 3.94E+02	+	**7.56E+02 ± 4.72E+02**
$w/t/l$	$12/12/1$		–	$10/11/4$		–	$17/8/0$		–

$\star$ indicates that when several algorithms obtain the global optimum, the intermediate results are reported at $\mathrm{NFFEs} = 50,000$.
"+", "–", and "=" indicate our approach is respectively better than, worse than, or similar to its competitor according to the Wilcoxon signed-rank test at $\alpha = 0.05$.

TABLE V
RESULTS OF THE MULTIPLE-PROBLEM WILCOXON'S TEST FOR ADVANCED DE VARIANTS FOR FUNCTIONS F01 - F25 AT $D = 30$.

Algorithm	R^+	R^-	p-value	at $\alpha = 0.05$	at $\alpha = 0.1$
rank-jDE vs jDE	202	74	5.22E-02	=	+
rank-ODE vs ODE	227	49	5.41E-03	+	+
rank-SaDE vs SaDE	223	53	8.26E-03	+	+
rank-JADE vs JADE	178	98	2.34E-01	=	=
rank-CoDE vs CoDE	176	100	2.59E-01	=	=
rank-DEGL vs DEGL	235	65	1.38E-02	+	+

TABLE VI
RESULTS OF THE MULTIPLE-PROBLEM WILCOXON'S TEST FOR ADVANCED DE VARIANTS FOR FUNCTIONS F01 - F25 AT $D = 50$.

Algorithm	R^+	R^-	p-value	at $\alpha = 0.05$	at $\alpha = 0.1$
rank-jDE vs jDE	203	73	4.84E-02	+	+
rank-ODE vs ODE	227	49	5.41E-03	+	+
rank-SaDE vs SaDE	265	35	4.94E-04	+	+
rank-JADE vs JADE	200	76	6.05E-02	=	+
rank-CoDE vs CoDE	190	86	1.19E-01	=	=
rank-DEGL vs DEGL	307	18	1.51E-05	+	+

are used for DE with two difference vectors, rank-jDE significantly improves jDE in the majority of the test functions in the three cases. rank-jDE is significantly better than jDE in $20, 12$, and 10 functions for "DE/rand/2/bin", "DE/current-to-best/2/bin", and "DE/rand-to-best/2/bin", respectively. It only respectively loses in $1, 4$, and 3 out of 25 functions.

With respect to the features of the benchmark functions, from the results shown in Table II, we can observe that:

- For the unimodal functions (F01 - F05), regardless of the mutation operator used in jDE, the ranking-based jDE variants consistently obtain better results than the non-ranking-based jDE variants. In 25 out of 30 cases, rank-jDEs significantly outperform non-rank-jDEs, and in the rest 5 cases there are no significant differences between rank-jDEs and their corresponding non-rank-jDEs. The reason is that the ranking-based mutation operator in-

TABLE VII
AVERAGE RANKINGS OF ALL DE VARIANTS BY THE FRIEDMAN TEST FOR ALL FUNCTIONS.

$D = 30$		$D = 50$	
Algorithm	Ranking	Algorithm	Ranking
jDE	8.34	jDE	6.38
rank-jDE	6.60	rank-jDE	5.44
ODE	8.06	ODE	7.46
rank-ODE	6.76	rank-ODE	5.64
SaDE	7.08	SaDE	8.76
rank-SaDE	5.70	rank-SaDE	7.02
JADE	5.10	JADE	**4.56**
rank-JADE	**4.76**	rank-JADE	**4.04**
CoDE	5.64	CoDE	6.42
rank-CoDE	**4.80**	rank-CoDE	5.28
DEGL	7.84	DEGL	9.06
rank-DEGL	7.32	rank-DEGL	7.94

creases the selection pressure on better solutions in the population, and hence, it can accelerate the original jDE method when solving the unimodal functions.

- For the basic multimodal functions (F06 - F12), the algorithm with over-exploitation may lead to trap into local optima. In our proposed ranking-based mutation operators, the solutions are selected proportionally to their selection probabilities; in this way, it can avoid over-exploiting the better solutions in the mutation. Therefore, our proposed ranking-based DE can improve the exploitation ability without deteriorating the exploration ability of the original DE method seriously. The results in Table II support this intuition. In the most of the cases (25 out of 42), the rank-jDEs still surpass non-rank-jDEs. While in the rest 17 cases, rank-jDEs provide similar results compared with their corresponding non-rank-jDEs.
- For the two expanded multimodal functions, there are no significant differences between rank-jDEs and their corresponding non-rank-jDEs in F14 for all mutation operators. However, in F13 ranking-based jDEs win in “DE/rand/1/bin” and “DE/rand/2/bin”, tie in “DE/current-to-best/2/bin” and “DE/rand-to-best/2/bin”, but lose in “DE/current-to-best/1/bin” and “DE/rand-to-best/1/bin” compared with their corresponding non-ranking-based jDEs. These further confirm that for the mutation operators with good exploration ability our method is able to balance the exploitation and exploration ability of DE, and hence, it can improve its performance. However, for the mutation operators with good exploitation ability (such as “DE/current-to-best/1/bin” and “DE/rand-to-best/1/bin”), the ranking-based mutation operators may slightly lead to over-exploitation when solving expanded multimodal functions.
- For the hybrid composition functions (F15 - F25), these functions are very difficult to solve for almost all existing optimizers. In all 66 cases, rank-jDEs win in 22 cases, tie in 33, but lose in 11 cases[3] compared with non-rank-jDEs. Similar to the results for expanded functions, rank-jDEs perform better when the mutation operators have good exploration ability. However, if the mutation operators are more exploitative, the ranking-based mutation operators may cause the algorithm over-exploitation, and thus they are not beneficial to the significant improvement of non-rank-jDEs for composition functions. To sum up, rank-jDEs still provide the better results in overall compared with non-rank-jDEs.

[3]By carefully looking at the results, we can see that in three functions (F18, F19, and F20), rank-jDEs lose in 8 cases compared with non-rank-jDEs. Indeed, these three functions have the same function but with different parameter settings [31].

In general, based on the results and analysis we can see that our proposed ranking-based mutation operators are able to enhance the exploitation ability. jDE with the ranking-based mutation operators improves the performance of the jDE algorithm, especially for the DE mutation operators with good exploration ability. The ranking-based jDEs are capable of surpassing the non-ranking-based jDEs in the unimodal and basic multimodal functions. In the more complex functions, such as expanded and/or composition multimodal functions, the rank-jDEs still slightly enhance the performance of the non-rank-jDEs. In the next section, we will test the influence of the ranking-based mutation on other advanced DE variants.

C. Effect on Advanced DE Variants

In order to better understand the effectiveness of the proposed ranking-based mutation operators, in this section, we incorporate the ranking-based mutation operators into some advanced DE variants. They are jDE [32], ODE [33], SaDE [18], JADE [25], CoDE [22], and DEGL [15], all of them obtained very promising results. jDE, SaDE, and JADE are adaptive DE variants, where the parameter adaptation are implemented. Note that in this work, for JADE the archive is employed for both JADE and rank-JADE. In ODE, the opposition-based learning is used for population initialization and jumping. In both SaDE and CoDE, ensemble of the multiple mutation strategies are presented. DEGL uses the local version and global version of “DE/current-to-best/1/bin”, and a parameter w is used to balance the influence of the two operators. In the above DE variants, when more than one strategies are adopted, the ranking-based vector selection technique is implemented for all of the strategies. For example, in DEGL, the ranking-based vector selection technique is used for both the local version and global version of “DE/current-to-best/1/bin”. To make a fair comparison between the advanced DEs and their corresponding ranking-based DEs, all parameters are kept the same as used in their original literature. The parameter settings are tabulated in Table I. The error values of all DE variants are respectively reported in Tables III and IV for functions F01 - F25 at $D = 30$ and $D = 50$. All results are averaged over 50 independent runs. The better results compared between ranking-based DE and non-ranking-based DE are highlighted in **boldface**. The Wilcoxon’s test is also used to compare the results between two algorithms. In addition, the multiple-problem statistical analysis based on the Wilcoxon’s test, as similar done in [35], [36], between ranking-based DE and non-ranking-based DE is reported for all functions in Tables V and VI, respectively. Moreover, according to the Friedman test, the final rankings of all DE variants for all functions are shown in Table VII. Note that the Friedman test, which is used to obtain the rankings of different algorithms for all problems, is calculated by the KEEL software [37]. In Table VII, the

TABLE VIII
COMPARISON ON THE ERROR VALUES FOR jDE VARIANTS WITH DIFFERENT PROBABILITY CALCULATION MODELS FOR FUNCTIONS F01 - F25 AT $D = 30$.

Prob	jDE		rank-jDE-q		rank-jDE-s		rank-jDE
F01*	7.37E+00 $\pm$ 3.02E+00	+	**6.46E-03 $\pm$ 3.97E-03**	–	**4.36E-02 $\pm$ 2.35E-02**	–	8.93E-02 $\pm$ 4.02E-02
F02	1.08E-05 $\pm$ 1.54E-05	+	**1.32E-14 $\pm$ 1.91E-14**	–	**7.55E-12 $\pm$ 2.00E-11**	–	1.44E-11 $\pm$ 2.64E-11
F03	1.89E+05 $\pm$ 1.04E+05	+	**9.97E+04 $\pm$ 8.66E+04**	=	1.18E+05 $\pm$ 6.56E+04	+	**8.12E+04 $\pm$ 3.87E+04**
F04	2.98E-01 $\pm$ 5.78E-01	+	8.26E-04 $\pm$ 4.13E-03	–	**6.56E-04 $\pm$ 1.30E-03**	=	**7.98E-04 $\pm$ 1.65E-03**
F05	**1.10E+03 $\pm$ 4.44E+02**	=	1.12E+03 $\pm$ 5.22E+02	=	**9.58E+02 $\pm$ 4.29E+02**	=	1.11E+03 $\pm$ 5.67E+02
F06	2.46E+01 $\pm$ 2.57E+01	+	1.36E+00 $\pm$ 1.91E+00	=	**9.20E-01 $\pm$ 1.63E+00**	=	**5.74E-01 $\pm$ 1.37E+00**
F07	1.31E-02 $\pm$ 9.30E-03	+	1.25E-02 $\pm$ 1.18E-02	=	**1.00E-02 $\pm$ 9.20E-03**	=	**9.75E-03 $\pm$ 8.92E-03**
F08	2.09E+01 $\pm$ 4.94E-02	=	2.09E+01 $\pm$ 4.71E-02	=	2.09E+01 $\pm$ 4.57E-02	=	2.09E+01 $\pm$ 4.98E-02
F09*	7.64E+01 $\pm$ 8.36E+00	+	**5.57E+01 $\pm$ 6.59E+00**	–	**6.11E+01 $\pm$ 9.91E+00**	–	6.42E+01 $\pm$ 9.08E+00
F10	5.86E+01 $\pm$ 1.05E+01	+	**3.44E+01 $\pm$ 9.63E+00**	–	**4.22E+01 $\pm$ 1.06E+01**	–	4.71E+01 $\pm$ 9.42E+00
F11	2.80E+01 $\pm$ 1.74E+00	=	**2.61E+01 $\pm$ 5.28E+00**	–	**2.79E+01 $\pm$ 1.95E+00**	=	2.79E+01 $\pm$ 2.29E+00
F12	1.16E+04 $\pm$ 8.08E+03	+	1.76E+03 $\pm$ 2.04E+03	=	**1.64E+03 $\pm$ 2.46E+03**	=	1.65E+03 $\pm$ 1.80E+03
F13	1.70E+00 $\pm$ 1.43E-01	+	**1.55E+00 $\pm$ 2.12E-01**	=	**1.55E+00 $\pm$ 1.76E-01**	–	1.60E+00 $\pm$ 1.26E-01
F14	**1.30E+01 $\pm$ 2.00E-01**	=	1.30E+01 $\pm$ 2.12E-01	=	**1.29E+01 $\pm$ 2.62E-01**	=	1.30E+01 $\pm$ 2.05E-01
F15	**3.40E+02 $\pm$ 1.09E+02**	=	3.78E+02 $\pm$ 6.48E+01	=	**3.64E+02 $\pm$ 5.63E+01**	=	3.66E+02 $\pm$ 5.58E+01
F16	7.56E+01 $\pm$ 8.99E+00	+	**5.91E+01 $\pm$ 1.85E+01**	–	6.34E+01 $\pm$ 1.63E+01	=	**6.12E+01 $\pm$ 9.00E+00**
F17	1.33E+02 $\pm$ 1.43E+01	+	**7.83E+01 $\pm$ 3.55E+01**	–	**8.94E+01 $\pm$ 3.19E+01**	–	1.06E+02 $\pm$ 3.81E+01
F18	**9.07E+02 $\pm$ 1.45E+00**	=	9.09E+02 $\pm$ 2.81E+00	+	**9.08E+02 $\pm$ 2.17E+00**	=	9.08E+02 $\pm$ 2.28E+00
F19	**9.06E+02 $\pm$ 1.72E+00**	–	9.09E+02 $\pm$ 2.03E+00	=	9.08E+02 $\pm$ 1.92E+00	=	**9.08E+02 $\pm$ 1.90E+00**
F20	**9.06E+02 $\pm$ 1.68E+00**	–	9.09E+02 $\pm$ 2.61E+00	+	9.08E+02 $\pm$ 2.22E+00	=	**9.08E+02 $\pm$ 1.87E+00**
F21	5.00E+02 $\pm$ 0.00E+00	=	5.00E+02 $\pm$ 0.00E+00	=	5.00E+02 $\pm$ 0.00E+00	=	5.00E+02 $\pm$ 0.00E+00
F22	9.04E+02 $\pm$ 1.03E+01	+	8.99E+02 $\pm$ 1.27E+01	=	**8.99E+02 $\pm$ 9.50E+00**	=	**8.97E+02 $\pm$ 1.16E+01**
F23	5.34E+02 $\pm$ 2.19E-04	=	5.34E+02 $\pm$ 1.20E-03	=	5.34E+02 $\pm$ 2.77E-04	=	5.34E+02 $\pm$ 1.20E-03
F24	2.00E+02 $\pm$ 0.00E+00	=	2.00E+02 $\pm$ 0.00E+00	=	2.00E+02 $\pm$ 0.00E+00	=	2.00E+02 $\pm$ 0.00E+00
F25	2.10E+02 $\pm$ 3.33E-01	+	2.09E+02 $\pm$ 3.01E-01	=	**2.09E+02 $\pm$ 2.63E-01**	=	**2.09E+02 $\pm$ 2.76E-01**
$w/t/l$	$14/9/2$		$2/15/8$		$1/18/6$		–

$\star$ indicates that when several algorithms obtain the global optimum, the intermediate results are reported at $\mathrm{NFFEs} = 20,000$.
"+", "–", and "=" indicate our approach is respectively better than, worse than, or similar to its competitor according to the Wilcoxon signed-rank test at $\alpha = 0.05$.

overall best and the second best results within all DE variants are highlighted in **grey boldface** and **boldface**, respectively.

TABLE IX
RESULTS OF THE MULTIPLE-PROBLEM WILCOXON'S TEST FOR rank-jDE VARIANTS WITH DIFFERENT PROBABILITY CALCULATION MODELS FOR FUNCTIONS F01 - F25 AT $D = 30$.

Algorithm	R^+	R^-	p-value	at $\alpha = 0.05$	at $\alpha = 0.1$
rank-jDE vs jDE	202	74	5.22E-02	=	+
rank-jDE vs rank-jDE-q	164	112	4.45E-01	=	=
rank-jDE vs rank-jDE-s	109	167	3.93E-01	=	=

For all functions at $D = 30$, Table III shows that in the majority of the test functions the ranking-based DE methods provide significantly better results compared with their corresponding non-ranking-based DE methods. For example, rank-JADE wins in 11 functions, ties in 14 functions compared with JADE. There is no function that JADE can significantly outperform rank-JADE. Additionally, according to the results of multiple-problem statistical analysis shown in Table V we can see that ranking-based DEs consistently get higher R^+ values than R^- values in all cases compared with the non-ranking-based DEs. This means that the ranking-based DE is better than its original DE for all functions. For the Wilcoxon's test at $\alpha = 0.05$ in three cases (rank-ODE vs ODE, rank-SaDE vs SaDE, and rank-DEGL vs DEGL) there are significant differences for all problems between ranking-based DE and non-ranking-based DE. At $\alpha = 0.1$ there are four cases (rank-jDE vs jDE, rank-ODE vs ODE, rank-SaDE vs SaDE, and rank-DEGL vs DEGL), where the significant differences are observed. This indicates that ranking-based DE is significantly better than its corresponding non-ranking-based DE based on the multiple-problem statistical analysis in these four cases at $\alpha = 0.1$. With respect to the rankings of different algorithms by the Friedman test, Table VII clearly shows that all rank-DEs consistently obtain better rankings compared with their corresponding non-rank-DEs. Overall, rank-JADE gets the first ranking, followed by rank-CoDE for all functions at $D = 30$.

TABLE XI
RESULTS OF THE MULTIPLE-PROBLEM WILCOXON'S TEST FOR jDE VARIANTS COMPARED WITH OUR PROPOSED rank-jDE METHOD FOR FUNCTIONS F01 - F25 AT $D = 30$.

Algorithm	R^+	R^-	p-value	at $\alpha = 0.05$	at $\alpha = 0.1$
rank-jDE vs jDE	202	74	5.22E-02	=	+
rank-jDE vs rank-jDE1	193	83	9.80E-02	=	+
rank-jDE vs rank-jDE2	191	85	1.11E-01	=	=

From Tables IV and VI, similar to the results for all functions at $D = 30$, it is clear that ranking-based DE approaches also consistently outperform their non-ranking-based DE methods in the majority of the test functions at $D = 50$. rank-jDE, rank-ODE, rank-SaDE, rank-JADE, rank-CoDE, and rank-DEGL significantly improve jDE, ODE, SaDE, JADE, CoDE, and DEGL in $15, 15, 17, 12, 9$, and 16 out of 25 functions, respectively. Also, with respect to the multiple-problem analysis DE based on ranking-based mutation operators obtains significantly better results in 4 cases at $\alpha = 0.05$ and in 5 cases at $\alpha = 0.1$. Moreover, considering the final rankings of all algorithms in Table VII, we can see that rank-JADE obtains the overall best ranking, followed by JADE and rank-CoDE for all functions at $D = 50$. Again, all rank-DEs get the better final rankings compared with their corresponding non-rank-DEs according to the Friedman test.

Overall, from results shown in Tables III - VI, we can conclude that our proposed ranking-based mutation operators are also capable of improving the performance of the recently presented advanced DE variants.

D. Influence of Different Probability Calculation Models

In the previous experiments, we verified the effectiveness of our proposed ranking-based mutation operators in various DE

TABLE X
COMPARISON ON THE ERROR VALUES FOR jDE VARIANTS WITH DIFFERENT VECTOR SELECTION METHODS FOR FUNCTIONS F01 - F25 AT $D = 30$.

Prob	jDE		rank-jDE1		rank-jDE2		rank-jDE
F01⋆	7.37E+00 ± 3.02E+00	+	1.42E-01 ± 6.36E-02	+	**5.61E-02 ± 2.60E-02**	–	**8.93E-02 ± 4.02E-02**
F02	1.08E-05 ± 1.54E-05	+	**1.22E-10 ± 1.86E-10**	+	4.78E-10 ± 1.05E-09	+	**1.44E-11 ± 2.64E-11**
F03	1.89E+05 ± 1.04E+05	+	**9.10E+04 ± 4.55E+04**	=	1.10E+05 ± 5.43E+04	+	**8.12E+04 ± 3.87E+04**
F04	2.98E-01 ± 5.78E-01	+	**4.09E-03 ± 9.52E-03**	+	1.79E-01 ± 5.23E-01	+	**7.98E-04 ± 1.65E-03**
F05	**1.10E+03 ± 4.44E+02**	=	**1.11E+03 ± 4.42E+02**	=	1.60E+03 ± 4.72E+02	+	1.11E+03 ± 5.67E+02
F06	2.46E+01 ± 2.57E+01	+	**2.31E+00 ± 2.43E+00**	+	8.45E+00 ± 1.72E+01	+	**5.74E-01 ± 1.37E+00**
F07	1.31E-02 ± 9.30E-03	+	**1.15E-02 ± 8.52E-03**	=	1.85E-02 ± 1.21E-02	+	**9.75E-03 ± 8.92E-03**
F08	2.09E+01 ± 4.94E-02	=	2.09E+01 ± 4.45E-02	=	2.09E+01 ± 6.07E-02	=	2.09E+01 ± 4.98E-02
F09⋆	7.64E+01 ± 8.36E+00	+	6.57E+01 ± 9.89E+00	+	**6.03E+01 ± 1.02E+01**	–	**6.42E+01 ± 9.08E+00**
F10	5.86E+01 ± 1.05E+01	+	**4.43E+01 ± 1.02E+01**	=	**3.99E+01 ± 1.22E+01**	–	4.71E+01 ± 9.42E+00
F11	2.80E+01 ± 1.74E+00	=	**2.80E+01 ± 1.63E+00**	=	2.81E+01 ± 2.81E+00	=	**2.79E+01 ± 2.29E+00**
F12	1.16E+04 ± 8.08E+03	+	2.43E+03 ± 4.16E+03	=	**1.91E+03 ± 2.54E+03**	=	**1.65E+03 ± 1.80E+03**
F13	1.70E+00 ± 1.43E-01	+	**1.58E+00 ± 1.43E-01**	=	**1.57E+00 ± 1.75E-01**	=	1.60E+00 ± 1.26E-01
F14	**1.30E+01 ± 2.00E-01**	=	1.31E+01 ± 1.83E-01	=	1.30E+01 ± 2.34E-01	=	**1.30E+01 ± 2.05E-01**
F15	**3.40E+02 ± 1.09E+02**	=	**3.60E+02 ± 7.59E+01**	=	3.84E+02 ± 6.18E+01	+	3.66E+02 ± 5.58E+01
F16	7.56E+01 ± 8.99E+00	+	6.25E+01 ± 1.12E+01	=	**6.03E+01 ± 1.78E+01**	=	**6.12E+01 ± 9.00E+00**
F17	1.33E+02 ± 1.43E+01	+	1.11E+02 ± 2.21E+01	=	**9.00E+01 ± 3.29E+01**	–	**1.06E+02 ± 3.81E+01**
F18	**9.07E+02 ± 1.45E+00**	=	**9.06E+02 ± 1.55E+01**	=	9.10E+02 ± 2.79E+00	+	9.08E+02 ± 2.28E+00
F19	**9.06E+02 ± 1.72E+00**	–	9.08E+02 ± 2.17E+00	=	9.10E+02 ± 2.49E+00	+	**9.08E+02 ± 1.90E+00**
F20	**9.06E+02 ± 1.68E+00**	–	9.09E+02 ± 2.08E+00	=	9.10E+02 ± 2.45E+00	+	**9.08E+02 ± 1.87E+00**
F21	5.00E+02 ± 0.00E+00	=	5.00E+02 ± 0.00E+00	=	5.00E+02 ± 0.00E+00	=	5.00E+02 ± 0.00E+00
F22	9.04E+02 ± 1.03E+01	+	**9.00E+02 ± 1.25E+01**	+	9.03E+02 ± 1.10E+01	+	**8.97E+02 ± 1.16E+01**
F23	5.34E+02 ± 2.19E-04	=	5.34E+02 ± 3.56E-04	=	5.34E+02 ± 3.35E-04	=	5.34E+02 ± 1.20E-03
F24	2.00E+02 ± 0.00E+00	=	2.00E+02 ± 0.00E+00	=	2.00E+02 ± 0.00E+00	=	2.00E+02 ± 0.00E+00
F25	2.10E+02 ± 3.33E-01	+	**2.09E+02 ± 3.01E-01**	=	2.10E+02 ± 3.64E-01	+	**2.09E+02 ± 2.76E-01**
$w/t/l$	14/9/2		6/19/0		12/9/4		–

⋆ indicates that when several algorithms obtain the global optimum, the intermediate results are reported at $\mathrm{NFFEs} = 20,000$.
"+", "–", and "=" indicate our approach is respectively better than, worse than, or similar to its competitor according to the Wilcoxon signed-rank test at $\alpha = 0.05$.

variants. The linear model is used as an illustration to calculate the selection probabilities according to the rankings. Actually, other models for calculating the selection probabilities can also be used in our proposed ranking-based mutation operators. Similar to the models presented in [29], in this section, two models, *i.e.*, quadratic model and sinusoidal modal, are adopted to evaluate the influence of different probability calculation models to rank-jDE. The quadratic model is as follows:

$$p_i = \left(\frac{R_i}{Np}\right)^2 \tag{11}$$

The sinusoidal model is formulated as:

$$p_i = 0.5 \cdot \left(1.0 - \cos\left(\frac{R_i \cdot \pi}{Np}\right)\right) \tag{12}$$

rank-jDE with the quadratic model is namely rank-jDE-q, and that with the sinusoidal modal is referred to as rank-jDE-s. For all compared algorithms, the parameter settings are used as described in Table I. The errors values are reported in Table VIII for all functions at $D = 30$. The overall best and the second best results among the four jDE variants are respectively highlighted in **grey boldface** and **boldface**. In addition, the results of the multiple-problem analysis are shown in Table IX. According the results we can see that rank-jDE-s obtains the overall best results among the four jDE methods, and all of these three rank-jDE methods get better results than jDE for all functions. The results also indicate that the linear model in the ranking-based mutation operators is a reasonable choice, but not the optimal one. It is worth pointing out that this experiment is not to seek the optimal probability calculation model, but only to evaluate the influence of different models. In the future work, we will comprehensively test different probability models in the ranking-based mutation operators.

E. Comparison on Vector Selection

In Section III-A, we mentioned that in our proposed ranking-based mutation operators only the base vector and the terminal point are chosen based on their rankings. In order to evaluate the influence of other different vector selection methods on the performance of DE, in this section, rank-jDE is compared with jDE, rank-jDE1, and rank-jDE2. In rank-jDE1, only the base vector is selected based on the ranking, while other vectors in the mutation are selected randomly as used in the original jDE method. In rank-jDE2, all vectors (including the starting point) are selected based on their rankings. All the parameter settings are kept the same as described in Table I. The results for all functions at $D = 30$ are reported in Table X, and the results of the multiple-problem analysis are tabulated in Table XI. The overall best and the second best results among the four jDE variants are highlighted in **grey boldface** and **boldface**, respectively.

According to the error values in Table X, the p-value computed by Iman-Daveport test, which is used to check the differences between all algorithms for all functions, is $2.61E-02$ for all functions at $D = 30$. It means that there are significant differences between the compared algorithms for all functions at $\alpha = 0.05$. rank-jDE wins in $14, 6$, and 12 out of 25 functions compared with jDE, rank-jDE1, and rank-jDE2, respectively. Compared with rank-jDE1, in 19 functions there are no significant differences between rank-jDE and rank-jDE1. There is no function that rank-jDE1 wins rank-jDE. In rank-jDE2, since all vectors are chosen based on the rankings, it is the most exploitative method among the four jDE variants. In some relatively simple functions (*e.g.* F01, F09, F10), rank-jDE2 obtains significantly better results than rank-jDE, however, it loses in 12 functions. In the rest 9 functions both rank-jDE2 and rank-jDE get similar error values.

Based on the results of the multiple-problem analysis shown

TABLE XII
COMPARISON ON THE ERROR VALUES OF jDE AND RANK-jDE WITH DIFFERENT POPULATION SIZE FOR FUNCTIONS F01 - F25 AT $D = 30$.

Prob	$Np = 50$			$Np = 150$			$Np = 200$		
	jDE		rank-jDE	jDE		rank-jDE	jDE		rank-jDE
F01⋆	4.10E-04 ± 3.09E-04	+	**2.08E-07 ± 1.99E-07**	1.93E+02 ± 4.15E+01	+	**1.24E+01 ± 3.76E+00**	9.10E+02 ± 1.57E+02	+	**1.22E+02 ± 2.92E+01**
F02	5.49E-10 ± 8.42E-10	+	**6.80E-17 ± 2.20E-16**	4.49E-03 ± 5.84E-03	+	**5.16E-08 ± 1.29E-07**	1.77E-01 ± 1.69E-01	+	**4.99E-06 ± 5.90E-06**
F03	1.34E+05 ± 7.17E+04	+	**8.20E+04 ± 6.12E+04**	2.20E+05 ± 1.03E+05	+	**1.44E+05 ± 9.75E+04**	2.87E+05 ± 1.33E+05	+	**1.48E+05 ± 7.76E+04**
F04	8.38E-01 ± 3.29E+00	+	**5.97E-01 ± 3.13E+00**	3.07E+00 ± 7.46E+00	+	**3.52E-03 ± 5.92E-03**	1.81E+01 ± 2.46E+01	+	**3.16E-02 ± 5.33E-02**
F05	**1.45E+03 ± 4.83E+02**	=	1.61E+03 ± 3.81E+02	8.46E+02 ± 3.93E+02	=	**7.89E+02 ± 3.55E+02**	9.66E+02 ± 4.60E+02	+	**6.29E+02 ± 3.57E+02**
F06	1.51E+00 ± 1.95E+00	+	**7.43E-01 ± 1.51E+00**	2.40E+01 ± 1.98E+01	+	**8.93E+00 ± 1.45E+01**	2.79E+01 ± 2.05E+01	+	**2.41E+01 ± 2.45E+01**
F07	**1.60E-02 ± 1.13E-02**	=	1.65E-02 ± 1.16E-02	**9.16E-03 ± 5.49E-03**	=	9.46E-03 ± 7.05E-03	9.36E-03 ± 5.99E-03	+	**7.69E-03 ± 6.34E-03**
F08	2.09E+01 ± 4.56E-02	=	2.09E+01 ± 4.63E-02	2.09E+01 ± 6.45E-02	=	2.09E+01 ± 5.43E-02	2.09E+01 ± 5.66E-02	=	2.09E+01 ± 5.14E-02
F09⋆	2.84E+01 ± 5.30E+00	+	**1.86E+01 ± 4.18E+00**	1.18E+02 ± 1.07E+01	+	**1.03E+02 ± 9.08E+00**	1.44E+02 ± 9.90E+00	+	**1.25E+02 ± 1.32E+01**
F10	4.06E+01 ± 9.35E+00	+	**3.76E+01 ± 8.30E+00**	7.21E+01 ± 1.09E+01	+	**6.07E+01 ± 1.01E+01**	8.34E+01 ± 1.07E+01	+	**7.16E+01 ± 1.17E+01**
F11	2.67E+01 ± 1.76E+00	+	**2.47E+01 ± 3.70E+00**	2.95E+01 ± 1.26E+00	=	**2.93E+01 ± 1.62E+00**	**2.97E+01 ± 1.28E+00**	=	3.00E+01 ± 1.21E+00
F12	2.87E+03 ± 4.01E+03	+	**1.70E+03 ± 2.15E+03**	2.21E+04 ± 8.45E+03	+	**2.23E+03 ± 3.66E+03**	3.07E+04 ± 8.80E+03	+	**5.27E+03 ± 8.11E+03**
F13	1.25E+00 ± 1.48E-01	=	**1.23E+00 ± 2.23E-01**	1.98E+00 ± 1.97E-01	+	**1.94E+00 ± 1.36E-01**	2.33E+00 ± 1.87E-01	+	**2.26E+00 ± 2.10E-01**
F14	1.28E+01 ± 2.56E-01	=	1.28E+01 ± 2.11E-01	1.31E+01 ± 2.51E-01	+	**1.30E+01 ± 2.17E-01**	1.32E+01 ± 1.54E-01	=	**1.31E+01 ± 2.02E-01**
F15	**3.47E+02 ± 9.41E+01**	=	3.51E+02 ± 8.44E+01	**3.50E+02 ± 1.04E+02**	=	3.74E+02 ± 4.87E+01	**3.09E+02 ± 1.44E+02**	–	3.70E+02 ± 5.05E+01
F16	**6.51E+01 ± 2.27E+01**	=	7.04E+01 ± 2.87E+01	8.93E+01 ± 9.05E+00	+	**7.33E+01 ± 1.17E+01**	1.06E+02 ± 1.17E+01	+	**8.90E+01 ± 1.13E+01**
F17	1.05E+02 ± 3.36E+01	+	**7.99E+01 ± 5.54E+01**	1.53E+02 ± 1.78E+01	+	**1.35E+02 ± 1.53E+01**	1.68E+02 ± 1.65E+01	+	**1.50E+02 ± 1.49E+01**
F18	9.09E+02 ± 2.47E+00	–	9.09E+02 ± 1.62E+01	9.07E+02 ± 1.53E+00	=	**9.06E+02 ± 1.77E+00**	9.07E+02 ± 1.68E+00	+	**9.06E+02 ± 1.80E+00**
F19	**9.08E+02 ± 2.42E+00**	–	9.11E+02 ± 2.96E+00	**9.06E+02 ± 1.30E+00**	=	9.07E+02 ± 1.66E+00	9.07E+02 ± 1.36E+00	+	**9.06E+02 ± 1.74E+00**
F20	**9.08E+02 ± 2.45E+00**	–	9.12E+02 ± 3.62E+00	**9.06E+02 ± 1.30E+00**	=	9.07E+02 ± 1.74E+00	9.06E+02 ± 1.44E+00	=	9.06E+02 ± 1.38E+00
F21	5.00E+02 ± 0.00E+00	=	5.00E+02 ± 0.00E+00	5.00E+02 ± 0.00E+00	=	5.00E+02 ± 0.00E+00	5.00E+02 ± 0.00E+00	=	5.00E+02 ± 0.00E+00
F22	**9.02E+02 ± 1.12E+01**	=	9.05E+02 ± 1.52E+01	9.06E+02 ± 9.63E+00	+	**8.99E+02 ± 1.07E+01**	9.10E+02 ± 7.36E+00	+	**9.00E+02 ± 9.28E+00**
F23	5.42E+02 ± 5.70E+01	+	**5.34E+02 ± 5.45E-03**	5.34E+02 ± 1.12E-04	+	5.34E+02 ± 2.20E-04	5.34E+02 ± 2.58E-06	+	5.34E+02 ± 2.23E-04
F24	2.00E+02 ± 0.00E+00	=	2.00E+02 ± 0.00E+00	2.00E+02 ± 0.00E+00	=	2.00E+02 ± 0.00E+00	2.00E+02 ± 0.00E+00	=	2.00E+02 ± 0.00E+00
F25	2.10E+02 ± 5.12E-01	=	2.10E+02 ± 6.24E-01	2.10E+02 ± 2.90E-01	+	**2.09E+02 ± 2.59E-01**	2.10E+02 ± 3.71E-01	+	**2.09E+02 ± 2.46E-01**
$w/t/l$	11/11/3		–	15/10/0		–	18/6/1		–

⋆ indicates that when several algorithms obtain the global optimum, the intermediate results are reported at $\text{NFFEs} = 20,000$.
"+", "–", and "=" indicate our approach is respectively better than, worse than, or similar to its competitor according to the Wilcoxon signed-rank test at $\alpha = 0.05$.

TABLE XIII
COMPARISON ON THE ERROR VALUES OF JADE AND RANK-JADE WITH DIFFERENT POPULATION SIZE FOR FUNCTIONS F01 - F25 AT $D = 30$.

Prob	$Np = 50$			$Np = 150$			$Np = 200$		
	JADE		rank-JADE	JADE		rank-JADE	JADE		rank-JADE
F01⋆	1.72E-10 ± 2.40E-10	+	**3.15E-11 ± 6.78E-11**	3.46E-01 ± 9.08E-02	+	**1.66E-01 ± 4.58E-02**	7.86E+00 ± 2.16E+00	+	**3.84E+00 ± 8.79E-01**
F02	5.13E-28 ± 2.10E-28	+	**4.67E-28 ± 3.72E-28**	3.04E-28 ± 1.18E-28	+	**2.15E-28 ± 7.69E-29**	2.67E-28 ± 7.80E-29	+	**1.77E-28 ± 7.33E-29**
F03	**4.52E+03 ± 3.34E+03**	=	5.21E+03 ± 4.53E+03	4.39E+03 ± 5.50E+03	=	**4.05E+03 ± 3.94E+03**	**3.33E+02 ± 8.67E+02**	–	6.06E+02 ± 1.06E+03
F04	1.15E+00 ± 7.91E+00	=	**1.11E+00 ± 7.70E+00**	3.01E-21 ± 9.25E-21	+	**1.17E-22 ± 6.91E-22**	4.62E-27 ± 1.00E-26	+	**6.29E-28 ± 4.42E-28**
F05	2.48E+02 ± 4.35E+02	=	**2.07E+02 ± 1.99E+02**	4.23E-03 ± 1.34E-02	+	**1.56E-04 ± 3.08E-04**	4.73E-03 ± 9.23E-03	+	**3.88E-04 ± 9.64E-04**
F06	**1.30E+01 ± 3.13E+01**	=	1.30E+01 ± 3.13E+01	8.93E+00 ± 2.68E+01	+	**3.46E+00 ± 1.71E+01**	4.11E+00 ± 9.32E+00	+	**6.07E-01 ± 3.76E+00**
F07	1.23E-02 ± 1.10E-02	=	**1.07E-02 ± 1.02E-02**	9.36E-03 ± 8.40E-03	+	**3.84E-03 ± 5.34E-03**	6.80E-03 ± 5.23E-03	+	**1.53E-03 ± 3.79E-03**
F08	2.09E+01 ± 2.23E-01	=	2.09E+01 ± 1.23E-01	2.09E+01 ± 4.26E-02	=	2.09E+01 ± 4.72E-02	2.09E+01 ± 5.20E-02	=	2.09E+01 ± 4.18E-02
F09⋆	2.24E+01 ± 2.58E+00	+	**2.06E+01 ± 2.43E+00**	1.31E+02 ± 7.79E+00	+	**1.26E+02 ± 1.02E+01**	1.58E+02 ± 1.15E+01	+	**1.53E+02 ± 1.06E+01**
F10	3.65E+01 ± 1.00E+01	+	**3.35E+01 ± 7.59E+00**	3.34E+01 ± 5.32E+00	+	**2.95E+01 ± 4.86E+00**	3.89E+01 ± 7.03E+00	=	**3.84E+01 ± 4.98E+00**
F11	2.68E+01 ± 1.83E+00	=	**2.65E+01 ± 1.67E+00**	2.57E+01 ± 1.32E+00	=	**2.56E+01 ± 1.50E+00**	2.66E+01 ± 1.60E+00	+	**2.58E+01 ± 1.44E+00**
F12	3.49E+03 ± 2.89E+03	+	**2.68E+03 ± 3.03E+03**	1.05E+04 ± 4.98E+03	+	**7.67E+03 ± 6.25E+03**	1.31E+04 ± 6.07E+03	+	**8.23E+03 ± 5.77E+03**
F13	**1.19E+00 ± 1.47E-01**	=	1.24E+00 ± 1.22E-01	1.86E+00 ± 1.33E-01	=	**1.85E+00 ± 1.21E-01**	**2.19E+00 ± 1.48E-01**	=	2.20E+00 ± 1.33E-01
F14	1.24E+01 ± 2.95E-01	=	**1.23E+01 ± 3.86E-01**	1.23E+01 ± 3.00E-01	=	1.23E+01 ± 3.01E-01	1.24E+01 ± 2.00E-01	=	**1.23E+01 ± 2.77E-01**
F15	3.56E+02 ± 9.72E+01	=	**3.38E+02 ± 1.09E+02**	3.69E+02 ± 7.89E+01	=	**3.58E+02 ± 8.59E+01**	3.60E+02 ± 8.30E+01	=	**3.52E+02 ± 5.80E+01**
F16	1.50E+02 ± 1.57E+02	=	**1.49E+02 ± 1.57E+02**	6.30E+01 ± 5.03E+01	+	**6.20E+01 ± 5.39E+01**	6.62E+01 ± 2.20E+01	+	**5.95E+01 ± 1.35E+01**
F17	**1.63E+02 ± 1.53E+02**	=	1.87E+02 ± 1.72E+02	1.05E+02 ± 5.39E+01	+	**9.46E+01 ± 5.00E+01**	1.11E+02 ± 4.70E+01	=	**1.09E+02 ± 5.11E+01**
F18	**8.98E+02 ± 4.01E+01**	=	9.03E+02 ± 3.09E+01	**8.92E+02 ± 4.07E+01**	=	9.06E+02 ± 1.54E+01	**9.00E+02 ± 2.98E+01**	=	9.06E+02 ± 1.54E+01
F19	9.03E+02 ± 3.50E+01	+	**8.95E+02 ± 4.19E+01**	9.02E+02 ± 2.62E+01	+	**9.01E+02 ± 2.60E+01**	**9.05E+02 ± 2.17E+01**	=	9.08E+02 ± 1.92E+00
F20	9.02E+02 ± 3.46E+01	=	**8.93E+02 ± 4.41E+01**	**9.02E+02 ± 2.62E+01**	=	9.04E+02 ± 2.15E+01	9.07E+02 ± 1.55E+01	+	**9.01E+02 ± 2.59E+01**
F21	5.00E+02 ± 0.00E+00	=	5.00E+02 ± 0.00E+00	5.00E+02 ± 0.00E+00	=	5.00E+02 ± 0.00E+00	5.00E+02 ± 0.00E+00	=	5.00E+02 ± 0.00E+00
F22	9.08E+02 ± 1.95E+01	+	**9.02E+02 ± 1.59E+01**	8.95E+02 ± 8.78E+00	=	**8.93E+02 ± 1.24E+01**	8.96E+02 ± 8.52E+00	=	**8.94E+02 ± 8.34E+00**
F23	5.86E+02 ± 1.45E+02	+	**5.50E+02 ± 7.97E+01**	2.00E+02 ± 0.00E+00	=	2.00E+02 ± 0.00E+00	5.34E+02 ± 8.45E-05	+	5.34E+02 ± 6.23E-05
F24	2.00E+02 ± 0.00E+00	=	2.00E+02 ± 0.00E+00	5.34E+02 ± 6.73E-05	+	5.34E+02 ± 1.69E-04	2.00E+02 ± 0.00E+00	=	2.00E+02 ± 0.00E+00
F25	2.34E+02 ± 1.38E+02	+	**2.10E+02 ± 4.63E-01**	2.09E+02 ± 9.55E-02	=	2.09E+02 ± 7.33E-02	2.09E+02 ± 6.90E-02	=	2.09E+02 ± 6.83E-02
$w/t/l$	9/16/0		–	13/12/0		–	12/12/1		–

⋆ indicates that when several algorithms obtain the global optimum, the intermediate results are reported at $\text{NFFEs} = 20,000$.
"+", "–", and "=" indicate our approach is respectively better than, worse than, or similar to its competitor according to the Wilcoxon signed-rank test at $\alpha = 0.05$.

in Table XI, it is clear that rank-jDE consistently provide better results than jDE, rank-jDE1, and rank-jDE2. For all cases, rank-jDE obtains better R^+ values than R^- values for all functions. In addition, at $\alpha = 0.1$, rank-jDE significantly improves the performance of jDE and rank-jDE1, respectively. Therefore, the above results confirm that our proposed ranking-based mutation operator is a reasonable method.

F. Influence of the Population Size

In our proposed ranking-based mutation operators, we use the simplest linear model to calculate the selection probabilities of different individual as shown in Equation (10). One of the advantages of this technique is that it does not add any new parameter to DE, however, it is worth mentioning that the selection probability is related to the population size. If the population size Np is small, all of the selection probabilities are increased, vice versa. In the previous experiments, for all DE variants the population size is set to as used in their corresponding literature as described in Table I. In this section, we set different Np values to evaluate the influence to our approach. To save the space, we only select jDE and JADE for illustration. For both algorithms, Np is set to $50, 150$, and 200. All other parameters are set the same as shown in Table I. The results of jDE and JADE are respectively reported in Tables XII and XIII for all functions at $D = 30$.

From Tables XII and XIII, we can see that for both jDE

TABLE XIV
COMPARISON ON THE ERROR VALUES BETWEEN ADVANCED DE AND ITS CORRESPONDING RANKING-BASED DE VARIANT FOR FUNCTIONS f_{01} - f_{13} AT $D = 100$.

Prob	jDE		rank-jDE	ODE		rank-ODE	SaDE		rank-SaDE
f_{01}	8.98E-08 ± 2.17E-08	+	**9.28E-14 ± 3.37E-14**	3.66E-06 ± 2.72E-06	+	**4.14E-09 ± 4.19E-09**	7.65E-12 ± 2.58E-12	+	**5.45E-18 ± 3.00E-18**
f_{02}	3.85E-05 ± 5.90E-06	+	**1.64E-08 ± 4.09E-09**	1.09E-01 ± 3.67E-02	+	**2.10E-02 ± 8.74E-03**	7.09E-07 ± 1.28E-07	+	**8.93E-10 ± 1.81E-10**
f_{03}	7.96E+04 ± 2.27E+04	+	**6.44E+03 ± 6.33E+03**	1.74E+04 ± 5.14E+03	+	**1.52E+04 ± 6.35E+03**	1.02E+01 ± 3.86E+00	+	**1.67E+00 ± 8.56E-01**
f_{04}	1.25E+01 ± 6.53E-01	+	**3.56E+00 ± 4.49E-01**	4.77E-10 ± 6.09E-10	+	**7.01E-11 ± 6.83E-11**	**7.81E-01 ± 2.56E-01**	–	1.03E+00 ± 3.21E-01
f_{05}	**1.00E+02 ± 1.64E+01**	=	1.05E+02 ± 2.46E+01	9.50E+01 ± 6.51E-01	+	**9.03E+01 ± 7.11E-01**	9.02E+01 ± 2.56E+00	+	**8.57E+01 ± 1.51E+01**
f_{06}	0.00E+00 ± 0.00E+00	=	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	=	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	=	0.00E+00 ± 0.00E+00
f_{07}	4.56E-02 ± 5.90E-03	+	**2.41E-02 ± 3.72E-03**	4.92E-03 ± 1.33E-03	+	**4.55E-03 ± 1.17E-03**	1.12E-02 ± 1.63E-03	+	**8.69E-03 ± 1.61E-03**
f_{08}	2.63E+03 ± 3.65E+02	+	**2.45E+02 ± 1.27E+02**	3.26E+04 ± 4.23E+02	+	**3.25E+04 ± 5.10E+02**	1.28E+04 ± 3.97E+02	+	**1.14E+04 ± 4.47E+02**
f_{09}	1.22E+02 ± 1.06E+01	+	**9.72E+01 ± 8.59E+00**	5.19E+02 ± 1.58E+02	=	**5.14E+02 ± 1.90E+02**	2.76E+02 ± 8.65E+00	+	**2.49E+02 ± 1.00E+01**
f_{10}	3.95E-05 ± 5.90E-06	+	**3.83E-08 ± 6.83E-09**	8.98E-04 ± 4.96E-04	+	**3.38E-05 ± 1.75E-05**	3.20E-07 ± 7.00E-08	+	**3.09E-10 ± 7.13E-11**
f_{11}	5.73E-08 ± 1.89E-08	+	**5.95E-14 ± 2.61E-14**	1.36E-03 ± 4.17E-03	+	**1.48E-04 ± 1.05E-03**	1.61E-11 ± 6.21E-11	+	**2.29E-19 ± 2.78E-19**
f_{12}	1.95E-08 ± 6.48E-09	+	**1.29E-14 ± 6.45E-15**	3.02E-08 ± 2.95E-08	+	**3.32E-11 ± 3.68E-11**	2.68E-14 ± 1.17E-14	+	**2.13E-20 ± 1.16E-20**
f_{13}	7.84E-05 ± 5.34E-05	+	**9.08E-12 ± 5.24E-12**	2.94E-03 ± 1.25E-02	+	**9.21E-08 ± 1.14E-07**	6.28E-11 ± 6.23E-11	+	**1.37E-17 ± 9.24E-18**
$w/t/l$	11/2/0		–	11/2/0		–	11/1/1		–

Prob	JADE		rank-JADE	CoDE		rank-CoDE	DEGL		rank-DEGL
f_{01}	2.09E-37 ± 4.19E-37	+	**2.58E-42 ± 3.21E-42**	1.23E+04 ± 1.97E+03	+	**6.23E+03 ± 1.09E+03**	1.15E+02 ± 4.66E+01	+	**6.54E+01 ± 3.26E+01**
f_{02}	5.84E-19 ± 5.41E-19	+	**2.23E-21 ± 1.75E-21**	1.23E+02 ± 9.36E+00	+	**1.08E+02 ± 9.10E+00**	4.27E+00 ± 1.16E+00	+	**2.82E+00 ± 8.42E-01**
f_{03}	5.41E-03 ± 2.60E-03	+	**2.51E-03 ± 1.24E-03**	5.26E+04 ± 9.62E+03	+	**2.80E+04 ± 7.07E+03**	6.02E-01 ± 9.03E-01	+	**4.17E-02 ± 4.01E-02**
f_{04}	2.16E+00 ± 3.45E-01	+	**1.88E+00 ± 2.62E-01**	4.65E+01 ± 3.79E+00	+	**3.99E+01 ± 3.48E+00**	1.11E+01 ± 1.67E+00	=	**1.07E+01 ± 1.39E+00**
f_{05}	7.17E+01 ± 1.52E+01	+	**6.14E+01 ± 2.34E+00**	5.25E+06 ± 1.61E+06	+	**1.75E+06 ± 5.71E+05**	1.08E+04 ± 6.90E+03	+	**5.56E+03 ± 5.08E+03**
f_{06}	0.00E+00 ± 0.00E+00	=	0.00E+00 ± 0.00E+00	1.16E+04 ± 1.63E+03	+	**6.66E+03 ± 1.09E+03**	2.23E+02 ± 8.58E+01	+	**1.36E+02 ± 5.14E+01**
f_{07}	2.78E-03 ± 5.17E-04	+	**2.46E-03 ± 5.36E-04**	6.40E+00 ± 2.08E+00	+	**2.71E+00 ± 8.68E-01**	1.01E-02 ± 1.95E-03	+	**7.77E-03 ± 1.67E-03**
f_{08}	6.30E+03 ± 2.82E+02	+	**5.97E+03 ± 2.70E+02**	2.29E+04 ± 5.23E+02	+	**2.28E+04 ± 3.27E+02**	3.31E+04 ± 4.59E+02	=	**3.29E+04 ± 5.33E+02**
f_{09}	1.01E+02 ± 4.95E+00	+	**9.78E+01 ± 4.74E+00**	8.10E+02 ± 1.58E+01	+	**7.99E+02 ± 1.61E+01**	9.92E+01 ± 9.23E+01	=	**8.30E+01 ± 1.51E+01**
f_{10}	7.55E-15 ± 0.00E+00	=	7.55E-15 ± 0.00E+00	1.19E+01 ± 4.74E-01	+	**1.02E+01 ± 5.81E-01**	3.14E+00 ± 3.03E-01	+	**2.73E+00 ± 3.20E-01**
f_{11}	2.96E-04 ± 1.46E-03	=	**5.42E-20 ± 0.00E+00**	1.11E+02 ± 1.77E+01	+	**5.70E+01 ± 9.84E+00**	2.03E+00 ± 4.19E-01	+	**1.59E+00 ± 2.93E-01**
f_{12}	4.71E-33 ± 0.00E+00	=	4.71E-33 ± 0.00E+00	4.03E+05 ± 3.76E+05	+	**1.52E+04 ± 3.06E+04**	2.36E+00 ± 7.26E-01	+	**2.09E+00 ± 7.24E-01**
f_{13}	1.27E-30 ± 8.72E-30	+	**1.42E-32 ± 3.03E-33**	5.46E+06 ± 3.12E+06	+	**8.83E+05 ± 6.54E+05**	3.08E+02 ± 8.45E+01	+	**2.43E+02 ± 8.75E+01**
$w/t/l$	9/4/0		–	13/0/0		–	10/3/0		–

"+", "–", and "=" indicate our approach is respectively better than, worse than, or similar to its competitor according to the Wilcoxon signed-rank test at $\alpha = 0.05$.

TABLE XV
COMPARISON ON THE ERROR VALUES BETWEEN ADVANCED DE AND ITS CORRESPONDING RANKING-BASED DE VARIANT FOR FUNCTIONS f_{01} - f_{13} AT $D = 200$.

Prob	jDE		rank-jDE	ODE		rank-ODE	SaDE		rank-SaDE
f_{01}	5.09E-03 ± 6.92E-04	+	**5.39E-07 ± 9.57E-08**	1.95E-01 ± 1.05E-01	+	**3.11E-03 ± 1.40E-03**	6.72E-06 ± 1.67E-06	+	**2.32E-10 ± 8.61E-11**
f_{02}	3.19E-02 ± 3.19E-03	+	**1.82E-04 ± 1.77E-05**	2.18E+00 ± 5.81E-01	+	**8.59E-01 ± 2.52E-01**	5.15E-04 ± 6.09E-05	+	**2.92E-06 ± 3.96E-07**
f_{03}	5.16E+05 ± 4.65E+04	+	**3.68E+05 ± 1.19E+05**	7.40E+04 ± 2.21E+04	=	**7.17E+04 ± 2.04E+04**	1.73E+02 ± 5.73E+01	+	**1.05E+02 ± 2.65E+01**
f_{04}	4.79E+01 ± 9.08E-01	+	**3.05E+01 ± 1.00E+00**	4.38E-08 ± 5.29E-08	+	**1.14E-08 ± 9.05E-09**	3.66E+00 ± 4.18E-01	=	**3.65E+00 ± 3.58E-01**
f_{05}	2.67E+02 ± 4.03E+01	+	**2.16E+02 ± 3.12E+01**	1.97E+02 ± 1.58E+00	+	**1.95E+02 ± 9.44E-01**	2.45E+02 ± 4.19E+01	+	**2.10E+02 ± 2.99E+01**
f_{06}	0.00E+00 ± 0.00E+00	=	0.00E+00 ± 0.00E+00	8.08E+00 ± 9.66E+00	+	**2.58E+00 ± 3.01E+00**	0.00E+00 ± 0.00E+00	=	0.00E+00 ± 0.00E+00
f_{07}	1.32E-01 ± 1.08E-02	+	**6.06E-02 ± 8.23E-03**	1.28E-02 ± 3.80E-03	+	**1.09E-02 ± 2.48E-03**	2.92E-02 ± 5.37E-03	+	**2.22E-02 ± 2.82E-03**
f_{08}	2.57E+04 ± 1.12E+03	+	**2.29E+04 ± 8.56E+02**	7.03E+04 ± 7.65E+02	=	7.03E+04 ± 6.34E+02	4.16E+04 ± 6.63E+02	+	**4.01E+04 ± 7.13E+02**
f_{09}	6.31E+02 ± 2.32E+01	+	**5.66E+02 ± 2.86E+01**	9.67E+02 ± 5.15E+02	=	**9.62E+02 ± 6.18E+02**	9.03E+02 ± 1.68E+01	+	**8.41E+02 ± 2.29E+01**
f_{10}	6.66E-03 ± 4.53E-04	+	**6.61E-05 ± 6.67E-06**	2.76E-01 ± 1.75E-01	+	**2.93E-02 ± 2.65E-02**	2.05E-01 ± 4.17E-01	+	**1.86E-06 ± 1.13E-06**
f_{11}	1.62E-03 ± 2.56E-04	+	**1.61E-07 ± 2.99E-08**	9.34E-02 ± 9.12E-02	+	**5.89E-03 ± 1.04E-02**	8.90E-04 ± 2.85E-03	+	**4.93E-04 ± 2.49E-03**
f_{12}	5.01E-03 ± 1.33E-03	+	**2.44E-07 ± 8.56E-08**	1.28E-04 ± 1.09E-04	+	**2.38E-06 ± 2.07E-06**	3.11E-04 ± 2.20E-03	+	**6.22E-06 ± 3.08E-05**
f_{13}	9.03E+00 ± 1.39E+00	+	**2.60E-04 ± 1.01E-04**	5.28E+00 ± 3.91E+00	+	**6.43E-01 ± 1.10E+00**	5.19E-04 ± 2.17E-03	+	**9.00E-06 ± 3.77E-05**
$w/t/l$	12/1/0		–	10/3/0		–	11/2/0		–

Prob	JADE		rank-JADE	CoDE		rank-CoDE	DEGL		rank-DEGL
f_{01}	1.65E-24 ± 1.32E-24	+	**4.88E-30 ± 6.49E-30**	5.99E+04 ± 6.23E+03	+	**4.81E+04 ± 7.26E+03**	1.99E+03 ± 3.98E+02	+	**1.43E+03 ± 3.63E+02**
f_{02}	1.15E-11 ± 7.76E-12	+	**5.36E-14 ± 5.60E-14**	1.22E+12 ± 6.46E+12	+	**6.16E+07 ± 3.60E+08**	3.27E+01 ± 3.96E+00	+	**2.67E+01 ± 3.65E+00**
f_{03}	9.17E+00 ± 3.06E+00	+	**7.46E+00 ± 2.16E+00**	3.26E+05 ± 4.52E+04	+	**2.58E+05 ± 3.82E+04**	8.19E+02 ± 1.75E+02	+	**4.53E+02 ± 1.44E+02**
f_{04}	4.71E+00 ± 4.06E-01	+	**4.33E+00 ± 3.76E-01**	6.30E+01 ± 3.36E+00	+	**5.81E+01 ± 4.16E+00**	1.64E+01 ± 1.44E+00	=	**1.62E+01 ± 1.51E+00**
f_{05}	1.95E+02 ± 2.79E+01	+	**1.76E+02 ± 1.59E+01**	**2.88E+07 ± 8.66E+06**	–	4.96E+07 ± 1.20E+07	2.33E+05 ± 7.79E+04	+	**1.39E+05 ± 3.98E+04**
f_{06}	0.00E+00 ± 0.00E+00	=	0.00E+00 ± 0.00E+00	6.38E+04 ± 7.41E+03	+	**4.56E+04 ± 7.32E+03**	2.25E+03 ± 4.10E+02	+	**1.71E+03 ± 3.62E+02**
f_{07}	6.98E-03 ± 1.09E-03	+	**5.21E-03 ± 8.33E-04**	1.44E+02 ± 3.29E+01	+	**8.52E+01 ± 1.73E+01**	2.45E-02 ± 4.32E-03	+	**1.88E-02 ± 2.69E-03**
f_{08}	3.31E+04 ± 6.04E+02	+	**3.27E+04 ± 6.23E+02**	5.67E+04 ± 6.58E+02	=	**5.66E+04 ± 5.71E+02**	**7.11E+04 ± 8.01E+02**	=	7.12E+04 ± 6.51E+02
f_{09}	5.61E+02 ± 1.44E+01	+	**5.57E+02 ± 1.47E+01**	2.11E+03 ± 3.31E+01	=	2.11E+03 ± 3.07E+01	2.20E+02 ± 2.46E+01	+	**2.06E+02 ± 2.73E+01**
f_{10}	1.76E-02 ± 1.24E-01	+	**1.46E-14 ± 5.02E-16**	1.54E+01 ± 4.88E-01	+	**1.44E+01 ± 6.05E-01**	5.24E+00 ± 3.72E-01	+	**4.84E+00 ± 3.92E-01**
f_{11}	4.93E-04 ± 1.99E-03	+	**3.94E-04 ± 1.95E-03**	5.40E+02 ± 5.61E+01	+	**4.33E+02 ± 6.53E+01**	1.89E+01 ± 3.58E+00	+	**1.39E+01 ± 3.26E+00**
f_{12}	1.20E-26 ± 3.76E-26	+	**1.30E-28 ± 5.36E-28**	2.23E+07 ± 1.06E+07	+	**9.01E+06 ± 6.32E+06**	6.39E+00 ± 1.16E+00	+	**5.47E+00 ± 1.00E+00**
f_{13}	7.06E-03 ± 3.47E-02	+	**7.01E-03 ± 3.41E-02**	9.45E+07 ± 3.57E+07	+	**5.14E+07 ± 2.33E+07**	2.82E+03 ± 4.55E+03	+	**1.05E+03 ± 1.22E+03**
$w/t/l$	12/1/0		–	10/2/1		–	11/2/0		–

"+", "–", and "=" indicate our approach is respectively better than, worse than, or similar to its competitor according to the Wilcoxon signed-rank test at $\alpha = 0.05$.

and JADE our proposed ranking-based mutation operators enhances their performance with different population size. For jDE, rank-jDE provides significantly better results in $11, 15$, and 18 out of 25 functions for $Np = 50, 150$, and 200, respectively. Additionally, rank-JADE respectively wins in $9, 13, 12$ functions for $Np = 50, 150, 200$ compared with JADE. Generally, when the population size is large, DE is more explorative, and our proposed ranking-based mutation operators are able to provide much better results.

G. Scalability Study

In the above experiments, all results are reported for functions F01 - F25 presented in CEC-2005 competition [31] at $D = 30$ and/or $D = 50$, since these functions are defined up to $D = 50$. In this section, we choose another test suite presented in [38] to conduct the scalability study. In [38], 23 benchmark functions are presented and the first 13 functions f_{01} - f_{13} are scalable. Thus, these 13 functions are selected for the scalability study, and the dimensions are scaled at $D = 100$

TABLE XVI
COMPARISON ON THE ERROR VALUES FOR jDE VARIANTS WITH DIFFERENT MUTATION OPERATORS FOR FUNCTIONS F01 - F25 AT $D = 30$.

Prob	jDE		jDERL		jRDDE		rank-jDE
F01*	7.37E+00 ± 3.02E+00	+	**1.68E-02 ± 2.52E-02**	–	**1.25E-02 ± 6.67E-03**	–	8.93E-02 ± 4.02E-02
F02	1.08E-05 ± 1.54E-05	+	**1.67E-11 ± 2.43E-11**	=	3.17E-04 ± 1.94E-03	+	**1.44E-11 ± 2.64E-11**
F03	1.89E+05 ± 1.04E+05	+	**1.12E+05 ± 6.12E+04**	+	4.89E+06 ± 3.01E+06	+	**8.12E+04 ± 3.87E+04**
F04	2.98E-01 ± 5.78E-01	+	**3.39E-03 ± 4.80E-03**	+	8.26E+00 ± 3.48E+01	+	**7.98E-04 ± 1.65E-03**
F05	**1.10E+03 ± 4.44E+02**	=	1.51E+03 ± 4.32E+02	+	1.32E+03 ± 6.20E+02	+	**1.11E+03 ± 5.67E+02**
F06	2.46E+01 ± 2.57E+01	+	**5.46E+00 ± 1.40E+01**	+	6.43E+00 ± 1.12E+01	+	**5.74E-01 ± 1.37E+00**
F07	1.31E-02 ± 9.30E-03	+	1.72E-02 ± 1.52E-02	+	**3.94E-03 ± 6.48E-03**	–	**9.75E-03 ± 8.92E-03**
F08	**2.09E+01 ± 4.94E-02**	=	2.10E+01 ± 4.76E-02	=	2.10E+01 ± 4.57E-02	+	**2.09E+01 ± 4.98E-02**
F09*	7.64E+01 ± 8.36E+00	+	**6.08E+01 ± 9.36E+00**	–	**5.50E+01 ± 6.93E+00**	–	6.42E+01 ± 9.08E+00
F10	5.86E+01 ± 1.05E+01	+	**4.19E+01 ± 7.98E+00**	–	**4.00E+01 ± 9.50E+00**	–	4.71E+01 ± 9.42E+00
F11	2.80E+01 ± 1.74E+00	=	**2.75E+01 ± 1.98E+00**	=	**2.78E+01 ± 1.87E+00**	=	2.79E+01 ± 2.29E+00
F12	1.16E+04 ± 8.08E+03	+	**1.94E+03 ± 2.38E+03**	=	5.29E+03 ± 5.18E+03	+	**1.65E+03 ± 1.80E+03**
F13	1.70E+00 ± 1.43E-01	+	**1.54E+00 ± 1.83E-01**	–	**1.56E+00 ± 1.71E-01**	=	1.60E+00 ± 1.26E-01
F14	1.30E+01 ± 2.00E-01	=	1.30E+01 ± 2.35E-01	=	1.30E+01 ± 2.40E-01	=	1.30E+01 ± 2.05E-01
F15	**3.40E+02 ± 1.09E+02**	=	3.69E+02 ± 8.36E+01	=	**3.33E+02 ± 6.91E+01**	–	3.66E+02 ± 5.58E+01
F16	7.56E+01 ± 8.99E+00	+	**6.16E+01 ± 1.77E+01**	=	6.63E+01 ± 1.36E+01	+	**6.12E+01 ± 9.00E+00**
F17	1.33E+02 ± 1.43E+01	+	**9.35E+01 ± 3.94E+01**	=	1.17E+02 ± 5.88E+01	+	**1.06E+02 ± 3.81E+01**
F18	**9.07E+02 ± 1.45E+00**	=	9.10E+02 ± 2.30E+00	+	**9.07E+02 ± 1.43E+00**	–	9.08E+02 ± 2.28E+00
F19	**9.06E+02 ± 1.72E+00**	–	**9.03E+02 ± 2.63E+01**	+	9.07E+02 ± 1.96E+00	–	9.08E+02 ± 1.90E+00
F20	9.06E+02 ± 1.68E+00	–	**9.03E+02 ± 2.64E+01**	+	**9.06E+02 ± 1.49E+00**	–	9.08E+02 ± 1.87E+00
F21	5.00E+02 ± 0.00E+00	=	5.00E+02 ± 0.00E+00	=	5.00E+02 ± 0.00E+00	=	5.00E+02 ± 0.00E+00
F22	9.04E+02 ± 1.03E+01	+	**9.03E+02 ± 1.18E+01**	+	8.98E+02 ± 1.15E+01	=	**8.97E+02 ± 1.16E+01**
F23	**5.34E+02 ± 2.19E-04**	=	**5.34E+02 ± 3.58E-04**	=	5.42E+02 ± 5.70E+01	=	5.34E+02 ± 1.20E-03
F24	2.00E+02 ± 0.00E+00	=	2.00E+02 ± 0.00E+00	=	2.00E+02 ± 0.00E+00	=	2.00E+02 ± 0.00E+00
F25	2.10E+02 ± 3.33E-01	+	2.10E+02 ± 3.91E-01	+	2.10E+02 ± 4.16E-01	+	**2.09E+02 ± 2.76E-01**
$w/t/l$	14/9/2		10/11/4		10/7/8		–

$\star$ indicates that when several algorithms obtain the global optimum, the intermediate results are reported at $\mathrm{NFFEs} = 20,000$.
"+", "–", and "=" indicate our approach is respectively better than, worse than, or similar to its competitor according to the Wilcoxon signed-rank test at $\alpha = 0.05$.

and $D = 200$. In the 13 functions, f_{01} - f_{07} are unimodal[4], while the rest 6 functions (f_{08} - f_{13}) are multimodal with many local optima. More details for these functions can be referred to [38]. For higher dimension problems, larger population is always required. Therefore, in this section, we set $Np = 4 \cdot D$ for all algorithms. Note that this setting may not be the optimal choice for all algorithms, however, we only evaluate the improved performance of our presented ranking-based mutation operators, but not try to obtain the best results for the problems. The Max_NFFEs are set as $D \cdot 5,000$ for all problems. All other parameters are kept unchanged as described in Table I.

The results are respectively shown in Tables XIV and XV at $D = 100$ and $D = 200$. From the results it is clear that the ranking-based DE approaches consistently get significantly better results than their corresponding non-ranking-based DE approaches in the majority of the test functions at both $D = 100$ and $D = 200$. Thus, we can expect that the proposed ranking-based mutation operators can also be effective in high-dimensional problems. We will verify our expectation in our near future work for the large-scale problems, such as presented in the special issue of Soft Computing [40].

H. Compared with Other Mutation Operators

In Section II-B, we mentioned that there are other new mutation operators, which are based on different vector selection techniques, such as DERL [24], Pro-DE [26], role differentiation based DE (referred to as RDDE in this work) [27]. In this section, we compare our proposed rank-DE with DERL and RDDE. We do not compare it with Pro-DE, since it is too time-consuming. In order to make a fair comparison, rank-DE, DERL, and RDDE are all utilized the parameter adaptation presented in jDE [32]. The three methods are referred to as rank-jDE, jDERL, and jRDDE, respectively. In addition, for all the three methods $Np = 100$ is used. All other parameters are kept the same as used in their original literature. For example, for jRDDE, $N_P = N_L = N_C = 40$. The results, averaged over 50 independent runs, are tabulated in Table XVI. The overall best and the second best results are respectively highlighted in **grey boldface** and **boldface**. In addition, the results of multiple-problem analysis based on the Wilcoxon's test are shown in Table XVII.

TABLE XVII
RESULTS OF THE MULTIPLE-PROBLEM WILCOXON'S TEST FOR jDE VARIANTS WITH DIFFERENT MUTATION OPERATORS FOR FUNCTIONS F01 - F25 AT $D = 30$.

Algorithm	R^+	R^-	p-value	at $\alpha = 0.05$	at $\alpha = 0.1$
rank-jDE vs jDE	202	74	5.22E-02	=	+
rank-jDE vs jDERL	158	118	5.60E-01	=	=
rank-jDE vs jRDDE	169	107	3.60E-01	=	=

The results in Table XVI indicate that in the majority of the test functions rank-jDE obtains significantly better results than jDERL. Compared with jDERL, rank-jDE wins in 10 functions, ties in 14 functions, but only loses in 4 functions. Compared with jRDDE, both rank-jDE and jRDDE obtain similar results. rank-jDE wins in 10 functions, ties in 7 functions, and loses in 8 functions. Among the four jDE variants, rank-jDE obtains the best results in 9 functions. According to the results of the multiple-problem analysis shown in Table XVII, we can see that rank-jDE gets higher R^+ values than R^- values, which means that rank-jDE obtains the overall better results compared with jDE, jDERL, and jRDDE.

I. Comparison in Real-world Application Problems

In the previous experiments, we have verified the effectiveness of our proposed ranking-based mutation operators in

[4]In [39], the authors pointed out that the extended rosenbrock function f_{05} are actually multimodal when $D > 3$.

TABLE XVIII
COMPARISON ON THE ERROR VALUES FOR ADVANCED DE VARIANTS FOR REAL-WORLD APPLICATION PROBLEMS.

Prob	Max_NFFEs	jDE	rank-jDE	ODE	rank-ODE	SaDE	rank-SaDE
P1	20 000	9.91E+02 ± 5.67E+02 +	**3.05E+01 ±7.26E+01**	4.32E+01 ± 2.93E+01 +	**6.39E-01 ±4.18E-01**	4.17E-01 ± 1.03E+00 +	**1.39E-02 ±9.17E-02**
	150 000	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00
P2	20 000	1.75E+01 ± 2.33E+00 +	**1.49E+01 ±2.27E+00**	1.15E+01 ± 7.23E+00 +	**5.38E+00 ±6.33E+00**	1.21E+01 ± 3.61E+00 +	**7.00E+00 ±5.09E+00**
	150 000	2.68E-01 ± 3.76E-01	**0.00E+00 ±0.00E+00**	**1.35E+00 ±3.65E+00**	2.88E+00 ± 5.05E+00	7.86E-02 ± 1.62E-01	**2.17E-04 ±1.53E-03**
P3, $D = 10$	150 000	8.84E-01 ± 9.29E-02 +	**8.25E-01 ±1.50E-01**	**7.15E-01 ±1.54E-01** +	7.29E-01 ± 1.59E-01	8.00E-01 ± 1.27E-01 +	**7.08E-01 ±1.63E-01**
P3, $D = 20$	300 000	1.84E+00 ± 1.10E-01 +	**1.81E+00 ±8.95E-02**	**1.11E+00 ±1.31E-01** –	1.19E+00 ± 1.61E-01	1.80E+00 ± 8.96E-02 =	**1.77E+00 ±1.28E-01**
P4	20 000	1.51E+02 ± 5.46E+01 +	**1.86E+01 ±1.05E+01**	3.42E+01 ± 9.26E+00 +	**5.14E+00 ±1.45E+00**	5.65E-01 ± 8.08E-01 =	**2.99E-01 ±9.38E-01**
	150 000	2.24E-07 ± 6.95E-07	**8.52E-08 ±4.97E-07**	7.08E-08 ± 3.76E-08	**1.17E-14 ±1.12E-14**	1.78E-14 ± 1.12E-13	**1.05E-14 ±7.38E-14**
P5, $D = 9$	150 000	3.91E-07 ± 3.64E-07 +	**2.20E-07 ±3.61E-07**	3.33E-07 ± 3.65E-07 +	**2.90E-07 ±3.58E-07**	3.48E-07 ± 3.65E-07 =	**3.33E-07 ±3.65E-07**
P5, $D = 25$	300 000	1.10E-03 ± 1.22E-03 +	**6.50E-04 ±8.98E-04**	1.79E-03 ± 1.29E-03 +	**7.20E-04 ±9.07E-04**	7.31E-04 ± 1.05E-03 +	**3.87E-04 ±6.49E-04**
$w/t/l$		$7/0/0$	–	$6/0/1$	–	$4/3/0$	–

Prob	Max_NFFEs	JADE	rank-JADE	CoDE	rank-CoDE	DEGL	rank-DEGL
P1	20 000	8.18E-02 ± 1.61E-01 +	**9.82E-03 ±2.15E-02**	5.45E-01 ± 6.89E-01 +	**2.98E-03 ±1.77E-02**	2.73E-01 ± 1.17E+00 +	**2.50E-02 ±9.98E-02**
	150 000	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	2.95E-02 ± 1.38E-01	**6.10E-04 ±3.20E-03**
P2	20 000	1.60E+01 ± 2.08E+00 +	**1.43E+01 ±3.51E+00**	1.34E+01 ± 2.74E+00 +	**1.20E+01 ±3.10E+00**	1.25E+01 ± 7.08E+00 +	**1.01E+01 ±7.00E+00**
	150 000	2.28E-01 ± 2.68E-01 =	**1.84E-01 ±6.98E-01**	**1.77E-04 ±1.18E-03**	4.38E-01 ± 2.17E+00	1.01E+01 ± 6.79E+00	**9.01E+00 ±6.98E+00**
P3, $D = 10$	150 000	**8.37E-01 ±8.43E-02** =	8.51E-01 ± 7.81E-02	**6.25E-01 ±1.19E-01** =	6.40E-01 ± 1.47E-01	**7.59E-01 ±2.29E-01** =	8.07E-01 ± 2.62E-01
P3, $D = 20$	300 000	1.68E+00 ± 8.85E-02 =	1.68E+00 ± 7.16E-02	1.16E+00 ± 1.45E-01 =	1.16E+00 ± 1.69E-01	2.41E+00 ± 1.29E-01 =	**2.40E+00 ±8.03E-02**
P4	20 000	2.21E+00 ± 3.01E+00 +	**1.48E+00 ±2.15E+00**	2.53E+00 ± 2.14E+00 =	1.33E+00 ± 8.78E+00	3.45E-01 ± 6.37E-01 +	**2.28E-01 ±6.38E-01**
	150 000	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	6.86E-14 ± 3.69E-13	**7.15E-15 ±4.96E-14**	1.05E-03 ± 4.68E-03	**4.11E-05 ±1.96E-04**
P5, $D = 9$	150 000	8.89E-06 ± 4.25E-05 +	**3.19E-07 ±3.63E-07**	**3.48E-07 ±3.65E-07** =	3.77E-07 ± 3.65E-07	4.21E-04 ± 2.96E-03 +	**4.19E-04 ±2.96E-03**
P5, $D = 25$	300 000	3.70E-03 ± 2.25E-03 +	**9.62E-04 ±1.62E-03**	8.75E-04 ± 1.14E-03 +	**4.81E-04 ±7.19E-04**	7.12E-04 ± 8.97E-04 +	**5.11E-04 ±1.62E-04**
$w/t/l$		$5/2/0$	–	$3/4/0$	–	$5/2/0$	–

"+", "–", and "=" indicate our approach is respectively better than, worse than, or similar to its competitor according to the Wilcoxon signed-rank test at $\alpha = 0.05$.

different DE variants via benchmark functions. In this section, we evaluate the potential of our approach for real-world application problems. Five real-world problems are chosen from the literature: P1) Chebychev polynomial fitting problem ($D = 9$) [4]; P2) frequency modulation sound parameter identification ($D = 6$) [41]; P3) spread spectrum radar polyphase code design problem ($D = 10$ and $D = 20$) [15]; P4) systems of linear equations problem ($D = 10$) [41]; and P5) optimization of geophysical potential field data inversion ($D = 9$ and $D = 25$) [42]. For all the DE variants the parameter settings are used as shown in Table I. The maximal NFFEs for all problems are tabulated in the second column of Table XVIII. The results, which are averaged over 50 independent runs, are shown in Table XVIII. The better results are highlighted in **boldface** compared between the ranking-based DE and its corresponding non-ranking-based DE. Similar to the methods used in [25], the *intermediate* results are also reported for the problems where several algorithms can obtain the global optimum of these problem. In these cases, the Wilcoxon signed-rank test is only compared with the intermediate results.

From the results shown in Table XVIII, we can see that the ranking-based DE is capable of obtaining significantly better results in the majority of the test cases compared with its corresponding non-ranking-based DE. Only in 1 case (P3 at $D = 20$), ODE outperforms rank-ODE significantly. In the rest 41 cases, ranking-based DEs provide significantly better, or competitive, results compared with non-ranking-based DEs. Therefore, the results in Table XVIII indicate that the ranking-based mutation operators can be an effective alternative for the real-world problems, due to their simplicity and effectiveness.

J. Discussions

Inspired by the natural phenomenon, in this work, we present the ranking-based mutation operators for the differential evolution algorithm. In the ranking-based mutation operators good solutions will obtain higher selection probabilities, and hence, they have more chance to propagate the offspring. In this way, the exploitation ability of DE can be enhanced. Experiments have been extensively conducted on 25 benchmark functions and five real-world application problems. From the experimental results and analysis, we can draw the following summaries.

- When the DE operators have good exploration ability, our proposed ranking-based mutation operators are very efficient. They are capable of balancing the exploration and exploitation abilities for the DE algorithm. It can be observed from the results where the explorative operator is used in DE, such as jDE with "DE/rand/1", jDE with "DE/rand/2", jDE with "DE/rand-to-best/2", DEGL, etc.
- On the other hand, when the DE operators utilize the best-so-far solution ($\mathbf{x}_{best}$) and only have one difference vector, they are more exploitative. In this situation, the ranking-based mutation operators may be over-exploitative and lead to premature convergence to the local optima in the multimodal problems. However, since the exploitative operators in DE (such as "DE/best/1", "DE/current-to-best/1") are more suitable to unimodal problems, our proposed ranking-based mutation operators are also useful when solving unimodal problems [43] (see for example the results of "DE/current-to-best/1" in Table II).
- In order to calculate the selection probabilities for the individuals, different models can be used. In this work, the simplest linear model is selected as the illustration, and two other models (*i.e.*, quadratic model and sinusoidal modal) are also compared in Section IV-D. The results show that rank-jDE with the three models improve the performance the original jDE algorithm. We believe that other different models can also be used in the ranking-based mutation operators, we will verify it in our near future work.
- Generally, the ranking-based mutation operators are very simple and generic. They can be easily incorporated into most of DE variants and improve their performance.

V. CONCLUSIONS AND FUTURE WORK

In the nature, good species always contain more useful information, and hence, they are more likely to be selected to propagate offspring. Inspired by this common phenomenon, in this paper, we propose simple, yet effective ranking-based mutation operators for the DE algorithm. The simplest linear model is selected as the illustration to assign the probabilities for the individuals in the population according to their rankings, which are measured by the fitness of individual. In the ranking-based mutation operators, the base vector and the terminal point are proportionally chosen based on the selection probability. The proposed ranking-based mutation operators do not add any new parameters and also do not significantly increase the overall complexity of DE any more.

Experiments have been conducted through the benchmark functions and five real-world problems. Through evaluating the effectiveness of our approach with different mutation operators, advanced DE variants, probability calculation models, vector selection methods, population size, scalability study, and other mutation operators based on different vector selection techniques, the results confirm that our presented ranking-based mutation operators are able to enhance the exploitation ability and improve the performance of different DE variants.

Stochastic ranking presented in [44] has been proven to be an efficient constraint-handling technique, another future direction is integrating the stochastic ranking into the DE mutation operators for constrained optimization problems. Large-scale continuous optimization gets more attention recently, some DE variants obtained very promising results, see for example [45], [46], [47], [48]. Thus, in our near future work, we will combine the ranking-based mutation operators with the above-mentioned DEs for the large-scale problems.

The source code of our proposed rank-jDE can be obtained from the first author upon request.

ACKNOWLEDGMENTS

The authors are very grateful to the Associate Editor and the anonymous reviewers for their constructive comments to this paper. The authors would like to thank Dr. Y. Wang to make the source code of CoDE available online. They are also grateful to Dr. J. Brest and Dr. P. N. Suganthan for providing the source code of jDE and SaDE, respectively. The first author would like to thank Dr. N. Qiu for providing the code for the problem of geophysical potential field data optimized inversion.

REFERENCES

[1] T. Bäck, *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford, UK: Oxford University Press, 1996.

[2] R. Storn and K. Price, "Differential evolution–A simple and efficient adaptive scheme for global optimization over continuous spaces," Berkeley, CA, Tech. Rep., 1995, tech. Rep. TR-95-012.

[3] ——, "Differential evolution–A simple and efficient heuristic for global optimization over continuous spaces," *J. of Global Optim.*, vol. 11, no. 4, pp. 341–359, Dec 1997.

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

[5] S. Das, A. Abraham, and A. Konar, "Automatic clustering using an improved differential evolution algorithm," *IEEE Trans. on Syst. Man, Cybern. A, Syst., Humans*, vol. 38, no. 1, pp. 218–237, Feb 2008.

[6] F. Neri and V. Tirronen, "Recent advances in differential evolution: A survey and experimental analysis," *Artificial Intelligence Review*, vol. 33, no. 1-2, pp. 61 – 106, 2010.

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

[8] N. Noman and H. Iba, "Accelerating differential evolution using an adaptive local search," *IEEE Trans. on Evol. Comput.*, vol. 12, no. 1, pp. 107–125, Feb 2008.

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

[10] J. Sun, Q. Zhang, and E. P. K. Tsang, "DE/EDA: A new evolutionary algorithm for global optimization," *Information Sciences*, vol. 169, no. 3-4, pp. 249–262, 2005.

[11] P. Kaelo and M. M. Ali, "Differential evolution algorithms using hybrid mutation," *Comput. Optim. Appl.*, vol. 37, no. 2, pp. 231–246, 2007.

[12] Z. Yang, K. Tang, and X. Yao, "Self-adaptive differential evolution with neighborhood search," in *Proc. IEEE Congress on Evol. Comput.*, 2008, pp. 1110–1116.

[13] Z. Cai, W. Gong, C. X. Ling, and H. Zhang, "A clustering-based differential evolution for global optimization," *Applied Soft Computing*, vol. 11, no. 1, pp. 1363 – 1379, January 2011.

[14] R. Storn and K. Price, "Home page of differential evolution," 2010. [Online]. Available: http://www.ICSI.Berkeley.edu/~storn/code.html

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

[16] A. K. Qin and P. N. Suganthan, "Self-adaptive differential evolution algorithm for numerical optimization," in *IEEE Congr. on Evol. Comput.*, 2005, pp. 1785–1791.

[17] E. Mezura-Montes, J. Velázquez-Reyes, and C. A. C. Coello, "A comparative study of differential evolution variants for global optimization," in *Proc. Genetic Evol. Comput. Conf.*, M. Cattolico, Ed., July 8-12, 2006, pp. 485–492.

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

[19] A. W. Iorio and X. Li, "Solving rotated multi-objective optimization problems using differential evolution," in *AI 2004: Advances in Artificial Intelligence, Proceedings*, 2004, pp. 861–872.

[20] R. Mallipeddi, P. Suganthan, Q. Pan, and M. Tasgetiren, "Differential evolution algorithm with ensemble of parameters and mutation strategies," *Applied Soft Computing*, vol. 11, no. 2, pp. 1679–1696, 2011.

[21] W. Gong, Z. Cai, C. X. Ling, and H. Li, "Enhanced differential evolution with adaptive strategies for numerical optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Part B – Cybernetics*, vol. 41, no. 2, pp. 397–413, 2011.

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

[23] A. P. Piotrowski, J. J. Napiorkowski, and A. Kiczko, "Differential evolution algorithm with separated groups for multi-dimensional optimization problems," *European Journal of Operational Research*, vol. 216, no. 1, pp. 33 – 46, 2012.

[24] P. Kaelo and M. Ali, "A numerical study of some modified differential evolution algorithms," *European Journal of Operational Research*, vol. 169, no. 3, pp. 1176–1184, March 2006.

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

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

[27] C. García-Martínez, F. Rodríguez, and M. Lozano, "Role differentiation and malleable mating for differential evolution: an analysis on large-scale optimisation," *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, vol. 15, no. 11, pp. 2109–2126, 2011.

[28] D. Simon, "Biogeography-based optimization," *IEEE Trans. on Evol. Comput.*, vol. 12, no. 6, pp. 702–713, Dec 2008.

[29] H. Ma, "An analysis of the equilibrium of migration models for biogeography-based optimization," *Information Sciences*, vol. 180, no. 8, pp. 3444–3464, Sept. 2010.

[30] J. Zhang and A. C. Sanderson, *Adaptive Differential Evolution: A Robust Approach to Multimodal Problem Optimization*. Berlin: Springer-Verlag, 2009.

[31] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y.-P. Chen, A. Auger, and S. Tiwari, "Problem definitions and evaluation criteria

for the CEC_2005 special session on real-parameter optimization," 2005. [Online]. Available: http://www.ntu.edu.sg/home/EPNSugan

[32] J. Brest, S. Greiner, B. Bošković, M. Mernik, and V. Žumer, "Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems," *IEEE Trans. on Evol. Comput.*, vol. 10, no. 6, pp. 646–657, Dec 2006.

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

[34] J. Arabas, A. Szczepankiewicz, and T. Wroniak, "Experimental comparison of methods to handle boundary constraints in differential evolution," in *Parallel Problem Solving from Nature C PPSN XI*, ser. Lecture Notes in Computer Science, R. Schaefer, C. Cotta, J. Kolodziej, and G. Rudolph, Eds., 2010, vol. 6239, pp. 411 – 420.

[35] S. García, A. Fernandez, 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.

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

[37] J. Alcalá-Fdez, L. Sánchez, and S. García, "KEEL: A software tool to assess evolutionary algorithms to data mining problems," 2012. [Online]. Available: http://www.keel.es/

[38] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Trans. on Evol. Comput.*, vol. 3, no. 2, pp. 82–102, Jul 1999.

[39] Y.-W. Shang and Y.-H. Qiu, "A note on the extended rosenbrock function," *Evol. Comput.*, vol. 14, no. 1, pp. 119–126, 2006.

[40] F. Herrera, M. Lozano, and D.Molina, "Special issue on the scalability of evolutionary algorithms and other metaheuristics for large scale continuous optimization problems," 2010. [Online]. Available: http://sci2s.ugr.es/eamhco/CFP.php

[41] C. García-Martínez, M. Lozano, F. Herrera, D. Molina, and A. M. Sánchez, "Global and local real-coded genetic algorithms based on parent-centric crossover operators," *European Journal of Operational Research*, vol. 185, no. 3, pp. 1088–1113, 2008.

[42] N. Qiu, Q.-S. Liu, Q.-Y. Gao, and Q.-L. Zeng, "Combining genetic algorithm and generalized least squares for geophysical potential field data optimized inversion," *IEEE Geoscience and Remote Sensing Letters*, vol. 7, no. 4, pp. 660 – 664, Oct. 2010.

[43] N. Padhye, P. Bhardawaj, and K. Deb, "Improving differential evolution through a unified approach," *Journal of Global Optimization*, pp. 1–29, 2012, in press.

[44] T. Runarsson and X. Yao, "Stochastic ranking for constrained evolutionary optimization," *IEEE Trans. on Evol. Comput.*, vol. 4, no. 3, pp. 284 – 294, 2000.

[45] J. Brest and M. S. Maučec, "Self-adaptive differential evolution algorithm using population size reduction and three strategies," *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, vol. 15, no. 11, pp. 2157–2174, 2011.

[46] H. Wang, Z. Wu, and S. Rahnamayan, "Enhanced opposition-based differential evolution for solving high-dimensional continuous optimization problems," *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, vol. 15, no. 11, pp. 2127–2140, 2011.

[47] Z. Yang, K. Tang, and X. Yao, "Scalability of generalized adaptive differential evolution for large-scale continuous optimization," *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, vol. 15, no. 11, pp. 2141–2155, 2011.

[48] S.-Z. Zhao, P. Suganthan, and S. Das, "Self-adaptive differential evolution with multi-trajectory search for large-scale optimization," *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, vol. 15, no. 11, pp. 2175–2185, 2011.

Wenyin Gong received the B.Eng., M.Eng., and PhD degrees in computer science from China University of Geosciences, Wuhan, China, in 2004, 2007, and 2010, respectively. He has published over 30 research papers in journals and international conferences, including IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: PART B - CYBERNETICS, *Information Sciences*, *European Journal of Operational Research*, *Soft Computing*, GECCO, CEC, EMO, etc. He served as a referee for over 10 international journals, such as IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: PART B - CYBERNETICS, *ACM Transactions on Intelligent Systems and Technology*, *Information Sciences*, *European Journal of Operational Research*, *Applied Soft Computing*, and so on. He is interested in differential evolution, memetic algorithms, multiobjective optimization, and their applications.

Zhihua Cai received the Bsc degree from Wuhan University, China, in 1986, the Msc degree from Beijing University of Technology, China, in 1992, and the PhD degree from China University of Geosciences, in 2003. He is currently a faculty member at School of Computer Science, China University of Geosciences, Wuhan, China. He has published over 50 research papers in journals and international conferences, such as IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: PART B - CYBERNETICS, *Applied Soft Computing*, *Information Sciences*, *Knowledge-Based Systems*, *Knowledge and Information Systems*, etc. His main research areas include data mining, machine learning, evolutionary computation, and their applications.