Computing and Informatics, Vol. 37, 2018, 49–75, doi: 10.4149/cai_2018_1_49

RANKING-BASED DIFFERENTIAL EVOLUTION FOR LARGE-SCALE CONTINUOUS OPTIMIZATION

Li Guo

School of Economics and Management
China University of Geosciences
Wuhan, 430074, China
e-mail: guoli_cn@163.com

Xiang Li, Wenyin Gong

School of Computer Science
China University of Geosciences
Wuhan, 430074, China
e-mail: {lixiang, wygong}@cug.edu.cn

Abstract. Large-scale continuous optimization has gained considerable attention in recent years. Differential evolution (DE) is a simple yet efficient global numerical optimization algorithm, which has been successfully used in diverse fields. Generally, the vectors in the DE mutation operators are chosen randomly from the population. In this paper, we employ the ranking-based mutation operators for the DE algorithm to improve DE's performance. In the ranking-based mutation operators, the vectors are selected according to their rankings in the current population. The ranking-based mutation operators are general, and they are integrated into the original DE algorithm, GODE, and GaDE to verify the enhanced performance. Experiments have been conducted on the large-scale continuous optimization problems. The results indicate that the ranking-based mutation operators are able to enhance the overall performance of DE, GODE, and GaDE in the large-scale continuous optimization problems.

Keywords: Differential evolution, ranking-based mutation, vector selection, large-scale continuous optimization

1 INTRODUCTION

During the last few decades, evolutionary algorithms and metaheuristics have been successfully used for the optimization problems. However, they are mainly applied for the low- or moderate-dimensional problems. Since there are many real-world problems (such as neural network training, bio-computing, etc.) that have large problem size, in recent years, large-scale continuous optimization has gained more attention [17, 31, 37, 18, 16].

Differential evolution (DE), which was proposed by Storn and Price in 1995 [27, 28], is a simple and powerful evolutionary algorithm for global optimization. Due to its simplicity, robustness, ease of use, and efficiency, DE has obtained many successful applications in diverse fields, such as data mining, engineering design, geophysical inversion, and so on [22, 14]. More details on the state-of-the-art research within DE can be found in two surveys [20] and [5] and the references therein.

In the original DE algorithm, the core operator is the *differential* mutation, and generally, the parents in the mutation are always randomly chosen 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 $r_1 \neq r_2 \neq r_3 \neq i$. Since the parent vectors in the mutation are selected randomly, it may lead to DE be good at exploring the search space and locating the region of global minimum, but be slow at exploitation of the solutions [21]. Based on this motivation, in this paper, we modify our previous proposed ranking-based mutation operators [11] to enhance the exploitation ability of DE and employ it for the large-scale continuous optimization problems.

In the proposed ranking-based mutation operators, each parent vector has a selection probability, which is calculated according to its ranking in the population. Then, the parent vectors in the mutation are proportionally selected based on the selection probabilities. The major advantage of our proposed ranking-based mutation operators is that they are very simple and do not introduce any new parameters at all. In addition, the ranking-based mutation operators are general, they can be easily incorporated into most of existing DE variants. In this paper, they are integrated into the original DE algorithm, GODE [32], and GaDE [36] to verify the enhanced performance. Experiments have been conducted on the large-scale continuous optimization problems. The results indicate that the ranking-based mutation operators are able to enhance the overall performance of DE, GODE, and GaDE in the large-scale continuous optimization problems.

The rest of this paper is organized as follows. In Section 2, we briefly introduce the related work, including the DE algorithm and large-scale optimization in DE. Section 3 describes the ranking-based mutation operators for the DE algorithm in detail. The experimental results and analysis are shown in Section 4. Finally, in Section 5, we draw the conclusions from this work. In addition, the detailed experimental results of rank-DE, rank-GODE, and rank-GaDE are described in Appendix A.

2 RELATED WORK

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, $\mathbf{x} = [x_1, x_2, \ldots, x_D]^T$, 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:

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

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

2.1 Differential Evolution

Similar to other evolutionary algorithms, differential evolution, which is mainly used for the numerical optimization problems, is a population-based optimization algorithm. The population consists of NP vectors. Each vector $\mathbf{x}_i$, $i = 1, \ldots, NP$ is initialized within the boundary. There are three operators in the DE algorithms, i.e. differential mutation, crossover, and selection. DE creates new candidate solutions through the differential mutation and crossover operations. The selection is applied between the target solution and its corresponding trial solution, and 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 $\text{rndreal}[0, 1)$ gives a uniformly distributed random real number in $[0, 1)$. $\langle \cdot \rangle_D$ is the modulo operation with divisor D. In Algorithm 1, the "DE/rand/1/exp" is illustrated, since the exponential crossover obtains very promising results in large-scale optimization. The binomial crossover and other mutation operators can be found in [22]. 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).

2.2 Large-Scale Optimization in DE

Many real-world problems can be formulated as numerical optimization problems, and many of them are large-scale, such as bio-computing, data mining, neural network training, etc. [18]. Due to the importance of the large-scale optimization, using the evolutionary algorithms and metaheuristics for the large-scale continuous optimization problems has gained considerable attention in recent years, such as the special sessions in conference [30, 29] and special issue in journal [12].

Algorithm 1 The DE algorithm with "DE/rand/1/exp"

Generate the initial population randomly
Evaluate the fitness for each individual in the population
while the stop criterion is not satisfied **do**
    for $i = 1$ to NP **do**
        Select uniform randomly $r_1 \neq r_2 \neq r_3 \neq i$
        $\mathbf{v}_i = \mathbf{x}_{r_1} + F \cdot (\mathbf{x}_{r_2} - \mathbf{x}_{r_3})$
        $\mathbf{u}_i = \mathbf{x}_i$
        $j_{rand} = \text{rndint}(1, D)$
        $u_{i,j_{rand}} = v_{i,j_{rand}}$
        $L = 0$
        while $\text{rndreal}[0, 1) < CR$ **and** $L < D$ **do**
            $j_{rand} = \langle j_{rand} + 1 \rangle_D$
            $L = L + 1$
            $u_{i,j_{rand}} = v_{i,j_{rand}}$
        end while
    end for
    for $i = 1$ to NP **do**
        Evaluate the offspring $\mathbf{u}_i$
        if $f(\mathbf{u}_i)$ is better than **or** equal to $f(\mathbf{x}_i)$ **then**
            Replace $\mathbf{x}_i$ with $\mathbf{u}_i$
        end if
    end for
end while

Since DE has obtained very promising performance in the numerical optimization [5], many researchers employed it for the large-scale continuous optimization recently. Yang et al. [34] presented two DE algorithms based on the cooperative coevolution framework for large-scale optimization problems. Later on, in order to handle the high-dimensional nonseparable problems, they extended their work in [34] and proposed a new cooperative coevolution framework [35], where the random grouping scheme and adaptive weighting are introduced. In [19], Muelas et al. proposed a hybrid memetic algorithm based on DE for large-scale optimization problems. Brest et al. [4] presented a self-adaptive DE, jDElsgo, on large-scale optimization. In [32], the authors presented a neighborhood search based sequential DE for the CEC2010 Special Session on Large Scale Global Optimization. Stanarevic [26] hybridized the artificial bee colony with DE for the large scale optimization problems.

Recently, in the special issue of Soft Computing on the large-scale continuous optimization, there are seven papers related to the DE algorithm [18]. Brest and Maučec proposed jDElscop [3], where parameter self-adaptation, three strategies, and a population size reduction mechanism are combined. In [10], García-Martínez et al. proposed the role differentiation mechanism and malleable mating for DE. The role differentiation mechanism differentiates the DE population into four groups, i.e., *receiving*, *placing*, *leading*, and *correcting* groups. The malleable mating ensures some similarity relations between chosen vectors. LaTorre et al. [15] proposed a memetic algorithm that combines the explorative and exploitative strength of differential evolution and MTS-LS1. In addition, the multiple offspring sampling framework has also been used in the hybrid memetic algorithm. Wang et al. [32] presented an improved DE algorithm based on generalized opposition-based learning (GOBL) for high dimensional optimization problems, where the opposition-based population initialization and generation jumping are applied with GOBL. SOUPDE, proposed by Weber et al. [33], is a shuffle or update parallel DE, where a structured population algorithm characterized by sub-populations is employed. Based on the analysis of the similarities and pitfalls of existing parameter adaptation techniques in DE, Yang et al. [36] proposed a generalized parameter adaptation method in DE for large-scale optimization problems. In [39], the authors presented the SaDE-MMTS algorithm to solve large-scale continuous optimization problems. In SaDE-MMTS, the strategy adaptation along with control parameter values presented in SaDE [23], the JADE mutation strategy [38], and the modified multi-trajectory search (MMTS) algorithm are hybridized.

3 RANKING-BASED DE

In this work, we modified our previous proposed ranking-based mutation operators in [11] and combine them with the original DE algorithm, GODE [32], and GaDE [36] to improve their performance on the large-scale continuous optimization problems.

3.1 Ranking-Based Mutation

3.1.1 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 ascendent 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:

$$R_i = NP - i, \quad i = 1, 2, \ldots, NP \tag{3}$$

where NP is the population size. According to Equation (3), the best vector in the current population will obtain the highest ranking.

3.1.2 Selection Probability

After assigning the ranking for each vector, the selection probability p_i of the i^{th} vector $\mathbf{x}_i$ is calculated based on the quadratic model as follows:

$$p_i = \left(\frac{R_i}{NP}\right)^2. \tag{4}$$

Different models can be used to calculate the selection probabilities, and they may lead to different selection pressure on the better solutions. Note that, in this work, the quadratic model is used, since it is able to provide better results than the linear and sinusoidal models. Interested readers can refer to our recent paper in [11].

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

Input: The target vector index i
Output: The selected vector indexes r_1, r_2, r_3
Randomly select $r_1 \in [1, NP]$
while rndreal$[0,1) > p_{r_1}$ **or** $r_1 == i$ **do**
    Randomly select $r_1 \in [1, NP]$
end while
Randomly select $r_2 \in [1, NP]$
while rndreal$[0,1) > p_{r_2}$ **or** $r_2 == r_1$ **or** $r_2 == i$ **do**
    Randomly select $r_2 \in [1, NP]$
end while
Randomly select $r_3 \in [1, NP]$
while rndreal$[0,1) <= p_{r_3}$ **or** $r_3 == r_2$ **or** $r_3 == r_1$ **or** $r_3 == i$ **do**
    Randomly select $r_3 \in [1, NP]$
end while

3.1.3 Ranking-Based Vector Selection

Inspired by the role differentiation mechanism proposed in [10], in our proposed ranking-based mutation operators, the vectors are selected based on their rankings and roles. Also, the vector can be classified into four different roles (i.e. *placing*, *leading*, *correcting*, and *receiving* vectors) as proposed in [10]. Solutions in the population with higher selection probabilities are more likely to be chosen as the placing and leading vectors, while poor solutions are more likely to be selected as the correcting vectors in the DE mutation. As an illustration, the ranking-based vector selection for the "DE/rand/1" mutation is shown in Algorithm 2. From Algorithm 2, different from the vector selection in the original DE algorithm, in the ranking-based vector selection the selection probabilities, which are calculated based on the rankings, are used to control the selection of different vectors. For example, in ranking-based "DE/rand/1" mutation, the placing vector $\mathbf{x}_{r_1}$ and the leading vector $\mathbf{x}_{r_2}$ try to select good solutions, but the correcting vector $\mathbf{x}_{r_3}$ proportionally chooses the poor solution. Different from the vector selection presented in [10], our approach does not introduce any new parameters, while in [10] there are three new parameters, i.e. N^P, N^L, and N^C. In addition, in our ranking-based vector selection there are no explicit groups to differentiate the vectors in the population.

It is worth pointing out that Algorithm 2 is only an illustration for the "DE/rand/1" mutation, our proposed ranking-based vector selection is simple and general. It is also applicable to other mutation operators. Compared with our previous work in [11], the major difference is that in this work the correcting vector is also selected according to its ranking, while in [11] it is only selected randomly.

Algorithm 3 rank-DE: ranking-based differential evolution

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 $\Leftarrow$
5:   Calculate the selection probability for each individual according to Equation (4) $\Leftarrow$
6:   **for** $i = 1$ to NP **do**
7:     Select r_1, r_2, r_3 as shown in Algorithm 2 $\Leftarrow$
8:     Generate the trial vector $\mathbf{u}_i$ with ranking-based "DE/rand/1/exp" strategy
9:   **end for**
10:   **for** $i = 1$ to NP **do**
11:     Evaluate the offspring $\mathbf{u}_i$
12:     **if** $f(\mathbf{u}_i)$ is better than **or** equal to $f(\mathbf{x}_i)$ **then**
13:       Replace $\mathbf{x}_i$ with $\mathbf{u}_i$
14:     **end if**
15:   **end for**
16: **end while**

3.2 DE with Ranking-Based Mutation

By combing the ranking-based mutation operators, we propose the ranking-based DE algorithm, referred to as rank-DE. The pseudo-code of rank-DE is shown in Algorithm 3. The differences between Algorithm 1 and Algorithm 3 are highlighted in "$\Leftarrow$". Note that in line 8 of Algorithm 3 other ranking-based DE strategies can also be used to generate trial vector $\mathbf{u}_i$. Compared with the original DE algorithm, Algorithm 3 indicates that our proposed ranking-based DE algorithm is very simple, it does not increase the overall complexity of DE. Additionally, rank-DE enhances the exploitation ability of DE due to its ranking-based mutation operator.

4 EXPERIMENTAL RESULTS AND ANALYSIS

In order to evaluate the performance of our proposed ranking-based DE for large-scale optimization problems, we employ the test suite presented for the special issue of Soft Computing on scalability of evolutionary algorithms and other metaheuristics for large-scale continuous optimization problems [12]. The test suite contains 19 test functions, which can be categorized into four groups:

- Shifted unimodal functions: F1–F2;
- Shifted multimodal functions: F3–F6;
- Other shifted unimodal functions: F7–F11;
- Hybrid composite functions: F12–F19.

All of these functions are tested at $D = 50$, 100, 200, 500, and 1 000. More details of these functions can be found in [13].

Algorithm	Parameter Settings
DE, rank-DE	$NP = 60,\ CR = 0.9,\ F = 0.5$
GODE, rank-GODE	$NP = 60,\ CR = 0.9,\ F = 0.5$
GaDE, rank-GaDE	$NP = 60,\ p = 0.2,\ c = 0.1$ $F_m = 0.5,\ CR_m = 0.9$

Table 1. Parameter settings for all compared DE variants

4.1 Parameter Settings

As mentioned above, our proposed ranking-based mutation operators are general, they can be used in different DE variants. In this work, the ranking-based mutation operators are integrated into the original DE algorithm, GODE [32], and GaDE [36], and they are respectively named as rank-DE, rank-GODE, and rank-GaDE. In order to make a fair comparison between rank-DE and its corresponding non-rank DE, we adopt the same parameter settings as used in their original literature. The parameter settings for all compared algorithms are shown in Table 1. The maximal number of fitness function evaluations (Max_NFFEs) are set to $5\,000 \times D$ as suggested in [12]. All algorithms are performed over 25 independent runs. In addition, in both rank-DE and rank-GODE the "DE/rand/1/exp" strategy is used as adopted in DE and GODE. In rank-GaDE, the same strategies are also employed as originally used in GaDE.

F	DE	rank-DE	GODE	rank-GODE	GaDE	rank-GaDE
F1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F2	3.60E−01	**8.15E−04**	2.57E−01	**1.33E−03**	1.46E+01	**2.69E+00**
F3	2.89E+01	**1.59E−01**	3.06E+01	**1.87E−09**	1.18E+01	**3.24E−12**
F4	3.98E−02	3.98E−02	**1.05E−13**	3.98E−02	0.00E+00	0.00E+00
F5	**0.00E+00**	9.85E−04	0.00E+00	0.00E+00	**0.00E+00**	8.88E−04
F6	1.43E−13	**0.00E+00**	1.24E−14	**0.00E+00**	0.00E+00	0.00E+00
F7	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F8	3.44E+00	**3.45E−03**	1.67E−01	**4.42E−08**	1.08E−08	**0.00E+00**
F9	2.73E+02	**9.91E−09**	7.77E−06	**4.39E−10**	6.24E−07	**0.00E+00**
F10	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F11	6.23E−05	**1.05E−08**	6.44E−06	**6.93E−10**	1.31E−06	**0.00E+00**
F12	5.35E−13	**0.00E+00**	1.33E−13	**0.00E+00**	0.00E+00	0.00E+00
F13	2.45E+01	**4.98E−02**	2.55E+01	**5.05E−02**	1.19E+01	**6.24E−01**
F14	4.16E−08	**3.35E−14**	6.24E−09	**5.79E−13**	9.78E−13	**0.00E+00**
F15	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F16	1.56E−09	**0.00E+00**	1.57E−10	**5.35E−14**	4.78E−12	**0.00E+00**
F17	7.98E−01	**2.21E−01**	1.17E+00	**3.96E−02**	4.97E−01	**2.49E−01**
F18	1.22E−04	**1.18E−10**	2.97E−07	**6.30E−10**	4.82E−08	**2.40E−10**
F19	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00

All the results below 1.00E−14 have been approximated to 0.

Table 2. Comparison of the mean error values between DEs and their corresponding rank-DEs for functions F1–F19 at $D = 50$

4.2 Influence of Ranking-Based Mutation

In this section, we evaluate the influence of ranking-based mutation operators to DE, GODE, and GaDE. The ranking-based DE is compared with its corresponding non-ranking-based DE, i.e., rank-DE vs. DE, rank-GODE vs. GODE, and rank-GaDE vs. GaDE. The results for all functions at $D = 50$, 100, 200, 500, and 1 000 are reported in Tables 2–6, respectively. Note that the results of DE, GODE, and GaDE are obtained from `http://sci2s.ugr.es/eamhco/SOCO-results.xls`. In Tables 2–6, the better results are highlighted in **boldface** compared between rank-DEs and their corresponding non-rank-DEs. In addition, as stated in [8, 9], the multiple-problem statistical analysis is also important to check the behavior of the stochastic algorithms. Therefore, in order to further prove statistical significance of the results, we also use the Wilcoxon's test to compare rank-DEs with their corresponding non-rank-DEs. The Wilcoxon's test is a non-parametric statistical hypothesis test, which can be used as an alternative to the paired t-test when the results cannot be assumed to be normally distributed [25]. The results, which are calculated by OriginPro software, are shown in Table 7. In addition, the detailed results of rank-DE, rank-GODE, and rank-GaDE for all functions at different dimensions are shown in Tables 15–17 in the Appendix A.

F	DE	rank-DE	GODE	rank-GODE	GaDE	rank-GaDE
F1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F2	4.45E+00	**1.69E−01**	3.65E+00	**2.10E−01**	3.88E+01	**4.74E+00**
F3	8.01E+01	**3.39E+01**	8.14E+01	**4.14E+01**	5.89E+01	**2.22E+00**
F4	**7.96E−02**	1.19E−01	8.32E−14	**0.00E+00**	0.00E+00	0.00E+00
F5	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F6	3.10E−13	**1.42E−14**	2.60E−14	**1.48E−14**	0.00E+00	0.00E+00
F7	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F8	3.69E+02	**1.75E+01**	7.53E+01	**8.50E−06**	1.23E−03	**3.34E−06**
F9	5.06E+02	**1.04E−07**	1.46E−05	**7.32E−10**	3.87E−07	**0.00E+00**
F10	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F11	1.28E−04	**1.13E−07**	1.58E−05	**7.30E−10**	4.34E−07	**0.00E+00**
F12	5.99E−11	**0.00E+00**	7.57E−12	**0.00E+00**	0.00E+00	0.00E+00
F13	6.17E+01	**2.49E+01**	6.32E+01	**2.87E+01**	4.99E+01	**8.96E−01**
F14	4.79E−02	**3.98E−02**	**4.13E−08**	3.98E−02	7.90E−13	**0.00E+00**
F15	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F16	3.58E−09	**1.46E−13**	3.75E−10	**1.24E−12**	2.45E−12	**4.21E−13**
F17	1.23E+01	**1.03E−01**	1.11E+01	**8.98E−02**	3.28E+00	**7.19E−01**
F18	2.98E−04	**2.66E−09**	1.11E−06	**1.30E−08**	1.96E−08	**2.47E−09**
F19	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00

All the results below 1.00E−14 have been approximated to 0.

Table 3. Comparison of the mean error values between DEs and their corresponding rank-DEs for functions F1–F19 at $D = 100$

4.2.1 Comparison Between DE and Rank-DE

First, the results of rank-DE is compare with those of DE. From the results shown in Tables 2–6, we can see that:

- For all functions at $D = 50$, in 5 functions (F1, F7, F10, F15, and F19) both DE and rank-DE are able to find the global optimum over all runs. In 12 out of 19 functions, our proposed rank-DE obtains better mean error values than DE. Only in one function (F5), DE is better than rank-DE. In F5, rank-DE occasionally converges to the local optima. The reason might be that the ranking-based mutation operator in rank-DE leads to over-exploitation in this problem. Therefore, this motivates us to study more sophisticated ranking technique that can control the selection pressure adaptively. We will leave it in our future work.
- When $D = 100$, there are 6 functions (F1, F5, F7, F10, F15, and F19) whose global optimum are obtained by both DE and rank-DE over all runs. rank-DE provides better results than DE in 12 out of 19 functions, but only loses in one function (F4).
- With respect to $D = 200$, similar to the results at $D = 100$, both DE and rank-DE get the global optimum in 6 functions (F1, F5, F7, F10, F15, and F19). In

F	DE	rank-DE	GODE	rank-GODE	GaDE	rank-GaDE
F1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F2	1.92E+01	**3.22E+00**	1.53E+01	**3.59E+00**	5.76E+01	**2.86E+01**
F3	1.78E+02	**1.36E+02**	1.80E+02	**1.42E+02**	1.61E+02	**9.03E+01**
F4	**1.27E−01**	1.59E−01	**4.17E−13**	3.98E−02	0.00E+00	0.00E+00
F5	0.00E+00	0.00E+00	0.00E+00	0.00E+00	**0.00E+00**	5.91E−04
F6	6.54E−13	**3.09E−14**	5.45E−14	**3.24E−14**	0.00E+00	0.00E+00
F7	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F8	5.53E+03	**1.15E+03**	2.10E+03	**9.33E−07**	3.02E+00	**6.94E−01**
F9	1.01E+03	**8.19E−07**	3.23E−05	**9.66E−11**	**4.53E−09**	7.09E−07
F10	0.00E+00	0.00E+00	0.00E+00	0.00E+00	4.20E−02	4.20E−02
F11	2.62E−04	**8.00E−07**	3.12E−05	**1.18E−10**	**1.85E−07**	2.21E−06
F12	9.76E−10	**2.38E−14**	1.20E−10	**2.30E−13**	4.92E−14	**0.00E+00**
F13	1.36E+02	**1.09E+02**	1.38E+02	**1.11E+02**	1.24E+02	**7.63E+01**
F14	1.38E−01	**1.19E−01**	**8.17E−02**	1.59E−01	2.87E−12	**2.17E−13**
F15	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F16	7.46E−09	**1.84E−12**	9.54E−10	**1.35E−11**	**1.58E−12**	5.96E−12
F17	3.70E+01	**1.13E+01**	3.74E+01	**1.26E+01**	2.45E+01	**7.54E−01**
F18	**4.73E−04**	7.96E−02	**1.91E−06**	3.98E−02	2.53E−08	**2.39E−08**
F19	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00

All the results below 1.00E−14 have been approximated to 0.

Table 4. Comparison of the mean error values between DEs and their corresponding rank-DEs for functions F1–F19 at $D = 200$

11 functions, rank-DE is better than DE in terms of the mean error values. In 2 functions (F4 and F18), DE provides better results than rank-DE.

- For all function at $D = 500$, also in 6 functions (F1, F5, F7, F10, F15, and F19) both DE and rank-DE obtain the global optimum over all runs. rank-DE is capable of providing better results in 10 out of 19 functions, but loses in three functions (F4, F14, and F18).
- When the dimension is scaled up to $D = 1\,000$, for functions F1, F5, F7, F10, F15, and F19, their global optimum are found by both rank-DE and DE over all 25 runs. In 12 out of 19 functions, rank-DE improves the results of DE. DE only gets better mean error value in one function (F18) than that of rank-DE.

4.2.2 Comparison Between GODE and Rank-GODE

In this section, the ranking-based mutation operator is integrated into GODE [32] to verify the enhanced performance of our approach. The mean error values of rank-GODE and GODE are shown in Tables 2–6. From the results, it can be observed that:

F	DE	rank-DE	GODE	rank-GODE	GaDE	rank-GaDE
F1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F2	5.35E+01	**2.35E+01**	5.81E+01	**2.31E+01**	7.42E+01	**4.69E+01**
F3	4.76E+02	**4.35E+02**	4.76E+02	**4.34E+02**	4.40E+02	**3.80E+02**
F4	**3.20E−01**	4.38E−01	**1.62E−03**	2.39E−01	0.00E+00	0.00E+00
F5	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F6	1.65E−12	**8.22E−14**	1.43E−13	**8.88E−14**	**1.46E−14**	3.44E−14
F7	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F8	6.09E+04	**2.68E+04**	3.93E+04	**0.00E+00**	1.33E+03	**1.32E+03**
F9	2.52E+03	**6.28E−06**	7.84E−05	**4.20E−14**	**0.00E+00**	4.44E−05
F10	0.00E+00	0.00E+00	0.00E+00	0.00E+00	3.78E−01	**1.26E−01**
F11	6.76E−04	**6.22E−06**	8.25E−05	**3.72E−14**	**0.00E+00**	4.04E−05
F12	7.07E−09	**2.43E−12**	7.39E−10	**1.81E−11**	**1.07E−12**	7.04E−12
F13	3.59E+02	**3.31E+02**	3.59E+02	**3.34E+02**	3.34E+02	**3.07E+02**
F14	**1.35E−01**	3.18E−01	**7.67E−02**	2.79E−01	2.79E−11	**8.42E−12**
F15	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F16	2.04E−08	**2.72E−11**	2.24E−09	**1.72E−10**	**1.67E−12**	1.38E−10
F17	1.11E+02	**8.69E+01**	1.12E+02	**8.84E+01**	9.26E+01	**5.24E+01**
F18	**1.22E−03**	3.98E−02	5.06E−06	**1.49E−06**	5.59E−08	**3.99E−10**
F19	0.00E+00	0.00E+00	0.00E+00	0.00E+00	4.20E−02	**0.00E+00**

All the results below 1.00E−14 have been approximated to 0.

Table 5. Comparison of the mean error values between DEs and their corresponding rank-DEs for functions F1–F19 at $D = 500$

- In 6 functions (F1, F5, F7, F10, F15, and F19) at $D = 50$, 100, 200, and 500, both rank-GODE and GODE consistently get the global optimum over all runs. At $D = 1\,000$, in 9 functions (F1, F5, F7–F11, F15, and F19) rank-GODE still obtains the global optimum over all 25 runs. While GODE finds the global optimum only in 4 functions.
- Regardless of the influence of dimensionality, in the majority of the test functions, our proposed rank-GODE consistently provides better results than those of GODE. In 12, 12, 10, 11, and 14 functions, rank-GODE respectively gets better mean error values than GODE at $D = 50$, 100, 200, 500, and 1 000.
- Rank-GODE is only worse than GODE in 1, 1, 3, 2, and 1 out of 19 functions at $D = 50$, $100, 200$, 500, and 1 000, respectively.

4.2.3 Comparison Between GaDE and Rank-GaDE

GaDE, proposed by Yang et al. [36], is an adaptive DE variant with a new proposed generalized parameter adaptation scheme and strategy adaptation. In this section, our proposed ranking-based vector selection technique is integrated into both of the mutation operators used in GaDE. The results of rank-GaDE and GaDE are

F	DE	rank-DE	GODE	rank-GODE	GaDE	rank-GaDE
F1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F2	8.46E+01	**5.03E+01**	9.02E+01	**4.79E+01**	8.93E+01	**4.34E+01**
F3	9.69E+02	**9.27E+02**	9.70E+02	**9.30E+02**	9.45E+02	**8.76E+02**
F4	1.44E+00	**5.97E−01**	1.03E+00	**7.56E−01**	0.00E+00	0.00E+00
F5	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F6	3.29E−12	**1.75E−13**	2.88E−13	**1.86E−13**	**1.66E−14**	5.41E−14
F7	0.00E+00	0.00E+00	INF	**0.00E+00**	0.00E+00	0.00E+00
F8	2.46E+05	**1.37E+05**	1.86E+05	**0.00E+00**	1.77E+04	**1.59E+04**
F9	5.13E+03	**2.26E−05**	1.70E−04	**0.00E+00**	**0.00E+00**	1.80E−04
F10	0.00E+00	0.00E+00	0.00E+00	0.00E+00	4.62E−01	**8.40E−02**
F11	1.35E−03	**2.29E−05**	1.73E−04	**0.00E+00**	**0.00E+00**	1.73E−04
F12	1.68E−08	**2.30E−11**	1.87E−09	**1.57E−10**	**3.85E−12**	1.49E−10
F13	7.30E+02	**7.06E+02**	7.31E+02	**7.08E+02**	7.15E+02	**6.80E+02**
F14	6.90E−01	**3.98E−01**	6.06E−01	**3.98E−01**	8.82E−11	**7.18E−12**
F15	0.00E+00	0.00E+00	INF	**0.00E+00**	0.00E+00	0.00E+00
F16	4.18E−08	**1.28E−10**	4.59E−09	**8.00E−10**	**2.35E−12**	6.78E−10
F17	2.36E+02	**2.11E+02**	2.36E+02	**2.14E+02**	2.19E+02	**1.80E+02**
F18	**2.37E−03**	3.98E−02	**3.29E−05**	3.98E−02	1.30E−07	**1.62E−08**
F19	0.00E+00	0.00E+00	0.00E+00	0.00E+00	3.78E−01	**0.00E+00**

All the results below 1.00E−14 have been approximated to 0.

Table 6. Comparison of the mean error values between DEs and their corresponding rank-DEs for functions F1–F19 at $D = 1\,000$

reported in Tables 2–6. All of the results are averaged over 25 independent runs. The results in Tables 2–6 show that:

- When $D = 50$, both rank-GaDE and GaDE can solve 8 functions (F1, F4, F6, F7, F10, F12, F15, and F19) over all runs. rank-GaDE improves the mean error values of GaDE in 10 out of 19 functions. GaDE only obtains better results than rank-GaDE in function F5.
- For all functions at $D = 100$, in 9 functions (F1, F4–F7, F10, F12, F15, and F19), their global optimum are obtained by rank-GaDE and GaDE consistently. In the rest of 10 functions, rank-GaDE gets better results than GaDE.
- For all functions at $D = 200$, rank-GaDE obtains better results in 8 functions, but loses in 4 functions compared with GaDE. In the rest of 7 functions, both rank-GaDE and GaDE get the same mean error values.
- With respect to $D = 500$, in 9 out of 19 functions rank-GaDE is capable of provide better results than GaDE. rank-GaDE is worse than GaDE in 5 functions. Both GaDE and rank-GaDE consistently find the global optimum in the rest of 5 functions (F1, F4, F5, F7, and F15).

$D = 50$					
Algorithm	R^+	R^-	p-value	significance at $\alpha = 0.05$	significance at $\alpha = 0.1$
rank-DE vs. DE	97	8	3.05E−03	+	+
rank-GODE vs. GODE	83	8	6.10E−03	+	+
rank-GaDE vs. GaDE	59	7	1.86E−02	+	+
$D = 100$					
Algorithm	R^+	R^-	p-value	significance at $\alpha = 0.05$	significance at $\alpha = 0.1$
rank-DE vs. DE	84	7	4.64E−03	+	+
rank-GODE vs. GODE	83	8	6.10E−03	+	+
rank-GaDE vs. GaDE	55	0	1.95E−03	+	+
$D = 200$					
Algorithm	R^+	R^-	p-value	significance at $\alpha = 0.05$	significance at $\alpha = 0.1$
rank-DE vs. DE	78	13	2.15E−02	+	+
rank-GODE vs. GODE	70	21	9.42E−02	=	+
rank-GaDE vs. GaDE	57	21	1.76E−01	=	=
$D = 500$					
Algorithm	R^+	R^-	p-value	significance at $\alpha = 0.05$	significance at $\alpha = 0.1$
rank-DE vs. DE	73	18	5.74E−02	=	+
rank-GODE vs. GODE	76	15	3.27E−02	+	+
rank-GaDE vs. GaDE	85	20	4.19E−02	+	+
$D = 1\,000$					
Algorithm	R^+	R^-	p-value	significance at $\alpha = 0.05$	significance at $\alpha = 0.1$
rank-DE vs. DE	86	5	2.44E−03	+	+
rank-GODE vs. GODE⋆	114	6	8.54E−04	+	+
rank-GaDE vs. GaDE	84	21	4.94E−02	+	+

⋆ In GODE, for functions F7 and F15 "INF" is approximated to 1.00E+20 to make the multiple-problem Wilcoxon's test.

Table 7. Results of the multiple-problem Wilcoxon's test for all DE variants on the mean error values of functions F1–F19

- When $D = 1\,000$, similar to the results at $D = 500$, in 5 functions (F1, F4, F5, F7, and F15) GaDE and rank-GaDE get the global optimum over all runs. rank-GaDE improves GaDE in 9 functions, but loses in 5 functions.

4.2.4 Summary

To summarize the results shown in Tables 2–6, the multiple-problem analysis on the mean error values in all functions is tabulated in Table 7. From Table 7, it is clear to

F	DE	CHC	G-CMA-ES	rank-DE	rank-GODE	rank-GaDE
F1	**0.00E+00**	**1.67E−11**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F2	3.60E−01	6.19E+01	**2.75E−11**	**8.15E−04**	1.33E−03	2.69E+00
F3	2.89E+01	1.25E+06	7.97E−01	1.59E−01	**1.87E−09**	**3.24E−12**
F4	**3.98E−02**	7.43E+01	1.05E+02	**3.98E−02**	**3.98E−02**	**0.00E+00**
F5	**0.00E+00**	1.67E−03	**2.96E−04**	9.85E−04	**0.00E+00**	8.88E−04
F6	**1.43E−13**	6.15E−07	2.09E+01	**0.00E+00**	**0.00E+00**	**0.00E+00**
F7	**0.00E+00**	2.66E−09	**1.01E−10**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F8	3.44E+00	2.24E+02	**0.00E+00**	3.45E−03	**4.42E−08**	**0.00E+00**
F9	2.73E+02	3.10E+02	1.66E+01	9.91E−09	**4.39E−10**	**0.00E+00**
F10	**0.00E+00**	7.30E+00	**6.81E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F11	6.23E−05	2.16E+00	3.01E+01	1.05E−08	**6.93E−10**	**0.00E+00**
F12	**5.35E−13**	9.57E−01	1.88E+02	**0.00E+00**	**0.00E+00**	**0.00E+00**
F13	2.45E+01	2.08E+06	1.97E+02	**4.98E−02**	**5.05E−02**	6.24E−01
F14	4.16E−08	6.17E+01	1.09E+02	**3.35E−14**	5.79E−13	**0.00E+00**
F15	**0.00E+00**	3.98E−01	**9.79E−04**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F16	1.56E−09	2.95E−09	4.27E+02	**0.00E+00**	**5.35E−14**	**0.00E+00**
F17	7.98E−01	2.26E+04	6.89E+02	**2.21E−01**	**3.96E−02**	2.49E−01
F18	1.22E−04	1.58E+01	1.31E+02	**1.18E−10**	6.30E−10	**2.40E−10**
F19	**0.00E+00**	3.59E+02	**4.76E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**

All the results below 1.00E−14 have been approximated to 0.

Table 8. Comparison of the mean error values among three baseline algorithms and rank-DEs for functions F1–F19 at $D = 50$

see that the ranking-based DE variants consistently provides higher R^+ values than those of non-ranking-based DEs, which means that the ranking-based DE variants are consistently better than the original DE mutation based methods. At $\alpha = 0.05$, in 12 out of 15 cases rank-DEs get significantly better results than non-rank-DEs according to the Wilcoxon's test. In addition, when $\alpha = 0.1$, rank-DEs significantly outperforms non-rank-DEs in 18 out of 19 cases.

In general, from the results shown in Tables 2–7 and the above analysis, we can conclude that our proposed ranking-based vector selection technique is really capable of improving the performance of DE. The reason is that the ranking-based mutation operators enhance the exploitation ability and make ranking-based DE balance the exploration and exploitation abilities.

4.3 Comparison with Baseline Algorithms

In the previous section, we have verified the enhanced performance of our proposed ranking-based mutation operators. In this section, in order to make an analysis of the scalability behavior of our proposed rank-DEs, the comparison to three baseline evolutionary algorithms for continuous optimization problems is performed as suggested in [12]. The three baseline algorithms are

F	DE	CHC	G-CMA-ES	rank-DE	rank-GODE	rank-GaDE
F1	**0.00E+00**	**3.56E−11**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F2	4.45E+00	8.58E+01	**1.51E−10**	**1.69E−01**	2.10E−01	4.74E+00
F3	8.01E+01	4.19E+06	**3.88E+00**	3.39E+01	4.14E+01	**2.22E+00**
F4	**7.96E−02**	2.19E+02	2.50E+02	1.19E−01	**0.00E+00**	**0.00E+00**
F5	**0.00E+00**	3.83E−03	**1.58E−03**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F6	3.10E−13	4.10E−07	2.12E+01	**1.42E−14**	1.48E−14	**0.00E+00**
F7	**0.00E+00**	1.40E−02	**4.22E−04**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F8	3.69E+02	1.69E+03	**0.00E+00**	1.75E+01	8.50E−06	**3.34E−06**
F9	5.06E+02	5.86E+02	1.02E+02	1.04E−07	**7.32E−10**	**0.00E+00**
F10	**0.00E+00**	3.30E+01	**1.66E+01**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F11	1.28E−04	7.32E+01	1.64E+02	1.13E−07	**7.30E−10**	**0.00E+00**
F12	**5.99E−11**	1.03E+01	4.17E+02	**0.00E+00**	**0.00E+00**	**0.00E+00**
F13	6.17E+01	2.70E+06	4.21E+02	**2.49E+01**	2.87E+01	**8.96E−01**
F14	4.79E−02	1.66E+02	2.55E+02	**3.98E−02**	**3.98E−02**	**0.00E+00**
F15	**0.00E+00**	8.13E+00	**6.30E−01**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F16	3.58E−09	2.23E+01	8.59E+02	**1.46E−13**	1.24E−12	**4.21E−13**
F17	1.23E+01	1.47E+05	1.51E+03	**1.03E−01**	**8.98E−02**	7.19E−01
F18	2.98E−04	7.00E+01	3.07E+02	**2.66E−09**	1.30E−08	**2.47E−09**
F19	**0.00E+00**	5.45E+02	**2.02E+01**	**0.00E+00**	**0.00E+00**	**0.00E+00**

All the results below 1.00E−14 have been approximated to 0.

Table 9. Comparison of the mean error values among three baseline algorithms and rank-DEs for functions F1–F19 at $D = 100$

- DE: the original DE algorithm with "DE/rand/1/exp" strategy, $CR = 0.9$, and $F = 0.5$ [28];
- CHC: the real-coded CHC proposed by Eshelman and Schaffer [7];
- G-CMA-ES: a restart CMA-ES with increasing population size [2].

We obtained the results of DE, CHC and G-CMA-ES from `http://sci2s.ugr.es/eamhco/SOCO-results.xls`. The results of DE, CHC, G-CMA-ES, rank-DE, rank-GODE, and rank-GaDE for all functions at $D = 50$, 100, 200, 500, and 1 000 are respectively reported in Tables 8–12. In these tables, the best and second best results are highlighted in **grey boldface** and **boldface**, respectively. In addition, the average rankings obtained by each above algorithm in the Friedman test[1] are tabulated in Table 13.

From the results shown in Tables 8–12, we can observe that regardless of the dimensionality the ranking-based DE variants always get the 1st best mean error values than the three baseline algorithms in the majority of the functions. For example, for all functions at $D = 500$, rank-DE, rank-GODE, rank-GaDE, DE, and

[1] The KEEL software [1] (`http://www.keel.es/`) is used to get the average rankings obtained by each algorithm based on the Friedman test.

F	DE	CHC	G-CMA-ES	rank-DE	rank-GODE	rank-GaDE
F1	**0.00E+00**	**8.34E−01**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F2	1.92E+01	1.03E+02	**1.16E−09**	**3.22E+00**	3.59E+00	2.86E+01
F3	1.78E+02	2.01E+07	**8.91E+01**	1.36E+02	1.42E+02	**9.03E+01**
F4	1.27E−01	5.40E+02	6.48E+02	1.59E−01	**3.98E−02**	**0.00E+00**
F5	**0.00E+00**	8.76E−03	**0.00E+00**	**0.00E+00**	**0.00E+00**	**5.91E−04**
F6	6.54E−13	1.23E+00	2.14E+01	**3.09E−14**	3.24E−14	**0.00E+00**
F7	**0.00E+00**	2.59E−01	**1.17E−01**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F8	5.53E+03	9.38E+03	**0.00E+00**	1.15E+03	**9.33E−07**	6.94E−01
F9	1.01E+03	1.19E+03	3.75E+02	8.19E−07	**9.66E−11**	**7.09E−07**
F10	**0.00E+00**	7.13E+01	4.43E+01	**0.00E+00**	**0.00E+00**	**4.20E−02**
F11	2.62E−04	3.85E+02	8.03E+02	**8.00E−07**	**1.18E−10**	2.21E−06
F12	9.76E−10	7.44E+01	9.06E+02	**2.38E−14**	2.30E−13	**0.00E+00**
F13	1.36E+02	5.75E+06	9.43E+02	**1.09E+02**	1.11E+02	**7.63E+01**
F14	1.38E−01	4.29E+02	6.09E+02	**1.19E−01**	1.59E−01	**2.17E−13**
F15	**0.00E+00**	2.14E+01	**1.75E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F16	7.46E−09	1.60E+02	1.92E+03	**1.84E−12**	1.35E−11	**5.96E−12**
F17	3.70E+01	1.75E+05	3.36E+03	**1.13E+01**	1.26E+01	**7.54E−01**
F18	**4.73E−04**	2.12E+02	6.89E+02	7.96E−02	3.98E−02	**2.39E−08**
F19	**0.00E+00**	2.06E+03	**7.52E+02**	**0.00E+00**	**0.00E+00**	**0.00E+00**

All the results below 1.00E−14 have been approximated to 0.

Table 10. Comparison of the mean error values among three baseline algorithms and rank-DEs for functions F1–F19 at $D = 200$

G-CMA-ES provides the 1st best results in 8, 9, 11, 6 and 3 functions, respectively. There are no functions that CHC obtains the best overall results.

The p-values computed by Iman-Davenport test on the mean error values shown in Tables 8–12 are respectively 1.50E−11, 1.00E−12, 2.30E−11, 2.90E−11, and 1.00E−11 at $D = 50$, 100, 200, 500, and 1 000. The results indicate that there are significant differences in the behavior of the compared six algorithms for all the functions at $\alpha = 0.05$, regardless of the dimensionality of the test functions.

According to the average rankings obtained by each algorithm in the Friedman test shown in Table 13, the results show that all of our proposed ranking-based DE variants obtain better rankings than the three compared baseline algorithms. Regardless of the dimensionality, in all cases, rank-GaDE gets the 1st ranking, followed by rank-GODE, rank-DE, DE, G-CMA-ES (except $D = 500$ and $D = 1\,000$)[2], and CHC.

[2] In G-CMA-ES, when $D = 500$ and $D = 1\,000$ the average error values of some functions are greater than $1.00E + 100$, therefore, the average rankings obtained by the Friedman test do not include the G-CMA-ES in these two cases.

F	DE	CHC	G-CMA-ES	rank-DE	rank-GODE	rank-GaDE
F1	**0.00E+00**	**2.84E−12**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F2	5.35E+01	1.29E+02	**3.48E−04**	2.35E+01	**2.31E+01**	4.69E+01
F3	4.76E+02	1.14E+06	**3.58E+02**	4.35E+02	4.34E+02	**3.80E+02**
F4	**3.20E−01**	1.91E+03	2.10E+03	4.38E−01	2.39E−01	**0.00E+00**
F5	**0.00E+00**	6.98E−03	**2.96E−04**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F6	1.65E−12	5.16E+00	2.15E+01	**8.22E−14**	8.88E−14	**3.44E−14**
F7	**0.00E+00**	**1.27E−01**	7.21E+153	**0.00E+00**	**0.00E+00**	**0.00E+00**
F8	6.09E+04	7.22E+04	**2.36E−06**	2.68E+04	**0.00E+00**	1.32E+03
F9	2.52E+03	3.00E+03	1.74E+03	**6.28E−06**	**4.20E−14**	4.44E−05
F10	**0.00E+00**	1.86E+02	1.27E+02	**0.00E+00**	**0.00E+00**	**1.26E−01**
F11	6.76E−04	1.81E+03	4.16E+03	**6.22E−06**	**3.72E−14**	4.04E−05
F12	7.07E−09	4.48E+02	2.58E+03	**2.43E−12**	1.81E−11	**7.04E−12**
F13	3.59E+02	3.22E+07	2.87E+03	**3.31E+02**	3.34E+02	**3.07E+02**
F14	**1.35E−01**	1.46E+03	1.95E+03	3.18E−01	2.79E−01	**8.42E−12**
F15	**0.00E+00**	**6.01E+01**	2.82E+262	**0.00E+00**	**0.00E+00**	**0.00E+00**
F16	2.04E−08	9.55E+02	5.45E+03	**2.72E−11**	1.72E−10	**1.38E−10**
F17	1.11E+02	8.40E+05	9.59E+03	**8.69E+01**	8.84E+01	**5.24E+01**
F18	1.22E−03	7.32E+02	2.05E+03	3.98E−02	**1.49E−06**	**3.99E−10**
F19	**0.00E+00**	**1.76E+03**	2.44E+06	**0.00E+00**	**0.00E+00**	**0.00E+00**

All the results below 1.00E−14 have been approximated to 0.

Table 11. Comparison of the mean error values among three baseline algorithms and rank-DEs for functions F1–F19 at $D = 500$

4.4 Comparison with Reported Results

In the special issue of Soft Computing [12], there are 13 papers published therein. All of the results are available online at `http://sci2s.ugr.es/eamhco/SOCO-results.xls`. In this subsection, we compare the results of rank-DE, rank-GODE, and rank-GaDE with those of the 13 advanced methods. With respect to the mean error values, the average rankings obtained by each algorithm in the Friedman test are reported in Table 14. From the results, it can be seen that MOS [15], which is a multiple offspring sampling method containing different search strategies, consistently obtains the best ranking regardless of the dimensionality. The ranking of jDElscop and rank-GaDE in different dimensions of problems are twisted: in $D = 50$, 200, and 500, jDElscop is better than rank-GaDE; while in $D = 100$ and 1 000, rank-GaDE provides better rankings than jDElscop. However, in overall, rank-GaDE obtains the 2$^{\text{nd}}$ ranking, following by jDElscop, rank-GODE, and rank-DE. It is worth noting that although the ranking-based DE variants are not the best one among all compared algorithms, they can provide promising results. More importantly, they improves their non-ranking-based DEs markedly, for example, the overall ranking of rank-GaDE is 2, while GaDE only ranks 7.

F	DE	CHC	G-CMA-ES	rank-DE	rank-GODE	rank-GaDE
F1	**0.00E+00**	**1.36E−11**	NA	**0.00E+00**	**0.00E+00**	**0.00E+00**
F2	8.46E+01	1.44E+02	NA	5.03E+01	**4.79E+01**	**4.34E+01**
F3	9.69E+02	8.75E+03	NA	**9.27E+02**	9.30E+02	**8.76E+02**
F4	1.44E+00	4.76E+03	NA	**5.97E−01**	7.56E−01	**0.00E+00**
F5	**0.00E+00**	**7.02E−03**	NA	**0.00E+00**	**0.00E+00**	**0.00E+00**
F6	3.29E−12	1.38E+01	NA	**1.75E−13**	1.86E−13	**5.41E−14**
F7	**0.00E+00**	**3.52E−01**	NA	**0.00E+00**	**0.00E+00**	**0.00E+00**
F8	2.46E+05	3.11E+05	NA	1.37E+05	**0.00E+00**	**1.59E+04**
F9	5.13E+03	6.11E+03	NA	**2.26E−05**	**0.00E+00**	1.80E−04
F10	**0.00E+00**	3.83E+02	NA	**0.00E+00**	**0.00E+00**	**8.40E−02**
F11	1.35E−03	4.82E+03	NA	**2.29E−05**	**0.00E+00**	1.73E−04
F12	1.68E−08	1.05E+03	NA	**2.30E−11**	1.57E−10	**1.49E−10**
F13	7.30E+02	6.66E+07	NA	**7.06E+02**	7.08E+02	**6.80E+02**
F14	6.90E−01	3.62E+03	NA	**3.98E−01**	**3.98E−01**	**7.18E−12**
F15	**0.00E+00**	**8.37E+01**	NA	**0.00E+00**	**0.00E+00**	**0.00E+00**
F16	4.18E−08	2.32E+03	NA	**1.28E−10**	8.00E−10	**6.78E−10**
F17	2.36E+02	2.04E+07	NA	**2.11E+02**	2.14E+02	**1.80E+02**
F18	**2.37E−03**	1.72E+03	NA	3.98E−02	3.98E−02	**1.62E−08**
F19	**0.00E+00**	**4.20E+03**	NA	**0.00E+00**	**0.00E+00**	**0.00E+00**

All the results below 1.00E−14 have been approximated to 0.

Table 12. Comparison of the mean error values among three baseline algorithms and rank-DEs for functions F1–F19 at $D = 1\,000$. "NA" means the results are not available.

Algorithm	Ranking ($D = 50$)	Ranking ($D = 100$)	Ranking ($D = 200$)	Ranking ($D = 500$)	Ranking ($D = 1\,000$)
DE	3.5	3.5263	3.3158	3.2368	3.3421
CHC	5.5	5.5	5.6316	4.8947	4.8947
G-CMA-ES	4.5789	4.6579	4.4474	NA	NA
rank-DE	2.6579	2.6579	2.6316	2.6053	2.3947
rank-GODE	**2.3947**	**2.5263**	**2.5263**	**2.1316**	**2.3421**
rank-GaDE	**2.3684**	**2.1316**	**2.4474**	**2.0893**	**2.0263**

Table 13. Average rankings obtained by each algorithm in the Friedman test. "NA" means not available.

5 CONCLUSIONS

In this paper, we employ our proposed modified ranking-based mutation operators to enhance the performance of differential evolution. In the ranking-based mutation operators, the vectors in the mutation operators are selected according to their rankings in the current population. Better solutions are more likely to be selected to be the placing and leading vectors, while worse solutions have more chance to be chosen as the correcting vector(s). In general, the proposed ranking-based vector technique

Algorithm	Ranking ($D = 50$)	Ranking ($D = 100$)	Ranking ($D = 200$)	Ranking ($D = 500$)	Ranking ($D = 1\,000$)	Average	Overall
SOUPDE	7.7632	8.2105	8.2632	8.0263	7.3684	7.9263	9
DE−D^{40}+M$^{\mathrm{m}}$	8.3947	8.5263	8.4737	7.8421	7.3158	8.1105	10
GODE	8.9211	8.9737	8.3947	8.3947	6.5789	8.2526	11
GaDE	7.6316	7.5000	6.7895	7.3158	NA	7.3092	7
jDElscop	**5.9211**	6.5789	**6.2105**	**6.2632**	6.5000	6.2947	3
SaDE−MMTS	6.6579	7.3421	7.3421	7.8421	7.1316	7.2632	6
MOS	**5.6053**	**5.7632**	**5.0263**	**5.0000**	**4.5526**	**5.1895**	**1**
MA-SSW-Chains	9.3684	10.1842	10.6579	12.0526	10.6053	10.5737	15
RPSO-vm	11.5526	10.9211	10.8684	10.3684	8.3684	10.4158	14
Tuned IPSOLS	9.6842	7.7105	7.7105	7.6053	6.5263	7.8474	8
EvoPROpt	15.1316	15.0000	14.8421	14.0000	12.8421	14.3632	16
EM323	10.1842	9.1053	9.1842	9.7895	NA	9.5658	12
VXQR1	10.3947	10.6316	11.0263	10.6842	8.9211	10.3316	13
rank-DE	6.5526	7.3158	7.5263	7.7105	6.5526	7.1316	5
rank-GODE	6.1842	6.4474	6.8421	6.6579	6.0000	6.4263	4
rank-GaDE	6.0526	**5.7895**	6.8421	6.4474	**5.7368**	**6.1737**	**2**

Table 14. Average rankings obtained by different algorithms in the Friedman test. "NA" means not available.

is very simple, and it does not introduce any new parameters. In order to verify the performance of our proposed ranking-based mutation operators, they are integrated into the original DE, GODE, and GaDE; rank-DE, rank-GODE, and rank-GaDE are evaluated on the large-scale continuous optimization problems presented in the special issue of Soft Computing. Experimental results verify our expectation that the ranking-based mutation operators are consistently able to enhance the performance of DE, GODE, and GaDE. Regardless of the dimensionality, ranking-based DEs achieve very promising results in the large-scale continuous optimization. Compared with the three baseline algorithms, statistical results show that ranking-based DEs still obtain better rankings.

The ranking-based mutation operators may also be useful in the constrained optimization and multiobjective optimization. For example, the stochastic ranking technique [24] and non-dominated sorting method [6] can be possibly used to rank solutions in the constrained optimization and multiobjective optimization. In our future, we will try to verify these expectations.

A APPENDIX

In this section, the detailed results of rank-DE, rank-GODE, and rank-GaDE are reported in Tables 15–17, respectively. In each function, each algorithm is performed over 25 independent runs. In Tables 15–17, the median value is highlighted in **boldface** when it is better than or equal to the mean value in the same function.

D	F	Best	Median	Worst	Mean	D	F	Best	Median	Worst	Mean
50	F1	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	50	F11	6.17E−09	**9.93E−09**	1.77E−08	1.05E−08
100		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	100		6.25E−08	**1.13E−07**	1.75E−07	1.13E−07
200		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	200		6.75E−07	**7.90E−07**	9.43E−07	8.00E−07
500		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	500		5.51E−06	6.25E−06	7.06E−06	6.22E−06
1 000		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	1 000		2.09E−05	**2.26E−05**	2.53E−05	2.29E−05
50	F2	5.21E−04	**7.79E−04**	1.40E−03	8.15E−04	50	F12	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
100		1.34E−01	**1.69E−01**	2.03E−01	1.69E−01	100		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
200		2.82E+00	3.24E+00	3.49E+00	3.22E+00	200		1.72E−14	2.47E−14	3.34E−14	2.38E−14
500		2.23E+01	2.36E+01	2.50E+01	2.35E+01	500		1.58E−12	2.49E−12	3.17E−12	2.43E−12
1 000		4.91E+01	**5.02E+01**	5.16E+01	5.03E+01	1 000		2.01E−11	**2.27E−11**	2.85E−11	2.30E−11
50	F3	1.77E−14	**2.61E−11**	3.99E+00	1.59E−01	50	F13	6.16E−08	**3.85E−07**	3.94E−01	4.98E−02
100		3.12E+01	**3.38E+01**	3.68E+01	3.39E+01	100		1.91E+01	2.52E+01	2.78E+01	2.49E+01
200		1.31E+02	**1.35E+02**	1.75E+02	1.36E+02	200		1.06E+02	**1.07E+02**	1.44E+02	1.09E+02
500		4.29E+02	**4.32E+02**	4.74E+02	4.35E+02	500		3.28E+02	**3.31E+02**	3.35E+02	3.31E+02
1 000		9.24E+02	**9.27E+02**	9.30E+02	9.27E+02	1 000		7.01E+02	**7.02E+02**	7.45E+02	7.06E+02
50	F4	0.00E+00	**0.00E+00**	9.95E−01	3.98E−02	50	F14	0.00E+00	**2.36E−14**	9.57E−14	3.35E−14
100		0.00E+00	**0.00E+00**	9.95E−01	1.19E−01	100		4.29E−12	**7.30E−12**	9.95E−01	3.98E−02
200		0.00E+00	**0.00E+00**	9.95E−01	1.59E−01	200		1.39E−10	**3.43E−10**	9.95E−01	1.19E−01
500		0.00E+00	**0.00E+00**	1.99E+00	4.38E−01	500		2.81E−09	**4.15E−09**	1.99E+00	3.18E−01
1 000		0.00E+00	**0.00E+00**	3.98E+00	5.97E−01	1 000		1.10E−08	**1.38E−08**	1.99E+00	3.98E−01
50	F5	0.00E+00	**0.00E+00**	1.48E−02	9.85E−04	50	F15	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
100		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	100		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
200		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	200		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
500		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	500		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
1 000		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	1 000		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
50	F6	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	50	F16	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
100		1.11E−14	1.47E−14	1.47E−14	1.42E−14	100		9.08E−14	**1.31E−13**	2.27E−13	1.46E−13
200		2.89E−14	3.24E−14	3.24E−14	3.09E−14	200		1.38E−12	**1.84E−12**	2.49E−12	1.84E−12
500		7.86E−14	**8.22E−14**	8.57E−14	8.22E−14	500		2.23E−11	2.74E−11	3.06E−11	2.72E−11
1 000		1.75E−13	**1.75E−13**	1.82E−13	1.75E−13	1 000		1.12E−10	**1.28E−10**	1.40E−10	1.28E−10
50	F7	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	50	F17	3.46E−07	**1.07E−02**	3.99E+00	2.21E−01
100		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	100		1.95E−05	**7.81E−02**	2.84E−01	1.03E−01
200		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	200		7.69E+00	1.16E+01	1.62E+01	1.13E+01
500		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	500		8.50E+01	8.72E+01	8.90E+01	8.69E+01
1 000		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	1 000		2.10E+02	**2.11E+02**	2.16E+02	2.11E+02
50	F8	1.29E−03	**3.45E−03**	8.21E−03	3.45E−03	50	F18	5.53E−11	**1.15E−10**	1.79E−10	1.18E−10
100		8.89E+00	**1.67E+01**	3.01E+01	1.75E+01	100		1.84E−09	**2.43E−09**	5.00E−09	2.66E−09
200		8.92E+02	**1.11E+03**	1.44E+03	1.15E+03	200		2.50E−08	**3.54E−08**	9.95E−01	7.96E−02
500		2.30E+04	**2.66E+04**	3.21E+04	2.68E+04	500		3.15E−07	**3.60E−07**	9.95E−01	3.98E−02
1 000		1.27E+05	**1.37E+05**	1.46E+05	1.37E+05	1 000		1.28E−06	**1.44E−06**	9.95E−01	3.98E−02
50	F9	5.68E−09	**9.27E−09**	2.06E−08	9.91E−09	50	F19	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
100		7.35E−08	1.09E−07	1.40E−07	1.04E−07	100		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
200		6.13E−07	**8.19E−07**	9.60E−07	8.19E−07	200		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
500		5.04E−06	6.42E−06	7.55E−06	6.28E−06	500		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
1 000		2.05E−05	**2.24E−05**	2.66E−05	2.26E−05	1 000		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
50	F10	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	All the results below 1.00E−14 have been approximated to 0.					
100		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00						
200		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00						
500		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00						
1 000		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00						

Table 15. Experimental results of rank-DE for functions F1–F19 at $D = 50$, 100, 200, 500, and 1 000, where the median value is highlighted in **boldface** when it is better than or equal to the mean value in the same function

D	F	Best	Median	Worst	Mean	D	F	Best	Median	Worst	Mean
50	F1	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	50	F11	2.29E−10	**6.18E−10**	1.48E−09	6.93E−10
100		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	100		8.30E−11	**6.86E−10**	1.58E−09	7.30E−10
200		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	200		1.53E−11	**7.12E−11**	5.04E−10	1.18E−10
500		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	500		0.00E+00	**1.46E−14**	2.11E−13	3.72E−14
1 000		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	1 000		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
50	F2	8.02E−04	**1.31E−03**	2.19E−03	1.33E−03	50	F12	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
100		1.72E−01	**2.05E−01**	2.57E−01	2.10E−01	100		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
200		2.93E+00	**3.62E+00**	3.94E+00	3.59E+00	200		1.21E−13	2.33E−13	3.28E−13	2.30E−13
500		2.23E+01	**2.30E+01**	2.42E+01	2.31E+01	500		1.42E−11	**1.77E−11**	2.46E−11	1.81E−11
1 000		4.68E+01	**4.78E+01**	4.96E+01	4.79E+01	1 000		1.37E−10	1.58E−10	1.79E−10	1.57E−10
50	F3	2.94E−13	**7.41E−10**	8.27E−09	1.87E−09	50	F13	1.23E−06	**4.12E−06**	3.48E−01	5.05E−02
100		3.28E+01	**3.60E+01**	8.42E+01	4.14E+01	100		2.49E+01	2.89E+01	3.31E+01	2.87E+01
200		1.34E+02	**1.37E+02**	1.82E+02	1.42E+02	200		1.07E+02	**1.10E+02**	1.49E+02	1.11E+02
500		4.32E+02	**4.34E+02**	4.38E+02	4.34E+02	500		3.31E+02	**3.32E+02**	3.72E+02	3.34E+02
1 000		9.28E+02	**9.30E+02**	9.32E+02	9.30E+02	1 000		7.02E+02	**7.05E+02**	7.41E+02	7.08E+02
50	F4	0.00E+00	**0.00E+00**	9.95E−01	3.98E−02	50	F14	1.03E−13	**5.08E−13**	1.34E−12	5.79E−13
100		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	100		2.28E−11	**9.93E−11**	9.95E−01	3.98E−02
200		0.00E+00	**0.00E+00**	9.95E−01	3.98E−02	200		1.18E−09	**2.67E−09**	1.99E+00	1.59E−01
500		0.00E+00	**0.00E+00**	1.99E+00	2.39E−01	500		1.92E−08	**2.68E−08**	9.95E−01	2.79E−01
1 000		2.38E−13	9.95E−01	2.98E+00	7.56E−01	1 000		7.16E−08	**9.30E−08**	1.99E+00	3.98E−01
50	F5	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	50	F15	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
100		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	100		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
200		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	200		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
500		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	500		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
1 000		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	1 000		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
50	F6	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	50	F16	2.45E−14	**4.99E−14**	1.36E−13	5.35E−14
100		1.47E−14	**1.47E−14**	1.82E−14	1.48E−14	100		6.95E−13	**1.21E−12**	1.76E−12	1.24E−12
200		2.89E−14	**3.24E−14**	3.60E−14	3.24E−14	200		8.54E−12	1.36E−11	1.72E−11	1.35E−11
500		8.57E−14	8.93E−14	9.28E−14	8.88E−14	500		1.44E−10	1.75E−10	1.95E−10	1.72E−10
1 000		1.82E−13	**1.85E−13**	1.92E−13	1.86E−13	1 000		7.25E−10	8.02E−10	8.98E−10	8.00E−10
50	F7	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	50	F17	4.39E−07	**6.60E−06**	2.36E−01	3.96E−02
100		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	100		1.66E−05	**6.28E−02**	2.50E−01	8.98E−02
200		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	200		9.17E+00	1.30E+01	1.45E+01	1.26E+01
500		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	500		8.54E+01	8.88E+01	9.03E+01	8.84E+01
1 000		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	1 000		2.11E+02	**2.14E+02**	2.16E+02	2.14E+02
50	F8	1.38E−10	**8.45E−09**	3.13E−07	4.42E−08	50	F18	2.93E−10	**6.11E−10**	1.08E−09	6.30E−10
100		9.28E−09	**2.84E−07**	7.14E−05	8.50E−06	100		7.10E−09	**1.25E−08**	2.45E−08	1.30E−08
200		1.03E−11	**5.35E−08**	1.42E−05	9.33E−07	200		9.94E−08	**1.36E−07**	9.95E−01	3.98E−02
500		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	500		1.16E−06	**1.47E−06**	1.79E−06	1.49E−06
1 000		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	1 000		4.39E−06	**5.13E−06**	9.95E−01	3.98E−02
50	F9	1.47E−10	**3.64E−10**	1.13E−09	4.39E−10	50	F19	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
100		1.87E−10	**6.01E−10**	2.68E−09	7.32E−10	100		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
200		7.27E−12	**7.92E−11**	3.24E−10	9.66E−11	200		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
500		4.14E−15	**2.19E−14**	2.06E−13	4.20E−14	500		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
1 000		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	1 000		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
50	F10	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	All the results below 1.00E−14 have been approximated to 0.					
100		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00						
200		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00						
500		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00						
1 000		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00						

Table 16. Experimental results of rank-GODE for functions F1–F19 at $D = 50$, 100, 200, 500, and 1 000, where the median value is highlighted in **boldface** when it is better than or equal to the mean value in the same function

D	F	Best	Median	Worst	Mean	D	F	Best	Median	Worst	Mean
50	F1	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	50	F11	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
100		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	100		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
200		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	200		0.00E+00	**4.33E−07**	1.15E−05	2.21E−06
500		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	500		2.64E−05	**3.57E−05**	7.15E−05	4.04E−05
1 000		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	1 000		1.05E−04	**1.71E−04**	2.59E−04	1.73E−04
50	F2	9.10E−01	**2.49E+00**	7.88E+00	2.69E+00	50	F12	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
100		3.59E−01	**3.65E+00**	1.79E+01	4.74E+00	100		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
200		1.84E+01	**2.64E+01**	3.80E+01	2.86E+01	200		0.00E+00	**0.00E+00**	2.59E−14	0.00E+00
500		2.89E+01	4.86E+01	6.32E+01	4.69E+01	500		3.01E−12	**6.40E−12**	1.78E−11	7.04E−12
1 000		2.85E+01	4.48E+01	6.03E+01	4.34E+01	1 000		8.11E−11	**1.40E−10**	2.51E−10	1.49E−10
50	F3	0.00E+00	**0.00E+00**	6.32E−11	3.24E−12	50	F13	4.10E−02	**2.94E−01**	4.32E+00	6.24E−01
100		5.04E−10	**1.04E−01**	1.21E+01	2.22E+00	100		1.73E−01	**8.50E−01**	2.26E+00	8.96E−01
200		4.80E+01	**8.57E+01**	1.39E+02	9.03E+01	200		6.02E+01	**6.59E+01**	1.13E+02	7.63E+01
500		3.38E+02	**3.79E+02**	4.11E+02	3.80E+02	500		2.89E+02	**3.06E+02**	3.30E+02	3.07E+02
1 000		7.95E+02	**8.71E+02**	9.67E+02	8.76E+02	1 000		6.22E+02	**6.78E+02**	7.40E+02	6.80E+02
50	F4	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	50	F14	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
100		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	100		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
200		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	200		5.32E−14	**1.98E−13**	6.29E−13	2.17E−13
500		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	500		4.13E−12	**8.06E−12**	1.25E−11	8.42E−12
1 000		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	1 000		4.05E−12	7.24E−12	1.02E−11	7.18E−12
50	F5	0.00E+00	**0.00E+00**	7.40E−03	8.88E−04	50	F15	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
100		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	100		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
200		0.00E+00	**0.00E+00**	1.48E−02	5.91E−04	200		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
500		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	500		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
1 000		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	1 000		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
50	F6	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	50	F16	0.00E+00	**0.00E+00**	5.48E−14	0.00E+00
100		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	100		1.16E−13	**3.37E−13**	1.12E−12	4.21E−13
200		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	200		2.98E−12	**5.54E−12**	1.25E−11	5.96E−12
500		2.84E−14	**3.20E−14**	6.75E−14	3.44E−14	500		7.88E−11	1.42E−10	1.92E−10	1.38E−10
1 000		4.26E−14	**4.97E−14**	1.28E−13	5.41E−14	1 000		4.13E−10	**6.66E−10**	1.00E−09	6.78E−10
50	F7	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	50	F17	8.12E−08	**2.35E−01**	4.71E−01	2.49E−01
100		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	100		2.82E−01	**6.94E−01**	1.53E+00	7.19E−01
200		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	200		3.85E−01	**7.39E−01**	1.10E+00	7.54E−01
500		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	500		4.44E+01	**5.20E+01**	5.96E+01	5.24E+01
1 000		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	1 000		1.73E+02	**1.80E+02**	1.85E+02	1.80E+02
50	F8	0.00E+00	**0.00E+00**	5.48E−14	0.00E+00	50	F18	5.10E−11	**1.76E−10**	5.92E−10	2.40E−10
100		3.89E−07	**2.41E−06**	9.87E−06	3.34E−06	100		9.64E−10	**2.41E−09**	5.62E−09	2.47E−09
200		1.58E−01	**5.69E−01**	3.68E+00	6.94E−01	200		1.74E−08	**1.99E−08**	7.88E−08	2.39E−08
500		7.36E+02	**1.21E+03**	2.04E+03	1.32E+03	500		2.05E−10	**3.83E−10**	5.98E−10	3.99E−10
1 000		1.33E+04	**1.55E+04**	1.93E+04	1.59E+04	1 000		1.17E−08	**1.51E−08**	2.11E−08	1.62E−08
50	F9	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	50	F19	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
100		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	100		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
200		0.00E+00	**0.00E+00**	5.27E−06	7.09E−07	200		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
500		1.97E−05	**4.39E−05**	7.60E−05	4.44E−05	500		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
1 000		9.73E−05	**1.60E−04**	4.11E−04	1.80E−04	1 000		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
50	F10	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00	All the results below 1.00E−14 have been approximated to 0.					
100		0.00E+00	**0.00E+00**	0.00E+00	0.00E+00						
200		0.00E+00	**0.00E+00**	0.00E+00	1.05E+00						
500		0.00E+00	**0.00E+00**	1.05E+00	1.26E−01						
1 000		0.00E+00	**0.00E+00**	1.05E+00	8.40E−02						

Table 17. Experimental results of rank-GaDE for functions F1–F19 at $D = 50$, 100, 200, 500, and 1 000, where the median value is highlighted in **boldface** when it is better than or equal to the mean value in the same function

From the results it is clear to see that in the majority of the cases the median values are much better than or equal to the corresponding mean values. For example, for rank-GODE there are 85 out of 95 cases where the median values are much better than or equal to the corresponding mean values. The results show that the ranking-based DEs sometimes occasionally converge to the local optima in some functions. But, in general, our proposed ranking-based DEs are able to obtain good solutions within the specified Max_NFFEs.

Acknowledgments

The source codes of GODE and GaDE are obtained available online at `http://sci2s.ugr.es/EAMHCO/contributionsSOCO.php`.

REFERENCES

[1] Alcalá-Fdez, J.—Sánchez, L.—García, S.—del Jesus, M. J.—Ventura, S.—Garrell, J. M.—Otero, J.—Romero, C.—Bacardit, J.—Rivas, V. M.—Fernández, J. C.—Herrera, F.: KEEL: A Software Tool to Assess Evolutionary Algorithms for Data Mining Problems. Soft Computing, Vol. 13, 2009, No. 3, pp. 307–318, doi: 10.1007/s00500-008-0323-y.

[2] Auger, A.—Hansen, N.: A Restart CMA Evolution Strategy with Increasing Population Size. The 2005 IEEE Congress on Evolutionary Computation, 2005, Vol. 2, pp. 1769–1776, doi: 10.1109/CEC.2005.1554902.

[3] Brest, J.—Sepesy Maučec, M.: Self-Adaptive Differential Evolution Algorithm Using Population Size Reduction and Three Strategies. Soft Computing – A Fusion of Foundations, Methodologies and Applications, Vol. 15, 2011, No. 11, pp. 2157–2174, doi: 10.1007/s00500-010-0644-5.

[4] Brest, J.—Zamuda, A.—Fister, I.—Sepesy Maučec, M.: Large Scale Global Optimization Using Self-Adaptive Differential Evolution Algorithm. 2010 IEEE Congress on Evolutionary Computation (CEC), 2010, pp. 1–8, doi: 10.1109/CEC.2010.5585927.

[5] Das, S.—Suganthan, P. N.: Differential Evolution: A Survey of the State-of-the-Art. IEEE Transactions on Evolutionary Computation, Vol. 15, 2011, No. 1, pp. 4–31, doi: 10.1109/TEVC.2010.2059031.

[6] Deb, K.—Pratap, A.—Agarwal, S.—Meyarivan, T.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, Vol. 6, 2002, No. 2, pp. 182–197, doi: 10.1109/4235.996017.

[7] Eshelman, L. J.—Schaffer, J. D.: Real-Coded Genetic Algorithms and Interval-Schemata. In: Whitley, D. L. (Ed.): Foundation of Genetic Algorithms, Vol. 2, 1993, pp. 187–202.

[8] García, S.—Fernández, A.—Luengo, J.—Herrera, F.: A Study of Statistical Techniques and Performance Measures for Genetics-Based Machine Learning: Accuracy and Interpretability. Soft Computing, Vol. 13, 2009, No. 10, pp. 959–977, doi: 10.1007/s00500-008-0392-y.

[9] García, S.—Molina, D.—Lozano, M.—Herrera, F.: 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, 2009, No. 6, pp. 617–644, doi: 10.1007/s10732-008-9080-4.

[10] García-Martínez, C.—Rodríguez, F.—Lozano, M.: 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, 2011, No. 11, pp. 2109–2126, doi: 10.1007/s00500-010-0641-8.

[11] Gong, W.—Cai, Z.: Differential Evolution with Ranking-Based Mutation Operators. IEEE Transactions on Cybernetics, Vol. 43, 2013, No. 6, pp. 2066–2081.

[12] Herrera, F.—Lozano, M.—Molina, D. (Eds.): Special Issue on Scalability of Evolutionary Algorithms and Other Metaheuristics for Large Scale Continuous Optimization Problems. Soft Computing – A Fusion of Foundations, Methodologies and Applications, Vol. 15, 2011, No. 11. Published online: 2010.

[13] Herrera, F.—Lozano, M.—Molina, D.: Test Suite for the Special Issue of Soft Computing on Scalability of Evolutionary Algorithms and Other Metaheuristics for Large Scale Continuous Optimization Problems. Technical report, Department of Computer Science and Artificial Intelligence, University of Granada, 2010.

[14] Korošec, P.—Šilc, J.: Using Stigmergy to Solve Numerical Optimization Problems. Computing and Informatics, Vol. 27, 2008, No. 3, pp. 377–402.

[15] LaTorre, A.—Muelas, S.—Peña, J.-M.: A MOS-Based Dynamic Memetic Differential Evolution Algorithm for Continuous Optimization: A Scalability Test. Soft Computing, Vol. 15, 2011, No. 11, pp. 2187–2199.

[16] Leong, W. J.—Hassan, M. A.: Scaled Memoryless Symmetric Rank One Method for Large-Scale Optimization. Applied Mathematics and Computation, Vol. 218, 2011, No. 2, pp. 413–418.

[17] Li, H.-M.—Zhang, K.-C.: A Decomposition Algorithm for Solving Large-Scale Quadratic Programming Problems. Applied Mathematics and Computation, Vol. 173, 2006, No. 1, pp. 394–403.

[18] Lozano, M.—Molina, D.—Herrera, F.: Editorial Scalability of Evolutionary Algorithms and Other Metaheuristics for Large-Scale Continuous Optimization Problems. Soft Computing – A Fusion of Foundations, Methodologies and Applications, Vol. 15, 2011, No. 11, pp. 2085–2087.

[19] Muelas, S.—La Torre, A.—Peña, J.-M.: A Memetic Differential Evolution Algorithm for Continuous Optimization. Ninth International Conference on Intelligent Systems Design and Applications (ISDA '09), 2009, pp. 1080–1084.

[20] Neri, F.—Tirronen, V.: Recent Advances in Differential Evolution: A Survey and Experimental Analysis. Artificial Intelligence Review, Vol. 33, 2010, No. 1-2, pp. 61–106.

[21] Noman, N.—Iba, H.: Accelerating Differential Evolution Using an Adaptive Local Search. IEEE Transactions on Evolutionary Computation, Vol. 12, 2008, No. 1, pp. 107–125.

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

[23] Qin, A. K.—Huang, V. L.—Suganthan, P. N.: Differential Evolution Algorithm with Strategy Adaptation for Global Numerical Optimization. IEEE Transactions on Evolutionary Computation, Vol. 13, 2009, No. 2, pp. 398–417.

[24] Runarsson, T. P.—Yao, X.: Stochastic Ranking for Constrained Evolutionary Optimization. IEEE Transactions on Evolutionary Computation, Vol. 4, 2000, No. 3, pp. 284–294, doi: 10.1109/4235.873238.

[25] Shaw, C.—Williams, K. S.—Assassa, R. P.: Patients' Views of a New Nurse-Led Continence Service. Journal of Clinical Nursing, Vol. 9, 2003, No. 4, pp. 574–582.

[26] Stanarevic, N.: Hybridizing Artificial Bee Colony (ABC) Algorithm with Differential Evolution for Large Scale Optimization Problems. International Journal of Mathematics and Computers in Simulation, Vol. 1, 2012, No. 6, pp. 194–202.

[27] Storn, R.—Price, K.: Differential Evolution – A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces. Technical report TR-95-012, Berkeley, CA, 1995.

[28] Storn, R.—Price, K.: Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. Journal of Global Optimization, Vol. 11, 1997, No. 4, pp. 341–359.

[29] Tang, K.—Li, X.—Suganthan, P. N.—Yang, Z.—Weise, T.: Benchmark Functions for the CEC '2010 Special Session and Competition on Large-Scale Global Optimization. Technical report, Nature Inspired Computation and Applications Laboratory, USTC, China, 2009.

[30] Tang, K.—Yao, X.—Suganthan, P. N.—MacNish, C.—Chen, Y. P.—Chen, C. M.—Yang, Z.: Benchmark Functions for the CEC '2008 Special Session and Competition on Large Scale Global Optimization. Technical report, Nature Inspired Computation and Applications Laboratory, USTC, China, 2007.

[31] Wang, H.—Ni, Q.: A New Method of Moving Asymptotes for Large-Scale Unconstrained Optimization. Applied Mathematics and Computation, Vol. 203, 2008, No. 1, pp. 62–71.

[32] Wang, H.—Wu, Z.—Rahnamayan, S.: Enhanced Opposition-Based Differential Evolution for Solving High-Dimensional Continuous Optimization Problems. Soft Computing – A Fusion of Foundations, Methodologies and Applications, Vol. 15, 2011, No. 11, pp. 2127–2140.

[33] Weber, M.—Neri, F.—Tirronen, V.: Shuffle or Update Parallel Differential Evolution for Large-Scale Optimization. Soft Computing – A Fusion of Foundations, Methodologies and Applications, Vol. 15, 2011, No. 11, pp. 2089–2107.

[34] Yang, Z.—Tang, K.—Yao, X.: Differential Evolution for High-Dimensional Function Optimization. IEEE Congress on Evolutionary Computation (CEC 2007), 2007, pp. 3523–3530.

[35] Yang, Z.—Tang, K.—Yao, X.: Large Scale Evolutionary Optimization Using Cooperative Coevolution. Information Sciences, Vol. 178, 2008, No. 15, pp. 2985–2999.

[36] Yang, Z.—Tang, K.—Yao, X.: Scalability of Generalized Adaptive Differential Evolution for Large-Scale Continuous Optimization. Soft Computing – A Fusion of Foundations, Methodologies and Applications, Vol. 15, 2011, No. 11, pp. 2141–2155.

[37] Yu, G.—Zhao, Y.—Wei, Z.: A Descent Nonlinear Conjugate Gradient Method for Large-Scale Unconstrained Optimization. Applied Mathematics and Computation, Vol. 187, 2007, No. 2, pp. 636–643.

[38] Zhang, J.—Sanderson, A. C.: JADE: Adaptive Differential Evolution with Optional External Archive. IEEE Transactions on Evolutionary Computation, Vol. 13, 2009, No. 5, pp. 945–958.

[39] Zhao, S.-Z.—Suganthan, P. N.—Das, S.: Self-Adaptive Differential Evolution with Multi-Trajectory Search for Large-Scale Optimization. Soft Computing – A Fusion of Foundations, Methodologies and Applications, Vol. 15, 2011, No. 11, pp. 2175–2185.

Li Guo received his Ph.D. degree from the China University of Geosciences, Wuhan, China, in 2015. He is currently working at CCCC Infrastructure Maintenance Group Co., Ltd., China.

Xiang Li received the Ph.D. degree from the China University of Geosciences, Wuhan, China, in 2008. He is currently Associate Professor with School of Computer Science, China University of Geosciences. He has published over 30 research papers in journals and international conferences.

Wenyin Gong received his B.Eng., M.Eng. and Ph.D. degrees in computer science from the China University of Geosciences, Wuhan, China, in 2004, 2007, and 2010, respectively. He is Professor with the School of Computer Science, China University of Geosciences. He has published over 50 research papers in journals and international conferences. His current research interests include evolutionary algorithms, evolutionary optimization, and their applications. He served as a reviewer for over 20 international journals, such as the IEEE Transactions on Evolutionary Computation, the IEEE Transactions on Cybernetics, the IEEE Computational Intelligence Magazine, the ACM Transactions on Intelligent Systems and Technology, the Information Sciences, the European Journal of Operational Research, the Applied Soft Computing, and the International Journal of Hydrogen Energy.