



Research article

Differential evolution with quasi-reflection-based mutation

Wei Li and Wenyin Gong*

School of Computer Science, China University of Geosciences, Wuhan 430074, China

* **Correspondence:** Email: wygong@cug.edu.cn; Tel: +86-2767883716.

Abstract: Differential evolution (DE) is one of the most successful evolutionary algorithms. However, the performance of DE is significantly influenced by its mutation strategies. Generally, different mutation strategies may obtain different search directions. The improper search direction misleads the search and results in the poor performance of DE. Therefore, it is vital to consider the search direction when designing new mutation strategies. Based on this consideration, in this paper, the quasi-reflection-based mutation is proposed to enhance the performance of DE. The quasi-reflection-based mutation is able to provide the promising search direction to guide the search. To extensively evaluate the performance of our approach, 30 benchmark functions are chosen as the test suite. Combined with SHADE, Re-SHADE is presented. Compared with different advanced DE methods, Re-SHADE can obtain better results in terms of the accuracy and the convergence rate. Additionally, further experiments on the CEC2013 test suite also confirm the effectiveness of the proposed method.

Keywords: differential evolution; quasi-reflection-based mutation; search direction; numerical optimization

1. Introduction

Differential evolution (DE) is a simple and powerful evolutionary algorithm (EA) for numerical optimization problems [1] proposed by Storn and Price in 1997. After that, DE has obtained considerable attention in the field of evolutionary computation (EC) community [2]. Due to its superior performance, DE has been applied to diverse fields in recent years [3, 4]. Although DE has been extensively studied in the last few years, it is still an active research direction in EC [5].

In DE, the core operator is the mutation, which is able to control the search direction and the stepsize. In [1], the “DE/rand/1” mutation was presented, where the base vector is perturbed by one scaled difference vector. In [6], other mutation strategies were proposed, such as “DE/best/1”, “DE/target-to-best/1”. Generally, different mutation strategies have different features and are suitable to different problems. As reviewed in [5], there are other mutation strategies proposed in the literature

to improve the performance of DE [7]. Although there are various mutation strategies, most of them do not carefully consider the search direction generated by the mutation.

In 1965, Nelder and Mead proposed the famous Nelder-Mead (N-M) algorithm or simplex search algorithm for the unconstrained optimization problems without derivatives [8]. In the N-M algorithm, one important operation is the *reflection*, which generates a reflection point away from the worst solution in the simplex. In this way, the reflection can possibly provide the search direction towards the promising area.

Inspired by the reflection in the N-M algorithm, in this paper, we propose a quasi-reflection-based mutation to further enhance the performance of DE. The proposed mutation is able to provide the promising search direction to guide the search. It is simple to be implemented. Additionally, it is readily to be integrated into different DE methods. To extensively evaluate its performance, the proposed mutation is first combined with SHADE [9]. The corresponding method is referred to as ReSHADE. Thirty benchmark functions are chosen from the literature as the test suite. ReSHADE is compared with other advanced DE variants. The results indicate the superior performance of our approach. To further study the performance of the proposed mutation, it is also integrated into other DE methods. The results also show its promising performance for improving other DE methods. In addition, further experiments on the CEC2013 test suite also confirm the effectiveness of the proposed method.

The main contributions of this paper are two-fold:

- A quasi-reflection-based mutation is proposed to enhance the performance of DE. The mutation is able to provide the promising search direction.
- The proposed mutation is combined with different DE methods, and obtains superior performance on the benchmark problems.

The rest of the paper is organized as follow. The background is briefly described in section 2. Section 3 introduces the proposed method in detail. In section 4, extensively experiments are conducted to show the performance of our approach. Finally, section 5 concludes the paper.

2. Background

In this section, first, the classical DE method is briefly introduced, followed by the description of the reflection operation in the N-M algorithm. In addition, the related work for the mutation of DE is also briefly described.

Without loss of generality, in this work, the following unconstrained numerical optimization problem is considered:

$$\min f(\mathbf{x}), \quad \mathbf{x} \in \mathcal{S}, \quad (2.1)$$

where $\mathcal{S} \in \mathbb{R}^n$ is a compact set, and $\mathbf{x} = (x_1, \dots, x_n)^T$ is a vector of decision variables with the dimension of n . For each variable, it satisfies the boundary constraint as follows:

$$x_j \in [\underline{x}_j, \bar{x}_j], \quad j = 1, \dots, n,$$

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

2.1. Differential evolution

In DE, there are four main operations to evolve the population towards the optimal solution for the optimization problem, that is, *initialization*, *mutation*, *crossover*, and *selection* [1]. First, the initial population is generated. After that, DE repeatedly executes the mutation, crossover, and selection, until the termination criterion is met. The four operations are introduced as follows.

2.1.1. Initialization

Suppose that there are μ solutions ($\mathbf{x}_1, \dots, \mathbf{x}_\mu$) in the DE population. Initially, for each variable of a solution \mathbf{x}_i , it is randomly initialized in its search range as

$$x_{i,j} = \text{rndreal}(\underline{x}_j, \bar{x}_j), \quad (2.2)$$

where $\text{rndreal}(\underline{x}_j, \bar{x}_j)$ generates a uniformly random real-valued number in $[\underline{x}_j, \bar{x}_j]$, $i = 1, \dots, \mu$, and $j = 1, \dots, n$.

2.1.2. Mutation

After initializing the population, the mutation is used to generate the mutant vector \mathbf{v}_i for each target solution \mathbf{x}_i . The classical “DE/rand/1” mutation is as follows:

$$\mathbf{v}_i = \mathbf{x}_{r1} + F \cdot (\mathbf{x}_{r2} - \mathbf{x}_{r3}), \quad (2.3)$$

where $r1, r2$, and $r3$ are randomly generated integers in $\{1, \mu\}$, satisfy $r1 \neq r2 \neq r3 \neq i$. $F \in (0, 1+)$ is a scaling factor. In DE, there are also other mutation strategies, such as

- “DE/best/1”:

$$\mathbf{v}_i = \mathbf{x}_{\text{best}} + F \cdot (\mathbf{x}_{r2} - \mathbf{x}_{r3}),$$

- “DE/current-to-best/1”:

$$\mathbf{v}_i = \mathbf{x}_i + F \cdot (\mathbf{x}_{\text{best}} - \mathbf{x}_i) + F \cdot (\mathbf{x}_{r2} - \mathbf{x}_{r3}),$$

- “DE/current-to-rand/1”:

$$\mathbf{v}_i = \mathbf{x}_i + F \cdot (\mathbf{x}_{r1} - \mathbf{x}_i) + F \cdot (\mathbf{x}_{r2} - \mathbf{x}_{r3}),$$

where \mathbf{x}_{best} is the best solution in the current population.

2.1.3. Crossover

In DE, the crossover is performed between the mutant vector \mathbf{v}_i and the target vector \mathbf{x}_i . The commonly used binomial crossover is formulated as:

$$u_{i,j} = \begin{cases} v_{i,j}, & \text{if } \text{rndreal}(0, 1) \leq Cr \text{ or } j == j_{\text{rand}} \\ x_{i,j}, & \text{otherwise} \end{cases}, \quad (2.4)$$

where $Cr \in [0, 1]$ is the crossover rate, and $j_{\text{rand}} \in \{1, \mu\}$ is a randomly generated integer.

2.1.4. Selection

For each target solution \mathbf{x}_i , DE uses the mutation and crossover to generate a trial vector \mathbf{u}_i . Then, the one-to-one tournament selection is used to select the better one between \mathbf{x}_i and \mathbf{u}_i to survive into the next population as follows.

$$\mathbf{x}_i = \begin{cases} \mathbf{u}_i, & \text{if } f(\mathbf{u}_i) \leq f(\mathbf{x}_i) \\ \mathbf{x}_i, & \text{otherwise} \end{cases}, \quad (2.5)$$

where $f(\mathbf{x}_i)$ is the objective function value of \mathbf{x}_i .

2.2. Reflection in N-M algorithm

In the N-M algorithm [8], to implement the simplex transformation, one of the main operations is the reflection. For the n -dimensional optimization problem, $n + 1$ solutions ($\mathbf{x}_1, \dots, \mathbf{x}_{n+1}$) are initially generated to form the simplex. Among these solutions, the worst one is marked as \mathbf{x}_w , then, the centroid \mathbf{x}_c of the *best* side is calculated as

$$\mathbf{x}_c = \frac{1}{n} \sum_{j \neq w} \mathbf{x}_j, \quad j = 1, \dots, n + 1. \quad (2.6)$$

After calculating the centroid, the reflection point is calculated as

$$\mathbf{x}_r = \mathbf{x}_c + \alpha \cdot (\mathbf{x}_c - \mathbf{x}_w), \quad (2.7)$$

where α is the reflection factor. The reflection operation generates the reflection point in the best side of the simplex, which is possible the promising direction towards the basin of attraction.

2.3. Related work

As mentioned above, the core operation in DE is the mutation, which is able to control the search direction and the stepsize. There are various mutation strategies proposed in the literature [2]. More details can be found in the survey paper in [5]. Herein, we only briefly introduce some related work.

2.3.1. DE with p best mutation

In [10, 11], the JADE algorithm was proposed, where four mutation strategies were developed, that is, “DE/current-to- p best/1” without archive, “DE/current-to- p best/1” with archive, “DE/rand-to- p best/1” without archive, and “DE/rand-to- p best/1” with archive. Let “DE/current-to- p best/1” without archive and “DE/rand-to- p best/1” without archive be examples, they are formulated as

- “DE/current-to- p best/1” without archive:

$$\mathbf{v}_i = \mathbf{x}_i + F_i \cdot (\mathbf{x}_{pbest} - \mathbf{x}_i) + F_i \cdot (\mathbf{x}_{r1} - \mathbf{x}_{r2}), \quad (2.8)$$

- “DE/rand-to- p best/1” without archive:

$$\mathbf{v}_i = \mathbf{x}_{r0} + F_i \cdot (\mathbf{x}_{pbest} - \mathbf{x}_{r0}) + F_i \cdot (\mathbf{x}_{r1} - \mathbf{x}_{r2}), \quad (2.9)$$

where $r_0, r_1, r_2 \in \{1, \mu\}$, and $r_0 \neq r_1 \neq r_2 \neq i$. F_i is generated for each target solution. \mathbf{x}_{pbest} is the p best solution, which is randomly selected as one of the top $100p\%$ solutions in the current population \mathbb{P} , and $p \in (0, 1]$.

When the archive \mathbb{A} is used, the solution \mathbf{x}_{r_2} is randomly chosen from the union of the current population and the archive, that is, $\mathbf{x}_{r_2} \in \mathbb{P} \cup \mathbb{A}$.

2.3.2. DE with adaptive guided mutation

Inspired by the p best mutation, in [12], a new mutation strategy was presented as follows:

$$\mathbf{v}_i = \mathbf{x}_r + F \cdot (\mathbf{x}_{pbest} - \mathbf{x}_{pworst}), \quad (2.10)$$

where \mathbf{x}_{pbest} and \mathbf{x}_{pworst} are respectively randomly selected as one of the top and bottom $100p\%$ solutions in the current population. \mathbf{x}_r is randomly chosen from the middle $\mu - 2(100p\%)$ solutions, and $p \in (0, 0.5)$.

2.3.3. DE with order-based mutation

In [13], two mutation strategies were proposed to improve the performance of SHADE [9] and LSHADE [14].

- “DE/current-to-ord_best/1”:

$$\mathbf{v}_i = \mathbf{x}_i + F_i \cdot (\mathbf{x}_{obest} - \mathbf{x}_i) + F_i \cdot (\mathbf{x}_{omedian} - \mathbf{x}_{oworst}), \quad (2.11)$$

where \mathbf{x}_{obest} , $\mathbf{x}_{omedian}$, and \mathbf{x}_{oworst} are the best, median, and worst solutions from three randomly selected solutions, respectively.

- “DE/current-to-ord_pbest/1”:

$$\mathbf{v}_i = \mathbf{x}_i + F_i \cdot (\mathbf{x}_{opbest} - \mathbf{x}_i) + F_i \cdot (\mathbf{x}_{opmedian} - \mathbf{x}_{opworst}). \quad (2.12)$$

Similar to “DE/current-to- p best/1” mutation, first, two solutions are randomly chosen, and one solution is randomly selected as one of the top $100p\%$ solution in the current population. Then, the three solutions are ordered based on their fitness values, and the best, median, worst one are respectively marked as \mathbf{x}_{opbest} , $\mathbf{x}_{opmedian}$, and $\mathbf{x}_{opworst}$.

2.3.4. DE with past difference vectors reuse

Based on the consideration that the successful previous difference vectors may provide the successful directions of perturbation in DE mutation, in [15], Ghosh et al. proposed the past difference vectors reuse (DVR) mechanism. The successful difference vectors are saved in an archive \mathbb{A}_V , and they will be probabilistically reused to generate the mutant vectors in the subsequent generations. For example, the DVR-based “DE/rand/1” mutation can be formulated as

$$\mathbf{v}_i = \begin{cases} \mathbf{x}_{r_1} + F \cdot (\mathbf{x}_{r_2} - \mathbf{x}_{r_3}), & \text{if } g == 1 \text{ or } \text{rndreal}(0, 1) \geq p_{dvr} \\ \mathbf{x}_{r_1} + F \cdot \vec{\Delta}_k, & \text{otherwise} \end{cases}, \quad (2.13)$$

where g is the generation counter, p_{dvr} is the probability, and $\vec{\Delta}_k$ is a randomly chosen difference vector from \mathbb{A}_V .

3. Our approach

In this section, first, the motivations of our approach is introduced. Then, the proposed quasi-reflection-based mutation is described in detail, followed by the introduction of the framework of DE with the proposed mutation.

3.1. Motivations

In the DE literature, there are various mutation strategies proposed. However, most of them do not carefully consider the search direction of the mutation. In [15], the successful directions introduced by the successful difference vectors are used. The results in [15] indicate that carefully considering the search direction of the mutation is able to remarkably improve the performance of DE.

As mentioned in section 2.2, the reflection in the N-M algorithm explicitly uses the search direction, which generates the reflection point towards the *best* side of the simplex. Therefore, inspired by the reflection, we try to develop the quasi-reflection-based mutation to carefully consider the search direction.

3.2. Quasi-reflection-based mutation

Suppose that there are four solutions (i.e., \mathbf{x}_{r1} , \mathbf{x}_{r2} , \mathbf{x}_{r3} , and \mathbf{x}_{r4}) are randomly chosen from the current population. Among the these solutions, the best and worst solutions are respectively marked as \mathbf{x}_b and \mathbf{x}_w , and the rest two solutions are referred to as \mathbf{x}_{r1} and \mathbf{x}_{r2} . Then, the proposed mutation is formulated as

$$\mathbf{v}_i = \begin{cases} \mathbf{x}_c + F \cdot (\mathbf{x}_b - \mathbf{x}_c) + F \cdot (\mathbf{x}_{r0} - \mathbf{x}_w), & \text{if } f(\mathbf{x}_{r0}) \leq f(\mathbf{x}_w) \\ \mathbf{x}_c + F \cdot (\mathbf{x}_b - \mathbf{x}_c) + F \cdot (\mathbf{x}_w - \mathbf{x}_{r0}), & \text{otherwise} \end{cases}, \quad (3.1)$$

where

$$\mathbf{x}_c = \beta_i \cdot \mathbf{x}_{r1} + (1 - \beta_i) \cdot \mathbf{x}_{r2}, \quad (3.2)$$

where $\beta_i = \text{rndreal}(0, 1)$, which is regenerated for each target solution at the current generation. \mathbf{x}_{r0} is another randomly selected solution from the current population.

3.2.1. Explanations

For the proposed quasi-reflection-based mutation shown in Eq (3.1), there are three parts to generate the mutant.

- Similar to the reflection operation in the N-M algorithm, \mathbf{x}_c is calculated by Eq (3.2). When $\beta_i = 0.5$, it is the center between \mathbf{x}_{r1} and \mathbf{x}_{r2} . There are three differences compared with the reflection operation shown in Eq (2.7): i) \mathbf{x}_c is not restricted to the centroid to avoid the mutation favoring the optimal solution at $\mathbf{x}^* = (0, \dots, 0)^T$; ii) only four solutions are used to reduce the complexity; and iii) the best solution \mathbf{x}_b among the four solutions is not used to avoid trapping into the local optima, since \mathbf{x}_b is used in the second part.
- For the second part $F_i \cdot (\mathbf{x}_b - \mathbf{x}_c)$, it is inspired by the second part of “DE/rand-to-*p*best/1” shown in Eq (2.9). In addition, since \mathbf{x}_b is the best one among the four solutions, the direction of the

difference vector $\mathbf{x}_b - \mathbf{x}_c$ points to the *best* side of the four solutions similar to the reflection in the N-M algorithm.

- For the third part, a random solution \mathbf{x}_{r0} is used to enhance the diversity of the mutation. Additionally, to ensure the direction of the difference vector point to the better solution, the better one between \mathbf{x}_{r0} and \mathbf{x}_w is set to be the final vector as shown in Eq (3.1).

In the literature, there are some studies that combined DE with the N-M algorithm [16, 17]. However, our approach has significant differences compared with the existing methods:

- In [16], the N-M algorithm is used to generate the initial population for DE.
- In [17], the N-M algorithm is used as a local search procedure to refine the solutions.
- In our method, inspired by the reflection of the N-M algorithm, a quasi-reflection mutation is proposed. Therefore, it is not a simple hybridization of DE and N-M like the methods in [16, 17].

3.2.2. Remarks

The proposed quasi-reflection-based mutation has the following advantages:

- 1) *Promising Search Direction*: Similar to the reflection in the N-M algorithm, the proposed mutation is able to guide the search towards the best side of the selected solutions. In this way, it is able to generate *good* mutant vectors with higher probability.
- 2) *Simplicity*: It is very simple to be implemented. It also does not increase the complexity of the original DE method.
- 3) *Generality*: The proposed mutation is generic. Like “DE/current-to-*p*best/1”, \mathbf{x}_i and \mathbf{x}_{pbest} can be used to replace two of the four solutions. Also, similar to “DE/rand-to-*p*best/1”, \mathbf{x}_{pbest} can be used to replace one of the four solutions. Additionally, in Eq (3.1), more than four solutions can also be used to obtain \mathbf{x}_c , \mathbf{x}_b , and \mathbf{x}_w .

3.3. DE with quasi-reflection-based mutation

Combined the quasi-reflection-based mutation with the original DE method, the framework of ReDE is shown in Algorithm 1, where $NFEs_{max}$ is the maximal number of function evaluations, and $NFEs$ is the current number of function evaluations. Compared with DE using the “DE/rand/1” mutation, the only differences are in lines 7–10. Note that, the proposed quasi-reflection-based mutation can be readily integrated into other advanced DE variants. As substantiations, in this work, it is combined with jDE [18], JADE [10], and SHADE [9]. The corresponding variants are referred to as RejDE, ReJADE, and ReSHADE, respectively. In RejDE, the four solutions in Eq (3.1) are randomly chosen from the population. While in ReJADE and ReSHADE, one of the four solutions is the *p*best solution, and the rest three ones are randomly chosen from the population.

4. Experimental results and analysis

In this section, the performance of the proposed method is extensively evaluate to show its superiority.

Algorithm 1: The framework of ReDE**Input:** Control parameters: $\mu, Cr, F, NFEs_{\max}$ **Output:** The optimal solution

```

1 Initialize the population  $\mathbb{P}$  randomly;
2 Calculate the fitness value of each solution in  $\mathbb{P}$ ;
3 Set  $NFEs = \mu$ ;
4 while  $NFEs \leq NFEs_{\max}$  do
5   for  $i = 1$  to  $\mu$  do
6     Randomly select  $r1, r2, r3, r4 \in \{1, \mu\}$  with  $r1 \neq r2 \neq r3 \neq r4 \neq i$ ;
7     Find  $\mathbf{x}_b$  and  $\mathbf{x}_w$  from the four solutions;
8      $\beta_i = \text{rndreal}(0, 1)$ ;
9     Randomly select  $r0 \in \{1, \mu\}$ ;
10    Generate the mutant solution  $\mathbf{v}_i$  according to Equation (3.1);
11    Generate the trial solution  $\mathbf{u}_i$  according to Equation (2.4);
12    Evaluate the trial solution  $\mathbf{u}_i$ ;
13     $NFEs = NFEs + 1$ ;
14  for  $i = 1$  to  $\mu$  do
15    if  $f(\mathbf{u}_i) \leq f(\mathbf{x}_i)$  then
16       $\mathbf{x}_i = \mathbf{u}_i$ ;

```

4.1. Benchmark functions

To compare the performance of different algorithms, we first select 30 benchmark functions as the test suite from the literature [19–22]. The detailed description of these functions are given in Table S1 in the supplementary file. Each function with the optimal value $f(\mathbf{x}^*) = 0$ is tested at $n = 30$.

4.2. Algorithms for comparison and parameter settings

First, the proposed mutation is integrated into SHADE. The corresponding ReSHADE method is compared with the following advanced DE variants: 1) jDE [18], 2) JADE [10], 3) SHADE [9], 4) AGDE [12], 5) CoBiDE [23], and 6) OrSHADE [13]. The parameter settings of these algorithms are given in Table 1, which are mainly set to be the same as used in their original literature.

For each function, the maximal NFEs is $NFEs_{\max} = 100,000$. Each function is optimized over 51 independent runs.

Table 1. Parameter settings of the compared methods.

Algorithm	Parameter settings
jDE	$\mu = 100, \tau_1 = 0.1, \tau_2 = 0.1$
JADE	$\mu = 100, p = 0.05, c = 0.1, \xi_{Cr} = 0.5, \xi_F = 0.5$
SHADE, OrSHADE, ReSHADE	$\mu = 100, p_{\min} = 0.02, H_M = \mu$
AGDE	$\mu = 50, p = 0.1$
CoBiDE	$\mu = 60, pb = 0.4, ps = 0.5$

4.3. Compared with other DE variants

This section compares ReSHADE with the above-mentioned 7 DE variants in two aspects: the accuracy of the final solutions and the convergence rate.

4.3.1. On the accuracy

The detailed results are reported in Table S2 in the supplementary file, where the best results among the 8 algorithms are highlighted in grey boldface. In the last column, the significance is calculated by the Wilcoxon test at $\alpha = 0.05$, where “ $w/t/l$ ” indicates ReSHADE wins in w functions, ties in t functions, and loses in l functions compared with its competitor. Note that, in Table S2, when there are different algorithms can approach the optimal solution for a function, the *intermediate* results are also given at $NFEs = 6,600^*$. To better compare the performance of different algorithms, the multiple-problem statistical analysis is used. In Figure 1, the average rankings of different algorithms are shown. In addition, Table 2 reports the results obtained by the Wilcoxon test for ReSHADE vs other DE variants[†].

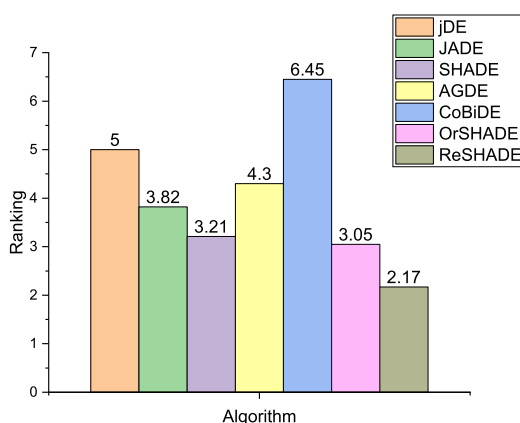


Figure 1. The average rankings of the eight algorithms by the Friedman test.

Table 2. Results obtained by the Wilcoxon test for ReSHADE vs other DE variants.

ReSHADE vs	R^+	R^-	p -value
jDE	430.0	35.0	7.99E-6
JADE	398.0	67.0	3.45E-4
SHADE	407.0	58.0	1.37E-4
AGDE	424.0	41.0	1.82E-5
CoBiDE	452.0	13.0	1.64E-7
OrSHADE	388.0	77.0	8.72E-4

From Table S2, we can observe that in 22 out of 30 functions ReSHADE obtains the best average results among the eight algorithms. Additionally, ReSHADE is able to get significant better results in

*For functions f06, f10–f13, and f23–f26, the intermediate results are reported in Table S2.

[†]The results of the Friedman and Wilcoxon tests are calculated by the KEEL software [24].

the majority of the functions according to the Wilcoxon test. The results in Figure 1 clearly indicates that ReSHADE yields the best ranking, followed by OrSHADE. Moreover, the results in Table 2 exhibits that ReSHADE significantly outperforms jDE, JADE, SHADE, AGDE, CoBiDE and OrSHADE based on the multiple-problem analysis of the Wilcoxon test.

4.3.2. On the convergence

In Figures 2 and 3, the convergence rate of different algorithm for the selected functions is illustrated, where the average convergence curve of each algorithm is shown[‡]. Since CoBiDE is the worst one among the eight algorithms, for the clarity, its convergence rate is not included herein.

From Figures 2, 3, S1 and S2, it can be seen that in the majority of the functions, ReSHADE converges the fastest from the begin of the evolution. OrSHADE is also an improved version of SHADE, its convergence rate is slightly faster than SHADE.

From the above the results on both the accuracy and the convergence rate, it is clear that ReSHADE consistently obtains the best results among the eight algorithms on the whole. This confirms that the proposed quasi-reflection-based mutation is effective and efficient. It can provide the promising search direction to guide the search, and hence, it results in the significant improvement of ReSHADE.

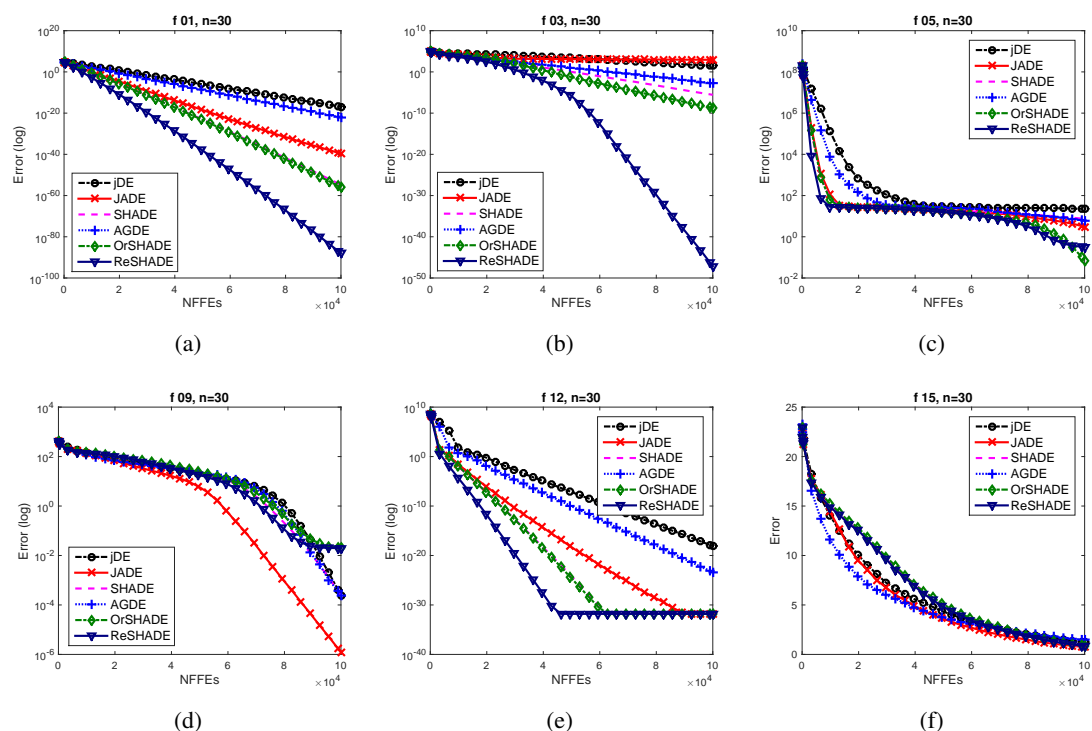


Figure 2. Comparison on the convergence rate of different algorithms for the selected functions. (a) f01, (b) f03, (c) f05, (d) f09, (e) f12 and (f) f15.

[‡]The convergence curves of the rest 18 functions are plotted in Figures S1 and S2 in the supplementary file.

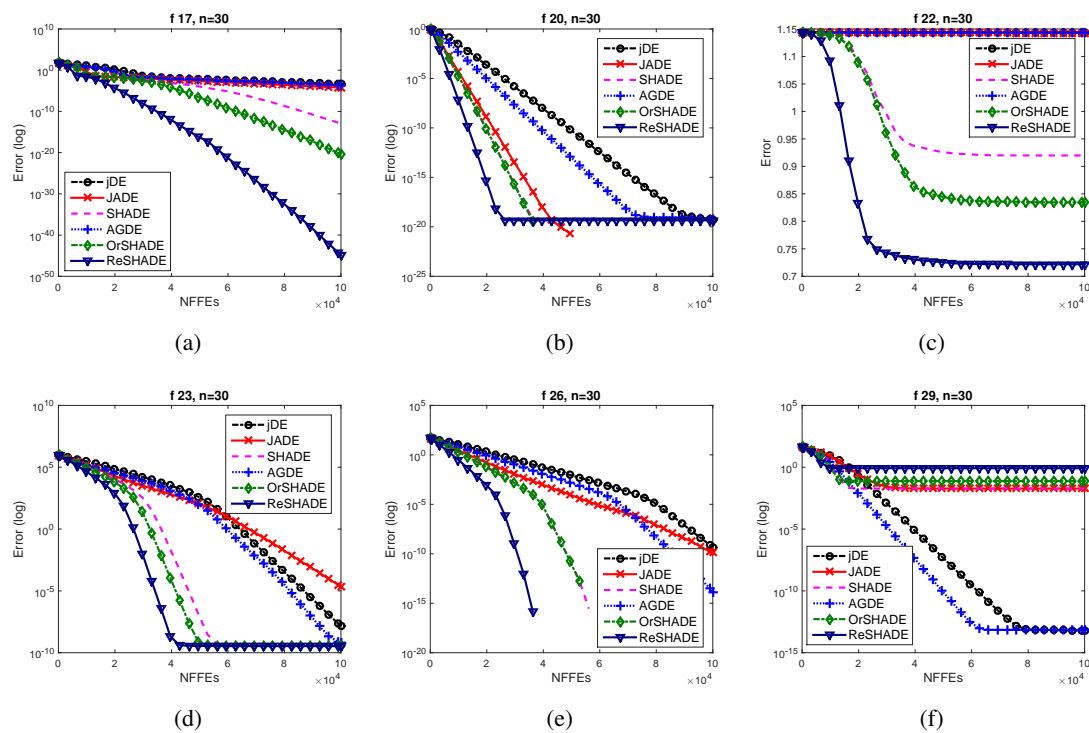


Figure 3. Comparison on the convergence rate of different algorithms for the selected functions. (a) f17, (b) f20, (c) f22, (d) f23, (e) f26 and (f) f29.

4.4. Study on other mutation strategies

In section 4.3, the proposed mutation is implemented in “DE/rand-to-*p*best/1” without archive. As mentioned in section 2.3.1, there are four mutation strategies presented in JADE [10, 11]. In this section, the quasi-reflection-based technique is also implemented into other three mutation strategies to evaluate its generality. Combined with SHADE, the detailed results are shown in Tables S3 and S4 in the supplementary file, where SHADE1 and ReSHADE1 use “rand-to-*p*best/1” without archive, SHADE2 and ReSHADE2 use “current-to-*p*best/1” without archive, SHADE3 and ReSHADE3 use “rand-to-*p*best/1” with archive, and SHADE4 and ReSHADE4 use “current-to-*p*best/1” with archive. In addition, the averaged rankings of SHADE and ReSHADE variants by the Friedman test is plotted in Figure 4. Table 3 reports the results obtained by the Wilcoxon test for ReSHADE vs SHADE with different mutation strategies.

Table 3. Results obtained by the Wilcoxon test for ReSHADE vs SHADE with different mutation strategies.

	R^+	R^-	p -value
ReSHADE1 vs SHADE1	407.0	58.0	1.37E-4
ReSHADE2 vs SHADE2	377.5	87.5	2.10E-3
ReSHADE3 vs SHADE3	397.0	68.0	3.80E-4
ReSHADE4 vs SHADE4	354.0	111.0	1.13E-2

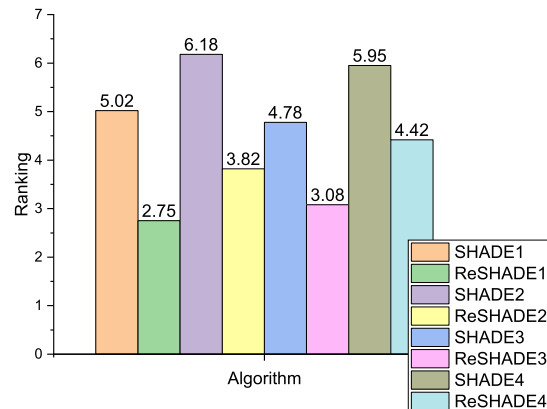


Figure 4. The average rankings of SHADE and ReSHADE with different mutation strategies by the Friedman test.

From the results in Tables S3, S4, 3 and Figure 4, we can see that

- In the majority of the functions, the proposed quasi-reflection-based mutation is able to obtain significant better results for different mutation strategies.
- Each ReSHADE method gets better ranking than its corresponding SHADE method.
- According to the multiple-problem analysis of the Wilcoxon test, ReSHADE significantly outperforms SHADE for different mutation strategies.

Therefore, we can conclude that the proposed quasi-reflection-based mutation has good generality. It may be of benefit to other DE mutation strategies to improve the performance of DE.

4.5. Effect on other DE methods

In the previous sections, the proposed mutation technique is combined with SHADE, and obtains very promising results compared with other DE variants. This section investigates the effect of the proposed mutation on other DE methods. It is integrated into jDE [18] and JADE [10], and the corresponding methods are referred to as RejDE and ReJADE, respectively. In RejDE, the four solutions in Eq (3.1) are randomly chosen from the current population. In ReJADE, the mutation strategy is the same as used in ReSHADE1. The parameters of RejDE and ReJADE are kept the same as the parameters of jDE and JADE in Table 1.

Table S5 in the supplementary file shows the detailed results of the four methods. The average rankings of these methods by the Friedman test are plotted in Figure 5. Additionally, the results obtained by the Wilcoxon test with the multiple-problem analysis are provided in Table 4. The results still substantiate the superiority of the proposed mutation, which can also significantly enhance the performance of other DEs (RejDE vs jDE and ReJADE vs JADE).

Table 4. Results obtained by the Wilcoxon test for RejDE vs jDE and ReJADE vs JADE.

	R^+	R^-	p -value
RejDE vs jDE	465.0	0.0	1.86E-9
ReJADE vs JADE	403.0	62.0	7.73E-5

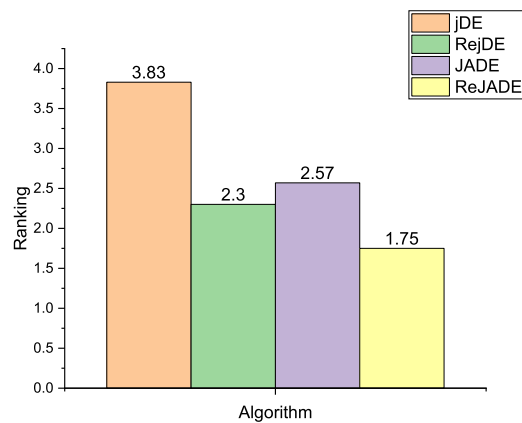


Figure 5. The average rankings of jDE, RejDE, JADE, and ReJADE by the Friedman test.

4.6. Results on the CEC2013 functions

In the previous experiments, the performance of different DE variants were evaluated on the functions shown in Table S1 in the supplementary file. To further study the performance of the proposed mutation, in this section, the CEC2013 functions [25] are used. There are 28 functions with three groups: unimodal functions (F01–F05), basic multimodal functions (F06–F20), and composition functions (F21–F28). In this work, all functions are tested at $n = 10$ with $NFE_{s_{\max}} = 100,000$. ReSHADE is compared with jDE, JADE, SHADE, AGDE, and OrSHADE. For the compared algorithms, the parameters are kept the same as presented in Table 1. Each function is executed over 51 independent runs.

In the supplementary file, Table S6 reports the detailed results of the compared DE variants. In addition, the average rankings of these algorithms are shown in Figure 6, and the results of the Wilcoxon test are given in Table 5. From the results, we observe that

Table 5. Results obtained by the Wilcoxon test for different algorithms on the CEC2013 functions.

ReSHADE vs	R^+	R^-	p -value
jDE	321.0	85.0	6.06E–3
JADE	338.0	68.0	1.42E–3
SHADE	253.0	153.0	≥ 0.2
AGDE	330.5	75.5	1.46E–3
OrSHADE	243.5	162.5	≥ 0.2

- Compared with jDE, JADE, and AGDE, ReSHADE obtains significantly better results in the majority of functions. It can also significantly outperform the three DE methods on the whole as shown in Table 5.
- Compared with SHADE and OrSHADE, ReSHADE wins in 11 and 10 functions, respectively. However, it loses in 3 functions. For the rest functions, there are no significant differences. Based on the Wilcoxon test shown in Table 5, ReSHADE still provides slightly better results than

SHADE and OrSHADE.

- On the whole, ReSHADE gets the best average ranking by the Friedman test among the six algorithms.

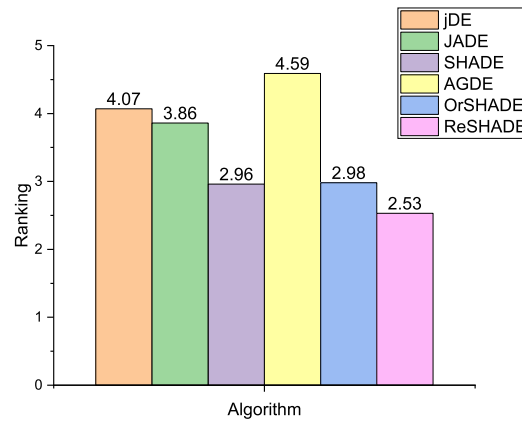


Figure 6. The average rankings of different algorithms on the CEC2013 functions by the Friedman test.

Therefore, it can be concluded that the proposed quasi-reflection-based mutation is also useful to improve the performance of DE on the complex optimization problems.

4.7. Results on the real-world problems

In this section, two real-world problems are chosen from the literature to verify the performance of our approach. The two problems are: i) RWP1) Chebychev polynomial fitting problem ($n = 9$) [6] and ii) RWP2) optimization of geophysical potential field data inversion ($n = 9$) [26]. The maximal NFEs $NFE_{s_{max}} = 30,000$. All other parameters are kept the same as shown in section 4.2. Each algorithm is executed over 51 independent runs for each problem. The results are reported in Table 6.

Table 6. Comparison on the results of different methods on two real-world problems.

Prob.	jDE	JADE	SHADE	AGDE	CoBiDE	OrSHADE	ReSHADE	ReJADE
RWP1	$6.8E+1 \pm 1.0E+2$	$0.0E+0 \pm 0.0E+0$	$2.2E-2 \pm 6.3E-2$	$9.7E-3 \pm 3.4E-2$	$6.5E-1 \pm 3.1E-1$	$3.7E-3 \pm 1.2E-2$	$1.3E-3 \pm 6.8E-3$	$0.0E+0 \pm 0.0E+0$
RWP2	$2.2E-3 \pm 1.2E-3$	$7.1E-4 \pm 1.5E-3$	$7.5E-5 \pm 7.9E-5$	$2.9E-5 \pm 2.7E-5$	$6.0E-4 \pm 3.9E-4$	$5.0E-5 \pm 6.7E-5$	$1.4E-5 \pm 2.2E-5$	$1.9E-4 \pm 6.6E-4$
ReSHADE vs	2/0/0	0/1/1	2/0/0	2/0/0	2/0/0	2/0/0	–	1/1/0
ReJADE vs	2/0/0	1/1/0	2/0/0	2/0/0	2/0/0	2/0/0	1/1/0	–

From the results in Table 6, it is clear that:

- For RWP1, both JADE and ReJADE get the global optimum. They perform significantly better than jDE, SHADE, AGDE, CoBiDE, OrSHADE, and ReSHADE. ReSHADE obtains the second best result.
- For RWP2, both ReSHADE and ReJADE significantly outperform jDE, SHADE, AGDE, CoBiDE, and OrSHADE.
- On the whole, the proposed ReSHADE and ReJADE obtain the best results on the two real-world problems.

Therefore, we can conclude that the proposed quasi-reflection mutation is also able to perform well on the real-world problems.

5. Conclusions

In this paper, inspired by the reflection operation in the N-M algorithm, the quasi-reflection-based mutation is proposed to improve the performance of DE. The proposed method is able to provide the promising search direction to guide the search. It is simple to be implemented, easy to be integrated into different DE variants, and generic to be extended to different mutation strategies. In this work, it is integrated into jDE, JADE, and SHADE as illustrations. Results on the benchmark functions and the CEC2013 functions clearly confirm the superiority of the quasi-reflection-based mutation.

In the near future, we will try to use the DE variants with the quasi-reflection-based mutation to solve the real-world optimization problems, such as the pollution isolation in water distribution network [27], mobile supplier selection problem [28], image segmentation [29].

The source code of ReSHADE can be obtained from the authors upon request.

Acknowledgments

This work was partly supported by the National Natural Science Foundation of China under Grant Nos. 62076225 and 62073300, the Natural Science Foundation for Distinguished Young Scholars of Hubei under Grant No. 2019CFA081, and the Fundamental Research Funds for the Central Universities, China University of Geosciences (Wuhan) under Grant No. CUGGC03.

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

1. R. Storn, K. Price, Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces, *J. Global Optim.*, **11** (1997), 341–359.
2. S. Das, P. Suganthan, Differential evolution: A survey of the state-of-the-art, *IEEE Trans. Evol. Comput.*, **15** (2011), 4–31.
3. S. Das, S. Mullick, P. Suganthan, Recent advances in differential evolution—An updated survey, *Swarm Evol. Comput.*, **27** (2016), 1–30.
4. J. Jang, K. Jang, H. Kwon, J. Lee, Feedback control of an HBV model based on ensemble kalman filter and differential evolution, *Math. Biosci. Eng.*, **15** (2018), 667–691.
5. M. Bilal, H. Zaheer, L. Garcia-Hernandez, A. Abraham, Differential evolution: A review of more than two decades of research, *Eng. Appl. Artif. Intell.*, **90** (2020), 103479.
6. K. Price, R. Storn, J. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*, Springer-Verlag, Berlin, 2005.

7. Y. Li, S. Wang, B. Yang, An improved differential evolution algorithm with dual mutation strategies collaboration, *Expert Syst. Appl.*, **153** (2020), 113451.
8. J. Nelder, R. Mead, A simplex method for function minimization, *Comput. J.*, **7** (1965), 308–313.
9. R. Tanabe, A. Fukunaga, Success-history based parameter adaptation for differential evolution, in *2013 IEEE Congress on Evolutionary Computation (CEC)*, 2013, 71–78.
10. J. Zhang, A. Sanderson, JADE: Adaptive differential evolution with optional external archive, *IEEE Trans. Evol. Comput.*, **13** (2009), 945–958, .
11. J. Zhang, A. Sanderson, *Adaptive Differential Evolution: A Robust Approach to Multimodal Problem Optimization*, Springer-Verlag, Berlin, 2009.
12. A. Mohamed, A. Mohamed, Adaptive guided differential evolution algorithm with novel mutation for numerical optimization, *Int. J. Mach. Learn. & Cyber.*, **10** (2019), 253–277.
13. A. Mohamed, A. Hadi, K. Jambi, Novel mutation strategy for enhancing SHADE and LSHADE algorithms for global numerical optimization, *Swarm Evol. Comput.*, **50** (2019), 100455.
14. R. Tanabe, A. Fukunaga, Improving the search performance of SHADE using linear population size reduction, in *2014 IEEE Congress on Evolutionary Computation (CEC)*, 2014, 1658–1665.
15. A. Ghosh, S. Das, A. K. Das, L. Gao, Reusing the past difference vectors in differential evolution - A simple but significant improvement, *IEEE Trans. on Cybern.*, **50** (2020), 4821–4834.
16. M. Ali, M. Pant, A. Abraham, Simplex differential evolution, *Acta Polytech. Hungarica.*, **6** (2009), 95–115.
17. Z. Gao, T. Xiao, W. Fan, Hybrid differential evolution and Nelder-Mead algorithm with re-optimization, *Soft Comput.*, **15** (2011), 581–594.
18. 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. Evol. Comput.*, **10** (2006), 646–657.
19. X. Yao, Y. Liu, G. Lin, Evolutionary programming made faster, *IEEE Trans. Evol. Comput.*, **3** (1999), 82–102.
20. S. Rahnamayan, H. Tizhoosh, M. Salama, Opposition-based differential evolution, *IEEE Trans. Evol. Comput.*, **12** (2008), 64–79.
21. M. Ali, C. Khompatporn, Z. Zabinsky, A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems, *J. Global Optim.*, **31** (2005), 635–672.
22. W. Gong, A. Zhou, Z. Cai, A multioperator search strategy based on cheap surrogate models for evolutionary optimization, *IEEE Trans. Evol. Comput.*, **19** (2015), 746–758.
23. Y. Wang, H. X. Li, T. Huang, L. Li, Differential evolution based on covariance matrix learning and bimodal distribution parameter setting, *Appl. Soft Comput.*, **18** (2014), 232–247.
24. J. Alcalá-Fdez, L. Sánchez, S. García, M. del Jesus, S. Ventura, J. M. Garrell, et al., KEEL: A software tool to assess evolutionary algorithms for data mining problems, *Soft Comput.*, **13** (2009), 307–318.

25. J. Liang, B. Qu, P. Suganthan, A. Hernández-Díaz, Problem definitions and evaluation criteria for the CEC 2013 special session and competition on real-parameter optimization, Technical Report 201212, Computational Intelligence Laboratory, Zhengzhou University, 2013.
26. N. Qiu, Q. Liu, Q. Gao, Q. Zeng, Combining genetic algorithm and generalized least squares for geophysical potential field data optimized inversion, *IEEE Geosci. Remote. Sens. Lett.*, **7** (2010), 660–664.
27. C. Hu, J. Cai, D. Zeng, X. Yan, W. Gong, L. Wang, Deep reinforcement learning based valve scheduling for pollution isolation in water distribution network, *Math. Biosci. Eng.*, **17** (2020), 105–121.
28. A. Alejo-Reyes, E. Olivares-Benitez, A. Mendoza, A. Rodriguez, Inventory replenishment decision model for the supplier selection problem using metaheuristic algorithms, *Math. Biosci. Eng.*, **17** (2020), 2016–2036.
29. X. Peng, H. Jia, C. Lang, Modified dragonfly algorithm based multilevel thresholding method for color images segmentation, *Math. Biosci. Eng.*, **16** (2019), 6467–6511.



AIMS Press

©2021 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)