



ARTICLE

Cooperative Metaheuristics with Dynamic Dimension Reduction for High-Dimensional Optimization Problems

Junxiang Li^{1,2}, Zhipeng Dong², Ben Han³, Jianqiao Chen³ and Xinxin Zhang^{1,2,*}

¹Hubei Key Laboratory of Digital Textile Equipment, Wuhan Textile University, Wuhan, 430200, China

²School of Mechanical Engineering and Automation, Wuhan Textile University, Wuhan, 430200, China

³Department of Mechanics, Huazhong University of Science and Technology, Wuhan, 430074, China

*Corresponding Author: Xinxin Zhang. Email: zxx@wtu.edu.cn

Received: 24 July 2025; Accepted: 29 August 2025; Published: 10 November 2025

ABSTRACT: Owing to their global search capabilities and gradient-free operation, metaheuristic algorithms are widely applied to a wide range of optimization problems. However, their computational demands become prohibitive when tackling high-dimensional optimization challenges. To effectively address these challenges, this study introduces cooperative metaheuristics integrating dynamic dimension reduction (DR). Building upon particle swarm optimization (PSO) and differential evolution (DE), the proposed cooperative methods C-PSO and C-DE are developed. In the proposed methods, the modified principal components analysis (PCA) is utilized to reduce the dimension of design variables, thereby decreasing computational costs. The dynamic DR strategy implements periodic execution of modified PCA after a fixed number of iterations, resulting in the important dimensions being dynamically identified. Compared with the static one, the dynamic DR strategy can achieve precise identification of important dimensions, thereby enabling accelerated convergence toward optimal solutions. Furthermore, the influence of cumulative contribution rate thresholds on optimization problems with different dimensions is investigated. Metaheuristic algorithms (PSO, DE) and cooperative metaheuristics (C-PSO, C-DE) are examined by 15 benchmark functions and two engineering design problems (speed reducer and composite pressure vessel). Comparative results demonstrate that the cooperative methods achieve significantly superior performance compared to standard methods in both solution accuracy and computational efficiency. Compared to standard metaheuristic algorithms, cooperative metaheuristics achieve a reduction in computational cost of at least 40%. The cooperative metaheuristics can be effectively used to tackle both high-dimensional unconstrained and constrained optimization problems.

KEYWORDS: Dimension reduction; modified principal components analysis; high-dimensional optimization problems; cooperative metaheuristics; metaheuristic algorithms

1 Introduction

Traditional gradient optimization algorithms, such as the steepest descent [1], Newton [2], and quasi Newton [3] methods, effectively utilize gradient information of the objective function, enabling rapid convergence and low computational cost. However, they are susceptible to converging on local optima rather than finding global optima. In practical engineering structural design, where optimization problems are typically high-dimensional and nonlinear, gradient methods are highly susceptible to converging to local optima, resulting in suboptimal solutions. To overcome these limitations, metaheuristic algorithms have emerged and have been employed in numerous fields, including machine learning hyperparameter optimization [4,5], path planning [6], scheduling problems [7], and structural optimization [8].



Metaheuristic algorithms require no gradient information and conduct global searches for optimal solutions, effectively avoiding local optima traps. These methods fall into two primary categories: swarm intelligence algorithms [9] and evolutionary algorithms [10]. Exhibiting strong robustness and adaptability, such algorithms are well-suited for complex engineering optimization problems and have emerged as powerful tools for solving high-dimensional nonlinear optimization challenges.

Swarm intelligence algorithms draw inspiration from the collective behaviors observed in bird flocks, fish schools, ant colonies, and bee swarms. Examples include particle swarm optimization (PSO) [11], ant colony optimization [12], and artificial bee colony algorithm [13]. Recently, Santhosh et al. [14] proposed a modified gray wolf optimization algorithm with feature selection to enable an effective prediction of Parkinson's disease. Wei et al. [15] developed an improved whale optimization algorithm incorporating multi-strategy to address engineering design challenges. Addressing data classification challenges, Bangyal et al. [16] employed a modified bat algorithm with strengthened exploitation capability to mitigate the issue of convergence to local minima. Later, the same authors [17] introduced a modern computerized bat algorithm to examine a comprehensive set of benchmark test functions.

Among these swarm intelligence algorithms, PSO has garnered significant popularity for its conceptual simplicity and ease of implementation [18]. The original formulation of the PSO algorithm was presented by Eberhart and Kennedy [19], and then improved by many researchers. Chen et al. [20] and Ge et al. [21] developed random mutation modifications to enhance search capabilities of PSO. After that, Bala et al. [22] proposed an improved PSO framework with orthogonal initialization and a crossover mechanism to improve both the exploration capability and global search performance. However, these algorithms are often computationally intensive, particularly in the context of optimization problems with high dimensions, strong nonlinearities and complex constraints.

Unlike swarm intelligence algorithms, evolutionary algorithms are modeled on the principles of biological evolution (natural selection, genetic inheritance, and mutation). Examples include differential evolution (DE) [23], genetic algorithm [24], and evolution strategy [25]. DE is particularly prevalent in current literature for simple program implementation and fewer adjustment parameters [26]. Storn and Price [27] originally proposed the DE algorithm. The algorithm is initially intended for continuous variables and further developed to handle discrete variables. Ho-Huu et al. [28] and Vo-Duy et al. [29] proposed adaptive elitist DE, which has demonstrated improved precision and efficiency for discrete and composite optimization problems. Despite these advancements, computational demands remain high, even for relatively high-dimensional optimization problems.

To tackle these issues, researchers have developed innovative strategies to enhance the performance of metaheuristic algorithms for high-dimensional problems. Carlsson et al. [30] employed the eigenstructure of the Hessian for dimension reduction (DR) in intensity modulated radiation therapy optimization, while Fernández Martínez and García Gonzalo [31] applied principal components analysis (PCA) to reduce dimensions of inverse problems. As evidenced by the aforementioned literature, different DR strategies are often combined with metaheuristic algorithms to tackle high-dimensional challenges.

To efficiently solve high-dimensional optimization problems, cooperative metaheuristics with a dynamic DR strategy (C-PSO and C-DE) are developed based on the metaheuristic algorithms (PSO and DE). In the proposed methods, the modified PCA is utilized to reduce the dimension of design variables, thereby decreasing computational costs. The dynamic DR strategy implements periodic execution of modified PCA after a fixed number of iterations, resulting in the important dimensions being dynamically identified. Compared with the static one, the dynamic DR strategy can achieve precise identification of important dimensions, thereby enabling accelerated convergence toward optimal solutions. Furthermore, the effect of different thresholds of cumulative contribution rate on optimization problems with different

dimensions is investigated. Though the threshold for a specific problem is set empirically, the adjustability of the threshold can guarantee the robustness of the cooperative methods. These strategies collectively enhance the computational efficiency and solution accuracy of the cooperative metaheuristics.

The primary advancements in this research are threefold: (1) unlike existing PCA, the modified PCA focuses on the correlation coefficients between variables and fitness values rather than the eigenvalues; (2) as the optimization progresses, the important dimensions will dynamically change; (3) the cumulative contribution rate threshold is investigated to determine the important dimensions.

This paper is organized as follows. [Section 2](#) presents a review of the metaheuristic algorithms. [Section 3](#) describes the cooperative metaheuristics. In [Section 4](#), 15 benchmark test functions are first examined, then a speed reducer and a composite pressure vessel design problem are performed to demonstrate the efficacy of the cooperative methods. [Section 5](#) gives the conclusions.

2 Review of the Metaheuristic Algorithms

PSO is a typical swarm intelligence algorithm, whereas DE represents a classical evolutionary algorithm. These two representative metaheuristic algorithms are employed in this article owing to their simple program implementation, other metaheuristic algorithms can also be incorporated with the DR strategy. PSO has been studied excessively, readers are encouraged to consult reference [32] for comprehensive details.

The DE framework comprises four key steps [23]: initialization, mutation, crossover, and selection. As a parallel direct search method, a population of N individuals in an n -dimensional space at generation k is represented by $x_i^k = (x_{i,1}^k, x_{i,2}^k, \dots, x_{i,n}^k)$, $i = 1, 2, \dots, N$. The specific procedure of the DE algorithm is outlined below:

Step 1: Initialization.

A randomly generated population is created throughout the entire design space. And the new generation of the population is developed by the following steps.

Step 2: Mutation.

During this step, a mutant vector is generated by combining a weighted differential between two population members with a third individual:

$$v_i^{k+1} = x_{r1}^k + F \times (x_{r2}^k - x_{r3}^k) \quad (1)$$

where the random indices r_1, r_2, r_3 should be distinct integers, and the constant $F \in [0, 2]$ serves as a real-valued coefficient that determines the magnitude of differential variation $(x_{r2}^k - x_{r3}^k)$. Additionally, the index i should be different from r_1, r_2, r_3 . As a result, the population size N must be at least four.

Step 3: Crossover.

A crossover process is employed to increase population diversity. Following the crossover, a trial individual is generated as $u_i^{k+1} = (u_{i,1}^{k+1}, u_{i,2}^{k+1}, \dots, u_{i,n}^{k+1})$. The crossover operation is defined as follows:

$$u_{i,j}^{k+1} = \begin{cases} v_{i,j}^{k+1}, & r(j) \leq CR \text{ or } j = r(i) \\ x_{i,j}^k, & \text{otherwise} \end{cases} \quad (2)$$

in which $r(j)$ represents a randomly generated value corresponding to the j -th dimension, which is uniformly distributed within the range $[0, 1]$; CR is a user-specified crossover rate, bounded between 0 and 1; while $r(i)$ denotes an index chosen stochastically from the set $\{1, 2, \dots, n\}$ that ensures the trial individual u_i^{k+1} inherits at least one parameter from the mutant individual v_i^{k+1} .

Step 4: Selection.

The selection operation is employed to retain individuals with better evaluation function values. In this way, the population can progressively converge toward the optimum. The selection operation is given by:

$$x_i^{k+1} = \begin{cases} u_i^{k+1}, & f(u_i^{k+1}) \leq f(x_i^k) \\ x_i^k, & \text{otherwise} \end{cases} \quad (3)$$

where $f()$ is the evaluation function.

3 The Proposed Cooperative Metaheuristics

As previously mentioned, metaheuristic algorithms are often incorporated with different DR strategies to address high-dimensional optimization challenges. Common DR strategies include: PCA [31], Kernel PCA [33], independent component analysis [34], and t-distributed stochastic neighbor embedding [35]. Compared to these alternatives, PCA offers superior computational efficiency and implementation simplicity, motivating its adoption in this study.

3.1 The Basic PCA

High-dimensional data often presents complex patterns, which makes graphical representation challenging. In such scenarios, PCA is an invaluable technique in data analysis, notably for its capability to reveal underlying data patterns while allowing for DR without significant loss of information. The detailed procedure for performing PCA on a dataset is as follows:

Step 1: Obtain the data.

In this step, an n -dimensional dataset is utilized to illustrate the entire PCA procedure. The variables are denoted as X_i , $i = 1, 2, \dots, n$ with the corresponding dataset represented as:

$$\begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m,1} & x_{m,2} & \cdots & x_{m,n} \end{bmatrix} \quad (4)$$

where m is the size of the dataset, and $x_i = [x_{1,i}, x_{2,i}, \dots, x_{m,i}]^T$ are the data of the variable X_i .

Step 2: Subtract the mean.

To center the data, we subtract the mean of each variable from the corresponding data, resulting in a dataset with a mean of zero:

$$\begin{bmatrix} x'_1 & x'_2 & \cdots & x'_n \end{bmatrix} = \begin{bmatrix} x_{1,1} - \bar{x}_1 & x_{1,2} - \bar{x}_2 & \cdots & x_{1,n} - \bar{x}_n \\ x_{2,1} - \bar{x}_1 & x_{2,2} - \bar{x}_2 & \cdots & x_{2,n} - \bar{x}_n \\ \vdots & \vdots & \ddots & \vdots \\ x_{m,1} - \bar{x}_1 & x_{m,2} - \bar{x}_2 & \cdots & x_{m,n} - \bar{x}_n \end{bmatrix} \quad (5)$$

where $\bar{x}_i = (x_{1,i} + x_{2,i} + \dots + x_{m,i})/m$ is the mean of the data x_i .

Step 3: Calculate the covariance matrix.

$$\text{Cov} = \begin{bmatrix} \text{cov}(X_1, X_1) & \text{cov}(X_1, X_2) & \cdots & \text{cov}(X_1, X_n) \\ \text{cov}(X_2, X_1) & \text{cov}(X_2, X_2) & \cdots & \text{cov}(X_2, X_n) \\ \vdots & \vdots & \ddots & \vdots \\ \text{cov}(X_n, X_1) & \text{cov}(X_n, X_2) & \cdots & \text{cov}(X_n, X_n) \end{bmatrix} \quad (6)$$

where each covariance term can be calculated using the formula:

$$\text{cov}(X_i, X_j) = \frac{x'_i \cdot x'_j}{m-1} = \frac{\sum_{k=1}^m x'_{k,i} x'_{k,j}}{m-1} \quad (7)$$

Step 4: Compute the eigenvalues and corresponding eigenvectors for the covariance matrix.

$$\text{eigenvalues: } \lambda_1, \lambda_2, \dots, \lambda_n \quad (8)$$

$$\text{eigenvectors: } \text{eig}_1 = \begin{bmatrix} a_{1,1} \\ a_{2,1} \\ \vdots \\ a_{n,1} \end{bmatrix}, \text{eig}_2 = \begin{bmatrix} a_{1,2} \\ a_{2,2} \\ \vdots \\ a_{n,2} \end{bmatrix}, \dots, \text{eig}_n = \begin{bmatrix} a_{1,n} \\ a_{2,n} \\ \vdots \\ a_{n,n} \end{bmatrix} \quad (9)$$

Step 5: Construct the principal components.

These new variables can be expressed in terms of the eigenvectors:

$$F_i = a_{i,1}X_1 + a_{i,2}X_2 + \dots + a_{i,n}X_n, i = 1, 2, \dots, n \quad (10)$$

Step 6: Determine the contribution rate of each principal component.

Based on the eigenvalues, the contribution rate of each principal component is calculated using the formula:

$$C_i = \frac{\lambda_i}{\sum_{j=1}^n \lambda_j}, i = 1, 2, \dots, n \quad (11)$$

in which C_i denotes the contribution rate corresponds to the i -th principal component. These rates inform the selection of the most significant components for analysis, while those with minimal contributions are typically excluded to streamline the process.

3.2 The Modified PCA

As previously mentioned, the contribution rates of principal components can be utilized to select certain dimensions for analysis. When addressing a high-dimensional optimization problem, it is feasible to identify and retain only the most important dimensions as the design variables. In this context, the modified PCA diverges slightly from the basic PCA.

In basic PCA, each dimension is interrelated, meaning that the dimensions can influence each other. Conversely, in optimization problems, while each dimension does affect the function value, they do not have direct influences on one another. To adapt this concept for optimization, the modified PCA is developed to calculate contribution rates using the correlation coefficients between each dimension and the function value, rather than the eigenvalues. Subsequently, we can select the most important dimensions for the optimization process. The detailed procedure is outlined below.

Consider an n -dimensional optimization problem represented by the following formulation:

$$\begin{aligned} \min: & f(\mathbf{X}) \\ \text{where: } & \mathbf{X} = [X_1, X_2, \dots, X_n] \\ \text{subject to: } & g_j(\mathbf{X}) \leq 0, j = 1, 2, \dots, l \\ & X_{i,\min} \leq X_i \leq X_{i,\max}, i = 1, 2, \dots, n \end{aligned} \quad (12)$$

in which \mathbf{X} is a vector of n -dimensional design variables; $f(\mathbf{X})$ represents the objective function; while $g_j(\mathbf{X}) \leq 0$ defines the j -th constraint; X_i is bounded by $X_{i,\min}$ and $X_{i,\max}$.

The implementation procedure for the modified PCA is described below:

Step 1: Obtain the data.

The $(n + 1)$ -dimensional data can be obtained by:

$$\begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} & f_1 \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} & f_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{m,1} & x_{m,2} & \cdots & x_{m,n} & f_m \end{bmatrix} \quad (13)$$

where m is the size of the data, and $x_i = [x_{1,i}, x_{2,i}, \dots, x_{m,i}]^T$ are the data of the variable X_i ; $f_j = f(x_{j,1}, x_{j,2}, \dots, x_{j,n})$ is the function value of the j -th data.

Step 2: Compute the correlation coefficients.

The correlation coefficients between each dimension and the function value can then be calculated as follows:

$$\text{cor}(X_i, f) = \frac{\sum_{j=1}^m (x_{j,i} - \bar{x}_i)(f_j - \bar{f})}{\sqrt{\sum_{j=1}^m (x_{j,i} - \bar{x}_i)^2} \sqrt{\sum_{j=1}^m (f_j - \bar{f})^2}}, i = 1, 2, \dots, n \quad (14)$$

where $\bar{x}_i = (x_{1,i} + x_{2,i} + \dots + x_{m,i})/m$ is the mean of the data x_i ; $\bar{f} = (f_1 + f_2 + \dots + f_m)/m$ is the mean of all function values f .

Step 3: Calculate the contribution rates.

The contribution rates can then be determined as follows:

$$C_i = \frac{|\text{cor}(X_i, f)|}{\sum_{j=1}^n |\text{cor}(X_j, f)|}, i = 1, 2, \dots, n \quad (15)$$

Step 4: Sort the contribution rates and calculate the cumulative contribution rates.

The contribution rates are sorted in descending order, and it is assumed that C'_1, C'_2, \dots, C'_n represent these sorted values with X'_1, X'_2, \dots, X'_n being the corresponding design variables. The cumulative contribution rate is then calculated using the following formula:

$$\text{CT}_k = \sum_{i=1}^k C'_i, k = 1, 2, \dots, n \quad (16)$$

Step 5: Select the important dimensions.

At this stage, the important dimensions can be selected based on a predefined threshold for the cumulative contribution rate C_{th} . The important dimensions are determined until the cumulative contribution rate reaches or exceeds the critical value C_{th} . For instance, if $CT_{k-1} < C_{th}$ and $CT_k \geq C_{th}$, the variables X'_1, X'_2, \dots, X'_k are then determined as the important dimensions, while the variables $X'_{k+1}, X'_{k+2}, \dots, X'_n$ are neglected.

Dynamic DR strategy implements periodic execution of modified PCA after a fixed number of iterations to dynamically identify the important dimensions. The important dimensions are chosen according to the threshold, so that variables with cumulative contribution rates exceeding the specified threshold are selected as important dimensions. However, the contribution rates of variables may vary with each iteration, ultimately leading to dynamic changes in cumulative contribution rates and important dimensions. During initial optimization stages, limited data availability compromises accurate identification of important dimensions, adversely affecting solution quality. As optimization progresses, accumulating data enables dynamic adjustment mechanisms to achieve precise identification of important dimensions, thereby enabling accelerated convergence toward optimal solutions.

3.3 The Cooperative Metaheuristics and the Procedure Description

Once the cumulative contribution rate threshold C_{th} is defined, the dimensions can be classified, allowing for the selection of important dimensions to serve as design variables in the optimization problem. The cooperative approaches integrate dynamic DR strategy with metaheuristic algorithms to effectively reduce computational costs meanwhile ensure solution precision.

The procedure for implementing the cooperative method can be outlined as follows (considering an n -dimensional problem with m data points):

Step 1: Generate an initial population.

An initial population with size m is first generated, these individuals are denoted as $x_i = [x_{1,i}, x_{2,i}, \dots, x_{m,i}]^T, i = 1, 2, \dots, n$.

Step 2: Evaluate the fitness values.

Corresponding fitness values $f = [f_1, f_2, \dots, f_m]^T, f_j = f(x_{j,1}, x_{j,2}, \dots, x_{j,n}), j = 1, 2, \dots, m$ are evaluated.

Step 3: Calculate the correlation coefficients.

Based on the data shown in Eq. (13), the correlation coefficients $\text{cor}(X_i, f)$ between X_i and f are then calculated using Eq. (14).

Step 4: Calculate the contribution rates.

The contribution rates C_i are computed using Eq. (15).

Step 5: Sort the contribution rates and calculate the cumulative contribution rates.

Sort the contribution rates as C_1', C_2', \dots, C_n' and calculate the cumulative contribution rates CT_k using Eq. (16).

Step 6: Classify the dimensions based on the cumulative contribution rates.

Suppose that $CT_{k-1} < C_{th}$ and $CT_k \geq C_{th}$, X'_1, X'_2, \dots, X'_k are determined as the important dimensions, while $X'_{k+1}, X'_{k+2}, \dots, X'_n$ are neglected dimensions. Corresponding data become $X'_i = [X'_{1,i}, X'_{2,i}, \dots, X'_{m,i}]^T, i = 1, 2, \dots, n$.

Step 7: Define the optimization problem after DR.

Treat the important dimensions X'_1, X'_2, \dots, X'_k as design variables and treat the neglected dimensions $X'_{k+1}, X'_{k+2}, \dots, X'_n$ as constants. In this way, the n -dimensional optimization problem is reduced to a k -dimensional optimization problem.

Step 8: Utilize the metaheuristic algorithms to update the population.

Employ the specified optimization algorithm (PSO, DE) to tackle the k -dimensional optimization problem, resulting in a new population denoted as $p'_i = [p'_{1,i}, p'_{2,i}, \dots, p'_{m,i}]^T$, $i = 1, 2, \dots, k$. The new population for the n -dimensional optimization problem is then defined as $p'_1, p'_2, \dots, p'_k, X'_{k+1}, X'_{k+2}, \dots, X'_n$.

Step 9: Determine whether the optimization process is terminated.

The optimization process terminates when either the convergence criterion is satisfied or the maximum iteration count is exhausted, outputting the current solution as optimal. If neither condition is met, the algorithm reverts to Step 2.

Fig. 1 illustrates the comprehensive flowchart of the optimization procedure.

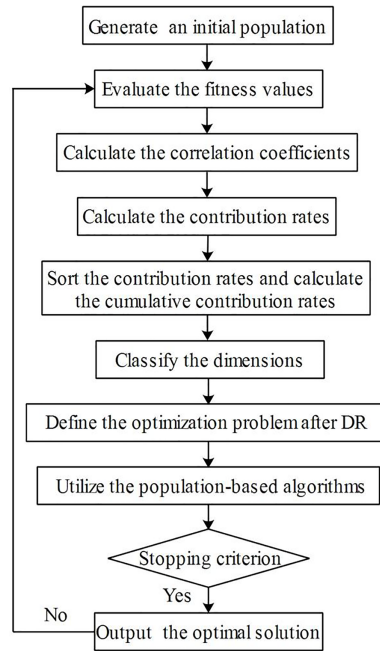


Figure 1: The flowchart of the cooperative metaheuristics

4 Numerical Applications

The parameters used in metaheuristic algorithms (PSO, DE) and cooperative methods (C-PSO, C-DE) are first defined. The population size and maximum number of iterations are specified as 200 for PSO and DE. In C-PSO and C-DE, the population size is reduced to 20, and the number of iterations for DR is set 20, i.e., the dynamic DR strategy is performed after every 20 iterations. The other parameters setting of algorithms are detailed in Table 1.

Table 1: The parameters setting of algorithms

Algorithms		Parameters
PSO	$c_1 = c_2 = 1.8$	
C-PSO	$\omega = 0.6$	$C_{th} = 0.2-0.4$ for 30-dimensional problems
DE	$F = 0.5$	$C_{th} = 0.05-0.2$ for 100-dimensional problems
C-DE	$CR = 0.9$	

Note: The threshold C_{th} for specific problem is set empirically.

To assess the overall performance of all algorithms, different algorithms (PSO, DE, C-PSO, and C-DE) are independently run 30 times for all examples.

4.1 Benchmark Functions

Within evolutionary computation, algorithmic performance is typically assessed through benchmark testing. For this purpose, 15 benchmark functions [36] are employed to examine the proposed C-PSO and C-DE algorithms. These test functions cover various problem types, comprising unimodal, multimodal, separable, and non-separable characteristics. The specifics of the variable ranges, problem characteristics, function types, and formulations are summarized in Table 2.

Table 2: Details of benchmark functions

No.	Range	Characteristic	Function	Formulation
1	$[-1.28, 1.28]$	US	Quartic	$f(x) = \sum_{i=1}^n ix_i^4$
2	$[-100, 100]$	US	Sphere	$f(x) = \sum_{i=1}^n x_i^2$
3	$[-100, 100]$	US	Step	$f(x) = \sum_{i=1}^n (x_i + 0.5)^2$
4	$[-5.12, 5.12]$	US	Stepint	$f(x) = \sum_{i=1}^n x_i $
5	$[-10, 10]$	US	SumSquares	$f(x) = \sum_{i=1}^n ix_i^2$
6	$[-5, 10]$	UN	Zakharov	$f(x) = \sum_{i=1}^n x_i^2 + \left(\sum_{i=1}^n 0.5ix_i \right)^2 + \left(\sum_{i=1}^n 0.5ix_i \right)^4$
7	$[-10, 10]$	UN	Schwefel 2.22	$f(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $
8	$[-100, 100]$	UN	Schwefel 1.2	$f(x) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$
9	$[-30, 30]$	UN	Rosenbrock	$f(x) = \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right]$
10	$[-10, 10]$	UN	Dixon-Price	$f(x) = (x_1 - 1)^2 + \sum_{i=2}^n i(2x_i^2 - x_{i-1})^2$
11	$[-5.12, 5.12]$	MS	Rastrigin	$f(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$
12	$[-32, 32]$	MN	Ackley	$f(x) = -20 \exp \left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right) - \exp \left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i) \right) + 20 + e$
13	$[-600, 600]$	MN	Griewank	$f(x) = 1/4000 \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1$
14	$[-50, 50]$	MN	Penalized	$f(x) = \frac{\pi}{n} \left\{ 10 \sin^2(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + \sin^2(\pi y_{i+1})] \right.$ $\left. + (y_n - 1)^2 \right\} + \sum_{i=1}^n u(x_i, 10, 100, 4)$ $y_i = 1 + 1/4(x_i + 1)$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \leq x_i \leq a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$

(Continued)

Table 2 (continued)

No.	Range	Characteristic	Function	Formulation
15	$[-50, 50]$	MN	Penalized2	$f(x) = 0.1 \left\{ \sin^2(\pi x_1) + \sum_{i=1}^{n-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] \right.$ $\left. + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)] \right\} + \sum_{i=1}^n u(x_i, 5, 100, 4)$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \leq x_i \leq a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$

Note: U: Unimodal, M: Multimodal, S: Separable, N: Non-Separable.

The optimizations are conducted for both 30-dimensional and 100-dimensional problems as detailed in Table 2. Comparative results from 30 individual runs for all benchmark functions are presented in Table 3. Notably, ‘Mean’ reflects the solution quality. For minimization problems, a lower ‘Mean’ indicates superior algorithm performance. ‘SD’ serves as a robustness indicator, where a smaller value demonstrates higher algorithmic stability. ‘N.F.’ measures computational efficiency, with fewer evaluations signifying greater efficiency. Since the execution time of the algorithm itself is negligible compared to that of response function evaluations or finite element analyses, the number of function calls or finite element program calls serves as a direct indicator of computational efficiency. Consequently, training and testing times are not reported herein.

Table 3: Comparative results from 30 individual runs for all benchmark functions

No.	Theoretical solution	Method	PSO		DE		C-PSO		C-DE	
		Dimension	30	100	30	100	30	100	30	100
1	0	Mean	0.008	5.364	0.007	6.216	3.36E-7	0.135	5.63E-7	3.145
		SD	0.009	2.392	0.005	1.124	6.25E-7	0.052	5.42E-7	0.044
		N.F.	4000	18,400	26,000	22,000	1400	5760	5080	4240
2	0	Mean	8.24E-6	4.225	3.54E-6	4.012	0	2.401	0	2.045
		SD	1.12E-6	1.024	2.44E-7	1.045	0	0.054	0	0.075
		N.F.	13,400	17,400	24,600	22,400	5600	6120	2920	4060
3	0	Mean	7.45E-7	6.245	6.48E-6	7.145	0	3.147	0	4.782
		SD	1.78E-6	1.057	1.26E-6	0.478	0	0.107	0	0.540
		N.F.	8400	19,400	9800	14,600	6140	5060	6240	4800
4	0	Mean	6.01E-6	5.46E-4	8.96E-7	8.66E-4	0	2.84E-4	0	4.77E-4
		SD	2.32E-6	3.25E-5	3.11E-7	2.43E-5	0	1.25E-5	0	1.43E-5
		N.F.	6200	20,400	14,600	18,400	5040	5400	4860	4440
5	0	Mean	1.16E-3	5.88E-2	1.36E-3	5.45E-2	0	0	0	0
		SD	5.04E-4	1.51E-2	7.61E-4	1.24E-2	0	0	0	0
		N.F.	13,400	18,600	18,200	24,200	4880	4240	5040	5400
6	0	Mean	5.64E-8	0.123	6.28E-8	0.134	0	0.063	0	0.045
		SD	3.17E-9	0.090	7.36E-9	4.41E-2	0	4.04E-2	0	4.41E-2
		N.F.	18,000	23,400	18,600	25,400	5100	5180	3050	4080
7	0	Mean	3.48E-7	2.54E-4	5.96E-7	3.07E-4	0	0	0	0
		SD	2.84E-7	1.56E-4	4.78E-7	2.57E-5	0	0	0	0
		N.F.	16,200	21,000	14,600	18,000	4060	6060	5060	4880
8	0	Mean	8.28E-6	8.25E-4	6.35E-6	6.67E-4	0	0	0	0
		SD	4.59E-6	3.17E-4	7.22E-6	3.97E-4	0	0	0	0
		N.F.	17,400	18,800	19,800	21,000	3860	4060	4020	4100
9	0	Mean	15.09	20.44	18.20	23.25	10.24	15.45	9.654	11.45
		SD	4.413	3.254	0.033	1.587	2.141	1.781	0.033	1.587
		N.F.	19,200	22,600	24,800	22,000	4200	5420	6140	8400

(Continued)

Table 3 (continued)

No.	Theoretical solution	Method	PSO		DE		C-PSO		C-DE	
		Dimension	30	100	30	100	30	100	30	100
10	0.667	Mean	0.667	0.667	0.667	0.667	0.667	0.667	0.667	0.667
		SD	3.11E−6	2.47E−4	4.54E−5	2.48E−3	0	0	0	0
		N.F.	18,400	20,200	23,600	22,400	3800	5040	5400	6040
11	0	Mean	43.97	45.12	11.71	12.78	43.97	45.12	11.71	12.78
		SD	2.141	3.874	0.463	0.547	2.141	3.874	0.463	0.547
		N.F.	38,000	36,400	40,000	38,000	1120	2860	3840	3420
12	0	Mean	1.73E−2	4.58E−2	1.47E−3	5.45E−3	0	0	0	0
		SD	3.80E−3	4.15E−3	5.43E−4	5.14E−4	0	0	0	0
		N.F.	18,400	24,600	35,000	32,400	2200	2440	3200	2840
13	0	Mean	0.165	0.198	3.64E−6	5.47E−4	0	0.087	0	0
		SD	9.01E−2	6.31E−2	1.24E−7	6.24E−5	0	3.11E−2	0	0
		N.F.	23,000	24,600	26,000	30,200	1520	1980	1280	2100
14	0	Mean	2.07E−2	2.14E−2	5.84E−7	3.04E−2	0	0	0	0
		SD	7.57E−3	5.47E−3	5.94E−7	4.15E−3	0	0	0	0
		N.F.	23,600	24,000	25,400	28,000	2760	3040	1980	2840
15	0	Mean	7.67E−3	8.34E−3	2.19E−3	4.14E−3	0	0	0	0
		SD	5.97E−2	4.57E−3	8.02E−4	7.12E−4	0	0	0	0
		N.F.	20,600	24,200	22,000	20,400	2840	2240	3040	2980

Note: Mean: mean of the optimal solutions, SD: standard deviation of the optimal solutions, N.F.: number of function calls.

The results clearly indicate declining solution accuracy across all algorithms with increasing dimension. Compared to standard PSO and DE, both C-PSO and C-DE achieve better optimization solutions. For 30-dimensional problems, the cooperative methods attain theoretical optimal solutions for all test functions with the exception of Nos. 9 and 11. However, for 100-dimensional problems, C-PSO and C-DE can obtain theoretical optimal solutions for the majority of functions (Nos. 4, 5, 7, 8, 10, and 12–15), demonstrating their remarkable efficacy in addressing high-dimensional optimization tasks.

For the MN functions (Nos. 12–15), both PSO and DE can achieve good optimization results, while C-PSO and C-DE can obtain theoretical optimal solutions. Conversely, for the MS function (No. 11), all methods fail to obtain satisfactory results. Nevertheless, C-PSO and C-DE notably achieve better solutions than the standard methods. This demonstrates that the cooperative metaheuristics substantially enhance solution precision even for challenging optimization problems.

According to the preceding analysis, a critical limitation is revealed: both standard metaheuristics (PSO, DE) and their cooperative variants (C-PSO, C-DE) exhibit poor performance for multimodal separable optimization problems, particularly in high-dimensional cases.

Beyond solution quality, robustness serves as a critical performance metric for evaluating algorithmic efficacy. As evidenced in Table 3, the proposed methods achieve low ‘SD’ values across diverse test functions, with ‘SD’ = 0 observed in several cases, demonstrating exceptional robustness. Furthermore, ‘N.F.’ values of C-PSO and C-DE are always lower than those of PSO and DE for all benchmark functions, showing that the cooperative metaheuristics require less computational cost and have higher computational efficiency.

As described in Sections 3.2 and 3.3, the cumulative contribution rate threshold C_{th} is the decisive factor in defining important dimensions. Therefore, the effect of different thresholds on optimization problems with different dimensions is investigated. The results of Sphere function (No. 2 in Table 2) are given in Table 4.

Table 4: The number of function calls under different C_{th} and dimension for Sphere function

C_{th} Dimension	0.05	0.1	0.2	0.3	0.4
20	30,200	18,800	18,000	22,400	38,000
30	28,400	18,800	12,800	16,400	37,200
50	26,000	15,600	22,000	32,400	–
100	22,800	34,800	71,600	–	–

Note: The desired precision may not be achieved when the maximum number of iterations is reached.

Table 4 demonstrates that the threshold C_{th} is not necessarily set as small as possible. For the low-dimensional problems, a too small threshold can lead to excessive information loss during DR, slowing down the optimization process. For high-dimensional problems, a small threshold can accelerate the optimization process in the early stage. The best outcomes for each dimension in Table 4 are highlighted in bold. As the dimension increases, the threshold needs to be set smaller to accelerate convergence and achieve optimal solutions more rapidly. Though the threshold for specific problem is set empirically, the adjustability of threshold can guarantee the robustness of the method.

4.2 A Speed Reducer Design

To further demonstrate the efficacy of the cooperative optimization algorithms, a more complex example involving a speed reducer design problem [20] is examined. The problem consists of seven dimensions, where x_3 is an integer variable while all other variables remain continuous. The problem is structured as follows:

$$\begin{aligned}
 \min: f(\mathbf{x}) &= 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) - 1.508x_1(x_6^2 + x_7^2) + \\
 & 7.4777(x_6^3 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2) \\
 \text{where: } \mathbf{x} &= [x_1, x_2, \dots, x_7] \\
 \text{subject to: } g_1(\mathbf{x}) &= \frac{27}{x_1x_2^2x_3} - 1 \leq 0, g_2(\mathbf{x}) = \frac{397.5}{x_1x_2^2x_3^2} - 1 \leq 0, g_3(\mathbf{x}) = \frac{1.93x_4^3}{x_2x_3x_6^4} - 1 \leq 0 \\
 g_4(\mathbf{x}) &= \frac{1.93x_5^3}{x_2x_3x_7^4} - 1 \leq 0, g_5(\mathbf{x}) = \frac{x_2x_3}{40} - 1 \leq 0, g_6(\mathbf{x}) = \frac{5x_2}{x_1} - 1 \leq 0 \\
 g_7(\mathbf{x}) &= \frac{x_1}{12x_2} - 1 \leq 0, g_8(\mathbf{x}) = \frac{1.5x_6 + 1.9}{x_4} - 1 \leq 0, g_9(\mathbf{x}) = \frac{1.1x_7 + 1.9}{x_5} - 1 \leq 0 \\
 g_{10}(\mathbf{x}) &= \frac{\sqrt{\left(\frac{745x_4}{x_2x_3}\right)^2 + 16.9 \times 10^6}}{110x_6^3} - 1 \leq 0, g_{11}(\mathbf{x}) = \frac{\sqrt{\left(\frac{745x_5}{x_2x_3}\right)^2 + 157.5 \times 10^6}}{85x_7^3} - 1 \leq 0
 \end{aligned} \tag{17}$$

where,

$$\begin{aligned}
 2.6 \leq x_1 \leq 3.6, 0.7 \leq x_2 \leq 0.8, 17 \leq x_3 \leq 28 \\
 7.3 \leq x_4 \leq 8.3, 7.8 \leq x_5 \leq 8.3, 2.9 \leq x_6 \leq 3.9 \\
 5 \leq x_7 \leq 5.5
 \end{aligned} \tag{18}$$

This is a single-objective problem with 11 behavioral constraints. PSO, DE, and their cooperative counterparts, C-PSO and C-DE, are all employed to solve this problem. Their results are compared with those of ‘Taguchi’ and ‘Sociobehavioural’ from the literature [20], as presented in Table 5. ‘Taguchi’ adopts a varying probe length value, which is initialized at 0.9 and reduced by one-half whenever the search direction shows no improvement. Meanwhile, ‘Sociobehavioural’ is executed using a population size of 100 over 200 iterations.

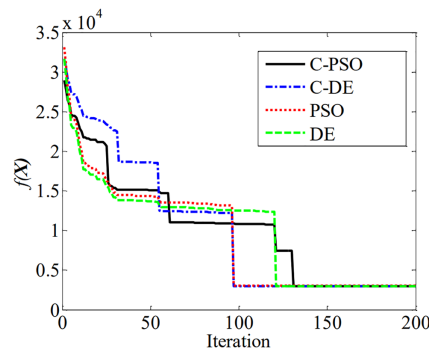
Table 5: Comparative results from 30 individual runs for the speed reducer ($C_{th} = 0.2$)

	Taguchi [20]	Sociobehavioural [20]	PSO	DE	C-PSO	C-DE
x_1	3.500	3.506	3.500	3.500	3.502	3.500
x_2	0.700	0.700	0.700	0.700	0.700	0.700
x_3	17	17	17	17	17	17
x_4	7.300	7.549	8.013	8.134	7.373	7.300
x_5	7.153	7.859	8.258	7.809	7.800	7.800
x_6	3.505	3.366	3.407	3.354	3.351	3.350
x_7	5.287	5.290	5.287	5.288	5.287	5.287
Best $f(\mathbf{x})$	2994.40	3008.08	3028.01	3006.52	2998.20	2996.35
Mean	2996.22	3012.12	3033.43	3012.61	3004.58	3003.46
SD	2.58	3.65	6.28	7.17	7.36	6.95
N.F.	9900	19,154	9600	12,000	4060	5200

'Best $f(\mathbf{x})$ ' reflects the best solution among 30 individual runs. In minimization problems, a superior solution corresponds to a lower value. As evidenced in Table 5, C-PSO (2998.20) and C-DE (2996.35) significantly outperform PSO (3028.01) and DE (3006.52), confirming superior performance of the cooperative methods.

In terms of algorithmic robustness, 'SD' values of all algorithms are very small, which verifies the robustness of these methods. Meanwhile, 'N.F.' values of C-PSO (4060) and C-DE (5200) are significantly less than those of PSO (9600) and DE (12,000), indicating that the cooperative methods require less computational cost and have higher computational efficiency.

In summary, the proposed methods demonstrate superior solutions with reduced computational costs compared to the standard algorithms. This example illustrates the efficacy of the cooperative methods in tackling constrained optimization challenges. The convergence process of different algorithms are provided in Fig. 2.

**Figure 2:** Convergence procedure for the speed reducer

It can be seen that PSO and DE no longer change after convergence, at which point the optimal solution is obtained. The convergence process of such algorithms is almost a one-time convergence solution without any sudden changes. The convergence process of C-PSO and C-DE is clearly a stepwise multiple convergence, mainly because in these two algorithms, the number of design variables dynamically changes with the intervention of the DR strategy.

4.3 A Composite Pressure Vessel Design

This study addresses a composite pressure vessel design problem [37], analyzing a Type 4 compressed hydrogen storage tank wrapped with carbon fiber. To ensure safety, the tank is designed with a safety factor of 2.25, resulting in a minimum burst pressure requirement of 158 MPa. The Toray T700S carbon fiber is used with a rated tensile strength of 4900 MPa, and the fiber-resin composite (60% fiber volume fraction) exhibits a tensile strength of 2550 MPa. The density of the composite material is 1800 kg/m^3 . The liner of the vessel is made from high-density polyethylene (HDPE), with a thickness of 5 mm. The material properties are summarized in Table 6.

Table 6: Material properties for T700S and HDPE

Materials	Parameters	Properties	Unit	Values
T700S	Young moduli	E_1, E_2, E_3	GPa	135, 9.66, 9.66
	Shear moduli	G_{12}, G_{13}, G_{23}	GPa	5.86, 5.86, 3.46
	Poisson's ratios	$\nu_{12}, \nu_{13}, \nu_{23}$		0.25, 0.25, 0.41
	Tensile strengths	F_{1t}, F_{1c}	MPa	2550, 1470
	Compressive strengths	F_{2t}, F_{2c}	MPa	69,224.1
	Shear strength	F_6	MPa	89.63
HDPE	Young modulus	E	MPa	900
	Poisson's ratio	ν		0.46
	Yield strength	S_y	MPa	26

Fig. 3 illustrates a schematic of a storage tank with 521 mm diameter and 1.5 length-to-diameter (L/D) ratio, designed for 5.6 kg usable hydrogen capacity (5.8 kg total) at 70 MPa operating pressure. The tank features 15° helical winding layers and 90° hoop wound layers. During fabrication, helical and hoop layers are applied alternately over the liner to minimize resin pooling.

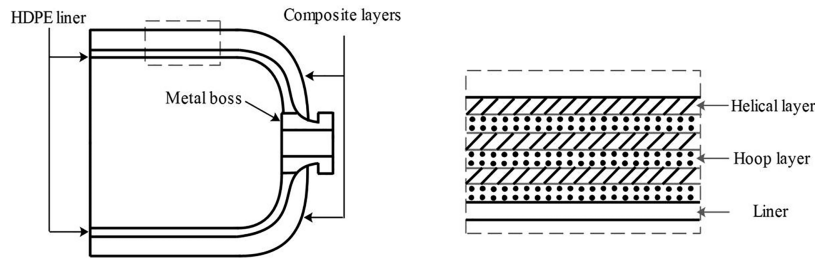


Figure 3: Schematic of a Type 4 compressed hydrogen storage tank

Fig. 4 shows the winding angle θ_0 at the cylinder and θ at the head. These angles are calculated as follows:

$$\theta_0 = \arcsin\left(\frac{r}{R_0}\right), \theta = \arcsin\left(\frac{r}{R}\right) \quad (19)$$

where r represents the outer radius of the vessel, R_0 denotes the outer radius of the cylindrical section, and R signifies the outer radius of the head at a specific location.

Finite element analysis (FEA) is performed using ANSYS, with the SHELL99 element modeling composite materials. This 8-node element features six degrees of freedom per node (translations and

rotations in x , y , z directions). The SOLID191 element is used to simulate liner materials, with 20 nodes and three translational degrees of freedom per node (x , y , and z directions). The total number of elements in the model is 1650.

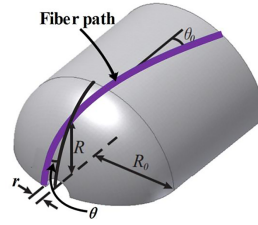


Figure 4: Geodesic fiber path along a pressure vessel surface

The design variables comprise thicknesses of five helical and four hoop layers, each ranging from 0.127 mm to 10 mm. The objective is to minimize the weight of the composite structure, subject to the constraint imposed by the Tsai-Wu strength criterion. The mathematical expression for this problem is:

$$\begin{aligned} &\min: W(\mathbf{d}) \\ &\text{where: } \mathbf{d} = [d_1, d_2, \dots, d_9] \\ &\text{subject to: } |\sigma| \leq \sigma_{\max} \end{aligned} \quad (20)$$

where $W(\mathbf{d})$ is the total weight of the pressure vessel, and \mathbf{d} is the thicknesses of nine layers, including five helical and four hoop layer thicknesses.

Comparative results obtained by 30 individual runs for the composite vessel are summarized in Table 7. The results are compared with those of genetic algorithm [37]. The population size and maximum number of iterations for genetic algorithm are set 200 and 100, respectively. The crossover probability is defined as 0.85 and the mutation probability is given as 0.0001.

Table 7: Comparative results from 30 individual runs for the composite vessel ($C_{th} = 0.2$)

	Genetic algorithm [37]	PSO	DE	C-PSO	C-DE
d_1	4.72	4.70	4.60	4.55	4.33
d_2	4.60	4.48	4.60	4.40	4.52
d_3	4.64	4.68	4.52	4.53	4.54
d_4	4.55	4.63	4.68	4.45	4.45
d_5	4.58	4.70	4.44	4.48	4.40
d_6	6.12	6.06	6.01	6.02	5.96
d_7	6.04	5.97	6.00	5.94	5.83
d_8	5.86	5.88	6.00	5.70	5.85
d_9	5.93	6.10	6.03	5.83	5.90
Best $W(\mathbf{d})$	105.9	106.3	105.6	103.6	103.1
Mean	107.4	106.8	106.2	105.3	104.6
SD	1.72	0.94	1.06	1.86	1.57
N.F.	7580	10,400	9600	6060	4940

‘Best $W(\mathbf{d})$ ’ denotes the best solution across 30 independent runs. For minimization problems, a superior solution is characterized by a smaller value. Table 6 demonstrates that C-PSO (103.6) and C-DE (103.1)

achieve significantly better solutions than PSO (106.3) and DE (105.6), confirming superior performance of the cooperative methods. This advantage mainly stems from design space simplification: despite nine design variables, they resolve to just two parameter types (helical and hoop layer thicknesses), reducing inherent problem complexity.

Regarding algorithmic robustness, consistently low 'SD' values across all methods confirm exceptional solution stability. Computationally, the cooperative methods demonstrate superior efficiency: C-PSO reduces 'N.F.' values by 41.7% (10,400 \rightarrow 6060), while C-DE achieves 48.5% reduction (9600 \rightarrow 4940). Consequently, FEA calls decrease substantially, lowering computational costs. In summary, the proposed cooperative methods achieve superior solutions with better efficiency compared to the standard algorithms.

5 Conclusions

Based on standard metaheuristics PSO and DE, this paper presents cooperative metaheuristics C-PSO and C-DE that integrate dynamic DR. The modified PCA is utilized to reduce the dimension of design variables, thereby lowering computational costs. The dynamic DR strategy implements periodic execution of modified PCA after a fixed number of iterations, enabling dynamic identification of important dimensions. Compared to a static strategy, this dynamic implementation achieves precise identification of important dimensions, thereby driving accelerated convergence toward optimal solutions. Furthermore, the influence of cumulative contribution rate thresholds on optimization problems with different dimensions is investigated. 15 benchmark functions and two engineering design problems (speed reducer and composite pressure vessel) are used to evaluate the proposed methods. The results are compared with those of PSO and DE, revealing that the cooperative methods achieve significantly superior performance compared to standard methods in both solution accuracy and computational efficiency.

Unlike existing PCA that prioritizes the eigenvalues, the cooperative algorithms develop the modified PCA to analyze the correlation coefficients between variables and fitness values. Crucially, the proposed methods dynamically identify important dimensions as optimization progresses. Another key feature is that the cumulative contribution rate threshold is investigated to determine the important dimensions. These strategies collectively improve the computational performance and solution precision of the cooperative methods.

In conclusion, the cooperative metaheuristics show promising performance in tackling the vast majority of high-dimensional optimization problems in terms of efficiency, accuracy, and robustness. Nevertheless, a key limitation is that both standard metaheuristics (PSO, DE) and their cooperative variants (C-PSO, C-DE) exhibit poor performance for multimodal separable optimization problems, particularly in high-dimensional cases. And this is the future direction of our work.

Acknowledgement: Not applicable.

Funding Statement: This research was funded by National Natural Science Foundation of China (Nos. 12402142, 11832013 and 11572134), Natural Science Foundation of Hubei Province (No. 2024AFB235), Hubei Provincial Department of Education Science and Technology Research Project (No. Q20221714), and the Opening Foundation of Hubei Key Laboratory of Digital Textile Equipment (Nos. DTL2023019 and DTL2022012).

Author Contributions: The authors confirm contribution to the paper as follows: Junxiang Li: Conceptualization, Methodology, Formal analysis, Writing—original draft, Writing—review & editing, Project administration, Investigation, Validation; Zhipeng Dong: Software, Data curation, Visualization, Writing—review & editing; Ben Han: Formal analysis, Writing—review & editing; Jianqiao Chen: Investigation, Visualization, Conceptualization, Project administration; Xinxin Zhang: Resources, Writing—review & editing, Supervision. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available from the corresponding author.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Sheikh-Hosseini M, Samareh Hashemi SR. Connectivity and coverage constrained wireless sensor nodes deployment using steepest descent and genetic algorithms. *Expert Syst Appl.* 2022;190(4):116164. doi:10.1016/j.eswa.2021.116164.
2. Wang L, Yuan Q, Zhao B, Zhu B, Zeng X. Parameter identification of photovoltaic cells/modules by using an improved artificial ecosystem optimization algorithm and Newton-Raphson method. *Alex Eng J.* 2025;123(3):559–91. doi:10.1016/j.aej.2025.03.024.
3. Liu C, Chen Y, Shen J, Chang Y. Solving higher-order nonlocal boundary value problems with high precision by the fixed quasi Newton methods. *Math Comput Simul.* 2025;232:211–26. doi:10.1016/j.matcom.2024.12.024.
4. Zhang C, He Y, Yuan L, Deng F. A novel approach for analog circuit fault prognostics based on improved RVM. *J Electron Test.* 2014;30(3):343–56. doi:10.1007/s10836-014-5454-8.
5. Zhang C, Luo L, Yang Z, Zhao S, He Y, Wang X, et al. Battery SOH estimation method based on gradual decreasing current, double correlation analysis and GRU. *Green Energy Intell Transp.* 2023;2(5):100108. doi:10.1016/j.geits.2023.100108.
6. Liu L, Wang X, Yang X, Liu H, Li J, Wang P. Path planning techniques for mobile robots: review and prospect. *Expert Syst Appl.* 2023;227(4):120254. doi:10.1016/j.eswa.2023.120254.
7. Schlenkrich M, Parragh SN. Solving large scale industrial production scheduling problems with complex constraints: an overview of the state-of-the-art. *Procedia Comput Sci.* 2023;217:1028–37. doi:10.1016/j.procs.2022.12.301.
8. Li J, Guo X, Zhang X, Wu Z. Time-dependent reliability assessment and optimal design of corroded reinforced concrete beams. *Adv Struct Eng.* 2024;27(8):1313–27. doi:10.1177/13694332241247923.
9. Pan Z, Lei D, Wang L. A knowledge-based two-population optimization algorithm for distributed energy-efficient parallel machines scheduling. *IEEE Trans Cybern.* 2022;52(6):5051–63. doi:10.1109/TCYB.2020.3026571.
10. Zhao F, Yin F, Wang L, Yu Y. A co-evolution algorithm with dueling reinforcement learning mechanism for the energy-aware distributed heterogeneous flexible flow-shop scheduling problem. *IEEE Trans Syst Man Cybern Syst.* 2025;55(3):1794–809. doi:10.1109/TSMC.2024.3510384.
11. Li J, Chen J. Solving time-variant reliability-based design optimization by PSO-t-IRS: a methodology incorporating a particle swarm optimization algorithm and an enhanced instantaneous response surface. *Reliab Eng Syst Saf.* 2019;191(3):106580. doi:10.1016/j.ress.2019.106580.
12. Mavrovouniotis M, Anastasiadou MN, Hadjimitsis D. Measuring the performance of ant colony optimization algorithms for the dynamic traveling salesman problem. *Algorithms.* 2023;16(12):545. doi:10.3390/a16120545.
13. Xing H, Xing Q. An air defense weapon target assignment method based on multi-objective artificial bee colony algorithm. *Comput Mater Contin.* 2023;76(3):2685–705. doi:10.32604/cmc.2023.036223.
14. Santhosh K, Dev PP, A BJ, Lynton Z, Das P, Ghaderpour E. A modified gray wolf optimization algorithm for early detection of Parkinson's disease. *Biomed Signal Process Control.* 2025;109(5):108061. doi:10.1016/j.bspc.2025.108061.
15. Wei J, Gu Y, Yan Y, Li Z, Lu B, Pan S, et al. LSEWOA: an enhanced whale optimization algorithm with multi-strategy for numerical and engineering design optimization problems. *Sensors.* 2025;25(7):2054. doi:10.3390/s25072054.
16. Bangyal WH, Ahmad J, Rauf HT. Optimization of neural network using improved bat algorithm for data classification. *J Med Imaging Health Inform.* 2019;9(4):670–81. doi:10.1166/jmihi.2019.2654.

17. Haider Bangyal W, Hameed A, Ahmad J, Nisar K, Reazul Haque M, Asri Ag Ibrahim A, et al. New modified controlled bat algorithm for numerical optimization problem. *Comput Mater Contin.* 2022;70(2):2241–59. doi:10.32604/cmc.2022.017789.
18. Yoshida H, Fukuyama Y, Takayama S, Nakanishi Y. A particle swarm optimization for reactive power and voltage control in electric power systems considering voltage security assessment. In: *IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man, and Cybernetics; 1999 Oct 12–15; Tokyo, Japan.* p. 497–502. doi:10.1109/ICSMC.1999.816602.
19. Eberhart R, Kennedy J. Particle swarm optimization. In: *Proceedings of the IEEE International Conference on Neural Networks; 1995 Nov 27–Dec 1; Perth, WA, Australia.* p. 1942–8.
20. Chen J, Ge R, Wei J. Probabilistic optimal design of laminates using improved particle swarm optimization. *Eng Optim.* 2008;40(8):695–708. doi:10.1080/03052150802010615.
21. Ge R, Chen J, Wei J. Reliability-based design of composites under the mixed uncertainties and the optimization algorithm. *Acta Mech Solida Sin.* 2008;21(1):19–27. doi:10.1007/s10338-008-0804-7.
22. Bala I, Karunaratne W, Mitchell L. Optimizing feature selection by enhancing particle swarm optimization with orthogonal initialization and crossover operator. *Comput Mater Contin.* 2025;84(1):727–44. doi:10.32604/cmc.2025.065706.
23. Meselhi MA, Elsayed SM, Essam DL, Sarker RA. Modified differential evolution algorithm for solving dynamic optimization with existence of infeasible environments. *Comput Mater Contin.* 2023;74(1):1–17. doi:10.32604/cmc.2023.027448.
24. Li J, Chen Z, Chen J, Song Q. Optimization design of variable stiffness composite cylinders based on non-uniform rational B-spline curve/surface. *J Compos Mater.* 2023;57(15):2405–19. doi:10.1177/00219983231170427.
25. Wang D, Zhu Y, Fei J, Guo M. CMAES-WFD: adversarial website fingerprinting defense based on covariance matrix adaptation evolution strategy. *Comput Mater Contin.* 2024;79(2):2253–76. doi:10.32604/cmc.2024.049504.
26. Zeng L, Shi J, Li Y, Wang S, Li W. A strengthened dominance relation NSGA-III algorithm based on differential evolution to solve job shop scheduling problem. *Comput Mater Contin.* 2024;78(1):375–92. doi:10.32604/cmc.2023.045803.
27. Storn R, Price K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J Glob Optim.* 1997;11(4):341–59. doi:10.1023/A:1008202821328.
28. Ho-Huu V, Nguyen-Thoi T, Le-Anh L, Nguyen-Trang T. An effective reliability-based improved constrained differential evolution for reliability-based design optimization of truss structures. *Adv Eng Softw.* 2016;92:48–56. doi:10.1016/j.advengsoft.2015.11.001.
29. Vo-Duy T, Ho-Huu V, Dang-Trung H, Nguyen-Thoi T. A two-step approach for damage detection in laminated composite structures using modal strain energy method and an improved differential evolution algorithm. *Compos Struct.* 2016;147:42–53. doi:10.1016/j.compstruct.2016.03.027.
30. Carlsson F, Forsgren A, Rehbinder H, Eriksson K. Using eigenstructure of the Hessian to reduce the dimension of the intensity modulated radiation therapy optimization problem. *Ann Oper Res.* 2006;148(1):81–94. doi:10.1007/s10479-006-0082-z.
31. Fernández Martínez JL, García Gonzalo E. The PSO family: deduction, stochastic analysis and comparison. *Swarm Intell.* 2009;3(4):245–73. doi:10.1007/s11721-009-0034-8.
32. Li J, Han B, Chen J, Wu Z. An improved optimization method combining particle swarm optimization and dimension reduction Kriging surrogate model for high-dimensional optimization problems. *Eng Optim.* 2024;56(12):2307–28. doi:10.1080/0305215X.2024.2309514.
33. Tonin F, Tao Q, Patrinos P, Suykens J. Deep kernel principal component analysis for multi-level feature learning. *Neural Netw.* 2024;170(1):578–95. doi:10.1016/j.neunet.2023.11.045.
34. Li Y, Zhang M, Bian X, Tian L, Tang C. Progress of independent component analysis and its recent application in spectroscopy quantitative analysis. *Microchem J.* 2024;202:110836. doi:10.1016/j.microc.2024.110836.
35. Cao YT, Lin SH, Hundangan RBL, Ang EB, Wu TL. Developing fault prognosis and detection modes for main hoisting motor in gantry crane based on t-SNE. *Comput Mater Contin.* 2025;84(3):5255–77. doi:10.32604/cmc.2025.066877.

36. Karaboga D, Akay B. A comparative study of artificial bee colony algorithm. *Appl Math Comput.* 2009;214(1):108–32. doi:10.1016/j.amc.2009.03.090.
37. Alcántar V, Aceves SM, Ledesma E, Ledesma S, Aguilera E. Optimization of Type 4 composite pressure vessels using genetic algorithms and simulated annealing. *Int J Hydrogen Energy.* 2017;42(24):15770–81. doi:10.1016/j.ijhydene.2017.03.032.