Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.060887

ARTICLE

Collaborative Decomposition Multi-Objective Improved Elephant Clan Optimization Based on Penalty-Based and Normal Boundary Intersection

Mengjiao Wei[1,*] and Wenyu Liu[2]

[1]School of Electrical Engineering, Northeast Electric Power University, Jilin, 132012, China
[2]Northeast Electric Power Design Institute, Changchun, 130000, China

*Corresponding Author: Mengjiao Wei. Email: 2202200366@neepu.edu.cn

Received: 12 November 2024; Accepted: 25 December 2024; Published: 16 April 2025

ABSTRACT: In recent years, decomposition-based evolutionary algorithms have become popular algorithms for solving multi-objective problems in real-life scenarios. In these algorithms, the reference vectors of the Penalty-Based boundary intersection (PBI) are distributed parallelly while those based on the normal boundary intersection (NBI) are distributed radially in a conical shape in the objective space. To improve the problem-solving effectiveness of multi-objective optimization algorithms in engineering applications, this paper addresses the improvement of the Collaborative Decomposition (CoD) method, a multi-objective decomposition technique that integrates PBI and NBI, and combines it with the Elephant Clan Optimization Algorithm, introducing the Collaborative Decomposition Multi-objective Improved Elephant Clan Optimization Algorithm (CoDMOIECO). Specifically, a novel subpopulation construction method with adaptive changes following the number of iterations and a novel individual merit ranking based on NBI and angle are proposed., enabling the creation of subpopulations closely linked to weight vectors and the identification of diverse individuals within them. Additionally, new update strategies for the clan leader, male elephants, and juvenile elephants are introduced to boost individual exploitation capabilities and further enhance the algorithm's convergence. Finally, a new CoD-based environmental selection method is proposed, introducing adaptive dynamically adjusted angle coefficients and individual angles on corresponding weight vectors, significantly improving both the convergence and distribution of the algorithm. Experimental comparisons on the ZDT, DTLZ, and WFG function sets with four benchmark multi-objective algorithms—MOEA/D, CAMOEA, VaEA, and MOEA/D-UR—demonstrate that CoDMOIECO achieves superior performance in both convergence and distribution.

KEYWORDS: Multi-objective optimization; elephant clan optimization algorithm; collaborative decomposition; new individual selection mechanism; diversity preservation

1 Introduction

Multi-objective Optimization Problems (MOP) refer to optimization problems involving more than two conflicting objective functions. Assuming the optimization problem is a minimization problem, MOP is typically represented as:

$$\begin{aligned} &minmize \quad F(x) = (f_1(x), \ldots, f_M(x))^T \\ &subject\,to \quad x \in \Omega \end{aligned} \tag{1}$$

where M represents the objective variables, D is the number of decision variables, $F = (f_1, \ldots, f_M)^T$ is the objective vector, $x = (x_1, \ldots, x_D)$ is the decision vector, and Ω is the D-dimensional decision space. Multi-objective algorithms can generally be divided into three main categories based on the techniques they use to handle multiple objectives: dominance-based multi-objective optimization algorithms, performance indicator-based multi-objective algorithms, and decomposition-based multi-objective optimization algorithms. A brief overview of representative methods for each category is provided below.

Dominance-based multi-objective algorithms utilize Pareto-based fitness allocation strategies to identify all non-dominated individuals from the current evolutionary population. Murata et al. [1] introduced a new Multi-objective Genetic Algorithm (MOGA). During the execution of the MOGA algorithm, a Pareto optimal solution set is maintained, where a certain number of individuals are preserved as elites and passed to the next generation. Zitzler et al. [2] introduced SPEA2, which includes fine-grained fitness assignment, density estimation, and a novel archive truncation method to enhance the balance and diversity of the non-dominated solution set, achieving a more optimal Pareto front. Deb et al. [3] proposed NSGA-II, which reduces computational complexity through fast non-dominated sorting and us-es crowding distance and elitism to maintain population diversity. It optimizes the population by using non-dominated rank and crowding distance, ensuring the generation of Pareto optimal solutions. Erickson et al. [4] proposed NPGA, which explores global information about the Pareto optimal front through population diversity and continuously improves individual fitness through evolutionary operators, thereby obtaining a set of well-balanced optimal solutions. Yicun Hua et al. [5] introduced a clustering-based adaptive MOEA (CAMOEA) aimed at solving multi-objective problems (MOPs) with irregular Pareto fronts. The core concept of CAMOEA is to adaptively generate a set of cluster centers that guide the selection process in each generation, promoting diversity while accelerating convergence. Gao et al. [6] proposed an algorithm for adaptively adjusting the balancing tendency, which is based on efficient non-dominated sorting to improve the distribution of the population. However, this type of approach generally lacks a robust diversity maintenance mechanism, often leading to poor distribution of the non-dominated solution set. Additionally, as the dimensionality continues to increase, the dominance relationship becomes unclear, selection pressure decreases, and the problem of insufficient selection pressure reappears, meaning this issue has not been fundamentally resolved.

Performance indicator-based multi-objective algorithms generally rely on performance metrics to guide the search process and select optimal solutions. Representative algorithm developments are as follows: Wang et al. proposed the hypervolume Newton method for solving unconstrained bi-objective optimization problems with objective functions [7]. For a given MOP, a deterministic numerical optimization is carried out by the Newton-Raphson method to maximize the hypervolume index. Phan et al. [8] proposed and evaluated an indicator-based EMOA (R2-IBEA), which uses a binary R2 indicator. This indicator adaptively adjusts the positions of reference points according to the distribution of individuals in the current generation within the objective space, helping to determine the dominant relationship between any two given individuals. Li et al. [9] proposed a multi-indicator algorithm based on stochastic ranking (SRA), which employs stochastic ranking techniques to balance the search biases of different indicators. Zhengping Liang et al. proposed a new MaOEA based on a boundary protection indicator (MaOEA-IBP) [10], developing a worst-case elimination mechanism grounded in the indicator and boundary protection strategy to improve the equilibrium of population convergence, diversity, and coverage. Peng et al. proposed a MaOEA (A Many-Objective Evolutionary Algorithm Based on Dual Selection Strategy, MaOEA/DS) [11]. A new distance function is designed as a diversity index, and a point congestion strategy based on the distance function is proposed to further enhance the ability of the algorithm to distinguish the optimal solutions overall.

The strength of performance indicator-based multi-objective evolutionary algorithms lies in their independence from Pareto dominance for exploring the objective space. However, their drawback is the high computational cost, which severely limits the application of these algorithms in high-dimensional objective spaces. Zhang et al. [12] introduced a decomposition-based multi-objective evolutionary algorithm (MOEA/D), which converts multi-objective problems into a set of single-objective subproblems using weight vectors. This approach significantly reduced the computational complexity of existing multi-objective algorithms and demonstrated good convergence, attracting considerable attention from scholars. Hui Li et al. [13] proposed a new implementation of MOEA/D based on DE operators and polynomial mutation (MOEA/D-DE), introducing two additional strategies to maintain population diversity. Ke et al. [14] combined MOEA/D with ant colony optimization (MOEA/D-ACO), using the positive feedback mechanism of pheromones to gradually converge toward high-quality solutions. In 2017, Xiang et al. [15] proposed a new MaOEA algorithm based on vector angles, VaEA, which achieves a balance between convergence and diversity. De Farias et al. [16] introduced MOEA/D-UR, a MOEA/D variant that updates as needed. This method employs an improvement detection metric to decide when to adjust the weight vectors while partitioning the objective space to enhance diversity. Wang et al. proposed a metric-based reference vector-adjusted MOEAs for the inconsistency between the reference vector distribution and the PF shape in MOEA/D [17]. It can be seen that decomposition-based MOEAs generally achieve better convergence when solving multi-objective problems. However, the weight vector setting makes these algorithms sensitive to the shape of the Pareto front, and their alignment with the true front can still be improved. Compared to the other two approaches, decomposition-based methods are currently the most widely utilized technique for solving multi-objective problems.

With the advancement of science and technology, various multi-objective problems are encountered in real life, such as robot path planning, industrial scheduling, and knapsack problems. Through mathematical analysis, these can be viewed as multi-objective optimization problems. Multi-objective optimization algorithms are currently an effective approach to solving these types of problems, comprising two aspects: first, the core evolutionary algorithm, and second, the multi-objective handling techniques. The former iteratively evolves to obtain optimal solutions, while the latter preserves these optimal solutions. In light of this, to further enhance the problem-solving capability of multi-objective optimization algorithms in engineering problems, this study focuses on two aspects: the core evolutionary strategy and the multi-objective framework, as outlined below: (1) Algorithmic evolution strategies can effectively solve multi-objective problems, so the merits of algorithmic evolution strategies are crucial to the optimization effect of multi-objective problems. Extensive experimental research has shown that compared to various evolutionary methods such as Genetic Algorithm and Differential Algorithm, the Elephant Clan Optimization Algorithm [18] exhibits certain advantages. Therefore, it will serve as the core evolutionary strategy for the multi-objective optimization algorithm. (2) Currently, multi-objective algorithms generally adopt decomposition-based multi-objective handling techniques. Experimental research indicates that in 2022, Wu et al. [19] proposed a Collaborative Decomposition (CoD) environmental selection method, which combines the advantages of the penalty-based boundary intersection (PBI) and normal boundary intersection (NBI) methods. This allows the multi-objective algorithms formed by CoD to achieve good convergence and distribution. Therefore, this paper selects CoD-based environmental selection and aims to improve it based on the characteristics of multi-objective optimization, constructing a Collaborative Decomposition Multi-objective Elephant Clan Optimization Algorithm with the Elephant Clan Optimization Algorithm as the core evolutionary strategy.

2 Elephant Clan Optimization Algorithm

In nature, elephant populations are organized into male herds and family groups led by matriarchs, who use their learning and memory abilities to find suitable survival resources and undertake long-distance migrations. Inspired by these behaviors, Jafari et al. proposed the Elephant Clan Optimization (ECO) algorithm in 2021. In the ECO algorithm, each individual represents the position of an elephant within the clan, and the fitness value of the algorithm represents the survival resources available at that position. The algorithm is divided into three stages: initialization and population division, individual updating of the population, and individual replacement within the population. The pseudocode for the ECO algorithm is shown in Algorithm 1.

Algorithm 1: ECO

Input: optimisation problem dimension: *D*; number of communities: *Nc;* number of members per community: *Ne;* number of mates available: *Nm;* scaling factors: α, β; maximum number of iterations: *Maxiter*
Output: optimal solution and its fitness value

---

01. Initialize of variables (*Nd, Nc, Ne, Nm,* α, β, *Maxiter*).
02. Individual elephant $X_i{}^0$ positions were initialised according to Eq. (2) and the herd was divided into Nc-1 family herds as well as 1 male elephant herd.

$$X_i{}^0(j) = X^{\min}(j) + rand \times \left(X^{\max}(j) - X^{\min}(j)\right) \tag{2}$$

03. Calculate the fitness value of each individual in each colony (each colony contains a total of $Ne = N/Nc$ individual elephant)
04. **While** *it* < = *Maxiter* **do**
05. In each family group, the individual elephant with the best fitness value is the matriarch of the family group updated according to Eq. (3), and the other individual elephants are members of the group updated according to Eq. (4).

$$X_{FCi,M}^{it+1}(j) = X_{FCi,M}^{it}(j) + r \times \beta \times \left[X_{Best}^{it}(j) - X_{FCi,M}^{it}(j)\right] \tag{3}$$
$$X_{FCi,m}^{it+1}(j) = X_{FCi,m}^{it}(j) + r \times \alpha \times \left[X_{FCi,M}^{it}(j) - X_{FCi,m}^{it}(j)\right] \tag{4}$$

where $X_{FCi,M}^{it}$ and $X_{FCi,m}^{it}$ represent, respectively, the matriarch of the female elephant in the i $(i = 1 \cdots Nc - 1)$th family herd X_{FCi} and the m $(m = 1 \cdots Ne)$th herd member after sorting at iteration; *r* is a random number uniformly distributed over a range of $[0, 1]$; $X_{Best}^{it}(j)$ represents the jth dimension of the optimal position obtained by the entire population in the it generation; $\alpha = \beta = 2$.
06. Updating the position of each elephant in the male elephant population according to Eq. (5).

$$p = 1 - \left(c \times \frac{it}{it_{\max}}\right)$$

$$X_{MC,n}^{it+1}(j) = X_{MC,n}^{it}(j) + r_1 \times p \times \left[X^{\max}(j) - X^{\min}(j)\right] \tag{5}$$

where $X_{MC,n}^{it+1}(j)$ is the *j*th dimension of the nth $(n = 1 \cdots Ne)$ elephant in the male elephant population X_{MC} of the it+1th iteration; r1 is a random number uniformly distributed within $[-1, 1]$; $it_{\max}$ and it represent the maximum number of iterations and the current number of iterations; and *c* is 0.5.
07. Calculate the fitness value for each new elephant
08. Sorting the elephants in each herd
09. **For** *i* = 1:*Nc*-1**do**

(Continued)

Algorithm 1 (continued)

10. Determine the worst elephant, the adult elephant that left, and the mateable elephant in the ith family group: each family group contains both mateable and adult elephants. The mateable elephant is the Nm = 3 elephant with the best fitness value in this family group, while the adult elephant is the one that left its family group and was randomly selected from the remaining Ne-Nm-1 elephants except for the mateable elephant and the one with the worst fitness value.
11. Eq. (6) produces 2 new individuals to replace the worst elephant in the family group and the adult elephant that left, respectively.

$$X_{FCi,Calf}^{it+1}(j) = r \times \left(\frac{X_{FCi,Rf}^{it+1}(j) + X_{MC,Rm}^{it+1}(j)}{2} \right) \tag{6}$$

where $X_{FCi,Calf}^{it+1}$ denotes a newly generated baby elephant from the *it*+1st iteration of family colony X_{FCi}^{it+1}, $X_{FCi,Rf}^{it+1}$ $(Rf \in [1, \ldots, Nm])$ denotes a randomly selected individual from the it+1st iteration of mateable elephants, and $X_{MC,Rm}^{it+1}$ denotes the best elephant from the *it*+1st iteration of male elephant colony X_{MC}.
12. Updating the worst elephant in the male elephant population according to Eq. (7).

$$X_{MC,worst}^{it+1} = \begin{cases} X_{FCi,Gm}^{it+1}, & if\ f\left(X_{FCi,Gm}^{it+1}\right) < f\left(X_{MC,worst}^{it+1}\right) \\ X_{MC,worst}^{it+1}, & else \end{cases} \tag{7}$$

where $X_{MC,worst}^{it+1}$ is the worst elephant in the male herd in it+1st iteration, and $X_{FCi,Gm}^{it+1}$ is the adult elephant that left the family herd X_{FCi}^{it+1} in *it*+1st iteration.
13. **End For**
14. **End While**
15. Output the global optimal solution

3 Multi-Objective Elephant Clan Optimization Algorithm

To enhance the convergence and distribution of multi-objective algorithms, this section proposes the Collaborative Decomposition Multi-objective Improved Elephant Clan Optimization Algorithm (CoD-MOIECO) by using the ECO algorithm introduced in Section 2 as the core evolutionary strategy and employing the Collaborative De-composition (CoD) environmental selection method [19]. This approach integrates the benefits of both the penalty-based boundary intersection (PBI) and normal boundary intersection (NBI) methods as a technique for handling multi-objective optimization. The pseudocode for CoDMOIECO is shown in Algorithm 2.

Algorithm 2: CoDMOIECO

Input: *N, D, M, U, L, it, it_max*
Output: Pareto optimal solution set

01 Initialize parameters, including *N,D,M,U,L,it,it_max*
02 Randomly generate populations in a defined domain$X$
03 The two-layer generation scheme in NSGAIII is used to generate a set of weight vectors uniformly distributed on the normalized hyperplane W
04 The rotation coefficient r is generated according to Algorithm 3
05 Calculate the target value f(X) for each individual and obtain the ideal point Z^*

(Continued)

Algorithm 2 (continued)

While $it < it\_max$ **do**
The Angle between each individual and each W is calculated according to Eq. (9), and each individual is associated with the W with the smallest Angle
The subpopulation corresponding to the individual Xi associated weight vector is determined according to the method in Section 3.1 (1)
The individuals within the subpopulation are sorted according to the sorting method in Section 3.1 (2)
For $i = 1\colon N$ **do**
Update individuals according to Section 3.2 to generate new individuals
The newly generated individuals enter the new population, and the target value of the new individuals is calculated, and the ideal point Z^* is updated
End For
Merge the new population with the old population to form a new population U
The population U is non-dominated
Move individuals from successive non-dominant hierarchies into population S until S is no less than N
Normalize all individuals in S and associate each individual in S with the corresponding subproblem in W
Sort all individuals associated with each subproblem based on the g^{NCoD} sorting strategy calculated according to Eq. (18) in Section 3.3
All individuals at the ith position associated with the corresponding subproblem form the ith $NCOD$-based R_i
While $|X| + |R_i| < N$ **do**
$X \leftarrow X \cup R_i$
$i \leftarrow i + 1$
End While
The remaining $X - |P|$ individuals were randomly selected from the final R_i to form a set $R_i^{'}$
$X \leftarrow X \cup R_i^{'}$
End While
Output Pareto optimal solution set

Algorithm 3 offers a comprehensive description of the process for constructing the rotation coefficient r based on the weight vector. In step 5 of Algorithm 3, α_i represents the boundary-aware rotation factor of W^i, while β_i denotes the vertex-aware rotation factor at position W^i. Here, m is the number of objectives.

Algorithm 3: $r(W)$

01 $r \leftarrow \varnothing$
02 **For** each of these weight vectors w^i **do**
03 $\alpha_i \leftarrow 1 - min_{j=1}^{m}\left\{w_j^i\right\} \times m$
04 $\beta_i = \left(1 - max_{j=1}^{m}\left\{w_j^i\right\}\right) \times 2$
05 $r_i = \frac{\alpha_i + \beta_i}{2}$
06 $r \leftarrow r \cup \{r_i\}$
07 **End For**
08 Output r

3.1 New Individual Selection Strategy

To enable the CoDMOIECO algorithm to adapt to solving multi-objective optimization problems, an individual selection method has been designed that includes subpopulation construction and a ranking method for individuals based on NBI and angles. The specific details are as follows.

3.1.1 Subpopulation Construction

Typically, multi-objective optimization problems are solved using a decomposition approach. In this process, related individuals from single-objective problems form a cohesive solution through mutual learning and communication. Since the optimal solutions of each subproblem differ, the individuals guided by them naturally carry evolutionary information in different directions. It is noteworthy that due to the stability of the non-dominated Pareto front, subproblems corresponding to neighboring weights exhibit some similarities in their optimal solutions, leading to significant overlap in the evolutionary information contained in the related individuals. Compared to individuals with distinct evolutionary tendencies or information, learning among these similar individuals can more effectively obtain optimized solutions for the corresponding single-objective problems.

In light of this, this section no longer adopts the population-based evolutionary strategy from the ECO algorithm. Instead, specific subpopulations are created for each weight vector to promote the collaborative evolution of related individuals. Specifically, we first determine the neighborhood range of the weight vector using Eq. (8). Then, by computing the Euclidean distances between the weight vectors, we select the T weight vectors that are closest to the current weight vector to form its neighborhood. Finally, each weight vector selects individuals associated with its neighboring weight vectors to form a subpopulation.

$$T = max\left(N \times \left(0.5 - it/it_{max}\right), 5\right) \tag{8}$$

Here, N represents the number of individuals, while it and it_{max} denote the current iteration number and the maximum iteration number, respectively.

The Eq. (8) presented in this section modifies the conventional approach of using a constant neighborhood size, allowing it to adaptively change with the iteration count. In the initial stages of the algorithm's evolution, the evolutionary levels among individuals differ significantly, and all are relatively distant from the optimal Pareto front. A larger number of neighboring individuals provide optimal individuals with more options, facilitating the use of evolutionary information from superior individuals in adjacent subproblems. This helps to rapidly narrow the evolutionary gap between themselves and other subproblems, thus accelerating convergence toward the optimal Pareto front. In the later stages of the algorithm, as all individuals are closer to the optimal Pareto front, reducing the number of neighboring individuals allows for a more precise selection of optimal individuals. This more accurately guides the evolution within the neighborhood, fostering a good distribution of solutions and effectively conserving computational resources.

3.1.2 Individual Ranking Based on NBI and Angles

To enhance the performance of the CoDMOIECO algorithm, which uses the ECO algorithm as its core evolutionary strategy, it is necessary to select the leader individual, male individuals, and juvenile individuals from the constructed subpopulations according to the individual update mechanism of the ECO algorithm.

Since the subpopulation construction is based on the weight vectors, the aggregation function values of the individuals within the subpopulation are utilized to assess their quality. The Chebyshev function of the NBI type can better measure the convergence of individuals, while the VaEA algorithm utilizes angles to assess the distribution of individuals. Therefore, this section proposes an individual evaluation method based on NBI and angles: first, calculate gNBI, then compute the angle using Eq. (9); next, calculate the aggregation function value using Eq. (10); finally, rank the individuals in each subpopulation based on the aggregation function values, with smaller aggregation function values indicating superior individuals.

The following formula calculates the angle between each individual X in the subpopulation and each weight vector W^i. Here, $F(X) = f(X) - Z^*$ represents the specific angle measurement.

$$angle\left(F\left(X\right), W^i, Z*\right) = \arccos\left|\sum_{m=1}^{M}\left(f_m\left(X\right) - Z_m^*\right) \times W^i / \left\|f\left(X\right) - Z^*\right\| \times \left\|W^i\right\|\right| \tag{9}$$

$$sort\left(X\right) = g^{NBI}\left(X\left|W^i\right.\right) + \delta \times angle\left(F\left(X\right), W^i, Z^*\right) \tag{10}$$

Here, $g^{NBI}\left(X|W^i\right) = \max_{j=1}^{m}\left\{f_j\left(x|w^i\right) - w_j^i\right\}$, δ represents the weighting factor, as shown in Eq. (11).

$$\delta = 2 + \left(6 \times it/it_{\max}\right) \tag{11}$$

The individual ranking method based on NBI and angles introduced in this section offers several advantages. On one hand, a smaller gNBI value for individuals on the same weight vector indicates that the individuals are closer to the Pareto front, suggesting better convergence; conversely, the larger the angle between individuals and the same weight vector, the more dispersed the individuals are, indicating better distribution. Thus, the aggregation function, as shown in Eq. (9), comprehensively reflects both convergence and distribution. On the other hand, the weighting factor δ increases with iterations, enhancing the effect of the angle. In the initial stages of evolution, since individuals are typically far from the ideal point, the algorithm should focus more on convergence. In the later stages of evolution, as individuals gradually approach the ideal point, the algorithm needs to emphasize distribution more. In the early stages of evolution, since the value of $angle\left(F\left(X\right), W^i, Z^*\right)$ ranges from [0, $\pi/2$], gNBI is likely to be much greater than $angle\left(F\left(X\right), W^i, Z^*\right)$. The initial setting of δ balances gNBI and $angle\left(F\left(X\right), W^i, Z^*\right)$. This section improves overall performance by reasonably adjusting the relationship between δ, gNBI, and $angle\left(F\left(X\right), W^i, Z^*\right)$ to satisfy the algorithm's requirements for convergence and distribution at various stages.

3.2 New Update Methods for Leaders, Male Elephants, and Juvenile Elephants

As mentioned in Section 2, the population individual updates in the ECO algorithm are divided into three methods: updating the matriarch leader individuals, updating the male elephant individuals, and updating the juvenile elephant individuals. In the CoDMOIECO algorithm, as introduced in Section 3.1, each individual to be updated will find its corresponding subpopulation. Within this subpopulation, individuals will be ranked based on their performance, and the updates will be categorized into three types: updating the leader individual at the optimal position, updating the male elephant individual at the worst position, and updating the juvenile elephant individuals at other positions. The specific update methods are as follows.

3.2.1 Leader Individual Update

According to the description in Section 3.1, in the CoDMOIECO algorithm, each weight vector selects a leader individual from the corresponding subpopulation. Generally, in the early stages of evolution, even though the leader individual possesses the best evolutionary information in the subpopulation, the information is not optimal because the individual is far from the theoretical Pareto front. In the later stages of evolution, as individuals gradually approach the theoretical Pareto front, the evolutionary information of the leader individual also approaches the theoretical optimal value. Although the evolutionary information of other individuals may be inferior to that of the leader individual, it is still similar. The exchange of information between these individuals can potentially generate superior evolutionary information; however, excessive communication may divert the leader individual from its original evolutionary direction, thereby reducing the algorithm's convergence speed. As the optimal individual within the subpopulation, the lead individual needs to enhance its coverage of the search space while improving the effectiveness of the evolutionary information. Based on this analysis, the new update method for the leader individual is shown in Algorithm 4.

Algorithm 4: New patriarch individual renewal method

According to Eq. (12), calculate the number of dimensions a that the individual to be updated should evolve

$$a = randperm\left(D, round\left((D/2) \times \left(exp\left(-((5 \times it)/it_{max})^2\right)\right)\right) + 2\right) \tag{12}$$

a dimensions are randomly selected to form a dimension set to be updated $rand\_D$
If the individual to be updated is the patriarch individual **do**
  If $rand <= 0.5$ **do**
    A simulated binary crossover with a distribution index of 30 with a probability of 1.0 and a polynomial mutation with a distribution index of 30 with a probability of 1.0 were used to update the patriarch individuals
  Else If $rand > 0.5$ && $rand <= 0.6$ **do**
    The patriarch individual is updated according to the opposite learning mode of Eq. (13).

$$X_j(it+1) = U + L - X_{i,j}(it) \tag{13}$$

  Else
    If $j \in rand\_D$ **do**
      The jth dimension of the patriarch is updated according to Eq. (14)

$$X_j(it+1) = X_j(it) + r \times \left(X_j(it) - r\_X_j\right) \tag{14}$$

  End
End

In the initial stages of evolution, it is advisable to retain a few characteristics of the leader individual while exchanging the majority of characteristics with another randomly selected individual from the population. Conversely, in the later stages of evolution, more characteristics of the leader individual should be preserved, as shown in Eq. (14). During the evolutionary process, a certain probability should be applied to introduce simulated binary crossover to enhance the leader individual's coverage of the search space. This enables the leader individual to guide other members in exploring the potential solution space more thoroughly. As a guide for the individuals in the subpopulation, the leader individual must preserve the effectiveness of its evolutionary information to ensure the successful evolution of the population. The opposition learning strategy typically validates the solutions generated by checking for any significant opposition between two solutions. This helps filter out unreasonable solutions while retaining those that are valid.

3.2.2 Leader Individual Update

In the ECO algorithm, the primary goal of the male elephant individuals is to generate superior evolutionary information while maintaining population diversity. In the realm of multi-objective optimization, each individual typically corresponds to a solution for a problem, necessitating the acceleration of evolution for the male elephant individuals within the subpopulation. As noted, the male elephant individuals occupy the last position in the current subpopulation. Similar to the leader individual updates, in the initial stages of evolution, a few characteristics of the male elephant individuals should be replaced while the majority of characteristics are exchanged with the leader center individual. Conversely, in the later stages of evolution, more characteristics of the male elephant individuals should be preserved. When performing characteristic exchanges, as shown in Eq. (15), this significantly enhances the effectiveness of the male elephant individuals' learning from superior individuals, accelerating the algorithm's convergence speed while maintaining diversity. For other characteristics of the male elephants, they can be randomly replaced by the leader in-dividual of their subpopulation with a certain probability, as indicated in Eqs. (16) and (17). Random replacement helps maintain the diversity of the male elephant individuals, while direct replacement can improve their convergence. The new update method for the male elephant individuals is detailed in Algorithm 5.

Algorithm 5: New male regeneration patterns

01 According to Eq. (12a), calculate the number of dimensions a that the individual to be updated should evolve

$$a = randperm\left(D, round\left((D/2) \times \left(exp\left(-\left((5 \times it)/it_{max}\right)^2\right)\right) + 2\right)\right) \tag{12a}$$

02 a dimension is randomly selected to form a dimension set $rand\_D$ to be updated
03 **If** the individual to be renewed is the male **do**
04 **If** $j \in rand\_D$ **do**
05 The jth dimension of male elephant is updated according to Eq. (15)

$$X_j(it+1) = \left(X_j(it) + best\_X_j(it)\right)/2 + r \times p \times \left(best\_center\_X_j(it) - X_j(it)\right) \tag{15}$$

06 **Else**
07 **If** $rand < 0.2$ **do**
08 The jth dimension of the male elephant is replaced according to Eq. (16)

$$X_j(it+1) = r \times best\_X_j(it) \tag{16}$$

09 **Else**
10 The ith dimension of the male elephant is replaced directly according to Eq. (17).

$$X_j(it+1) = best\_X_j(it) \tag{17}$$

11 **End**
12 **End**

3.2.3 Update of the Juvenile Elephant Individuals

The new update method for the juvenile elephant individuals is detailed in Algorithm 6.

In CoDMOIECO, the juvenile elephant individuals need to accelerate the convergence speed while maintaining a certain level of diversity, providing more evolutionary information for communication with the leader individuals. Therefore, the individual features that require communication, as shown in Eq. (18), evolve toward the optimal individual in the subpopulation to enhance their convergence speed while also learning from a randomly selected individual in the entire population to preserve diversity. For the individual features that do not require communication, they are replaced by the leader individual in the subpopulation with a certain probability, thereby speeding up convergence and increasing the diversity of the evolutionary information.

Algorithm 6: New ways of individual calf renewal

---

01 According to Eq. (12b), calculate the number of dimensions a that the individual to be updated should evolve

$$a = randperm\left(D, round\left((D/2) \times \left(exp\left(-\left((5 \times it)/it_{max}\right)^2\right)\right)\right) + 2\right) \tag{12b}$$

02 Randomly select a dimensions to form the set $rand\_D$ of dimensions to be updated 21 **If** the individual to be renewed is a juvenile elephant

03 **If** $j \in rand\_D$ **do**

04 According to Eq. (18), the jth dimension of the juvenile elephant was updated.

$$X_j(it+1) = X_j(it) + r \times \alpha \times \left(best\_X_j(it) - X_j(it)\right) + r \times \beta \times \left(r\_X_j - X_j(it)\right) \tag{18}$$

05 **Else**

06 **If** $rand < 0.2$ **do**

07 The jth dimension of the juvenile individual is replaced by the jth dimension of the best individual within the subpopulation

08 **End**

09 **End**

As mentioned above, with the evolution of individuals, the updating strategies introduced in this section gradually reduce the features updated for each individual. This aligns with the evolutionary requirement of individuals approaching the optimal front and conducting a more refined search. At the same time, different updating methods are designed for various individuals, ensuring that different information between individuals can be fully communicated. This, to some extent, accelerates the algorithm's convergence speed toward the front. Additionally, through methods such as SBX, oppositional learning, random individual learning, and communication, as well as replacement strategies, new excellent evolutionary information is provided for each subpopulation, minimizing the likelihood of the algorithm becoming trapped in local optima during the evolutionary process.

3.3 New CoD-Based Environmental Selection Method

The PBI type of Chebyshev function is better able to maintain population diversity, and the specific aggregation function is shown in Eq. (19).

$$g^{PBI}\left(x|w^i\right) = d_1\left(f(x), w^i\right) + \theta \times d_2\left(f(x), w^i\right) \tag{19}$$

where $d_1\left(f\left(x\right),w^i\right)$ denotes the projection length of the target vector $f\ (x)$ under the reference vector w^i, the constant θ is used to balance convergence and distribution, and $d_2\left(f\left(x\right),w^i\right)$ denotes the perpendicular distance from the target vector $f\ (x)$ to the reference vector w^i.

While PBI can maintain population diversity, it is unable to obtain uniform boundaries due to the radial spatial distribution of its reference lines, and while NBI can maintain population convergence, its parallel reference lines may result in poorer diversity due to undersampling near the boundaries. The CoD environment selection method, a synergistic decomposition method of PBI and NBI, inherits the advantages of the two methods in an integrated way. Therefore, this paper addresses the improvement of the CoD environment selection method. The aggregation function is specifically shown in Eq. (20).

$$minimize\ g^{CoD}\left(X\middle|W^i\right)=g^{NBI}\left(X\middle|W^i\right)+r_i\times d_2\left(f\left(X\right),W^i\right) \tag{20}$$

where r_i denotes the rotation factor.

Upon further analysis, the NBI-type Tchebycheff function and vertical distance d_2 can only evaluate the convergence and distribution of individuals to a certain extent, and they do not effectively assess these aspects in a coordinated manner. It is well recognized that in multi-objective evolutionary algorithms, the emphasis shifts between convergence in the early stages and distribution in the later stages of evolution. However, the current methods do not adequately address the differing emphasis on convergence and distribution across these evolutionary phases.

In light of this, to effectively enhance the evaluation of convergence and distribution by the g^{CoD} aggregation function, a new *CoD* aggregation function is proposed, as shown in Eq. (21). This function combines the NBI-type Tchebycheff function with the angle of the individual's objective vector relative to the corresponding weight vector and the vertical distance.

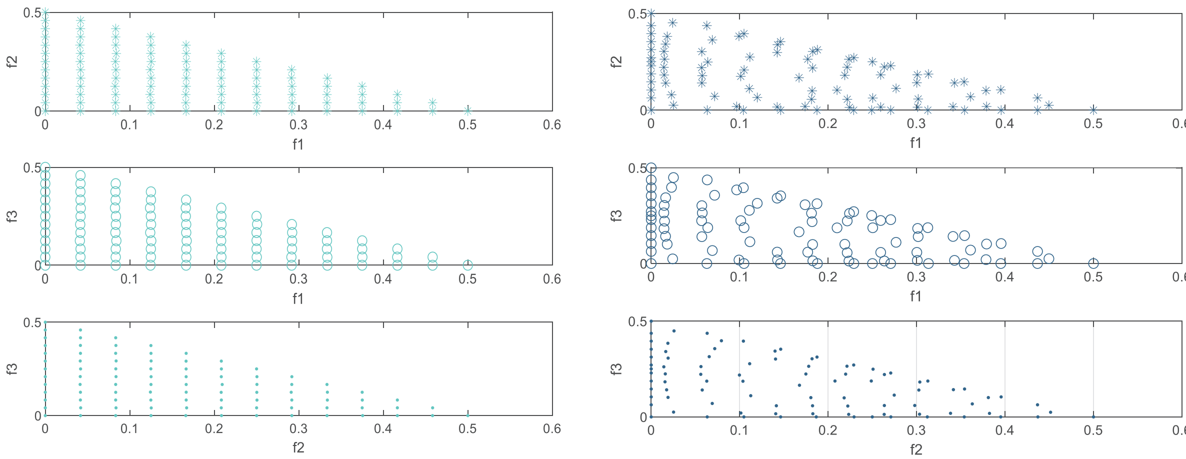
$$minimize\ g^{NCoD}\left(X\middle|W^i\right)=g^{NBI}\left(X\middle|W^i\right)+\varepsilon\times angle\left(F\left(X\right),W^i,Z^*\right)+r_i\times d_2\left(f\left(X\right),W^i\right) \tag{21}$$

The coefficient ε of the angle ranges from 1 to 1.5 and increases with the number of iterations.

As seen from the above formula, this paper introduces the angle of the individual's objective vector relative to the corresponding weight vector into the original CoD aggregation function. A smaller angle not only guides individuals to search toward the Pareto front along the associated weight vectors but also directs individuals' distribution on the Pareto front as they converge near it. The dynamic adjustment of the angle coefficient allows the algorithm to focus more on exploration or distribution at different stages. In the initial iterations, the coefficient is relatively small, emphasizing exploration and helping to discover new solution spaces. In the later iterations, the coefficient gradually increases, leading to a greater distribution of individuals near the Pareto front, thereby enhancing the algorithm's diversity and distribution and facilitating a more comprehensive exploration of the Pareto solution set. Therefore, the aggregation function proposed in this section, which combines the NBI-style Tchebycheff function, dynamic updates of the angle, and vertical distance, considers all three factors to offer a more accurate evaluation of an individual's convergence and distribution. This approach improves the performance and effectiveness of multi-objective optimization algorithms.

CoDMOIECO represents the optimization algorithm when the aggregation function proposed in this paper is g^{NcoD}, and CoDMOIECO1 represents the algorithm when the aggregation function is g^{coD}, and to ensure the fairness of the experiments, all parameters are set as shown in Section 4.1.

As can be seen from Fig. 1, the distribution of individuals in CoDMOIECO is significantly more dispersed than in ITLCO under the same conditions. This suggests that the CoDMOIECO proposed in this section can better balance the convergence speed and population diversity.



(a) CoDMOIECO (b) CoDMOIECO1

Figure 1: Scatter plot of different algorithms on DTLZ1

4 Experiments

To verify the performance of CoDMOIECO, it is compared with four currently outstanding algorithms on the DTLZ [20], WFG [21], and ZDT [22] test suites. In the DTLZ problems, the decision variable D = M + k – 1, where M represents the number of objectives, k is 5 for DTLZ1 and 10 for the others. In WFG, the decision variable D = k + l, where k = 2(M – 1) and l = 20. In ZDT, the decision variable is fixed at 30. The comparison algorithms include MOEA/D [12], CAMOEA [5], VaEA [16], and MOEA/D-UR [17].

4.1 Basic Parameter Settings

To ensure the fairness of the comparison experiments, for the ZDT problem with 2 objective functions, the population size N is set to 100, and the maximum number of function evaluations is 25,000. For the DTLZ and WFG problems with 3 objectives, the population size N is set to 91, and the maximum number of function evaluations is 45,500. Other parameter settings for each algorithm are shown in Table 1.

Table 1: Parameter settings for each algorithm

Algorithm	Parameter
MOEA/D	p_c = 1.0; xc = 20; p_m = 1/n; F = 0.5; T = 20; $delta$ = 0.9
VaEA	p_c = 1.0; p_m = 1/n; η_c = 30; η_m = 20
MOEA/D-UR	T = 0.1; δ = 0.9; nr = 2; k = 10
CAMOEA	p_c = 1.0; p_m = 1/n; η_c = 30; η_m = 20
CoDMOIECO	T_{min} = 10; p_c = 1.0; p_m = 1/n; η_c = 30; η_m = 20

4.2 Experimental Results and Analysis

The values of HV and IGD for the CoDMOIECO algorithm and the comparison algorithms on three test sets are shown in Tables 2 and 3. The mean and standard deviation of HV and IGD are indicated by "±," with the best results for each function highlighted in bold. Table 4 presents the Wilcoxon rank-sum test results between the CoDMOIECO algorithm and the comparison algorithms.

Table 2: HV results of each algorithm on the test sets

	M	MOEA/D Mean ± Std	CAMOEA Mean ± Std	VaEA Mean ± Std	MOEA/D-UR Mean ± Std	CoDMOIECO Mean ± Std
ZDT1	2	7.05E-01 ± 1.08E-02	7.19E-01 ± 2.02E-04	7.18E-01 ± 6.14E-04	7.18E-01 ±1.53E-03	**7.20E-01 ± 1.45E-09**
ZDT2	2	3.61E-01 ± 5.94E-02	4.44E-01 ± 2.09E-04	4.44E-01 ± 1.85E-04	4.43E-01 ±6.79E-04	**4.45E-01 ± 2.18E-10**
ZDT3	2	**6.20E-01 ± 3.93E-02**	5.99E-01 ± 4.48E-04	6.02E-01 ± 1.62E-02	5.99E-01 ±1.90E-03	5.99E-01 ± 3.12E-08
ZDT4	2	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	1.94E-03 ± 4.71E-03	**7.20E-01 ± 4.84E-09**
ZDT6	2	2.03E-01 ± 2.60E-02	1.99E-01 ± 2.15E-02	1.60E-01 ± 2.96E-02	3.01E-01 ± 1.32E-02	**3.89E-01 ± 7.66E-11**
DTLZ1	3	8.41E-01 ± 8.21E-04	8.35E-01 ± 2.08E-03	8.00E-01 ± 4.55E-02	8.35E-01 ± 1.57E-03	**8.42E-01 ± 7.88E-10**
DTLZ2	3	5.60E-01 ± 8.91E-06	5.48E-01 ± 2.28E-03	5.54E-01 ± 1.49E-03	5.58E-01 ± 7.45E-04	**5.60E-01 ± 9.64E-10**
DTLZ3	3	5.18E-01 ± 3.69E-02	5.14E-01 ± 9.76E-02	5.08E-01 ± 9.96E-02	5.36E-01 ± 1.03E-02	**5.59E-01 ± 4.21E-08**
DTLZ4	3	4.62E-01 ± 1.42E-01	5.49E-01 ± 1.51E-03	5.54E-01 ± 1.48E-03	4.95E-01 ± 1.32E-01	**5.60E-01 ± 1.35E-09**
DTLZ5	3	1.82E-01 ± 1.42E-05	1.99E-01 ± 2.03E-04	**1.99E-01 ± 1.58E-04**	1.98E-01 ± 2.11E-04	1.81E-01 ± 8.46E-06
DTLZ6	3	1.82E-01 ± 6.37E-06	2.00E-01 ± 1.05E-04	**2.00E-01 ± 7.42E-05**	1.98E-01 ± 4.77E-04	1.80E-01 ± 2.80E-07
DTLZ7	3	2.56E-01 ± 1.10E-03	**2.74E-01 ± 1.34E-03**	2.73E-01 ± 1.52E-02	2.67E-01 ± 2.09E-03	2.62E-01 ± 2.47E-05
WFG1	3	7.17E-01 ± 4.92E-02	8.13E-01 ± 2.48E-02	8.11E-01 ± 3.06E-02	**8.91E-01 ± 3.15E-02**	6.64E-01 ± 1.85E-03
WFG2	3	7.31E-01 ± 4.87E-02	8.64E-01 ± 6.65E-02	8.35E-01 ± 6.80E-02	8.30E-01 ± 6.81E-02	**9.20E-01 ± 1.01E-05**
WFG3	3	3.08E-01 ± 2.97E-02	3.41E-01 ± 8.10E-03	3.57E-01 ± 6.91E-03	3.84E-01 ± 1.00E-02	**3.89E-01 ± 2.97E-05**
WFG4	3	5.09E-01 ± 4.81E-03	5.14E-01 ± 4.61E-03	5.26E-01 ± 4.10E-03	5.38E-01 ± 2.23E-03	**5.38E-01 ± 1.29E-05**
WFG5	3	4.92E-01 ± 2.84E-03	4.90E-01 ± 3.28E-03	**5.03E-01 ± 2.86E-03**	4.97E-01 ± 2.10E-03	4.98E-01 ± 9.54E-06
WFG6	3	4.80E-01 ± 1.40E-02	4.91E-01 ± 4.86E-03	5.04E-01 ± 6.41E-03	**5.07E-01 ± 7.61E-03**	4.81E-01 ± 1.18E-07
WFG7	3	5.02E-01 ± 1.51E-02	5.34E-01 ± 3.10E-03	5.44E-01 ± 2.03E-03	5.44E-01 ± 4.65E-03	**5.46E-01 ± 5.03E-06**
WFG8	3	4.42E-01 ± 1.87E-02	4.33E-01 ± 4.55E-03	4.51E-01 ± 4.07E-03	4.62E-01 ± 4.08E-03	**4.68E-01 ± 7.42E-06**
WFG9	3	4.29E-01 ± 2.67E-02	4.70E-01 ± 1.98E-02	4.72E-01 ± 2.01E-02	**4.84E-01 ± 2.36E-02**	4.57E-01 ± 5.85E-06

Table 3: IGD results of each algorithm on the test sets

	M	MOEA/D Mean ± Std	CAMOEA Mean ± Std	VaEA Mean ± Std	MOEA/D-UR Mean ± Std	CoDMOIECO Mean ± Std
ZDT1	2	1.80E-02 ± 1.71E-02	4.60E-03 ± 1.18E-04	5.03E-03 ± 3.04E-04	5.50E-03 ± 1.33E-03	**3.56E-03 ± 1.85E-09**
ZDT2	2	7.09E-02 ± 6.74E-02	4.49E-03 ± 1.16E-04	4.33E-03 ± 8.19E-05	4.69E-03 ± 2.21E-04	**3.47E-03 ± 6.26E-09**
ZDT3	2	3.47E-02 ± 1.66E-02	**5.14E-03 ± 1.59E-04**	6.93E-03 ± 5.27E-03	1.09E-02 ± 2.10E-03	7.71E-03 ± 2.15E-07
ZDT4	2	2.79E+00 ± 7.23E-01	2.46E+00 ± 5.57E-01	3.13E+00 ± 7.31E-01	1.29E+00 ± 4.10E-01	**3.58E-03 ± 1.28E-08**
ZDT6	2	1.57E-01 ± 2.57E-02	1.75E-01 ± 2.51E-02	2.26E-01 ± 4.07E-02	6.96E-02 ± 1.11E-02	**3.16E-03 ± 3.99E-09**
DTLZ1	3	2.07E-02 ± 9.15E-05	2.27E-02 ± 5.11E-04	3.65E-02 ± 2.02E-02	2.31E-02 ± 7.44E-04	**2.66E-04 ± 5.19E-09**
DTLZ2	3	5.45E-02 ± 5.73E-07	6.01E-02 ± 8.95E-04	5.78E-02 ± 8.27E-04	5.61E-02 ± 4.81E-04	**1.16E-03 ± 6.64E-08**
DTLZ3	3	6.86E-02 ± 2.02E-02	9.82E-02 ± 1.74E-01	1.05E-01 ± 1.76E-01	6.96E-02 ± 3.41E-03	**4.39E-03 ± 7.81E-07**
DTLZ4	3	2.61E-01 ± 2.91E-01	6.03E-02 ± 1.09E-03	5.80E-02 ± 5.90E-04	1.84E-01 ± 2.65E-01	**1.61E-03 ± 1.03E-07**
DTLZ5	3	3.39E-02 ± 2.76E-05	5.64E-03 ± 1.72E-04	**5.50E-03 ± 1.39E-04**	7.35E-03 ± 3.65E-04	3.55E-02 ± 1.91E-05
DTLZ6	3	3.39E-02 ± 1.60E-05	**5.16E-03 ± 1.48E-04**	5.18E-03 ± 1.69E-04	7.60E-03 ± 7.64E-04	3.73E-02 ± 1.39E-06
DTLZ7	3	1.54E-01 ± 1.59E-03	**6.46E-02 ± 1.39E-03**	9.84E-02 ± 1.43E-01	1.01E-01 ± 5.39E-03	1.09E-01 ± 1.46E-04
WFG1	3	4.88E-01 ± 8.17E-02	3.27E-01 ± 4.10E-02	3.08E-01 ± 5.00E-02	**2.19E-01 ± 3.52E-02**	5.92E-01 ± 8.16E-03
WFG2	3	5.88E-01 ± 1.99E-01	3.13E-01 ± 1.51E-01	3.59E-01 ± 1.65E-01	3.86E-01 ± 1.57E-01	**2.42E-01 ± 2.02E-04**
WFG3	3	2.57E-01 ± 8.08E-02	1.69E-01 ±1.36E-02	1.46E-01 ± 1.29E-02	9.27E-02 ± 1.86E-02	**8.62E-02 ± 7.73E-05**
WFG4	3	2.66E-01 ± 5.31E-03	2.51E-01 ± 3.42E-03	2.49E-01 ± 4.21E-03	2.44E-01 ± 5.57E-03	**2.44E-01 ± 3.31E-04**
WFG5	3	2.53E-01 ± 2.83E-03	2.54E-01 ± 3.83E-03	2.43E-01 ± 3.33E-03	2.48E-01 ± 4.82E-03	**2.24E-01 ± 3.00E-04**
WFG6	3	2.86E-01 ± 1.29E-02	2.68E-01 ± 5.52E-03	**2.52E-01 ± 6.53E-03**	2.60E-01 ± 4.83E-02	2.93E-01 ± 4.85E-06
WFG7	3	3.04E-01 ± 2.38E-02	2.40E-01 ± 5.22E-03	**2.30E-01 ± 2.82E-03**	2.41E-01 ± 7.44E-03	2.69E-01 ± 2.17E-05
WFG8	3	3.21E-01 ± 2.77E-02	3.44E-01 ± 6.96E-03	3.14E-01 ± 5.29E-03	3.11E-01 ± 6.58E-03	**3.06E-01 ± 7.95E-05**
WFG9	3	3.22E-01 ± 3.91E-02	2.66E-01 ± 1.97E-02	**2.57E-01 ± 1.87E-02**	2.62E-01 ± 1.61E-02	2.98E-01 ± 8.82E-05

Table 4: Wilcoxon rank-sum test results for HV and IGD of CoDMOIECO and comparison algorithms on the test sets

Test function	M	HV				IGD			
		MOEA/D	VaEA	MOEA/D-UR	CAMOEA	MOEA/D	VaEA	MOEA/D-UR	CAMOEA
ZDT1	2	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)
ZDT2	2	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)
ZDT3	2	0.000(+)	0.000(+)	0.000(−)	0.000(−)	0.000(−)	0.000(+)	0.000(−)	0.000(+)
ZDT4	2	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)
ZDT6	2	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)
DTLZ1	3	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)
DTLZ2	3	0.428(=)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)
DTLZ3	3	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)
DTLZ4	3	0.079(=)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)
DTLZ5	3	0.157 (=)	0.000(+)	0.000(+)	0.000(+)	0.000(+)	0.000(+)	0.000(+)	0.000(+)
DTLZ6	3	0.000(+)	0.000(+)	0.000(+)	0.000(+)	0.000(+)	0.000(+)	0.000(+)	0.000(+)
DTLZ7	3	0.000(−)	0.000(+)	0.000(+)	0.000(+)	0.000(−)	0.000(+)	0.000(+)	0.000(+)
WFG1	3	0.000(+)	0.000(+)	0.000(+)	0.000(+)	0.000(+)	0.000(+)	0.000(+)	0.000(+)
WFG2	3	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.074(=)	0.186(=)
WFG3	3	0.000(−)	0.000(−)	0.429(=)	0.000(−)	0.000(−)	0.000(−)	0.830(=)	0.000(−)
WFG4	3	0.000(−)	0.000(−)	0.912(=)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.171(=)
WFG5	3	0.000(−)	0.000(+)	0.084(=)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)
WFG6	3	0.233(=)	0.000(+)	0.000(+)	0.000(+)	0.000(+)	0.000(+)	0.000(+)	0.000(+)
WFG7	3	0.000(−)	0.000(−)	0.492(=)	0.000(−)	0.000(−)	0.006(+)	0.000(+)	0.000(+)
WFG8	3	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)
WFG9	3	0.000(−)	0.000(+)	0.000(+)	0.841(=)	0.000(−)	0.000(+)	0.000(+)	0.000(+)
+/=/-		3/4/14	8/0/13	6/4/11	5/1/15	4/0/17	8/0/13	7/2/12	8/2/11

From Tables 2 and 4, it can be observed that in terms of the HV metric, the CoDMOIECO algorithm achieves the best performance compared to the comparison algorithms on ZDT1, ZDT2, ZDT4, ZDT6, DTLZ1–4, WFG2–4, WFG7, and WFG8. The VaEA algorithm demonstrates optimal performance on DTLZ5, DTLZ6, and WFG5. The MOEA/D algorithm and CAMOEA algorithm achieve the best results only on ZDT3 and DTLZ7, respectively. The MOEA/D-UR algorithm excels on WFG1, WFG6, and WFG9. In general evaluation, compared to the CoDMOIECO algorithm, the VaEA algorithm has an advantage on 8 functions but falls short on 13 functions. The CAMOEA algorithm performs better on 5 functions but is inferior to 15 functions. The MOEA/D-UR algorithm shows strong performance on 6 functions but is less effective on 11 functions. The MOEA/D algorithm exhibits a clear advantage on 3 functions but is outperformed on 14 functions.

According to Tables 3 and 4, in terms of the IGD metric, the CoDMOIECO algorithm achieves optimal results on ZDT1, ZDT2, ZDT4, ZDT6, DTLZ1–4, WFG2–5, and WFG8. The VaEA algorithm excels on DTLZ5–7 and WFG9. The CAMOEA algorithm performs best on ZDT3, DTLZ6, and DTLZ7. The MOEA/D-UR algorithm only achieves the best results on WFG1. The MOEA/D algorithm does not achieve the best results on any of the functions. Overall, compared to the CoDMOIECO algorithm, the MOEA/D algorithm performs better on 4 functions but is inferior on 17 functions. The VaEA algorithm has superior performance on 8 functions but is outperformed on 13 functions. The CAMOEA algorithm performs better on 8 functions but is less effective on 11 functions. The MOEA/D-UR algorithm excels on 7 functions but falls short on 12 functions.

As shown in Table 5, from the Friedman's test results for IGD and HV, it is clear that CoDMOIECO performs best on both IGD and HV compared to other algorithms.

To clearly illustrate the overall performance of the CoDMOIECO algorithm, Figs. 2–4 show the Pareto front graphs of all algorithms for three representative functions in the ZDT, DTLZ, and WFG test sets. By observation, it can be noted that in terms of convergence, the CoDMOIECO algorithm converges to the true front on ZDT1, DTLZ3, and WFG2. Regarding distribution, although there is still a certain gap between the

Pareto front obtained by CoDMOIECO and the true front for the WFG2 test function, it still demonstrates significant advantages in distribution compared to other comparative algorithms.

In summary, the CoDMOIECO algorithm exhibits greater advantages in both convergence and distribution compared to the benchmark algorithms. This advantage allows the CoDMOIECO algorithm to more effectively identify high-quality solutions that are near the true Pareto front when addressing multi-objective optimization problems, thereby improving the quality of the optimization outcomes.

Table 5: Friedman test results for HV and IGD of CoDMOIECO and comparison algorithms on the test sets

	MOEAD	**CAMOEA**	**VaEA**	**MOEADUR**	**CoDMOIECO**
HV	1.98	2.76	3.12	3.38	3.76
	5	4	3	2	1
IGD	4.14	2.9	2.76	2.83	2.36
	5	4	2	3	1

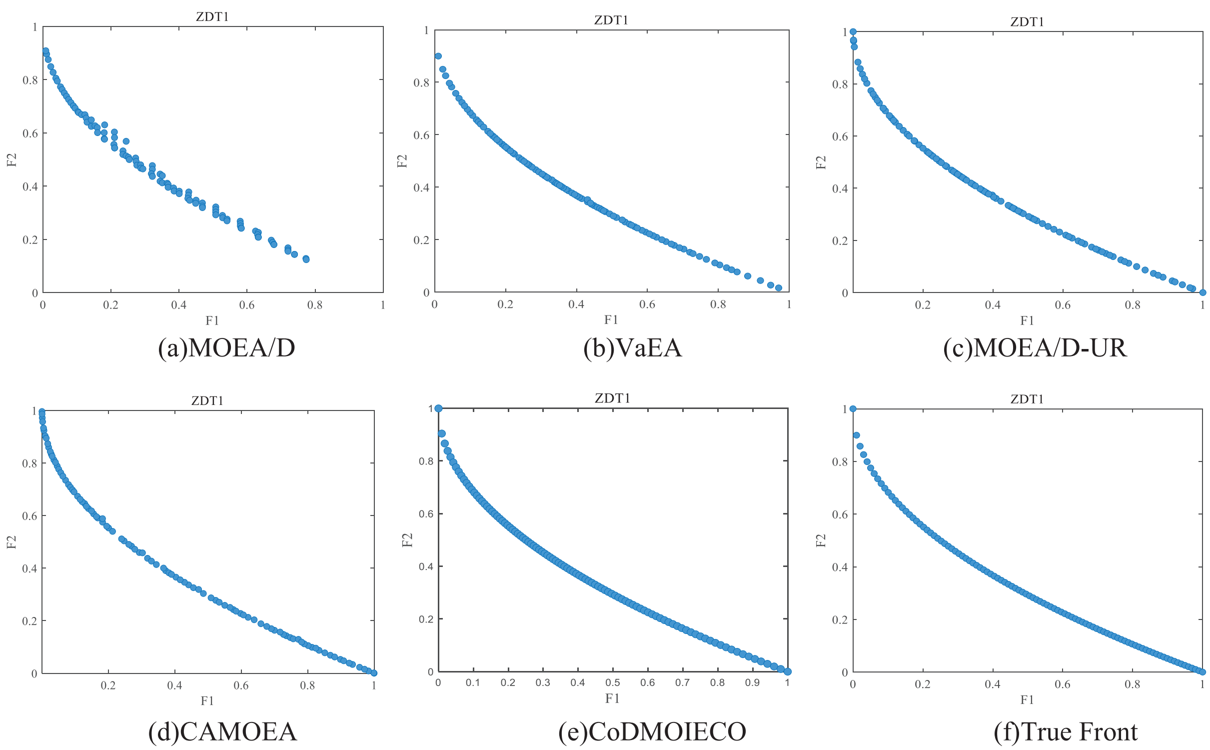


Figure 2: Pareto front graph of various algorithms on ZDT1

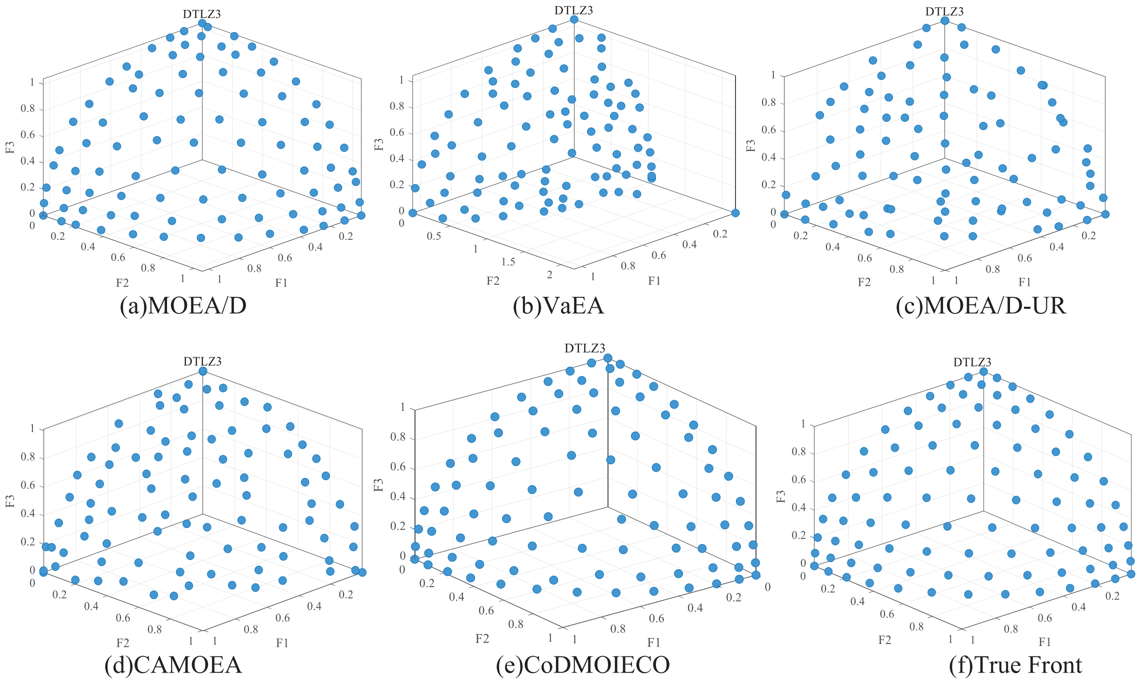


Figure 3: Pareto front graph of various algorithms on ZDTZ3

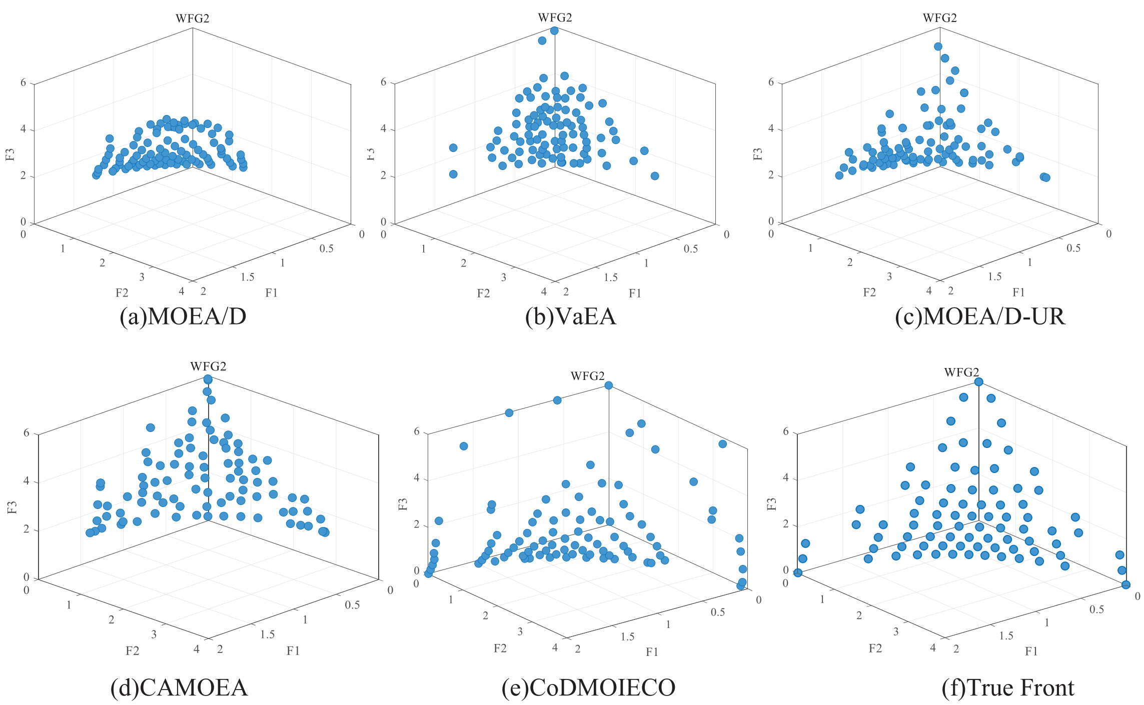


Figure 4: Pareto front graph of various algorithms on WFG2

5 Conclusions

To tackle more complex and practical optimization challenges and improve the performance of multi-objective optimization algorithms in addressing these issues, this paper integrates the elephant clan optimization algorithm with multi-objective processing techniques, proposing the CoDMOIECO algorithm. Based on the characteristics of multi-objective optimization problems, new individual selection methods, updated strategies for the leader, male, and juvenile elephants, and a new CoD-based environmental selection method are introduced. Experiments on the ZDT, DTLZ, and WFG benchmark function sets demonstrate that CoDMOIECO exhibits superior convergence and distribution compared to benchmark algorithms. However, as the number of optimization objectives increases, the time complexity increases, and the performance of the algorithm decreases. Therefore, reducing the complexity and improving the performance of algorithms in large-scale multi-objective optimization is an important part of the research. CoDMOIECO can be further extended and applied to the engineering field to solve optimization problems in real engineering and maximize the benefits. For example, fuzzy overlapping community detection problem, and optimization problem of edge computing.

Acknowledgement: The authors would like to thank the anonymous reviewers and the editor for their valuable suggestions, which greatly contributed to the improved quality of this article.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Mengjiao Wei, Wenyu Liu; data collection: Mengjiao Wei; analysis and interpretation of results: Mengjiao Wei, Wenyu Liu; draft manuscript preparation: Mengjiao Wei, Wenyu Liu. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The authors confirm that the data supporting the findings of this study are available at https://gitcode.com/open-source-toolkit/4265b (accessed on 05 September 2024).

Ethics Approval: Not applicable.

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

References

1. Murata T, Ishibuchi H. MOGA: multi-objective genetic algorithms. In: Proceedings of 1995 IEEE International Conference on Evolutionary Computation; 1995 Nov; Perth, Australia; p. 289–94.
2. Zitzler E, Laumanns M, Thiele L. SPEA2: improving the strength pareto evolutionary algorithm for multiobjective optimization. In: Proceedings of the EUROGEN; 2001 Sep; Athens, Greece; p. 95–100.
3. Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transact Evolu Comput. 2002 Apr;6(2):182–97. doi:10.1109/4235.996017.
4. Erickson M, Mayer A, Horn J. Multi-objective optimal design of groundwater remediation systems: application of the niched Pareto genetic algorithm (NPGA). Adv Water Resour. 2002 Jan;25(1):51–65. doi:10.1016/S0309-1708(01)00020-3.
5. Hua Y, Jin Y, Hao K. A clustering-based adaptive evolutionary algorithm for multiobjective optimization with irregular Pareto fronts. IEEE Trans Cybern. 2018 Jul;49(7):2758–70. doi:10.1109/TCYB.2018.2834466.
6. Gao X, He F, Zhang S, Luo J, Fan B. A fast nondominated sorting-based MOEA with convergence and diversity adjusted adaptively. J Supercomput. 2024 Jan;80(2):1426–63. doi:10.1007/s11227-023-05516-5.
7. Wang H, Emmerich M, Deutz A, Hernández VAS, Schütze O. The hypervolume newton method for constrained multi-objective optimization problems. Math Comput Appl. 2023 Jan;28(1):10. doi:10.3390/mca28010010.

8. Phan DH, Suzuki J. R2-IBEA: R2 indicator based evolutionary algorithm for multiobjective optimization. In: Proceedings of 2013 IEEE Congress on Evolutionary Computation; 2013 Jun 20–23; Cancun, Mexico; p. 1836–45.
9. Li B, Tang K, Li J, Yao X. Stochastic ranking algorithm for many-objective optimization based on multiple indicators. IEEE Trans Evol Comput. 2016 Mar;20(6):924–38. doi:10.1109/TEVC.2016.2549267.
10. Liang Z, Luo T, Hu K, Ma X, Zhu Z. An indicator-based many-objective evolutionary algorithm with boundary protection. IEEE Trans Cybern. 2020 Jan;51(9):4553–66. doi:10.1109/TCYB.2019.2960302.
11. Peng C, Dai C, Xue X. A many-objective evolutionary algorithm based on dual selection strategy. Entropy. 2023 Jul;25(7):1015. doi:10.3390/e25071015.
12. Zhang Q, Li H. MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans Evol Comput. 2007 Dec;11(6):712–31. doi:10.1109/TEVC.2007.892759.
13. Li H, Zhang Q. Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-I I. IEEE Trans Evol Comput. 2008 Apr;13(2):284–302. doi:10.1109/TEVC.2008.925798.
14. Ke L, Zhang Q, Battiti R. MOEA/D-ACO: a multiobjective evolutionary algorithm using decomposition and antcolony. IEEE Trans Cybern. 2013 Dec;43(6):1845–59. doi:10.1109/TSMCB.2012.2231860.
15. Xiang Y, Zhou Y, Li M, Chen Z. A vector angle-based evolutionary algorithm for unconstrained many-objective optimization. IEEE Trans Evol Comput. 2016 Feb;21(1):131–52. doi:10.1109/TEVC.2016.2587808.
16. De Farias LRC, Araújo AFR. A decomposition-based many-objective evolutionary algorithm updating weights when required. Swarm Evol Comput. 2022 Feb;68:100980. doi:10.1016/j.swevo.2021.100980.
17. Wang X, Zhang F, Yao M. A many-objective evolutionary algorithm with metric-based reference vector adjustment. Comp Intell Syst. 2024 Feb;10(1):207–31. doi:10.1007/s40747-023-01161-w.
18. Malihe J, Eysa S, Javad S. Elephant clan optimization: a nature-inspired metaheuristic algorithm for the optimal design of structures. Appl Soft Comput. 2021 Dec;113:107892. doi:10.1016/j.asoc.2021.107892.
19. Wu Y, Wei J, Ying W, Lan Y, Cui Z, Wang Z. A collaborative decomposition-based evolutionary algorithm integrating normal and penal-ty-based boundary intersection methods for many-objective optimization. Inf Sci. 2022 Nov;616:505–25. doi:10.1016/j.ins.2022.10.136.
20. Liu HL, Gu F, Zhang Q. Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems. IEEE Trans Evol Comput. 2014 Sep;18(3):450–5. doi:10.1109/TEVC.2013.2281533.
21. Huband S, Hingston P, Barone L, While L. A review of multiobjective test problems and a scalable test problem toolkit. IEEE Trans Evol Comput. 2006 Oct;10(5):477–506. doi:10.1109/TEVC.2005.861417.
22. Zitzler E, Deb K, Thiele L. Comparison of multi-objective evolutionary algorithms: empirical results. Evol Comput. 2000 Jun;8(2):173–95. doi:10.1162/106365600568202.