Repairing the Crossover Rate in Adaptive Differential Evolution

Wenyin Gong[*,a], Zhihua Cai[a], Yang Wang[a]

[a]School of Computer Science,
China University of Geosciences, Wuhan 430074, P.R. China

Abstract

Differential evolution (DE) is a simple yet powerful evolutionary algorithm (EA) for global numerical optimization. However, its performance is significantly influenced by its parameters. Parameter adaptation has been proven to be an efficient way for the enhancement of the performance of the DE algorithm. Based on the analysis of the behavior of the crossover in DE, we find that the trial vector is directly related to its binary string, but not directly related to the crossover rate. Based on this inspiration, in this paper, we propose a crossover rate repair technique for the adaptive DE algorithms that are based on successful parameters. The crossover rate in DE is repaired by its corresponding binary string, *i.e.* by using the average number of components taken from the mutant. The average value of the binary string is used to replace the original crossover rate. To verify the effectiveness of the proposed technique, it is combined with an adaptive DE variant, JADE, which is a highly competitive DE variant. Experiments have been conducted on 25 functions presented in CEC-2005 competition. The results indicate that our proposed crossover rate technique is able to enhance the performance of JADE. In addition, compared with other DE variants and state-of-the-art EAs, the improved JADE method obtains better, or at least comparable, results in terms of the quality of final solutions and the convergence rate.

Key words: Differential evolution, parameter adaptation, crossover rate repair, binary string, numerical optimization

1. Introduction

Differential evolution (DE), proposed by Storn and Price in 1995 [1, 2], is a simple, efficient, and versatile population-based evolutionary algorithm (EA) for the global numerical optimization. The advantages are its simple structure, ease of use, speed, and robustness. Due to these advantages, DE has been successfully applied in diverse fields, such as data mining, pattern recognition, digital filter design, etc. [3, 4]. In addition, recent studies demonstrate the highly competitive performance provided by DE in constrained optimization problems, multi-objective optimization problems, and other complex problems. More details on the state-of-the-art research within DE can be found in two surveys [5] and [6] and the references therein.

There are three algorithmic parameters in the original DE algorithm, which are i) the population size NP; ii) the crossover rate CR; and iii) the scaling factor F. Originally, these parameters are user-specified and kept fixed during the run. However, recent studies indicate that the performance of DE is very sensitive to the parameter setting and the choice of the best parameters is always problem-dependent [7, 8, 9]. In order to obtain acceptable results, we need different parameter settings for different problems at hand. Even for the same problem, different parameters are required at different stages of evolution. Thus, some researchers investigated the parameter adaptation techniques to adaptively choose the parameters for the DE algorithm, such as jDE [9], SaDE [10], JADE [11], and so on. These adaptive DE variants obtained very promising results in the DE literature.

In this paper, we first analyze the behavior of the crossover operator. Then, we propose a crossover rate repair technique for the adaptive DE algorithm. The crossover rate in DE is repaired by its corresponding binary string, *i.e.* by using the average number of components taken from the mutant. As it will be explained in the following sections, we can see that the crossover rate repair technique is very simple. In order to evaluate the efficiency of our proposed technique, it is combined with an adaptive DE variant, JADE [11], which is a highly competitive DE variant. Experiments have been conducted on 25 benchmark functions presented in CEC-2005 competition [12] on real-parameter numerical optimization. In addition, the proposed crossover rate repair technique is also incorporated into SaDE [10] and EPSDE [13]. Experimental results indicate that this technique is able to enhance the performance of JADE, SaDE, and EPSDE in the test functions at $D = 30$ and $D = 50$. Moreover, compared with other DE variants and state-of-the-art EAs, the improved JADE method obtains better, or at least comparable, results in terms of the quality of final solutions and the convergence rate.

The rest of this paper is organized as follows. Section 2 briefly introduces the original DE algorithm and the related work. In Section 3 we present our proposed crossover rate repair technique in detail. In Section 4, we comprehensively evaluate the performance of our approach through different ex-

*Corresponding author. Tel: +86-27-67883716.
Email addresses: wenyingong@yahoo.com; wygong@cug.edu.cn (Wenyin Gong), zhcai@cug.edu.cn (Zhihua Cai)

periments. In this last section, Section 5, we conclude the work of this paper.

2. Related Work

In this section, we first briefly introduce the original DE algorithm. Then, the studies on the influence of crossover in DE are briefly introduced. Finally, the recently proposed adaptive DE variants in the literature are surveyed.

2.1. Differential Evolution

DE algorithm is initially proposed to solve numerical optimization problems. Without loss of generality, in this work, we consider the following numerical optimization problem:

$$\text{Minimize} \quad f(\mathbf{x}), \qquad \mathbf{x} \in \mathbb{R}^D, \tag{1}$$

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

$$L_j \le x_j \le U_j, \; j = 1, 2, \cdots, D. \tag{2}$$

where L_j and U_j are respectively the lower bound and upper bound of x_j.

2.1.1. Initialization

The DE population consists of NP vectors. Initially, the population is generated at random. For example, for the i-th vector $\mathbf{x}_i$ it is initialized as follows:

$$x_{i,j} = L_j + \text{rndreal}(0,1) \cdot \left(U_j - L_j\right) \tag{3}$$

where $i = 1, \cdots, NP$, $j = 1, \cdots, D$, and rndreal$(0, 1)$ is a uniformly distributed random real number in $(0, 1)$.

2.1.2. Mutation

After initialization, the mutation operation is applied to generate the mutant vector $\mathbf{v}_i$ for each target vector $\mathbf{x}_i$ in the current population. There are many mutation strategies available in the literature [3, 14, 11], the classical one is "DE/rand/1":

$$\mathbf{v}_i = \mathbf{x}_{r_1} + F \cdot (\mathbf{x}_{r_2} - \mathbf{x}_{r_3}) \tag{4}$$

where F is the mutation scaling factor, $r_1, r_2, r_3 \in \{1, \cdots, NP\}$ are mutually different integers randomly generated, and $r_1 \neq r_2 \neq r_3 \neq i$.

2.1.3. Crossover

In order to diversify the current population, following mutation, DE employs the crossover operator to produce the trial vector $\mathbf{u}_i$ between $\mathbf{x}_i$ and $\mathbf{v}_i$. The most commonly used operator is the *binomial* or *uniform* crossover performed on each component as follows:

$$u_{i,j} = \begin{cases} v_{i,j}, & \text{if } (\text{rndreal}(0,1) < CR \text{ or } j = j_{rand}) \\ x_{i,j}, & \text{otherwise} \end{cases} \tag{5}$$

where CR is the crossover rate and j_{rand} is a randomly generated integer within $[1, D]$. It is worth noting that there are other crossover operators in DE, such as the *exponential* crossover [3]. However, in this paper, we only focus on the binomial crossover mentioned above due to its promising performance obtained.

2.1.4. Selection

Finally, to keep the population size constant in the following generations, the selection operation is employed to determine whether the trial or the target vector survives to the next generation. In DE, the *one-to-one tournament selection* is used as follows:

$$\mathbf{x}_i = \begin{cases} \mathbf{u}_i, & \text{if } f(\mathbf{u}_i) \le f(\mathbf{x}_i) \\ \mathbf{x}_i, & \text{otherwise} \end{cases} \tag{6}$$

where $f(\mathbf{x})$ is the objective function to be optimized. For the sake of clarity, the pseudo-code of DE with "DE/rand/1/bin" is given in Algorithm 1, where rndint$(1, D)$ returns a uniformly distributed random integer number between 1 and D.

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

Generate the initial population
Evaluate the fitness for each individual
while the halting criterion is not satisfied **do**
  for $i = 1$ to NP **do**
    Select uniform randomly $r_1 \neq r_2 \neq r_3 \neq i$
    $j_{rand} = \text{rndint}(1, D)$
    for $j = 1$ to D **do**
      if rndreal$_j(0,1) < CR$ **or** j is equal to j_{rand} **then**
        $u_{i,j} = x_{r_1,j} + F \cdot (x_{r_2,j} - x_{r_3,j})$
      else
        $u_{i,j} = x_{i,j}$
      end if
    end for
  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

2.2. Influence of Crossover in DE

The crossover operator, which is designated to enhance the potential diversity of the population, plays an important role in DE. In the DE family of algorithms there are mainly two kinds of crossover methods: *binomial* and *exponential* [3]. Between the two crossover methods, there are two essential differences: i) the probability distribution of crossover length; and ii) the inheritance continuity [15]. In the binomial crossover, the relation between the probability distribution and its crossover rate CR is linear; while in the exponential crossover the relation is nonlinear [16, 17]. Through exponential crossover the trial vector gets a fraction of the mutant consecutively (in cyclic sense) while the inheritance by binomial crossover is non-consecutive [15].

In the DE literature, there are some studies that have examined the influence of crossover. In [16, 17], Zaharie analyzed the influence of the crossover operator and the crossover rate *CR* on the behavior of DE. The relation between mutation probability p_m and crossover rate *CR* is also theoretically analyzed for several variants of crossover in [16, 17]. Lin *et al.* presented theoretical analysis and comparative study of different crossover methods in DE to better understand the role of crossover [15]. They also designed two new crossover methods, namely consecutive binomial crossover and non-consecutive exponential crossover. In [15], the authors concluded that the choice of the proper crossover method and its associated parameters is dependent on the features of the problems.

The crossover rate *CR* is used to control which and how many components to be mutated in each element of the current population [17]. Low values of *CR* result in a small number of parameters to be changed in each generation, and hence, to make moves to be orthogonal to the current axes. On the other hand, high values of *CR* (near 1) cause moves at angles to the search space's axes [6, 18]. Rönkkönen *et al.* suggest that for separable problems $CR \leq 0.2$ was appropriate, while for non-separable problems $CR > 0.9$ was best [19]. In [20], the properties of the moves with different values of *CR* and their effects on DE's search behavior was studied. Montgomery and Chen analyzed the operation of DE at low and high crossover rates in [18]. DE with low values of $CR \leq 0.1$ is able to maintain a highly diverse population throughout its course, especially in complex landscapes. On the other hand, DE with high values of *CR* causes rapid convergence but loses the diversity so early [18]. The authors suggest that both low and high values of *CR* are able to produce effective moves: low values can conduct gradual and frequently successful exploration; while high values are capable of producing rapid improvements in solution quality and contraction of the search space [20, 18].

2.3. Parameter Adaptation in DE

As above-mentioned, there are three parameters (NP, CR, and F) in DE. The performance of DE is significantly influenced by the parameter settings, and the choice of the best parameters is difficult and problem-dependent [7, 8]. There are some empirical guidance for the parameter setting in the DE literature [2, 7, 3]. However, most of the claims are mutually countered and lack sufficient experimental justifications [6]. Therefore, in order to improve the performance of DE and make it use more easily, DE researchers investigate the parameter adaptation techniques to adaptively control the parameters of DE during the run.

In [21], the scaling factor F is controlled by a fitness-based adaptation, while the crossover rate *CR* is fixed to 0.5. Liu and Lampinen [8] proposed a Fuzzy Adaptive DE (FADE), which employs fuzzy logic controllers to adapt the mutation and crossover control parameters. Brest *et al*. [9] proposed self-adapting control parameter settings. Their proposed approach encodes the F and *CR* parameters into the chromosome and uses a self-adaptive control mechanism to change them. Salman *et al*. [22] proposed a self-adaptive DE (SDE) algorithm that eliminates the need for manual tuning of control parameters. In SDE, the mutation weighting factor F is self-adapted by a mutation strategy similar to the mutation operator of DE. Nobakhti and Wang [23] proposed a Randomized Adaptive Differential Evolution (RADE) method, where a simple randomized self-adaptive scheme was proposed for the mutation weighting factor F. Das *et al.* [24] proposed two variants of DE, DERSF and DETVSF, that use varying scale factors. They concluded that those variants outperform the original DE. Teo [25] presented a dynamic self-adaptive populations DE, where the population size is self-adapting. Through five De Jong's test functions, they showed that DE with self-adaptive populations produced highly competitive results. Brest and Maučec [26] proposed an improved DE method, where the population size is gradually reduced. Qin *et al.* [10] presented the SaDE algorithm, where both the mutation strategies and their associated crossover *CR* is adaptively controlled according to their previous successful experience; the scaling factor F is generated for each target vector as $F_i = N(0.5, 0.3)$, where $N(0.5, 0.3)$ is a normal distribution with mean value 0.5 and standard deviation 0.3. Zhang and Sanderson [11, 27] proposed an adaptive DE variant, namely JADE. In JADE, the parameters (*CR* and F) of DE are updated iteratively according to their previous successful experience. Recently, Ghosh *et al.* [28] proposed the FiADE algorithm, in which both *CR* and F are adapted based on the objective value of individuals in the DE population. As analyzed in [18] large values of *CR* are able to accelerate the convergence, Li *et al.* presented an improved JADE variant [29], where the power mean is employed to calculate the mean value to replace the arithmetic mean used in JADE [11]. In [30, 13], the authors presented an adaptive DE variant with ensemble of parameters and mutation strategies. The parameters are initially chosen from fixed pools. During the evolution process, if the trial vector is worse than its target vector, then they are updated randomly with new parameter values from the respective pools or from the successful combinations stored in the previous generations. Islam *et al.* [31] proposed an adaptive DE algorithm (called MDE_*p*BX). In MDE_*p*BX, the authors presented a novel mutation strategy ("DE/current-to-gr_best/1") and a new crossover strategy (namely "*p*BX" crossover). In addition, similar to the parameter adaptation used in [11], *CR* and F are updated iteratively according to their previous successful experience in MDE_*p*BX.

3. Crossover Rate Repair in Adaptive DE

As the above literature survey to the adaptive DE variants, we notice that there are some algorithms that update the parameters based on their previous successful experience in the last generations, such as SaDE [10], JADE [11], MDE_*p*BX [31]. The rationale of these parameter adaptation techniques is that "Better control parameter values tend to generate individuals that are more likely to survive and thus these values should be propagated" [11]. In this work, we mainly focus on the adaptive DE algorithm, and try to enhance its performance based on our proposed crossover rate repair technique, which will be presented in Section 3.2. In addition, combined the crossover rate repair method with JADE, the improved JADE variant, R_{cr}-JADE, is proposed in Section 3.3.

3.1. Motivations

3.1.1. Behavior of Binomial Crossover in DE

The most commonly used crossover operator is the *binomial* or *uniform* crossover (see (5)) in the DE algorithm. In order to analyze the behavior of the binomial crossover, we let $\mathbf{b}_i$ be a binary string generated for each target vector $\mathbf{x}_i$ as follows:

$$b_{i,j} = \begin{cases} 1, & \text{if } (\text{rndreal}(0,1) < CR \text{ or } j = j_{rand}) \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

Therefore, the binomial crossover of DE in (5) can be reformulated as

$$u_{i,j} = b_{i,j} \cdot v_{i,j} + (1 - b_{i,j}) \cdot x_{i,j} \tag{8}$$

where $i = 1, \cdots, NP$ and $j = 1, \cdots, D$. According to (7) and (8), we can see that the binary string $\mathbf{b}_i$ is stochastically related to CR; however, the trial vector $\mathbf{u}_i$ is *directly* related to its binary string $\mathbf{b}_i$, but not directly related to its crossover rate CR.

3.1.2. Adaptive DE Variants based on Successful Parameters

In the adaptive DE variants, let CR_i and F_i be the associated parameters of the target vector $\mathbf{x}_i$, in this context, we give two definitions:

Definition 1 (**Successful trial vector**). *In DE, if the trial vector $\mathbf{u}_i$ produced by its target vector $\mathbf{x}_i$ survives to the next generation according to* (6), *we say $\mathbf{u}_i$ is a successful trial vector.*

Definition 2 (**Successful parameters**). *The parameters CR_i and F_i for generating the successful trial vector $\mathbf{u}_i$ are called successful parameters.*

Algorithm 2 Adaptive DEs based on successful parameters

```
1:  Generate the initial population P(0) at random;
2:  Set the generation counter t ← 1;
3:  while The halting criteria are not satisfied do
4:      Calculate CR_i and F_i for each target vector with some
        distributions (such as Gaussian, Cauchy);
5:      Generate trial vector from the parents using mutation and
        crossover;
6:      Get the next population P(t + 1) by the DE selection operation;
7:      Save the successful CR_i and F_i in S_CR and S_F, respectively;
8:      Update the distribution parameters with S_CR and S_F;
9:      t ← t + 1
10: end while
```

Based on the above definitions, the pseudo-code of the adaptive DE variants based on successful parameters can be described in Algorithm 2. Note that in Algorithm 2 (also in Algorithm 3), S_{CR} is used to store the successful CR values, however, in different adaptive DE variants it may be applied in different manners. For example, in SaDE [10] S_{CR} saves the successful CR values in the previous few generations (learning period). While in JADE [11], S_{CR} saves the successful CR values only in the last generation. As mention above, in the adaptive DE algorithms, SaDE [10], JADE [11], and MDE_*p*BX [31] are the representative variants based on successful parameters in the previous generations. These three algorithms have obtained very promising results [10, 11, 31]. However, their performance might be influenced by the initial distribution parameters (*e.g.*, the initial mean value μ_{CR} and location factor μ_F in JADE).

For example, for JADE we set the initial $\mu_{CR} \in [0.1, 1.0]$ with step size by 0.1, and keep the initial $\mu_F = 0.5$[1]. JADE is used to minimize the sphere function (f_{01} in [32]) at $D = 30$ over 50 independent runs. The convergence curves and evolution trend of μ_{CR} of JADE are shown in Fig. 1(a) and 1(b), respectively. From Fig. 1(a), it can be seen that JADE with the initial $\mu_{CR} = 0.8, 0.9, 1.0$ values obtain similar results. For other initial μ_{CR} values, the results are significantly different. In addition, from Fig. 1(b), we see that the optimal μ_{CR} value is around 0.8 for the sphere function. However, if the initial μ_{CR} value in JADE is far away from 0.8 (*e.g.*, $\mu_{CR} = 0.3$), JADE is difficult to converge to the optimal μ_{CR} value, and hence, its performance is poor.

Algorithm 3 Procedure of crossover rate repair

1: Generate CR_i and F_i for each target vector $\mathbf{x}_i$;
2: Generate the mutant vector $\mathbf{v}_i$ by a specific DE mutation strategy;
3: Get the binary string $\mathbf{b}_i$:

$$b_{i,j} = \begin{cases} 1, & \text{if } (\text{rndreal}(0,1) < CR_i \text{ or } j = j_{rand}) \\ 0, & \text{otherwise} \end{cases}$$

4: Calculate the repaired crossover rate CR'_i using (9);
5: Obtain the trial vector $\mathbf{u}_i$ by (8);
6: Save the successful CR'_i and F_i in S_{CR} and S_F, respectively;
7: Update the distribution parameters with S_{CR} and S_F;

3.2. Crossover Rate Repair Technique

From (7), we know that the successful trial vector $\mathbf{u}_i$ is directly related to its binary string $\mathbf{b}_i$, but not directly related to its original crossover rate CR_i. In addition, in the adaptive DE variants based on successful parameters, the performance might be significantly influenced by the initial distribution parameters. Based on these considerations, in this work, we propose a crossover rate repair technique to enhance the adaptive DE methods that update CR and F based on successful parameters. The crossover rate is repaired by its corresponding binary string, *i.e.* by using the average number of components taken from the mutant. Suppose that CR'_i is the repaired crossover rate, it is calculated as

$$CR'_i = \frac{\sum_{j=1}^{D} b_{i,j}}{D} \tag{9}$$

[1] Since we only focus on repairing the crossover rate in this work, the initial $\mu_F = 0.5$ is adopted for all experiments.

where $\mathbf{b}_i$ is the binary string calculated in (7), $i = 1, \cdots, NP$, and $j = 1, \cdots, D$. The crossover rate is repaired after its binary string is generated in (7) based on CR_i. If the trial vector $\mathbf{u}_i$ is a successful vector, CR'_i will be stored in S_{CR}, instead of storing CR_i. The procedure of crossover rate repair technique in adaptive DE is shown in Algorithm 3. From Algorithm 3 we can see that this technique is very simple without adding any additional parameter.

Algorithm 4 R_{cr}-JADE: Crossover rate repaired JADE

Initialize the population $\mathbf{P}(0)$ at random;
Set $\mu_{CR} = 0.5, \mu_F = 0.5, \mathbf{A} = \phi, c = 0.1, p = 0.05, t = 1$;
while The halting criterion is not satisfied **do**
  $S_{CR} = \phi, S_F = \phi$;
  for $i = 1$ to NP **do**
    Generate CR_i and F_i with (14) and (16), respectively;
    Produce the mutant vector $\mathbf{v}_i$ with one of JADE mutation strategy as described in Appendix A.1;
    Get the binary string $\mathbf{b}_i$ as stated in Algorithm 3; $\Leftarrow$
    Calculate the repaired crossover rate CR'_i with (9); $\Leftarrow$
    for $j = 1$ to D **do**
      $u_{i,j} = b_{i,j} \cdot v_{i,j} + (1 - b_{i,j}) \cdot x_{i,j}$; $\Leftarrow$
    end for
  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**
      Update the archive $\mathbf{A}$ with the inferior solution $\mathbf{x}_i$;
      $CR'_i \to S_{CR}$;
      $F_i \to S_F$;
      Replace $\mathbf{x}_i$ with $\mathbf{u}_i$;
    end if
  end for
  Update the μ_{CR} and μ_F with (15) and (17), respectively;
  $t = t + 1$;
end while

After the crossover rate is repaired, we now use the repaired JADE to minimize the sphere function at $D = 30$. We also set the initial $\mu_{CR} \in [0.1, 1.0]$ with step size by 0.1, and keep the initial $\mu_F = 0.5$. The convergence curves and evolution trend of μ_{CR} of the repaired JADE are respectively shown in Fig. 1(d) and 1(e). From Fig. 1(e), it is clear that for all initial values the μ_{CR} can finally converges to the optimal value around 0.85 in the sphere function. Compared the convergence rate between JADE and the repaired JADE, Fig. 1(a) and 1(d) indicate that the repaired JADE converges faster than JADE, especially when the initial μ_{CR} is far away from the optimal value. The reason is that saving the repaired CR'_i is more reasonable than saving CR_i, since the trial vector is *directly* related to its binary string. In order to further explain it, we select a multimodal function, the Ackley function (f_{10} in [32]) at $D = 30$, to perform the same experiments with different initial μ_{CR} values. The results are plotted in Fig. 2. From Fig. 2 we can also observe the similar phenomenon like Fig. 1. The crossover rate repair technique is also able to improve the performance of JADE with different initial μ_{CR} vlaues for the Ackley function. Additionally, the enhanced performance of the repaired JADE algorithm will also be observed in Section 4.

3.3. R_{cr}-*JADE: Crossover Rate Repaired JADE*

By combining our proposed crossover rate repair technique with JADE[2], the repaired JADE algorithm is proposed, referred to as R_{cr}-JADE. The only difference between JADE and R_{cr}-JADE is that in R_{cr}-JADE the repaired crossover rate CR'_i is stored into S_{CR} if it can produce a successful trial vector; while in JADE the original CR_i is saved into S_{CR}. The pseudo-code of R_{cr}-JADE is illustrated in Algorithm 4. Modified steps with respect to JADE are marked with a left arrow "$\Leftarrow$". As analyzed in [27, pp. 52], in general, the overall complexity of JADE is $O(G \cdot NP \cdot D)$, where G is the maximal generations. Since our proposed R_{cr}-JADE does not increase the complexity of JADE at all, the overall complexity of R_{cr}-JADE is also $O(G \cdot NP \cdot D)$. Note that R_{cr}-JADE is only an illustration of combing the crossover rate repair technique with JADE, our proposed technique is also able to integrate into other adaptive DE variants based on successful parameters, such as SaDE [10] and EPSDE [13].

4. Experimental Results and Analysis

In order to verify the performance of our approach, we choose 25 benchmark functions presented in CEC-2005 competition [12] on real-parameter optimization as the test suite. The detailed description of these functions can be found in [12]. Briefly, they can be categorized into four groups:

- Unimodal functions: F01 - F05;
- Basic multimodal functions: F06 - F12;
- Expanded multimodal functions: F13 - F14;
- Hybrid composition functions: F15 - F25.

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 [33], *i.e.*, all of the compared algorithms are started from the same initial population in each out of 50 runs. The error value $f(\mathbf{x}) - f(\mathbf{x}^*)$ is recorded for the solution $\mathbf{x}$, where $\mathbf{x}^*$ is the global minimum of the function. The average and standard deviation of the error values over all independent runs are calculated. The results are compared using three nonparametric statistical hypothesis tests: i) the Friedman test (to obtain the final rankings of different algorithms for all functions); ii) Iman-Davenport test (to check the differences between all algorithms for all functions); and iii) the paired Wilcoxon signed-rank test at $\alpha = 0.05$ (to compare the significance between two algorithms in multi-problem and single-problem). The first two statistical tests and the multi-problem analysis by the Wilcoxon signed-rank test are calculated by the KEEL software tool [34]. When the Wilcoxon signed-rank test is applied to a single problem in all runs, the results are obtained by the OriginPro software, since in the KEEL software the values less than $5.0E - 11$ have been approximated to 0.

[2]The original JADE algorithm is briefly described in Appendix A. More details can be found in [11, 27].

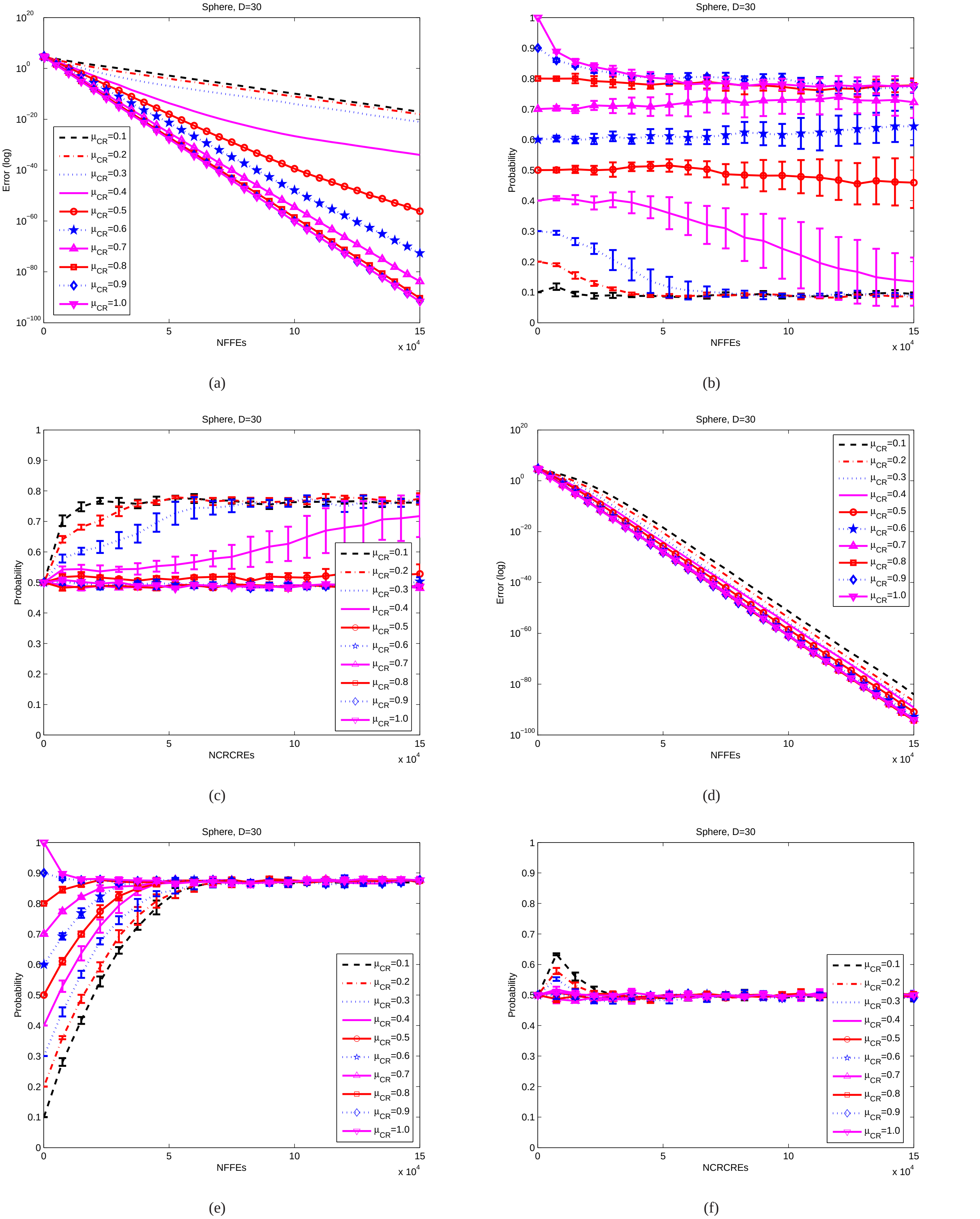

Figure 1: Convergence curves (1(a),1(d)), evolution trend of μ_{CR} (1(b),1(e)) and evolution trend of μ_F (1(c),1(f)) of JADE and R_{cr}-JADE in sphere function at $D = 30$ with different initial μ_{CR} values. (1(a),1(b),1(c)) for JADE; (1(d),1(e),1(f)) for R_{cr}-JADE.

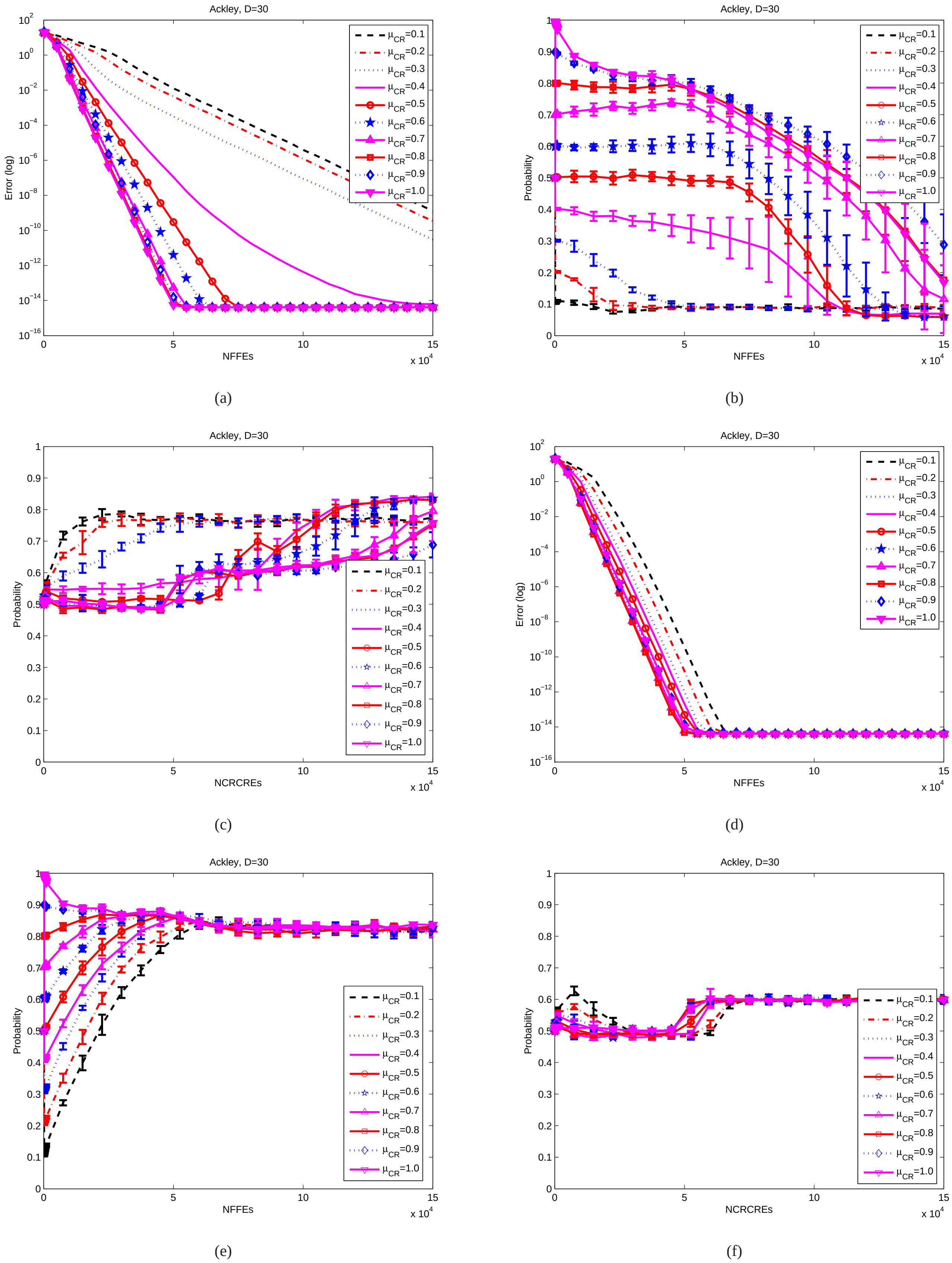

Figure 2: Convergence curves (2(a),2(d)), evolution trend of μ_{CR} (2(b),2(e)) and evolution trend of μ_F (2(c),2(f)) of JADE and R_{cr}-JADE in Ackley's function at $D = 30$ with different initial μ_{CR} values. (2(a),2(b),2(c)) for JADE; (2(d),2(e),2(f)) for R_{cr}-JADE.

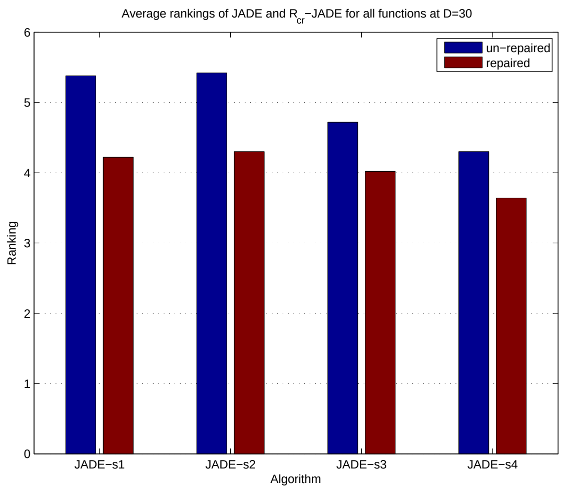

Figure 3: Average Rankings of JADE and R_{cr}-JADE variants (Friedman) for all functions at $D = 30$.

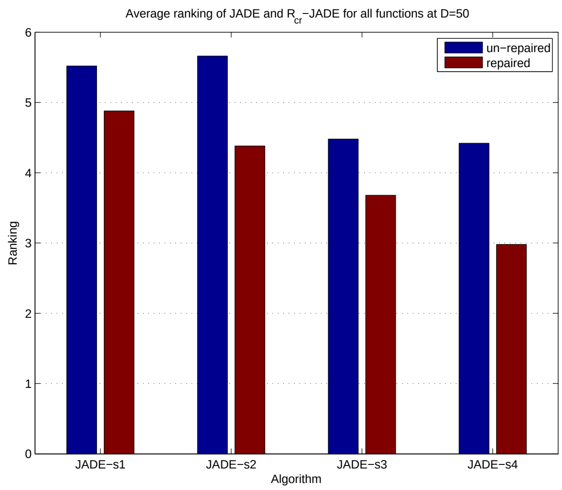

Figure 4: Average rankings of JADE and R_{cr}-JADE variants (Friedman) for all functions at $D = 50$.

4.1. Parameter Setting

In all experiments, we use the following parameters for JADE and R_{cr}-JADE unless a change is mentioned.

- Dimension of each function: $D = 30$ and $D = 50$;
- Population size: $NP = 100$ [11, 27];
- Initial distribution parameters: $\mu_{CR} = 0.5$ and $\mu_F = 0.5$ [11, 27];
- $c = 0.1$ and $p = 0.05$ [11, 27];
- Maximal number of fitness function evaluations (Max_NFFEs): Max_NFFEs = $D \times 10,000$ [12].

4.2. Comparison Among Different JADE Variants

At first, we need to evaluate the effectiveness of our proposed crossover rate repair technique for enhancing the original JADE algorithm. To address this issue, we compare JADE with R_{cr}-JADE for all test instances at $D = 30$ and $D = 50$. Since there are four mutation strategies in JADE [11, 27] (see Appendix A.1), there are four JADE and four R_{cr}-JADE variants based on each of the four mutation strategies. They are:

- JADE-s1 and R_{cr}-JADE-s1: based on "DE/current-to-*p*best/1 (without archive)";
- JADE-s2 and R_{cr}-JADE-s2: based on "DE/rand-to-*p*best/1 (without archive)";
- JADE-s3 and R_{cr}-JADE-s3: based on "DE/current-to-*p*best/1 (with archive)";
- JADE-s4 and R_{cr}-JADE-s4: based on "DE/rand-to-*p*best/1 (with archive)".

The error values of all JADE and R_{cr}-JADE algorithms are shown in Tables 1 and 2 for all functions at $D = 30$ and $D = 50$, respectively[3]. All results are averaged over 50 independent runs. The overall best and the second best results among the eight JADE variants are highlighted in **gray boldface** and **boldface**, respectively. In addition, according to the Wilcoxon's test, the results are summarized as "$w/t/l$", which means that R_{cr}-JADE wins in w functions, ties in t functions, and loses in l functions, compared with its corresponding JADE. Moreover, the final rankings of all JADE variants for all functions at $D = 30$ and $D = 50$ are plotted in Figures 3 and 4, respectively.

According to the error values in Tables 1 and 2, the p-values computed by Iman-Daveport test are $1.09E-01$ and $1.10E-03$ for all functions at $D = 30$ and $D = 50$, respectively. The results indicate that there are no significant differences between the compared algorithms for all functions at $D = 30$. However, when the dimension is scaled up to 50, the differences are significant between the compared algorithms for all functions at $\alpha = 0.05$. In addition, based on the Wilcoxon's test we can see that in the majority of the test functions R_{cr}-JADE performs significantly better than its corresponding JADE. For example, at $D = 30$ R_{cr}-JADE-s3 wins in 10 cases, ties in 14 cases, and only loses in 1 case, compared with JADE-s3. The only exception is for R_{cr}-JADE-s1 and JADE-s1 at $D = 50$, both algorithms obtain similar results in the most of functions (18 out of 25). R_{cr}-JADE-s1 only wins in 5 cases, but loses in 2 cases. The reason is that for the higher dimensional problems, "DE/current-to-*p*best/1" strategy used in the two algorithms does not provide sufficient diversity, and hence, the performance of both of them are poor (see the rankings in Figure 4). The insufficient diversity causes that R_{cr}-JADE-s1 only slightly improves JADE-s1 for higher dimensional problems.

[3]Note that we also test all JADE and R_{cr}-JADE variants for all functions at $D = 10$. Like $D = 30$ and $D = 50$, similar results can be observed, thus, we omit to present the results at $D = 10$ to save the space.

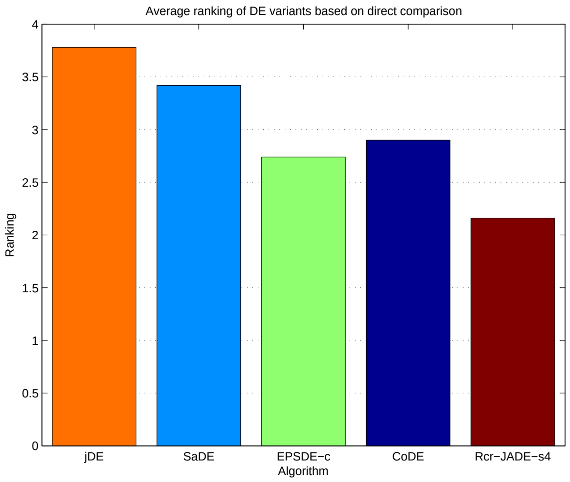

Figure 5: Average rankings of the state-of-the-art DE variants (Friedman) for all functions at $D = 30$, where the direct comparison is performed.

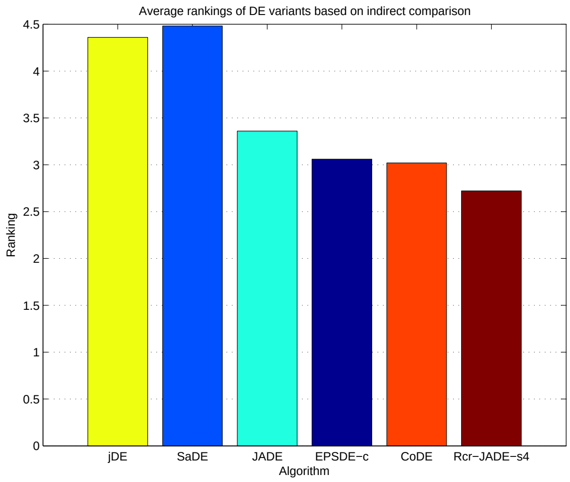

Figure 6: Average rankings of the state-of-the-art DE variants (Friedman) for all functions at $D = 30$, where the indirect comparison is performed.

With respect to the average rankings of all algorithms according to the Friedman test, the results are respectively shown in Figures 3 and 4 for all functions at $D = 30$ and $D = 50$. The lower the bar, the better ranking the algorithm obtains. It is clear that R_{cr}-JADE consistently ranks better than its corresponding JADE regardless of the dimensions of the test functions.

In general, from the above analysis of the results shown in Tables 1 - 2 and Figures 3 - 4, we can conclude that our proposed crossover rate repair technique is effective and it can enhance the performance of JADE. By carefully looking at the results presented in Figures 3 and 4, we see that R_{cr}-JADE-s4 obtains the overall best rankings. Therefore, in the following experiments, we only compare R_{cr}-JADE-s4 with other algorithms.

Table 4: Ranks Computed by the Wilcoxon Test for State-of-the-Art DE Variants on CEC-2005 Benchmark Functions at $D = 30$. • = the Method in the Row Improves the Method of the Column. ○ = the Method in the Column Improves the Method of the Row. Upper Diagonal of Level Significance $\alpha = 0.1$, Lower Diagonal Level of Significance $\alpha = 0.05$.

	(1)	(2)	(3)	(4)	(5)
jDE (1)	–	56.0	27.0○	30.0○	30.0○
SaDE (2)	115.0	–	23.0○	84.0	42.5○
EPSDE-c (3)	163.0•	167.0•	–	117.0	58.0
CoDE (4)	141.0•	69.0	73.0	–	64.0
R_{cr}-JADE-s4 (5)	180.0•	167.5•	132.0	146.0	–

4.3. Comparison With Other DE Variants

In this section, R_{cr}-JADE-s4 is compared with other state-of-the-art DE variants. Both the direct comparison and indirect comparison are presented to evaluate the performance of R_{cr}-JADE-s4.

4.3.1. Direct Comparison

First, R_{cr}-JADE-s4 is directly compared with four DE variants, which have obtained competitive results in the literature.

Table 6: Ranks Computed by the Wilcoxon Test for State-of-the-Art DE Variants on CEC-2005 Benchmark Functions at $D = 30$, Where the Indirect Comparison is Performed. • = the Method in the Row Improves the Method of the Column. ○ = the Method in the Column Improves the Method of the Row. Upper Diagonal of Level Significance $\alpha = 0.1$, Lower Diagonal Level of Significance $\alpha = 0.05$.

	(1)	(2)	(3)	(4)	(5)	(6)
jDE (1)	–	129.0	29.0○	69.0○	8.0○	30.0○
SaDE (2)	102.0	–	100.0	65.0○	69.5○	51.0○
JADE (3)	107.0•	153.0	–	122.0	60.0	79.0
EPSDE-c (4)	184.0•	235.0•	131.0	–	142.0	96.0
CoDE (5)	128.0•	183.5	76.0	111.0	–	68.0
R_{cr}-JADE-s4 (6)	180.0•	202.0•	152.0	157.0	122.0	–

The four DE variants are jDE [9], SaDE [10], EPSDE-c [30][4], and CoDE [35]. Since R_{cr}-JADE-s4 has been compared with JADE in the previous section, we do not compare them again. jDE is a self-adaptive DE algorithm, where the parameters *CR* and *F* are self-adaptively controlled during the evolution. In the other three DE algorithms, the ensemble of different mutation strategies is implemented. In addition, in SaDE and EPSDE-c the parameters are also adaptively updated. While in CoDE the parameters are randomly selected for each strategy in a specific pool. In order to make a fair comparison, for jDE, SaDE, EPSDE-c, and CoDE, all the parameters are set as the same used in their original literature. All algorithms are evaluated for all the functions at $D = 30$. The error values are shown in Table 3. Figure 5 shows the average rankings of the considered DE algorithms based on the Friedman test. In addition, due to the importance of the multiple-problem statistical analysis [36], we present the results of the multiple-problem Wilcoxon signed-rank test in Table 4, where "•" means that the method in the row improves the method of the column, and "○" means that the method in the column improves the method of the

[4] There are two versions of EPSDE in [30] and [13]. We refer to EPSDE-c and EPSDE-j for the conference and journal version, respectively.

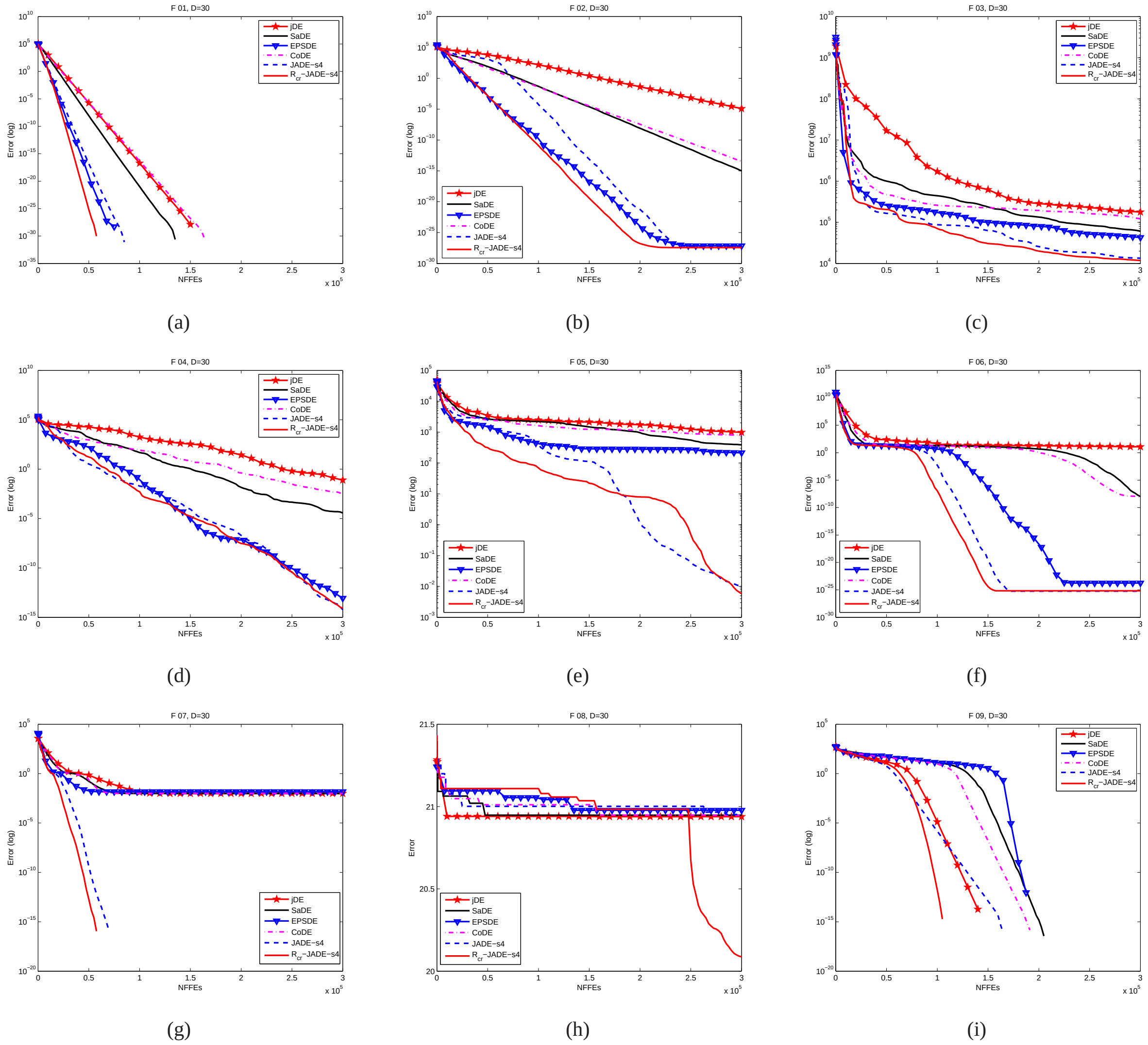

Figure 7: Convergence curves of different DE variants for the selected functions. (a) - (i): F01 - F09.

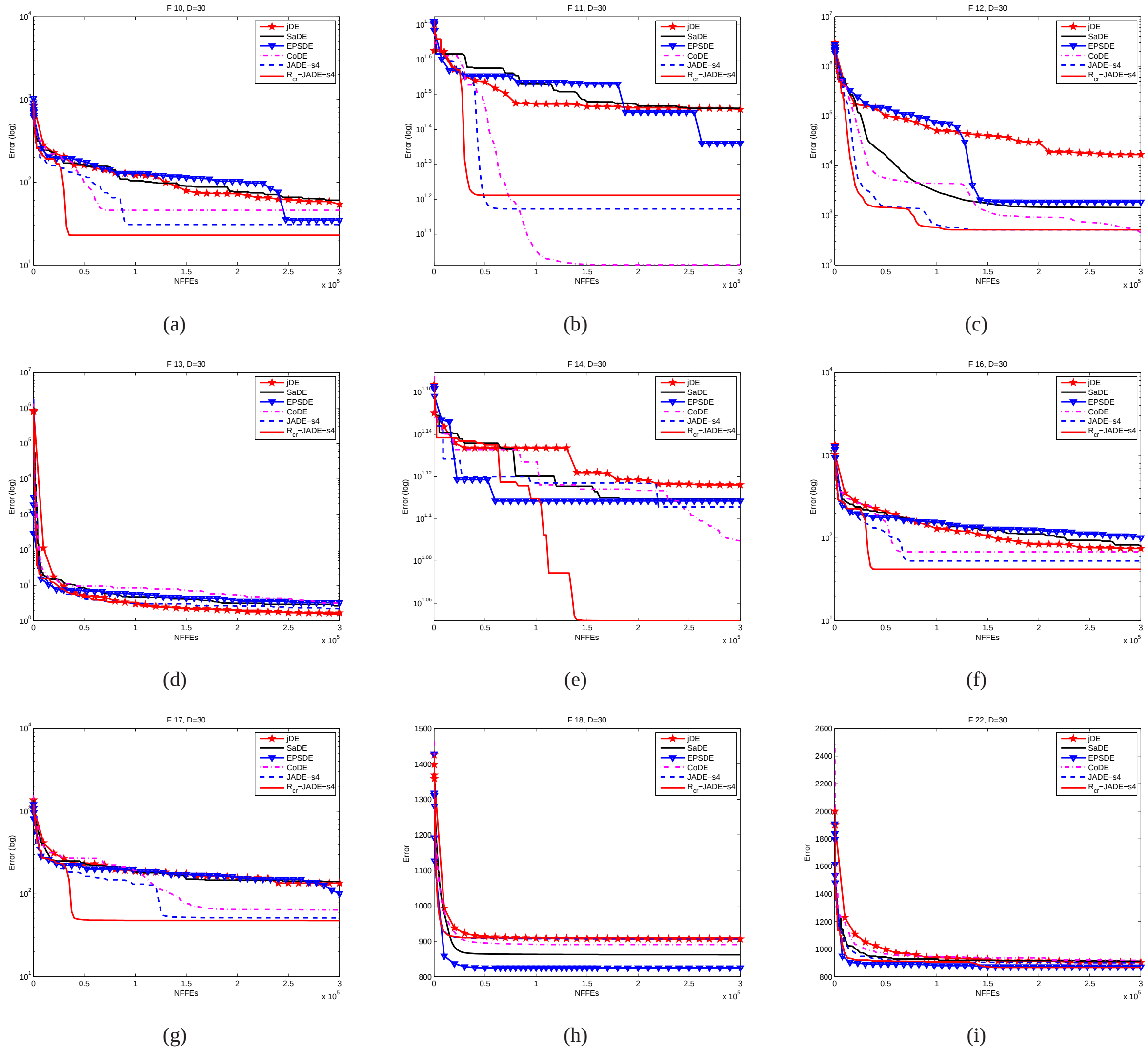

Figure 8: Convergence curves of different DE variants for the selected functions. (a) - (e): F10 - F14; (f) - (h): F16 - F18; (i): F22.

row. Upper diagonal of level significance $\alpha = 0.1$, and lower diagonal level of significance $\alpha = 0.05$. Furthermore, in order to compare the convergence rate among different algorithms, some representative convergence graphs of the DE algorithms are shown in Fig. 7 and 8. Note that the convergence graphs show the *median* error performance of the best solution over the total runs [12].

The p-value computed by Iman-Davenport test on the average error values shown in Table 3 is $2.37E-03$, which states that there are significant differences on the behavior of the compared DE algorithms for all the functions at $\alpha = 0.05$. From Table 3, we can observe that the proposed R_{cr}-JADE-s4 consistently provides the best error values in the majority of all test cases. R_{cr}-JADE-s4 significantly outperforms in 15, 17, 16, and 15 functions compared with jDE, SaDE, EPSDE-c, and CoDE, respectively. Additionally, in 14 out of 25 functions, R_{cr}-JADE-s4 obtains the best final results compared with other four DE algorithms.

According to the average rankings of all considered DE algorithms based on the Friedman test, Figure 5 shows that R_{cr}-JADE-s4 obtains the first ranking, followed by EPSDE-c, CoDE, SaDE, and jDE.

In Table 4, the multiple-problem Wilcoxon signed-rank test is applied based on the average error values shown in Table 3. The results in Table 4 are the positive ranks R^+ computed by the Wilcoxon signed-rank test when the algorithm in the row is compare with one in the column. R_{cr}-JADE-s4 obtains higher R^+ values than R^- in all cases, which means that R_{cr}-JADE-s4 is better than other compared DE variants for all functions.

With respect to the convergence rate, Fig. 7 and 8 show that R_{cr}-JADE-s4 converges fastest in most of the functions compared with other four DE algorithms.

4.3.2. Indirect Comparison

Since there are other DE variants that have conducted experiments on the CEC-2005 benchmark functions, in this section, we compare the results of R_{cr}-JADE-s4 with the reported results of other DE variants on the CEC-2005 benchmark functions at $D = 30$. R_{cr}-JADE-s4 is indirectly compared with jDE, SaDE, JADE, ESPDE-c, and CoDE. The results of jDE, SaDE, JADE, and CoDE are all obtained from Table I in [35]. The results of EPSDE-c are gotten from Table 2 in [30]. The error values are reported in Table 5. In the six DE variants, the best and the second best results are respectively highlighted in **gray boldface** and **boldface**. Averaged rankings obtained by each method in the Friedman test are shown in Figure 6. Also, the results of the multiple-problem Wilcoxon signed-rank test are tabulated in Table 6.

Table 5 shows that in 10 out of 25 cases R_{cr}-JADE-s4 provides the 1^{st} best error values, and in 9 functions it obtains the 2^{nd} best error values. According to Figure 6, we can see that R_{cr}-JADE-s4 gets the first ranking, followed by CoDE, EPSDE-c, JADE, jDE, and SaDE. In addition, Table 6 indicates that R_{cr}-JADE-s4 obtains higher R^+ values than R^- compared with other five DE variants, which means that R_{cr}-JADE-s4 is able to provide overall better results than other five compared DE variants for all functions. In general, based on the indirect comparison with other state-of-the-art DE variants, we can see that R_{cr}-JADE-s4 is still highly competitive.

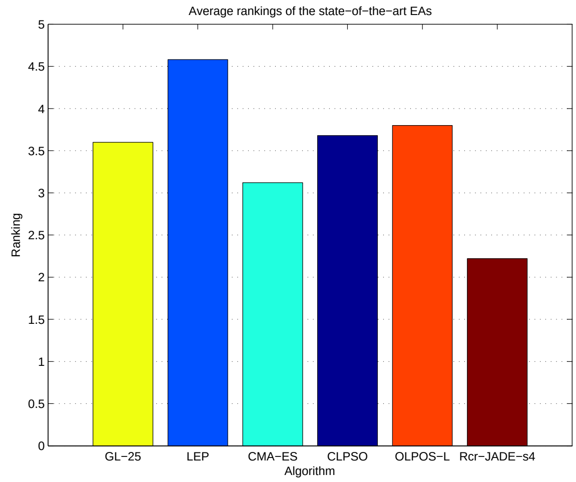

Figure 9: Average rankings of the state-of-the-art EAs (Friedman) for all functions at $D = 30$.

4.4. Comparison With State-of-the-Art EAs

In the previous experiments, R_{cr}-JADE-s4 is compared with other state-of-the-art DE variants in the literature. In this section, it is also compared with other state-of-the-art non-DE EAs: GL-25 [37], LEP [38], CMA-ES [39], CLPSO [40], and OLPSO-L [41]. GL-25, proposed by García-Martínez *et al.* [37], is a hybrid real-coded genetic algorithm based on parent-centric crossover operators. In [38], Lee and Yao proposed the LEP algorithm, which is an improved evolutionary programming based on the Lévy probability distribution. Hansen *et al.* [39] proposed CMA-ES, which is a very efficient evolution strategy for global numerical optimization. Actually, there are several variants of CMA-ES, such as restart CMA-ES proposed in [42]. In this work, we only use its basic version for comparison. CLPSO and OLPSO-L are two particle swarm optimization (PSO) algorithms, which obtain promising results in the PSO literature. CLPSO, proposed by Liang *et al.* [40], updates a particle's velocity using all other particles' historical best information. In OLPSO-L [41], Zhan *et al.* proposed an orthogonal learning strategy to discover more useful information between its historical best experience and its neighborhood's best experience. In [41], the authors presented two versions of OLPSO, *i.e.* OLPSO-G (based on global best experience) and OLPSO-L (based on local best experience). Since OLPSO-L obtains better results than OLPSO-G, it is selected for comparison.

In order to make a fair comparison, for GL-25, LEP, CMA-ES, CLPSO, and OLPSO-L, all the parameters are set as the same used in their original literature. All algorithms are evaluated for all the functions at $D = 30$ over 50 independent runs. Table 7 describes the error values of all compared algorithms.

Table 8: Ranks Computed by the Wilcoxon Test for State-of-the-Art EAs on CEC-2005 Benchmark Functions at $D = 30$. • = the Method in the Row Improves the Method of the Column. ∘ = the Method in the Column Improves the Method of the Row. Upper Diagonal of Level Significance $\alpha = 0.1$, Lower Diagonal Level of Significance $\alpha = 0.05$.

	(1)	(2)	(3)	(4)	(5)	(6)
GL-25 (1)	–	195.0•	137.5	144.0	156.0	35.0∘
LEP (2)	81.0∘	–	143.0	134.5	107.0	7.0∘
CMA-ES (3)	138.5	157.0	–	161.0	196.0	93.0
CLPSO (4)	109.0	190.5	139.0	–	121.5	34.0∘
OLPSO-L (5)	120.0	193.0	129.0	154.5	–	33.0∘
R_{cr}-JADE-s4 (6)	218.0•	293.0•	207.0	197.0•	220.0•	–

The average rankings of the considered EAs based on the Friedman test are shown in Figure 9. In addition, we also present the results of the multiple-problem Wilcoxon signed-rank test in Table 8.

According to Ivan-Davenport test, there are significant differences among the compared algorithms (the p-value obtained is $2.67E-04$). Compared with GL-25, LEP, CLPSO, and OLPSO-L, Table 7 shows that R_{cr}-JADE-s4 performs significantly better in 18, 24, 17, and 17 functions. Based on the multiple-problem Wilcoxon test shown in Table 8, the results also confirm that R_{cr}-JADE-s4 is significantly better than GL-25, LEP, CLPSO, and OLPSO-L with 95% confidence. Compared with CMA-ES, R_{cr}-JADE-s4 wins in 13 cases, ties in 3 cases, but loses in 9 cases. And the multiple-problem Wilcoxon test indicates that both of the two algorithm have no significant differences at $\alpha = 0.05$ and $\alpha = 0.1$. In Table 7, it also shows that R_{cr}-JADE-s4 obtains the best results in 12 out of 25 cases, and CMA-ES gets the best results in 9 out of 25 cases. However, R_{cr}-JADE-s4 obtains higher positive ranks ($R^+ = 207.0$) than that of CMA-ES ($R^+ = 93.0$). In addition, from Table 8, we can see that only R_{cr}-JADE-s4 is able to significantly outperform GL-25, LEP, CLPSO, and OLPSO-L; while there are no significant differences among CMA-EA, GL-25, LEP, CLPSO, and OLPSO-L for all test problems at $D = 30$. Moreover, from Figure 9, it is clear that R_{cr}-JADE-s4 ranks the first, followed by CMA-ES, GL-25, CLPSO, OLPSO-L, and LEP.

In summary, according to the results shown in Tables 7 - 8, we can conclude that our proposed R_{cr}-JADE-s4 is highly competitive to the above-mentioned state-of-the-art EAs. The results of R_{cr}-JADE-s4 are better than, or at least comparable to, those of the state-of-the-art EAs in terms of the quality of the final solutions.

4.5. Study on the Influence to Other Adaptive DE Variants

In the above sections, our proposed crossover rate repair technique is integrated into JADE, and the proposed R_{cr}-JADE is compared with other state-of-the-art DE and non-DE algorithms. The results demonstrate the superiority of our approach. Thus, we will be asked: "Can the proposed crossover rate repair technique be used to enhance other adaptive DE algorithms based on successful parameters?" To address this question, in this section, we integrate this technique into SaDE [10] and EPSDE-j [13][5], and the modified SaDE and EPSDE-j is respectively referred to as R_{cr}-SaDE and R_{cr}-EPSDE-j. It is worth noting that the crossover rate repair technique can also be combined with MDE_pBX [31]. However, since MDE_pBX employed the similar parameter adaptation method to JADE, we do not verify it again in this work.

4.5.1. Influence to SaDE

All the parameters are set as used in [10] for both SaDE and R_{cr}-SaDE. The error values of SaDE and R_{cr}-SaDE are given in Table 9 for all functions at $D = 30$ and $D = 50$. All results are averaged over 50 independent runs. The better results are highlighted in **boldface** compared between SaDE and R_{cr}-SaDE for $D = 30$ and $D = 50$, respectively.

With respect to $D = 30$, the p-value of the multi-problem analysis between SaDE and R_{cr}-SaDE by the Wilcoxon signed-rank test is $1.24E-03$, which leads to rejection of H0 at $\alpha = 0.05$. It indicates that there are significant differences between the two algorithms for all functions. From Table 9, we see that R_{cr}-SaDE is significantly better than SaDE in 10 out of 25 functions. In the rest 15 functions there are no significant differences between SaDE and R_{cr}-SaDE. Additionally, in 15 out of 25 functions, R_{cr}-SaDE obtains better error values than SaDE.

For all functions at $D = 50$, there are no significant differences between SaDE and R_{cr}-SaDE at $\alpha = 0.05$ (the p-value of the multi-problem analysis by the Wilcoxon signed-rank test is $5.51E-02$). However, according to Table 9, it can be seen that R_{cr}-SaDE significantly outperforms SaDE in 9 out of 25 functions. While there are no functions that SaDE obtains significantly better results compared with R_{cr}-SaDE. Table 9 also shows that in 13 cases R_{cr}-SaDE is better than SaDE; but only in 4 functions (F03, F07, F15, and F18) R_{cr}-SaDE is worse than SaDE.

It is worth pointing out that R_{cr}-SaDE improves SaDE significantly for all functions at $D = 30$; while the improvement of R_{cr}-SaDE is not significant at $D = 50$, according to the multi-problem analysis by the Wilcoxon signed-rank test at $\alpha = 0.05$. The reason might be that in R_{cr}-SaDE the mutation strategy "DE/current-to-rand/1" is selected in the strategy pool. This strategy, which is not controlled by the crossover operator, is a rotation-invariant strategy [14]. As stated in [3, pp. 101], *rotational invariance* is very important to obtain good performance for parameter-dependent problems. In the benchmark functions presented in CEC-2005 [12], most of them are rotated and parameter-dependent. Thus, for the rotated problems at $D = 50$, "DE/current-to-rand/1" maybe dominate other three strategies that are controlled by crossover operator during the evolution. As a result, the improvement of R_{cr}-SaDE is decreased for the problems at $D = 50$.

4.5.2. Influence to EPSDE-j

In EPSDE-j [13], if trial vector is better than its target vector,

[5] We do not integrate the crossover rate repair technique into EPSDE-c [30], since in the original implementation of EPSDE-c the successful parameters were not used.

the crossover rate associated with mutation strategy and scaling factor is retained with trial vector which becomes the target vector in the next generation. The successful parameter values and strategy are also saved in the archive. Otherwise, if trial vector is worse than its target vector, then the strategy and parameter values of the target vector will be reinitialized or chosen from the archive randomly. In this section, our proposed crossover rate repair technique is also used in EPSDE-j, and the repaired EPSDE-j (R_{cr}-EPSDE-j) is compared with EPSDE-j for all functions at $D = 30$ and $D = 50$. All parameters are kept the same as the original literature in [13]. The results, which are averaged over 50 independent runs, are shown in Table 10.

For the functions at $D = 30$, Table 10 describes that in 13 out of 25 functions R_{cr}-EPSDE-j obtains better error values compared with those of EPSDE-j. In 12 functions, R_{cr}-EPSDE-j is significantly better than EPSDE-j. R_{cr}-EPSDE-j loses in 7 functions. In the remaining 6 functions, the differences between the two algorithms are not significant. The p-value of the multi-problem analysis between EPSDE-j and R_{cr}-EPSDE-j by the Wilcoxon signed-rank test is $4.30E-02$. Thus, there are significant differences at $\alpha = 0.05$ between the two algorithms in all functions at $D = 30$.

When the dimensions are scaled to $D = 50$, the differences between EPSDE-j and R_{cr}-EPSDE-j are also significant in all functions, since the p-value is $2.54E-02$ based on the multi-problem analysis between the two algorithms by the Wilcoxon signed-rank test at $\alpha = 0.05$. From Table 10 it can be seen that in 14 out of 25 functions R_{cr}-EPSDE-j provides significantly better results. In addition, in the majority of the functions (18 out of 25), R_{cr}-EPSDE-j obtain better error values than those of EPSDE-j. Only in 5 functions (F04, F17, F19, F20, and F25), EPSDE-j is significantly better than R_{cr}-EPSDE-j.

In general, from the results shown in Tables 9 and 10 and the above analysis, it confirms that our proposed crossover rate repair technique is also able to enhance both of the performance of SaDE and EPSDE-j. Hence, we can except that this crossover rate repair technique can be similarly useful to the performance enhancement of other adaptive DE approaches, which update the crossover rate CR based on its successful experience.

4.6. Performance on Moderate-dimensional Problems

In order to better understand the performance of our approach, in this section, R_{cr}-JADE-s4 is compared with JADE-s4 on the moderate-dimensional problems at $D = 100$. Because functions F15 - F25 are too time-consuming, we only select functions F01 - F14 for comparison[6]. In [11], the population size $NP = 400$ is used for problems at $D = 100$. Therefore, we also set $NP = 400$ for both JADE-s4 and R_{cr}-JADE-s4. All other parameters are kept unchanged as mentioned in Section 4.1. The results are described in Table 11. Table 11 shows that in 6 out of 14 functions R_{cr}-JADE-s4 significantly outperforms JADE-s4 in terms of the error values. Only in function F13,

Table 11: Comparison on the Error Values Between JADE-s4 and R_{cr}-JADE-s4 for Functions F01 - F14 at $D = 100$.

Prob	JADE-s4		R_{cr}-JADE-s4
F01	0.00E+00 ± 0.00E+00	=	0.00E+00 ± 0.00E+00
F02	1.33E+05 ± 5.08E+04	+	**3.09E+04 ± 6.25E+04**
F03	**2.20E-06 ± 2.21E-06**	=	2.87E-06 ± 5.90E-06
F04	1.82E+05 ± 8.14E+04	+	**5.73E+04 ± 9.50E+04**
F05	2.27E+03 ± 4.36E+02	=	**2.13E+03 ± 4.13E+02**
F06	2.13E+01 ± 5.40E+00	=	**1.91E+01 ± 6.82E+00**
F07	3.45E+00 ± 6.95E-01	+	**1.61E+00 ± 1.71E-01**
F08	2.13E+01 ± 5.38E-02	=	2.13E+01 ± 3.74E-02
F09	5.27E-02 ± 1.19E-02	+	**4.55E-02 ± 2.84E-02**
F10	8.56E+01 ± 3.83E+01	=	**8.34E+01 ± 1.32E+01**
F11	1.21E+02 ± 1.70E+01	+	**7.65E+01 ± 2.47E+01**
F12	1.72E+05 ± 2.60E+05	+	**1.83E+04 ± 6.76E+04**
F13	**1.28E+01 ± 5.91E-01**	–	1.32E+01 ± 5.03E-01
F14	**4.62E+01 ± 4.42E-01**	=	4.63E+01 ± 6.73E-01
$w/t/l$	6/7/1		–

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

JADE-s4 is significantly better than R_{cr}-JADE-s4. There are no significant differences between R_{cr}-JADE-s4 and JADE-s4 in the rest 7 functions. In addition, the p-value of the multi-problem analysis between JADE-s4 and R_{cr}-JADE-s4 by the Wilcoxon signed-rank test is $1.22E-02$, which means that there are significant differences at $\alpha = 0.05$ between the two algorithms for functions F01 - F14 at $D = 100$. Thus, according to the results in Table 11, we can conclude that the crossover rate repair technique is also capable of enhancing the performance of JADE on the moderate-dimensional problems.

However, it is worth pointing out that the potential advantage of the crossover rate repair technique might be decrease when the problem size is large, especially for the large-scale problems, because the sample mean becomes closer to the real mean when the sample size increases. In our future work, we will evaluate the proposed crossover rate repair technique in the large-scale problems [43].

4.7. Parameter Study

In the previous experiments, we set the default parameter settings originally used in JADE [11, 27]. In [11], parameter study on c and p was conducted, and the recommended values are $1/c \in [5, 20]$ and $p \in [5\%, 20\%]$. In addition, the study on the effect of the initial μ_{CR} and μ_F values indicate that an initial setting of $\mu_{CR} = \mu_F = 0.5$ works well for a wide range of test functions [11]. In this section, we perform the parameter study on the population size NP and the initial μ_F value to investigate the enhanced performance of R_{cr}-JADE. Note that in this study we do not try to find the optimal values for NP and μ_F, but to verify the improved performance obtained after integrating the crossover rate repairing technique into JADE.

4.7.1. Influence of Population Size

To study the influence of the population size to the performance of R_{cr}-JADE and JADE, in this section, R_{cr}-JADE-s4 is compared with JADE-s4 for all functions at $D = 30$. The population size $NP = 50$ and $NP = 200$ are used. All other

[6] Although all functions are originally defined up to $D = 50$ in CEC-2005, it is easy to make some changes to scale them to $D = 100$.

parameters are kept the same as mentioned in Section 4.1. The results are tabulated in Table 12. All results are averaged over 50 independent runs.

When $NP = 50$, the results in Table 12 shows that in 9 out of 25 functions R$_{cr}$-JADE-s4 improves JADE-s4 significantly in terms of the error values. However, in 6 functions (F10, F13, F16, F18, F19 and F20), R$_{cr}$-JADE-s4 is statistically worse than JADE-s4. In 10 functions, both of them obtains similar results. R$_{cr}$-JADE-s4 is able to obtain higher R^+ value ($157.0 > 53.0$). According to the multi-problem analysis between the two algorithms by the Wilcoxon signed-rank test, the p-value is $5.20E-2$, which means that the differences between R$_{cr}$-JADE-s4 and JADE-s4 are not significant at $\alpha = 0.05$ in all functions. The reason might be that the small population size is not sufficient to R$_{cr}$-JADE-s4 and JADE-s4 in the majority of the test functions at $D = 30$.

When $NP = 200$, the p-value of the multi-problem analysis between the two algorithms by the Wilcoxon signed-rank test is $3.05E-4$, which leads to rejection of H0 at $\alpha = 0.05$. It indicates that there are significant differences between R$_{cr}$-JADE-s4 and JADE-s4 in all functions. In 13 functions, R$_{cr}$-JADE-s4 is significantly better than JADE-s4. R$_{cr}$-JADE-s4 only loses in function F13. In the rest 11 functions, there are not significant differences between the two algorithms.

Although there are no significant differences between R$_{cr}$-JADE-s4 and JADE-s4 with $NP = 50$ in all functions, however, in general, the population size does not influence the enhanced performance compared between R$_{cr}$-JADE-s4 and JADE-s4. With different population size ($NP = 50, 100$, and 200), R$_{cr}$-JADE-s4 consistently obtains better results in the majority of test functions.

Table 14: Statistical Results Between R$_{cr}$-JADE-s4 and JADE-s4 (R$_{cr}$-JADE-s4 vs JADE-s4) by the Wilcoxon Signed-rank Test for All Functions with Different Initial μ_F Values. The **Boldface** and *Italic* of the p-value Indicate that the Differences Are Significant at $\alpha = 0.05$ and $\alpha = 0.1$, Respectively.

μ_F	R^+	R^-	p-value	$w/t/l$
0.1	173.0	37.0	**9.13E-03**	6/18/1
0.2	146.0	25.0	**6.58E-03**	9/14/2
0.3	129.5	41.5	*5.50E-02*	8/15/2
0.4	167.0	23.0	**2.31E-03**	11/13/1
0.5	113.0	77.0	≥ 0.2	8/17/0
0.6	138.0	15.0	**2.09E-03**	12/12/1
0.7	151.0	39.0	**2.23E-02**	12/12/1
0.8	160.0	30.0	**7.11E-03**	16/8/1
0.9	161.0	29.0	**5.99E-03**	15/7/3

4.7.2. Influence of Initial μ_F Value

In the previous experiments the recommended initial $\mu_F = 0.5$ value is used. In order to test the influence of different initial μ_F values to the enhanced performance of R$_{cr}$-JADE, in this section, R$_{cr}$-JADE-s4 is compared with JADE-s4 with different initial μ_F values. The initial μ_F value is set to $\mu_F = \{0.1, 0.2, 0.3, 0.4, 0.6, 0.7, 0.8, 0.9\}$. All other parameters do not changed as described in Section 4.1. Due to the space limitation, we only give the results of R$_{cr}$-JADE-s4 and JADE-s4 with initial $\mu_F = 0.1, 0.6, 0.9$ in Table 13. In addition, the statistical results between the two algorithms by the Wilcoxon signed-rank test with all initial μ_F values are shown in Table 14.

From Table 13, we can see that R$_{cr}$-JADE-s4 consistently provides the better error values than those of JADE with different initial μ_F values in the majority of test functions. R$_{cr}$-JADE-s4 obtains better error values in 14, 14, and 16 out of 25 functions with the initial $\mu_F = 0.1, 0.6$, and 0.9, respectively. When the initial $\mu_F = 0.1$, in 6 functions R$_{cr}$-JADE-s4 is significantly better than JADE-s4. R$_{cr}$-JADE-s4 only loses in function F13. For the initial $\mu_F = 0.6$, R$_{cr}$-JADE-s4 provides significantly better results than JADE-s4 in 12 functions, but only loses in 1 function. With respect to the initial $\mu_F = 0.9$, in 15 out of 25 functions R$_{cr}$-JADE-s4 significantly improves the error values compared with JADE-s4. In three functions (F12, F13, and F15), JADE-s4 obtains statistically better results than R$_{cr}$-JADE-s4.

In addition, Table 14 shows that in all cases R$_{cr}$-JADE-s4 obtains higher R^+ values, which means that R$_{cr}$-JADE-s4 is overall better than JADE-s4 in terms of the error values in all functions based on the multi-problem analysis. Moreover, in the cases of the initial $\mu_F = 0.1, 0.2, 0.4, 0.6, 0.7, 0.8$, and 0.9, the differences are significant in all functions according to the multi-problem analysis between the two algorithms by the Wilcoxon signed-rank test at $\alpha = 0.05$. For the initial $\mu_F = 0.3$, there are significant differences between R$_{cr}$-JADE-s4 and JADE-s4 by the Wilcoxon signed-rank test at $\alpha = 0.1$.

In general, from the results in Tables 13 and 14 and the above analysis, we can conclude that the proposed crossover rate repair technique is consistently capable of improving the performance of the original JADE algorithm with different initial μ_F values.

4.8. Real-World Applications

According to benchmark functions we see that R$_{cr}$-JADE obtains highly competitive results with other DE and non-DE algorithms. In this section, R$_{cr}$-JADE-s4 is also evaluated in 5 real-world problems to test its capability of solving real-world problems. R$_{cr}$-JADE-s4 is compared with jDE, SaDE, CoDE, and JADE-s4. The five real-world problems are: P1) Chebychev polynomial fitting problem ($D = 9$) [3]; P2) frequency modulation sound parameter identification ($D = 6$) [37]; P3) spread spectrum radar poly-phase code design problem ($D = 20$) [44]; P4) systems of linear equations problem ($D = 10$) [37]; and P5) circular antenna array design problem ($D = 12$) [45]. In the five problems, P2, P3, and P5 are also appeared in CEC-2011 competition on real-world numerical optimization problems [46].

For all algorithms we use the same parameter settings as in Section 4.3. The Max_NEEFs are 150,000 for all problems. All results are averaged over 50 runs. The results are described in Table 15. The *intermediate* results are also reported for the functions where several algorithms can obtain the global optimum of these problems. According to the results in Table 15, we see that R$_{cr}$-JADE-s4 still provides highly competitive results compared with other DE variants. It obtains the best results in 4 (P1 - P4) out of 5 problems. In these four problems, R$_{cr}$-JADE-s4 is significantly better than jDE, SaDE, CoDE, and

JADE-s4. In P5, R_{cr}-JADE-s4 is worse than SaDE and jDE, but better than CoDE and JADE-s4.

5. Conclusions and Future Work

With the aim of enhancing the performance of adaptive DE algorithms based on successful parameters, in this paper, we propose a very simple technique for repairing the crossover rate according to its corresponding binary string, *i.e.* by using the average number of components taken from the mutant. Furthermore, this crossover rate repair technique does not add any additional parameter when integrating into adaptive DE algorithms. In order to evaluate the effectiveness of our proposed technique, it is integrated into two representative adaptive DE variants, *i.e.* JADE, SaDE, and EPSDE. Experimental results demonstrate that the proposed crossover rate repair technique is capable of enhancing the performance of JADE and SaDE. Moreover, compared with other state-of-the-art DE and non-DE approaches, one of the improved JADE (R_{cr}-JADE-s4) obtains better, or at least comparable, results in terms of the quality of final solutions and the convergence speed. In addition, extensive experiments on the influence of moderate-dimensional problems, different population size, and different initial μ_F values indicate that crossover rate repair technique consistently enhances the performance of the original JADE algorithm.

Ensemble of multiple strategies is able to to improve the performance of DE [10, 13, 35, 47], we will try to incorporate the repaired JADE into multiple-strategy DE variants in our future work.

Large-scale optimization has been one of the most interesting trends in recent years [43], some DE variants have obtained promising results (see [48, 49, 50]). Thus, another future direction is that the repaired JADE algorithm will be combined with cooperative coevolution [51, 52] or other local search techniques for the large-scale continuous optimization problems.

Funding

This work was partly supported by the National Natural Science Foundation of China under Grant Nos. 61203307, 61075063, and 61375066, the Fundamental Research Funds for the Central Universities at China University of Geosciences (Wuhan) under Grant Nos. CUG130413 and CUG090109, and the Research Fund for the Doctoral Program of Higher Education under Grant No. 20110145120009.

Acknowledgments

The authors would like to thank the Associate Editor and the anonymous reviewers for their great comments. They also thank Dr. Y. Wang to make the source codes of CoDE, CMA-ES, EPSDE-j, and CLPSO available online. The source codes of SaDE and EPSDE-c are appreciatively provided by Dr. P. N. Suganthan. They are also grateful to Dr. J. Brest, Dr. C. García-Martínez, and Dr. J. He for providing the source codes of jDE, GL-25, and LEP, respectively. The first author also thanks Dr. C. Li and Mr. Y. Cai for their constructive suggestions to this paper.

A. The JADE Algorithm

Since this work is mainly based on the JADE algorithm [11, 27], for the sake of completeness, the original JADE algorithm is briefly described herein. There are three main contributions in JADE: i) the modified mutation strategies based on the *p*best vector; ii) adaptation of the crossover rate; and iii) adaptation of the scaling factor.

A.1. Modified Mutation Strategies

In [11] and [27], the authors presented four modified "DE/current-to-best/1" and "DE/rand-to-best/1" strategies as follows:

1) "DE/current-to-*p*best/1 (without archive)":

$$\mathbf{v}_i = \mathbf{x}_i + F_i \cdot (\mathbf{x}^p_{best} - \mathbf{x}_i) + F_i \cdot (\mathbf{x}_{r_2} - \mathbf{x}_{r_3}) \tag{10}$$

2) "DE/rand-to-*p*best/1 (without archive)":

$$\mathbf{v}_i = \mathbf{x}_{r_1} + F_i \cdot (\mathbf{x}^p_{best} - \mathbf{x}_{r_1}) + F_i \cdot (\mathbf{x}_{r_2} - \mathbf{x}_{r_3}) \tag{11}$$

3) "DE/current-to-*p*best/1 (with archive)":

$$\mathbf{v}_i = \mathbf{x}_i + F_i \cdot (\mathbf{x}^p_{best} - \mathbf{x}_i) + F_i \cdot (\mathbf{x}_{r_2} - \tilde{\mathbf{x}}_{r_3}) \tag{12}$$

4) "DE/rand-to-*p*best/1 (with archive)":

$$\mathbf{v}_i = \mathbf{x}_{r_1} + F_i \cdot (\mathbf{x}^p_{best} - \mathbf{x}_{r_1}) + F_i \cdot (\mathbf{x}_{r_2} - \tilde{\mathbf{x}}_{r_3}) \tag{13}$$

In the latter two strategies in (12) and (13), an archive $\mathbf{A}$ is used to store the inferior solutions recently explored in the evolutionary search. $\mathbf{x}^p_{best}$ refers to the *p*best solution, which is randomly selected from the top $100p\%$ solutions, with $p \in (0, 1]$. $\mathbf{x}_i$, $\mathbf{x}_{r_2}$, and $\mathbf{x}^p_{best}$ are chosen from the current population $\mathbf{P}$; $\tilde{\mathbf{x}}_{r_3}$ is randomly chosen from the union between the archive and current populations ($\mathbf{P} \cup \mathbf{A}$).

A.2. Adaptation of the Crossover Rate

In JADE, for each target vector $\mathbf{x}_i$, the crossover rate CR_i is independently generated at each generation:

$$CR_i = \text{rndn}_i(\mu_{CR}, 0.1) \tag{14}$$

and truncated to the interval $[0, 1]$. In (14), $\text{rndn}_i(\mu_{CR}, 0.1)$ is a normal distribution with mean value μ_{CR} and standard deviation 0.1. The μ_{CR} is initially set to 0.5 and updated as

$$\mu_{CR} = (1 - c) \cdot \mu_{CR} + c \cdot \text{mean}_A(S_{CR}) \tag{15}$$

where c is a constant in $[0, 1]$; $\text{mean}_A(\cdot)$ is the usual arithmetic mean operation; and S_{CR} is the set of all successful crossover rates CR_i in the current generation.

A.3. Adaptation of the Scaling Factor

Similar to the adaptation of the crossover rate, at each generation, the scaling factor F_i is independently calculated for each target vector $\mathbf{x}_i$ as follows:

$$F_i = \text{rndc}_i(\mu_F, 0.1) \tag{16}$$

and then truncated to be 1.0 if $F_i > 1.0$ or regenerated if $F_i \leq 0$. $\text{rndc}_i(\mu_F, 0.1)$ is a random number generated according to the Cauchy distribution with location parameter μ_F and scale parameter 0.1. The location parameter μ_F is updated in the following manner:

$$\mu_F = (1 - c) \cdot \mu_F + c \cdot \text{mean}_L(S_F) \tag{17}$$

where S_F is the set of all successful mutation factors F_i in the current generation; and $\text{mean}_L(\cdot)$ is the Lehmer mean:

$$\text{mean}_L(S_F) = \frac{\sum_{i=1}^{|S_F|} F_i^2}{\sum_{i=1}^{|S_F|} F_i} \tag{18}$$

References

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

[2] R. Storn, K. Price, Differential evolution–A simple and efficient heuristic for global optimization over continuous spaces, J. of Global Optim. 11 (4) (1997) 341–359.

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

[4] S. Das, A. Abraham, A. Konar, Automatic clustering using an improved differential evolution algorithm, IEEE Trans. on Syst. Man, Cybern. A, Syst., Humans 38 (1) (2008) 218–237.

[5] F. Neri, V. Tirronen, Recent advances in differential evolution: A survey and experimental analysis, Artificial Intelligence Review 33 (1-2) (2010) 61 – 106.

[6] S. Das, P. N. Suganthan, Differential evolution: A survey of the state-of-the-art, IEEE Trans. on Evol. Comput. 15 (1) (2011) 4–31. doi:10.1109/TEVC.2010.2059031.

[7] R. Gämperle, S. Müler, P. Koumoutsakos, A parameter study for differential evolution, in: Proc. WSEAS Int. Conf. Advances Intell. Syst., Fuzzy Syst., Evol. Comput., 2002, pp. 293–298.

[8] J. Liu, J. Lampinen, A fuzzy adaptive differential evolution algorithm, Soft Comput. 9 (6) (2005) 448–462.

[9] J. Brest, S. Greiner, B. Bošković, M. Mernik, V. Žumer, Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems, IEEE Trans. on Evol. Comput. 10 (6) (2006) 646–657.

[10] A. K. Qin, V. L. Huang, P. N. Suganthan, Differential evolution algorithm with strategy adaptation for global numerical optimization, IEEE Trans. on Evol. Comput. 13 (2) (2009) 398–417.

[11] J. Zhang, A. C. Sanderson, JADE: Adaptive differential evolution with optional external archive, IEEE Trans. on Evol. Comput. 13 (5) (2009) 945–958.

[12] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y.-P. Chen, A. Auger, S. Tiwari, Problem definitions and evaluation criteria for the CEC_2005 special session on real-parameter optimization (2005).
URL `http://www.ntu.edu.sg/home/EPNSugan`

[13] R. Mallipeddi, P. Suganthan, Q. Pan, M. Tasgetiren, Differential evolution algorithm with ensemble of parameters and mutation strategies, Applied Soft Computing 11 (2) (2011) 1679–1696.

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

[15] C. Lin, A. Qing, Q. Feng, A comparative study of crossover in differential evolution, Journal of Heuristics 17 (2011) 675–703.

[16] D. Zaharie, A comparative analysis of crossover variants in differential evolution, in: Proceedings of the International Multiconference on Computer Science and Information Technology, 2007, p. 171C181.

[17] D. Zaharie, Influence of crossover on the behavior of differential evolution algorithms, Applied Soft Computing 9 (3) (2009) 1126 – 1138.

[18] J. Montgomery, S. Chen, An analysis of the operation of differential evolution at high and low crossover rates, in: Evolutionary Computation (CEC), 2010 IEEE Congress on, 2010, pp. 1–8.

[19] J. Rönkkönen, S. Kukkonen, K. Price, Real-parameter optimization with differential evolution, in: Evolutionary Computation, 2005. The 2005 IEEE Congress on, Vol. 1, 2005, pp. 506–513.

[20] J. Montgomery, Crossover and the different faces of differential evolution searches, in: Evolutionary Computation (CEC), 2010 IEEE Congress on, 2010, pp. 1–8.

[21] M. Ali, A. Törn, Population set-based global optimization algorithms: some modifications and numerical studies, Computers & Operations Research 31 (10) (2004) 1703 – 1725.

[22] A. Salman, A. P. Engelbrecht, M. G. H. Omran, Empirical analysis of self-adaptive differential evolution, European Journal of Operational Research 183 (2) (2007) 785–804.

[23] A. Nobakhti, H. Wang, A simple self-adaptive differential evolution algorithm with application on the alstom gasifier, Appl. Soft Comput. 8 (1) (2008) 350–370.

[24] S. Das, A. Konar, U. K. Chakraborty, Two improved differential evolution schemes for faster global search, in: H.-G. Beyer, U.-M. O'Reilly (Eds.), Proc. Genetic Evol. Comput. Conf., June 25-29, 2005, 2005, pp. 991–998.

[25] J. Teo, Exploring dynamic self-adaptive populations in differential evolution, Soft Comput. 10 (8) (2006) 673–686.

[26] J. Brest, M. S. Maučec, Population size reduction for the differential evolution algorithm, Applied Intelligence 29 (3) (2008) 228–247.

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

[28] A. Ghosh, S. Das, A. Chowdhury, R. Giri, An improved differential evolution algorithm with fitness-based adaptation of the control parameters, Information Sciences 181 (18) (2011) 3749 – 3765.

[29] J. Li, W. Zhu, M. Zhou, H. Wang, Power mean based crossover rate adaptive differential evolution, in: H. Deng, D. Miao, J. Lei, F. Wang (Eds.), Artificial Intelligence and Computational Intelligence, Vol. 7003 of Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2011, pp. 34–41.

[30] R. Mallipeddi, P. N. Suganthan, Differential evolution algorithm with ensemble of parameters and mutation and crossover strategies, in: Swarm, Evolutionary, and Memetic Computing, Vol. LNCS 6466, 2010, pp. 71–78.

[31] S. M. Islam, S. Das, S. Ghosh, S. Roy, P. N. Suganthan, An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 42 (2) (2012) 482 – 500.

[32] X. Yao, Y. Liu, G. Lin, Evolutionary programming made faster, IEEE Trans. on Evol. Comput. 3 (2) (1999) 82–102.

[33] N. Noman, H. Iba, Accelerating differential evolution using an adaptive local search, IEEE Trans. on Evol. Comput. 12 (1) (2008) 107–125.

[34] J. Alcalá-Fdez, L. Sánchez, S. García, KEEL: A software tool to assess evolutionary algorithms to data mining problems (2012).
URL `http://www.keel.es/`

[35] Y. Wang, Z. Cai, Q. Zhang, Differential evolution with composite trial vector generation strategies and control parameters, IEEE Trans. on Evol. Comput. 15 (1) (2011) 55–66.

[36] S. García, D. Molina, M. Lozano, 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 15 (6) (2009) 617 – 644.

[37] C. García-Martínez, M. Lozano, F. Herrera, D. Molina, A. M. Sánchez, Global and local real-coded genetic algorithms based on parent-centric crossover operators, European Journal of Operational Research 185 (3) (2008) 1088–1113.

[38] C.-Y. Lee, X. Yao, Evolutionary programming using mutations based on the Lévy probability distribution, IEEE Trans. on Evol. Comput. 8 (1)

(2004) 1–13.
[39] N. Hansen, S. D. Müller, P. Koumoutsakos, Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES), Evolutionary Computation 11 (1) (2003) 1 – 18.
[40] J. J. Liang, A. K. Qin, P. N. Suganthan, S. Baskar, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, IEEE Trans. on Evol. Comput. 10 (3) (2006) 281–295.
[41] Z.-H. Zhan, J. Zhang, Y. Li, Y.-H. Shi, Orthogonal learning particle swarm optimization, IEEE Transactions on Evolutionary Computation 15 (6) (2011) 832 – 847.
[42] A. Auger, N. Hansen, A restart CMA evolution strategy with increasing population size, in: IEEE Congr. on Evol. Comput. (CEC2005), Vol. 2, 2005, pp. 1769–1776.
[43] M. Lozano, D. Molina, F. Herrera, Editorial scalability of evolutionary algorithms and other metaheuristics for large-scale continuous optimization problems, Soft Computing - A Fusion of Foundations, Methodologies and Applications 15 (2011) 2085–2087.
[44] S. Das, A. Abraham, U. K. Chakraborty, A. Konar, Differential evolution using a neighborhood-based mutation operator, IEEE Trans. on Evol. Comput. 13 (3) (2009) 526–553.
[45] L. Gurel, O. Ergul, Design and simulation of circular arrays of trapezoidal-tooth logperiodic antennas via genetic optimization, Progress In Electromagnetics Research PIER 85 (2008) 243 – 260.
[46] S. Das, P. N. Suganthan, Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems (2010).
URL `http://www.ntu.edu.sg/home/EPNSugan`
[47] W. Gong, Z. Cai, C. X. Ling, H. Li, Enhanced differential evolution with adaptive strategies for numerical optimization, IEEE Transactions on Systems, Man, and Cybernetics: Part B – Cybernetics 41 (2) (2011) 397–413.
[48] A. LaTorre, S. Muelas, J.-M. Peña, A MOS-based dynamic memetic differential evolution algorithm for continuous optimization: A scalability test, Soft Computing 15 (11) (2011) 2187 – 2199.
[49] S.-Z. Zhao, P. Suganthan, S. Das, Self-adaptive differential evolution with multi-trajectory search for large-scale optimization, Soft Computing 15 (11) (2011) 2175–2185.
[50] J. Brest, M. S. Maučec, Self-adaptive differential evolution algorithm using population size reduction and three strategies, Soft Computing 15 (11) (2011) 2157–2174.
[51] Z. Yang, K. Tang, X. Yao, Large scale evolutionary optimization using cooperative coevolution, Inf. Sci. 178 (2008) 2985–2999.
[52] A. Zamuda, J. Brest, B. Bošković, V. Žumer, Large scale global optimization using differential evolution with self-adaptation and cooperative co-evolution, in: 2008 IEEE World Congr. on Comput. Intell., 2008, pp. 3719–3726.

Table 1: Comparison on the Error Values Between JADE and Its Corresponding R_{cr}-JADE for All Functions at D = 30.

Prob	JADE-s1		R_{cr}-JADE-s1	JADE-s2		R_{cr}-JADE-s2
F01	0.00E+00 ± 0.00E+00	=	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	=	0.00E+00 ± 0.00E+00
F02	1.22E-27 ± 1.20E-27	+	8.26E-28 ± 4.54E-28	1.35E-27 ± 2.65E-27	+	6.38E-28 ± 3.92E-28
F03	1.55E+04 ± 1.06E+04	=	1.60E+04 ± 1.04E+04	2.82E+04 ± 1.53E+04	+	2.22E+04 ± 1.71E+04
F04	3.88E-09 ± 1.65E-08	–	3.47E-08 ± 1.41E-07	1.02E+03 ± 2.46E+03	=	2.78E-07 ± 1.02E-06
F05	1.69E+01 ± 3.90E+01	=	4.28E+01 ± 1.14E+02	9.61E+01 ± 1.55E+02	=	1.33E+02 ± 2.18E+02
F06	1.77E+01 ± 3.53E+01	=	8.77E-01 ± 1.67E+00	5.78E+00 ± 2.10E+01	=	**4.78E-01 ± 1.31E+00**
F07	1.29E-02 ± 9.11E-03	=	1.50E-02 ± 1.32E-02	1.33E-02 ± 1.01E-02	=	1.48E-02 ± 1.36E-02
F08	2.09E+01 ± 1.39E-01	+	**2.02E+01 ± 3.28E-01**	2.09E+01 ± 1.37E-01	+	**2.02E+01 ± 3.61E-01**
F09	0.00E+00 ± 0.00E+00	=	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	=	0.00E+00 ± 0.00E+00
F10	3.53E+01 ± 5.74E+00	+	**2.38E+01 ± 4.88E+00**	3.36E+01 ± 9.82E+00	+	2.73E+01 ± 8.69E+00
F11	2.74E+01 ± 1.57E+00	=	2.71E+01 ± 1.81E+00	1.69E+01 ± 3.48E+00	=	1.70E+01 ± 3.43E+00
F12	4.99E+03 ± 4.32E+03	+	1.70E+03 ± 2.09E+03	**1.16E+03 ± 1.88E+03**	=	1.50E+03 ± 2.17E+03
F13	1.87E+00 ± 1.52E-01	+	**1.52E+00 ± 1.23E-01**	2.18E+00 ± 1.77E-01	+	1.71E+00 ± 1.08E-01
F14	1.26E+01 ± 2.21E-01	+	1.22E+01 ± 3.17E-01	1.27E+01 ± 2.44E-01	+	**1.10E+01 ± 9.76E-01**
F15	3.69E+02 ± 9.08E+01	=	**3.46E+02 ± 1.16E+02**	3.48E+02 ± 9.31E+01	=	3.50E+02 ± 7.35E+01
F16	7.20E+01 ± 5.46E+01	+	7.78E+01 ± 1.06E+02	9.32E+01 ± 1.04E+02	+	**6.39E+01 ± 7.30E+01**
F17	1.35E+02 ± 8.02E+01	+	8.72E+01 ± 5.94E+01	**8.17E+01 ± 8.38E+01**	+	8.55E+01 ± 1.15E+02
F18	8.96E+02 ± 3.93E+01	=	**8.80E+02 ± 5.27E+01**	9.00E+02 ± 3.37E+01	=	9.02E+02 ± 3.05E+01
F19	**8.89E+02 ± 4.49E+01**	=	**8.92E+02 ± 4.36E+01**	9.06E+02 ± 2.19E+01	–	9.09E+02 ± 1.59E+01
F20	**8.93E+02 ± 4.12E+01**	=	**8.92E+02 ± 4.37E+01**	9.01E+02 ± 3.02E+01	–	9.07E+02 ± 2.22E+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
F22	9.10E+02 ± 1.04E+01	+	9.01E+02 ± 1.77E+01	9.10E+02 ± 9.17E+00	+	**8.87E+02 ± 1.80E+01**
F23	**5.34E+02 ± 7.89E-05**	+	5.50E+02 ± 7.97E+01	5.42E+02 ± 5.46E+01	=	5.34E+02 ± 2.34E-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	2.12E+02 ± 1.33E-01	+	2.11E+02 ± 2.03E-01	2.10E+02 ± 4.24E-01	=	2.10E+02 ± 2.04E-01
$w/t/l$	11/13/1		–	9/14/2		–

Prob	JADE-s3		R_{cr}-JADE-s3	JADE-s4		R_{cr}-JADE-s4
F01	0.00E+00 ± 0.00E+00	=	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	=	0.00E+00 ± 0.00E+00
F02	4.77E-28 ± 1.84E-28	+	**3.74E-28 ± 1.19E-28**	4.35E-28 ± 2.60E-28	+	**3.78E-28 ± 1.98E-28**
F03	**9.45E+03 ± 7.33E+03**	=	**1.06E+04 ± 8.05E+03**	1.65E+04 ± 1.28E+04	=	1.50E+04 ± 1.29E+04
F04	**2.28E-14 ± 1.34E-13**	=	**2.89E-12 ± 1.75E-11**	8.29E+02 ± 2.14E+03	=	6.37E-11 ± 3.17E-10
F05	**3.97E-02 ± 1.34E-01**	=	**1.85E-01 ± 6.42E-01**	5.60E+00 ± 2.77E+01	=	2.04E-01 ± 8.02E-01
F06	7.08E+00 ± 2.65E+01	=	7.18E-01 ± 1.55E+00	2.34E+00 ± 1.29E+01	=	**1.59E-01 ± 7.89E-01**
F07	7.83E-03 ± 8.86E-03	=	7.63E-03 ± 7.65E-03	**4.83E-03 ± 5.56E-03**	=	**5.12E-03 ± 6.94E-03**
F08	2.09E+01 ± 6.23E-02	+	2.03E+01 ± 4.46E-01	2.09E+01 ± 6.14E-02	+	2.04E+01 ± 4.56E-01
F09	0.00E+00 ± 0.00E+00	=	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	=	0.00E+00 ± 0.00E+00
F10	3.20E+01 ± 8.31E+00	+	**2.28E+01 ± 5.15E+00**	3.13E+01 ± 8.62E+00	+	2.47E+01 ± 9.35E+00
F11	2.18E+01 ± 6.88E+00	=	2.05E+01 ± 6.83E+00	**1.51E+01 ± 3.32E+00**	=	**1.60E+01 ± 3.25E+00**
F12	3.76E+03 ± 4.16E+03	+	2.37E+03 ± 3.09E+03	**1.14E+03 ± 1.40E+03**	=	1.51E+03 ± 2.77E+03
F13	1.82E+00 ± 1.57E-01	+	**1.55E+00 ± 1.18E-01**	2.16E+00 ± 1.48E-01	+	1.69E+00 ± 1.11E-01
F14	1.25E+01 ± 2.40E-01	+	1.20E+01 ± 3.41E-01	1.27E+01 ± 1.98E-01	+	**1.12E+01 ± 1.02E+00**
F15	3.54E+02 ± 9.73E+01	=	3.64E+02 ± 1.06E+02	**3.40E+02 ± 8.33E+01**	=	3.48E+02 ± 6.46E+01
F16	6.86E+01 ± 5.47E+01	+	7.88E+01 ± 1.09E+02	7.57E+01 ± 8.21E+01	+	**5.60E+01 ± 5.53E+01**
F17	1.62E+02 ± 1.20E+02	+	1.14E+02 ± 1.15E+02	**8.15E+01 ± 8.72E+01**	=	8.75E+01 ± 1.12E+02
F18	**8.88E+02 ± 4.45E+01**	=	8.91E+02 ± 4.29E+01	9.07E+02 ± 1.56E+01	=	9.10E+02 ± 2.20E+00
F19	8.99E+02 ± 3.35E+01	=	9.06E+02 ± 2.21E+01	9.07E+02 ± 1.56E+01	=	9.10E+02 ± 2.49E+00
F20	8.99E+02 ± 3.35E+01	–	9.07E+02 ± 2.21E+01	9.07E+02 ± 1.56E+01	=	9.10E+02 ± 2.49E+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	9.06E+02 ± 1.19E+01	+	8.92E+02 ± 1.48E+01	9.00E+02 ± 8.73E+00	+	**8.63E+02 ± 1.47E+01**
F23	5.50E+02 ± 7.76E+01	=	5.42E+02 ± 5.70E+01	**5.34E+02 ± 3.51E-04**	+	5.34E+02 ± 3.71E-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
F25	2.12E+02 ± 1.05E-01	+	2.10E+02 ± 3.85E-01	**2.09E+02 ± 1.32E-01**	=	**2.09E+02 ± 8.67E-02**
$w/t/l$	10/14/1		–	8/17/0		–

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

Table 2: Comparison on the Error Values Between JADE and Its Corresponding R$_{cr}$-JADE for All Functions at D = 50.

Prob	JADE-s1		R$_{cr}$-JADE-s1	JADE-s2		R$_{cr}$-JADE-s2
F01	0.00E+00 ± 0.00E+00	=	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	=	0.00E+00 ± 0.00E+00
F02	7.69E-21 ± 1.50E-20	=	1.09E-20 ± 2.17E-20	5.18E+03 ± 8.61E+03	+	5.08E-19 ± 1.62E-18
F03	1.95E+04 ± 9.19E+03	=	2.31E+04 ± 1.06E+04	1.45E+06 ± 7.01E+06	=	3.26E+04 ± 1.38E+04
F04	1.12E+01 ± 1.87E+01	–	2.76E+01 ± 4.24E+01	1.43E+04 ± 1.90E+04	+	6.18E+02 ± 4.21E+03
F05	2.48E+03 ± 4.87E+02	=	2.50E+03 ± 4.55E+02	2.65E+03 ± 5.87E+02	=	2.54E+03 ± 3.71E+02
F06	3.97E+00 ± 1.39E+01	=	2.07E+00 ± 2.01E+00	3.61E+00 ± 1.53E+01	=	1.28E+00 ± 1.88E+00
F07	6.78E-03 ± 1.15E-02	=	8.90E-03 ± 1.27E-02	**1.77E-03 ± 4.14E-03**	=	2.46E-03 ± 9.37E-03
F08	2.11E+01 ± 2.71E-01	+	**2.03E+01 ± 5.06E-01**	2.11E+01 ± 2.52E-01	+	**2.05E+01 ± 5.38E-01**
F09	0.00E+00 ± 0.00E+00	=	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	=	0.00E+00 ± 0.00E+00
F10	6.57E+01 ± 1.06E+01	=	6.49E+01 ± 1.16E+01	5.15E+01 ± 1.07E+01	+	**4.87E+01 ± 1.29E+01**
F11	5.26E+01 ± 2.44E+00	=	5.30E+01 ± 2.30E+00	5.28E+01 ± 8.21E+00	+	**4.80E+01 ± 1.20E+01**
F12	1.56E+04 ± 1.76E+04	+	**5.96E+03 ± 7.43E+03**	2.81E+04 ± 2.67E+04	+	9.10E+03 ± 1.12E+04
F13	**2.65E+00 ± 1.91E-01**	–	2.77E+00 ± 2.20E-01	2.89E+00 ± 1.74E-01	–	3.06E+00 ± 1.72E-01
F14	2.17E+01 ± 3.24E-01	+	2.14E+01 ± 3.96E-01	2.19E+01 ± 9.25E-01	+	**2.11E+01 ± 1.08E+00**
F15	3.34E+02 ± 9.20E+01	=	3.25E+02 ± 9.54E+01	3.26E+02 ± 9.43E+01	=	**3.04E+02 ± 1.07E+02**
F16	7.55E+01 ± 7.40E+01	+	5.66E+01 ± 5.17E+01	9.88E+01 ± 1.25E+02	+	6.28E+01 ± 7.38E+01
F17	1.11E+02 ± 4.96E+01	+	1.11E+02 ± 6.57E+01	**6.60E+01 ± 4.22E+01**	=	7.58E+01 ± 9.85E+01
F18	9.40E+02 ± 3.10E+01	=	9.34E+02 ± 3.64E+01	9.39E+02 ± 8.35E+00	=	9.36E+02 ± 2.94E+01
F19	9.40E+02 ± 2.28E+01	=	9.39E+02 ± 1.90E+01	9.39E+02 ± 8.73E+00	–	9.43E+02 ± 7.45E+00
F20	9.39E+02 ± 2.27E+01	=	9.41E+02 ± 1.70E+01	9.39E+02 ± 9.06E+00	–	9.42E+02 ± 7.41E+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	9.48E+02 ± 9.61E+00	=	9.50E+02 ± 8.82E+00	9.25E+02 ± 2.09E+01	+	**9.19E+02 ± 1.31E+01**
F23	5.59E+02 ± 1.04E+02	=	5.46E+02 ± 4.94E+01	5.39E+02 ± 5.21E-03	=	5.39E+02 ± 1.75E-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
F25	2.14E+02 ± 8.67E-01	=	2.14E+02 ± 7.00E-01	2.15E+02 ± 7.66E-01	=	2.15E+02 ± 8.85E-01
$w/t/l$	5/18/2		–	9/13/3		–

Prob	JADE-s3		R$_{cr}$-JADE-s3	JADE-s4		R$_{cr}$-JADE-s4
F01	0.00E+00 ± 0.00E+00	=	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	=	0.00E+00 ± 0.00E+00
F02	**1.09E-26 ± 7.27E-27**	=	**1.15E-26 ± 5.27E-27**	6.98E+03 ± 9.70E+03	+	2.14E-26 ± 1.64E-26
F03	**1.70E+04 ± 1.01E+04**	=	**1.57E+04 ± 7.74E+03**	3.24E+06 ± 8.22E+06	=	2.46E+04 ± 1.35E+04
F04	**3.79E+00 ± 1.70E+01**	=	**2.97E+00 ± 1.05E+01**	1.15E+04 ± 1.69E+04	+	8.21E+02 ± 5.80E+03
F05	1.89E+03 ± 3.98E+02	+	**1.81E+03 ± 4.43E+02**	2.08E+03 ± 9.91E+02	+	**1.74E+03 ± 3.74E+02**
F06	1.12E+00 ± 1.81E+00	=	1.67E+00 ± 1.99E+00	**3.99E-01 ± 1.21E+00**	=	**5.58E-01 ± 1.40E+00**
F07	4.92E-03 ± 9.15E-03	=	3.20E-03 ± 5.95E-03	4.38E-03 ± 7.43E-03	+	**1.87E-03 ± 5.36E-03**
F08	2.11E+01 ± 2.69E-01	+	2.07E+01 ± 5.33E-01	2.11E+01 ± 2.72E-01	+	2.07E+01 ± 5.51E-01
F09	0.00E+00 ± 0.00E+00	=	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	=	0.00E+00 ± 0.00E+00
F10	6.42E+01 ± 8.91E+00	+	5.61E+01 ± 9.72E+00	**4.90E+01 ± 1.13E+01**	=	5.12E+01 ± 1.18E+01
F11	5.23E+01 ± 2.20E+00	=	5.24E+01 ± 2.27E+00	5.53E+01 ± 7.90E+00	+	**4.32E+01 ± 1.15E+01**
F12	2.09E+04 ± 2.24E+04	+	8.87E+03 ± 1.45E+04	3.00E+04 ± 2.67E+04	+	**6.89E+03 ± 1.15E+04**
F13	**2.69E+00 ± 1.90E-01**	–	2.86E+00 ± 1.66E-01	2.94E+00 ± 1.69E-01	–	3.04E+00 ± 2.05E-01
F14	2.17E+01 ± 3.53E-01	+	2.15E+01 ± 4.86E-01	2.17E+01 ± 1.03E+00	+	**2.08E+01 ± 1.24E+00**
F15	3.46E+02 ± 8.80E+01	=	3.22E+02 ± 9.51E+01	**3.06E+02 ± 9.77E+01**	=	3.10E+02 ± 1.04E+02
F16	6.73E+01 ± 6.99E+01	+	6.27E+01 ± 7.12E+01	**5.21E+01 ± 5.22E+01**	=	**5.02E+01 ± 2.47E+01**
F17	1.17E+02 ± 6.24E+01	+	9.79E+01 ± 2.67E+01	8.08E+01 ± 6.91E+01	+	**6.33E+01 ± 7.27E+01**
F18	9.33E+02 ± 3.60E+01	=	**9.29E+02 ± 4.06E+01**	9.31E+02 ± 2.03E+01	=	**9.30E+02 ± 2.78E+01**
F19	9.36E+02 ± 2.30E+01	=	9.38E+02 ± 2.98E+01	**9.29E+02 ± 2.78E+01**	–	**9.35E+02 ± 2.29E+01**
F20	9.35E+02 ± 2.26E+01	=	9.36E+02 ± 2.97E+01	**9.28E+02 ± 2.77E+01**	–	**9.35E+02 ± 2.24E+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
F22	9.48E+02 ± 9.49E+00	+	9.44E+02 ± 1.12E+01	9.21E+02 ± 2.63E+01	+	**9.05E+02 ± 1.33E+01**
F23	5.39E+02 ± 3.26E-03	=	5.39E+02 ± 7.48E-03	5.39E+02 ± 6.38E-03	=	5.39E+02 ± 8.89E-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	2.14E+02 ± 9.13E-01	=	**2.14E+02 ± 6.18E-01**	2.14E+02 ± 9.23E-01	+	**2.14E+02 ± 5.07E-01**
$w/t/l$	8/16/1		–	11/11/3		–

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

Table 3: Direct Comparison on the Error Values Among Different State-of-the-Art DE Variants for All Functions at $D = 30$.

Prob	jDE		SaDE		EPSDE-c		CoDE		R_{cr}-JADE-s4
F01	**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**
F02	1.22E-05 ± 2.22E-05	+	1.01E-15 ± 1.89E-15	+	**1.20E-27 ± 3.88E-27**	+	3.57E-14 ± 8.14E-14	+	**3.78E-28 ± 1.98E-28**
F03	1.94E+05 ± 1.15E+05	+	7.65E+04 ± 6.50E+04	+	**5.99E+04 ± 2.77E+04**	+	1.41E+05 ± 7.39E+04	+	**1.50E+04 ± 1.29E+04**
F04	1.86E-01 ± 2.33E-01	+	4.25E-02 ± 2.02E-01	+	**2.02E-09 ± 4.51E-09**	+	6.79E-02 ± 2.87E-01	+	**6.37E-11 ± 3.17E-10**
F05	1.06E+03 ± 4.38E+02	+	6.93E+02 ± 6.33E+02	+	**2.25E+02 ± 2.38E+02**	+	8.27E+02 ± 4.12E+02	+	**2.04E-01 ± 8.02E-01**
F06	2.93E+01 ± 2.79E+01	+	9.41E-01 ± 1.84E+00	+	**1.59E-01 ± 2.18E-01**	+	**3.29E-08 ± 1.22E-07**	+	**1.59E-01 ± 7.89E-01**
F07	1.17E-02 ± 9.90E-03	+	1.68E-02 ± 1.15E-02	+	9.86E-03 ± 8.94E-03	=	**9.60E-03 ± 8.84E-03**	+	**5.12E-03 ± 6.94E-03**
F08	2.09E+01 ± 5.18E-02	+	2.09E+01 ± 5.48E-02	+	2.09E+01 ± 4.14E-02	+	2.09E+01 ± 4.66E-02	+	**2.04E+01 ± 4.56E-01**
F09	**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**
F10	5.54E+01 ± 9.44E+00	+	5.99E+01 ± 1.13E+01	+	3.14E+01 ± 4.31E+00	+	**4.63E+01 ± 1.03E+01**	+	**2.47E+01 ± 9.35E+00**
F11	2.85E+01 ± 1.71E+00	+	2.79E+01 ± 4.38E+00	+	2.03E+01 ± 1.20E+01	+	**1.10E+01 ± 2.99E+00**	–	**1.60E+01 ± 3.25E+00**
F12	1.45E+04 ± 7.82E+03	+	3.69E+03 ± 5.80E+03	+	2.41E+03 ± 2.15E+03	+	**1.68E+03 ± 2.21E+03**	=	**1.51E+03 ± 2.77E+03**
F13	**1.67E+00 ± 1.54E-01**	=	2.64E+00 ± 1.85E-01	+	3.76E+00 ± 3.54E+00	+	3.25E+00 ± 1.16E+00	+	**1.69E+00 ± 1.11E-01**
F14	1.30E+01 ± 2.20E-01	+	1.29E+01 ± 1.99E-01	+	1.26E+01 ± 2.40E-01	+	**1.23E+01 ± 4.73E-01**	+	**1.12E+01 ± 1.02E+00**
F15	3.54E+02 ± 9.33E+01	=	4.04E+02 ± 4.02E+01	+	**2.15E+02 ± 1.88E+02**	–	4.04E+02 ± 1.98E+01	+	**3.48E+02 ± 6.46E+01**
F16	7.47E+01 ± 1.12E+01	+	7.89E+01 ± 9.65E+00	+	9.23E+01 ± 4.38E+01	+	**6.80E+01 ± 1.33E+01**	+	**5.60E+01 ± 5.53E+01**
F17	1.33E+02 ± 1.70E+01	+	1.38E+02 ± 2.35E+01	+	1.36E+02 ± 3.66E+01	+	**6.58E+01 ± 1.36E+01**	=	**8.75E+01 ± 1.12E+02**
F18	9.06E+02 ± 1.74E+00	–	**8.62E+02 ± 5.56E+01**	–	**8.21E+02 ± 4.22E+00**	–	8.91E+02 ± 4.01E+01	–	9.10E+02 ± 2.20E+00
F19	9.07E+02 ± 1.75E+00	–	**8.55E+02 ± 5.61E+01**	–	**8.22E+02 ± 3.87E+00**	–	8.95E+02 ± 3.57E+01	–	9.10E+02 ± 2.49E+00
F20	9.07E+02 ± 1.79E+00	–	**8.58E+02 ± 5.60E+01**	–	**8.21E+02 ± 4.66E+00**	–	8.96E+02 ± 3.57E+01	–	9.10E+02 ± 2.49E+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
F22	9.02E+02 ± 9.14E+00	+	9.15E+02 ± 1.23E+01	+	**8.77E+02 ± 1.61E+01**	+	9.18E+02 ± 1.23E+01	+	**8.63E+02 ± 1.47E+01**
F23	5.34E+02 ± 2.14E-04	–	5.34E+02 ± 1.60E-04	=	5.34E+02 ± 1.76E-02	+	5.34E+02 ± 4.29E-04	+	5.34E+02 ± 3.71E-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
F25	**2.10E+02 ± 3.33E-01**	+	2.10E+02 ± 3.34E-01	+	2.11E+02 ± 5.17E-01	+	2.10E+02 ± 4.11E-01	+	**2.09E+02 ± 2.51E-01**
$w/t/l$	15/6/4		17/5/3		16/5/4		15/6/4		–

"+", "–", 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 5: Indirect Comparison on the Error Values Among Different State-of-the-Art DE Variants for All Functions at $D = 30$.

Prob	jDE [35]	SaDE [35]	JADE [35]	EPSDE-c [30]	CoDE [35]	R_{cr}-JADE-s4
F01	**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**
F02	1.11E-06 ± 1.96E-06	8.26E-06 ± 1.65E-05	**1.07E-28 ± 1.00E-28**	3.37E-27 ± 4.73E-27	1.69E-15 ± 3.95E-15	**3.78E-28 ± 1.98E-28**
F03	1.98E+05 ± 1.10E+05	4.27E+05 ± 2.08E+05	**8.42E+03 ± 7.26E+03**	7.74E+04 ± 3.77E+04	1.05E+05 ± 6.25E+04	**1.50E+04 ± 1.29E+04**
F04	4.40E-02 ± 1.26E-01	1.77E+02 ± 2.67E+02	**1.73E-16 ± 5.43E-16**	**1.76E-12 ± 2.97E-12**	5.81E-03 ± 1.38E-02	6.37E-11 ± 3.17E-10
F05	5.11E+02 ± 4.40E+02	3.25E+03 ± 5.90E+02	**8.59E-08 ± 5.23E-07**	2.26E+02 ± 2.61E+02	3.31E+02 ± 3.44E+02	**2.04E-01 ± 8.02E-01**
F06	2.35E+01 ± 2.50E+01	5.31E+01 ± 3.25E+01	1.02E+01 ± 2.96E+01	**2.12E-20 ± 1.13E-19**	1.60E-01 ± 7.85E-01	**1.59E-01 ± 7.89E-01**
F07	1.18E-02 ± 7.78E-03	1.57E-02 ± 1.38E-02	8.07E-03 ± 7.42E-03	**5.60E-03 ± 6.11E-03**	7.46E-03 ± 8.55E-03	**5.12E-03 ± 6.94E-03**
F08	2.09E+01 ± 4.86E-02	2.09E+01 ± 4.95E-02	2.09E+01 ± 1.68E-01	2.08E+01 ± 1.31E-01	**2.01E+01 ± 1.41E-01**	**2.04E+01 ± 4.56E-01**
F09	**0.00E+00 ± 0.00E+00**	**2.39E-01 ± 4.33E-01**	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**
F10	5.54E+01 ± 8.46E+00	4.72E+01 ± 1.01E+01	**2.41E+01 ± 4.61E+00**	4.71E+01 ± 1.52E+01	4.15E+01 ± 1.16E+01	**2.47E+01 ± 9.35E+00**
F11	2.79E+01 ± 1.61E+00	1.65E+01 ± 2.42E+00	2.53E+01 ± 1.65E+00	2.86E+01 ± 9.61E-01	**1.18E+01 ± 3.40E+00**	**1.60E+01 ± 3.25E+00**
F12	8.63E+03 ± 8.31E+03	**3.02E+03 ± 2.33E+03**	6.15E+03 ± 4.79E+03	1.32E+04 ± 1.35E+04	3.05E+03 ± 3.80E+03	**1.51E+03 ± 2.77E+03**
F13	1.66E+00 ± 1.35E-01	3.94E+00 ± 2.81E-01	**1.49E+00 ± 1.09E-01**	**1.19E+00 ± 1.24E-01**	1.57E+00 ± 3.27E-01	1.69E+00 ± 1.11E-01
F14	1.30E+01 ± 2.00E-01	1.26E+01 ± 2.83E-01	**1.23E+01 ± 3.11E-01**	1.25E+01 ± 1.64E-01	**1.23E+01 ± 4.81E-01**	**1.12E+01 ± 1.02E+00**
F15	3.77E+02 ± 8.02E+01	3.76E+02 ± 7.83E+01	3.51E+02 ± 1.28E+02	**2.12E+02 ± 1.98E+01**	3.88E+02 ± 6.85E+01	**3.48E+02 ± 6.46E+01**
F16	7.94E+01 ± 2.96E+01	8.57E+01 ± 6.94E+01	1.01E+02 ± 1.24E+02	9.08E+01 ± 2.98E+01	**7.37E+01 ± 5.13E+01**	**5.60E+01 ± 5.53E+01**
F17	1.37E+02 ± 3.80E+01	**7.83E+01 ± 3.76E+01**	1.47E+02 ± 1.33E+02	1.04E+02 ± 7.27E+01	**6.67E+01 ± 2.12E+01**	8.75E+01 ± 1.12E+02
F18	9.04E+02 ± 1.08E+01	**8.68E+02 ± 6.23E+01**	9.04E+02 ± 1.03E+00	**8.20E+02 ± 3.35E+00**	9.04E+02 ± 1.04E+00	9.10E+02 ± 2.20E+00
F19	9.04E+02 ± 1.11E+00	**8.74E+02 ± 6.22E+01**	9.04E+02 ± 8.40E-01	**8.21E+02 ± 3.35E+00**	9.04E+02 ± 9.42E-01	9.10E+02 ± 2.49E+00
F20	9.04E+02 ± 1.10E+00	**8.78E+02 ± 6.03E+01**	9.04E+02 ± 8.47E-01	**8.22E+02 ± 4.17E+00**	9.04E+02 ± 9.01E-01	9.10E+02 ± 2.49E+00
F21	**5.00E+02 ± 4.80E-13**	**5.52E+02 ± 1.82E+02**	**5.00E+02 ± 4.67E-13**	**5.00E+02 ± 6.64E-14**	**5.00E+02 ± 4.88E-13**	**5.00E+02 ± 0.00E+00**
F22	8.75E+02 ± 1.91E+01	9.36E+02 ± 1.83E+01	**8.66E+02 ± 1.91E+01**	8.85E+02 ± 6.82E+01	**8.63E+02 ± 2.43E+01**	**8.63E+02 ± 1.47E+01**
F23	**5.34E+02 ± 2.77E-04**	**5.34E+02 ± 3.57E-03**	5.50E+02 ± 8.05E+01	**5.07E+02 ± 7.26E+00**	**5.34E+02 ± 4.12E-04**	**5.34E+02 ± 3.71E-04**
F24	**2.00E+02 ± 2.85E-14**	**2.00E+02 ± 6.20E-13**	**2.00E+02 ± 2.85E-14**	2.13E+02 ± 1.52E+00	**2.00E+02 ± 2.85E-14**	**2.00E+02 ± 0.00E+00**
F25	**2.11E+02 ± 7.32E-01**	2.14E+02 ± 2.00E+00	**2.11E+02 ± 7.92E-01**	2.13E+02 ± 2.55E+00	**2.11E+02 ± 9.02E-01**	**2.09E+02 ± 2.51E-01**

Table 7: Comparison on the Error Values Among Different State-of-the-Art EAs for All Functions at $D = 30$.

Prob	GL-25	LEP	CMA-ES	CLPSO	OLPSO-L	R_{cr}-JADE-s4
F01	**1.60E-27 ± 2.11E-27** +	6.24E-06 ± 9.56E-07 +	1.25E-25 ± 2.38E-26 +	**0.00E+00 ± 0.00E+00** =	**0.00E+00 ± 0.00E+00** =	**0.00E+00 ± 0.00E+00**
F02	2.87E+01 ± 5.45E+01 +	4.46E+01 ± 4.49E+01 +	**6.70E-25 ± 2.45E-25** +	6.91E+01 ± 2.06E+01 +	1.29E+02 ± 2.84E+02 +	**3.78E-28 ± 1.98E-28**
F03	2.36E+06 ± 1.11E+06 +	3.30E+06 ± 1.65E+06 +	**5.31E-21 ± 1.51E-21** –	1.39E+07 ± 2.83E+06 +	6.99E+06 ± 3.40E+06 +	**1.50E+04 ± 1.29E+04**
F04	7.83E+02 ± 4.58E+02 +	6.18E+03 ± 3.14E+03 +	4.79E+05 ± 1.75E+06 +	1.78E+03 ± 4.56E+02 +	**4.53E+02 ± 4.40E+02** +	**6.37E-11 ± 3.17E-10**
F05	2.59E+03 ± 2.57E+02 +	5.91E+03 ± 1.19E+03 +	**3.54E-10 ± 7.46E-11** –	2.06E+03 ± 4.00E+02 +	2.73E+03 ± 1.12E+03 +	**2.04E-01 ± 8.02E-01**
F06	2.17E+01 ± 1.41E+00 +	1.71E+02 ± 2.90E+02 +	**6.38E-01 ± 1.48E+00** =	2.83E+01 ± 9.68E+00 +	1.60E+01 ± 4.02E+01 +	**1.59E-01 ± 7.89E-01**
F07	3.10E-02 ± 6.78E-02 +	4.81E-02 ± 3.77E-02 +	**2.15E-03 ± 3.36E-03** –	3.63E-02 ± 2.93E-02 +	9.59E-01 ± 8.68E-01 +	**5.12E-03 ± 6.94E-03**
F08	2.10E+01 ± 4.70E-02 +	2.10E+01 ± 4.91E-02 +	**2.03E+01 ± 5.96E-01** –	2.09E+01 ± 5.11E-02 +	2.10E+01 ± 6.09E-02 +	**2.04E+01 ± 4.56E-01**
F09	2.55E+01 ± 6.56E+00 +	1.81E-03 ± 6.18E-04 +	4.01E+02 ± 1.28E+02 +	**0.00E+00 ± 0.00E+00** =	**0.00E+00 ± 0.00E+00** =	**0.00E+00 ± 0.00E+00**
F10	1.51E+02 ± 5.34E+01 +	8.23E+01 ± 2.08E+01 +	**4.54E+01 ± 1.15E+01** +	1.15E+02 ± 1.56E+01 +	6.88E+01 ± 1.68E+01 +	**2.47E+01 ± 9.35E+00**
F11	3.15E+01 ± 8.14E+00 +	3.93E+01 ± 1.21E+00 +	**5.89E+00 ± 2.12E+00** –	2.62E+01 ± 1.51E+00 +	2.93E+01 ± 4.38E+00 +	**1.60E+01 ± 3.25E+00**
F12	8.32E+03 ± 6.50E+03 +	**6.91E+03 ± 6.63E+03** +	8.31E+03 ± 1.03E+04 +	3.36E+04 ± 6.54E+03 +	9.84E+03 ± 5.22E+03 +	**1.51E+03 ± 2.77E+03**
F13	5.17E+00 ± 4.17E+00 +	2.18E+00 ± 6.40E-01 +	3.47E+00 ± 7.17E-01 +	7.01E+00 ± 7.09E-01 +	**1.11E+00 ± 3.61E-01** –	**1.69E+00 ± 1.11E-01**
F14	1.30E+01 ± 2.00E-01 +	**1.18E+01 ± 7.73E-01** +	1.47E+01 ± 2.84E-01 +	1.29E+01 ± 1.90E-01 +	1.33E+01 ± 3.28E-01 +	**1.12E+01 ± 1.02E+00**
F15	3.00E+02 ± 2.94E-02 –	3.45E+02 ± 7.75E+01 =	5.02E+02 ± 2.92E+02 +	**2.92E+02 ± 4.54E+01** –	**2.96E+02 ± 7.54E+01** –	3.48E+02 ± 6.46E+01
F16	**1.01E+02 ± 8.30E+01** +	1.43E+02 ± 1.05E+02 +	3.63E+02 ± 2.47E+02 +	1.98E+02 ± 3.34E+01 +	1.32E+02 ± 4.21E+01 +	**5.60E+01 ± 5.53E+01**
F17	2.09E+02 ± 7.79E+00 +	**1.21E+02 ± 6.88E+01** +	4.16E+02 ± 3.99E+02 +	2.33E+02 ± 2.97E+01 +	1.61E+02 ± 4.01E+01 +	**8.75E+01 ± 1.12E+02**
F18	9.06E+02 ± 1.56E+00 –	9.30E+02 ± 2.07E+01 +	9.06E+02 ± 1.20E+01 –	9.06E+02 ± 6.21E-01 –	9.07E+02 ± 1.41E+00 –	9.10E+02 ± 2.20E+00
F19	9.07E+02 ± 2.67E+00 –	9.27E+02 ± 2.84E+01 +	**9.04E+02 ± 2.35E-01** –	**9.06E+02 ± 6.86E-01** –	9.07E+02 ± 1.36E+00 –	9.10E+02 ± 2.49E+00
F20	**9.05E+02 ± 3.65E+00** –	9.33E+02 ± 1.02E+01 +	**9.04E+02 ± 2.78E-01** –	9.06E+02 ± 6.90E-01 –	9.07E+02 ± 1.34E+00 –	9.10E+02 ± 2.49E+00
F21	5.00E+02 ± 0.00E+00 =	5.32E+02 ± 1.18E+02 +	5.16E+02 ± 7.92E+01 =	5.00E+02 ± 0.00E+00 =	5.19E+02 ± 2.59E+01 +	5.00E+02 ± 0.00E+00
F22	9.32E+02 ± 8.69E+00 +	9.14E+02 ± 2.78E+01 +	**8.25E+02 ± 1.72E+01** –	9.02E+02 ± 8.88E+00 +	8.91E+02 ± 1.84E+01 +	**8.63E+02 ± 1.47E+01**
F23	5.34E+02 ± 6.00E-04 =	5.89E+02 ± 1.71E+02 +	5.44E+02 ± 5.42E+01 +	5.34E+02 ± 8.42E-05 =	5.78E+02 ± 4.52E+01 +	5.34E+02 ± 3.71E-04
F24	**2.00E+02 ± 0.00E+00** =	**2.00E+02 ± 1.09E-04** +	**2.00E+02 ± 0.00E+00** =	2.01E+02 ± 6.48E+00 +	6.99E+02 ± 2.63E+02 +	**2.00E+02 ± 0.00E+00**
F25	2.15E+02 ± 2.35E+00 +	2.20E+02 ± 7.58E+00 +	2.14E+04 ± 3.25E-01 +	2.10E+02 ± 3.81E-01 +	**2.09E+02 ± 4.21E-01** =	**2.09E+02 ± 2.51E-01**
$w/t/l$	18/3/4	24/1/0	13/3/9	17/4/4	17/3/5	–

"+", "–", 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 9: Comparison on the Error Values Between SaDE and Its Corresponding R_{cr}-SaDE for All Functions at $D = 30$ and $D = 50$, Respectively.

Prob	$D = 30$			$D = 50$		
	SaDE		R_{cr}-SaDE	SaDE		R_{cr}-SaDE
F01	0.00E+00 ± 0.00E+00	=	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	=	0.00E+00 ± 0.00E+00
F02	1.01E-15 ± 1.89E-15	+	**2.03E-16 ± 4.98E-16**	1.33E-08 ± 2.77E-08	+	**6.11E-09 ± 1.48E-08**
F03	7.65E+04 ± 6.50E+04	+	**4.84E+04 ± 3.17E+04**	**1.07E+05 ± 4.52E+04**	=	1.14E+05 ± 5.53E+04
F04	4.25E-02 ± 2.02E-01	=	**4.62E-04 ± 1.20E-03**	3.24E+02 ± 3.52E+02	+	**1.67E+02 ± 1.56E+02**
F05	6.93E+02 ± 6.33E+02	+	**3.72E+02 ± 3.51E+02**	3.49E+03 ± 5.08E+02	+	**3.23E+03 ± 5.66E+02**
F06	**9.41E-01 ± 1.84E+00**	=	1.51E+00 ± 1.95E+00	5.36E+00 ± 1.42E+01	=	**3.15E+00 ± 2.48E+00**
F07	1.68E-02 ± 1.15E-02	+	**1.34E-02 ± 1.19E-02**	**4.48E-03 ± 8.70E-03**	=	4.52E-03 ± 1.03E-02
F08	2.09E+01 ± 5.48E-02	=	2.09E+01 ± 5.36E-02	2.11E+01 ± 4.50E-02	=	2.11E+01 ± 3.18E-02
F09	0.00E+00 ± 0.00E+00	=	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	=	0.00E+00 ± 0.00E+00
F10	5.99E+01 ± 1.13E+01	+	**4.39E+01 ± 1.78E+01**	1.48E+02 ± 1.83E+01	+	**1.30E+02 ± 3.72E+01**
F11	2.79E+01 ± 4.38E+00	+	**2.21E+01 ± 8.99E+00**	5.73E+01 ± 2.04E+00	+	**5.54E+01 ± 8.01E+00**
F12	3.69E+03 ± 5.80E+03	=	**3.06E+03 ± 5.27E+03**	1.20E+04 ± 1.44E+04	=	**9.11E+03 ± 7.54E+03**
F13	**2.64E+00 ± 1.85E-01**	=	2.66E+00 ± 2.10E-01	5.51E+00 ± 3.31E-01	=	**5.47E+00 ± 3.67E-01**
F14	1.29E+01 ± 1.99E-01	=	1.29E+01 ± 1.89E-01	2.25E+01 ± 2.46E-01	=	2.25E+01 ± 2.10E-01
F15	4.04E+02 ± 4.02E+01	=	**3.98E+02 ± 5.53E+01**	**3.76E+02 ± 6.57E+01**	=	3.80E+02 ± 6.06E+01
F16	7.89E+01 ± 9.65E+00	+	**6.13E+01 ± 1.88E+01**	9.93E+01 ± 1.54E+01	+	**9.03E+01 ± 5.07E+01**
F17	1.38E+02 ± 2.35E+01	+	**8.63E+01 ± 4.79E+01**	1.97E+02 ± 1.35E+01	+	**1.79E+02 ± 3.90E+01**
F18	**8.62E+02 ± 5.56E+01**	=	8.64E+02 ± 5.49E+01	**9.21E+02 ± 4.58E+01**	=	9.29E+02 ± 3.51E+01
F19	8.55E+02 ± 5.61E+01	=	**8.51E+02 ± 5.55E+01**	9.14E+02 ± 5.42E+01	=	9.14E+02 ± 5.21E+01
F20	8.58E+02 ± 5.60E+01	=	**8.51E+02 ± 5.53E+01**	9.24E+02 ± 4.25E+01	=	**9.19E+02 ± 4.96E+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
F22	9.15E+02 ± 1.23E+01	=	**9.14E+02 ± 1.35E+01**	9.68E+02 ± 5.87E+00	+	**9.63E+02 ± 6.72E+00**
F23	5.34E+02 ± 1.60E-04	+	5.34E+02 ± 1.24E-04	5.39E+02 ± 4.14E-05	=	5.39E+02 ± 2.46E-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 ± 3.34E-01	+	**2.09E+02 ± 4.25E-01**	2.16E+02 ± 8.10E-01	+	**2.15E+02 ± 7.70E-01**
$w/t/l$	10/15/0		–	9/16/0		–

"+", "–", 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 10: Comparison on the Error Values Between EPSDE-j and Its Corresponding R_{cr}-EPSDE-j for All Functions at $D = 30$ and $D = 50$, Respectively.

Prob	$D = 30$			$D = 50$		
	EPSDE-j		R_{cr}-EPSDE-j	EPSDE-j		R_{cr}-EPSDE-j
F01	0.00E+00 ± 0.00E+00	=	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	=	0.00E+00 ± 0.00E+00
F02	**2.44E-26 ± 1.30E-26**	–	1.52E-24 ± 4.24E-24	4.41E-08 ± 2.84E-08	+	**1.67E-10 ± 2.35E-10**
F03	3.85E+05 ± 2.61E+05	+	**7.52E+04 ± 5.63E+04**	6.27E+05 ± 3.20E+05	+	**5.86E+05 ± 3.60E+05**
F04	6.54E+03 ± 6.37E+03	+	**5.96E+03 ± 5.08E+03**	**2.65E+04 ± 1.69E+04**	–	5.40E+04 ± 1.60E+04
F05	3.25E+03 ± 8.23E+02	+	**2.12E+03 ± 7.80E+02**	7.44E+03 ± 1.38E+03	+	**6.59E+03 ± 1.33E+03**
F06	5.38E-01 ± 1.56E+00	=	**4.31E-01 ± 1.76E+00**	5.95E-01 ± 1.18E+00	=	**3.92E-01 ± 1.18E+00**
F07	**1.43E-02 ± 1.37E-02**	–	2.33E-02 ± 1.82E-02	1.18E-02 ± 1.50E-02	=	**1.14E-02 ± 1.55E-02**
F08	2.09E+01 ± 6.36E-02	+	2.09E+01 ± 4.54E-02	2.11E+01 ± 3.49E-02	+	**2.10E+01 ± 6.76E-02**
F09	0.00E+00 ± 0.00E+00	=	0.00E+00 ± 0.00E+00	6.50E+01 ± 6.34E+00	+	**0.00E+00 ± 0.00E+00**
F10	5.20E+01 ± 1.16E+01	+	**4.17E+01 ± 7.87E+00**	1.90E+02 ± 5.01E+01	+	**1.19E+02 ± 3.17E+01**
F11	3.26E+01 ± 2.78E+00	+	**2.94E+01 ± 1.65E+00**	7.15E+01 ± 1.79E+00	+	**5.76E+01 ± 6.28E+00**
F12	3.74E+04 ± 6.61E+03	+	**1.84E+04 ± 2.41E+03**	3.04E+05 ± 7.11E+04	+	**7.11E+04 ± 1.22E+04**
F13	1.99E+00 ± 2.28E-01	+	**1.15E+00 ± 8.40E-02**	6.35E+00 ± 3.22E-01	+	**2.16E+00 ± 1.61E-01**
F14	1.34E+01 ± 3.02E-01	+	**1.30E+01 ± 1.68E-01**	2.28E+01 ± 7.34E-01	+	**2.26E+01 ± 1.36E-01**
F15	2.16E+02 ± 2.07E+01	+	**2.11E+02 ± 3.14E+01**	2.66E+02 ± 6.36E+01	+	**2.03E+02 ± 3.91E+00**
F16	1.69E+02 ± 1.45E+02	=	1.69E+02 ± 1.63E+02	1.56E+02 ± 4.74E+01	+	**1.41E+02 ± 6.35E+01**
F17	1.81E+02 ± 1.02E+02	+	**1.41E+02 ± 1.12E+02**	**2.22E+02 ± 1.31E+02**	–	2.99E+02 ± 1.34E+02
F18	**8.22E+02 ± 3.51E+00**	–	8.26E+02 ± 4.06E+00	8.71E+02 ± 4.65E+01	+	**8.53E+02 ± 5.12E+00**
F19	**8.25E+02 ± 2.82E+00**	–	8.29E+02 ± 5.86E+00	**8.49E+02 ± 2.41E+00**	–	8.77E+02 ± 3.67E+01
F20	**8.23E+02 ± 3.55E+00**	–	8.28E+02 ± 5.69E+00	**8.46E+02 ± 2.91E+00**	–	8.53E+02 ± 5.06E+00
F21	8.62E+02 ± 3.12E+00	+	**7.94E+02 ± 1.55E+02**	7.32E+02 ± 3.20E+00	=	**6.92E+02 ± 1.07E+02**
F22	**5.12E+02 ± 8.21E+00**	–	5.52E+02 ± 1.05E+02	5.00E+02 ± 7.07E-01	=	5.00E+02 ± 5.67E-01
F23	8.89E+02 ± 6.18E+01	=	**8.73E+02 ± 4.00E+00**	7.40E+02 ± 3.12E+00	+	**6.60E+02 ± 1.05E+02**
F24	**2.20E+02 ± 4.43E+00**	–	2.21E+02 ± 7.14E+00	4.72E+02 ± 4.10E+02	=	**4.37E+02 ± 4.09E+02**
F25	2.10E+02 ± 3.25E-01	=	2.10E+02 ± 4.15E-01	**2.13E+02 ± 1.98E+00**	–	2.14E+02 ± 6.58E+00
$w/t/l$	12/6/7		–	14/6/5		–

"+", "–", 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 12: Comparison on the Error Values Between JADE-s4 and R_{cr}-JADE-s4 for All Functions at $D = 30$ with $NP = 50$ and $NP = 200$, Respectively.

Prob	$NP = 50$			$NP = 200$		
	JADE-s4		R_{cr}-JADE-s4	JADE-s4		R_{cr}-JADE-s4
F01	0.00E+00 ± 0.00E+00	=	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	=	0.00E+00 ± 0.00E+00
F02	2.17E+02 ± 4.31E+02	+	**1.48E+01 ± 1.05E+02**	1.33E+03 ± 2.47E+03	+	**1.53E-28 ± 1.02E-28**
F03	2.55E+06 ± 4.26E+06	+	**8.47E+03 ± 6.05E+03**	1.54E+06 ± 4.02E+06	+	**1.81E+03 ± 2.51E+03**
F04	4.80E+02 ± 1.69E+03	=	**1.72E-02 ± 1.21E-01**	2.41E+03 ± 4.01E+03	+	**5.44E-21 ± 3.81E-20**
F05	8.48E+02 ± 1.16E+03	+	**2.19E+02 ± 2.54E+02**	9.71E+01 ± 4.81E+02	+	**5.81E-04 ± 2.18E-03**
F06	1.61E+01 ± 3.19E+01	+	**1.28E+00 ± 1.88E+00**	1.23E-22 ± 8.69E-22	=	**1.19E-26 ± 3.57E-26**
F07	**7.93E-03 ± 5.96E-03**	=	1.03E-02 ± 9.04E-03	7.15E-03 ± 4.35E-03	+	**2.17E-03 ± 4.21E-03**
F08	2.09E+01 ± 2.51E-01	+	**2.04E+01 ± 3.99E-01**	2.09E+01 ± 5.23E-02	=	2.09E+01 ± 1.54E-01
F09	0.00E+00 ± 0.00E+00	=	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	=	0.00E+00 ± 0.00E+00
F10	**2.75E+01 ± 6.95E+00**	–	3.23E+01 ± 9.45E+00	3.69E+01 ± 1.93E+01	+	**2.11E+01 ± 6.50E+00**
F11	2.78E+01 ± 4.37E+00	+	**2.36E+01 ± 4.02E+00**	2.81E+01 ± 5.12E+00	+	**1.16E+01 ± 4.06E+00**
F12	5.98E+03 ± 4.40E+03	+	**3.15E+03 ± 4.44E+03**	2.26E+04 ± 4.90E+03	+	**1.00E+04 ± 1.10E+04**
F13	**1.18E+00 ± 1.02E-01**	–	1.27E+00 ± 1.21E-01	**2.18E+00 ± 1.85E-01**	–	2.34E+00 ± 1.83E-01
F14	1.23E+01 ± 7.64E-01	+	**1.19E+01 ± 8.78E-01**	1.28E+01 ± 2.53E-01	+	**1.12E+01 ± 7.52E-01**
F15	3.22E+02 ± 8.87E+01	=	**3.12E+02 ± 1.12E+02**	3.60E+02 ± 5.71E+01	=	**3.58E+02 ± 6.73E+01**
F16	**1.31E+02 ± 1.58E+02**	–	1.36E+02 ± 1.50E+02	5.82E+01 ± 2.43E+01	+	**3.97E+01 ± 1.42E+01**
F17	1.50E+02 ± 1.37E+02	+	**1.25E+02 ± 1.38E+02**	7.62E+01 ± 6.35E+01	+	**4.58E+01 ± 5.16E+01**
F18	**9.04E+02 ± 2.68E+01**	–	9.07E+02 ± 2.76E+01	9.08E+02 ± 2.05E+00	=	9.08E+02 ± 1.79E+00
F19	**9.07E+02 ± 2.25E+01**	–	9.13E+02 ± 4.16E+00	9.08E+02 ± 1.85E+00	=	9.08E+02 ± 1.70E+00
F20	**9.07E+02 ± 2.21E+01**	–	9.14E+02 ± 4.18E+00	9.08E+02 ± 1.94E+00	=	9.08E+02 ± 1.83E+00
F21	5.31E+02 ± 1.14E+02	=	**5.06E+02 ± 4.24E+01**	5.00E+02 ± 0.00E+00	=	5.00E+02 ± 0.00E+00
F22	**8.96E+02 ± 2.03E+01**	=	9.02E+02 ± 2.26E+01	8.90E+02 ± 1.61E+01	+	**8.79E+02 ± 2.59E+01**
F23	5.34E+02 ± 9.86E-03	=	5.34E+02 ± 1.73E-02	5.34E+02 ± 2.13E-04	+	5.34E+02 ± 2.88E-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
F25	2.10E+02 ± 6.77E-01	=	2.10E+02 ± 3.69E-01	2.09E+02 ± 3.26E-02	=	2.09E+02 ± 3.76E-02
$w/t/l$	9/10/6		–	13/11/1		–

"+", "–", 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 13: Comparison on the Error Values Between JADE-s4 and R_{cr}-JADE-s4 for All Functions at $D = 30$ with Different Initial μ_F Values.

Prob	$\mu_F = 0.1$		$\mu_F = 0.6$		$\mu_F = 0.9$	
	JADE-s4	R_{cr}-JADE-s4	JADE-s4	R_{cr}-JADE-s4	JADE-s4	R_{cr}-JADE-s4
F01	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
F02	**3.59E-28 ± 1.99E-28** =	4.13E-28 ± 1.99E-28	1.31E+03 ± 1.89E+03 +	**3.96E-28 ± 2.34E-28**	3.79E+03 ± 2.14E+03 +	**7.33E+02 ± 1.61E+03**
F03	1.70E+05 ± 1.11E+06 +	**1.33E+04 ± 8.32E+03**	4.57E+06 ± 5.64E+06 +	**1.44E+04 ± 9.71E+03**	1.17E+07 ± 4.83E+06 +	**2.72E+06 ± 5.35E+06**
F04	5.44E+02 ± 2.08E+03 +	**1.55E-11 ± 1.01E-10**	1.66E+03 ± 3.31E+03 +	**1.57E+02 ± 1.11E+03**	8.67E+03 ± 5.22E+03 +	**1.76E+03 ± 3.97E+03**
F05	1.12E-01 ± 2.96E-01 =	**3.71E-02 ± 7.32E-02**	8.13E+02 ± 1.27E+03 +	**2.53E+00 ± 1.49E+01**	2.88E+03 ± 1.30E+03 +	**1.41E-01 ± 3.76E-01**
F06	5.97E+00 ± 2.17E+01 =	**6.38E-01 ± 1.48E+00**	6.84E+00 ± 2.34E+01 =	**7.97E-02 ± 5.64E-01**	2.27E+01 ± 2.31E+01 +	**1.59E-01 ± 7.89E-01**
F07	**6.31E-03 ± 5.25E-03** =	6.79E-03 ± 9.82E-03	7.98E-03 ± 5.46E-03 =	**7.04E-03 ± 7.57E-03**	8.03E-03 ± 5.98E-03 =	**7.54E-03 ± 6.68E-03**
F08	2.09E+01 ± 2.16E-01 +	**2.03E+01 ± 4.31E-01**	2.09E+01 ± 1.44E-01 +	**2.03E+01 ± 4.35E-01**	2.09E+01 ± 2.32E-01 +	**2.04E+01 ± 4.50E-01**
F09	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
F10	2.99E+01 ± 8.36E+00 =	**2.95E+01 ± 8.86E+00**	2.85E+01 ± 6.16E+00 +	**2.34E+01 ± 6.82E+00**	2.95E+01 ± 1.04E+01 +	**2.45E+01 ± 9.11E+00**
F11	2.61E+01 ± 4.53E+00 +	**1.79E+01 ± 5.60E+00**	2.67E+01 ± 5.69E+00 +	**1.99E+01 ± 6.60E+00**	2.80E+01 ± 4.85E+00 +	**1.90E+01 ± 4.80E+00**
F12	3.08E+03 ± 3.87E+03 +	**1.17E+03 ± 1.76E+03**	1.52E+04 ± 3.91E+03 +	**1.21E+04 ± 8.05E+03**	**1.83E+04 ± 3.83E+03** –	2.05E+04 ± 4.52E+03
F13	**1.52E+00 ± 1.02E-01** –	1.63E+00 ± 1.23E-01	**1.56E+00 ± 1.45E-01** –	1.73E+00 ± 1.19E-01	**1.63E+00 ± 1.02E-01** –	1.70E+00 ± 1.19E-01
F14	1.23E+01 ± 9.00E-01 +	**1.12E+01 ± 7.25E-01**	1.26E+01 ± 5.61E-01 +	**1.13E+01 ± 1.00E+00**	1.25E+01 ± 8.27E-01 +	**1.14E+01 ± 9.32E-01**
F15	**3.36E+02 ± 6.31E+01** =	3.40E+02 ± 8.57E+01	**3.56E+02 ± 6.75E+01** =	3.66E+02 ± 8.46E+01	**3.12E+02 ± 1.08E+02** –	3.53E+02 ± 9.81E+01
F16	1.07E+02 ± 1.35E+02 =	**9.17E+01 ± 1.26E+02**	8.59E+01 ± 1.15E+02 =	**7.26E+01 ± 9.68E+01**	6.49E+01 ± 4.05E+01 +	**4.27E+01 ± 9.01E+00**
F17	**1.11E+02 ± 1.18E+02** =	1.13E+02 ± 1.33E+02	1.00E+02 ± 1.05E+02 +	**7.57E+01 ± 1.03E+02**	9.30E+01 ± 4.64E+01 +	**4.78E+01 ± 2.08E+01**
F18	8.94E+02 ± 4.14E+01 =	**8.86E+02 ± 4.88E+01**	**9.09E+02 ± 1.90E+00** =	9.10E+02 ± 1.89E+00	9.11E+02 ± 1.59E+00 +	**9.09E+02 ± 1.84E+00**
F19	8.94E+02 ± 4.14E+01 =	**8.91E+02 ± 4.60E+01**	9.09E+02 ± 1.98E+00 =	9.09E+02 ± 1.95E+00	9.10E+02 ± 1.90E+00 +	**9.08E+02 ± 2.08E+00**
F20	8.94E+02 ± 4.14E+01 =	**8.93E+02 ± 4.44E+01**	9.09E+02 ± 1.87E+00 =	9.09E+02 ± 2.07E+00	9.10E+02 ± 1.85E+00 +	**9.09E+02 ± 2.07E+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	**8.70E+02 ± 1.58E+01** =	8.71E+02 ± 1.66E+01	8.86E+02 ± 4.06E+01 +	**8.63E+02 ± 1.24E+01**	9.54E+02 ± 2.99E+01 +	**8.85E+02 ± 4.05E+01**
F23	5.34E+02 ± 3.53E-04 =	5.34E+02 ± 1.34E-02	5.34E+02 ± 3.44E-04 +	5.34E+02 ± 3.86E-04	5.34E+02 ± 2.68E-04 =	5.34E+02 ± 4.11E-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	2.13E+02 ± 2.30E+00 =	**2.09E+02 ± 5.03E-01**	2.09E+02 ± 3.09E-01 =	2.09E+02 ± 1.26E-01	2.09E+02 ± 6.23E-01 =	2.09E+02 ± 2.33E-01
$w/t/l$	6/18/1	–	12/12/1	–	15/7/3	–

"+", "–", 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 15: Comparison on the Performance of Different DE Variants in Five Real-World Problems.

Prob	NFFEs	jDE		SaDE		CoDE		JADE-s4		R_{cr}-JADE-s4
P1	10,000	1.16E+04 ± 4.60E+03	+	2.80E+02 ± 1.16E+02	+	**1.41E+02 ± 1.03E+02**	+	5.77E+03 ± 3.11E+03	+	**2.93E+00 ± 3.64E+00**
	150,000	**0.00E+00 ± 0.00E+00**		**0.00E+00 ± 0.00E+00**		**0.00E+00 ± 0.00E+00**		1.10E+01 ± 7.76E+01		**0.00E+00 ± 0.00E+00**
P2	10,000	2.09E+01 ± 2.43E+00	+	2.14E+01 ± 2.18E+00	+	**1.86E+01 ± 2.72E+00**	+	2.07E+01 ± 1.96E+00	+	**2.00E+01 ± 2.54E+00**
	150,000	**2.82E-01 ± 5.21E-01**		5.50E-01 ± 4.64E-01		9.34E-01 ± 3.29E+00		7.72E-01 ± 1.07E+00		**3.58E-02 ± 2.01E-01**
P3	150,000	1.32E+00 ± 9.25E-02	+	1.95E+00 ± 9.78E-02	+	1.23E+00 ± 1.62E-01	+	**1.21E+00 ± 1.71E-01**	+	**9.02E-01 ± 4.23E-01**
P4	10,000	4.89E+02 ± 1.18E+02	+	**3.36E+01 ± 7.39E+00**	+	3.83E+01 ± 1.46E+01	+	2.60E+02 ± 6.46E+01	+	**8.27E+00 ± 3.43E+00**
	150,000	5.87E-07 ± 2.16E-06		**0.00E+00 ± 0.00E+00**		**6.86E-14 ± 3.69E-13**		1.53E+00 ± 1.08E+01		**0.00E+00 ± 0.00E+00**
P5	150,000	**-2.16E+01 ± 1.75E-01**	=	**-2.17E+01 ± 1.40E-01**	–	-1.84E+01 ± 1.85E+00	+	-1.98E+01 ± 1.43E+00	+	-2.14E+01 ± 4.49E-01
$w/t/l$		4/1/0		4/0/1		5/0/0		5/0/0		–

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