



---

*Research article*

## **An efficient eigenvector-based crossover for differential evolution: Simplifying with rank-one updates**

**Tae Jong Choi\***

Graduate School of Data Science, Chonnam National University, Gwangju 61186, Republic of Korea

\* **Correspondence:** Email: [ctj17@jnu.ac.kr](mailto:ctj17@jnu.ac.kr); Tel: +82625305796; Fax: +82625305796.

**Abstract:** We propose a new approach to enhancing the efficiency of the differential evolution (DE) algorithm, specifically targeting rotational invariance. The performance of the DE algorithm can be hampered by the crossover's dependency on the coordinate system, particularly in optimization problems involving strongly correlated variables. Previous attempts to achieve rotational invariance in the DE algorithm have involved estimating the covariance matrix using the population's distribution information and executing the crossover operation in an eigen coordinate system. However, these methods are computationally intensive. Our approach exclusively employs the rank-one update method, estimating the covariance matrix using the means of the current and previous generations' populations. This lightweight technique reduces the computational costs from  $O(Np \cdot D^2)$  to  $O(D^2)$  (where  $Np$  is the population size and  $D$  is the dimension) operations, yet still preserves the critical rotational invariance property. Experiments conducted on 57 benchmark functions demonstrated that our method finds quicker and more accurate solutions than previous methods. This represents a substantial improvement in achieving rotational invariance in the DE algorithm.

**Keywords:** artificial intelligence; evolutionary algorithms; evolutionary computation; differential evolution; eigenvector-based crossover

**Mathematics Subject Classification:** 68T01, 68W50

---

### **1. Introduction**

Differential evolution (DE) [35, 39–41] is a widely used evolutionary algorithm (EA) designed to tackle optimization problems, particularly continuous optimization challenges. Compared with other EAs, DE is more straightforward and involves fewer control parameters, making it relatively easy to implement. Its high flexibility also allows for enhanced performance by integrating various techniques.

The DE algorithm operates through three main processes: mutation, crossover, and selection. These processes are governed by three control parameters: the amplification factor, crossover probability, and

population size. The performance of DE substantially depends on the interplay of these operations and parameters. The binomial crossover is commonly utilized due to its advantageous performance [1, 11, 12, 33, 34]. However, its heavy reliance on the coordinate system can limit the algorithm's effectiveness in solving problems involving variables with strong correlations [3, 22, 42].

Some researchers have proposed methods to make the DE algorithm rotationally invariant to address this limitation. These methods involve estimating the covariance matrix using population distribution information and conducting the crossover operation in an eigen coordinate system [15, 16, 28, 43, 44]. The underlying premise is that the population's distribution information highlights the promising regions of the function landscape. However, these techniques have not yet incorporated the rank-one update [20, 21, 23–25] for covariance matrix estimation and require extensive computational resources, which presents practical challenges. Consequently, further improvements are necessary for these approaches to be more effective and efficient.

This study proposes a straightforward yet effective method employing the rank-one update for estimating the covariance matrix. The method calculates a rank-one matrix using the means of the current previous generation's populations, which is then used to update the covariance matrix. This approach makes the DE algorithm rotationally invariant by forming a new eigen coordinate system based on the updated covariance matrix and applying this system to the crossover operation. Consequently, this method allows quicker updates of the covariance matrix, enabling faster crossover operations than previous methods.

We conducted experiments using 57 IEEE CEC benchmark functions [2, 27] to demonstrate the proposed method's performance. The proposed method was applied to two state-of-the-art DE algorithms [37, 38] in these experiments and compared against previous techniques. The results showed that the proposed method finds solutions faster and discovers more accurate or similar solutions than previous methods. Consequently, it was confirmed that using the rank-one update to make the DE algorithm rotationally invariant is more efficient than earlier approaches.

The main contributions of this study are as follows:

- While previous methods require  $O(Np \cdot D^2)$  operations to estimate the covariance matrix, the proposed method exclusively uses the rank-one update, thus requiring  $O(D^2)$  operations.  $Np$  and  $D$  denote the population size and dimensionality, respectively.
- The proposed method can be easily applied to the crossover operation of the DE algorithm.
- The performance of the proposed method was demonstrated by applying it to two state-of-the-art DE algorithms, NL-SHADE-LBC and NL-SHADE-RSP.

The structure of this paper is outlined as follows: Section 2 delves into related work, exploring the DE algorithm and eigenvector-based crossover operators. Section 3 introduces our proposed method, detailing its motivation and the procedures of the proposed eigenvector-based crossover operator. Section 4 describes the experimental settings, benchmark functions, and test algorithms. Section 5 discusses the results of our experiments, featuring comparisons with NL-SHADE-LBC and NL-SHADE-RSP, along with an algorithm complexity analysis. The paper concludes in Section 6, where we summarize our findings.

## 2. Related work

### 2.1. Differential evolution

The DE algorithm [35, 39–41] starts by initializing a population of  $Np$  target vectors, each with  $D$  real-valued parameters, represented as  $\vec{x}_{i,g} = (x_{1,i,g}, \dots, x_{D,i,g})$  for  $i = 1, \dots, Np$ . The subscript  $g$  denotes the generation to which the target vectors belong. After initialization, the algorithm randomly mutates the target vectors to produce a population of  $Np$  mutant vectors,  $\vec{v}_{i,g} = (v_{1,i,g}, \dots, v_{D,i,g})$  for  $i = 1, \dots, Np$ . These mutant vectors then undergo recombination with their corresponding target vectors, creating a population of  $Np$  trial vectors,  $\vec{u}_{i,g} = (u_{1,i,g}, \dots, u_{D,i,g})$  for  $i = 1, \dots, Np$ . If a trial vector has a function error value that is less than or equal to that of its target vector, it replaces the target vector. This process repeats until one of the termination criteria is met. The DE algorithm is highly flexible for optimizing continuous problems and can integrate various techniques to enhance performance [4–10, 17–19, 32, 46]. DE has been successfully applied to a variety of problems, including the development of a three-stage dynamic false data injection attack model for cyber-physical power systems [30], a stealthy sparse cyber-attack model for AC smart grids [29], and a multi-objective feature selection [45]. For a more comprehensive description of recent advancements in DE, please refer to the following survey papers [1, 11, 12, 33, 34, 36].

#### 2.1.1. Mutation

A mutant vector  $\vec{v}_{i,g}$  is generated for each target vector  $\vec{x}_{i,g}$  by forming a linear combination of other target vectors. Two commonly used mutation operators are described as follows:

- The DE/rand/1 is defined as  $\vec{v}_{i,g} = \vec{x}_{r1,g} + F \cdot (\vec{x}_{r2,g} - \vec{x}_{r3,g})$ .
- The DE/best/1 is defined as  $\vec{v}_{i,g} = \vec{x}_{best,g} + F \cdot (\vec{x}_{r1,g} - \vec{x}_{r2,g})$ .

Here, the subscripts  $r1$ ,  $r2$ , and  $r3$  denote distinct random integers selected from the set  $\{1, \dots, Np\}$ , each of which is different from  $i$ , and  $\vec{x}_{best,g}$  represents the target vector with the lowest function error value at generation  $g$ . The amplification factor  $F$  adjusts the magnitude of the mutation.

#### 2.1.2. Crossover

The DE algorithm commonly utilizes binomial crossover to generate a trial vector  $\vec{u}_{i,g}$ :

$$u_{j,i,g} = \begin{cases} v_{j,i,g}, & \text{if } \text{rand}_j(0, 1) \leq Cr \text{ or } j == j_{rand} \\ x_{j,i,g}, & \text{otherwise.} \end{cases} \quad (2.1)$$

$j_{rand}$  and  $\text{rand}_j(0, 1)$  refer to a random integer selected from the set  $\{1, \dots, D\}$  and a random number uniformly sampled from the interval  $[0, 1]$ , respectively.  $Cr$  denotes the crossover probability.

#### 2.1.3. Selection

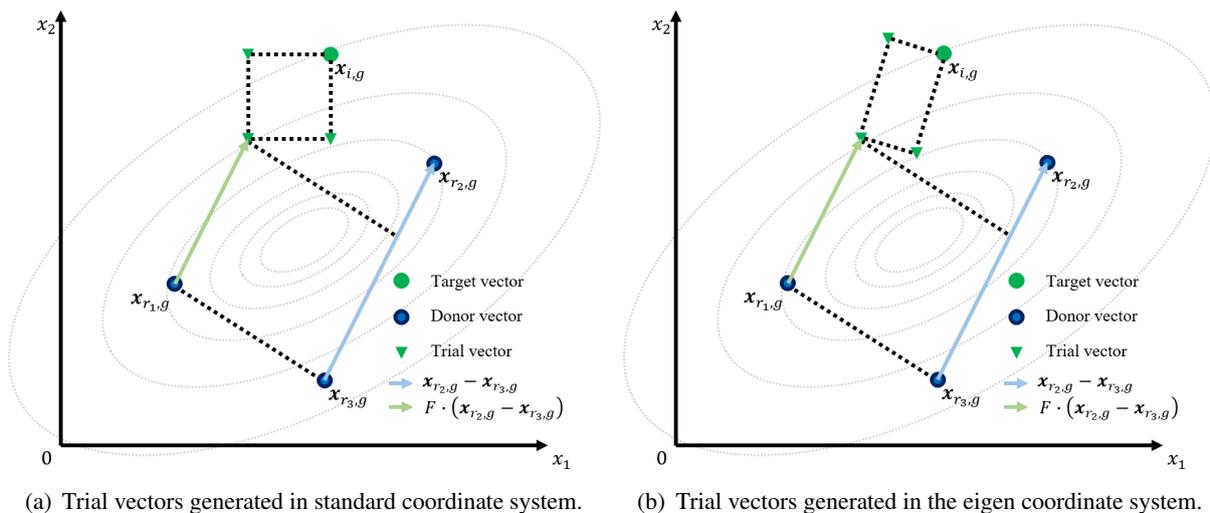
When the error value of a trial vector is less than or equal to that of its corresponding target vector, the trial vector replaces the target vector. Otherwise, the target vector remains unchanged.

$$\vec{x}_{i,g+1} = \begin{cases} \vec{u}_{i,g}, & \text{if } f(u_{i,g}) \leq f(x_{i,g}) \\ \vec{x}_{i,g}, & \text{otherwise.} \end{cases} \quad (2.2)$$

## 2.2. Eigenvector-based crossover operators

The DE algorithm may encounter difficulties when optimizing problems with strongly correlated variables [3, 22, 42]. Researchers have introduced techniques that estimate the covariance matrix from the population's distribution information to overcome this. These techniques utilize an eigen coordinate system for the crossover operation, ensuring the rotational invariance of the algorithm. This approach is grounded in the observation that the population's distribution information reflects promising regions within the function's landscape [3]. By establishing an eigen coordinate system derived from the covariance matrix, these methods effectively mitigate correlations among variables [16]. These strategies yield offspring distributions aligned with the function landscape, guiding the evolution of the population toward the global optimum [16]. Figure 1 illustrates the benefits of eigenvector-based crossover operators. These operators can be categorized according to three perspectives:

- (1) Whether they utilize the cumulative distribution information of the population to estimate the covariance matrix;
- (2) Whether they incorporate all individuals when utilizing the population's distribution information;
- (3) Whether the eigenvector ratio remains fixed or varies.



**Figure 1.** Eigenvector-based crossover operators can eliminate correlations between variables by constructing an eigen coordinate system using the covariance matrix. Consequently, such methods can generate an offspring distribution that aligns with the function's landscape, guiding the population's evolution toward the global optimum.

DE/eig [16] estimates the covariance matrix using the distribution information of the population in the current generation. When computing the population's distribution information, this method incorporates all individuals and employs a fixed eigenvector ratio. CoBiDE [43] estimates the covariance matrix using the distribution information of the current generation's population. This method includes the top  $ps\%$  individuals when calculating the population's distribution information and utilizes a fixed eigenvector ratio. CPI-DE [44] estimates the covariance matrix using the rank- $Np$

update, leveraging the cumulative distribution information of the population. This method incorporates the top half of all offspring when computing the distribution information. It generates offspring, half within the standard coordinate system and half within an eigen coordinate system, without relying on control parameters such as the eigenvector ratio. ACoS [28] also employs the rank- $Np$  update for covariance matrix estimation. This method includes offerings stored in the archive when calculating the population distribution information. Each individual has an eigenvector ratio, with these control parameters adjusted based on adaptive parameter control. STCS [15] utilizes the rank- $Np$  update to estimate the covariance matrix, incorporating superior individuals stored in the archive when computing the population distribution information. Each individual has an eigenvector ratio, and these control parameters are adjusted according to the policy gradient.

### 3. Proposed method

#### 3.1. Motivation

In the DE algorithm, the crossover operation is critical for generating offspring. However, the commonly used binomial crossover heavily depends on the coordinate system [3, 22, 42], necessitating improved rotational invariance to solve optimization problems involving highly correlated variables effectively. Previous methods [15, 16, 28, 43, 44] have attempted to address this by estimating the covariance matrix using the population's distribution information, applying eigen-decomposition to create a new eigen coordinate system, and performing the crossover operation within this system. Despite these efforts, several limitations persist:

- **DE/eig** and **CoBiDE**: These methods estimate the covariance matrix on the basis of the population's distribution information at the current generation, which may not accurately capture promising regions of the function landscape. Additionally, they require substantial computational resources ( $O(Np \cdot D^2)$ ), limiting their practicality.
- **CPI-DE**, **ACoS**, and **STCS**: These methods use the rank- $Np$  update to estimate the covariance matrix, better reflecting promising regions of the function landscape compared with DE/eig and CoBiDE. However, they do not utilize the rank-one update for covariance matrix estimation and still demand substantial computational resources ( $O(Np \cdot D^2)$ ). Moreover, ACoS and STCS maintain an archive to store individuals, requiring considerable memory space.

Therefore, a more efficient method is needed to update the covariance matrix and perform the crossover operation rapidly.

#### 3.2. Proposed eigenvector-based crossover operator

The proposed method uses the eigenvector ratio  $P$  as a control parameter to determine the proportion of crossover operations performed in the eigen coordinate system versus the standard coordinate system. When a random number uniformly sampled from  $[0, 1]$  is less than the eigenvector ratio, the eigen coordinate system performs the crossover operation. Otherwise, it is performed in the standard coordinate system. The proposed method executes the rank-one update following the selection operation. Figure 2 shows the flowchart of the proposed method, and Algorithm 1 presents the pseudo-code.

**Algorithm 1:** DE/ $r_1$ 

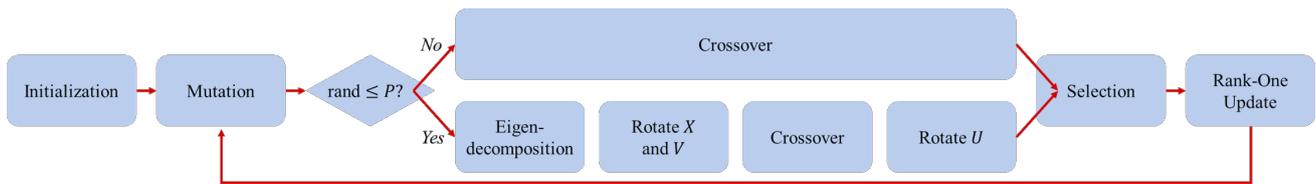
**Input** : Amplification factor  $F$ , Crossover probability  $Cr$ , Population size  $Np$ , Eigenvector ratio  $P$

**Output:** Best target vector  $\vec{x}_{\text{best}}$

```

1 Initialize target vectors;
2 for  $i = 1; i \leq Np; i = i + 1$  do
3   |  $w'_i = \ln(Np + 0.5) - \ln(i)$ ;
4 end
5 for  $i = 1; i \leq Np; i = i + 1$  do
6   |  $w_i = \frac{w'_i}{\sum_{i=1}^{Np} w'_i}$ ;
7 end
8  $\vec{C}_0 = \vec{I}$  and  $\vec{m}_0 = \frac{\sum_{i=1}^{Np} \vec{x}_{\text{best},i,0}}{Np}$ ;
9  $\mu_{\text{eff}} = \frac{(\sum_{i=1}^{Np} w_i)^2}{\sum_{i=1}^{Np} w_i^2}$ ,  $c_1 = \frac{2}{(D+1.3)^2 + \mu_{\text{eff}}}$ , and  $c_c = \frac{4 + \mu_{\text{eff}}/D}{D + 4 + 2 \cdot \mu_{\text{eff}}/D}$ ;
10  $g = 0$ ;
11 while None of the termination criteria is met do
12   Generate mutant vectors (Mutation);
13   if  $\text{rand}(0, 1) \leq P$  then
14     // Eigen coordinate system
15     Factor  $\vec{C}_g = \vec{B}\vec{D}^2\vec{B}^T$ ;
16     for  $i = 1; i \leq Np; i = i + 1$  do
17       | Rotate  $\vec{x}_{i,g} = \vec{B}^T \vec{x}_{i,g}$  and  $\vec{v}_{i,g} = \vec{B}^T \vec{v}_{i,g}$ ;
18     end
19     Generate trial vectors (Crossover);
20     for  $i = 1; i \leq Np; i = i + 1$  do
21       | Rotate  $\vec{u}_{i,g} = \vec{B}\vec{u}'_{i,g}$ ;
22     end
23   else
24     // Standard coordinate system
25     Generate trial vectors (Crossover);
26   end
27   Select target vectors at  $g + 1$  (Selection);
28    $\vec{m}_{g+1} = \frac{\sum_{i=1}^{Np} w_i \cdot \vec{x}_{\text{best},i,g+1}}{\sum_{i=1}^{Np} w_i}$ ;
29    $\vec{p}_{g+1} = (1 - c_c) \cdot \vec{p}_g + \sqrt{c_c \cdot (2 - c_c) \cdot \mu_{\text{eff}}} \cdot \frac{\vec{m}_{g+1} - \vec{m}_g}{\sigma_g}$ ;
30    $\vec{C}_{g+1} = (1 - c_1) \cdot \vec{C}_g + c_1 \cdot \vec{p}_{g+1} \cdot \vec{p}_{g+1}^T$ ;
31    $g = g + 1$ ;
32 end

```



**Figure 2.** Flowchart of DE with the proposed eigenvector-based crossover operator.

### 3.2.1. Rank-one update of the covariance matrix

The proposed method starts by initializing the covariance matrix  $\vec{C}_g$  as an identity matrix  $\vec{I} \in \mathbb{R}^{d \times d}$  and the search distribution's mean vector  $\vec{m}_g$  as the arithmetic mean of the population. After the selection operation, the mean vector is updated by taking the weighted arithmetic mean of the population as follows:

$$\vec{m}_{g+1} = \frac{\sum_{i=1}^{Np} w_i \cdot \vec{x}_{best_i, g+1}}{\sum_{i=1}^{Np} w_i} \quad (3.1)$$

$$w_i = \frac{w'_i}{\sum_{i=1}^{Np} w'_i} \quad (3.2)$$

$$w'_i = \ln(Np + 0.5) - \ln(i) \quad (3.3)$$

where  $\vec{x}_{best_i, g+1}$  and  $w_i$  are the  $i$ th best individual in the population and the  $i$ th weight coefficient, respectively. The weight coefficient is set according to the information provided in [20, 21, 23–25]. Assigning different values to the weight coefficient introduces a search bias toward promising regions.

The rank-one update of the covariance matrix is as follows:

$$\vec{C}_{g+1} = (1 - c_1) \cdot \vec{C}_g + c_1 \cdot \vec{p}_{g+1} \cdot \vec{p}_{g+1}^\top \quad (3.4)$$

$$\vec{p}_{g+1} = (1 - c_c) \cdot \vec{p}_g + \sqrt{c_c \cdot (2 - c_c) \cdot \mu_{\text{eff}}} \cdot \frac{\vec{m}_{g+1} - \vec{m}_g}{\sigma_g} \quad (3.5)$$

where  $\mu_{\text{eff}} = \frac{(\sum_{i=1}^{Np} w_i)^2}{\sum_{i=1}^{Np} w_i^2}$  is the variance effective selection mass,  $c_1 = \frac{2}{(D+1.3)^2 + \mu_{\text{eff}}}$  and  $c_c = \frac{4 + \mu_{\text{eff}}/D}{D+4+2 \cdot \mu_{\text{eff}}/D}$  are the learning rates, and  $\sigma_g = 1$  is the overall standard deviation. CMA-ES [20, 21, 23–25] involves step-size control to adjust the overall distribution scale. However, the proposed method does not require approximating the overall step size, as it generates offspring using the mutation and crossover operations. Therefore, the covariance matrix at each generation holds equal importance in the proposed method.

### 3.2.2. Eigenvector-based crossover operator

The proposed method establishes an eigen coordinate system by identifying an eigenvector basis. This process involves performing eigendecomposition on the covariance matrix. Since the covariance matrix is both symmetric and positive semi-definite, eigendecomposition allows the covariance matrix to be factored into a canonical form, represented as follows:

$$\vec{C}_g = \vec{B}\vec{D}^2\vec{B}^\top \quad (3.6)$$

where  $\vec{B}$  denotes an orthogonal matrix with its columns being eigenvectors, while  $\vec{D}^2$  represents a diagonal matrix with eigenvalues along its diagonal. These eigenvectors define the principal axes of the covariance matrix, indicating the directions of maximum variance in the evolutionary path. Simultaneously, the eigenvalues quantify the variance accounted for by each respective eigenvector along this path.

The proposed method transforms the target and mutant vectors into the eigen coordinate system using the orthogonal matrix in the following manner:

$$\vec{x}'_{i,g} = \vec{B}^\top \vec{x}_{i,g} \quad (3.7)$$

$$\vec{v}'_{i,g} = \vec{B}^\top \vec{v}_{i,g} \quad (3.8)$$

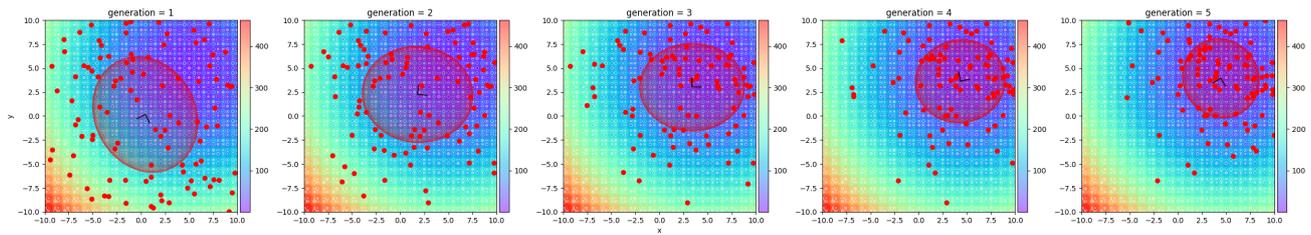
where  $\vec{x}'_{i,g}$  and  $\vec{v}'_{i,g}$  represent the transformed target and mutant vectors into the eigen coordinate system, respectively. After this transformation, a trial vector is created through the crossover operation. To convert the trial vector back to the standard coordinate system, the proposed method utilizes the orthogonal matrix once again in the following manner:

$$\vec{u}_{i,g} = \vec{B}\vec{u}'_{i,g} \quad (3.9)$$

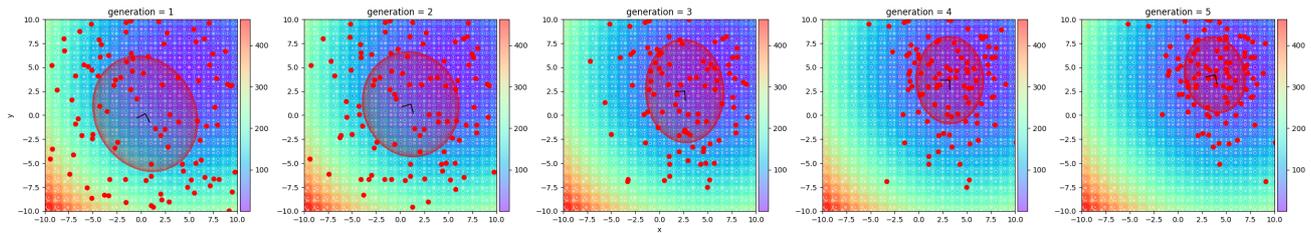
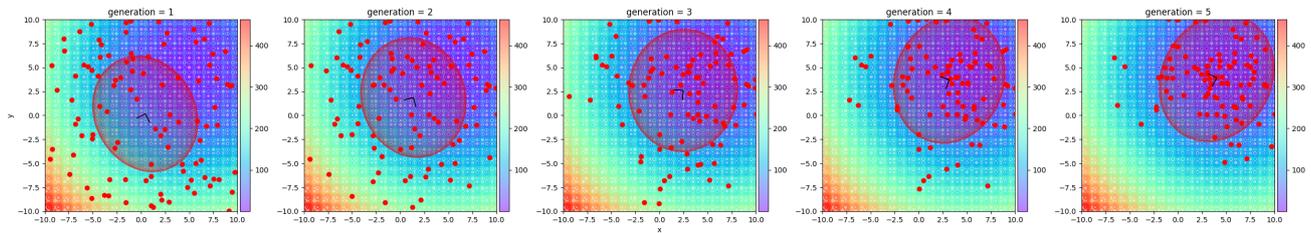
where  $\vec{u}'_{i,g}$  denotes the transformed trial vector into the eigen coordinate system.

### 3.3. Remarks

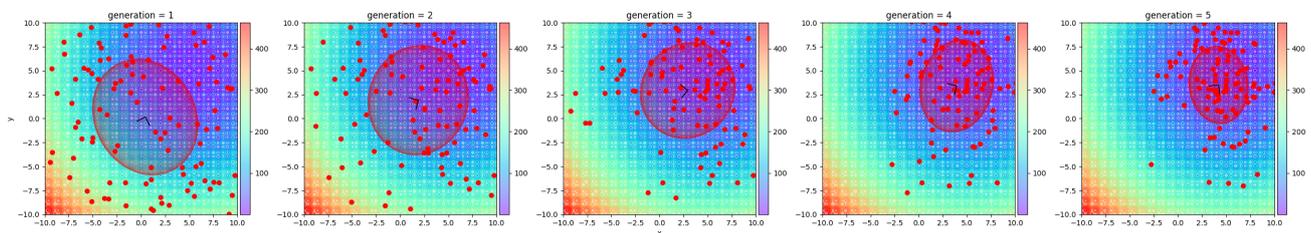
In this section, we present a method for updating the covariance matrix in a simple yet effective manner when establishing a new eigen coordinate system. Whereas previous methods require  $O(Np \cdot D^2)$  operations to update the covariance matrix, our proposed method achieves this with  $O(D^2)$  operations. This substantially speeds up crossover operations, even with a large population size in the DE algorithm. Figure 3 compares the performance of the proposed method and previous approaches on a shifted Rastrigin function. In Figure 3(a), the sample covariance matrix is derived directly from the current population to guide eigenvector-based crossover; here, red dots represent candidate solutions, while the color map indicates the objective function's contour. Figures 3(b)–3(d) illustrate three different covariance update strategies—rank- $Np$ , rank-one, and a combination of both, respectively—each offering a distinct way of incrementally estimating and refining the covariance matrix based on the population's performance. Although they differ in methodology, all strategies successfully guide the population toward promising regions of the search space, ultimately converging to the same optimal region. Thus, while their final results show no significant differences in performance, the key conclusion is that the proposed rank-one approach effectively captures the principal directions of the landscape with notably less computational cost.



(a) DE with the eigenvector-based crossover operator via a sample covariance matrix.

(b) DE with the eigenvector-based crossover operator via rank- $Np$  update.

(c) DE with the eigenvector-based crossover operator via rank-one update.

(d) DE with the eigenvector-based crossover operator via rank- $Np$  and rank-one updates.

**Figure 3.** Comparison of the proposed method with existing eigenvector-based DE variants on a shifted Rastrigin function. In (a), the covariance matrix is estimated from the current population (DE/eig, CoBiDE), while (b), (c), and (d) employ rank- $Np$ , rank-one, and combined rank updates (CPI-DE, ACoS, STCS, and CMA-ES), respectively. Despite the differing update strategies, all converge to the same region, confirming that the proposed rank-one approach effectively captures promising areas in the landscape with a lower computational cost.

## 4. Experimental settings

### 4.1. Benchmark functions

A test algorithm's performance is assessed using two widely recognized test suites: CEC2013 and CEC2017. The CEC2013 test suite [27] comprises 28 benchmark functions categorized into three types: unimodal functions (f1–f5), essential multimodal functions (f6–f20), and composition functions (f21–f28). Similarly, the CEC2017 test suite [2] includes 29 benchmark functions categorized as unimodal functions (g1 and g2), simple multimodal functions (g4–g10), hybrid functions (g11–g20), and composition functions (g21–g30). For more detailed information on the CEC2013 and CEC2017 test suites, please refer to [2, 27].

The test algorithm is executed independently 51 times for each benchmark function, with recorded outcomes including the average function error value (labeled as the mean) and the standard deviation (std).

To validate the statistical reliability of our findings, we conducted the Wilcoxon rank-sum [31] and Friedman tests [13, 14], followed by Hochberg's post hoc analysis [26] for all comparative assessments. Substantial improvements identified by these tests are highlighted in bold text.

### 4.2. Test algorithms

We compared the proposed method with five selected methods: DE/eig [16], CoBiDE [43], CPI-DE [44], the proposed framework with rank- $Np$  update (DE/ $r_{Np}$ ), and the proposed framework with rank- $Np$  and rank-one updates (DE/ $r_{Np+1}$ ). The rationale for selecting these methods is as follows.

- **DE/eig** and **CoBiDE**: These methods allow us to compare the difference in performance between using the rank-one update and utilizing the distribution information of the current population generation.
- **CPI-DE**: This method compares the difference in performance between the rank-one and rank- $Np$  updates. Notably, it employs the CPI framework, which removes problem-dependent parameters.
- **DE/ $r_{Np}$**  and **DE/ $r_{Np+1}$** : These methods compare the rank-one update, the rank- $Np$  update, and the combined rank- $Np$  and rank-one updates for covariance matrix estimation. CPI-DE and DE/ $r_{Np}$  use the rank- $Np$  update, but a direct comparison of the rank-one update versus the rank- $Np$  update is challenging due to CPI-DE's use of the CPI framework.

We excluded ACoS [28] and STCS [15] in our experiments because they employ adaptive parameter control methods similar to DE/ $r_{Np}$ . Our primary objective was to evaluate various covariance matrix estimation techniques rather than comparing adaptive parameter control methods for eigenvector ratios.

The control parameters of the compared methods are set to the default values as specified in their respective references. The proposed methods, DE/ $r_{Np}$  and DE/ $r_{Np+1}$ , utilize the eigenvector ratio as a control parameter set to 0.05. Table 1 presents the parameter configurations of the proposed and compared methods. It is important to note that CPI-DE does not employ control parameters such as the eigenvector ratio.

**Table 1.** Parameter configurations of proposed and compared methods.

Algorithm	Parameter settings
DE/ $r_1$	$P = 0.05$
DE/eig	$P = 0.5$
CoBiDE	$pb = 0.5, ps = 0.4$
DE/ $r_{Np}$	$P = 0.05$
DE/ $r_{Np+1}$	$P = 0.05$

We applied the proposed and compared methods to two state-of-the-art DE algorithms, NL-SHADE-LBC [38] and NL-SHADE-RSP [37], which were the winners of the CEC and GECCO competitions in 2022 and 2021, respectively. The control parameters of NL-SHADE-LBC and NL-SHADE-RSP were set to the default values specified in their references. Table 2 presents the parameter configurations of NL-SHADE-LBC and NL-SHADE-RSP.

**Table 2.** Parameter configurations of NL-SHADE-LBC and NL-SHADE-RSP.

Algorithm	Parameter settings
NL-SHADE-LBC	$Np_{max} = 23D, M_{F,r} = 0.5, M_{Cr,r} = 0.9$ $H = 20D, k = 1, NA = 1.0 \cdot Np, n_A = 0.5$
NL-SHADE-RSP	$Np_{max} = 30D, M_{F,r} = 0.2, M_{Cr,r} = 0.2$ $H = 20D, k = 1, NA = 2.1 \cdot Np, n_A = 0.5$

All comparative assessments were conducted on a Ubuntu 20.04.5 LTS PC equipped with an AMD Ryzen Threadripper 2990WX CPU and 64 GB of RAM. All test algorithms were implemented in C++.

## 5. Experimental results

This section presents comparative assessments demonstrating the effectiveness of DE/ $r_1$  in solving optimization problems. Section 5.1 compares NL-SHADE-LBC using DE/ $r_1$  and NL-SHADE-LBC, employing the compared methods on the CEC2013 and CEC2017 test suites across three different dimensional settings. Subsequently, Section 5.2 offers an extensive comparison between NL-SHADE-RSP using DE/ $r_1$  and NL-SHADE-RSP with the compared methods on the CEC2013 and CEC2017 test suites across three different dimensional settings.

### 5.1. Comparative assessments with NL-SHADE-LBC

Tables 3 and 6 present the aggregated comparison results derived from the Friedman test on the CEC'2013 and CEC'2017 test suites, respectively. Additionally, Tables 4 and 5 present the aggregated comparison results derived from the Wilcoxon rank-sum test on the CEC'2013 and CEC'2017 test suites, respectively. The symbols “+/-” indicate that NL-SHADE-LBC with the corresponding method performed substantially better (+), similar (=), or substantially worse (-) compared with NL-SHADE-LBC with DE/ $r_1$  based on the Wilcoxon rank-sum test with an  $\alpha = 0.05$  significance level.

- **Comparison of the results on CEC2013:** Table 3 summarizes the results derived from Tables S2, S4, and S6 in the supplementary material. The proposed method outperformed all the compared methods at all dimensionalities. The Friedman test rankings for the proposed method were 3.45 for 10D, 3.57 for 30D, and 3.36 for 50D, resulting in an average ranking of 3.32, the highest observed. Hochberg's post hoc analysis revealed substantial differences in performance favoring the proposed method at the 30D and 50D dimensions, with adjusted  $P$ -values of  $2.27e-03$  and  $7.74e-05$ , respectively, below the 0.05 significance threshold. The average rankings for the proposed framework with the rank- $Np$  update ( $DE/r_{Np}$ ) and the rank- $Np$  and rank-one updates ( $DE/r_{Np+1}$ ) were 3.72 and 3.75, placing them in second and third positions, respectively.

**Table 3.** Aggregated comparison results between NL-SHADE-LBC with  $DE/r_1$  and NL-SHADE-LBC with the compared methods derived from the Friedman test with Hochberg's post hoc analysis on the CEC2013 test suite.

NL-SHADE-LBC				
Rank	10D	30D	50D	Mean
$DE/r_1$	<b>3.45</b>	<b>3.57</b>	<b>3.36</b>	<b>3.46</b>
DE-eig	4.05	<b>3.57</b>	4.07	3.90
CoBiDE	4.00	3.79	3.48	3.76
CPI-DE	5.00	5.57	5.54	5.37
$DE/r_{Np}$	3.71	4.05	3.39	3.72
$DE/r_{Np+1}$	4.14	3.66	3.45	3.75
Original	3.64	3.79	4.71	4.05
$P$ -value	7.10.E-02	2.27.E-03	7.74.E-05	

Table 4, derived from Tables S1, S3, and S5 in the supplementary material, supports the Friedman test's conclusions. For example, comparing DE-eig vs.  $DE/r_1$ , the total result is 3/65/16, meaning that NL-SHADE-LBC with the proposed method outperformed NL-SHADE-LBC with DE-eig in 16 cases and was outperformed in 3 cases. The proposed method substantially enhances NL-SHADE-LBC's performance from both the statistical and accuracy perspectives, with NL-SHADE-LBC with  $DE/r_1$  outperforming its original version in 28 cases and being outperformed in 12 cases. The most notable differences in performance are between  $DE/r_1$  and CPI-DE, where NL-SHADE-LBC with  $DE/r_1$  outperformed NL-SHADE-LBC with CPI-DE on 44 functions and was outperformed on 5 functions. No substantial differences in performance were observed between the proposed method and the proposed framework with the rank- $Np$  update ( $DE/r_{Np}$ ) and the rank- $Np$  and rank-one updates ( $DE/r_{Np+1}$ ).

**Table 4.** Aggregated comparison results between NL-SHADE-LBC with  $DE/r_1$  and NL-SHADE-LBC with the compared methods derived from the Wilcoxon rank-sum test with the  $\alpha = 0.05$  significance level on the CEC2013 test suite.

NL-SHADE-LBC				
+/=/-	10D	30D	50D	Total
DE-eig vs. $DE/r_1$	0/24/4	2/20/6	1/21/6	3/65/16
CoBiDE vs. $DE/r_1$	0/26/2	1/22/5	2/23/3	3/71/10
CPI-DE vs. $DE/r_1$	1/12/15	3/9/16	1/14/13	5/35/44
$DE/r_{Np}$ vs. $DE/r_1$	0/26/2	1/26/1	0/28/0	1/80/3
$DE/r_{Np+1}$ vs. $DE/r_1$	0/26/2	0/28/0	1/26/1	1/80/3
Original vs. $DE/r_1$	8/11/9	6/14/8	2/15/11	16/40/28

- Comparison of the results on CEC2017:** Table 6, derived from Tables S8, S10, and S12 in the supplementary material, indicates that the proposed method was more effective than most compared methods across all tested dimensions. The Friedman test results showed rankings of 3.36 for 10D, 3.53 for 30D, and 3.07 for 50D, with an average ranking of 3.32, the best among the compared methods. Hochberg's post hoc analysis identified substantial differences in performance across all tested dimensions, with adjusted  $P$ -values of  $1.79e-04$  for 10D,  $1.78e-04$  for 30D, and  $3.34e-06$  for 50D, all below the 0.05 significance level. The average rankings for the proposed framework combined with the rank- $Np$  update ( $DE/r_{Np}$ ) and the rank- $Np$  and rank-one updates ( $DE/r_{Np+1}$ ) were 3.45 and 3.56, respectively, securing the second and third positions. Table 5, derived from Tables S7, S9, and S11 in the supplementary material, supports the Friedman test's conclusion. The proposed method substantially enhances NL-SHADE-LBC's performance from both the statistical and accuracy perspectives, with the proposed method outperforming its original version in 25 cases and 12 cases. The most notable differences in performance are between  $DE/r_1$  and CPI-DE, where NL-SHADE-LBC with the proposed method outperformed NL-SHADE-LBC with CPI-DE on 46 functions and was outperformed on 3 functions. No substantial differences in performance were observed between the proposed method and the proposed framework with the rank- $Np$  update ( $DE/r_{Np}$ ) and the rank- $Np$  and rank-one updates ( $DE/r_{Np+1}$ ).

**Table 5.** Aggregated comparison results between NL-SHADE-LBC with  $DE/r_1$  and NL-SHADE-LBC with the compared methods derived from the Wilcoxon rank-sum test with the  $\alpha = 0.05$  significance level on the CEC2017 test suite.

NL-SHADE-LBC				
+/=/-	10D	30D	50D	Total
DE-eig vs. $DE/r_1$	1/18/10	2/15/12	3/12/14	6/45/36
CoBiDE vs. $DE/r_1$	0/25/4	2/19/8	4/11/14	6/55/26
CPI-DE vs. $DE/r_1$	2/16/11	1/11/17	0/11/18	3/38/46
$DE/r_{Np}$ vs. $DE/r_1$	1/25/3	1/26/2	0/29/0	2/80/5
$DE/r_{Np+1}$ vs. $DE/r_1$	1/28/0	0/29/0	0/29/0	1/86/0
Original vs. $DE/r_1$	4/20/5	6/10/13	2/20/7	12/50/25

**Table 6.** Aggregated comparison results between NL-SHADE-LBC with  $DE/r_1$  and NL-SHADE-LBC with the compared methods derived from the Friedman test with Hochberg's post hoc analysis on the CEC2017 test suite.

NL-SHADE-LBC				
Rank	10D	30D	50D	Mean
$DE/r_1$	3.36	3.53	<b>3.07</b>	<b>3.32</b>
DE-eig	5.38	4.31	4.72	4.80
CoBiDE	4.31	3.93	4.21	4.15
CPI-DE	4.33	5.34	5.69	5.12
$DE/r_{Np}$	3.83	<b>3.31</b>	3.21	3.45
$DE/r_{Np+1}$	3.53	3.53	3.62	3.56
Original	<b>3.26</b>	4.03	3.48	3.59
<i>P</i> -value	1.79.E-04	1.78.E-04	3.34.E-06	

The experimental results demonstrate the proposed method's superiority over various other methods. According to the Friedman test, it consistently achieved the best average ranking, with substantial differences in performance identified via the Wilcoxon rank-sum test, particularly at higher dimensionalities. The proposed method substantially improved NL-SHADE-LBC's performance in many cases, outperforming methods like DE-eig and CPI-DE across various functions. Despite these achievements, no substantial differences were observed between the proposed method and the proposed framework with either the rank- $Np$  update ( $DE/r_{Np}$ ) or the rank- $Np$  and rank-one updates ( $DE/r_{Np+1}$ ). Notably, the proposed method requires substantially fewer computational resources than these approaches.

## 5.2. Comparative assessments with NL-SHADE-RSP

Tables 7 and 9 present the aggregated comparison results derived from the Friedman test on the CEC2013 and CEC2017 test suites, respectively. Additionally, Tables 8 and 10 present the aggregated

comparison results derived from the Wilcoxon rank-sum test on the CEC2013 and CEC2017 test suites, respectively.

**Table 7.** Aggregated comparison results between NL-SHADE-RSP with DE/ $r_1$  and NL-SHADE-RSP with the compared methods derived from the Friedman test with Hochberg's post hoc analysis on the CEC2013 test suite.

NL-SHADE-RSP				
Rank	10D	30D	50D	Mean
DE/ $r_1$	<b>3.34</b>	3.89	3.63	3.62
DE-eig	3.64	3.54	<b>3.09</b>	<b>3.42</b>
CoBiDE	3.86	<b>3.18</b>	3.34	3.46
CPI-DE	5.34	4.79	5.71	5.28
DE/ $r_{Np}$	4.23	4.07	3.39	3.90
DE/ $r_{Np+1}$	4.00	3.89	3.79	3.89
Original	3.59	4.64	5.05	4.43
<i>P</i> -value	6.87.E-03	2.17.E-02	7.52.E-07	

**Table 8.** Aggregated comparison results between NL-SHADE-RSP with DE/ $r_1$  and NL-SHADE-RSP with the compared methods derived from the Wilcoxon rank-sum test with the  $\alpha = 0.05$  significance level on the CEC2013 test suite.

NL-SHADE-RSP				
+/=/-	10D	30D	50D	Total
DE-eig vs. DE/ $r_1$	2/22/4	6/15/7	7/15/6	15/52/17
CoBiDE vs. DE/ $r_1$	4/20/4	4/18/6	4/19/5	12/57/15
CPI-DE vs. DE/ $r_1$	2/12/14	4/14/10	0/13/15	6/39/39
DE/ $r_{Np}$ vs. DE/ $r_1$	0/26/2	1/24/3	1/26/1	2/76/6
DE/ $r_{Np+1}$ vs. DE/ $r_1$	0/27/1	0/28/0	0/28/0	0/83/1
Original vs. DE/ $r_1$	6/14/8	3/13/12	4/12/12	13/39/32

**Table 9.** Aggregated comparison results between NL-SHADE-RSP with DE/ $r_1$  and NL-SHADE-RSP with the compared methods derived from the Friedman test with Hochberg's post hoc analysis on the CEC2017 test suite.

NL-SHADE-RSP				
Rank	10D	30D	50D	Mean
DE/ $r_1$	<b>3.21</b>	2.95	<b>2.34</b>	<b>2.83</b>
DE-eig	4.57	5.17	5.19	4.98
CoBiDE	4.66	4.50	4.45	4.53
CPI-DE	4.53	6.19	5.84	5.52
DE/ $r_{Np}$	3.91	2.98	3.10	3.33
DE/ $r_{Np+1}$	3.69	<b>2.86</b>	3.00	3.18
Original	3.43	3.34	4.07	3.61
<i>P</i> -value	1.62.E-02	2.51.E-12	5.70.E-11	

**Table 10.** Aggregated comparison results between NL-SHADE-RSP with DE/ $r_1$  and NL-SHADE-RSP with the compared methods derived from the Wilcoxon rank-sum test with the  $\alpha = 0.05$  significance level on the CEC2017 test suite.

NL-SHADE-RSP				
+/=/-	10D	30D	50D	Total
DE-eig vs. DE/ $r_1$	1/15/13	2/5/22	2/7/20	5/27/55
CoBiDE vs. DE/ $r_1$	1/18/10	2/7/20	3/9/17	6/34/47
CPI-DE vs. DE/ $r_1$	5/13/11	1/6/22	2/6/21	8/25/54
DE/ $r_{Np}$ vs. DE/ $r_1$	0/27/2	2/27/0	0/28/1	2/82/3
DE/ $r_{Np+1}$ vs. DE/ $r_1$	0/27/2	0/29/0	0/28/1	0/84/3
Original vs. DE/ $r_1$	7/15/7	5/13/11	1/20/8	13/48/26

- Comparison of the results on CEC2013:** The results in Table 7 were obtained from Tables S14, S16, and S18 in the supplementary material. In light of the results presented in Table 7, it can be observed that the proposed method outperformed all the compared methods except for DE-eig and CoBiDE at all dimensionalities. As obtained by the Friedman test, the final rankings of the proposed method were 3.34 at 10D, 3.89 at 30D, and 3.63 at 50D, with a mean ranking of 3.62, the third best. The mean rankings of DE-eig and CoBiDE were 3.42 and 3.46, respectively, and ranked first and second best, respectively. Hochberg's post hoc analysis revealed significant differences in performance between the proposed and compared methods at all dimensionalities, with adjusted *P*-values of 6.87e-03 at 10D, 2.17e-02 at 30D, and 7.52e-07 at 50D; all values are less than the significance level of 0.05.

The results presented in Table 8, derived from Tables S13, S15, and S17 in the supplementary material, support the conclusions of the Friedman test. It is evident that the proposed method significantly improves NL-SHADE-RSP's effectiveness, as seen from both the statistical analyses

and the accuracy measures. Specifically, in 32 cases, NL-SHADE-RSP enhanced by the proposed method surpassed its original form, whereas in 13 cases, it did not perform as well. Compared with DE-eig, the proposed method yielded better results in 17 cases while underperforming in 15 cases. Similarly, compared with CoBiDE, the proposed method yielded better results in 15 cases while underperforming in 12 cases. Therefore, no notable performance disparities were observed between the proposed method and either DE-eig or CoBiDE despite these having better final rankings than the proposed method.

- Comparison of the results on CEC2017:** The results in Table 9 were obtained from Tables S20, S22, and S24 in the supplementary material. According to the results presented in Table 9, the proposed method outperformed all the compared methods at most of the dimensionalities. As obtained by the Friedman test, the final rankings of the proposed method were 3.21 at 10D, 2.95 at 30D, and 2.34 at 50D, with a mean ranking of 2.83, the best. Hochberg's post hoc analysis revealed significant differences in performance between the proposed and compared methods at all dimensionalities, with adjusted  $P$ -values of  $1.62e-02$  at 10D,  $2.51e-12$  at 30D, and  $5.70e-11$  at 50D; all values are less than the significance level of 0.05. Notably, the mean rankings of the proposed framework with the rank- $Np$  and rank-one updates ( $DE/r_{Np+1}$ ) and the proposed framework with the rank- $Np$  update ( $DE/r_{Np}$ ) were 3.18 and 3.33, respectively, and ranked second and third best, respectively.

The results in Table 10 were obtained from Tables S19, S21, and S23 in the supplementary material, supporting the Friedman test's conclusion. The proposed method can observably enhance the performance of NL-SHADE-RSP from both the statistical and accuracy perspectives, where NL-SHADE-RSP with the proposed method outperformed its corresponding original version in 26 cases, whereas it was outperformed in 13 cases. The differences in performance between  $DE/r_1$  and DE-eig are the most obvious, where NL-SHADE-RSP with the proposed method outperformed NL-SHADE-RSP with DE-eig on 55 functions, whereas it was outperformed on 5 functions. In addition, we observed no significant differences in performance between the proposed method and the proposed framework with the rank- $Np$  update ( $DE/r_{Np}$ ) and the rank- $Np$  and rank-one updates ( $DE/r_{Np+1}$ ).

The experimental results highlight the proposed method's effectiveness, showing it outperforms most of the compared methods across several dimensions. Despite achieving the third-highest average ranking on the CEC2013 suite according to the Friedman test, individual comparisons using the Wilcoxon rank-sum test revealed its performance was on par with DE-eig and CoBiDE, the top two performers. On the CEC2017 suite, the proposed method distinguished itself as the leading performer. Additionally, it significantly improved the performance of NL-SHADE-RSP, exceeding the achievements of its original version.

### 5.3. Algorithm complexity

The proposed method is designed to be computationally more efficient than the compared methods. We evaluated its algorithmic complexity using the CEC2017 test suite to substantiate this claim. The computing time  $T_0$  is measured by executing the test program outlined below.

---

```

1 x=0.55;
2 for i = 1; i < 1000000; i = i + 1 do
3   | x=x+x; x=x/2; x=x*x; x=sqrt(x);
4   | x=log(x); x=exp(x); x=x/(x+2);
5 end

```

---

The computing time  $T_1$  is determined by running the function g18 for 200,000 evaluations. Similarly, computing time  $T_2$  is obtained by executing the same function, g18, under an algorithm that also performs 200,000 evaluations. The average computing time,  $\hat{T}_2$ , is calculated as the mean of five instances of  $T_2$  measurements.

Table 11 presents the algorithm complexity results comparing NL-SHADE-LBC with DE/ $r_1$  against NL-SHADE-LBC with other methods. As shown in this table, except for the original algorithm, the proposed method requires the least computational time relative to the compared methods. The proposed method exhibits complexities of 4.133 for 10D, 20.830 for 30D, and 42.282 for 50D. Compared with DE-eig, the proposed method requires three times less computation time for 10D, six times less for 30D, and eight times less for 50D. Similarly, CoBiDE requires approximately three times more computational time for all dimensions (10D, 30D, and 50 D) than the proposed method. CPI-DE requires less computation time than the other methods because it consistently generates  $Np$  offspring in each generation's standard coordinate system. However, CPI-DE still requires more computational time than the proposed method. Lastly, DE/ $r_{Np}$  and DE/ $r_{Np+1}$  consume more computational time than the proposed method due to their additional use of the rank- $Np$  update. Overall, the proposed method performs better and requires less computational time than the other methods.

Table 12 presents the algorithm complexity results comparing NL-SHADE-RSP with DE/ $r_1$  against NL-SHADE-RSP with other methods. This table illustrates that, except for the original algorithm, the proposed method demands the least computational time relative to the compared methods. It has a complexity of 4.069 for 10D, 23.095 for 30D, and 42.903 for 50D. Compared with DE-eig, the computational time needed by the proposed method is three times less for 10D, five times less for 30D, and eight times less for 50D. Compared with CoBiDE, the computational time needed by the proposed method is three times less for 10D, three times less for 30D, and four times less for 50D. CPI-DE, DE/ $r_{Np}$ , and DE/ $r_{Np+1}$  also require greater computational time than the proposed method. Overall, the proposed method requires less computational time while maintaining superior performance compared with the other methods.

**Table 11.** Algorithm complexity results between NL-SHADE-LBC with DE/ $r_1$  and NL-SHADE-LBC with the compared methods on the CEC2017 test suite.

D	DE/ $r_1$			DE-eig			CoBiDE			CPI-DE		
	10	30	50	10	30	50	10	30	50	10	30	50
$\hat{T}_2$	0.156	0.784	1.830	0.320	2.577	6.887	0.314	1.618	3.958	0.241	1.035	2.294
$\frac{\hat{T}_2 - T_1}{T_0}$	4.133	20.830	42.282	13.886	127.298	342.669	13.507	70.370	168.674	9.181	35.718	69.868

D	DE/ $r_{Np}$			DE/ $r_{Np+1}$			Original					
	10	30	50	10	30	50	10	30	50			
$\hat{T}_2$				0.311	2.095	5.490	0.364	2.120	5.550	0.120	0.527	1.399
$\frac{\hat{T}_2 - T_1}{T_0}$				13.363	98.671	259.671	16.476	100.199	263.221	2.002	5.551	16.719

$T_0 = 0.016837s$ ,  $T_1(10D) = 0.086452s$ ,  $T_1(30D) = 0.433232s$ , and  $T_1(50D) = 1.117972s$ .

**Table 12.** Algorithm complexity results between NL-SHADE-RSP with DE/ $r_1$  and NL-SHADE-RSP with the compared methods on the CEC2017 test suite.

D	DE/ $r_1$			DE-eig			CoBiDE			CPI-DE		
	10	30	50	10	30	50	10	30	50	10	30	50
$\hat{T}_2$	0.150	0.798	1.802	0.314	2.276	6.861	0.366	1.747	4.142	0.272	1.107	2.405
$\frac{\hat{T}_2 - T_1}{T_0}$	4.069	23.095	42.903	14.232	114.649	356.367	17.490	81.858	187.857	11.680	42.255	80.294

D	DE/ $r_{Np}$			DE/ $r_{Np+1}$			Original					
	10	30	50	10	30	50	10	30	50			
$\hat{T}_2$				0.315	2.140	5.573	0.353	2.722	6.888	0.116	0.505	1.222
$\frac{\hat{T}_2 - T_1}{T_0}$				14.351	106.236	276.562	16.684	142.311	358.025	1.966	4.948	7.000

$T_0 = 0.01614s$ ,  $T_1(10D) = 0.083841s$ ,  $T_1(30D) = 0.425319s$ , and  $T_1(50D) = 1.109502s$ .

## 6. Conclusions

We have introduced DE/ $r_1$ , a simple yet effective eigenvector-based crossover operator, to enhance the capability of the DE algorithm in tackling optimization problems with variables exhibiting strong correlations. Like other eigenvector-based crossover operators, this method calculates the covariance matrix, applies eigendecomposition to establish a new eigen coordinate system, and performs crossover operations within this system. The main innovation of our approach lies in exclusively using the rank-one update for covariance matrix estimation, substantially reducing the computational costs. Specifically, while the previous methods require  $O(Np \cdot D^2)$  computational operations for covariance matrix estimation, our method only requires  $O(D^2)$  computational operations. Despite concerns that our method's covariance matrix may not capture the function's landscape as precisely as previous operators, our experimental analysis reveals no noticeable decline in performance. Our method discovered solutions more quickly and accurately than previous methods.

We did not include ACoS [28] and STCS [15] in our experiments, as they are similar to DE/ $r_{Np}$  but use adaptive parameter control methods. Our focus was on evaluating different techniques for estimating the covariance matrix within an eigenvector-based crossover operator rather than assessing adaptive parameter control methods for the eigenvector ratio. However, future research will explore efficient adaptive parameter control methods for our approach.

Additionally, we aim to examine the applicability and effectiveness of our method in tackling various other types of optimization problems, including dynamic and multi-objective optimization,

---

in subsequent studies.

### Use of Generative-AI tools declaration

The author declares he has not used Artificial Intelligence (AI) tools in the creation of this article.

### Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2023-00214326).

### Conflict of interest

The author declares no conflict of interest.

### References

1. R. D. Al-Dabbagh, F. Neri, N. Idris, M. S. Baba, Algorithmic design issues in adaptive differential evolution schemes: Review and taxonomy, *Swarm Evol. Comput.*, **43** (2018), 284–311. <https://doi.org/10.1016/j.swevo.2018.03.008>
2. N. Awad, M. Ali, J. Liang, B. Qu, P. Suganthan, *Problem definitions and evaluation criteria for the cec 2017 special session and competition on single objective bound constrained real-parameter numerical optimization*, in Technical report, Nanyang Technological University, Singapore, 2016, 1–34.
3. F. Caraffini, F. Neri, A study on rotation invariance in differential evolution, *Swarm Evol. Comput.*, **50** (2019), 100436. <https://doi.org/10.1016/j.swevo.2018.08.013>
4. T. J. Choi, A rotationally invariant stochastic opposition-based learning using a beta distribution in differential evolution, *Expert Syst. Appl.*, **231** (2023), 120658. <https://doi.org/10.1016/j.eswa.2023.120658>
5. T. J. Choi, C. W. Ahn, Artificial life based on boids model and evolutionary chaotic neural networks for creating artworks, *Swarm Evol. Comput.*, **47** (2019), 80–88. <https://doi.org/10.1016/j.swevo.2017.09.003>
6. T. J. Choi, C. W. Ahn, An improved lshade-rsp algorithm with the cauchy perturbation: ilshade-rsp, *Knowl.-Based Syst.*, **215** (2021), 106628. <https://doi.org/10.1016/j.knosys.2020.106628>
7. T. J. Choi, J. H. Lee, H. Y. Youn, C. W. Ahn, Adaptive differential evolution with elite opposition-based learning and its application to training artificial neural networks, *Fund. Inform.*, **164** (2019), 227–242. <https://doi.org/10.3233/FI-2019-1764>
8. T. J. Choi, J. Togelius, Y. G. Cheong, Advanced cauchy mutation for differential evolution in numerical optimization, *IEEE Access*, **8** (2020), 8720–8734. <https://doi.org/10.1109/ACCESS.2020.2964222>

9. T. J. Choi, J. Togelius, Y. G. Cheong, A fast and efficient stochastic opposition-based learning for differential evolution in numerical optimization, *Swarm Evol. Comput.*, **60** (2021), 100768. <https://doi.org/10.1016/j.swevo.2020.100768>
10. M. Dai, X. Feng, H. Yu, W. Guo, An opposition-based differential evolution clustering algorithm for emotional preference and migratory behavior optimization, *Knowl.-Based Syst.*, **259** (2023), 110073. <https://doi.org/10.1016/j.knosys.2022.110073>
11. S. Das, S. S. Mullick, P. N. Suganthan, Recent advances in differential evolution—An updated survey, *Swarm Evol. Comput.*, **27** (2016), 1–30. <https://doi.org/10.1016/j.swevo.2016.01.004>
12. S. Das, P. N. Suganthan, Differential evolution: A survey of the state-of-the-art, *IEEE T. Evol. Comput.*, **15** (2010), 4–31. <https://doi.org/10.1109/TEVC.2010.2059031>
13. M. Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, *J. Am. Stat. Assoc.*, **32** (1937), 675–701. <https://doi.org/10.1080/01621459.1937.10503522>
14. M. Friedman, A comparison of alternative tests of significance for the problem of m rankings, *Ann. Math. Stat.*, **11** (1940), 86–92. <https://doi.org/10.1214/aoms/1177731944>
15. W. Gao, Q. Dang, M. Gong, An adaptive framework to select the coordinate systems for evolutionary algorithms, *Appl. Soft Comput.*, **129** (2022), 109585. <https://doi.org/10.1016/j.asoc.2022.109585>
16. S. M. Guo, C. C. Yang, Enhancing differential evolution utilizing eigenvector-based crossover operator, *IEEE T. Evol. Comput.*, **19** (2014), 31–49. <https://doi.org/10.1109/TEVC.2013.2297160>
17. S. Gupta, W. Shu, Y. Zhang, R. Su, Differential evolution-driven traffic light scheduling for vehicle-pedestrian mixed-flow networks, *Knowl.-Based Syst.*, **274** (2023), 110636. <https://doi.org/10.1016/j.knosys.2023.110636>
18. S. Gupta, R. Su, Multiple individual guided differential evolution with time varying and feedback information-based control parameters, *Knowl.-Based Syst.*, **259** (2023), 110091. <https://doi.org/10.1016/j.knosys.2022.110091>
19. Y. Han, H. Peng, C. Mei, L. Cao, C. Deng, H. Wang, et al., Multi-strategy multi-objective differential evolutionary algorithm with reinforcement learning, *Knowl.-Based Syst.*, **277** (2023), 110801. <https://doi.org/10.1016/j.knosys.2023.110801>
20. N. Hansen, *The cma evolution strategy: A comparing review*, Towards a new evolutionary computation: Advances in the estimation of distribution algorithms, Springer, Berlin, Heidelberg, 2006, 75–102. [https://doi.org/10.1007/3-540-32494-1\\_4](https://doi.org/10.1007/3-540-32494-1_4)
21. N. Hansen, The cma evolution strategy: A tutorial, *arXiv preprint*, arXiv:1604.00772, 2016.
22. N. Hansen, S. Kern, *Evaluating the cma evolution strategy on multimodal test functions*, in International conference on parallel problem solving from nature, Springer, 2004, 282–291.
23. N. Hansen, S. D. Müller, P. Koumoutsakos, Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es), *Evol. Comput.*, **11** (2003), 1–18. <https://doi.org/10.1162/106365603321828970>
24. N. Hansen, A. Ostermeier, Completely derandomized self-adaptation in evolution strategies, *Evol. Comput.*, **9** (2001), 159–195. <https://doi.org/10.1162/106365601750190398>

25. N. Hansen-INRIA, *Variable metrics in evolutionary computation*, 2009.
26. Y. Hochberg, A sharper bonferroni procedure for multiple tests of significance, *Biometrika*, **75** (1988), 800–802. <https://doi.org/10.1093/biomet/75.4.800>
27. J. J. Liang, B. Qu, P. N. Suganthan, A. G. Hernández-Díaz, *Problem definitions and evaluation criteria for the cec 2013 special session on real-parameter optimization*, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, Technical Report, **201212** (2013), 281–295.
28. Z. Z. Liu, Y. Wang, S. Yang, K. Tang, An adaptive framework to tune the coordinate systems in nature-inspired optimization algorithms, *IEEE T. Cybernetics*, **49** (2018), 1403–1416. <https://doi.org/10.1109/TCYB.2018.2802912>
29. K. D. Lu, Z. G. Wu, Constrained-differential-evolution-based stealthy sparse cyber-attack and countermeasure in an ac smart grid, *IEEE T. Ind. Inform.*, **18** (2021), 5275–5285. <https://doi.org/10.1109/TII.2021.3129487>
30. K. D. Lu, Z. G. Wu, T. Huang, Differential evolution-based three stage dynamic cyber-attack of cyber-physical power systems, *IEEE-ASME T. Mech.*, **28** (2022), 1137–1148. <https://doi.org/10.1109/TMECH.2022.3214314>
31. H. B. Mann, D. R. Whitney, On a test of whether one of two random variables is stochastically larger than the other, *Ann. Math. Stat.*, 1947, 50–60. <https://doi.org/10.1214/aoms/1177730491>
32. R. R. Mostafa, A. M. Khedr, Z. A. Aghbari, I. Afyouni, I. Kamel, N. Ahmed, An adaptive hybrid mutated differential evolution feature selection method for low and high-dimensional medical datasets, *Knowl.-Based Syst.*, **283** (2024), 111218. <https://doi.org/10.1016/j.knosys.2023.111218>
33. F. Neri, V. Tirronen, Recent advances in differential evolution: A survey and experimental analysis, *Artif. Intell. Rev.*, **33** (2010), 61–106. <https://doi.org/10.1007/s10462-009-9137-2>
34. B. Mirza, M. Pant, H. Zaheer, L. Garcia-Hernandez, A. Abraham, Differential evolution: A review of more than two decades of research, *Eng. Appl. Artif. Intel.*, **90** (2020), 103479. <https://doi.org/10.1016/j.engappai.2020.103479>
35. K. V. Price, *Differential evolution vs. the functions of the 2/sup nd/iceo*, in Proceedings of 1997 IEEE International Conference on Evolutionary Computation (ICEC'97), IEEE, 1997, 153–157.
36. X. Song, Y. Zhang, W. Zhang, C. He, Y. Hu, J. Wang, et al., Evolutionary computation for feature selection in classification: A comprehensive survey of solutions, applications and challenges, *Swarm Evol. Comput.*, **90** (2024), 101661. <https://doi.org/10.1016/j.swevo.2024.101661>
37. V. Stanovov, S. Akhmedova, E. Semenkin, *Nl-SHADE-RSP algorithm with adaptive archive and selective pressure for CEC 2021 Numerical Optimization*, in 2021 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2021, 809–816. <https://doi.org/10.1109/CEC45853.2021.9504959>
38. V. Stanovov, S. Akhmedova, E. Semenkin, *Nl-SHADE-LBC algorithm with linear parameter adaptation bias change for CEC 2022 Numerical Optimization*, in 2022 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2022, 01–08. <https://doi.org/10.1109/CEC55065.2022.9870295>

39. R. Storn, *On the usage of differential evolution for function optimization*, in Proceedings of north american fuzzy information processing, IEEE, 1996, 519–523. <https://doi.org/10.1109/NAFIPS.1996.534789>
40. R. Storn, K. Price, *Minimizing the real functions of the icec'96 contest by differential evolution*, in Proceedings of IEEE international conference on evolutionary computation, IEEE, 1996, 842–844. <https://doi.org/10.1109/ICEC.1996.542711>
41. R. M. Storn, K. Price, Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces, *J. Global Optim.*, **11** (1997), 341–359. <https://doi.org/10.1023/A:1008202821328>
42. A. M. Sutton, M. Lunacek, L. D. Whitley, *Differential evolution and non-separability: Using selective pressure to focus search*, in Proceedings of the 9th annual conference on Genetic and evolutionary computation, 2007, 1428–1435.
43. 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. <https://doi.org/10.1016/j.asoc.2014.01.038>
44. Y. Wang, Z. Z. Liu, J. Li, H. X. Li, G. G. Yen, Utilizing cumulative population distribution information in differential evolution, *Appl. Soft Comput.*, **48** (2016), 329–346. <https://doi.org/10.1016/j.asoc.2016.07.012>
45. Y. Zhang, D. W. Gong, X. Z. Gao, T. Tian, X. Y. Sun, Binary differential evolution with self-learning for multi-objective feature selection, *Inform. Sciences*, **507** (2020), 67–85. <https://doi.org/10.1016/j.ins.2019.08.040>
46. X. Zhong, P. Cheng, Z. You, An improved differential evolution algorithm based on basis vector type and its application in fringe projection 3D imaging, *Knowl.-Based Syst.*, **268** (2023), 110470. <https://doi.org/10.1016/j.knosys.2023.110470>



AIMS Press

© 2025 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<https://creativecommons.org/licenses/by/4.0>)