Swarm and Evolutionary Computation 84 (2024) 101447

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

journal homepage: www.elsevier.com/locate/swevo

Differential Evolution with perturbation mechanism and covariance matrix based stagnation indicator for numerical optimization

Zhenghao Song, Chongle Ren, Zhenyu Meng *

Institute of Artificial Intelligence, Fujian University of Technology, Fuzhou, China

ARTICLE INFO

Keywords:
Differential evolution (DE)
Crossover strategy
Parameter control
Perturbation mechanism

ABSTRACT

Differential Evolution (DE), as a promising population-based stochastic optimization algorithm, has drawn attention from researchers of various fields owing to its simple operation, strong robustness and few control parameters. However, classic DE suffers from drawbacks such as premature convergence and stagnation resulted from reduced population diversity when tackling complicated optimization problems with high dimensions. To mitigate these deficiencies, Differential Evolution with Perturbation mechanism and Covariance Matrix based stagnation indicator (PCM-DE) is proposed in this paper. There are three main modifications in the algorithm. First, a two-phase parameter adaptation strategy based on fitness value is proposed to guide the search towards promising direction according to different stages of evolution. Second, a perturbation mechanism for archived population is proposed, where a new weight coefficient is defined by exploiting information of fitness value and position in archived individuals. Third, a stagnation indicator based on covariance matrix is proposed to assess the population diversity and the information of variance is employed to perturb stagnant individuals. To validate the effectiveness of PCM-DE, it is compared with several state-of-the-art DE variants and non-DE based algorithms under a large test suite containing 100 benchmark functions. Besides, three truss optimization problems are also employed to verify the feasibility and scalability of the algorithm. The experiment results confirm its highly competitive performance in terms of solution accuracy and convergence speed.

1. Introduction

Differential Evolution (DE), proposed by Price and Storn [1], is an branch of evolutionary algorithms (EAs) based on population. Inspired by the theory of "survival of the fittest", DE comprises four evolutionary operations including population initialization, mutation, crossover and selection. Recently, DE and its variants have attracted extensive attention from many engineering fields and been applied into various real-world applications [2–12].

During the past few years, researchers have incorporated distinct parameter control strategies into the basic framework of DE to avoid parameter tuning and cater for different optimization problems [13–15]. DE has only three control parameters that need to be predefined: the scale factor F, crossover rate CR, and population size PS. Draa et al. [16] introduced cosine functions to adjust F and CR during optimization process. By encoding control parameters into individuals, Wang et al. [17] employed bimodal distribution which contains several probability density function to achieve parameter adaptation. Zhang et al. [18] utilized historical information of successful parameters to adjust the values of parameters during optimization. Tanaba et al. [19] proposed a parameter adaptation scheme based on success-history information, and the new DE variant was known as SHADE. Viktorin et al. [20] replaced the fitness-based parameter adaptation with distance-based one to enhance the robustness of DE for high-dimensional optimization. In [21], a parameter adaptation mechanism based on sine function was proposed, in which two schemes based on sine functions were incorporated into SHADE to adjust control parameters at different stages of evolution. Meng et al. [22] introduced an parameter adaptation mechanism by randomly dividing the population into several subpopulations and updating their corresponding control parameters based on the subpopulation's performance. Yu et al. [23] divided the process of parameter adaptation into two layers, where diversity information of the population is used in one layer and difference between current individuals and global optimum is used in another layer. In [24], wavelet basis function was incorporated into parameter adaptation to control the generation of control parameters. Zhou et al. [25] modified JADE algorithm by proposing a sorting CR scheme. In [26], information of fitness deviation is used to adjust control parameters. Although the above strategies have demonstrated

* Corresponding author.
E-mail address: mzy1314@gmail.com (Z. Meng).

https://doi.org/10.1016/j.swevo.2023.101447
Received 15 May 2023; Received in revised form 15 November 2023; Accepted 6 December 2023
Available online 16 December 2023
2210-6502/© 2023 Elsevier B.V. All rights reserved.

effectiveness in improving search ability, most of them pay attention to the fitness and position information of individuals. However, few focuses have been concentrated on the parameter setting at different phases of evolution.

Besides the modifications made to parameter control, researchers have proposed many strategies based on evolutionary paths by using historical search information to predict evolutionary direction [27,28]. Hansen et al. [29] introduced a covariance matrix to learn the evolutionary state of the current population and using a weighted combination of historical information to predict the direction of evolution. Li et al. [30] proposed a novel DE variant, in which the mean weighted sum of dominant individuals in the historical population is set as the search fixed point. To speed up convergence rate, He et al. [31] utilized historical information of individuals to predict the evolutionary trend and developed a novel mutation operator based on the evolutionary path. Wang et al. [32] proposed a search method by combining current individual historical individuals with their offspring according to their fitness. To avoid premature convergence problem of the Gaussian distribution estimation algorithm, Liang et al. [33] proposed an improved distribution estimation algorithm by using current and historical information to estimate the covariance matrix of the Gaussian model. Zheng et al. [34] proposed a mutation operator based on collective information, generating a search direction by linearly combining the m best individuals in the population according to their fitness. Liu et al. [35] utilized historical and heuristic information to adapt mutation strategies and control parameters. Meng and Yang [36] proposed a novel mutation operator based on historical population to perceive the landscape of the objective function. These methods have improved the search efficiency to some extent, but they did not consider the individual-based information for determining the search direction, which may waste computational resources. Numerical experiments indicate that these methods can indeed improve the local search capacity, but they will weaken the ability of escaping from local optima, which may cause premature convergence or population stagnation.

DE has also found broad applications in real-world problems [37]. For instance, Hu et al. [38] applied DE in parameter identification of the least squares support vector machine for building a multi-level regression model that predicts carbon efficiency while minimizing energy consumption generated during iron ore sintering process. Santucci et al. [39] employed a novel DE to solve permutation flowshop scheduling problems. Hancer et al. [40] introduced a novel multi-objective filtering method which utilizes DE to select optimal feature subsets by incorporating fuzzy and kernel measures as filtering criteria, thus maximizing prediction accuracy. Rivera et al. [41] developed a permutation-based DE algorithm to address feature subset selection problems without the need for a fixed subset size. In the realm of energy systems, Ozyon et al. [42] employed DE to optimize the short-term operation of pumped storage hydropower generator systems. Gong et al. [43] employed DE to tackle the pressing challenge of superpixel segmentation. Yang et al. [44] proposed a multi-level threshold image segmentation method on the basis of DE.

Based on the above analyses, the search capability is greatly enhanced by modifications proposed in those DE variants, but there still exist shortcomings including premature convergence and stagnation. In addition, the increasing decision variables and more complicating relations between them have posed greater challenges to existing algorithms. Therefore, a novel DE with perturbation mechanism and covariance matrix based stagnation indicator known as PCM-DE is proposed to mitigate existing limitations for tackling numerical optimization. The highlights of PCM-DE are described below:

1. A two-phase parameter adaptation strategy is proposed, where F is generated based on the wavelet basis function during the first phase of evolution and according to the Cauchy distribution during the second phase. By dividing the adaptation process into two phases, a sound exploration and exploitation balance can be attained.
2. A perturbation mechanism based on exponential distribution is advanced to deal with the shortcomings of low utilization of individuals stored in external archive. In this mechanism, the archived population are randomly prescreened and perturbed by exponential distribution, thus enhancing the diversity in the external archive.
3. A stagnation indicator based on covariance matrix is employed and a corresponding population diversity enhancement mechanism is performed when a certain individual is detected as stagnant one. Therefore, the population diversity of DE can be enhanced by this mechanism.
4. PCM-DE is compared with eight advanced meta-heuristic algorithms including four DE variants and four non-DE based algorithms under a large test-bed containing 90 benchmark functions. In addition, it is tested on three truss optimization problems and yields satisfactory solution.

The rest of this paper is structured as follows: Section 2 briefly introduce basic operations of DE. The details of PCM-DE are provided in Section 3. Section 4 contains analysis and discussion of experiment results. Section 5 presents the experiment results on real-world application. Finally, Section 6 concludes the paper.

2. The basic operations of DE

In DE, initialization and evolution are the two main stages. The population is created randomly in the stage of initialization. In the stage of evolution, the population is undergone three evolutionary operations including mutation, crossover, and selection, which are iterated until certain termination criteria are met.

Initialization: In DE, initialization is the first step in locating the global optimal solution in the D-dimensional search space. In real-world application, there should be a specific boundary for each dimension of the candidate solution. The target vector can be described as Eq. (1).

$$X_{i,G} = [X_{i,1,G}, X_{i,2,G}, \dots, X_{i,D,G}], \qquad i = 1, 2, \dots, PS \tag{1}$$

where $X_{i,G}$ denotes the ith candidate solution at the Gth generation, and D denotes the dimension of the individual. The jth dimension of ith individual is generated according to Eq. (2).

$$X_{i,j} = X_{min,j} + rand(0,1) \cdot (X_{max,j} - X_{min,j}) \tag{2}$$

where $X_{max,j}$ and $X_{min,j}$ are the upper and lower bounds of the jth dimension of the individual and $rand(0,1)$ is a random number in the interval [0,1].

Mutation: During mutation, a mutant vector $V_{i,G}$ is generated by combining between target vector and difference vector according to a certain mutation strategy. For dealing with various optimization problems, different mutation strategies have been proposed, as presented below:

1. "DE/rand/1"

$$V_{i,G} = X_{r_1,G} + F \cdot (X_{r_2,G} - X_{r_3,G}) \tag{3}$$

2. "DE/best/1"

$$V_{i,G} = X_{best,G} + F \cdot (X_{r_1,G} - X_{r_2,G}) \tag{4}$$

3. "DE/rand/2":

$$V_{i,G} = X_{r_0,G} + F \cdot (X_{r_1,G} - X_{r_2,G}) + F \cdot (X_{r_3,G} - X_{r_4,G}) \tag{5}$$

4. "DE/best/2":

$$V_{i,G} = X_{best,G} + F \cdot (X_{r_1,G} - X_{r_2,G}) + F \cdot (X_{r_3,G} - X_{r_4,G}) \tag{6}$$

5. "DE/target-to-best/1":

$$V_{i,G} = X_{i,G} + F \cdot (X_{best,G} - X_{i,G}) + F \cdot (X_{r_1,G} - X_{r_2,G}) \tag{7}$$

6. "DE/rand/2/dir"

$$V_{i,G} = X_{r_1,G} + \frac{F}{2} \cdot (X_{r_1,G} - X_{r_2,G} - X_{r_3,G}) \tag{8}$$

where r_0 r_1, r_2, r_3 and r_4 are different integers randomly selected from the set $\{1, 2, \ldots, PS\}$. $X_{best,G}$ denotes the best vector at the Gth generation. F is used to controls the scale of difference vector.

Crossover: Through crossover, a trial vector $U_{i,G}$ is produced by exchanging information between the mutant vector $V_{i,G}$ and the target vector $X_{i,G}$.

$$U_{j,i,G} = \begin{cases} V_{j,i,G}, & if\ rand(0,1) \le CR_i\ or j = j_{rand} \\ X_{j,i,G}, & otherwise \end{cases} \tag{9}$$

where j_{rand} is a random integer within [1, D] to ensure that at least one dimension of donor vector is utilized. CR is used to control the number of dimension inherited from the mutant vector.

Selection: The selection in DE is used to determine whether the trial vector or its target vector will survive to the next generation. Before selection, all trial vectors will be evaluated by objective function. The vector with worse fitness will be discarded and the one with better fitness will be sustained.

$$X_{i,G+1} = \begin{cases} U_{i,G}, & if\ f(U_{i,G}) < f(X_{i,G}) \\ X_{i,G}, & otherwise \end{cases} \tag{10}$$

3. The proposed PCM-DE algorithm

This section presents the details of the proposed PCM-DE algorithm, including the two-phase parameter adaptation scheme, the perturbation mechanism for archived individuals, and the population diversity enhancement mechanism based on covariance matrix.

3.1. Two-phase parameter adaptation strategy

The key issue of DE is the high sensitivity of control parameters. A set of control parameters which excels in one optimization problem may perform badly in another optimization problems. F is the key control parameter to determine the size of difference vector. Generally, the value of F is set within [0, 2], and remains unchanged at 0.5 in classic DE. However, different parameter settings are required at different phases of the algorithm, and the fixed control parameters may fail to balance the exploration and exploitation. Therefore, a two-phase parameter adaptation strategy is proposed.

During the first phase of evolution, the generation of F obeys the wavelet basis function owing to its ability on capturing both smoothly and rapidly changing features. By introducing wavelet basis function, the value of F can be kept within the range of [0.4, 0.6], thus mitigating the problem of premature convergence in "DE/current-to-$p$best". During the second phase of evolution, F obeys the Cauchy distribution $F \sim C(\mu_F, 0.1)$. The value of CR should also adapt to the different stage of evolution. When the value of CR is low, the probability of the trial vector inheriting information from the mutant vector decreases. Conversely, when the value of CR is high, the probability of the trial vector inheriting information from the mutant vector increases. In this paper, CR obeys the Gaussian distribution $CR \sim C(\mu_{CR}, 0.1)$, as used in [19]. The range of CR should be defined based on the different stages of evolution. According to the above analysis, assigning a relatively high value to CR during the initial phase of evolution would be beneficial, as it enhances the global exploration ability. The generation of F and CR is presented as follows:.

$$F_i = \begin{cases} \sqrt{2\pi}^{-\frac{1}{3}} \cdot (1 - \mu_{F,r_i}^2) \cdot e^{-\mu_{F,r_i}^2} + 0.1 \cdot rand_i, & if\ nfes < \perp \\ randc_i(\mu_{F,r_i}, 0.1), & otherwise \end{cases} \tag{11}$$

$$CR_i = \begin{cases} 0, & if\ \mu_{CR,r_i} = \emptyset \\ randn_i(\mu_{CR,r_i}, 0.1), & otherwise \end{cases} \tag{12}$$

$$CR_i = \begin{cases} max(CR_i, 0.6), & if\ nfes < \perp \\ max(CR_i, 0), min(CR_i, 1), & otherwise \end{cases} \tag{13}$$

where μ_{F,r_i} and μ_{CR,r_i} denote the control parameters randomly selected in the H entries of memory pool. $\perp$ denotes the threshold between the first and second phases of evolution, equaling to $nfes_{max} \cdot 0.2$. $randn_i$ and $randc_i$ denote the Gaussian distribution and the Cauchy distribution, respectively.

The μ_F and μ_{CR} in the historical memory are also renewed at the end of each generation. If the trial vector is better than its target vector, CR_i and F_i will be labeled as S_{CR} and S_F and stored in successful parameter set. The detailed equations for adaptation are showed in Eqs. (14) and (15).

$$\begin{cases} w_k = \dfrac{\Delta f_k}{\sum_{k=1}^{|S_F|} \Delta f_k} \\ \Delta f_k = |f(X_{i,G}) - f(U_{i,G})| \\ mean_{WL}(S_F) = \dfrac{\sum_{k=1}^{|S_F|} w_k \cdot S_F^2(k)}{\sum_{k=1}^{|S_F|} w_k \cdot S_F(k)} \\ \mu_{F,k,G+1} = \begin{cases} mean_{WL}(S_F), & if\ S_F \neq \emptyset \\ \mu_{F,k,G}, & otherwise \end{cases} \end{cases} \tag{14}$$

$$\begin{cases} mean_{WL}(S_{CR}) = \dfrac{\sum_{k=1}^{|S_{CR}|} w_k \cdot S_{CR}^2(k)}{\sum_{k=1}^{|S_{CR}|} w_k \cdot S_{CR}(k)} \\ \mu_{CR,k,G+1} = \begin{cases} mean_{WL}(S_{CR}), & if\ S_{CR} \neq \emptyset \\ \mu_{CR,k,G}, & otherwise \end{cases} \end{cases} \tag{15}$$

where $mean_{WL}(\cdot)$ is the weighted Lehmer mean. For another parameter population size PS, we employ the population reduction scheme in [45], as shown below:

$$PS_{G+1} = round[\frac{PS_{min} - PS_{max}}{nfes_{max}} \cdot nfes + PS_{max}] \tag{16}$$

where PS_{min} and PS_{max} denote the minimum and maximum population size, respectively.

3.2. Perturbation mechanism for archived population

The importance of the mutation strategy in determining the search direction of DE is widely acknowledged. As a result, selecting an appropriate mutation strategy is crucial for enhancing the algorithm's optimization capability. Among distinct mutation strategies with different characteristics, DE/target-to-$p$best/1 introduced in [18] can achieve a sound balance between exploration and exploitation. Specifically, it employed the external archive to record inferior individuals that are discarded by selection, aiming to enhance the overall population diversity. PCM-DE adopts this mutation strategy, as shown below:

$$V_{i,G} = X_{i,G} + F \cdot (X_{best,G}^p - X_{i,G}) + F \cdot (X_{r_1,G} - \widetilde{X}_{r_2,G}) \tag{17}$$

where X_{best}^p is randomly selected from the top $100p\%$ individuals with $p \in (0, 1]$. X_i and X_{r_1} are selected from P, and $\widetilde{X}_{r_2}$ is chosen from the combination of $\mathbf{P}$ and $\mathbf{A}$, which represents the current population and external archive, respectively.

Although the inferior individuals are utilized in this strategy, they may fail to provide useful information if they are unchanged during the evolution. In addition, the selection used in DE is "one-to-one" competition, which means that the trial vector only compete with its target vector. Even though the target vector fails in the competition, it does not mean that it is indeed "inferior", because it may still have better fitness than other vectors. Therefore, this paper proposes an perturbation mechanism based on exponential distribution to renew the

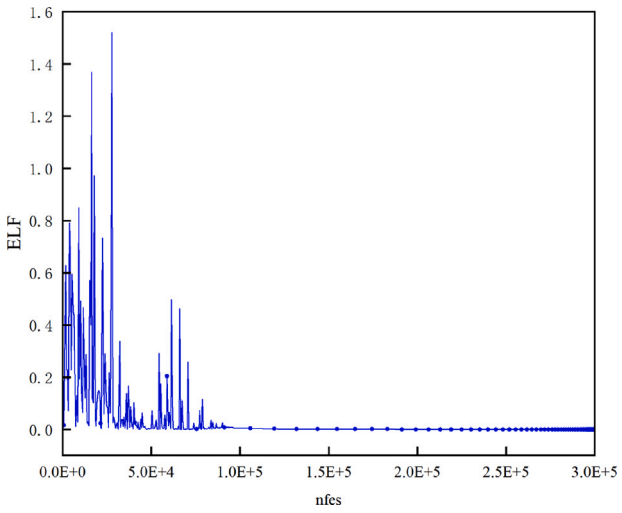


Fig. 1. The changing process of ELF during evolution.

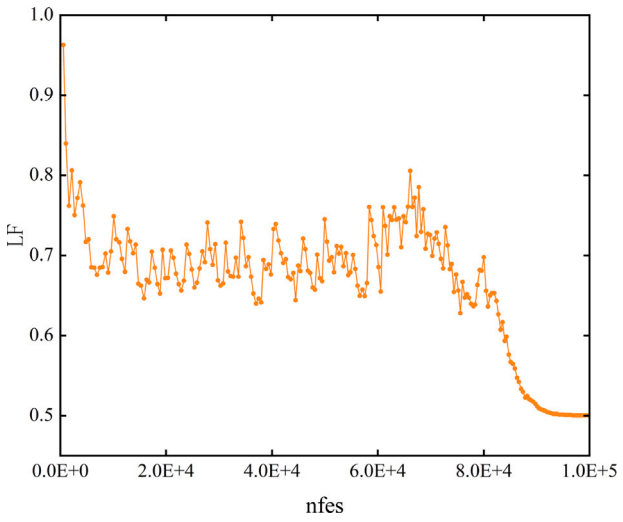


Fig. 2. The changing process of LF during evolution.

archived individuals for better utilization. In this mechanism, a random selection is conducted for the archived individuals, and a quarter of them are chosen to undergo the perturbation. The motivation for this operation is to address the problem that certain individuals in the archive may have poor fitness due to reasons other than being unchanged, such as being assigned to a bad F or CR, or both. Therefore, a random selection is necessary to prevent bias. This approach enables the preservation of useful individuals while introducing perturbations. The perturbation mechanism is described as follows:

$$\begin{cases} LF_i = \frac{f(X_{i,G})}{f(U_{i,G})+f(X_{best,G})} \\ ELF = k \cdot e^{k \cdot LF} \\ \widetilde{X}_{r2} = \widetilde{X}_{r_2} + ELF \cdot (X^p_{best} - \widetilde{X}_{r_2}) \end{cases} \tag{18}$$

where LF denotes the independent variable of the exponential distribution, and k represents a variable that is linearly reduced over time. The LF changes according to the fitness values of the parent, offspring and the best-so-far individual, as shown in Fig. 2. From Fig. 2, we can observe that as the evolution proceeds, the LF tend to converge to 0.5, indicating that individuals are gradually approaching the local optimum. The process is necessary to prevent larger perturbation values caused by a larger LF. The curve of exponential distribution based on LF and k is presented in Fig. 1. It can be observed that in the early stage of evolution, the perturbation value is relatively large, which can enhance the diversity of archived population. As the evolution proceeds, the perturbation value decreases to speed up convergence. Employing the perturbation mechanism based on exponential distribution is able to improve the diversity of the archived population, enabling them to contribute more useful information during mutation operation.

3.3. Population diversity enhancement mechanism based on covariance matrix

The population undergoes different evolutionary states at various stages of evolution. In the early stages, the exploration behavior is preferred. Therefore, a higher degree of population diversity is needed. In the later stages, as the population converges to the optimal solutions, the population diversity inevitably reduces. However, excessively low population diversity will incur problems of premature convergence and stagnation, thus deteriorating the overall optimization performance. As a result, another direction of our research is to analyze the different evolutionary states of the algorithm by analyzing population diversity. Statistical features such as average, variance, and covariance can be used to reflect the population's distribution, among which covariance matrix reflects both population diversity and variable correlations. Therefore, we employed covariance matrix as stagnation indicator to measure the population diversity.

The mechanism comprises two main steps. In the first step, the covariance matrix is constructed to measure the spatial distribution of individuals and a counter Co is used to counter the number of individuals in the current covariance matrix whose variance is less than 0.00001. Then we define a threshold N_1. Once the threshold is met, we assume that current population encounters stagnation. In the second step, the stagnant individual is perturbed using a normalized covariance matrix.

$$CQ_G = Q_G \cdot \sum_{k=1}^{PS} (X_{i,k,G} - \overline{X}_{i,G}) \cdot (X_{j,k,G} - \overline{X}_{j,G}) \tag{19}$$

where Q_G is the D · D matrix consisting of all eigenvectors of the covariance matrix and $(Q_G)^{-1}$ is the inverse matrix of Q_G. Then, the new population after dimension reduction is formed by Eq. (19) for subsequent perturbation.

Algorithm 1 Calculate *count*

```
for i = 1 to PS do
    if f(X_{i,G}) > f(U_{i,G}) then
        count(i) = count(i) + 1
    else
        count(i) = 0;
    end if
end for
```

During the evolution, some individuals may get trapped in local optima, thus potentially resulting in stagnation. Such individuals are referred to as stagnant individuals in this paper. and *count* is used to record the total number of stagnant individuals, when both conditions are met, the stagnation mechanism is started, where N_2 is set to 2 ·D. When the threshold is reached, the stagnant individuals are updated using the following equation:

$$X_{i,G} = CQ_{i,G} + X_{i,G} \tag{20}$$

where $X_{i,G}$ denotes the stagnant individual. CQ denotes the normalization of individuals in the population using the covariance matrix.

Algorithm 2 Pseudocode of PCM-DE

Input: Maximum number of function evaluations $nfes_{max}$, objective function $f(X)$, constraint boundary $[UB, LB]$
Output: Global optimal solution X_{best} and its fitness $f(X_{best})$;

for $i = 1$ **to** PS **do**
  $X_{i,G} = X_i$; Calculate $f(X_{i,G})$;
end for
Find the global best $X_{gbest,G}$ and fitness value $f(X_{gbest,G})$;
while $nfes < nfes_{\max}$ **do**
  $S_{CR} = \emptyset$, $S_F = \emptyset$;
  for $i = 1$ **to** PS **do** ;
    r_i = Select from $[1, H]$ randomly;
    if $\mu_{CR,r_i} = \perp$, $CR_{i,G} = 0$. Otherwise **then**
    end if
    Assign F and CR to each individual using;
    (Eqs. (11) and (12) and (13));
  end for
  for $i = 1$ **to** PS **do**
    if $f(U_{i,G} \le f(X_{i,G})$ **then**
      $X_{i,G+1} = U_{i,G}$;
    else
      $X_{i,G+1} = X_{i,G}$;
    end if
    if $f(U_{i,G} < f(X_{i,G})$ **then**
      $X_{i,G} \to A$;
      $CR_{i,G} \to S_{CR}$, $F_{i,G} \to S_F$;
    end if
  end for
  Update μ_{CR} and μ_F according to Eqs. (14) and (15);
  Calculate A according to Eq. (18);
  It will be randomly removed from the archive,
  making the population size save A;
  Execute Algorithm 1;
  for $j = 1$ **to** D **do**
    if $Co(j, j) > N_1$ **then**
      for $i = 1$ **to** PS **do**
        if $count(i) > N_2$ **then**
          Calculate $X_{i,G}$ according to Eq. (20);
          calculate the fitness value $f(X_{i,G})$;
          $nfes = nfes + 1$;
          $count(i) = 0$;
        end if
      end for
      $Co(j, j) = 0$;
    end if
  end for
  Calculate PS_{G+1} according to Eq. (16);
  $G = G + 1$;
end while

4. Experiment analysis

To verify the overall performance of the newly proposed algorithms, researchers have designed benchmark problems with different characteristics. PCM-DE is firstly validated under CEC2017 test suite [46]. At the same time, in order to avoid problem of overfitting, PCM-DE is also tested under CEC2013 and CEC2014 test suites, and experiment results are provided in Table S-I to Table S-VI of the supplementary file. According to the suggestion of one reviewer, we also examined the performance of PCM-DE on CEC2022 test suite. The comparison results are shown in Table S-XIX to Table S-XXII of the supplementary file. For the function representation, this paper uses f_{a_1}–$f_{a_{28}}$ to represent 28 benchmark functions from CEC2013, f_{b_1}–$f_{b_{30}}$ to represent 30 benchmark functions from CEC2014, and f_{c_1}–$f_{c_{30}}$ to represent 30 benchmark functions from CEC2017 and f_{d_1}–$f_{d_{12}}$ to represent 12 benchmark functions from CEC2022.

Benchmark problems from CEC2017 can be categorized into four groups: f_{c_1}–f_{c_2} belongs to unimodal functions, f_{c_3}–f_{c_9} belongs to simple multimodal functions, $f_{c_{10}}$–$f_{c_{19}}$ belongs to hybrid functions and $f_{c_{20}}$–$f_{c_{30}}$ belongs to composition functions. Besides, the unimodal functions f_{c_1} and f_{c_2} are employed to examine the convergence speed.

All experiments were conducted on a PC with an Intel i7-12700k CPU @3.2 GHz and 32 GB memory on Matlab R2021a. PCM-DE and the comparing algorithms were both run 51 times, and their performance was measured by taking the mean and standard deviation of multiple independent runs. According to the guidance of the CEC2013 special conference [47], $nfes_{max}$ is set to $10\,000 \cdot D$. Eight powerful meta-heuristic algorithms including four DE variants and four non-DE based algorithms are used for comparison. All algorithms are evaluated with the same parameter settings as reported in their original papers, which are summarized in Table 1. For the sake of fairness and clarity, the Wilcoxon signed rank test with a significance level of 0.05 is used to illustrate the differences between the algorithms on all test problems, and the Friedman test is used to provide the overall ranking of the algorithms.

4.1. Comparison with 4 well-known DE variants

Four advanced DE variants, including LSHADE [45], jSO [48], CS-DE [49], and TDE [50], are selected for performance comparison. The experiments are performed on the CEC2017 test suite across different dimensions.

Statistical results on CEC2017 with dimensions ranging from 10, 30 to 50 are presented in Tables 2–4. The symbols “>”, “≈” and “<” indicate that the comparing algorithm outperforms, performs similarly with, and underperforms PCM-DE, respectively. The best result for each function is highlighted in bold.

For unimodal functions f_{c_1}–f_{c_2}, all algorithms can converge to the optimal solution when D = 10. When D = 30, PCM-DE can still converge to the global optimum, whereas CS-DE and TDE only converge to the optimal solution on f_{c_1} and LSHADE and jSO fail to obtain the global optimal solution. When D = 50, PCM-DE outperforms the four well-known DE variants significantly in terms of mean value.

For simple multimodal functions f_{c_3}–f_{c_9}, PCM-DE achieves the optimal solution on f_{c_3}, f_{c_4}, f_{c_6} and f_{c_9} when D = 10, and has the best mean for f_{c_5}, f_{c_7}, and f_{c_8}. When D = 30, PCM-DE converges to the global optimal solution for f_{c_3} and f_{c_9}, while achieving the best mean for f_{c_4} and f_{c_7}. The performance of PCM-DE was only slightly worse than that of LSHADE and TDE for f_{c_5}, f_{c_6}, and f_{c_8}. When D = 50, PCM-DE obtains the best mean for f_{c_4} and achieved the global optimal solution for f_{c_9}, with a mean that is slightly worse than LSHADE but better than other algorithms for f_{c_5}, f_{c_7}, and f_{c_8}. TDE had the best mean for f_{c_6}. Overall, PCM-DE demonstrates competitive optimization accuracy and convergence speed for simple multimodal functions, which highlights the effectiveness of the diversity-based population archiving strategy.

For hybrid Functions $f_{c_{10}}$–$f_{c_{19}}$, when D = 10, PCM-DE obtains the global optimal solution on functions $f_{c_{11}}$ and $f_{c_{14}}$, and achieves the best mean on functions $f_{c_{10}}$, $f_{c_{12}}$, $f_{c_{13}}$, $f_{c_{15}}$, and $f_{c_{18}}$, being only worse than LSHADE on $f_{c_{16}}$ and $f_{c_{17}}$, and worse than jSO on $f_{c_{19}}$. When D = 30, PCM-DE achieves the best mean on functions $f_{c_{12}}$, $f_{c_{13}}$, $f_{c_{17}}$, and $f_{c_{18}}$, and is only slightly worse than LSHADE on $f_{c_{11}}$ and $f_{c_{16}}$, and slightly worse than jSO on $f_{c_{11}}$ and $f_{c_{15}}$, while being worse than TDE only on $f_{c_{14}}$ and $f_{c_{19}}$. When D = 50, PCM-DE achieves the best mean on $f_{c_{11}}$–$f_{c_{19}}$.

For composition functions $f_{c_{20}}$–$f_{c_{30}}$, when D = 10, PCM-DE achieves the global optimal solution on function $f_{c_{20}}$ and obtains the best mean on functions $f_{c_{21}}$, $f_{c_{28}}$, and $f_{c_{29}}$, while being worse than CS-DE on functions f_{c_2}–$f_{c_{25}}$ and $f_{c_{30}}$. When D = 30, PCM-DE has the best mean on functions $f_{c_{27}}$–$f_{c_{29}}$, while being worse than TDE on $f_{c_{23}}$, $f_{c_{24}}$, and $f_{c_{26}}$, and worse than LSHADE and jSO on functions $f_{c_{21}}$ and $f_{c_{30}}$. When

Table 1
Recommended parameter settings of all algorithms.

Algorithm	Parameters initial settings
LSHADE	$\mu_F = 0.5,\ \mu_{CR} = 0.5,\ F \sim C(\mu_F, 0.1),\ PS = 18 \cdot D \sim 4,\ CR \sim C(\mu_{CR}, 0.1),\ r^{rac} = 2.6,\ p = 0.11,\ H = 6$
jSO	$\mu_F = 0.3,\ \mu_{CR} = 0.8,\ F, CR\&r^{rac}$ same as LSHADE, $PS = 25 \cdot ln(D) \cdot \sqrt{D} \sim 4,\ p = 0.25 \sim 0.125,\ H = 5$
CS-DE	$\mu_F = 0.8,\ \mu_{CR} = 0.6,\ F\&CR$ same as LSHADE, $p = 0.25 \sim 0.05,\ PS = 25 \cdot ln(D) \cdot \sqrt{D} \sim 4,\ r^{rac} = 1.6,\ r^{rac,A} = 1.6,\ r^{rac,B} = 5,\ T_0 = \frac{G_{max}}{2},\ K = 4$
TDE	$\mu_F = 0.3,\ \mu_{CR} = 0.8,\ F\&CR$ same as LSHADE, $p = 0.11,\ PS = 25 \cdot ln(D) \cdot \sqrt{D} \sim 4,\ r^{rac,A} = 1.6,\ r^{rac,B} = 3,\ p = \frac{2}{3} \cdot nfes_{max}$
PSO-sono	$PS = 100,\ r = 0.5,\ \epsilon = \frac{D}{PS} \cdot 0.01,\ iw \in [0.4, 0.9]$
HSES	$cc = 0.96,\ I = 20,\ M = 200,\ PS = 100$
E-QUATRE	$PS = 100,\ T_0 = 70,\ p_{max} = 0.2,\ p_{min} = 0,\ r^{rac} = 1.6$
EA4eig	$PS_{ini} = 100,\ PS_{min} = 10,\ q_h = \frac{1}{H},\ H = 4$
PCM-DE	$\mu_F = 0.5,\ \mu_{CR} = 0.8,\ F\&CR$ same as LSHADE, $p = 0.11,\ r^{rac} = 1.4,\ PS = 25 \cdot ln(D) \cdot \sqrt{D} \sim 4,\ H = 4,\ N_2 = 2 \cdot D,\ N_1 = 0.0001,\ k = 4 \sim 40$

Table 2
Algorithm comparison between LSHADE, jSO, CS-DE, TDE and PCM-DE on 10D optimization of our test suite.

DE variants No.	LSHADE Mean/Std	jSO Mean/Std	CS-DE Mean/Std	TDE Mean/Std	PCM-DE Mean/Std
f_{c_1}	*0/0(≈)*	*0/0(≈)*	*0/0(≈)*	*0/0(≈)*	*0/0*
f_{c_2}	*0/0(≈)*	*0/0(≈)*	*0/0(≈)*	*0/0(≈)*	*0/0*
f_{c_3}	*0/0(≈)*	*0/0(≈)*	*0/0(≈)*	*0/0(≈)*	*0/0*
f_{c_4}	*0/0(≈)*	*0/0(≈)*	*0/0(≈)*	*0/0(≈)*	*0/0*
f_{c_5}	2.6361E+00/8.1587E−01(<)	1.8729E+00/8.3593E−01(<)	2.4606E+00/7.7859E−01(<)	3.0073E+00/1.4136E+00(<)	**1.6543E+00/1.7640E+00**
f_{c_6}	4.4583E−15/2.2287E−14(<)	*0/0(≈)*	4.4583E−14/5.6058E−14(<)	8.6937E−14/4.8704E−14(<)	*0/0*
f_{c_7}	1.2097E+01/6.3890E−01(<)	1.2078E+01/7.2070E−01(<)	1.2301E+01/6.0065E−01(<)	1.2308E+01/1.1318E+00(<)	**1.1998E+01/1.4285E+00**
f_{c_8}	2.4414E+00/8.7339E−01(<)	2.0289E+00/8.4333E−01(<)	2.5518E+00/7.9801E−01(<)	2.7738E+00/1.7145E+00(<)	**1.8129E+00/1.8406E+00**
f_{c_9}	*0/0(≈)*	*0/0(≈)*	*0/0(≈)*	*0/0(≈)*	*0/0*
$f_{c_{10}}$	2.8095E+01/4.2078E+01(<)	2.8622E+01/4.5822E+01(<)	5.6217E+01/5.9176E+01(<)	4.5781E+01/9.2213E+01(<)	**2.0125E+01/3.6804E+01**
$f_{c_{11}}$	3.9590E−01/6.8874E−01(<)	4.4583E−15/3.1839E−14(<)	1.5529E+00/6.5403E−01(<)	8.5579E−02/3.2067E−01(<)	**0/0**
$f_{c_{12}}$	3.4019E+01/5.4563E+01(<)	5.0792E+00/2.3459E+01(<)	1.6843E+01/4.1374E+01(<)	3.0609E−01/1.9236E−01(<)	**5.7100E−03/4.7695E+01**
$f_{c_{13}}$	3.5270E+00/2.2884E+00(<)	2.9004E+00/2.3060E+00(<)	1.5403E+00/2.1830E+00(<)	2.0295E+00/2.3799E+00(<)	**8.2650E−01/2.9901E+00**
$f_{c_{14}}$	2.0907E−01/4.3055E−01(<)	1.3656E−01/3.4579E−01(<)	3.3571E−01/5.1432E−01(<)	5.5177E−01/7.4937E−01(<)	**0/0**
$f_{c_{15}}$	1.8762E−01/2.1487E−01(<)	2.9709E−01/2.1862E−01(<)	1.2969E−01/1.6833E−01(<)	1.3775E−01/1.9055E−01(<)	**1.0215E−01/1.5607E−01**
$f_{c_{16}}$	**3.4327E−01/1.7852E−01**(>)	5.1955E−01/2.9781E−01(<)	5.0583E−01/1.6584E−01(<)	5.3994E−01/2.7718E−01(<)	3.8878E−01/1.9546E−01
$f_{c_{17}}$	**1.3891E−01/1.4731E−01**(>)	4.9298E−01/3.9129E−01(<)	2.2755E−01/2.2790E−01(<)	1.6916E−01/1.7433E−01(<)	1.6671E−01/2.2737E−01
$f_{c_{18}}$	2.3745E−01/2.0231E−01(<)	2.5619E−01/1.9455E−01(<)	2.6464E−01/1.8584E−01(<)	2.2620E−01/1.9216E−01(<)	**2.2388E−01/1.9686E−01**
$f_{c_{19}}$	1.1055E−02/1.0560E−02(<)	**9.4262E−03/1.0798E−02**(>)	1.8034E−02/1.2768E−02(<)	1.2058E−02/1.0601E−02(<)	1.0199E−02/1.3424E−02
$f_{c_{20}}$	*0/0(≈)*	3.6114E−01/1.3055E−01(<)	*0/0(≈)*	1.2242E−02/6.1198E−02(<)	*0/0*
$f_{c_{21}}$	1.5359E+02/5.1750E+01(<)	1.4656E+02/5.1886E+01(<)	1.4607E+02/4.8991E+01(<)	1.4680E+02/5.2163E+01(<)	**1.4490E+02/5.2290E+01**
$f_{c_{22}}$	1.0001E+02/5.6308E−02(<)	1.0000E+02/1.0925E−13(≈)	**9.8902E+01/7.8434E+00**(>)	1.0005E+02/1.1986E−01(<)	1.0000E+02/1.4215E−13
$f_{c_{23}}$	3.0328E+02/1.4013E+00(<)	3.0146E+02/1.6873E+00(<)	**2.9630E+02/4.2299E+01**(>)	3.0218E+02/2.0513E+00(<)	3.0078E+02/1.4386E+00
$f_{c_{24}}$	3.0366E+02/7.5121E+01(<)	2.7055E+02/1.0084E+02(>)	**2.6545E+02/1.0244E+02**(>)	3.0064E+02/7.4376E+01(<)	2.8148E+02/9.0945E+01
$f_{c_{25}}$	4.0881E+02/1.9694E+01(>)	4.0953E+02/1.9977E+01(>)	**4.0684E+02/1.8204E+01**(>)	4.1305E+02/2.1643E+01(>)	4.1575E+02/2.2387E+01
$f_{c_{26}}$	*3.0000E+02/0(≈)*	*3.0000E+02/0(≈)*	*3.0000E+02/0(≈)*	*3.0000E+02/0(≈)*	*3.0000E+02/0*
$f_{c_{27}}$	3.8941E+02/2.1287E−01(>)	**3.8936E+02/2.4015E−01**(>)	3.9288E+02/1.7221E+00(>)	3.9278E+02/1.8967E+00(>)	3.9378E+02/1.1343E+00
$f_{c_{28}}$	3.5894E+02/1.2069E+02(<)	3.1340E+02/5.7489E+01(<)	3.0556E+02/3.9732E+01(<)	3.1056E+02/5.2812E+01(<)	**3.0448E+02/9.5485E+01**
$f_{c_{29}}$	2.3410E+02/2.7555E+00(<)	2.3394E+02/3.0671E+00(<)	2.3795E+02/5.0444E+00(<)	2.3419E+02/4.2552E+00(<)	**2.3162E+02/2.8417E+00**
$f_{c_{30}}$	3.2446E+04/1.6020E+05(<)	3.9452E+02/4.3407E−02(>)	**3.9451E+02/2.3305E−02**(>)	3.9453E+02/5.4511E−02(>)	4.0315E+02/1.9983E+01
>/≈/<	4/7/19	5/8/17	6/7/17	3/6/21	–/–/–

D = 50, PCM-DE is only worse than LSHADE and TDE on functions $f_{c_{21}}$ and $f_{c_{22}}$, and achieves the best mean on all other functions.

In summary, the experimental results demonstrate that PCM-DE outperforms four DE variants for most benchmark functions. In total 90 cases comparing with LSHADE, jSO, CS-DE and TDE, PCM-DE achieves superior performance on 66, 67, 63 and 66 cases, achieves similar performance on 9, 11, 13 and 12 cases, and achieves worse performance on 15, 12, 14 and 12 cases, respectively. From the comparison results, we can conclude that the improved optimization accuracy with increasing dimensionality can be attributed to diversity enhancement mechanism which prevents individuals from trapping into local optima.

4.2. Comparison with 4 non-DE based algorithms

In this subsection, we conducted a comparison between PCM-DE and four non-DE based algorithms, namely PSO-sono [51], HSES [52], E-QUATRE [53], and EA4eig [54]. PSO-sono is a novel PSO variant based on sorted particle swarm with hybrid paradigms, new adaptation scheme and a fully-informed search method. In HSES, a hybrid sampling evolutionary strategy is presented to address single-peak and multi-peak problems. E-QUATRE is a population-based cooperative optimization algorithm that utilizes an automatically generated evolution matrix M to mitigate the exploration bias in DE. EA4eig develops a collaborative model of four well-performing evolution algorithms that are enhanced by feature crossover. The experimental results under CEC2017 on 10D, 30D and 50D optimization are presented in Tables 5–7, respectively.

From the tables, it is obvious that PCM-DE achieves the best mean value for f_{c_1}–f_{c_4}, f_{c_6}, f_{c_7}–$f_{c_{14}}$, $f_{c_{17}}$, $f_{c_{20}}$, and $f_{c_{23}}$ for f_{c_1}–f_{c_3}, f_{c_5}, f_{c_7}–f_{c_9}, $f_{c_{11}}$, $f_{c_{13}}$, $f_{c_{15}}$–$f_{c_{17}}$, $f_{c_{20}}$–$f_{c_{23}}$, and $f_{c_{26}}$–$f_{c_{28}}$ on 30D; and for f_{c_1}–f_{c_3}, f_{c_7}, f_{c_9}, $f_{c_{11}}$, $f_{c_{13}}$–$f_{c_{14}}$, $f_{c_{16}}$–$f_{c_{18}}$, $f_{c_{21}}$, $f_{c_{23}}$, and $f_{c_{29}}$ on 50D. Additionally, from Table 9, it is evident that PCM-DE secures 87 performance improvements out of 90 cases comparing with PSO-sono, secures 67 performance improvements out of 90 cases comparing with E-QUATER, 62 out of 90 cases in comparison with HSES, and 66 out of 90 cases in comparison with EA4eig. The experimental results indicate that PCM-DE has a highly competitive advantage not only over advanced DE variants but also non-DE based algorithms.

4.3. Statistical analysis

A comparative study was conducted between PCM-DE and four DE variants and non-DE based algorithms under CEC2017 test suite with dimensions ranging from 10, 30 to 50.

Table 3
Algorithm comparison between LSHADE, jSO, CS-DE, TDE and PCM-DE on 30D optimization of our test suite.

DE variants No.	LSHADE Mean/Std	jSO Mean/Std	CS-DE Mean/Std	TDE Mean/Std	PCM-DE Mean/Std
f_{c_1}	2.7864E−16/1.9899E−15(<)	1.3932E−15/4.2679E−15(<)	*0/0(≈)*	*0/0(≈)*	*0/0*
f_{c_2}	3.3437E−15/9.2483E−15(<)	2.7864E−15/8.5358E−15(<)	8.3593E−15/1.5352E−14(<)	2.2292E−15/7.7172E−15(<)	**0/0**
f_{c_3}	3.3437E−15/1.3508E−14(<)	4.2354E−14/2.5019E−14(<)	*0/0(≈)*	*0/0(≈)*	*0/0*
f_{c_4}	5.8562E+01/2.1269E−14(<)	5.8562E+01/2.7847E−14(<)	5.7413E+01/8.2003E+00(<)	5.5480E+01/1.3810E+01(<)	**5.2522E+01/8.2526E+00**
f_{c_5}	**6.3199E+00/1.5399E+00**(>)	9.0011E+00/1.9905E+00(<)	1.1038E+01/1.6227E+00(<)	9.7720E+00/1.8951E+00(<)	6.7782E+00/1.4887E+00
f_{c_6}	5.3677E−09/2.6833E−08(>)	8.0803E−09/3.2519E−08(>)	1.7445E−08/4.4548E−08(>)	**1.1146E−13/1.5919E−14**(>)	1.6707E−06/4.5762E−06
f_{c_7}	3.7385E+01/1.3641E+00(<)	3.8746E+01/1.9227E+00(<)	4.0215E+01/1.7286E+00(<)	3.8798E+01/1.9240E+00(<)	**3.7102E+01/1.0312E+00**
f_{c_8}	**7.3618E+00/1.4181E+00**(>)	8.8054E+00/1.6649E+00(<)	1.1463E+01/1.9147E+00(<)	9.9245E+00/2.4457E+00(<)	7.4233E+00/1.7401E+00
f_{c_9}	*0/0(≈)*	*0/0(≈)*	*0/0(≈)*	*0/0(≈)*	*0/0*
$f_{c_{10}}$	**1.4191E+03/2.0794E+02**(>)	1.5401E+03/2.3970E+02(<)	1.6281E+03/2.1799E+02(<)	1.5419E+03/2.2276E+02(<)	1.4202E+03/2.3505E+02
$f_{c_{11}}$	1.8634E+01/2.4363E+01(<)	**4.1677E+00/8.5999E+00**(>)	9.3239E+00/1.6467E+01(<)	9.1811E+00/1.6329E+01(<)	5.4838E+00/1.1406E+01
$f_{c_{12}}$	1.0373E+03/3.3002E+02(<)	1.6009E+02/9.2030E+01(<)	7.8296E+02/2.9598E+02(<)	4.6572E+02/2.3659E+02(<)	**1.1594E+02/7.0382E+01**
$f_{c_{13}}$	1.5708E+01/5.2229E+00(<)	1.6778E+01/3.3253E+00(<)	1.4849E+01/5.9753E+00(<)	1.3915E+01/5.7923E+00(<)	**1.2908E+01/5.8705E+00**
$f_{c_{14}}$	2.1348E+01/4.3814E+00(<)	2.1457E+01/3.2159E+00(<)	2.2452E+01/3.3134E+00(<)	**1.9994E+01/8.2217E+00**(>)	2.0450E+01/5.1838E+00
$f_{c_{15}}$	4.0008E+00/2.3587E+00(<)	**1.0103E+00/7.0740E−01**(>)	2.8842E+00/1.3246E+00(<)	2.1867E+00/1.2635E+00(<)	1.1445E+00/7.7129E−01
$f_{c_{16}}$	**5.1347E+01/4.3559E+01**(>)	5.4949E+01/6.7056E+01(>)	1.9707E+02/1.0213E+02(<)	1.6426E+02/8.9100E+01(<)	6.3177E+01/6.7323E+01
$f_{c_{17}}$	3.1879E+01/6.2712E+00(<)	3.3496E+01/6.4682E+00(<)	3.6153E+01/6.7814E+00(<)	3.2809E+01/7.3068E+00(<)	**2.6186E+01/7.0129E+00**
$f_{c_{18}}$	2.1763E+01/8.8785E−01(<)	2.0765E+01/3.3103E−01(<)	2.1506E+01/8.0943E−01(<)	2.1268E+01/7.1838E−01(<)	**2.0573E+01/1.4791E−01**
$f_{c_{19}}$	5.3009E+00/1.2616E+00(<)	4.6243E+00/1.7513E+00(<)	5.5169E+00/2.0586E+00(<)	**3.9648E+00/9.8463E−01**(>)	4.0954E+00/1.2936E+00
$f_{c_{20}}$	3.1817E+01/6.3336E+00(<)	**2.9930E+01/7.9389E+00**(>)	3.9653E+01/8.0348E+00(<)	3.3117E+01/9.1059E+00(<)	3.1486E+01/6.1854E+00
$f_{c_{21}}$	**2.0710E+02/1.4812E+00**(>)	2.0957E+02/1.8002E+00(<)	2.1145E+02/2.0328E+00(<)	2.0913E+02/2.1731E+00(<)	2.0728E+02/1.6532E+00
$f_{c_{22}}$	*1.0000E+02/1.4352E−14(≈)*	*1.0000E+02/1.0047E−13(≈)*	*1.0000E+02/1.4352E−14(≈)*	*1.0000E+02/1.4352E−14(≈)*	*1.0000E+02/6.3901E−14*
$f_{c_{23}}$	3.4950E+02/2.7650E+00(<)	3.5041E+02/3.8216E+00(<)	3.4809E+02/3.5422E+00(<)	**3.4314E+02/3.4879E+00**(>)	3.4457E+02/2.8645E+00
$f_{c_{24}}$	4.2552E+02/1.7469E+00(<)	4.2729E+02/2.5303E+00(<)	4.2180E+02/2.3801E+00(<)	**4.1962E+02/3.5323E+00**(>)	4.2082E+02/1.6896E+00
$f_{c_{25}}$	3.8674E+02/2.3732E−02(<)	*3.8670E+02/1.0235E−02(≈)*	3.8675E+02/1.1362E−02(<)	*3.8670E+02/2.2770E−01(≈)*	*3.8670E+02/9.7496E−03*
$f_{c_{26}}$	9.2546E+02/3.4801E+01(<)	9.3878E+02/4.2029E+01(<)	9.3195E+02/4.1173E+01(<)	**8.5829E+02/4.1708E+01**(>)	8.7413E+02/4.3716E+01
$f_{c_{27}}$	5.0334E+02/5.8744E+00(<)	4.9773E+02/6.7188E+00(<)	5.0302E+02/5.3567E+00(<)	4.9585E+02/6.4060E+00(<)	**4.9284E+02/8.7525E+00**
$f_{c_{28}}$	3.3305E+02/5.2139E+01(<)	3.1320E+02/3.6533E+01(<)	3.2333E+02/4.4992E+01(<)	3.2821E+02/4.8782E+01(<)	**3.0873E+02/3.0250E+01**
$f_{c_{29}}$	4.3156E+02/7.1649E+00(<)	4.3590E+02/1.1192E+01(<)	4.4216E+02/6.9851E+00(<)	4.3323E+02/9.1769E+00(<)	**4.3150E+02/7.9950E+00**
$f_{c_{30}}$	1.9910E+03/5.1984E+01(<)	**1.9708E+03/2.6363E+01**(>)	2.0122E+03/4.4439E+01(<)	1.9965E+03/2.3905E+01(<)	1.9866E+03/2.0265E+01
>/≈/<	6/2/22	6/3/21	1/4/25	6/5/19	–/–/–

Table 4
Algorithm comparison between LSHADE, jSO, CS-DE, TDE and PCM-DE on 50D optimization of our test suite.

DE variants No.	LSHADE Mean/Std	jSO Mean/Std	CS-DE Mean/Std	TDE Mean/Std	PCM-DE Mean/Std
f_{c_1}	1.7555E−14/6.0880E−15(<)	2.9258E−14/8.7194E−15(<)	1.5604E−14/4.2679E−15(<)	1.3654E−14/2.7859E−15(<)	**1.2604E−14/4.2679E−15**
f_{c_2}	1.0092E−12/5.7168E−12(<)	1.8391E−13/7.1747E−13(<)	6.2416E−14/9.3579E−14(<)	2.0062E−14/1.9103E−14(<)	**1.2260E−14/1.5311E−14**
f_{c_3}	1.6496E−13/5.8253E−14(<)	2.6750E−13/7.2166E−14(<)	1.2260E−13/3.8339E−14(<)	1.1257E−13/3.1114E−14(<)	**7.2447E−14/2.5620E−14**
f_{c_4}	7.4401E+01/5.0815E+01(<)	5.2699E+01/4.3116E+01(<)	8.1661E+01/4.7635E+01(<)	7.5297E+01/4.6764E+01(<)	**4.3170E+01/3.6193E+01**
f_{c_5}	**1.2781E+01/2.0421E+00**(>)	1.6472E+01/3.3435E+00(<)	2.2319E+01/2.8008E+00(<)	1.9692E+01/3.5045E+00(<)	1.3944E+01/2.0279E+00
f_{c_6}	1.3996E−07/2.6902E−07(>)	4.5306E−07/1.0754E−06(>)	5.0715E−07/6.7352E−07(>)	**5.6404E−09/1.5600E−08**(>)	1.7639E−03/1.3880E−03
f_{c_7}	**6.2661E+01/2.0497E+00**(>)	6.6291E+01/2.8331E+00(<)	6.7415E+01/3.0906E+00(<)	6.6370E+01/3.8529E+00(<)	6.2981E+01/1.6221E+00
f_{c_8}	**1.2592E+01/2.1023E+00**(>)	1.7579E+01/2.9018E+00(<)	2.1839E+01/2.4886E+00(<)	2.0500E+01/3.3224E+00(<)	1.3739E+01/2.0600E+00
f_{c_9}	6.0187E−14/5.7310E−14(<)	7.5791E−14/5.4126E−14(<)	*0/0(≈)*	*0/0(≈)*	*0/0*
$f_{c_{10}}$	3.1037E+03/3.2509E+02(<)	3.1900E+03/2.9785E+02(<)	3.1401E+03/3.7133E+02(<)	3.1186E+03/3.1459E+02(<)	**3.0814E+03/2.7442E+02**
$f_{c_{11}}$	4.9702E+01/8.0592E+00(<)	2.7732E+01/3.2713E+00(<)	4.5959E+01/7.3351E+00(<)	3.9154E+01/5.9297E+00(<)	**2.2328E+01/1.8919E+00**
$f_{c_{12}}$	2.1818E+03/4.5692E+02(<)	1.6365E+03/4.2894E+02(<)	2.1772E+03/4.6676E+02(<)	1.7829E+03/5.1572E+02(<)	**1.6271E+03/4.7303E+02**
$f_{c_{13}}$	4.9925E+01/2.3914E+01(<)	3.1840E+01/1.7620E+01(<)	4.9846E+01/1.9987E+01(<)	3.9641E+01/2.3152E+01(<)	**2.1928E+01/1.5427E+01**
$f_{c_{14}}$	2.9563E+01/3.3497E+00(<)	2.4844E+01/1.9528E+00(<)	2.8735E+01/2.4162E+00(<)	2.7634E+01/2.3058E+00(<)	**2.4748E+01/1.8219E+00**
$f_{c_{15}}$	3.8002E+01/8.6118E+00(<)	2.3728E+01/3.0491E+00(<)	3.6408E+01/8.7737E+00(<)	2.8765E+01/3.2275E+00(<)	**2.2053E+01/2.4843E+00**
$f_{c_{16}}$	3.4340E+02/1.2380E+02(<)	4.4427E+02/1.3553E+02(<)	5.0217E+02/1.2300E+02(<)	3.9703E+02/1.0713E+02(<)	**3.1555E+02/1.1429E+02**
$f_{c_{17}}$	2.4250E+02/6.8884E+01(<)	2.7824E+02/9.4804E+01(<)	3.6134E+02/8.0999E+01(<)	3.3001E+02/8.7503E+01(<)	**2.4060E+02/4.8050E+01**
$f_{c_{18}}$	4.0906E+01/1.4363E+01(<)	2.4922E+01/2.1399E+00(<)	3.3606E+01/6.2667E+00(<)	2.7780E+01/3.3531E+00(<)	**2.1157E+01/8.7311E−01**
$f_{c_{19}}$	2.4516E+01/6.3755E+00(<)	1.4514E+01/2.5875E+00(<)	2.1504E+01/4.0598E+00(<)	1.5292E+01/2.3821E+00(<)	**1.1106E+01/2.1103E+00**
$f_{c_{20}}$	1.5170E+02/5.2989E+01(<)	1.2969E+02/7.4729E+01(<)	2.0718E+02/6.9671E+01(<)	1.9350E+02/8.1794E+01(<)	**1.1661E+02/3.2452E+01**
$f_{c_{21}}$	**2.1289E+02/2.5945E+00**(>)	2.1880E+02/3.1300E+00(<)	2.2416E+02/2.6529E+00(<)	2.2150E+02/3.9995E+00(<)	2.1473E+02/2.1676E+00
$f_{c_{22}}$	2.0179E+03/1.7758E+03(<)	1.5263E+03/1.8031E+03(<)	3.3004E+02/8.8254E+02(<)	**1.6813E+02/4.7342E+02**(>)	2.9593E+02/7.9543E+02
$f_{c_{23}}$	4.3021E+02/3.9508E+00(<)	4.3274E+02/6.0158E+00(<)	4.2996E+02/4.9945E+00(<)	4.2353E+02/6.0631E+00(<)	**4.2024E+02/7.7018E+00**
$f_{c_{24}}$	5.0668E+02/2.9836E+00(<)	5.0930E+02/3.7851E+00(<)	5.0145E+02/5.5717E+00(<)	5.0076E+02/5.9937E+00(<)	**4.9901E+02/5.2595E+00**
$f_{c_{25}}$	4.8336E+02/1.2014E+01(<)	4.8116E+02/3.1464E+00(<)	4.8468E+02/1.0853E+01(<)	4.9015E+02/2.0005E+01(<)	**4.8097E+02/5.4072E+00**
$f_{c_{26}}$	1.1394E+03/4.4333E+01(<)	1.1578E+03/5.1982E+01(<)	1.1598E+03/6.1031E+01(<)	1.0764E+03/6.9792E+01(<)	**1.0371E+03/7.7311E+01**
$f_{c_{27}}$	5.3343E+02/1.8105E+01(<)	5.1152E+02/1.0414E+01(<)	5.2731E+02/6.9044E+00(<)	5.1365E+02/1.4209E+01(<)	**5.0851E+02/6.5458E+00**
$f_{c_{28}}$	4.6938E+02/2.0290E+01(<)	4.6172E+02/1.1607E+01(<)	4.8279E+02/2.4661E+01(<)	4.8077E+02/2.1005E+01(<)	**4.5885E+02/2.2278E−13**
$f_{c_{29}}$	3.5285E+02/1.0007E+01(<)	3.6471E+02/1.4654E+01(<)	3.8120E+02/1.3547E+01(<)	3.5675E+02/1.5458E+01(<)	**3.5152E+02/1.0478E+01**
$f_{c_{30}}$	6.5923E+05/7.0107E+04(<)	6.0465E+05/3.5617E+04(<)	6.0998E+05/3.5421E+04(<)	6.0586E+05/3.2428E+04(<)	**5.9243E+05/2.1917E+04**
>/≈/<	5/0/25	1/0/29	1/1/28	2/1/27	–/–/–

Table 5
Algorithm comparison between PSO-sono, E-QUATRE, HSES, EA4eig and PCM-DEs on 10D optimization of our test suite.

Non-DE algorithms No.	PSO-sono Mean/Std	E-QUATRE Mean/Std	HSES Mean/Std	EA4eig Mean/Std	PCM-DE Mean/Std
f_{c_1}	6.3116E+02/1.1615E+03(<)	*0/0(≈)*	5.9226E−11/5.9044E−11(<)	7.9196E−09/1.3179E−09(<)	*0/0*
f_{c_2}	1.3423E−05/1.3838E−05(<)	*0/0(≈)*	1.1171E−10/1.9178E−10(<)	7.4903E−09/2.0053E−09(<)	*0/0*
f_{c_3}	*0/0(≈)*	*0/0(≈)*	*0/0(≈)*	7.9623E−09/1.5244E−09(<)	*0/0*
f_{c_4}	1.7870E−09/4.3894E−09(<)	*0/0(≈)*	1.2874E−11/1.6782E−11(<)	7.7607E−09/1.7080E−09(<)	*0/0*
f_{c_5}	6.6916E+00/2.7435E+00(<)	2.6525E+00/1.1569E+00(<)	**7.2183E−01/8.2235E−01**(>)	1.4046E+00/1.1802E+00(>)	1.6543E+00/1.7640E+00
f_{c_6}	3.8940E−03/9.3890E−03(<)	1.1039E−11/7.2404E−12(<)	3.1190E−11/1.9589E−11(<)	9.0013E−09/9.1598E−10(<)	**0/0**
f_{c_7}	1.7566E+01/4.3691E+00(<)	1.3843E+01/1.3892E+00(<)	**1.1306E+01/7.0038E−01**(>)	1.1759E+01/9.5348E−01(>)	1.1998E+01/1.4285E+00
f_{c_8}	7.6670E+00/3.8137E+00(<)	2.8627E+00/1.1253E+00(<)	**4.8773E−01/7.2768E−01**(>)	1.3851E+00/1.2117E+00(>)	1.8129E+00/1.8406E+00
f_{c_9}	2.1197E−02/9.3206E−02(<)	*0/0(≈)*	2.1199E−12/8.0378E−12(<)	8.2683E−09/1.3542E−09(<)	*0/0*
$f_{c_{10}}$	3.7609E+02/2.2854E+02(<)	1.0447E+02/7.1206E+01(<)	8.4365E+01/1.2413E+02(<)	5.3758E+01/6.9689E+01(<)	**2.0125E+01/3.6804E+01**
$f_{c_{11}}$	1.4363E+01/1.1600E+01(<)	7.8615E−02/2.6103E−01(<)	1.1705E−01/3.8002E−01(<)	8.0232E−09/1.3697E−09(<)	**0/0**
$f_{c_{12}}$	6.8123E+03/7.7539E+03(<)	7.2638E+00/2.8434E+01(<)	3.2353E+01/8.2139E+01(<)	4.8742E+00/2.3327E+01(<)	**5.7100E−03/4.7695E+01**
$f_{c_{13}}$	1.3453E+02/1.2087E+02(<)	1.3243E+00/1.6830E+00(<)	3.2175E+00/2.7429E+00(<)	1.0727E+00/1.9334E+00(<)	**8.2650E−01/2.9901E+00**
$f_{c_{14}}$	4.5170E+01/2.0260E+01(<)	1.6192E−03/5.8597E−03(<)	2.0780E+00/6.7180E+00(<)	8.2413E−09/1.4386E−09(<)	**0/0**
$f_{c_{15}}$	4.0651E+01/3.5330E+01(<)	9.1302E−02/8.7164E−02(>)	5.3020E−01/5.0977E−01(<)	**1.1985E−02/3.5835E−02**(>)	1.0215E−01/1.5607E−01
$f_{c_{16}}$	1.0114E+01/2.9761E+01(<)	6.2828E−01/1.7731E−01(<)	5.4388E+00/2.3174E+01(<)	**1.8707E−01/1.8006E−01**(>)	3.8878E−01/1.9546E−01
$f_{c_{17}}$	2.8422E+01/1.0135E+01(<)	2.3333E−01/1.6406E−01(<)	1.4729E+01/1.1649E+01(<)	2.1754E−01/2.6978E−01(<)	**1.6671E−01/2.2737E−01**
$f_{c_{18}}$	5.7336E+02/3.6656E+03(<)	1.3001E−01/1.4268E−01(>)	1.6115E+00/7.7912E+00(<)	**9.9269E−02/1.3889E−01**(>)	2.2388E−01/1.9686E−01
$f_{c_{19}}$	2.2310E+01/1.8084E+01(<)	5.4933E−02/3.7395E−02(<)	9.2282E−01/1.9952E+00(<)	**8.9779E−03/1.0455E−02**(>)	1.0199E−02/1.3424E−02
$f_{c_{20}}$	2.2088E+01/9.7098E+00(<)	*0/0(≈)*	9.4811E+00/1.0279E+01(<)	8.0719E−09/1.7641E−09(<)	*0/0*
$f_{c_{21}}$	1.8310E+02/4.6390E+01(<)	**1.2892E+02/4.7480E+01**(>)	1.9414E+02/2.7768E+01(<)	1.5429E+02/5.1700E+01(<)	1.4490E+02/5.2290E+01
$f_{c_{22}}$	9.8571E+01/1.1530E+01(>)	**9.6085E+01/1.9605E+01**(>)	1.0000E+02/0(≈)	1.0001E+02/4.0101E−02(<)	1.0000E+02/1.4215E−13
$f_{c_{23}}$	3.0787E+02/3.8636E+00(<)	3.0384E+02/2.0214E+00(<)	3.0141E+02/1.6354E+00(<)	3.0091E+02/1.7027E+00(<)	**3.0078E+02/1.4386E+00**
$f_{c_{24}}$	3.0402E+02/8.2255E+01(<)	**2.3238E+02/1.1680E+02**(>)	3.1961E+02/4.4814E+01(<)	2.7840E+02/9.4578E+01(>)	2.8148E+02/9.0945E+01
$f_{c_{25}}$	4.3133E+02/2.2791E+01(<)	**3.9963E+02/8.9210E+00**(>)	4.4657E+02/9.8873E−01(<)	4.1133E+02/2.0987E+01(>)	4.1575E+02/2.2387E+01
$f_{c_{26}}$	3.0081E+02/1.8240E+01(<)	**2.9412E+02/4.2008E+01**(>)	2.9913E+02/1.6144E+01(>)	3.0000E+02/0(≈)	3.0000E+02/0
$f_{c_{27}}$	3.8337E+02/1.3616E+01(>)	3.8842E+02/1.2301E+00(>)	3.9718E+02/2.3956E+00(<)	**3.7955E+02/9.2742E+00**(>)	3.9378E+02/1.1343E+00
$f_{c_{28}}$	4.6461E+02/1.3806E+02(<)	**2.9968E+02/5.8396E+01**(>)	5.8623E+02/6.2471E+01(<)	3.3065E+02/4.8587E+01(<)	3.0448E+02/9.5485E+01
$f_{c_{29}}$	2.6864E+02/2.2545E+01(<)	2.3695E+02/5.0439E+00(<)	2.6374E+02/1.1297E+01(<)	**2.3121E+02/2.7910E+00**(>)	2.3162E+02/2.8417E+00
$f_{c_{30}}$	1.8822E+05/4.2336E+05(<)	3.9527E+02/3.0722E+00(>)	4.0983E+02/2.0294E+01(<)	**2.0267E+02/4.5353E+00**(>)	4.0315E+02/1.9983E+01
>/≈/<	2/1/27	10/6/14	4/2/24	12/1/17	–/–/–

Table 6
Algorithm comparison between PSO-sono, E-QUATRE, HSES, EA4eig and PCM-DE on 30D optimization of our test suite.

Non-DE algorithms No.	PSO-sono Mean/Std	E-QUATRE Mean/Std	HSES Mean/Std	EA4eig Mean/Std	PCM-DE Mean/Std
f_{c_1}	1.5758E+03/1.9114E+03(<)	1.0867E−14/6.0880E−15(<)	2.6968E−10/2.7008E−10(<)	9.0249E−09/8.3123E−10(<)	**0/0**
f_{c_2}	4.0345E+13/1.8381E+14(<)	6.7989E−14/6.8937E−14(<)	1.5171E−07/1.9033E−07(<)	2.7500E+00/1.9639E+01(<)	**0/0**
f_{c_3}	5.8438E−02/3.1588E−01(<)	5.6843E−14/1.1369E−14(<)	2.1580E−10/2.2087E−10(<)	9.5115E−09/4.5406E−10(<)	**0/0**
f_{c_4}	1.0236E+02/5.2555E+01(<)	2.9234E+01/3.0498E+01(>)	**4.3019E+00/9.8107E+00**(>)	1.8368E+01/1.4059E+01(>)	5.2522E+01/8.2526E+00
f_{c_5}	3.2326E+01/9.0347E+00(<)	3.6128E+01/7.1430E+00(<)	8.4279E+00/2.8723E+00(<)	2.5011E+01/6.6298E+00(<)	**6.7782E+00/1.4887E+00**
f_{c_6}	2.8914E−01/4.2848E−01(<)	4.7120E−11/1.0970E−10(>)	**1.1369E−13/0**(>)	1.2185E−07/3.5047E−07(>)	1.6707E−06/4.5762E−06
f_{c_7}	6.4112E+01/1.3983E+01(<)	6.7158E+01/8.5441E+00(<)	4.0901E+01/6.1020E+00(<)	5.8696E+01/5.9282E+00(<)	**3.7102E+01/1.0312E+00**
f_{c_8}	3.3633E+01/9.4872E+00(<)	3.8090E+01/7.7163E+00(<)	7.8621E+00/2.8019E+00(<)	2.7820E+01/8.5450E+00(<)	**7.4233E+00/1.7401E+00**
f_{c_9}	2.8888E+00/8.0568E+00(<)	*0/0(≈)*	*0/0(≈)*	3.5109E−03/1.7551E−02(<)	*0/0*
$f_{c_{10}}$	3.0802E+03/6.4300E+02(<)	1.9947E+03/4.0893E+02(<)	**9.7003E+02/3.1135E+02**(>)	1.8450E+03/2.8897E+02(<)	1.4202E+03/2.3505E+02
$f_{c_{11}}$	1.4136E+02/4.7455E+01(<)	1.2560E+01/1.1842E+01(<)	9.9998E+00/1.8378E+01(<)	1.0875E+01/5.6216E+00(<)	**5.4838E+00/1.1406E+01**
$f_{c_{12}}$	1.6141E+05/6.0326E+05(<)	8.1707E+02/4.8904E+02(<)	**6.7426E+01/1.2517E+02**(>)	3.7680E+02/2.2126E+02(<)	1.1594E+02/7.0382E+01
$f_{c_{13}}$	1.5922E+04/2.0479E+04(<)	2.5021E+01/1.8180E+01(<)	3.6408E+01/1.6625E+01(<)	2.3614E+01/1.7710E+01(<)	**1.2908E+01/5.8705E+00**
$f_{c_{14}}$	1.9490E+03/6.3754E+03(<)	**1.5355E+01/9.7451E+00**(>)	1.7009E+01/8.8094E+00(>)	1.5759E+01/1.0795E+01(>)	2.0450E+01/5.1838E+00
$f_{c_{15}}$	5.6144E+03/6.6936E+03(<)	9.0078E+00/3.8733E+00(<)	6.0466E+00/3.6024E+00(<)	4.9745E+00/2.7908E+00(<)	**1.1445E+00/7.7129E−01**
$f_{c_{16}}$	6.3594E+02/2.7452E+02(<)	3.7708E+02/1.6342E+02(<)	2.5703E+02/1.8323E+02(<)	2.6101E+02/1.4074E+02(<)	**6.3177E+01/6.7323E+01**
$f_{c_{17}}$	2.2415E+02/1.2619E+02(<)	3.9135E+01/9.0990E+00(<)	4.1548E+01/6.1590E+01(<)	2.7706E+01/1.2990E+01(<)	**2.6186E+01/7.0129E+00**
$f_{c_{18}}$	6.0574E+04/1.3025E+05(<)	2.6041E+01/3.5569E+00(<)	**1.9113E+01/6.3259E+00**(>)	2.1546E+01/4.5306E+00(<)	2.0573E+01/1.4791E−01
$f_{c_{19}}$	6.7466E+03/8.7875E+03(<)	1.1719E+01/2.5459E+00(<)	**4.0660E+00/1.8995E+00**(>)	5.3706E+00/1.4107E+00(<)	4.0954E+00/1.2936E+00
$f_{c_{20}}$	2.1479E+02/9.6852E+01(<)	6.2579E+01/5.7720E+01(<)	1.6421E+02/5.8428E+01(<)	3.2403E+01/3.8201E+01(<)	**3.1486E+01/6.1854E+00**
$f_{c_{21}}$	2.3744E+02/1.0700E+01(<)	2.3774E+02/7.3901E+00(<)	2.0813E+02/2.8177E+00(<)	2.2536E+02/6.6971E+00(<)	**2.0728E+02/1.6532E+00**
$f_{c_{22}}$	1.0045E+02/1.2035E+00(<)	*1.0000E+02/1.4352E−14(≈)*	*1.0000E+02/1.4352E−14(≈)*	*1.0000E+02/8.0991E−06(≈)*	*1.0000E+02/6.3901E−14*
$f_{c_{23}}$	3.8588E+02/1.0111E+01(<)	3.8304E+02/8.2400E+00(<)	3.5067E+02/7.5643E+00(<)	3.7703E+02/9.2290E+00(<)	**3.4457E+02/2.8645E+00**
$f_{c_{24}}$	4.5245E+02/1.1272E+01(<)	4.5177E+02/7.2973E+00(<)	**4.1919E+02/4.4718E+00**(>)	4.4465E+02/8.3676E+00(<)	4.2082E+02/1.6896E+00
$f_{c_{25}}$	3.9849E+02/1.2548E+01(<)	3.8682E+02/9.8589E−02(<)	3.8675E+02/2.4560E−02(<)	**3.7967E+02/2.3702E+00**(>)	3.8670E+02/9.7496E−03
$f_{c_{26}}$	1.2751E+03/3.7716E+02(<)	1.2978E+03/1.8236E+02(<)	8.9408E+02/1.2325E+02(<)	1.2079E+03/2.5055E+02(<)	**8.7413E+02/4.3716E+01**
$f_{c_{27}}$	5.3660E+02/2.7631E+01(<)	5.0721E+02/5.8780E+00(<)	5.1600E+02/5.6741E+00(<)	4.9395E+02/1.8589E+01(<)	**4.9284E+02/8.7525E+00**
$f_{c_{28}}$	3.8931E+02/5.9323E+01(<)	3.0894E+02/3.0947E+01(<)	3.2430E+02/4.4248E+01(<)	3.4037E+02/5.3390E+01(<)	**3.0873E+02/3.0250E+01**
$f_{c_{29}}$	7.7091E+02/1.4341E+02(<)	4.6216E+02/3.7766E+01(<)	4.6742E+02/5.5686E+01(<)	**4.1194E+02/4.9578E+01**(>)	4.3150E+02/7.9950E+00
$f_{c_{30}}$	1.6864E+04/1.5445E+04(<)	2.1701E+03/1.1632E+02(<)	2.0528E+03/3.8234E+01(<)	**2.7323E+02/1.6829E+01**(>)	1.9866E+03/2.0265E+01
>/≈/<	0/0/30	3/2/25	8/2/20	6/1/23	–/–/–

Table 7
Algorithm comparison between PSO-sono, E-QUATRE, HSES, EA4eig and PCM-DE on 50D optimization of our test suite.

Non-DE algorithms No.	PSO-sono Mean/Std	E-QUATRE Mean/Std	HSES Mean/Std	EA4eig Mean/Std	PCM-DE Mean/Std
f_{c_1}	2.7795E+03/3.4768E+03(<)	3.7338E−14/1.9887E−14(<)	1.8344E−09/4.0623E−09(<)	9.8152E−09/2.3919E−10(<)	**1.2604E−14/4.2679E−15**
f_{c_2}	1.3791E+36/9.2480E+36(<)	1.1699E−10/6.1009E−10(<)	1.6330E−08/1.4901E−08(<)	5.9754E−08/1.0982E−07(<)	**1.2260E−14/1.5311E−14**
f_{c_3}	2.0982E+02/3.2578E+02(<)	1.6496E−13/4.2923E−14(<)	1.3443E−09/2.8802E−09(<)	9.8809E−09/9.1158E−11(<)	**7.2447E−14/2.5620E−14**
f_{c_4}	2.2800E+02/5.3103E+01(<)	4.3511E+01/4.6421E+01(<)	5.4468E+01/5.2818E+01(<)	6.0442E+01/4.3603E+01(<)	**4.3170E+01/3.6193E+01**
f_{c_5}	6.3664E+01/1.5403E+01(<)	8.2260E+01/1.4179E+01(<)	**1.1705E+00/1.0474E+00**(>)	7.2417E+01/1.5186E+01(<)	1.3944E+01/2.0279E+00
f_{c_6}	5.1183E−01/1.4133E+00(<)	**2.0285E−13/4.7224E−14**(>)	4.0728E−07/7.1243E−08(>)	2.5008E−06/4.1812E−06(>)	1.7639E−03/1.3880E−03
f_{c_7}	1.2032E+02/2.0509E+01(<)	1.2410E+02/1.3557E+01(<)	6.5148E+01/7.2259E−01(<)	1.2089E+02/9.3407E+00(<)	**6.2981E+01/1.6221E+00**
f_{c_8}	6.6020E+01/1.4720E+01(<)	8.3561E+01/1.3336E+01(<)	**1.6583E+00/1.2374E+00**(>)	6.8008E+01/1.4322E+01(<)	1.3739E+01/2.0600E+00
f_{c_9}	1.0334E+01/1.1035E+01(<)	6.3983E−02/1.7731E−01(<)	*0/0(≈)*	3.3396E−01/4.6692E−01(<)	*0/0*
$f_{c_{10}}$	5.9742E+03/8.9391E+02(<)	3.9573E+03/6.5769E+02(<)	**3.7447E+02/3.1645E+02**(>)	3.7967E+03/4.8984E+02(<)	3.0814E+03/2.7442E+02
$f_{c_{11}}$	2.4999E+02/6.5333E+01(<)	5.4777E+01/1.4297E+01(<)	2.4963E+01/2.4071E+00(<)	5.8310E+01/1.6760E+01(<)	**2.2328E+01/1.8919E+00**
$f_{c_{12}}$	1.8894E+06/5.6260E+06(<)	2.1826E+04/2.0600E+04(<)	**1.1876E+02/1.2886E+02**(>)	3.3423E+04/5.8807E+04(<)	1.6271E+03/4.7303E+02
$f_{c_{13}}$	4.5487E+03/4.6911E+03(<)	1.8778E+02/1.6112E+02(<)	4.5119E+01/3.1742E+01(<)	9.8452E+01/3.7358E+01(<)	**2.1928E+01/1.5427E+01**
$f_{c_{14}}$	2.9340E+04/5.3191E+04(<)	5.1434E+01/1.0881E+01(<)	2.7600E+01/7.2174E+00(<)	5.3667E+01/1.1247E+01(<)	**2.4748E+01/1.8219E+00**
$f_{c_{15}}$	3.6908E+03/3.2790E+03(<)	4.7484E+01/1.0893E+01(<)	**1.9600E+01/3.9950E+00**(>)	4.9206E+01/1.8719E+01(<)	2.2053E+01/2.4843E+00
$f_{c_{16}}$	1.3775E+03/3.6473E+02(<)	8.1635E+02/2.3891E+02(<)	5.1862E+02/1.8154E+02(<)	7.4657E+02/1.7412E+02(<)	**3.1555E+02/1.1429E+02**
$f_{c_{17}}$	1.1184E+03/2.8239E+02(<)	4.9635E+02/1.4422E+02(<)	2.7398E+02/1.4031E+02(<)	4.5612E+02/1.3678E+02(<)	**2.4060E+02/4.8050E+01**
$f_{c_{18}}$	6.4114E+05/1.0881E+06(<)	1.1442E+02/1.0565E+02(<)	2.1929E+01/1.5951E−01(<)	4.7030E+01/1.1258E+01(<)	**2.1157E+01/8.7311E−01**
$f_{c_{19}}$	1.5369E+04/1.0469E+04(<)	2.0847E+01/4.4584E+00(<)	1.1325E+01/1.0163E+01(<)	2.6899E+01/4.6559E+00(<)	**1.1106E+01/2.1103E+00**
$f_{c_{20}}$	6.5418E+02/2.3121E+02(<)	3.0861E+02/1.4236E+02(<)	**4.3114E+01/5.4283E+01**(>)	1.8228E+02/1.1392E+02(<)	1.1661E+02/3.2452E+01
$f_{c_{21}}$	2.7369E+02/1.4861E+01(<)	2.8310E+02/1.4911E+01(<)	2.1731E+02/1.1838E+00(<)	2.7114E+02/1.4883E+01(<)	**2.1473E+02/2.1676E+00**
$f_{c_{22}}$	5.3900E+03/2.3066E+03(<)	2.7848E+03/2.3094E+03(<)	**1.0000E+02/3.8849E−09**(>)	3.2736E+03/2.0159E+03(<)	2.9593E+02/7.9543E+02
$f_{c_{23}}$	5.1712E+02/2.3779E+01(<)	5.0782E+02/1.4048E+01(<)	4.2475E+02/8.3800E+00(<)	5.0179E+02/1.6629E+01(<)	**4.2024E+02/7.7018E+00**
$f_{c_{24}}$	5.7142E+02/2.4573E+01(<)	5.6700E+02/1.3977E+01(<)	4.9913E+02/2.5690E+00(<)	5.6377E+02/1.5682E+01(<)	**4.9901E+02/5.2595E+00**
$f_{c_{25}}$	6.4803E+02/3.3328E+01(<)	5.2426E+02/2.8178E+01(<)	5.4781E+02/2.4122E+01(<)	**4.5946E+02/2.0193E+01**(>)	4.8097E+02/5.4072E+00
$f_{c_{26}}$	1.8216E+03/4.0653E+02(<)	1.9458E+03/1.4631E+02(<)	**5.8505E+02/1.4274E+02**(>)	1.8322E+03/2.0942E+02(<)	1.0371E+03/7.7311E+01
$f_{c_{27}}$	8.9218E+02/1.2642E+02(<)	5.3159E+02/1.4916E+01(<)	5.8831E+02/1.8040E+01(<)	**4.9884E+02/8.3540E+00**(>)	5.0851E+02/6.5458E+00
$f_{c_{28}}$	6.1682E+02/4.9693E+01(<)	4.8299E+02/2.3320E+01(<)	5.0428E+02/9.7321E+00(<)	**4.5452E+02/1.7839E+01**(>)	4.5885E+02/2.2278E−13
$f_{c_{29}}$	1.0573E+03/3.0111E+02(<)	4.6162E+02/9.9445E+01(<)	4.2065E+02/1.2706E+02(<)	4.0460E+02/1.0270E+02(<)	**3.5152E+02/1.0478E+01**
$f_{c_{30}}$	4.0126E+06/6.0698E+06(<)	5.9851E+05/2.3237E+04(<)	6.0237E+05/2.0419E+04(<)	**1.1640E+03/4.3426E+02**(>)	5.9243E+05/2.1917E+04
>/≈/<	0/0/30	1/0/29	9/1/20	5/0/25	−/−/−

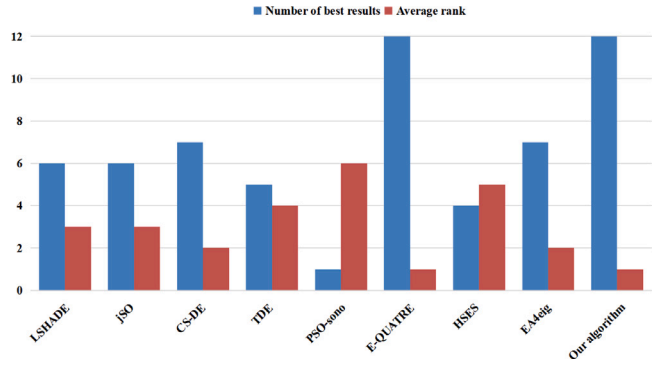


Fig. 3. Statistical results of PCM-DE and 8 advanced algorithms under CEC2017 test suite on 10D.

Table 8
Average ranking results based on Friedman test.

Algorithms	Mean rank			Total mean rank
	D = 10	D = 30	D = 50	
LSHADE	4.3	4	4.3	4.2
jSO	3.8	3.9	3.9	3.9
CS-DE	3.9	4.8	5.1	4.6
TDE	4.8	3.2	4	4
PSO-sono	8.2	8.9	8.7	8.6
E-QUATRE	3.7	6.5	6.9	5.7
HSES	6.5	4.5	3.6	4.9
EA4eig	4.1	5.2	6.4	5.2
PCM-DE	2.6	2.1	1.7	2.1

To better demonstrate the performance comparisons between all algorithms, the Friedman test is employed. The average ranks of the nine algorithms are presented in Table 8. From the table, it is evident that PCM-DE has significantly better average ranks than the eight comparing algorithms on 10D, 30D, and 50D. The average ranks of LSHADE, jSO, and PSO-sono are 4.2, 3.9, and 8.6, respectively. Moreover, it can be seen that the three algorithms fail to mitigate the curse of dimensionality. The average ranks of CS-DE, E-QUATRE, and EA4eig are 4.6, 5.7, and 5.2, respectively, and with the increase in dimension, the solution accuracy obtained by the above three algorithms decreases significantly. Although EA4eig is the champion algorithm of the CEC2022 test suite, which involves test problems with dimensions of 5D, 10D, and 20D, it is observed that PCM-DE is still better than EA4eig even on low dimensions. Only HSES and PCM-DE have better optimization performance as dimensionality increases. Therefore, we can conclude that PCM-DE not only performs well on low dimensions, but also obtains better competitive performances on high-dimensional search space.

In addition, to better illustrate the above statistical results, Figs. 3 to 5 provide bar charts of the statistical results between PCM-DE and other algorithms under CEC2017 test suite. It can be seen that PCM-DE obtains the best average ranking and the optimal results for CEC2017 test suite across all dimensions.

The summary of comparison results under CEC2013, CEC2014 and CEC2017 and CEC2022 across different dimension is presented in Table 9. In terms of comparison with DE variants, PCM-DE secures 176 performance enhancements out of 288 cases comparing with LSHADE, secures 192 performance enhancements out of 288 cases comparing with jSO, 177 out of 288 cases comparing with CS-DE and 171 out of 288 cases comparing with TDE. In terms of comparison with non-DE based algorithms, PCM-DE secures 268 performance improvements out of 288 cases comparing with PSO-sono, 205 out of 288 cases comparing with E-QUATRE, 180 out of 288 cases comparing with HSES and 200 out of 288 cases comparing with EA4eig.

To conclude, the reasons why PCM-DE is superior to the above eight comparison algorithms may be lie in that the archived population perturbation strategy and the population diversity mechanism.

Table 9
The comparison results of PCM-DE and other algorithms for 10D, 30D, 50D under CEC2013, CEC2014, CEC2017 test suites and 10D and 20D under CEC2022 test suites are summarized.

Test suit:	CEC2013			CEC2014			CEC2017			CEC2022		All
>/≈/<	D = 10	D = 30	D = 50	D = 10	D = 30	D = 50	D = 10	D = 30	D = 50	D = 10	D = 20	∑
LSHADE	6/10/12	5/4/19	8/2/18	9/6/15	10/4/16	10/1/19	4/7/19	6/2/22	5/0/25	2/5/5	2/4/6	67/45/176
jSO	9/7/12	2/4/22	7/2/19	8/7/15	4/4/22	5/2/23	5/8/17	6/3/21	1/0/29	1/5/6	2/4/6	50/46/192
CS-DE	13/8/7	5/5/18	12/3/13	13/5/12	6/4/20	8/2/20	6/7/17	1/4/25	1/2/27	1/3/8	1/3/8	67/44/177
TDE	11/10/7	5/5/18	11/2/15	7/7/16	9/4/17	10/1/19	3/6/21	6/5/19	3/1/26	2/2/8	4/3/5	71/46/171
PSO-sono	2/2/24	1/1/26	0/0/28	3/0/27	2/0/28	2/0/28	2/1/27	0/0/30	0/0/30	2/1/9	1/0/11	15/5/268
E-QUATRE	9/6/13	4/4/20	7/2/19	6/5/19	2/1/27	5/1/24	10/6/14	3/2/25	1/0/29	2/4/6	0/3/9	49/34/205
HSES	8/6/14	5/2/21	12/3/13	9/1/20	7/1/22	13/1/16	4/2/24	7/2/21	15/1/14	2/3/7	2/2/8	84/24/180
EA4eig	2/4/22	5/3/20	5/2/21	16/1/13	5/0/25	8/0/22	12/1/17	6/1/23	5/0/25	3/5/4	4/0/8	71/17/200

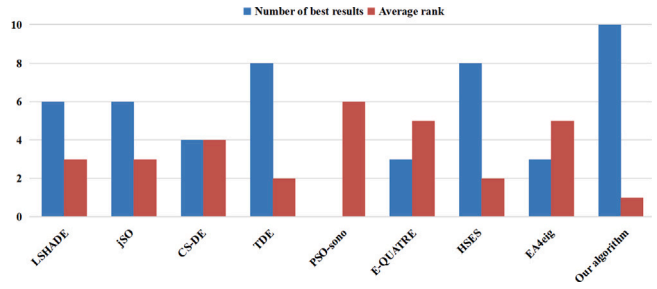


Fig. 4. Statistical results of PCM-DE and 8 advanced algorithms under CEC2017 test suite on 30D.

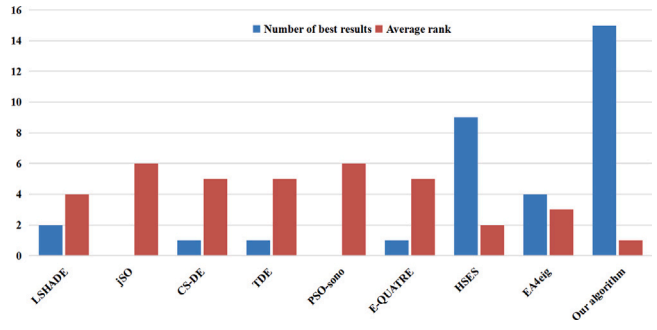


Fig. 5. Statistical results of PCM-DE and 8 advanced algorithms under CEC2017 test suite on 50D.

These strategies can avoid premature convergence and stagnation when dealing with composition functions problems. In addition, the two-phase parameter control strategy can balance the exploration and exploitation capacities when dealing with unimodal functions problems. Therefore, PCM-DE is a promising optimization approach for complex optimization problems with high dimensions.

4.4. Convergence analysis

To demonstrate the convergence performance of PCM-DE, Figs. 6 and 7 present the convergence curves of PCM-DE and eight other algorithms under the CEC2017 test suite on 30D optimization, with the median of 51 runs. As shown in Figs. 6–7, PCM-DE generally exhibits a faster convergence speed than most of comparing algorithms. In comparison with LSHADE, PCM-DE achieves superior convergence speed in f_{c_1}–f_{c_4}, f_{c_6}–f_{c_7}, and f_{c_9}–$f_{c_{30}}$; in comparison with jSO, PCM-DE performs better in f_{c_1}–$f_{c_{15}}$ and $f_{c_{17}}$–$f_{c_{30}}$; with CS-DE, PCM-DE performs better in f_{c_1}–$f_{c_{30}}$; with TDE, PCM-DE shows better convergence speed in f_{c_1}–$f_{c_{30}}$; with PSO-sono, PCM-DE is superior in f_{c_1}–$f_{c_{30}}$; with E-QUATRE, PCM-DE outperforms E-QUATRE in f_{c_1}–$f_{c_{12}}$ and $f_{c_{14}}$–$f_{c_{30}}$; with HSES, PCM-DE has better convergence speed in f_{c_1}–$f_{c_{30}}$; and with EA4eig, PCM-DE performs better in f_{c_1}–$f_{c_{30}}$. The improved convergence performance of PCM-DE may be attributed to the perturbation mechanism for the archived population and the population diversity mechanism. These methods enhance the diversity of the population and accelerate the algorithm's convergence speed in the later stages of evolution, allowing it to escape from local optima. As a result, PCM-DE exhibits higher convergence efficiency.

4.5. Ablation experiments

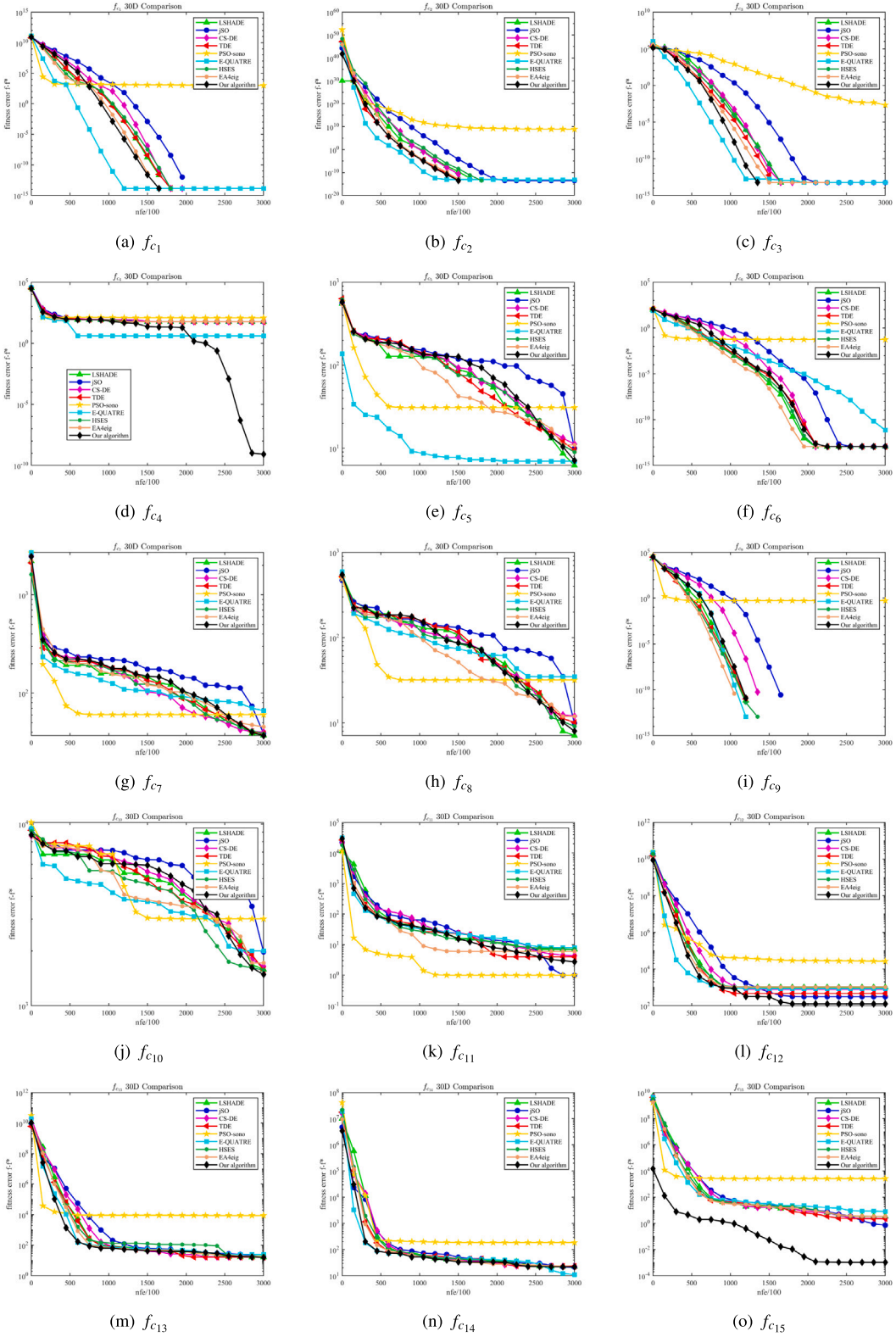
This paper proposes three main modifications including the two-phase parameter adaptation strategy, the perturbation mechanism for archived population, and the population diversity mechanism. In order to validate the effectiveness of three strategies, three variants based on PCM-DE are created: V_1 represents the PCM-DE variant without the two-phase parameter adaptation strategy; V_2 represents the PCM-DE variant without the perturbation mechanism for archived population; V_3 represents the PCM-DE variant without the population diversity enhancement mechanism. Table 10 shows comparison results between PCM-DE and three variants under CEC2017 test suite with 30 dimensions. From the table, one can observe that PCM-DE obtains the best optimization accuracy on 15 functions, V_1 obtains the best optimization accuracy 7 functions, V_2 on 6 functions, and V_3 on 7 functions. For unimodal functions, PCM-DE performs significantly better than other variants. For simple multimodal functions f_{c_3}–f_{c_9}, PCM-DE has the best performance on half of the functions, indicating that the exploration ability of PCM-DE has been greatly enhanced by these modifications. For hybrid functions $f_{c_{10}}$–$f_{c_{19}}$, PCM-DE still obtains better performance on half of the functions, which proves that PCM-DE has achieved a balance between exploration and exploitation. Other variants have also shown improved performance on several functions, demonstrating their effectiveness. In conclusion, the combination of these strategies can significantly enhance the optimization ability of DE.

4.6. Time complexity analysis

For complexity analysis, the method in IEEE CEC2013 is employed to evaluate the time complexity of PCM-DE by comparing with LSHADE, jSO, CS-DE, TDE, PSO-sono, E-QUATRE, HSES, EA4eig. The time complexity is determined using $(\hat{T}_2 - T_1)/T_0$, and the meanings of $\hat{T}_2$, T_1 and T_0 is defined in [47]. The results of time complexity among all algorithms are listed in Table 11. From Table 11, we can observe that PCM-DE has less time complexity than CS-DE, TDE, PSO-sono, E-QUATRE, HSES, EA4eig, which indicates that PCM-DE not only improves the performance but also does not increase much time complexity.

5. Real-world applications

Truss optimization problem is a complicated optimization problem with non-linear constraints and non-convex region [55]. In this paper, three truss structures with 3, 10 and 37 bars are employed to access the performance of PCM-DE. For performance comparison, we do not employ the aforementioned meta-heuristic algorithms because they are not specially tuned for constrained optimization problems. For the sake



(a) f_{c_1} (b) f_{c_2} (c) f_{c_3}

(d) f_{c_4} (e) f_{c_5} (f) f_{c_6}

(g) f_{c_7} (h) f_{c_8} (i) f_{c_9}

(j) $f_{c_{10}}$ (k) $f_{c_{11}}$ (l) $f_{c_{12}}$

(m) $f_{c_{13}}$ (n) $f_{c_{14}}$ (o) $f_{c_{15}}$

Fig. 6. Here we present the convergence comparison by employing the median value of 51 runs on 30D optimization under f_{c_1}–$f_{c_{15}}$ of CEC2017.

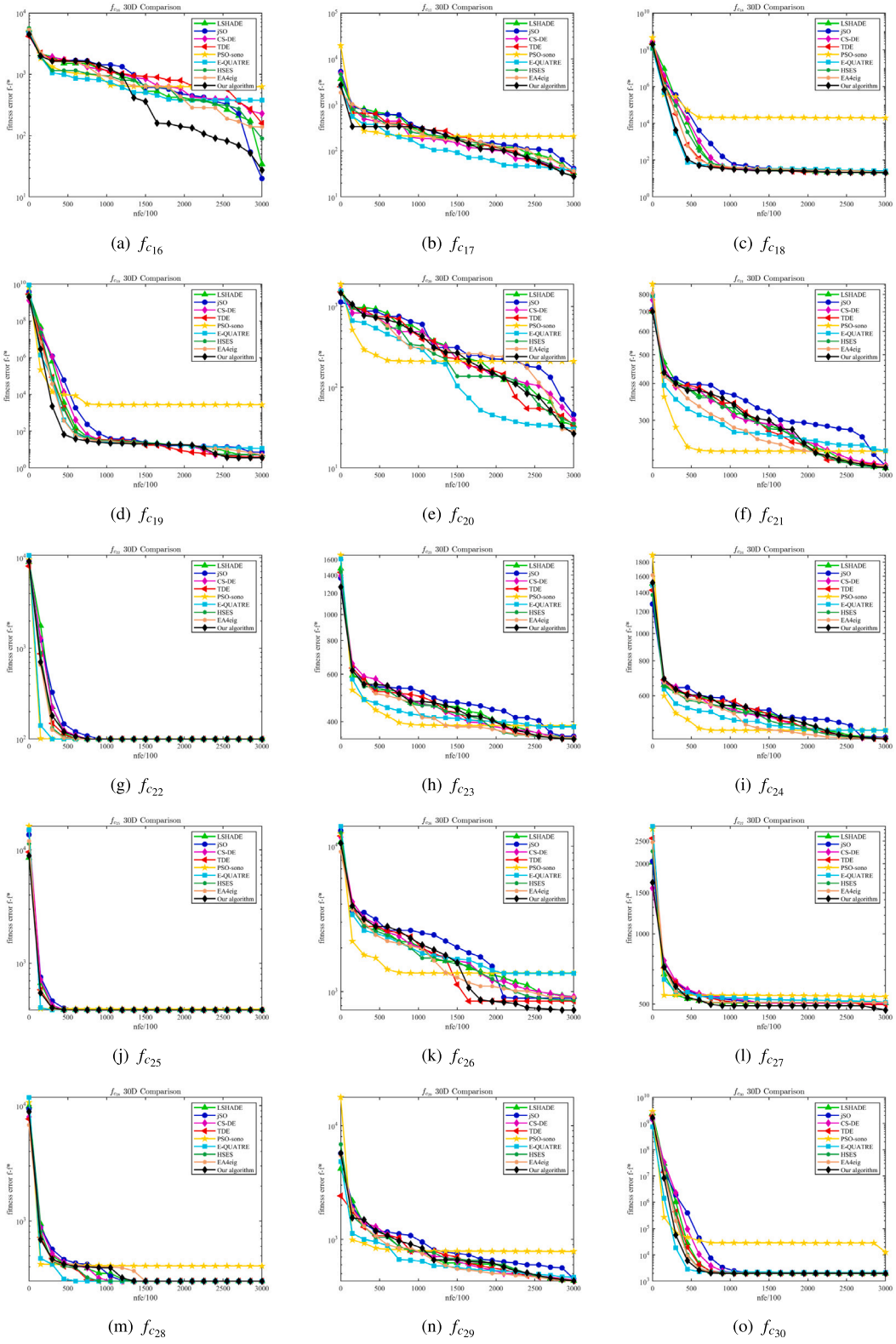


Fig. 7. Here we present the convergence comparison by employing the median value of 51 runs on 30D optimization under $f_{c_{16}}$–$f_{c_{30}}$ of CEC2017.

of fairness, we select three recently proposed algorithms, including EnMODE [56], BP-MAg-ES [57], COLSHADE [58], which are designed for constrained optimization problems, to conduct comparison with PCM-DE. The parameter setting of each algorithm is in accordance with its original paper. To ensure fairness, all algorithms are run 30 times independently and the best result ("Min"), the worst result ("Max"), the mean ("Mean") and standard deviation ("Std") among 30 runs are used for comparison.

The layout of 3-bar truss is presented in Fig. 8. The objective of the 3-bar truss optimization is to minimize the weight of bars, which can be formulated as constrained optimization problem with two design variables and three constraints. The mathematical expression of the problem is described as follows:

Table 10
Performance comparison between V_1, V_2, V_3 and PCM-DE under CEC2017 with 30D.

DE variants No.	V_1 Mean/Std	V_2 Mean/Std	V_3 Mean/Std	PCM-DE Mean/Std
f_{c_1}	*0/0(≈)*	*0/0(≈)*	*0/0(≈)*	*0/0*
f_{c_2}	2.7864E−15/8.5358E−15(<)	5.0156E−15/1.0943E−14(<)	*0/0(≈)*	*0/0*
f_{c_3}	1.1146E−15/7.9597E−15(<)	*0/0(≈)*	*0/0(≈)*	*0/0*
f_{c_4}	5.6374E+01/1.1529E+01(<)	5.8888E+01/1.3203E+00(<)	**4.9177E+01/2.2878E+01**(>)	5.2522E+01/8.2526E+00
f_{c_5}	6.8636E+00/1.5668E+00(<)	6.8151E+00/1.3550E+00(<)	7.4650E+00/1.5622E+00(<)	**6.7782E+00/1.4887E+00**
f_{c_6}	3.4422E−06/1.9531E−05(<)	1.7265E−06/4.0148E−06(<)	**5.3677E−09/2.6833E−08**(>)	1.6707E−06/4.5762E−06
f_{c_7}	3.7316E+01/1.5553E+00(<)	3.7185E+01/1.4132E+00(<)	**3.7089E+01/1.2740E+00**(>)	3.7102E+01/1.0312E+00
f_{c_8}	7.8260E+00/1.5296E+00(<)	7.4713E+00/1.5996E+00(<)	7.9382E+00/1.5652E+00(<)	**7.4233E+00/1.7401E+00**
f_{c_9}	*0/0(≈)*	*0/0(≈)*	*0/0(≈)*	*0/0*
$f_{c_{10}}$	1.4473E+03/1.9489E+02(<)	1.4468E+03/2.7064E+02(<)	1.5180E+03/2.3795E+02(<)	**1.4202E+03/2.3505E+02**
$f_{c_{11}}$	**3.9219E+00/8.5301E+00**(>)	5.0341E+00/1.1908E+01(>)	1.7052E+01/2.2800E+01(<)	5.4838E+00/1.1406E+01
$f_{c_{12}}$	1.2454E+02/6.7157E+01(<)	**1.0935E+02/6.0531E+01**(>)	1.0348E+03/3.5631E+02(<)	1.1594E+02/7.0382E+01
$f_{c_{13}}$	1.3750E+01/5.6133E+00(<)	1.3692E+01/5.5512E+00(<)	1.4572E+01/6.4929E+00(<)	**1.2908E+01/5.8705E+00**
$f_{c_{14}}$	2.0772E+01/4.6113E+00(<)	2.0500E+01/5.3475E+00(<)	2.1544E+01/4.0077E+00(<)	**2.0450E+01/5.1838E+00**
$f_{c_{15}}$	9.7837E−01/8.5050E−01(>)	**7.8875E−01/5.6957E−01**(>)	2.8133E+00/1.1922E+00(<)	1.1445E+00/7.7129E−01
$f_{c_{16}}$	**5.8108E+01/7.1729E+01**(>)	7.2394E+01/7.8434E+01(<)	1.4020E+02/9.2777E+01(<)	6.3177E+01/6.7323E+01
$f_{c_{17}}$	2.8279E+01/6.5870E+00(<)	2.8428E+01/6.0985E+00(<)	2.8764E+01/7.0955E+00(<)	**2.6186E+01/7.0129E+00**
$f_{c_{18}}$	2.0835E+01/5.0627E−02(<)	2.0596E+01/5.0021E−02(<)	2.1839E+01/1.1988E+00(<)	**2.0573E+01/1.4791E−01**
$f_{c_{19}}$	4.4627E+00/1.2256E+00(<)	4.4097E+00/1.6746E+00(<)	4.9646E+00/1.3852E+00(<)	**4.0954E+00/1.2936E+00**
$f_{c_{20}}$	**2.8299E+01/8.7816E+00**(>)	2.9623E+01/6.3573E+00(>)	3.8592E+01/7.2179E+00(<)	3.1486E+01/6.1854E+00
$f_{c_{21}}$	**2.0683E+02/1.7799E+00**(>)	2.0701E+02/1.4404E+00(>)	2.0709E+02/1.4543E+00(>)	2.0728E+02/1.6532E+00
$f_{c_{22}}$	*1.0000E+02/1.4352E−14(≈)*	*1.0000E+02/1.4352E−14(≈)*	*1.0000E+02/6.3901E−14(≈)*	*1.0000E+02/6.3901E−14*
$f_{c_{23}}$	**3.4359E+02/3.5680E+00**(>)	3.4400E+02/2.6033E+00(>)	3.4396E+02/2.9016E+00(>)	3.4457E+02/2.8645E+00
$f_{c_{24}}$	4.2095E+02/2.1491E+00(<)	4.2162E+02/2.2791E+00(<)	4.2176E+02/2.3838E+00(<)	**4.2082E+02/1.6896E+00**
$f_{c_{25}}$	*3.8670E+02/8.4907E−03(≈)*	*3.8670E+02/9.8527E−03(≈)*	3.8677E+02/2.8792E−02(<)	*3.8670E+02/9.7496E−03*
$f_{c_{26}}$	8.7851E+02/4.6338E+01(<)	8.8594E+02/3.7832E+01(<)	9.0404E+02/5.1043E+01(<)	**8.7413E+02/4.3716E+01**
$f_{c_{27}}$	4.9389E+02/8.1717E+00(<)	4.9318E+02/9.2562E+00(<)	5.0273E+02/7.7968E+00(<)	**4.9284E+02/8.7525E+00**
$f_{c_{28}}$	3.1096E+02/3.3615E+01(<)	*3.0873E+02/3.0250E+01(≈)*	3.2634E+02/4.8427E+01(<)	*3.0873E+02/3.0250E+01*
$f_{c_{29}}$	4.3263E+02/6.9542E+00(<)	**4.2968E+02/7.3006E+00**(>)	4.3252E+02/5.7667E+00(<)	4.3150E+02/7.9950E+00
$f_{c_{30}}$	1.9847E+03/2.0116E+01(>)	1.9904E+03/2.3454E+01(<)	1.9587E+03/3.6598E+01	1.9866E+03/2.0265E+01
>/≈/<	7/4/19	6/6/18	5/5/19	–/–/–

Table 11
Time complexity of PCM-DE and eight comparing algorithms.

Algorithms	T_0	T_1	$\hat{T}_2$	$\frac{\hat{T}_2-T_1}{T_0}$
LSHADE			1.4834	14.9520
jSO			1.5077	15.3688
CS-DE			1.7714	19.8919
TDE			1.7714	19.8919
PSO-sono	0.0583	0.6117	3.1893	44.2126
E-QUATRE			2.2580	23.2384
HSES			1.9719	19.8919
EA4eig			10.9493	177.3173
PCM-DE			1.6741	18.2220

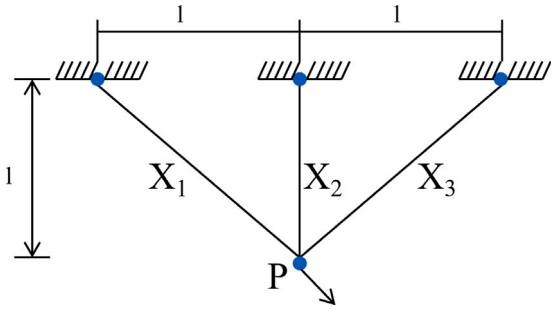


Fig. 8. Layout of the three-bar truss.

Minimize:

$$f(x) = (2\sqrt{2}x_1 + x_2) \cdot l \tag{21}$$

Subject to:

$$\begin{cases} g_1(x) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2} \cdot P - \sigma \le 0 \\ g_2(x) = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2} \cdot P - \sigma \le 0 \\ g_3(x) = \frac{1}{\sqrt{2}x_2 + x_1} \cdot P - \sigma \le 0 \end{cases} \tag{22}$$

With bounds:

$$0 \le x_i \le 1, i = 1, 2. l = 100 \text{ cm}, P = 2 \text{ kN/cm}^2, \sigma = 2 \text{ kN/cm}^2. \tag{23}$$

To verify the scalability of PCM-DE, it is extended to the 10-bar and 37-bar truss optimization problems with frequency constraints. The optimization objective is to search for the optimal design of size and shape of the structure, where total weight is minimized. In the optimization process, the calculation considers truss member sizes and/or structural shape elements as continuous variables. The design constraints are limits on natural frequencies. The objective function can be expressed as follows:

Minimize: Subject to:

$$f(\mathbf{A}, \mathbf{x}) = \sum_{i=1}^{m} \rho_i A_i L_i(x_j) \tag{24}$$

Subject to: Subject to:

$$\begin{cases} \omega_k \le \overline{\omega_k} \\ \omega_l \le \overline{\omega_l} \\ A_{i,\min} \le A_i \le A_{i,\max} \\ x_{j,\min} \le x_j \le x_{j,\max} \end{cases} \tag{25}$$

where $\mathbf{A} = \{A_1, \ldots, A_i, \ldots, A_m\}$ and $\mathbf{x} = \{x_1, \ldots, x_i, \ldots, x_n\}$ denote vector of m cross-sectional areas and that of n element nodal coordinates, respectively. rho_i and L_i denote the material density and the length of the ith member of the structure, respectively. $\overline{\omega_k}$ and $\overline{\omega_l}$ are the upper and lower constraints of natural frequency, respectively. The data of problems are presented in Table

The 10-bar truss structure, which is illustrated in Fig. 9, belongs to group of size optimization problems. It contains 10 cross-sectional

Table 12
Statistical results obtained by all algorithms on 3-bar, 10-bar and 37-bar truss optimization problems.

Model	Algorithm	Max	Min	Mean	Std	>/≈/<
3-bar truss	PCM-DE	**6.1797189E+03**	**6.0597E+03**	6.1297162E+03	3.3431E+01	–
	EnMODE	6.2597142E+03	6.0597138E+03	**6.1197141E+03**	**2.4575E−13**	(>)
	BPMAgES	6.1905916E+03	6.0597426E+03	6.1671363E+03	1.3431E+01	(<)
	COLSHADE	6.3905262E+03	6.0597143E+03	6.2621792E+03	8.3590E+01	(<)
10-bar truss	PCM-DE	*5.2445076E+02*	*5.2445076E+02*	*5.2445076E+02*	0	–
	EnMODE	5.2445079E+02	5.2445078E+02	5.2445078E+02	2.5685E−07	(<)
	BPMAgES	5.3057057E+02	*5.2445076E+02*	5.2765487E+02	1.2225E+00	(<)
	COLSHADE	*5.2445076E+02*	*5.2445076E+02*	*5.2445076E+02*	*0*	(<)
37-bar truss	PCM-DE	3.5988320E+02	3.5980720E+02	**3.5984856E+02**	**2.6987E−02**	–
	EnMODE	3.5989541E+02	3.5982652E+02	3.5986482E+02	4.5698E−01	(<)
	BPMAgES	**3.5987651E+02**	3.5989541E+02	3.5989541E+02	8.6547E−01	(<)
	COLSHADE	3.5987985E+02	**3.5980566E+02**	3.5984647E+02	7.5612E−02	(<)

areas of the elements as decision variables (A_i with $i \in [1, 10]$). Different from the 10-bar truss, the 37-bar truss are size and shape optimization problem, as shown in Fig. 10. Therefore, both cross-sectional areas and nodal coordinates are taken into consideration. It contains 14 elements and 5 nodal coordinates as decision variables (A_i with $i \in [1, 4]$ and x_j with $i \in [3, 5, 7, 9, 11]$).

Experiment results of 3-bar, 10-bar and 37-bar truss optimization problems are presented in Table 12. For optimization of 3-bar truss structure, we can observe that PCM-DE obtains the best performance in terms of Max and Min, and the solution accuracy is significantly higher than those of other algorithms. EnMODE obtains the best results in terms of Mean and Std compared to other algorithms. For optimization of 10-bar truss structure, PCM-DE and COLSHADE secures the best performance in terms of Max and Mean. For optimization of 37-bar truss structure, PCM-DE obtains the best performance in terms of Mean and Std, demonstrating its scalability. In addition, COLSHADE ranks the second in terms of Min, closely followed by PCM-DE.

Based on the above analyses, PCM-DE, owing to its powerful optimization capacity and robust scalability, can be used to tackle complicated real-world optimization problems.

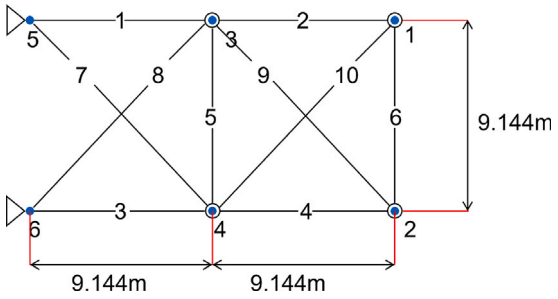


Fig. 9. Layout of the ten-bar truss.

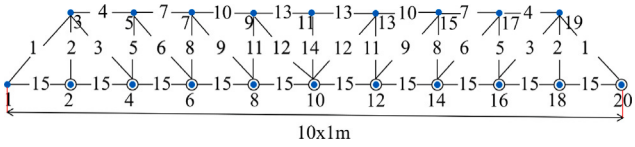


Fig. 10. Layout of the ten-bar truss.

	10-bar truss	37-bar truss
Design variables	$A_i, i = 1, 2, \ldots, 10$	$A_i, i = 1, 2, \ldots, 14$ $x_j, j = 3, 5, 7, 9, 11$
Frequency constraints (Hz)	$f_1 \geq 7$ $f_2 \geq 15$ $f_3 \geq 20$	$f_1 \geq 20$ $f_2 \geq 40$ $f_3 \geq 60$
Size variable range (cm^2)	$0.645 \leq A_i \leq 50$	$1 \leq A_i \leq 10$
Shape variable range (m)	–	$0.1 \leq x_j \leq 3$
Material density (kg/m^3)	2770	7800
Modulus of elasticity (N/m^2)	6.98×10^{10}	2.1×10^{11}
Added mass (kg)	454	10

6. Conclusion

In this paper, differential evolution with perturbation mechanism and covariance matrix based stagnation indicator(PCM-DE) is proposed to mitigate the problem of premature convergence and population stagnation resulted from the reduced population diversity especially during the later stage of evolution. Three strategies are proposed in PCM-DE: Firstly, a two-phase parameter adaptation strategy is proposed to attain a sound balance between global and local search capabilities. Secondly, to make full use of archived population, a perturbation strategy based on exponential distribution is employed to renew a certain number of individuals from the external archive. Lastly, a population diversity mechanism is proposed, where covariance matrix is used to measure the population diversity and stagnation individuals will be perturbed by variance information. Ablation experiments have been conducted with the proposed modifications. For performance verification, PCM-DE is compared with four advanced DE variants and four non-DE based algorithms under a large test suite containing 100 benchmark functions. In addition, the algorithm is applied to three different truss structure optimization problems to verify its feasibility and scalability. The experiment results confirm that PCM-DE is able to achieve highly competitive performance both on the standard test suite and real-world applications.

While the strategies proposed in this paper have made significant progress compared to existing methods, they still face challenges in accurately measuring the search characteristics of individuals and the evolutionary state of populations. In addition, the time complexity slightly increases due to the calculation of covariance matrix along with evolution. Therefore, future researches will focus on developing more efficient methods for evaluation of search behavior and evolutionary states, as well as intervention mechanisms for perturbation of stagnation individual.

CRediT authorship contribution statement

Zhenghao Song: Methodology, Software, Writing – original draft. **Chongle Ren:** Writing – original draft, Reviewing & editing. **Zhenyu Meng:** Conceptualization, Methodology, Software, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.swevo.2023.101447.

References

[1] Rainer Storn, Kenneth Price, Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (4) (1997) 341.

[2] Zhenyu Meng, Junyuan Zhang, QUATRE-EMS: QUATRE algorithm with novel adaptation of evolution matrix and selection operation for numerical optimization, Inform. Sci. 651 (2023) 119714.

[3] Zhenyu Meng, Yuxin Chen, Differential Evolution with exponential crossover can be also competitive on numerical optimization, Appl. Soft Comput. 146 (2023) 110750.

[4] T. Rogalsky, S. Kocabiyik, R.W. Derksen, Differential evolution in aerodynamic optimization, Can. Aeronaut. Space J. 46 (4) (2000) 183–190.

[5] Tapas Karmaker, Ranjan Das, Estimation of riverbank soil erodibility parameters using genetic algorithm, Sādhanā 42 (2017) 1953–1963.

[6] Ranjan Das, Bahriye Akay, Rohit K. Singla, Kuljeet Singh, Application of artificial bee colony algorithm for inverse modelling of a solar collector, Inverse Probl. Sci. Eng. 25 (6) (2017) 887–908.

[7] Mahamed G.H. Omran, Andries P. Engelbrecht, Ayed Salman, Differential evolution methods for unsupervised image classification, in: 2005 IEEE Congress on Evolutionary Computation, Vol. 2, IEEE, 2005, pp. 966–973.

[8] Dominique Douguet, E-LEA3D: a computational-aided drug design web server, Nucl. Acids Res. 38 (suppl_2) (2010) W615–W621.

[9] Xin Yao, Evolving artificial neural networks, Proc. IEEE 87 (9) (1999) 1423–1447.

[10] Maoguo Gong, Jia Liu, Hao Li, Qing Cai, Linzhi Su, A multiobjective sparse feature learning model for deep neural networks, IEEE Trans. Neural Netw. Learn. Syst. 26 (12) (2015) 3263–3277.

[11] Jia Liu, Maoguo Gong, Qiguang Miao, Xiaogang Wang, Hao Li, Structure learning for deep neural networks based on multiobjective optimization, IEEE Trans. Neural Netw. Learn. Syst. 29 (6) (2017) 2450–2463.

[12] Ammar Mansoor Kamoona, Jagdish Chandra Patra, A novel enhanced cuckoo search algorithm for contrast enhancement of gray scale images, Appl. Soft Comput. 85 (2019) 105749.

[13] Mohamad Faiz Ahmad, Nor Ashidi Mat Isa, Wei Hong Lim, Koon Meng Ang, Differential evolution: A recent review based on state-of-the-art works, Alexandria Eng. J. 61 (5) (2022) 3831–3872.

[14] Xuewen Xia, Haojie Song, Yinglong Zhang, Ling Gui, Xing Xu, Kangshun Li, Yuanxiang Li, A particle swarm optimization with adaptive learning weights tuned by a multiple-input multiple-output fuzzy logic controller, IEEE Trans. Fuzzy Syst. (2022).

[15] Huixian Qiu, Xuewen Xia, Yuanxiang Li, Xianli Deng, A dynamic multipopulation genetic algorithm for multiobjective workflow scheduling based on the longest common sequence, Swarm Evol. Comput. 78 (2023) 101291.

[16] Amer Draa, Samira Bouzoubia, Imene Boukhalfa, A sinusoidal differential evolution algorithm for numerical optimisation, Appl. Soft Comput. 27 (2015) 99–126.

[17] Yong Wang, Han-Xiong Li, Tingwen Huang, Long Li, Differential evolution based on covariance matrix learning and bimodal distribution parameter setting, Appl. Soft Comput. 18 (2014) 232–247.

[18] Jingqiao Zhang, Arthur C. Sanderson, JADE: adaptive differential evolution with optional external archive, IEEE Trans. Evol. Comput. 13 (5) (2009) 945–958.

[19] Ryoji Tanabe, Alex Fukunaga, Success-history based parameter adaptation for differential evolution, in: 2013 IEEE Congress on Evolutionary Computation, IEEE, 2013, pp. 71–78.

[20] Adam Viktorin, Roman Senkerik, Michal Pluhacek, Tomas Kadavy, Ales Zamuda, Distance based parameter adaptation for success-history based differential evolution, Swarm Evol. Comput. 50 (2019) 100462.

[21] Noor H. Awad, Mostafa Z. Ali, Ponnuthurai N. Suganthan, Ensemble of parameters in a sinusoidal differential evolution with niching-based population reduction, Swarm Evol. Comput. 39 (2018) 141–156.

[22] Zhenyu Meng, Jeng-Shyang Pan, Kuo-Kun Tseng, Pade: An enhanced differential evolution algorithm with novel control parameter adaptation schemes for numerical optimization, Knowl.-Based Syst. 168 (2019) 80–99.

[23] Wei-Jie Yu, Meie Shen, Wei-Neng Chen, Zhi-Hui Zhan, Yue-Jiao Gong, Ying Lin, Ou Liu, Jun Zhang, Differential evolution with two-level parameter adaptation, IEEE Trans. Cybern. 44 (7) (2013) 1080–1099.

[24] Zhenghao Song, Zhenyu Meng, Differential evolution with wavelet basis function based parameter control and dimensional interchange for diversity enhancement, Appl. Soft Comput. (2023) 110492.

[25] Yin-Zhi Zhou, Wen-Chao Yi, Liang Gao, Xin-Yu Li, Adaptive differential evolution with sorting crossover rate for continuous optimization problems, IEEE Trans. Cybern. 47 (9) (2017) 2742–2753.

[26] Zhenyu Meng, Zhenghao Song, Xueying Shao, Junyuan Zhang, Huarong Xu, FD-DE: Differential evolution with fitness deviation based adaptation in parameter control, ISA Trans. (2023).

[27] Xuewen Xia, Lei Tong, Yinglong Zhang, Xing Xu, Honghe Yang, Ling Gui, Yuanxiang Li, Kangshun Li, NFDDE: A novelty-hybrid-fitness driving differential evolution algorithm, Inform. Sci. 579 (2021) 33–54.

[28] Xuewen Xia, Ling Gui, Yinglong Zhang, Xing Xu, Fei Yu, Hongrun Wu, Bo Wei, Guoliang He, Yuanxiang Li, Kangshun Li, A fitness-based adaptive differential evolution algorithm, Inform. Sci. 549 (2021) 116–141.

[29] Nikolaus Hansen, Andreas Ostermeier, Completely derandomized self-adaptation in evolution strategies, Evol. Comput. 9 (2) (2001) 159–195.

[30] Yuan-Long Li, Zhi-Hui Zhan, Yue-Jiao Gong, Wei-Neng Chen, Jun Zhang, Yun Li, Differential evolution with an evolution path: A DEEP evolutionary algorithm, IEEE Trans. Cybern. 45 (9) (2014) 1798–1810.

[31] Xiaoyu He, Yuren Zhou, Zefeng Chen, An evolution path-based reproduction operator for many-objective optimization, IEEE Trans. Evol. Comput. 23 (1) (2017) 29–43.

[32] Gai-Ge Wang, Ying Tan, Improving metaheuristic algorithms with information feedback models, IEEE Trans. Cybern. 49 (2) (2017) 542–555.

[33] Yongsheng Liang, Zhigang Ren, Xianghua Yao, Zuren Feng, An Chen, Wenhua Guo, Enhancing Gaussian estimation of distribution algorithm by exploiting evolution direction with archive, IEEE Trans. Cybern. 50 (1) (2018) 140–152.

[34] Li Ming Zheng, Sheng Xin Zhang, Kit Sang Tang, Shao Yong Zheng, Differential evolution powered by collective information, Inform. Sci. 399 (2017) 13–29.

[35] Xiao-Fang Liu, Zhi-Hui Zhan, Ying Lin, Wei-Neng Chen, Yue-Jiao Gong, Tian-Long Gu, Hua-Qiang Yuan, Jun Zhang, Historical and heuristic-based adaptive differential evolution, IEEE Trans. Syst. Man Cybern. Syst. 49 (12) (2018) 2623–2635.

[36] Zhenyu Meng, Cheng Yang, Hip-DE: Historical population based mutation strategy in differential evolution with parameter adaptive mechanism, Inform. Sci. 562 (2021) 44–77.

[37] Xuewen Xia, Ling Gui, Fei Yu, Hongrun Wu, Bo Wei, Ying-Long Zhang, Zhi-Hui Zhan, Triple archives particle swarm optimization, IEEE Trans. Cybern. 50 (12) (2019) 4862–4875.

[38] Jie Hu, Min Wu, Xin Chen, Sheng Du, Pan Zhang, Weihua Cao, Jinhua She, A multilevel prediction model of carbon efficiency based on the differential evolution algorithm for the iron ore sintering process, IEEE Trans. Ind. Electron. 65 (11) (2018) 8778–8787.

[39] Valentino Santucci, Marco Baioletti, Alfredo Milani, Algebraic differential evolution algorithm for the permutation flowshop scheduling problem with total flowtime criterion, IEEE Trans. Evol. Comput. 20 (5) (2015) 682–694.

[40] Emrah Hancer, Fuzzy kernel feature selection with multi-objective differential evolution algorithm, Connect. Sci. 31 (4) (2019) 323–341.

[41] Rafael Rivera-López, Efrén Mezura-Montes, Juana Canul-Reich, Marco Antonio Cruz-Chávez, A permutational-based differential evolution algorithm for feature subset selection, Pattern Recognit. Lett. 133 (2020) 86–93.

[42] Serdar Özyön, Optimal short-term operation of pumped-storage power plants with differential evolution algorithm, Energy 194 (2020) 116866.

[43] Yue-Jiao Gong, Yicong Zhou, Differential evolutionary superpixel segmentation, IEEE Trans. Image Process. 27 (3) (2017) 1390–1404.

[44] Xiao Yang, Rui Wang, Dong Zhao, Fanhua Yu, Ali Asghar Heidari, Zhangze Xu, Huiling Chen, Abeer D Algarni, Hela Elmannai, Suling Xu, Multi-level threshold segmentation framework for breast cancer images using enhanced differential evolution, Biomed. Signal Process. Control 80 (2023) 104373.

[45] Ryoji Tanabe, Alex S. Fukunaga, Improving the search performance of SHADE using linear population size reduction, in: 2014 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2014, pp. 1658–1665.

[46] Guohua Wu, Rammohan Mallipeddi, Ponnuthurai Nagaratnam Suganthan, Problem Definitions and Evaluation Criteria for the CEC 2017 Competition on Constrained Real-Parameter Optimization, Technical Report, National University of Defense Technology, Changsha, Hunan, PR China and Kyungpook National University, Daegu, South Korea and Nanyang Technological University, Singapore, 2017.

[47] Jing J Liang, BY Qu, Ponnuthurai Nagaratnam Suganthan, Alfredo G Hernández-Díaz, Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session on Real-Parameter Optimization, Technical Report, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, 2013, pp. 281–295, 201212(34).

[48] Janez Brest, Mirjam Sepesy Maučec, Borko Bošković, Single objective real-parameter optimization: Algorithm jSO, in: 2017 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2017, pp. 1311–1318.

[49] Zhenyu Meng, Yuxin Zhong, Cheng Yang, Cs-de: Cooperative strategy based differential evolution with population diversity enhancement, Inform. Sci. 577 (2021) 663–696.
[50] Zhenyu Meng, Cheng Yang, Two-stage differential evolution with novel parameter control, Inform. Sci. 596 (2022) 321–342.
[51] Zhenyu Meng, Yuxin Zhong, Guojun Mao, Yan Liang, PSO-sono: A novel PSO variant for single-objective numerical optimization, Inform. Sci. 586 (2022) 176–191.
[52] Geng Zhang, Yuhui Shi, Hybrid sampling evolution strategy for solving single objective bound constrained problems, in: 2018 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2018, pp. 1–7.
[53] Zhenyu Meng, Yuxin Chen, Xiaoqing Li, Cheng Yang, Yuxin Zhong, Enhancing quasi-affine transformation evolution (QUATRE) with adaptation scheme on numerical optimization, Knowl.-Based Syst. 197 (2020) 105908.
[54] Petr Bujok, Patrik Kolenovsky, Eigen crossover in cooperative model of evolutionary algorithms applied to cec 2022 single objective numerical optimisation, in: 2022 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2022, pp. 1–8.
[55] Hoang Anh Pham, Truss optimization with frequency constraints using enhanced differential evolution based on adaptive directional mutation and nearest neighbor comparison, Adv. Eng. Softw. 102 (2016) 142–154.
[56] Karam M Sallam, Saber M Elsayed, Ruhul A Sarker, Daryl L Essam, Landscape-assisted multi-operator differential evolution for solving constrained optimization problems, Expert Syst. Appl. 162 (2020) 113033.
[57] Michael Hellwig, Hans-Georg Beyer, A modified matrix adaptation evolution strategy with restarts for constrained real-world problems, in: 2020 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2020, pp. 1–8.
[58] Javier Gurrola-Ramos, Arturo Hernàndez-Aguirre, Oscar Dalmau-Cedeño, COLSHADE for real-world single-objective constrained optimization problems, in: 2020 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2020, pp. 1–8.