mathematics

MDPI

Article

An Improved Parrot Optimization Algorithm Applied to Engineering Optimization Problems

Yueqiao Yang [1], Mingyuan Li [1], Minhao Qu [2,3,*], Yang Gao [2], Liang Zhao [4] and Boni Du [1]

[1] School of Emergency Technology and Management, University of Emergency Management, Langfang 065201, China; yangyueqiao@cidp.edu.cn (Y.Y.); 24661702@st.cidp.edu.cn (M.L.); 25661726@st.cidp.edu.cn (B.D.)
[2] National Earthquake Response Support Service, Beijing 100049, China; xyy816@126.com
[3] College of Mechanical and Transportation Engineering, Hunan University, Changsha 410082, China
[4] School of Urban Economics and Management, Beijing University of Civil Engineering and Architecture, Beijing 102616, China; zhaoliang91@bucea.edu.cn
* Correspondence: nerssquminhao@163.com

Abstract

Metaheuristic algorithms have become effective tools for solving complex engineering optimization problems. The Parrot Optimization (PO) algorithm is a recently proposed method with promising performance; however, it still suffers from premature convergence and slow convergence when handling complex tasks. To address these issues, this paper proposes an improved variant termed the Elite Cauchy Multi-scale Multi-directional Parrot Optimization algorithm (ECMPO). The proposed ECMPO incorporates three coordinated strategies. First, an elite learning strategy is introduced before the behavioral update phase to guide the population toward promising regions and accelerate convergence. Second, after the behavioral update, individuals are adaptively processed based on their fitness: elite individuals perform a multi-scale multi-directional search to enhance local exploitation. Third, non-elite individuals undergo Cauchy mutation to increase population diversity and strengthen global exploration. These strategies enable ECMPO to achieve a better balance between exploration and exploitation. To evaluate its performance, ECMPO is examined through ablation studies on the CEC2017 benchmark functions and further compared with eleven algorithms on the CEC2020–2022 benchmark suites. Finally, it is applied to six constrained engineering design problems. The results demonstrate that ECMPO exhibits superior performance compared with competitive algorithms. ECMPO shows strong robustness and applicability in practical engineering optimization tasks.

Keywords: engineering design; parrot optimization algorithm; elite learning; Cauchy mutation; metaheuristic optimization algorithms

MSC: 68T20; 68W25; 68Q25; 90C27; 90C59

Check for updates

Academic Editor: Ioannis Tsoulos

Received: 20 March 2026
Revised: 14 April 2026
Accepted: 15 April 2026
Published: 19 April 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Engineering design optimization problems often involve multivariable coupling and high-dimensional search spaces. Consequently, their scale and complexity have increased significantly, making the limitations of traditional optimization methods increasingly evident [1]. In particular, traditional numerical optimization approaches such as gradient descent [2] not only tend to be trapped in local optima when solving large-scale problems [3] but also require substantial computational resources, which limits their applicability in practical engineering scenarios [4].

In this context, achieving efficient and reliable solutions for complex engineering design optimization problems has become a fundamental challenge [5–7]. Due to the combined effects of strong nonlinearity, high-dimensional constraints, and multimodal characteristics, it remains challenging for current optimization methods to maintain a proper balance between exploration and exploitation [8]. As a result, there is a critical need to develop advanced optimization algorithms that can enhance convergence accuracy, avoid premature convergence, and maintain solution diversity in complex search spaces [9].

Metaheuristic algorithms are a class of intelligent optimization techniques inspired by cooperative behaviors and information sharing among individuals in a population [10]. These algorithms search for optimal or near-optimal solutions in the solution space through collective behaviors [11]. They exhibit strong flexibility and robustness. Therefore, they are well-suited for addressing complex challenges in engineering optimization problems [12]. Consequently, these methods have found extensive applications across various practical engineering domains, such as engineering design [13], resource allocation [2], and data mining [14].

The Parrot Optimization (PO) algorithm is a recently proposed metaheuristic optimization algorithm developed by Lian et al. in 2024 [15]. Inspired by the natural behaviors of parrots, PO simulates activities such as foraging, staying, communicating, and fear of strangers and balances global exploration and local exploitation through the random switching of different search behaviors. In addition, the PO algorithm reduces dependence on external parameters through inherent behavioral mechanisms and dynamic adjustment strategies, resulting in a relatively simple algorithmic structure [16]. Consequently, when PO is applied to practical optimization tasks, complex parameter tuning can be avoided, thereby simplifying its practical implementation. Furthermore, owing to its stochastic search mechanism and multi-behavior collaborative strategy, PO exhibits strong adaptability in solving complex optimization problems [17,18]. However, this balance primarily relies on random behavior selection and lacks effective control mechanisms [19]. In complex engineering optimization problems, the algorithm remains susceptible to insufficient local exploitation capability, loss of population diversity, and reduced search efficiency, which ultimately limit its overall optimization performance [20,21]. Therefore, numerous studies have attempted to improve the PO algorithm to enhance its search capability and stability when addressing complex optimization problems.

Scholars have proposed various mathematical strategies to compensate for the limitations of the original PO algorithm, thereby significantly improving the performance of the standalone algorithm. Yang et al. [22] proposed the CGBPO algorithm integrating Logistic mapping, Gaussian mutation, and centroid opposition-based learning to enhance population diversity and avoid local optima. Results show that CGBPO outperforms seven representative algorithms in terms of convergence speed, accuracy, and stability. Wu et al. [23] developed a binary multi-strategy enhanced PO (EMEPO) for feature selection, demonstrating superior feature quality and classification performance. Li et al. [24] proposed a multi-strategy Parrot Optimizer (MSPO) for breast cancer image classification, incorporating Sobol sequence initialization, nonlinear decreasing inertia weights, and chaotic parameters to improve exploration and convergence stability. Furthermore, Lei et al. [25] proposed AWTPO, which employs hybrid mutation and elite retention mechanisms to balance exploration and exploitation, and experimental results on benchmark and engineering problems show superior optimization performance.

In addition to strategy-based improvements, some researchers combine PO with other optimization algorithms or advanced techniques to construct hybrid models, further enhancing the performance of the original algorithm. Huang et al. [26] proposed an IPO–RF hybrid model for autonomous underwater vehicle (AUV) fault diagnosis, achieving a fault

identification accuracy of 99.59%. Soundradevi et al. [27] developed a PO-DWHCN hybrid model integrating PO with a dynamic weighted hypergraph convolutional network, which improves prediction accuracy, reduces operational costs, and increases renewable energy system utilization. Shaheen et al. [19] combined the Osprey Optimization Algorithm (OOA) with PO for parameter identification, leveraging the exploration ability of OOA and the local search capability of PO to improve overall optimization performance.

Furthermore, several studies extend PO to multi-objective optimization frameworks, where it demonstrates promising performance in solving problems involving multiple objectives and constraints. Saad et al. [28] proposed MOPO, which integrates an external archive and Pareto dominance, and benchmark comparisons show that it outperforms several leading multi-objective algorithms. Yan et al. [29] developed an adaptive quantum multi-objective Parrot Optimization algorithm (AQMOPO) for task allocation in multi-UUV underwater search missions. Zhang et al. [30] proposed an MOPO with improved Lévy flight and adaptive elliptical screening for wireless network deployment optimization, and experimental results rank the algorithm first in Friedman tests.

In summary, despite these improvements, the PO algorithm still exhibits limitations when addressing complex engineering optimization problems. Specifically, the limitations can be summarized as follows:

(a) Many existing improvement mechanisms focus on enhancing exploration while overlooking the deep exploitation of high-quality individuals, leading to insufficient exploitation capability in the later search stages.
(b) In multimodal and high-dimensional search spaces, population diversity may diminish rapidly, increasing the risk of premature convergence.
(c) Traditional iterative strategies often rely on fixed directions or single-scale updates, limiting adaptability to complex landscapes and reducing search efficiency.

To overcome these limitations, this paper proposes an improved Elite Cauchy Multi-scale Multi-directional Parrot Optimization (ECMPO) algorithm. ECMPO introduces a staged and condition-based optimization framework. To address limitation (a), an elite learning strategy is incorporated before the behavioral update phase to guide the population toward promising regions, thereby enhancing exploitation of high-quality individuals and improving convergence behavior. To address limitation (b), a Cauchy mutation mechanism is applied to non-elite individuals after the behavioral update to maintain population diversity and reduce the risk of premature convergence. To address limitation (c), a multi-scale multi-directional search strategy is assigned to elite individuals, enabling flexible search across different scales and directions and improving adaptability to complex landscapes. Through this staged and adaptive mechanism, ECMPO effectively coordinates exploration and exploitation.

The primary contributions of this work are outlined as follows:

(1) The proposed ECMPO introduces an elite learning strategy prior to the behavioral update phase and employs a fitness-based adaptive branching mechanism to assign differentiated search strategies to individuals, in which elite individuals perform a multi-scale multi-directional search to enhance exploitation, while non-elite individuals undergo Cauchy mutation to maintain population diversity and promote exploration.
(2) The performance of the ECMPO algorithm is validated through comparative experiments with multiple representative algorithms, indicating competitive performance across diverse optimization problems.
(3) The ECMPO algorithm is further employed to solve several practical optimization problems, demonstrating its effectiveness and potential in practical engineering scenarios.

This article is organized as follows: Section 2 introduces the original PO algorithm and the proposed improved ECMPO algorithm. Section 3 evaluates ECMPO's performance on a set of benchmark suites and analyzes the results. Section 4 applies ECMPO to six engineering optimization problems to further demonstrate its effectiveness. Section 5 summarizes the findings of this paper and discusses prospective research directions.

2. Methods

2.1. Parrot Optimization Algorithm

The PO algorithm [15] is inspired by several characteristic behaviors of domesticated Pyrrhura molinae, including foraging, staying, communication, and fear of strangers. In natural settings, these parrots tend to forage in small groups and are capable of guiding their search toward resource-rich regions using environmental cues such as vision and smell. Notably, domesticated parrots exhibit a strong attachment to their owners, often using the owner's position as a primary spatial reference (analogous to the current best solution) when moving or searching for resources.

In addition, parrots frequently perch at different positions on their owner's body for short periods, reflecting a stochastic yet locally bounded movement pattern. They also communicate through vocal signals to exchange information and maintain population-level coordination. When encountering unfamiliar individuals, parrots instinctively move away and return toward their owner, which serves as a perceived safe location.

These behavioral patterns are abstracted in the PO algorithm as four distinct search strategies. During the optimization process, individuals update their positions by probabilistically selecting one of the four behavioral strategies at each iteration.

During the initialization phase of the algorithm, a set of predefined solutions are generated as the initial population through Equation (1):

$$X_i^0 = lb + rand(0,1)\cdot(ub - lb) \tag{1}$$

where X_i^0 denotes the initial position of the i-th parrot, and lb and ub represent the lower and upper bounds of the search space.

(1) Foraging Behavior

In the foraging stage of PO, parrots identify a rough food location using visual signals or additional cues from the environment. Considering the owner's location, they move toward their respective target positions. Therefore, the position update is defined as follows:

$$X_i^{t+1} = (X_i^t - X_{best})\cdot Levy(dim) + rand(0,1)\cdot\left(1 - \frac{t}{Max_{iter}}\right)^{\frac{2t}{Max_{iter}}}\cdot X_{mean}^t \tag{2}$$

where X_i^t represents the current position and X_i^{t+1} denotes the updated position. X_{mean}^t indicates the average position of the current population, and $Levy(dim)$ is defined as a Lévy flight distribution describing the parrots' movement trajectories. X_{best} denotes the best position found so far, corresponding to the owner's current location, t represents the current iteration, and Max_{iter} is defined as the maximum allowable number of iterations.

(2) Staying Behavior

Parrots exhibit strong social characteristics and often perch in different locations of their owner for short periods. Within the PO algorithm, this behavior is mathematically described as individuals randomly moving toward the best solution while selecting random stopping positions. Its updating mechanism is given as follows:

$$X_i^{t+1} = X_i^t + X_{best}\cdot Levy(dim) + rand(0,1)\cdot ones(1,dim) \tag{3}$$

where $ones(1, dim)$ represents the vector of all ones with dimension dim. $X_{best} \cdot Levy(dim)$ indicates the movement of a parrot flying toward the owner, while $rand(0, 1)$ represents the random selection of a stopping location on the owner.

(3) Communicating Behavior

Parrots exhibit strong social characteristics and frequently interact within their groups. Such communication may involve either moving toward the group or interacting without physically approaching it. In the PO framework, these two interaction patterns are considered to occur with equal likelihood, while the population mean is adopted to approximate the group center. The corresponding formulation is defined as follows:

$$X_i^{t+1} = \begin{cases} 0.2 \cdot rand(0,1) \cdot \left(1 - \frac{t}{Max_{iter}}\right) \cdot (X_i^t - X_{mean}^t), P \leq 0.5 \\ 0.2 \cdot rand(0,1) \cdot exp\left(-\frac{t}{rand(0,1) \cdot Max_{iter}}\right), \ P > 0.5 \end{cases} \tag{4}$$

where $P \in [0, 1]$ is a randomly generated number. When $P \leq 0.5$, an individual parrot moves toward the population and engages in communication; otherwise, it departs from the group after the interaction.

(4) Fear Behavior

The fear behavior of parrots is characterized by individuals moving away from unfamiliar groups and seeking a safe environment near the owner. This process is described by Equation (5):

$$\begin{aligned} X_i^{t+1} = X_i^t + rand(0,1) \cdot cos\left(0.5\pi \cdot \frac{t}{Max_{iter}}\right) \cdot (X_{best} - X_i^t) \\ - cos(rand(0,1) \cdot \pi) \left(\frac{t}{Max_{iter}}\right)^{\frac{2}{Max_{iter}}} \cdot (X_i^t - X_{best}) \end{aligned} \tag{5}$$

where $rand(0,1) \cdot cos\left(0.5\pi \cdot \frac{t}{Max_{iter}}\right) \cdot (X_{best} - X_i^t)$ represents the process of a parrot adjusting its direction and flying toward the owner, while $cos(rand(0,1) \cdot \pi) \left(\frac{t}{Max_{iter}}\right)^{\frac{2}{Max_{iter}}} \cdot (X_i^t - X_{best})$ describes the behavior of moving away from unfamiliar individuals.

2.2. The Proposed ECMPO

To enhance the optimization capability of the original PO algorithm, this study proposes an enhanced variant termed the ECMPO algorithm. Specifically, an elite learning mechanism is introduced before the behavioral update stage of the PO algorithm to fully exploit information from high-quality individuals and guide the population toward promising regions of the search space. After the standard behavioral updates of the PO algorithm are applied, a differentiated search mechanism is used to further refine the search. In this mechanism, elite individuals perform a multi-scale, multi-directional local search to refine solutions in promising areas, while non-elite individuals undergo Cauchy mutation to increase population diversity and expand the algorithm's exploration range.

By integrating these strategies cooperatively, the ECMPO algorithm achieves a better balance between exploration and exploitation during the optimization process. The detailed mechanisms of these strategies are described in the following subsections.

2.2.1. Elite Learning Strategy

In the conventional PO algorithm, individuals mainly update their positions through stochastic behaviors such as foraging and imitation during the iterative process. Although these mechanisms help maintain population diversity, they do not sufficiently exploit the information contained in high-quality solutions. As a result, the search process may suffer from slow convergence and limited exploitation capability.

To address this limitation, an elite learning strategy is introduced before the behavioral update stage of the PO algorithm. In this strategy, several top-performing individuals in the population are selected as elite exemplars. The other individuals update their positions by referencing the statistical center of the selected elite solutions, thereby obtaining a reliable search direction derived from high-quality candidate solutions. By incorporating elite information into the position update process, the population can move more efficiently toward promising regions of the search space, which improves both convergence speed and optimization performance.

Furthermore, an adaptive learning factor is introduced to coordinate the search requirements across different phases of the optimization process. A relatively large step size is adopted in the initial phase to strengthen global exploration capability. As iterations continue, the step size is gradually reduced to prevent excessive perturbations that could compromise convergence stability. Meanwhile, a random perturbation term is incorporated to preserve population diversity and reduce the risk of excessive reliance on elite individuals. The position update rule can be expressed as

$$X_i^{t+1} = X_i^t + \alpha \cdot \left(\frac{1}{m} \sum_{j=1}^{m} e_j^t - X_i^t \right) + \beta \cdot \mathrm{rand}(0,1) \cdot (X_{ub} - X_{lb}) \tag{6}$$

where X_i^t denotes the position of the i–th individual at iteration t, e_j^t represents the position of the j-th elite individual at iteration t, and m is the number of elite individuals. The parameter α is the elite learning coefficient that controls the strength of elite guidance, β is the random perturbation coefficient used to maintain population diversity, and X_{ub} and X_{lb} are the upper and lower bounds of the search space.

2.2.2. Cauchy Mutation Strategy

During the iterative process of the original PO algorithm, the continuous communication and imitation among individuals may gradually drive the population to cluster around locally dominant solutions. As the search progresses, the randomness of the population decreases, which increases the risk of search stagnation and premature convergence.

To alleviate this issue, a Cauchy mutation strategy is introduced for non-elite individuals after the behavioral update stage. By selectively applying mutation to non-elite individuals, the proposed mechanism enhances population diversity while preserving the stability of elite solutions. This differentiated search mechanism allows non-elite individuals to explore a broader region of the solution space, thereby reducing the likelihood of premature convergence.

The proposed mutation strategy exploits the sharp peak and heavy-tailed characteristics of the Cauchy distribution. This property enables most individuals to perform small local perturbations while allowing a small portion of individuals to generate large position jumps with a certain probability. Consequently, some individuals can escape from the current search region and explore previously unvisited areas of the solution space, which effectively enhances the global exploration capability of the algorithm. The Regin Cauchy mutation operation can be expressed as

$$X_i^{t+1} = X_i^t + \gamma \cdot \mathrm{Cauchy}(0,1) \cdot (X_{ub} - X_{lb}) \tag{7}$$

where X_{ub} and X_{lb} denote the upper and lower bounds of the decision variables, respectively. $\mathrm{Cauchy}(0,1)$ denotes a Cauchy-distributed random variable with location parameter 0 and scale parameter 1, whose probability density function is

$$f(x) = \frac{1}{\pi(1+x^2)} \tag{8}$$

To better coordinate the search requirements across different stages of the optimization process, an adaptive mutation intensity factor γ, which gradually decays with the iteration number, is introduced. This factor enables the algorithm to maintain strong random perturbation capability in the early stage to enhance global exploration, while gradually reducing the mutation intensity in the later stage to avoid excessive perturbations that may affect convergence stability. The definition of the mutation intensity factor is given as follows:

$$\gamma = \sigma_0 \cdot \left(1 - \frac{t}{Max_{iter}}\right) \tag{9}$$

where t denotes the current iteration number, Max_{iter} represents the maximum number of iterations, and σ_0 is a constant parameter used to control the initial mutation intensity.

2.2.3. Multi-Scale Multi-Directional Search Strategy

The combination of the elite learning mechanism and the Cauchy mutation strategy enables the algorithm to achieve a trade-off between convergence speed and global exploration ability. Nevertheless, in the later phase of optimization, as the population progressively converges toward promising regions of the search space, purely random perturbations are often inadequate for fine-grained local search. Consequently, the algorithm may find it difficult to further approach the global optimum.

A Multi-scale Multi-directional Search Strategy is introduced to further enhance the local exploitation capability of the algorithm. The proposed strategy performs a greedy neighborhood search around the current candidate solution by combining multiple search scales with randomized search directions. Specifically, perturbation step sizes are generated at different scales, and a random unit direction vector is constructed at each scale. The candidate solution is then explored along multiple directions within its neighborhood. Once a better solution is identified, the individual position is immediately updated, allowing the algorithm to progressively refine the solution and approach the global optimum.

Let X_i^t denote the position of the i-th individual at iteration t. A random unit direction vector d is generated to probe the neighborhood of the current solution. Under the search scale S_k, the local exploration step can be formulated as follows:

$$X_i^{t+1} = X_i^t + S_k \cdot d \tag{10}$$

If no improvement is observed along the sampled direction, the search continues in the opposite direction to further examine the surrounding region:

$$X_i^{t+1} = X_i^t - S_k \cdot d \tag{11}$$

where S_k denotes the k-th search scale, d represents a randomly generated unit direction vector, k = 1, 2, . . ., K, and K is the total number of search scales.

To better accommodate the search requirements at different stages of the optimization process, an adaptive search scale mechanism is incorporated. The search scale gradually decreases with the iteration number, allowing the algorithm to perform broader neighborhood exploration in the early stage and progressively transition to a fine-grained local search in the later stage. In this way, the strategy maintains a balance between global exploration and local exploitation throughout the optimization process. The search scale is defined as follows:

$$S_k = S_k^0 \left(1 - \frac{t}{Max_{iter}}\right) \tag{12}$$

where S_k^0 denotes the initial search scale, t is the current iteration number, and Max_{iter} represents the maximum number of iterations. The detailed pseudocode of the Multi-scale Multi-directional Search Strategy is presented in Algorithm 1.

Algorithm 1. Pseudocode of Multi-scale Multi-directional Search Strategy

Require: N : Total population size; T : Max_{iter}; dim: Problem dimension;
Ensure: X_{best}: global best solution;
For i = 1: N
  For k = 1: K
    Calculate search scale S_k by Equation (12)
    Generate a random unit direction vector d
    Generate a candidate solution X_{new} using Equation (10)
    If f (X_{new}) < f (X_i^t)
      $X_i^{t+1} = X_{new}$
    Else
      Generate candidate solution X_{new} using Equation (11)
      If f (X_{new}) < f (X_i^t)
        $X_i^{t+1} = X_{new}$
      End If
    End If
  End For
End For

Based on the above design, the overall framework of the ECMPO algorithm is illustrated in Figure 1, where the blue-highlighted components correspond to the three strategies proposed in this study. The detailed pseudocode of ECMPO is provided in Algorithm 2.

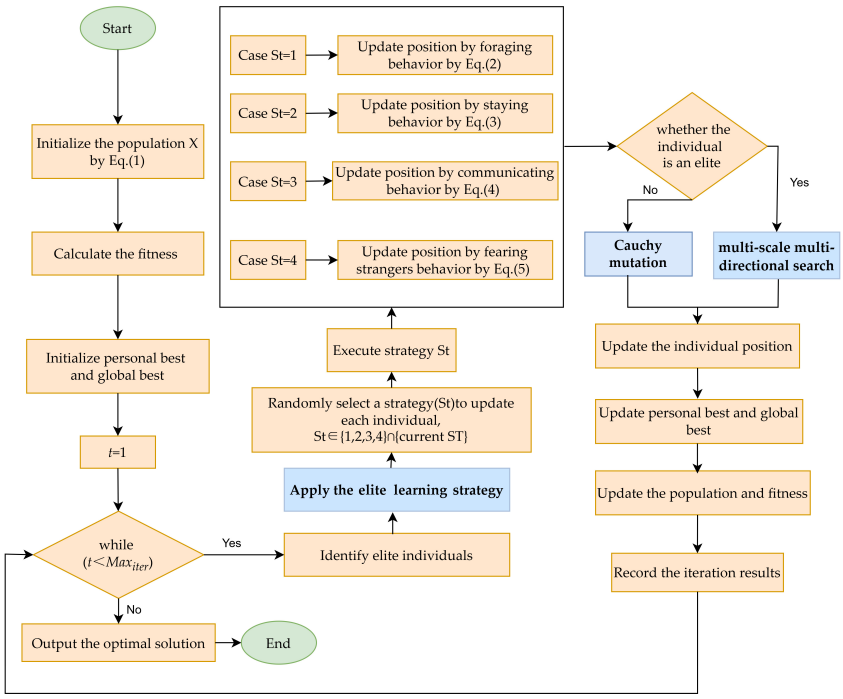


Figure 1. Flowchart of the ECMPO.

Algorithm 2. Pseudocode of the ECMPO algorithm

Require: N : Total population size; T : Max_{iter}; dim: Problem dimension;
Ensure : X_{best}: global best solution;
Initialize population positions X_i(i = 1, 2. . ., N) using Equation (1)
Evaluate the fitness value f (X_i) for each individual
Determine the global best solution X_{best}
t = 1
While (t ≤ T)
//elite learning strategy
Determine elite individuals
Update non-elite individuals in X using Equation (6)
For i = 1: N
St = randi ([1, 4])
IF St = 1
Update X_i using foraging behavior Equation (2)
ELSE IF St = 2
Update X_i using staying behavior Equation (3)
ELSE IF St = 3
Update X_i using communicating behavior Equation (4)
ELSE IF St = 4
Update X_i using fear behavior Equation (5)
END IF
IF X_i is an elite individual
Perform **multi-scale multi-directional search** by Algorithm 1
ELSE
Update X_i using **Cauchy mutation** Equation (7)
END IF
End For
Evaluate fitness values
Update X_{best}
t = t + 1
End While
Return X_{best}

2.2.4. Time Complexity Analysis

Computational complexity is a critical metric for evaluating the efficiency of the ECMPO algorithm. The computational complexity of the proposed algorithm can be analyzed by examining each of its components individually.

In the initialization phase, the positions of all individuals in the population are randomly generated, leading to a time complexity of $O(N \times dim)$, where N denotes the population size and dim represents the dimensionality of the decision variables.

In the iterative optimization phase, the total computational cost mainly consists of three components. First, in the elite learning strategy, the population needs to be sorted to identify elite individuals, which incurs a complexity of $O(NlogN)$. Subsequently, an elite-guided position update is conducted across all individuals, contributing an additional computational complexity of $O(N \times dim)$. Second, the four behavioral update mechanisms inherited from the original PO algorithm (foraging, staying, communicating, and fear behaviors) perform vector operations on all individuals, yielding a complexity of $O(N \times dim)$. Third, in the differentiated search mechanism, elite individuals perform a multi-scale multi-directional search, where each elite individual conducts K local searches.

This results in a computational cost of $O(m \times K \times dim)$, where m is the number of elite individuals. Meanwhile, non-elite individuals undergo Cauchy mutation with a complexity of $O((N-m) \times dim)$. Since m and K are typically small constants in practical implementations, the computational complexity of this part is simplified to $O(N \times dim)$.

Consequently, the overall computational complexity per iteration is expressed as $O(N \times dim + NlogN)$. The total time complexity of ECMPO is $T_{total} \approx O(Max_{iter} \times (N \times dim + NlogN))$.

In summary, compared to the original PO algorithm's complexity of $O(Max_{iter} \times N \times dim)$, ECMPO introduces a moderate additional computational cost while significantly improving optimization performance, maintaining good scalability for high-dimensional problems.

2.2.5. Convergence Analysis

To establish the theoretical reliability of our proposed ECMPO algorithm in solving complex optimization problems, this subsection presents a global convergence analysis based on the Markov chain framework and stochastic optimization theory [31,32].

A. Markov Chain Modeling of ECMPO

Let the population at iteration t be denoted as

$$S(t) = \{X_1(t), X_2(t), \ldots X_N(t)\}, X_i(t) \in \Omega \tag{13}$$

where N denotes the population size, $X_i(t)$ represents the position of the i-th individual, and Ω is the feasible solution space.

Since the generation of $S(t+1)$ depends only on the current state $S(t)$ and stochastic transition rules (including behavioral updates, elite learning, and mutation operators), the sequence $\{S(t)\}$ satisfies the Markov property. Therefore, ECMPO can be modeled as a homogeneous Markov chain defined on the state space Ω^N, where Ω is the feasible solution space.

Furthermore, due to the stochastic perturbation mechanisms embedded in the algorithm, the induced Markov chain is irreducible and aperiodic, which ensures that the chain has the ability to explore the entire feasible space without being trapped in periodic cycles.

B. Convergence Properties

(1) Satisfaction of Elitism

In ECMPO, an elite learning strategy is incorporated to maintain the global best solution $X_{gb}(t)$. Specifically, at each iteration, the fitness of newly generated solutions is evaluated, and the global best solution is updated only if a better candidate is found. Otherwise, the previous best solution is retained. Therefore, the following relationship always holds:

$$f\left(X_{gb}(t+1)\right) \leq f\left(X_{gb}(t)\right) \tag{14}$$

This ensures that the sequence of best fitness values is monotonically non-increasing. Hence, ECMPO satisfies the elitism condition.

(2) Satisfaction of Global Reachability

ECMPO integrates multiple stochastic search mechanisms, including behavioral updates, multi-scale multi-directional search, and Cauchy mutation.

Among these, the Cauchy mutation plays a crucial role in ensuring global exploration. The Cauchy distribution has the following probability density function:

$$f(x) = \frac{1}{\pi(1+x^2)} \tag{15}$$

Due to its heavy-tailed property, the Cauchy distribution enables the generation of large perturbations with non-negligible probability. Consequently, for any measurable subset A $\subseteq \Omega$ with nonzero measure, there exists a nonzero probability that a candidate solution falls within A, i.e.,

$$P(X_{t+1} \in A \mid X_t) > 0 \tag{16}$$

This indicates that the algorithm possesses a nonzero probability of visiting any neighborhood in the feasible search domain, including regions surrounding the global optimum X^*. Hence, ECMPO satisfies the global reachability condition.

Since ECMPO satisfies both the elitism condition and the global reachability condition, and the induced Markov chain is irreducible and aperiodic, according to standard convergence results for stochastic optimization algorithms, it follows that

$$\lim_{t \to \infty} P(X_t \in R_\varepsilon(X^*)) = 1 \tag{17}$$

where $R_\varepsilon(X^*)$ denotes an ε-neighborhood of the global optimal solution X^*.

This result indicates that, as the number of iterations increases, the probability that ECMPO reaches the neighborhood of the global optimum approaches 1. Therefore, the proposed algorithm achieves probabilistic global convergence and offers strong theoretical support for complex engineering optimization problems.

3. Experimental Results and Discussion

3.1. Experimental Setup

To assess the effectiveness of the proposed strategies, ablation experiments are first performed on the CEC2017 benchmark suite to examine the individual contributions of elite learning, Cauchy mutation, and the multi-scale multi-directional search strategy to the performance improvement of the original PO algorithm.

Subsequently, the proposed ECMPO algorithm was compared with eleven representative optimization algorithms on three challenging benchmark suites, namely the CEC2020, CEC2021, and CEC2022 benchmark suites. The algorithms used for comparison consist of the following: the Grey Wolf Optimizer (GWO) [33], Vulture Optimization Algorithm (VOA) [34], Hippopotamus Optimization Algorithm (HOA) [35], Walrus Optimization Algorithm (WaOA) [36], Arctic Sea Eagle Optimization (ASEO) [37], Blood Sucker Leech Optimization (BSLO) [38], Emperor Penguin Optimizer (EPO) [39], Schrödinger Optimizer (SRA) [40], Chaotic Evolution optimization (CEO) [41], Chaotic–Gaussian Barycenter Parrot Optimization (CGBPO) [22] and an enhanced version of the PO(EMEPO) [23]. The parameter configurations of both the compared algorithms and the proposed ECMPO are summarized in Table 1. To ensure the reproducibility of the results, the source code is publicly available at https://github.com/ze190612/cautious-memory (accessed on 14 April 2026). These values are obtained from the corresponding original literature or represent their optimal settings. To investigate the effect of key parameters of our proposed ECMPO, a preliminary sensitivity analysis was conducted on several representative benchmark functions. The results indicate that ECMPO is not highly sensitive to parameter variations within reasonable ranges. Therefore, the parameter settings used in this paper are selected based on common practices and empirical experience.

Table 1. Parameters of algorithms.

Algorithm	Parameter	Value
GWO (2013)	Parameter (a)	Linear reduction from 2 to 0
VOA (2021)	p1	0.6
	p2	0.4
	p3	0.6
	Starvation rate (F)	Adaptive in [−2, 2]
HOA (2024)	Energy parameter (E)	Linear reduction from 2 to 0
	Spiral coefficient (l)	[−1, 1]
WaOA (2023)	Temperature (T)	Decreases from 1 to 0
	Coefficient vector (A)	[−T, T]
	Coefficient vector C	[0, 2]
ASEO (2024)	Storm probability (Ps)	[0.05, 0.2]
	Competition factor (Pc)	[0.4, 0.8]
	Environmental temperature (T)	Exponential decay
BSLO (2024)	Detection range	[0.1, 0.5]
	Tracking strength	[0.3, 1.0]
	Random walk probability (P_{rw})	[0.2, 0.5]
	Blood-sucking factor	[0.2, 0.5]
	Temperature profile (T)	Exponential decay
EPO (2018)	T_{min}	0.1
SRA (2025)	Wave function factor (ψ)	Adaptive
	Potential well depth (V)	[0, 1]
	Quantum control parameter ($\hbar$)	Decreasing
	Collapse probability (P_{collapse})	[0.1, 0.5]
CEO (2025)	Chaotic map parameter (μ)	(0, 1)
	Initial chaotic value (x_0)	[0, 2]
	Control parameter (β)	[0.1, 0.3]
	Chaotic perturbation probability (P_c)	
CGBPO (2025)	Logistic map control parameter (a)	4
	Levy flight parameter(γ)	(1,2)
EMEPO (2025)	Exponential decay factor	$exp(-t/(rand \cdot Maxiter))$
	Cosine control factor	$cos(0.5\pi \cdot t/Maxiter)$
Our proposed ECMPO	Elite proportion	0.1
	Cauchy mutation scale factor (γ)	(0, 1]
	Number of search scales (K)	[3, 10]

To quantitatively compare the performance of different algorithms, the Friedman test was employed to rank the algorithms based on their optimization performance. Furthermore, the Wilcoxon rank-sum test was applied to test whether ECMPO significantly outperforms the compared algorithms in terms of performance.

The benchmark suites CEC2017, CEC2020, CEC2021, and CEC2022 consist of four categories of functions, as summarized in Table 2. These categories evaluate different aspects of algorithmic performance. Unimodal functions contain a single global optimum and are mainly used to assess the exploitation capability of an algorithm. Multimodal functions feature multiple local optima and are commonly utilized for evaluating exploration performance. Hybrid functions and composition functions combine multiple characteristics of basic functions and are designed to test the robustness of algorithms and their ability to avoid being trapped in local optima.

Table 2. The composition of the benchmark functions in the four test suites.

Test Suite	CEC2017	CEC2020	CEC2021	CEC2022
Unimodal Functions	F1–F2	F1	F1	F1
Multimodal Functions	F3–F9	F2–F4	F2–F4	F2–F5
Multimodal/Basic Functions	F10–F19	F5–F7	F5–F7	F6–F8
Composition Functions	F20–F29	F8–F10	F8–F10	F9–F12

All algorithms were implemented in Python 3.9, and all experiments were conducted on a laptop equipped with an Intel (R) Core (TM) i7-8565U processor and 16 GB RAM. For a fair comparison, a population size of 30 and a maximum of 1000 iterations were used consistently across all algorithms. Each benchmark function was run independently 30 times, and both the mean (Aver) and standard deviation (Std) were recorded. The parameter settings of the comparison algorithms were set according to the default values suggested in their original publications.

3.2. Ablation Experiment

To comprehensively analyze the impact of each enhancement strategy on the performance of the ECMPO algorithm, the original PO algorithm was progressively enhanced through the incorporation of three key strategies: the elite learning strategy, the Cauchy mutation strategy, and the multi-scale multi-directional search strategy.

To evaluate the individual contribution of each strategy, several algorithm variants were constructed by selectively removing one or more components of ECMPO. The detailed configurations of these algorithm variants and the strategies adopted in each variant are summarized in Table 3, where "1" indicates the inclusion of the corresponding strategy in the algorithm, whereas "0" indicates its absence.

Table 3. Variants of the PO algorithm with different strategies.

Algorithm	Elite Learning (E)	Cauchy Mutation (C)	Multi-Scale Multi-Directional Search (M)
PO	0	0	0
EPO	1	0	0
CPO	0	1	0
MPO	0	0	1
ECMPO	1	1	1

The experimental results on the CEC 2017 benchmark functions with a 30−dimensional search space are summarized in Table 4. In this table, the symbols "−/=/+" denote the number of benchmark problems on which the compared algorithms perform worse than, are comparable to, or outperform the ECMPO algorithm, respectively. In addition, "Friedman Aver Rank" refers to the mean ranking of each algorithm obtained via the Friedman test, while "Final Rank" represents the overall ranking of the algorithms across all test functions.

Table 4. Comparison results on CEC2017 (CEC2027 F1–CEC2017 F29, dim = 30).

ID	Criteria	PO	EPO	CPO	MPO	ECMPO
F1	Aver	4.8926×10^{10}	2.5638×10^{10}	2.8055×10^{8}	3.0692×10^{10}	$\mathbf{1.2778 \times 10^{7}}$
F2	Aver	8.8558×10^{4}	8.3182×10^{4}	$\mathbf{3.7456 \times 10^{3}}$	7.4359×10^{4}	1.0389×10^{4}
F3	Aver	1.1899×10^{4}	4.9493×10^{3}	5.5312×10^{2}	6.3758×10^{3}	$\mathbf{4.8992 \times 10^{2}}$
F4	Aver	6.4995×10^{4}	3.8677×10^{4}	$\mathbf{1.2730 \times 10^{3}}$	4.3797×10^{4}	1.6714×10^{3}
F5	Aver	5.0003×10^{2}	5.0002×10^{2}	5.0000×10^{2}	5.0001×10^{2}	$\mathbf{5.0000 \times 10^{2}}$
F6	Aver	2.0798×10^{6}	1.2538×10^{6}	1.3738×10^{4}	1.3545×10^{6}	$\mathbf{2.8806 \times 10^{3}}$
F7	Aver	1.2907×10^{3}	6.4543×10^{1}	1.1002×10^{3}	1.1487×10^{3}	$\mathbf{1.0358 \times 10^{3}}$
F8	Aver	8.2868×10^{2}	8.1700×10^{2}	$\mathbf{8.0745 \times 10^{2}}$	8.1632×10^{2}	8.0752×10^{2}
F9	Aver	7.9796×10^{3}	7.5061×10^{3}	6.2507×10^{3}	5.1004×10^{3}	$\mathbf{4.5348 \times 10^{3}}$
F10	Aver	1.0085×10^{9}	3.6450×10^{8}	1.5182×10^{5}	3.1580×10^{8}	$\mathbf{8.2605 \times 10^{4}}$
F11	Aver	7.1196×10^{9}	3.3568×10^{9}	4.4476×10^{7}	4.7364×10^{9}	$\mathbf{9.9542 \times 10^{6}}$
F12	Aver	7.7917×10^{9}	2.9724×10^{9}	7.6060×10^{6}	4.5277×10^{9}	$\mathbf{3.4790 \times 10^{5}}$
F13	Aver	6.0517×10^{6}	4.5057×10^{6}	$\mathbf{1.5604 \times 10^{6}}$	3.7123×10^{6}	1.9395×10^{6}
F14	Aver	3.2454×10^{9}	9.5646×10^{8}	1.7421×10^{6}	1.4611×10^{9}	$\mathbf{1.6600 \times 10^{5}}$
F15	Aver	6.9957×10^{8}	8.4864×10^{7}	4.2745×10^{5}	1.1526×10^{8}	$\mathbf{3.0298 \times 10^{5}}$
F16	Aver	6.4533×10^{12}	1.3801×10^{11}	3.5962×10^{4}	9.5442×10^{11}	$\mathbf{1.4691 \times 10^{4}}$
F17	Aver	1.2891×10^{7}	4.5442×10^{6}	6.8456×10^{5}	2.4557×10^{6}	$\mathbf{2.8206 \times 10^{5}}$
F18	Aver	4.6011×10^{12}	2.2322×10^{11}	3.0154×10^{6}	1.0938×10^{12}	$\mathbf{8.5035 \times 10^{4}}$
F19	Aver	7.9936×10^{3}	6.0148×10^{3}	4.1082×10^{3}	6.3283×10^{3}	$\mathbf{3.5065 \times 10^{3}}$
F20	Aver	3.4883×10^{4}	2.3232×10^{4}	$\mathbf{2.8568 \times 10^{3}}$	2.3416×10^{4}	2.9094×10^{3}
F21	Aver	3.8077×10^{3}	3.1525×10^{3}	$\mathbf{2.3655 \times 10^{3}}$	3.6803×10^{3}	2.3691×10^{3}
F22	Aver	5.3881×10^{4}	4.0018×10^{4}	4.2839×10^{3}	3.7134×10^{4}	$\mathbf{3.1786 \times 10^{3}}$
F23	Aver	3.1878×10^{4}	2.2888×10^{4}	3.6842×10^{3}	2.3584×10^{4}	$\mathbf{2.9022 \times 10^{3}}$
F24	Aver	6.2787×10^{3}	4.4469×10^{3}	2.9345×10^{3}	4.7740×10^{3}	$\mathbf{2.9190 \times 10^{3}}$
F25	Aver	9.0523×10^{3}	5.0952×10^{3}	$\mathbf{3.3850 \times 10^{3}}$	7.0880×10^{3}	3.3907×10^{3}
F26	Aver	4.0427×10^{3}	3.6066×10^{3}	$\mathbf{3.2085 \times 10^{3}}$	3.8366×10^{3}	3.3004×10^{3}
F27	Aver	5.8060×10^{3}	4.5575×10^{3}	3.1880×10^{3}	5.2775×10^{3}	$\mathbf{3.1515 \times 10^{3}}$
F28	Aver	3.6191×10^{11}	7.0310×10^{10}	3.3212×10^{6}	3.2469×10^{11}	$\mathbf{2.3766 \times 10^{6}}$
F29	Aver	1.1905×10^{11}	2.0957×10^{10}	7.2939×10^{6}	7.7589×10^{9}	$\mathbf{1.3299 \times 10^{6}}$
</=/>		29/0/0	29/0/0	20/1/8	29/0/0	/
Friedman Aver Rank		5.0000	3.3448	1.7586	3.6207	1.2759
Final Rank		5	3	2	4	1

Bold is the best result of all algorithms.

As shown in Table 4, the original PO algorithm obtains a Final Rank of 4, indicating relatively weak optimization capability and an inability to achieve the best performance across the four categories of benchmark functions. In contrast, the proposed ECMPO achieves the first position in the Final Rank, demonstrating that the introduced strategies significantly improve the overall optimization performance.

In addition, the algorithm variants EPO, CPO, and MPO achieve lower average values (Aver) than the original PO algorithm on most benchmark functions, indicating that each enhancement strategy contributes positively to the optimization performance. Specifically, the elite learning strategy provides effective directional guidance for the population by exploiting information from high-quality individuals, thereby accelerating the convergence process. The Cauchy mutation strategy introduces long-distance perturbations through the heavy-tailed distribution of the Cauchy function, which enhances the algorithm's ability to escape local optima while maintaining population diversity. Meanwhile, the multi-scale multi-directional search strategy performs refined local exploration around elite individuals, improving the local exploitation capability of the algorithm in the vicinity of promising solutions. Notably, CPO achieves the best performance on seven benchmark functions, indicating that the Cauchy mutation strategy plays a crucial role in improving solution quality. This result suggests that the introduction of long-distance stochastic perturbations significantly enhances the global exploration capability of the algorithm in complex non-convex optimization environments.

For unimodal functions, ECMPO achieves the best result on F1, while CPO performs best on F2, with ECMPO ranking second. This observation indicates that Cauchy mutation plays an important role in maintaining search diversity. For more challenging functions (F3–F29), where numerous local optima and rugged search landscapes make the optimization process more difficult, ECMPO achieves the best results on 19 functions, further confirming the effectiveness of integrating these three strategies.

Figure 2 illustrates the convergence curves of the proposed ECMPO algorithm, the original PO algorithm, and its ablation variants (EPO, CPO, and MPO) on several representative CEC2017 benchmark functions. The purpose is to analyze the influence of different improvement strategies on the convergence behavior of the algorithm. Overall, ECMPO demonstrates faster convergence speed and better final solution quality on most benchmark functions, indicating that the proposed multi-strategy collaborative mechanism effectively enhances the overall optimization performance.

Specifically, for unimodal functions such as F1, EPO converges significantly faster than the original PO algorithm. This result indicates that the elite learning strategy can effectively guide the search process by exploiting information from high-quality individuals within the population, thereby accelerating the convergence toward promising regions and enhancing local exploitation capability.

For simple multimodal functions such as F5, the convergence curve of CPO outperforms that of the original PO algorithm. This observation suggests that the Cauchy mutation strategy helps the algorithm avoid being trapped in local optima. Due to the heavy-tailed property of the Cauchy distribution, the strategy introduces large stochastic perturbations into the population, thereby improving the algorithm's capability to escape local optima while preserving population diversity.

For complex multimodal landscapes such as F9, MPO exhibits a more stable convergence trend compared with the original PO and continues to approach better solutions during the optimization process. This phenomenon indicates that the multi-scale multi-directional search strategy enables more comprehensive exploration of the search space from different scales and directions, thereby improving the global exploration capability of the algorithm in complex landscapes.

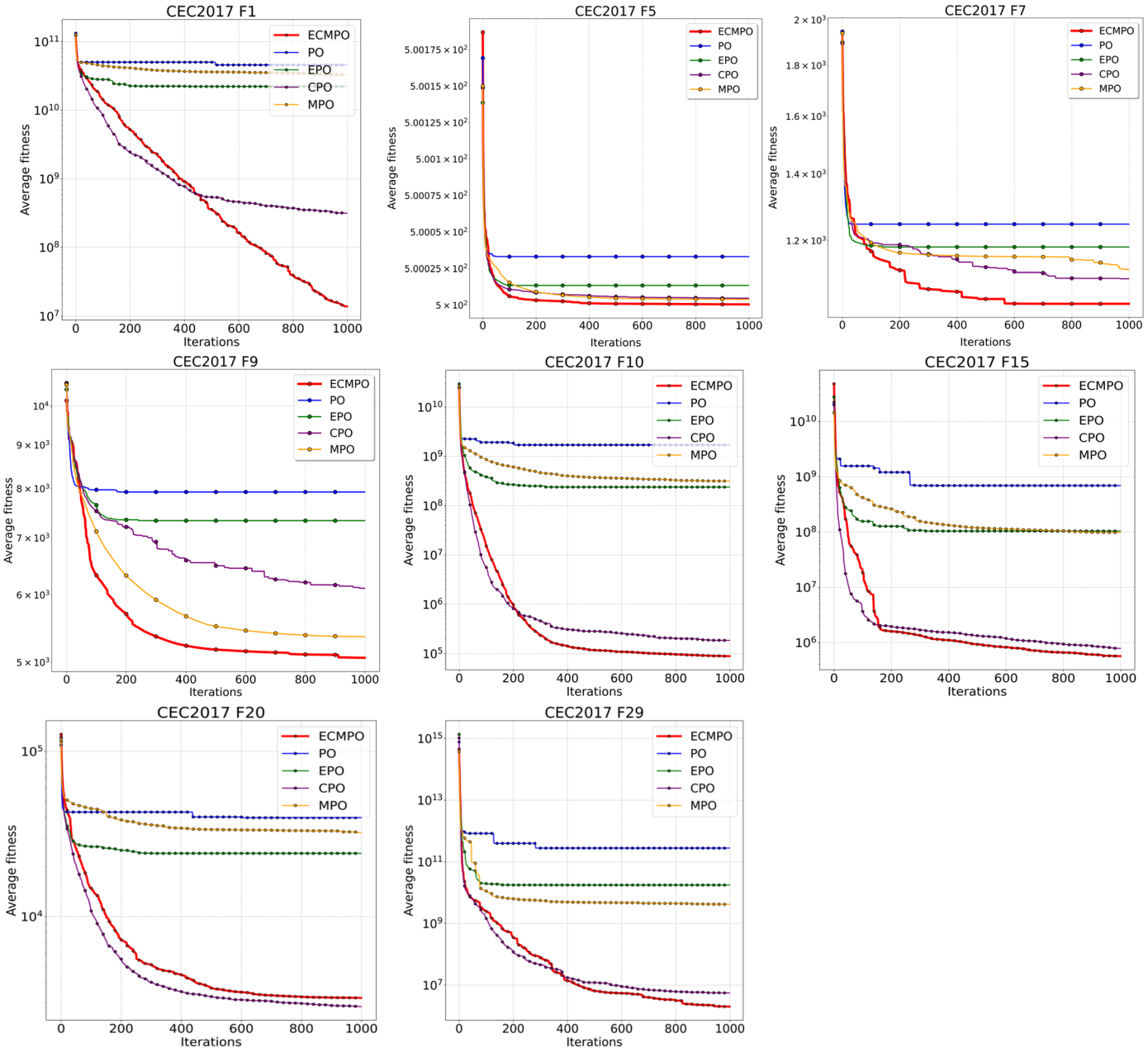


Figure 2. Convergence comparison of the original PO and its variants with different strategies.

Furthermore, it can be observed that ECMPO consistently outperforms all ablation variants on most benchmark functions, demonstrating the coordinated effects of the three improvement strategies. By effectively integrating elite learning, Cauchy mutation, and multi-scale multi-directional search strategies, ECMPO achieves superior convergence performance and stability across different types of CEC2017 benchmark functions, which further validates the effectiveness of the proposed improvements.

To evaluate the efficiency and scalability of the algorithms in addressing high-dimensional optimization problems, Table 5 summarizes the average execution times of ECMPO and the original PO algorithm across 30, 50, and 100 dimensions. Compared to the original PO, the integration of the elite learning strategy, Cauchy mutation, and multi-scale multi-directional search inevitably leads to an increased computational overhead. Nevertheless, the experimental results confirm that ECMPO yields superior solution quality, demonstrating a favorable balance between computational cost and optimization accuracy.

Table 5. Average runtime comparison on CEC2017 under different dimensions.

Dimensional	Dim 30		Dim 50		Dim 100	
Algorithms	ECMPO	PO	ECMPO	PO	ECMPO	PO
Aver_Time (s)	5.322	3.395	12.388	9.343	19.334	13.670

3.3. Comparison on the CEC2020, CEC2021, and CEC2022 Benchmark Suites

To further assess the exploration capability, exploitation ability, and local optima avoidance performance of the ECMPO algorithm, comparative experiments were carried out on the CEC2020, CEC2021, and CEC2022 benchmark suites with eleven competing algorithms. Tables 6–8 present the optimization results of ECMPO and the competing algorithms on the benchmark functions, where the best results are highlighted in bold.

Table 6. Comparison results on CEC2020 (CEC2020 F1–CEC2020 F10, dim = 30).

ID	Criteria	ECMPO	GWO	VOA	HOA	WaOA	ASEO	BSLO	EPO
F1	Aver	**1.2778×10^{4}**	2.0951×10^{10}	6.6350×10^{10}	2.0735×10^{7}	1.0789×10^{10}	2.2178×10^{10}	1.2147×10^{11}	6.3315×10^{9}
	Std	**1.4349×10^{4}**	4.3897×10^{9}	1.4263×10^{10}	1.6889×10^{9}	3.8640×10^{9}	4.7668×10^{9}	1.6349×10^{10}	2.5910×10^{9}
F2	Aver	**2.3686×10^{3}**	7.3759×10^{3}	8.9162×10^{3}	6.0469×10^{3}	8.4562×10^{3}	8.8807×10^{3}	1.0731×10^{4}	5.6085×10^{3}
	Std	**3.2024×10^{2}**	5.9083×10^{2}	5.9571×10^{2}	1.2610×10^{3}	2.8492×10^{2}	3.8696×10^{2}	5.9729×10^{2}	6.1502×10^{2}
F3	Aver	**8.9382×10^{2}**	6.7770×10^{5}	2.4620×10^{6}	2.0000×10^{5}	3.6185×10^{5}	7.5021×10^{5}	4.0553×10^{6}	2.1863×10^{5}
	Std	**3.1103×10^{1}**	1.9153×10^{5}	5.2040×10^{5}	1.3317×10^{5}	8.9355×10^{4}	1.7365×10^{5}	4.3072×10^{5}	8.9260×10^{4}
F4	Aver	**1.9091×10^{3}**	7.2983×10^{4}	3.1539×10^{6}	6.5979×10^{3}	2.1216×10^{4}	7.4912×10^{4}	1.3057×10^{7}	6.2360×10^{3}
	Std	**2.1044×10^{0}**	5.5654×10^{4}	2.0743×10^{6}	4.9138×10^{3}	2.8830×10^{4}	7.7577×10^{4}	6.7661×10^{6}	3.0949×10^{3}
F5	Aver	1.4295×10^{6}	1.4054×10^{7}	3.6562×10^{8}	5.2525×10^{5}	1.4090×10^{7}	3.1486×10^{7}	7.2531×10^{8}	8.5547×10^{6}
	Std	5.0199×10^{6}	1.7725×10^{7}	4.1172×10^{8}	5.1302×10^{5}	1.1348×10^{7}	1.4908×10^{7}	3.2382×10^{8}	8.6821×10^{6}
F6	Aver	5.2546×10^{4}	5.7230×10^{5}	1.1763×10^{9}	1.9343×10^{4}	1.2535×10^{6}	7.0495×10^{6}	3.6280×10^{9}	3.0296×10^{4}
	Std	4.1789×10^{3}	1.3277×10^{6}	9.3305×10^{8}	2.0774×10^{4}	1.3769×10^{6}	6.1776×10^{6}	2.3526×10^{9}	1.0489×10^{4}
F7	Aver	**8.2777×10^{4}**	4.3837×10^{7}	2.9654×10^{9}	7.8992×10^{5}	4.2414×10^{7}	9.4799×10^{7}	6.2065×10^{9}	7.7474×10^{6}
	Std	**4.5496×10^{5}**	2.8927×10^{7}	3.5719×10^{9}	1.0989×10^{6}	3.2201×10^{7}	6.0494×10^{7}	2.3851×10^{9}	9.3418×10^{6}
F8	Aver	**2.3917×10^{3}**	3.2953×10^{3}	6.6181×10^{3}	2.9386×10^{3}	2.5831×10^{3}	2.8921×10^{3}	8.9234×10^{3}	3.5628×10^{3}
	Std	1.8432×10^{2}	4.8089×10^{2}	1.4853×10^{3}	2.7841×10^{2}	7.6926×10^{1}	1.8915×10^{2}	1.2787×10^{3}	3.8432×10^{2}
F9	Aver	2.9767×10^{3}	1.8372×10^{4}	4.2171×10^{4}	1.3913×10^{4}	9.9682×10^{3}	1.7496×10^{4}	6.9814×10^{4}	1.4442×10^{4}
	Std	**3.2238×10^{1}**	4.4698×10^{3}	4.5064×10^{3}	6.2914×10^{3}	1.6047×10^{3}	3.5020×10^{3}	1.1377×10^{4}	4.8483×10^{3}
F10	Aver	**2.9229×10^{3}**	4.0523×10^{3}	9.5847×10^{3}	3.2405×10^{3}	3.7374×10^{3}	4.4690×10^{3}	2.2930×10^{4}	3.4542×10^{3}
	Std	**1.7062×10^{0}**	4.6772×10^{2}	2.3304×10^{3}	1.5500×10^{2}	2.5456×10^{2}	3.8571×10^{2}	5.6661×10^{3}	1.4278×10^{2}
	<=>	/	10/0/0	10/0/0	9/0/1	10/0/0	10/0/0	10/0/0	10/0/0

ID	Criteria	SRA	CEO	CGBPO	EMEPO
F1	Aver	7.9648×10^{10}	4.4482×10^{10}	3.9918×10^{8}	1.2778×10^{7}
	Std	1.3358×10^{10}	1.2249×10^{10}	1.8819×10^{9}	5.3773×10^{6}
F2	Aver	5.3523×10^{3}	7.3676×10^{3}	5.4036×10^{3}	3.3802×10^{3}
	Std	3.9420×10^{2}	6.2830×10^{2}	7.5260×10^{2}	6.7699×10^{2}
F3	Aver	2.5918×10^{6}	1.3957×10^{6}	1.0940×10^{3}	2.3389×10^{4}
	Std	2.8532×10^{5}	3.8721×10^{5}	9.3143×10^{4}	6.9204×10^{2}
F4	Aver	1.9275×10^{6}	6.4504×10^{5}	2.0232×10^{3}	2.0083×10^{3}
	Std	9.0055×10^{5}	4.2206×10^{5}	2.3097×10^{2}	2.7667×10^{1}
F5	Aver	1.0880×10^{8}	1.5006×10^{8}	**1.8749×10^{5}**	5.5760×10^{5}
	Std	6.1423×10^{7}	8.4302×10^{7}	2.7295×10^{5}	**3.0894×10^{5}**
F6	Aver	3.1894×10^{8}	1.3449×10^{8}	**9.8946×10^{3}**	2.7421×10^{4}
	Std	2.9621×10^{8}	1.4403×10^{8}	2.1029×10^{5}	1.2071×10^{4}
F7	Aver	7.0370×10^{8}	5.3457×10^{8}	6.4665×10^{5}	1.1669×10^{7}
	Std	4.2550×10^{8}	4.8327×10^{8}	1.1538×10^{6}	4.3918×10^{7}
F8	Aver	4.6937×10^{3}	4.0278×10^{3}	2.4463×10^{3}	2.4558×10^{3}
	Std	5.2881×10^{2}	6.2402×10^{2}	**3.1572×10^{1}**	4.4262×10^{1}
F9	Aver	4.5495×10^{4}	3.2244×10^{4}	4.8250×10^{3}	**2.6032×10^{3}**
	Std	6.1673×10^{3}	4.4920×10^{3}	2.2072×10^{3}	3.7304×10^{2}
F10	Aver	1.0573×10^{4}	5.9394×10^{3}	2.9788×10^{3}	3.0268×10^{3}
	Std	1.9465×10^{3}	8.4536×10^{2}	7.8673×10^{1}	5.0091×10^{1}
	<=>	10/0/0	10/0/0	8/0/2	9/0/1

Bold is the best result of all algorithms.

Table 7. Comparison results on CEC2021(CEC2021 F1–CEC2021 F10, dim = 20).

ID	Criteria	ECMPO	GWO	VOA	HOA	WaOA	ASEO	BSLO	EPO
F1	Aver	**4.7618 × 10³**	7.3414 × 10⁹	3.9674 × 10¹⁰	1.6214 × 10⁸	2.5876 × 10⁹	8.6267 × 10⁹	5.9481 × 10¹⁰	1.0746 × 10⁹
	Std	**2.4079 × 10³**	3.2697 × 10⁹	1.2412 × 10¹⁰	3.6556 × 10⁸	8.2298 × 10⁸	3.2678 × 10⁹	9.8554 × 10⁹	1.0459 × 10⁹
F2	Aver	**2.1916 × 10³**	4.2632 × 10³	5.7627 × 10³	3.5148 × 10³	5.0461 × 10³	5.5221 × 10³	6.9958 × 10³	3.7080 × 10³
	Std	**4.5921 × 10²**	4.8670 × 10²	5.1323 × 10²	5.8710 × 10²	3.5891 × 10²	2.5639 × 10²	3.7394 × 10²	5.3708 × 10²
F3	Aver	**7.5446 × 10²**	1.5751 × 10⁵	7.8007 × 10⁵	4.5188 × 10⁴	8.9345 × 10⁴	1.8947 × 10⁵	1.5455 × 10⁶	1.2222 × 10⁴
	Std	**8.4074 × 10⁰**	8.1207 × 10⁴	1.9367 × 10⁵	4.2264 × 10⁴	3.0895 × 10⁴	4.1774 × 10⁴	2.8222 × 10⁵	1.0837 × 10⁴
F4	Aver	**1.9357 × 10³**	8.7120 × 10³	3.0906 × 10⁶	2.5700 × 10³	2.8816 × 10³	1.6566 × 10⁴	9.6346 × 10⁶	2.6187 × 10³
	Std	**1.2457 × 10¹**	8.0606 × 10³	3.0783 × 10⁶	4.6901 × 10²	1.4169 × 10³	1.3605 × 10⁴	6.3245 × 10⁶	7.5088 × 10²
F5	Aver	6.9270 × 10⁵	4.0658 × 10⁶	3.1737 × 10⁷	3.3960 × 10⁵	3.4916 × 10⁶	4.9002 × 10⁶	1.4171 × 10⁸	2.7953 × 10⁶
	Std	7.9756 × 10⁵	3.7726 × 10⁶	1.9034 × 10⁷	5.0745 × 10⁵	2.1914 × 10⁶	2.8602 × 10⁶	6.7091 × 10⁷	2.2347 × 10⁶
F6	Aver	**1.2579 × 10⁴**	3.3960 × 10⁴	1.9916 × 10⁸	2.4799 × 10⁴	9.6424 × 10⁴	2.2607 × 10⁵	8.8238 × 10⁸	2.1272 × 10⁴
	Std	**6.5635 × 10³**	1.7958 × 10⁴	4.1658 × 10⁸	6.8327 × 10⁴	6.3838 × 10⁴	1.7581 × 10⁵	7.4485 × 10⁸	1.1000 × 10⁴
F7	Aver	**4.4515 × 10⁴**	1.9400 × 10⁷	8.1685 × 10⁸	9.3360 × 10⁵	1.2398 × 10⁷	1.5566 × 10⁷	2.0734 × 10⁹	3.3352 × 10⁶
	Std	**9.7891 × 10⁴**	3.0014 × 10⁷	1.0141 × 10⁹	1.3212 × 10⁶	8.8927 × 10⁶	8.7821 × 10⁶	1.3595 × 10⁹	2.8283 × 10⁶
F8	Aver	**2.3031 × 10³**	2.4488 × 10³	3.8505 × 10³	2.4346 × 10³	2.3727 × 10³	2.4204 × 10³	4.5288 × 10³	2.6826 × 10³
	Std	**8.6805 × 10⁰**	4.8494 × 10¹	1.0304 × 10³	4.7291 × 10¹	1.7905 × 10¹	3.3450 × 10¹	6.8080 × 10²	1.5911 × 10²
F9	Aver	**2.5863 × 10³**	9.8914 × 10³	2.6226 × 10⁴	5.0983 × 10³	6.1620 × 10³	1.0104 × 10⁴	4.0747 × 10⁴	6.0132 × 10³
	Std	**2.4857 × 10¹**	3.3170 × 10³	4.7722 × 10³	2.9514 × 10³	6.0441 × 10²	1.8940 × 10³	5.2423 × 10³	2.3297 × 10³
F10	Aver	**3.1693 × 10³**	4.7889 × 10³	1.3870 × 10⁴	3.4418 × 10³	3.8179 × 10³	5.2626 × 10³	1.9520 × 10⁴	4.3704 × 10³
	Std	**4.6648 × 10¹**	8.5401 × 10²	5.0975 × 10³	2.0896 × 10²	3.9076 × 10²	6.6808 × 10²	5.0297 × 10³	4.7310 × 10²
	<=>	/	10/0/0	10/0/0	10/0/0	10/0/0	10/0/0	10/0/0	10/0/0

ID	Criteria	SRA	CEO	CGBPO	EMEPO
F1	Aver	3.4889 × 10¹⁰	1.8888 × 10¹⁰	7.1631 × 10⁷	8.6004 × 10⁴
	Std	5.2467 × 10⁹	5.3453 × 10⁹	3.0612 × 10⁸	1.2604 × 10⁵
F2	Aver	3.2027 × 10³	4.3904 × 10³	3.0800 × 10³	2.7490 × 10³
	Std	3.0955 × 10²	5.3556 × 10²	4.3978 × 10²	1.0403 × 10³
F3	Aver	8.5571 × 10⁵	4.2048 × 10⁵	9.5025 × 10²	1.2222 × 10³
	Std	1.5759 × 10⁵	1.2677 × 10⁵	1.0914 × 10²	1.6590 × 10²
F4	Aver	7.3238 × 10⁵	2.7111 × 10⁵	2.4301 × 10³	2.0357 × 10³
	Std	4.3256 × 10⁵	2.6101 × 10⁵	2.6679 × 10³	5.5698 × 10¹
F5	Aver	9.3398 × 10⁶	1.6904 × 10⁷	1.1725 × 10⁵	**8.6158 × 10⁴**
	Std	8.6612 × 10⁶	1.0026 × 10⁷	5.0664 × 10⁴	**1.8621 × 10⁵**
F6	Aver	1.0890 × 10⁷	4.8063 × 10⁷	4.7111 × 10⁴	2.5127 × 10⁴
	Std	2.4464 × 10⁷	7.3411 × 10⁷	1.0427 × 10⁴	4.6162 × 10⁶
F7	Aver	1.3846 × 10⁸	2.2631 × 10⁸	3.8109 × 10⁵	6.4528 × 10⁶
	Std	1.0472 × 10⁸	3.4118 × 10⁸	6.9460 × 10⁴	6.4533 × 10⁵
F8	Aver	2.8680 × 10³	2.7774 × 10³	2.3429 × 10³	2.3327 × 10³
	Std	1.6562 × 10²	1.7264 × 10²	1.4533 × 10¹	4.8007 × 10¹
F9	Aver	2.2634 × 10⁴	1.7302 × 10⁴	3.3396 × 10³	2.6083 × 10³
	Std	3.3114 × 10³	2.8820 × 10³	1.0081 × 10³	1.6093 × 10²
F10	Aver	9.7156 × 10³	8.2547 × 10³	4.2179 × 10³	3.3749 × 10³
	Std	1.8058 × 10³	1.8769 × 10³	7.1613 × 10¹	1.1524 × 10²
	<=>	10/0/0	10/0/0	10/0/0	9/0/1

Bold is the best result of all algorithms.

Table 8. Comparison results on CEC2022(CEC2022 F1–CEC2022 F12, dim = 20).

ID	Criteria	ECMPO	GWO	VOA	HOA	WaOA	ASEO	BSLO	EPO
F1	Aver	**3.9865 × 10²**	1.2872 × 10⁴	6.1059 × 10⁴	1.9266 × 10³	9.7707 × 10³	1.4010 × 10⁴	8.6498 × 10⁴	4.0469 × 10³
	Std	**4.1521 × 10²**	3.8891 × 10³	3.0966 × 10⁴	1.9597 × 10³	3.6638 × 10³	4.4581 × 10³	2.5518 × 10⁴	2.4307 × 10³
F2	Aver	**5.0074 × 10²**	8.0591 × 10²	4.1695 × 10³	5.1830 × 10²	7.0735 × 10²	9.7353 × 10²	8.7171 × 10³	6.0199 × 10²
	Std	**4.4459 × 10¹**	1.4525 × 10²	1.3706 × 10³	5.3654 × 10¹	1.2076 × 10²	1.9523 × 10²	2.2722 × 10³	4.0918 × 10¹
F3	Aver	**6.0000 × 10²**	6.0030 × 10²	6.0211 × 10²	6.0000 × 10²	6.0010 × 10²	6.0044 × 10²	6.0273 × 10²	6.0011 × 10²
	Std	**2.1860 × 10⁻⁶**	1.5276 × 10⁻¹	5.4223 × 10⁻¹	6.9797 × 10⁻³	3.4182 × 10⁻²	1.6103 × 10⁻¹	4.0062 × 10⁻¹	9.0574 × 10⁻²
F4	Aver	9.5856 × 10²	1.0103 × 10³	1.1150 × 10³	9.9942 × 10²	1.1090 × 10³	1.0744 × 10³	1.3933 × 10³	9.8280 × 10²
	Std	4.8979 × 10¹	3.4542 × 10¹	5.7150 × 10¹	4.7408 × 10¹	4.0113 × 10¹	3.2860 × 10¹	6.2131 × 10¹	4.5233 × 10¹
F5	Aver	**4.4770 × 10²**	9.0560 × 10²	9.1329 × 10²	9.0319 × 10²	9.0682 × 10²	9.0569 × 10²	9.3318 × 10²	9.0305 × 10²
	Std	**8.0899 × 10⁻¹**	2.2785 × 10⁰	4.3805 × 10⁰	1.5984 × 10⁰	2.6850 × 10⁰	1.7478 × 10⁰	7.1224 × 10⁰	1.2222 × 10⁰
F6	Aver	4.3688 × 10⁴	9.6983 × 10⁷	5.4650 × 10⁹	4.7101 × 10⁴	7.2309 × 10⁷	2.4648 × 10⁸	9.6853 × 10⁹	5.3367 × 10⁴
	Std	1.7362 × 10⁴	1.4974 × 10⁸	3.1910 × 10⁹	1.5764 × 10⁴	4.5101 × 10⁷	1.1723 × 10⁸	3.3805 × 10⁹	2.3383 × 10⁴
F7	Aver	**2.2537 × 10³**	2.7039 × 10³	8.0245 × 10³	2.7557 × 10³	2.9372 × 10³	2.7685 × 10³	1.3426 × 10⁴	2.9568 × 10³
	Std	1.2605 × 10²	4.8814 × 10²	4.7049 × 10³	3.9036 × 10²	3.7334 × 10²	2.3493 × 10²	3.5910 × 10³	3.6752 × 10²
F8	Aver	**9.0813 × 10²**	4.7858 × 10³	6.8761 × 10¹³	3.9336 × 10³	2.7089 × 10⁵	1.9077 × 10⁶	2.9738 × 10¹⁴	3.9013 × 10³
	Std	9.7777 × 10²	1.3406 × 10³	3.2894 × 10¹⁴	8.7074 × 10²	8.7756 × 10⁵	3.3084 × 10⁶	8.1091 × 10¹⁴	1.0992 × 10³
F9	Aver	**2.1827 × 10³**	3.1068 × 10³	6.8103 × 10³	2.6902 × 10³	2.8459 × 10³	3.2387 × 10³	9.5241 × 10³	3.0881 × 10³
	Std	**7.3354 × 10¹**	1.8636 × 10²	2.5849 × 10³	3.6323 × 10¹	9.8751 × 10¹	1.9273 × 10²	2.1704 × 10³	1.9275 × 10²
F10	Aver	3.1077 × 10³	5.0554 × 10³	6.7296 × 10³	3.9944 × 10³	**2.8601 × 10³**	2.9277 × 10³	8.1591 × 10³	4.6286 × 10³
	Std	8.0055 × 10²	1.2182 × 10³	1.0935 × 10³	1.2852 × 10³	**4.1582 × 10¹**	9.9648 × 10¹	8.1792 × 10²	1.0450 × 10³
F11	Aver	**2.2427 × 10³**	3.5009 × 10³	1.7047 × 10⁴	2.8355 × 10³	3.2864 × 10³	2.9896 × 10³	2.3900 × 10⁴	2.9278 × 10³
	Std	**1.1703 × 10²**	7.1639 × 10²	8.5173 × 10³	8.1358 × 10²	7.7467 × 10²	3.7573 × 10²	1.0500 × 10⁴	3.4753 × 10²
F12	Aver	3.0852 × 10³	3.1784 × 10³	3.6219 × 10³	3.1681 × 10³	3.0071 × 10³	3.3094 × 10³	4.3386 × 10³	3.6231 × 10³
	Std	7.6671 × 10¹	1.1949 × 10²	3.3006 × 10²	1.2811 × 10²	**3.4324 × 10¹**	1.0966 × 10²	3.3495 × 10²	2.7203 × 10²

Table 8. *Cont.*

ID	Criteria	ECMPO	GWO	VOA	HOA	WaOA	ASEO	BSLO	EPO
	<=>	/	12/0/0	12/0/0	12/0/0	11/0/1	11/0/1	12/0/0	12/0/0
ID	**Criteria**	**SRA**		**CEO**		**CGBPO**		**EMEPO**	
F1	Aver	3.7950×10^{4}		2.5414×10^{4}		9.1345×10^{3}		3.8346×10^{3}	
	Std	7.5705×10^{3}		8.6668×10^{3}		3.3042×10^{2}		3.8728×10^{3}	
F2	Aver	4.3023×10^{3}		5.0738×10^{3}		4.3088×10^{3}		3.4435×10^{3}	
	Std	5.2299×10^{2}		1.2533×10^{3}		9.6685×10^{2}		6.3214×10^{2}	
F3	Aver	7.9365×10^{3}		6.9613×10^{3}		2.9285×10^{3}		2.6714×10^{3}	
	Std	2.4543×10^{3}		2.8340×10^{3}		5.4069×10^{2}		3.9075×10^{2}	
F4	Aver	3.5983×10^{3}		3.8961×10^{3}		2.9955×10^{3}		$\mathbf{8.4183 \times 10^{2}}$	
	Std	1.4178×10^{2}		2.2743×10^{2}		$\mathbf{2.7377 \times 10^{1}}$		3.3275×10^{2}	
F5	Aver	3.3234×10^{3}		1.8353×10^{3}		9.0929×10^{2}		9.0228×10^{2}	
	Std	7.7121×10^{2}		7.1760×10^{2}		3.5611×10^{1}		3.8570×10^{0}	
F6	Aver	6.0160×10^{2}		6.0114×10^{2}		6.0001×10^{2}		$\mathbf{6.0000 \times 10^{2}}$	
	Std	2.7663×10^{-1}		2.3250×10^{-1}		5.6678×10^{-2}		$\mathbf{1.3418 \times 10^{-7}}$	
F7	Aver	1.1612×10^{3}		1.0468×10^{3}		3.0554×10^{3}		9.3348×10^{3}	
	Std	5.5133×10^{1}		4.4398×10^{1}		6.5141×10^{1}		$\mathbf{2.9129 \times 10^{1}}$	
F8	Aver	9.1277×10^{2}		9.0807×10^{2}		3.2508×10^{3}		9.6161×10^{2}	
	Std	2.5748×10^{0}		2.8358×10^{0}		2.9546×10^{0}		$\mathbf{6.7328 \times 10^{-1}}$	
F9	Aver	2.5998×10^{9}		1.9897×10^{9}		2.6715×10^{3}		8.0720×10^{4}	
	Std	1.2367×10^{9}		1.4677×10^{9}		2.1692×10^{4}		7.5574×10^{7}	
F10	Aver	4.7112×10^{3}		4.4315×10^{3}		4.3282×10^{3}		3.3308×10^{3}	
	Std	1.0083×10^{3}		1.5394×10^{3}		1.0535×10^{2}		4.4979×10^{2}	
F11	Aver	8.1140×10^{8}		2.1039×10^{12}		2.5918×10^{3}		2.4732×10^{3}	
	Std	2.4433×10^{9}		9.6160×10^{12}		6.3150×10^{2}		2.4270×10^{2}	
F12	Aver	5.0007×10^{3}		5.4934×10^{3}		$\mathbf{2.6454 \times 10^{3}}$		3.4873×10^{3}	
	Std	8.2617×10^{2}		8.2475×10^{2}		2.2440×10^{2}		9.1901×10^{2}	
	<=>	12/0/0		12/0/0		11/0/1		10/0/2	

Bold is the best result of all algorithms.

Table 6 presents the experimental results obtained on the 30-dimensional CEC2020 benchmark suite. As shown in the table, ECMPO achieves the best mean fitness values on seven benchmark functions (F1, F2, F3, F4, F7 and F10), whereas CGBPO performs best on F5 and F6 and EMEPO performs best on F9. The remaining algorithms do not obtain the best results on any of the benchmark functions. This indicates that ECMPO demonstrates superior search capability and higher solution accuracy on most benchmark functions. Further examination of the mean fitness values reveals that ECMPO significantly outperforms the competing algorithms on the unimodal function F1, with differences reaching several orders of magnitude. For example, the mean fitness value obtained by ECMPO is significantly lower than those of algorithms such as GWO, VOA, and WaOA, indicating stronger global search capability in the solution space. In addition, CGBPO and EMEPO slightly outperform ECMPO on functions F5, F6 and F9, suggesting that they have certain advantages when dealing with hybrid function structures. Overall, ECMPO achieves the best results on seven out of the 10 benchmark functions and outperforms most competing algorithms in terms of overall optimization performance and stability, demonstrating strong competitiveness on the CEC2020 benchmark suite.

Table 7 presents the experimental results obtained on the 20-dimensional CEC2021 benchmark suite. As shown in the table, ECMPO achieves the best mean fitness values on nine out of the ten benchmark functions (F1, F2, F3, F4, F6, F7, F8, F9, and F10) and performs slightly worse than EMEPO only on F5. These results indicate that ECMPO demonstrates significant advantages on the vast majority of benchmark functions, indicating higher optimization accuracy and stronger global search capability.

Table 8 presents the experimental results obtained on the 20-dimensional CEC2022 benchmark suite. Compared with the other algorithms, EMEPO obtains the best results on functions F4 and F6. CGBPO obtains the best results on function F12. ECMPO outperforms the competing algorithms on nine benchmark functions in the CEC2022 suite, indicating that ECMPO can effectively maintain a balance between global exploration and local exploitation in complex search spaces, thereby improving convergence accuracy.

3.4. Statistical Analysis

To further analyze statistical differences among the compared algorithms, the Friedman test was applied to results obtained from the CEC2020, CEC2021, and CEC2022 benchmark suites. The mean rank of each algorithm was subsequently computed, and the outcomes are summarized in Table 9. The Friedman test reveals statistically significant differences among the evaluated methods. Based on the average ranking, ECMPO attains the lowest mean rank across all three benchmark suites, indicating its superior overall optimization capability relative to the competing algorithms and demonstrating strong search effectiveness and robustness.

Table 9. Friedman average rank.

Algorithms	CEC 2020	CEC 2021	CEC 2022
GWO	6.1433	6.4877	6.6854
VOA	10.6211	10.0433	10.9444
HOA	4.3267	4.4600	4.5112
WaOA	5.8112	5.8112	5.9639
ASEO	7.6300	7.3833	7.0389
BSLO	11.3223	11.8700	11.8611
EPO	5.9000	5.3118	5.1389
SAR	9.0223	9.4967	9.5639
CEO	8.1000	8.4900	8.3889
CGBPO	2.3700	2.4500	3.2136
EMEPO	2.7000	2.9067	2.5311
ECMPO	1.5322	1.6233	1.5766

In addition, the Wilcoxon rank-sum test was performed to compare ECMPO with the eleven competing algorithms on the CEC2020, CEC2021, and CEC2022 benchmark suites based on 30 independent runs. In Tables 10–12, the symbol "+" indicates that ECMPO performs better than the compared algorithm, "=" indicates comparable performance, and "−" indicates that the compared algorithm outperforms ECMPO.

Table 10 presents the Wilcoxon rank-sum test results comparing ECMPO with eleven benchmark algorithms on the ten benchmark functions of the CEC2020 test suite. The statistical results indicate that ECMPO achieves superior performance in 98 out of 110 comparisons (98/110), demonstrating a clear advantage over the other algorithms.

Table 10. Results of the Wilcoxon rank-sum test on the CEC2020 benchmark functions.

ID	GWO	VOA	HOA	WaOA	ASEO	BSLO	EPO	SRA	CEO	CGBPO	EMEPO
F1	+	+	+	+	+	+	+	+	+	+	+
F2	+	+	+	+	+	+	+	+	+	+	+
F3	+	+	+	+	+	+	+	+	+	+	+
F4	+	+	+	+	+	+	+	+	+	=	=
F5	+	+	=	+	+	+	+	+	+	−	+
F6	+	+	−	+	+	+	=	+	+	−	−
F7	+	+	=	+	+	+	+	+	+	+	+
F8	+	+	+	+	+	+	+	+	+	=	=
F9	+	+	+	+	+	+	+	+	+	+	−
F10	+	+	+	+	+	+	+	+	+	+	+
+/=/−	10/0/0	10/0/0	7/2/1	10/0/0	10/0/0	10/0/0	9/1/0	10/0/0	10/0/0	6/2/2	6/2/2

Table 11. Results of the Wilcoxon rank-sum test on the CEC2021 benchmark functions.

ID	GWO	VOA	HOA	WaOA	ASEO	BSLO	EPO	SRA	CEO	CGBPO	EMEPO
F1	+	+	+	+	+	+	+	+	+	+	+
F2	+	+	+	+	+	+	+	+	+	+	+
F3	+	+	+	+	+	+	+	+	+	+	+
F4	+	+	+	+	+	+	+	+	+	=	=
F5	+	+	=	+	+	+	+	+	+	+	−
F6	+	+	=	+	+	+	+	+	+	+	+
F7	+	+	=	+	+	+	+	+	+	+	+
F8	+	+	+	+	+	+	+	+	+	=	=
F9	+	+	+	+	+	+	+	+	+	+	=
F10	+	+	=	+	+	+	+	+	+	+	+
+/=/−	10/0/0	10/0/0	6/4/0	10/0/0	10/0/0	10/0/0	10/0/0	10/0/0	10/0/0	8/2/0	6/3/1

Table 12. Results of the Wilcoxon rank-sum test on the CEC2022 benchmark functions.

ID	GWO	VOA	HOA	WaOA	ASEO	BSLO	EPO	SRA	CEO	CGBPO	EMEPO
F1	+	+	+	+	+	+	+	+	+	+	+
F2	+	+	=	+	+	+	+	+	+	=	+
F3	+	+	+	+	+	+	+	+	+	=	=
F4	+	+	+	+	+	+	+	+	+	+	−
F5	+	+	+	+	+	+	+	+	+	+	+
F6	+	+	=	+	+	+	−	+	+	+	+
F7	+	+	=	+	+	+	+	+	+	+	+
F8	+	+	=	+	+	+	=	+	+	=	=
F9	+	+	−	+	+	+	+	+	+	+	+
F10	+	+	+	+	=	+	+	+	+	+	+
F11	+	+	+	+	+	+	+	+	+	+	+
F12	+	+	+	−	+	+	+	+	+	−	+
+/=/−	12/0/0	12/0/0	8/3/1	11/0/1	11/1/0	12/0/0	10/1/1	12/0/0	12/0/0	8/3/1	9/2/1

Table 11 summarizes the Wilcoxon rank-sum test results obtained from the ten benchmark functions of the CEC2021 test suite. The results indicate that ECMPO performs comparably to HOA on F5, F6, F7, and F10. On F4 and F8, ECMPO shows performance comparable to CGBPO and EMEPO, and on F9, it is comparable to EMEPO. For the remaining functions, ECMPO outperforms all other competing algorithms. Among the 110 total comparisons, ECMPO again achieves 100 superior results (100/110), further indicating its strong overall optimization capability.

Table 12 presents the Wilcoxon rank-sum test results comparing ECMPO with other algorithms on the twelve benchmark functions of the CEC2022 test suite. The results indicate that ECMPO performs similarly to HOA on functions F2, F6, F7, and F8 and shows comparable performance to ASEO on F10. Furthermore, it demonstrates similar performance to CGBPO on F2, F3, and F8 and to EMEPO on F3 and F8. However, ECMPO performs slightly worse than HOA and WaOA on functions F9 and F12, worse than CGBPO on F12, and worse than EMEPO on F4. Out of 132 total comparisons, ECMPO achieves 117 superior results (117/132), indicating that its overall performance on the CEC2022 test functions remains better than that of most comparison algorithms. Overall, considering the Wilcoxon rank-sum test results across CEC2020, CEC2021, and CEC2022, ECMPO demonstrates clear superiority in the majority of comparisons, further verifying its stability and effectiveness across different types of optimization problems.

3.5. Convergence Performance Analysis

Convergence analysis plays a critical role in evaluating whether an algorithm can efficiently approach the global optimum while maintaining an appropriate balance between exploration and exploitation. Figures 3–5 present the convergence curves of ECMPO compared with other algorithms, providing a visual comparison of their convergence behavior, optimization accuracy, and stability.

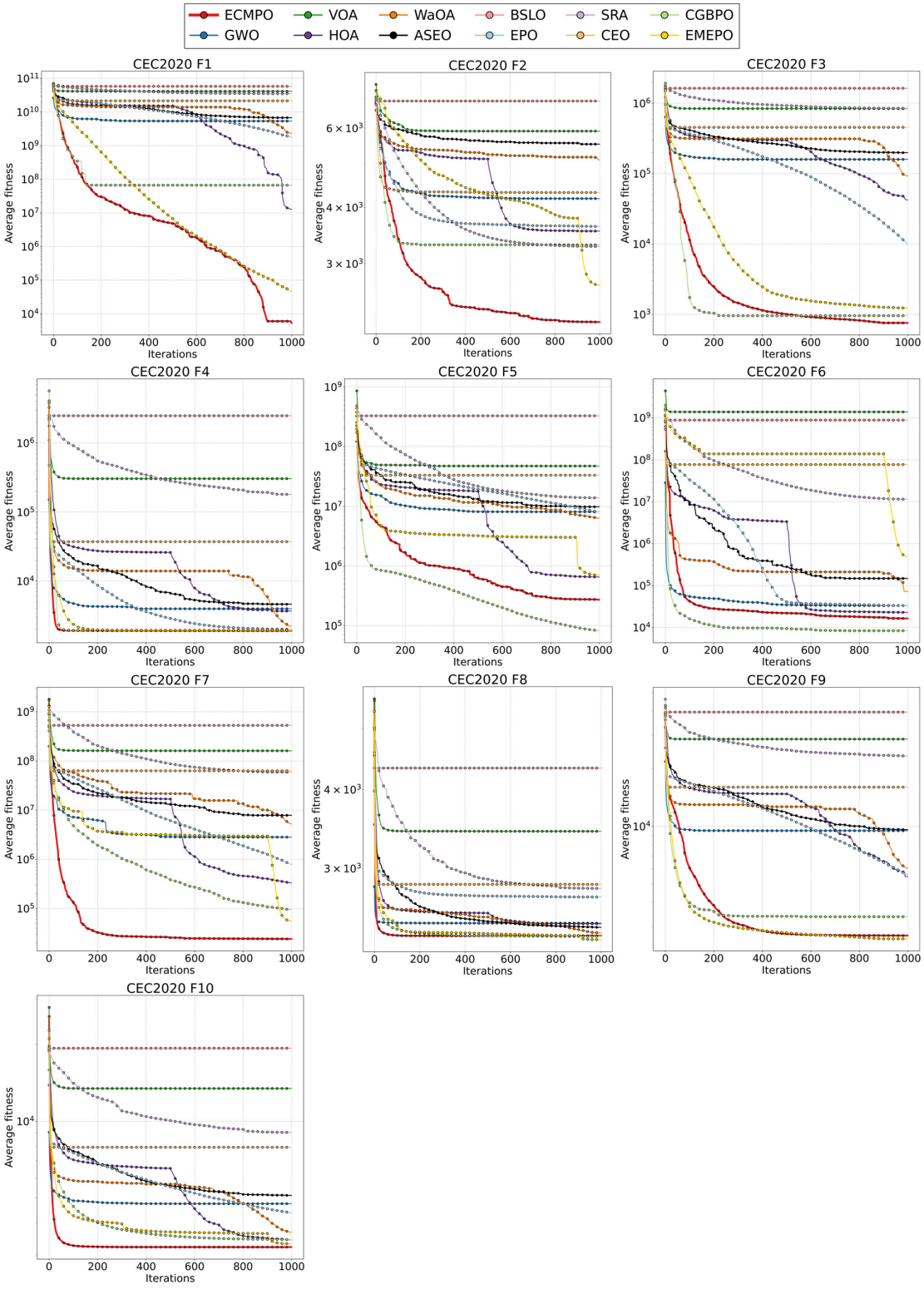


Figure 3. Convergence curves obtained by different algorithms on the CEC2020 benchmark suite.

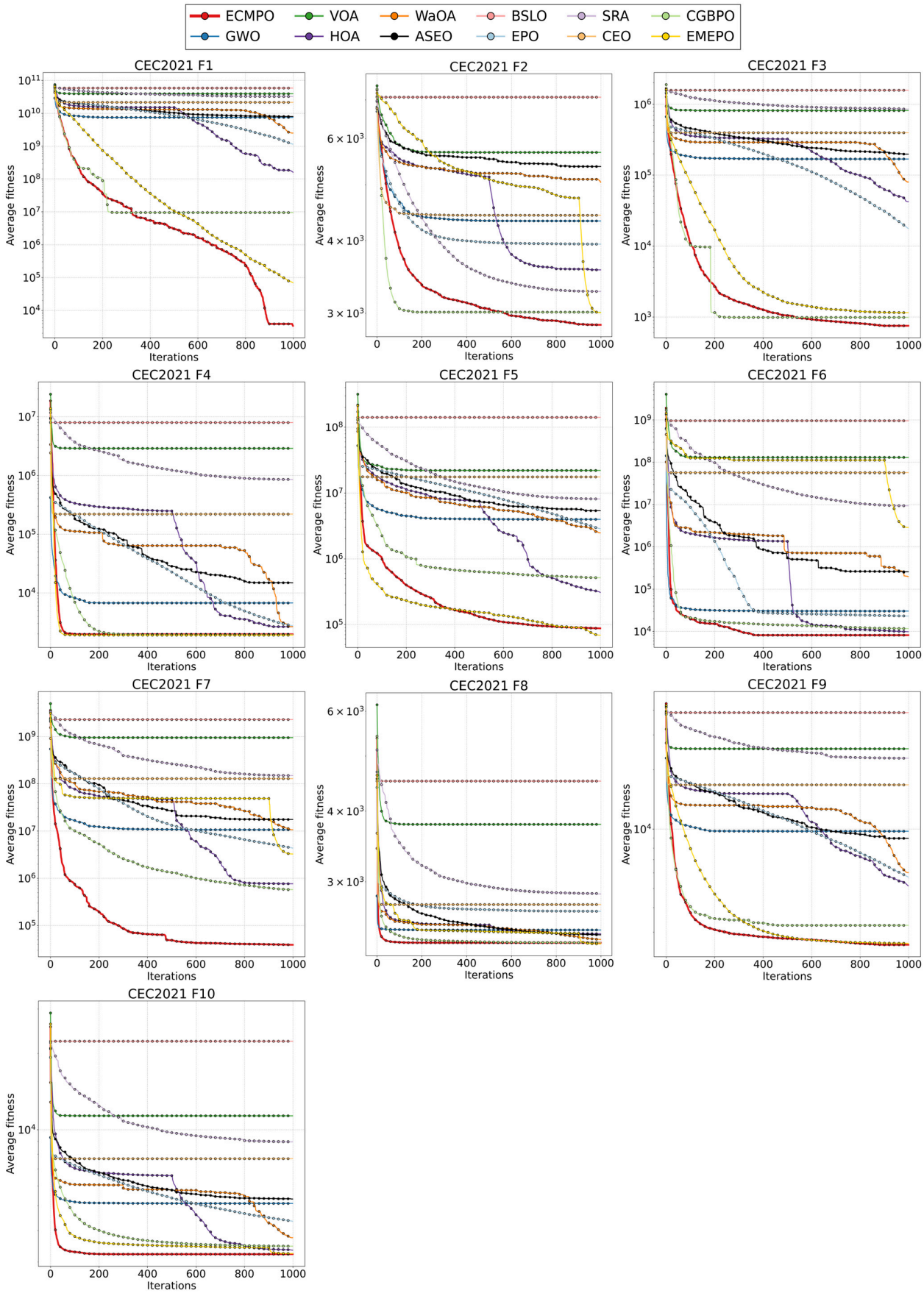


Figure 4. Convergence curves obtained by different algorithms on the CEC2021 benchmark suite.

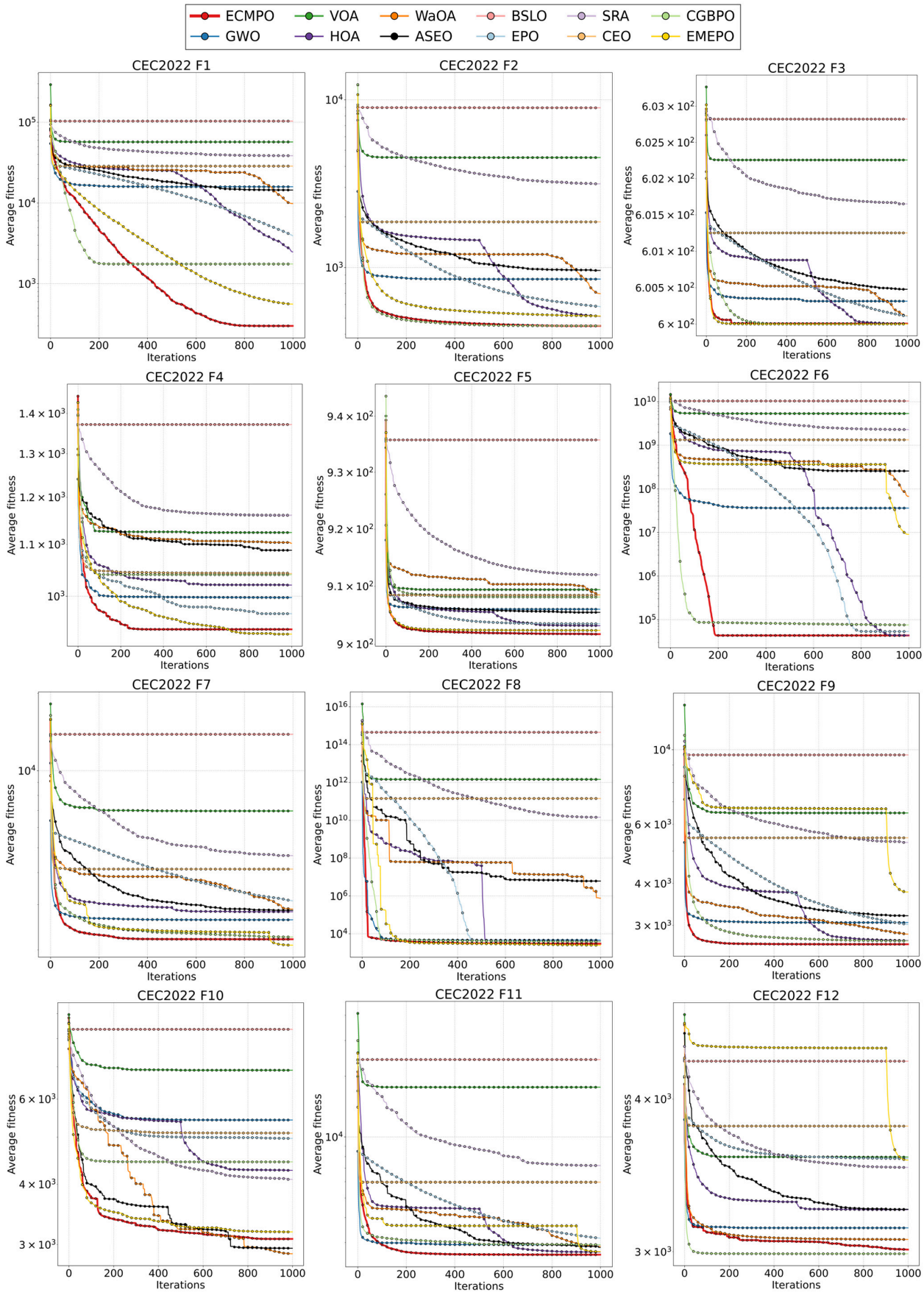


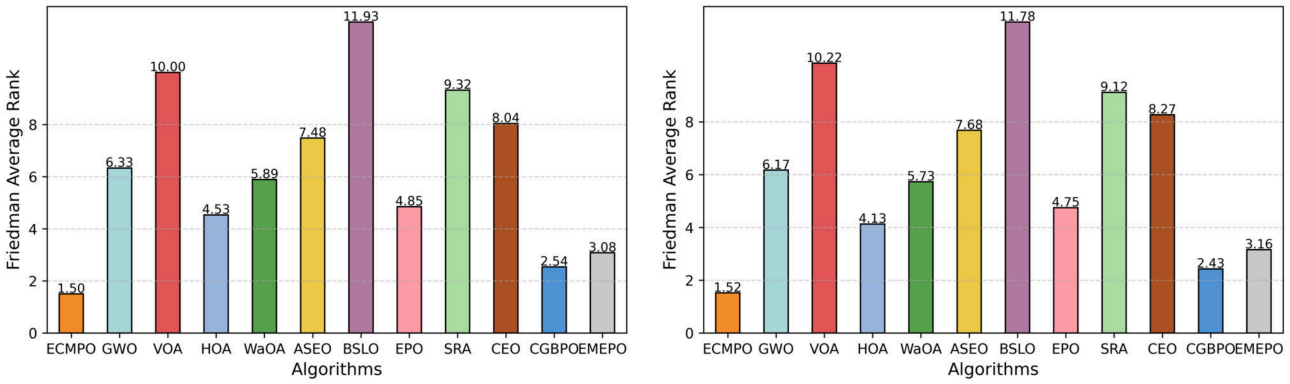
Figure 5. Convergence curves obtained by different algorithms on the CEC2022 benchmark suite.

Overall, ECMPO exhibits superior convergence speed and higher optimization accuracy compared with the competing algorithms. For unimodal functions, such as CEC2020 F1, CEC2021 F1, and CEC2022 F1, ECMPO consistently converges faster than the other swarm intelligence algorithms and rapidly approaches the global optimum, demonstrating strong exploitation capability in fine-grained local search. For multimodal functions, including CEC2020 F2–F4, CEC2021 F2–F4, and CEC2022 F2–F5, ECMPO demonstrates strong global exploration capability and effectively avoids premature convergence to local optima, a limitation commonly observed in many competing algorithms. In composite and hybrid functions, which involve more complex landscapes and interactions among variables, ECMPO still maintains high solution quality and efficient convergence, further confirming its robustness and effectiveness across diverse optimization scenarios.

3.6. High-Dimensional Analysis

As problem dimensionality increases, the search space expands exponentially, which substantially increases the complexity of optimization tasks. Many algorithms exhibit strong performance in low-dimensional settings but tend to experience a marked decline in effectiveness when applied to high-dimensional problems, a phenomenon known as the "curse of dimensionality." Therefore, experimental evaluations on high-dimensional benchmark functions are crucial for assessing the robustness and scalability of optimization algorithms.

To further evaluate the performance of ECMPO on high-dimensional optimization problems, comparative experiments were conducted on the CEC2020 benchmark suite with dimensional settings of 50 and 100. Figure 6 presents the average ranking results obtained from the Friedman test. It can be observed that ECMPO achieves average ranks of 1.50 and 1.52 under the 50-dimensional and 100-dimensional settings, respectively, ranking first among all comparison algorithms in both cases. These results indicate that ECMPO maintains stable and superior optimization performance as the dimensionality increases, demonstrating strong capability in solving high-dimensional problems and good algorithm scalability.



(**a**) (**b**)

Figure 6. (**a**) Friedman average rankings on the CEC2020 benchmark functions at 50 dimensions. (**b**) Friedman average rankings on the CEC2020 benchmark functions at 100 dimensions.

4. ECMPO Applications in Engineering Optimization Problems

In this section, the performance of the ECMPO algorithm is assessed on six representative real-world engineering optimization problems, namely the Pressure Vessel Design (PVD) [42], Robot Gripper Optimization Problem (GROP) [43], Tubular Column Design (TCD) [44], Cantilever Beam Design (CBD) [45], Gear Train Design (GTD) [46], and Reducer Design Problem (RDP) [47]. Compared with standard numerical benchmark functions, these problems are characterized by the need to achieve optimality while simultaneously satisfying multiple strict constraints.

To address constraint violations, various constraint-handling strategies have been proposed in the literature, such as static penalty, dynamic penalty, annealing-based penalty, adaptive penalty, and co-evolutionary methods. In this study, a static penalty approach is adopted due to its simplicity and effectiveness.

Accordingly, the constrained optimization problem is transformed into an unconstrained one using the following penalized objective function in Equation (18).

$$F(x) = f(x) + \sum_{i=1}^{m} \left(R_i \cdot max[0, g_i(x)]^2\right) \tag{18}$$

where $f(x)$ represents the original objective function, $g_i(x)$ represents the i-th constraint function, m refers to the total number of constraints, and R_i are the penalty coefficients.

For comparative evaluation, ECMPO is tested against several well-established metaheuristic algorithms, including GWO, VOA, HOA, WaOA, ASEO, BSLO, EPO, SRA CEO, CGBPO, and EMEPO. The population size of all algorithms is uniformly set to 30, while the maximum iteration number is fixed at 300. To mitigate the influence of randomness, each algorithm is run independently 30 times. The performance measures include the mean value (Aver), standard deviation (Std), best value (Best), worst value (Worst), and the corresponding rank (Rank).

4.1. Pressure Vessel Design (PVD)

The main objective of the Pressure Vessel Design (PVD) problem is to minimize the manufacturing cost of the pressure vessel while satisfying functional and safety requirements. The problem involves four design variables, namely the shell thickness (T_s), head thickness (T_h), inner radius (R), and the length of the cylindrical section excluding the head (L). The schematic diagram of the structure is shown in Figure 7, and the corresponding mathematical model can be expressed as follows:

$$\begin{aligned}
&\text{Consider:} && \vec{x} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix} = \begin{bmatrix} T_s & T_h & R & L \end{bmatrix} \\
&\text{Minimize:} && f\left(\vec{x}\right) = 0.6224 x_1 x_3 x_4 + 1.7781 x_2 x_3^2 + 3.1661 x_1^2 x_4 + 19.84 x_1^2 x_3, \\
&\text{Subject to:} && g_1\left(\vec{x}\right) = -x_1 + 0.0193 x_3 \le 0, \\
& && g_2\left(\vec{x}\right) = -x_2 + 0.00954 x_3 \le 0, \\
& && g_3\left(\vec{x}\right) = -\pi x_3^2 x_4 - \tfrac{4}{3}\pi x_3^3 1296000 \le 0, \\
& && g_4\left(\vec{x}\right) = x_4 - 240 \le 0, \\
&\text{Parameter range:} && 0 \le x_1, x_2 \le 99, \cdots\cdots 10 \le x_3, x_4 \le 200.
\end{aligned}$$

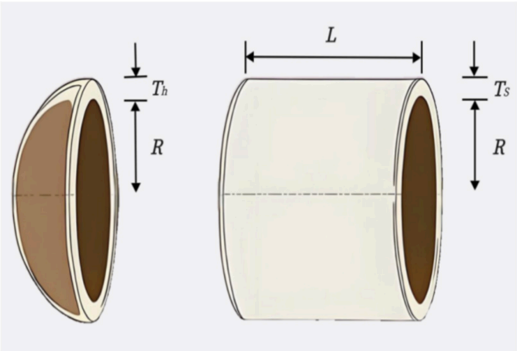


Figure 7. PVD diagram.

Table 13 presents the comparison results for the PVD problem. It can be observed that ECMPO achieves the lowest cost among all compared algorithms.

Table 13. Comparison results for the PVD problem.

Algorithm	Optimal Design Variables				Aver	Std	Best	Worst	Rank
	T_s	T_h	R	L					
ECMPO	0.8125	0.4375	42.2649	174.5846	6.5194×10^3	4.4366×10^2	6.0499×10^3	7.3814×10^3	1
GWO	1.1250	0.5625	56.8805	52.1517	4.6524×10^4	5.0056×10^4	6.9503×10^3	2.4191×10^5	6
VOA	0.8750	0.4375	44.8146	145.6549	2.1794×10^4	2.5971×10^4	6.1510×10^3	1.2401×10^5	4
HOA	1.0625	0.5625	54.4210	150.6048	1.5877×10^5	1.0497×10^5	1.0139×10^4	4.1427×10^5	10
WaOA	0.8125	0.3750	40.7651	193.8899	5.5701×10^4	1.6077×10^5	6.2375×10^3	8.7923×10^5	7
ASEO	1.3750	1.6875	48.3730	162.5112	3.2668×10^5	2.0483×10^5	1.6536×10^4	1.0041×10^6	11
BSLO	1.0000	0.5000	49.8210	99.8434	2.8760×10^4	4.4567×10^4	6.6073×10^3	1.8002×10^5	5
EPO	3.9375	1.1250	68.0905	17.6691	5.9081×10^5	6.7500×10^5	3.4035×10^4	3.6287×10^6	12
SRA	0.8125	0.4375	42.7512	168.7119	1.1906×10^5	1.2335×10^5	6.1405×10^3	3.5733×10^5	9
CEO	1.0000	3.6875	46.2703	200.0000	8.3464×10^4	5.3582×10^4	2.1349×10^4	2.7129×10^5	8
CGBPO	1.0000	1.0000	52.0915	82.6865	1.3515×10^4	6.0823×10^3	8.8299×10^3	3.0748×10^4	2
EMEPO	1.0625	0.5000	51.1938	99.2472	1.8918×10^4	1.3465×10^4	7.1913×10^3	7.0402×10^4	3

4.2. Robot Gripper Optimization Problem (GROP)

The Robot Gripper Optimization problem aims to minimize the difference between the maximum and minimum gripping forces within a specified displacement range of the gripper tip. The schematic diagram of the robot gripper is shown in Figure 8. In this model, a, b and c are the link lengths of the various links of the mechanism, α, β,and δ are the angles that define the geometric relationships between the links of the mechanism, e is the offset of point C from point A perpendicular to the direction of manipulator displacement, F_k is the gripping force applied by the mechanism on the object to be gripped, z is the manipulator displacement, and l is the distance of point H from the actuator end. The mathematical model of the GROP problem can be expressed as follows:

Consider: $\vec{x} = \left[x_1\ x_2\ \ x_3\ \ x_4\ \ x_5\ \ x_6\ \ x_7\right] = \left[\ a\ b\ c\ e\ f\ l\ \delta\right]$

Minimize: $f(\overline{x}) = -minF_k(x,z) + maxF_k(x,z)$

Subject to:

$$g_1(\overline{x}) = -Y_{min} + y\left(\left(\vec{x}\right), Z_{max}\right) \le 0,$$
$$g_2(\overline{x}) = -y\left(\left(\vec{x}\right), Z_{max}\right) \le 0,$$
$$g_3(\overline{x}) = Y_{max} - y\left(\left(\vec{x}\right), 0\right) \le 0,$$
$$g_4(\overline{x}) = y\left(\left(\vec{x},\right), 0\right) - Y_G \le 0,$$
$$g_5(\overline{x}) = l^2 + e^2 - (a+b)^2 \le 0,$$
$$g_6(\overline{x}) = b^2 - (a-e)^2 - (l - Z_{max})^2 \le 0,$$
$$g_7(\overline{x}) = Z_{max} - l \le 0,$$

where

$$\alpha = cos^{-1}\left(\frac{a^2+g^2-b^2}{2ag}\right) + \phi, g = \sqrt{e^2 + (z-l)^2},$$
$$\beta = cos^{-1}\left(\frac{b^2+g^2-a^2}{2bg}\right) - \phi, \phi = tan^{-1}\left(\frac{e}{l-z}\right),$$
$$y(x,z) = 2(f + e + csin(\beta + \delta)),$$
$$F_k = \frac{Pbsin(\alpha+\beta)}{2ccos(\alpha)},\ Y_{min} = 50,\ Y_{max} = 100,\ Y_G = 150,\ Z_{max} = 100,\ P = 100,$$

With bounds: $0 \le e \le 50,\ 100 \le c \le 200,\ 10 \le f, a, b \le 150,\ 1 \le \delta \le 3.14,\ 100 \le L \le 300$

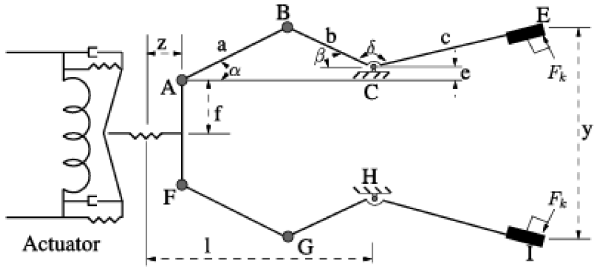


Figure 8. GROP diagram.

The optimization results of different GROP algorithms are shown in Table 14. The results show that the ECMPO algorithm is better than the other eleven algorithms.

Table 14. Comparison results of GROP problem.

Algorithm	Optimal Design Variables							Aver	Std	Best	Worst	Rank
	x_1	x_2	x_3	x_4	x_5	x_6	x_7					
ECMPO	122.4435	10.0000	200.0000	50.0000	132.7003	119.1162	2.6580	4.8482	1.5183	4.4662	5.0450	1
GWO	144.0167	84.9710	134.5603	35.4259	83.7046	180.4498	2.6973	13.1223	3.1421	11.0319	15.2432	12
VOA	149.9761	43.2794	175.1532	27.2210	146.8511	191.3287	2.6883	9.2332	2.8100	8.5259	11.8526	6
HOA	150.0000	99.4549	197.2028	28.0538	147.8372	192.9368	2.7469	10.7550	2.2325	8.7036	11.7435	11
WaOA	122.5831	11.7293	194.6981	49.9997	96.1206	118.3053	2.4124	8.5420	2.9653	6.5545	9.1329	4
ASEO	150.0000	41.5326	120.9059	19.0128	94.5677	188.8497	2.3300	9.4500	1.4298	8.8418	10.3564	7
BSLO	116.4612	42.3062	149.3149	42.7387	116.8214	105.0830	2.7245	10.3321	2.3362	9.5326	12.2258	10
EPO	116.9830	34.0759	181.1639	44.8692	58.2042	115.3345	2.2330	8.7623	2.1776	7.0996	10.3123	5
SRA	120.9055	24.0089	171.1653	46.4564	28.5195	134.8827	2.0869	10.4662	1.2023	9.0016	13.1931	9
CEO	144.9137	102.9441	101.0969	33.0172	81.5286	213.0759	2.4618	9.4558	1.3642	8.2291	10.4603	8
CGBPO	135.8845	92.0971	108.9536	28.2760	57.4066	199.9467	2.2528	6.9829	2.5561	4.6795	7.2349	2
EMEPO	150.0000	66.2049	121.5421	27.6788	52.7258	203.6263	2.3194	7.3533	1.8233	6.4946	8.8976	3

4.3. Tubular Column Design (TCD)

The objective of the Tubular Column Design problem is to obtain a uniform column with minimum cost. The schematic diagram is shown in Figure 9, where the column is required to withstand a compressive load of $P = 2500$ kgf. The mean diameter (d) of the column is restricted to lie between 2 and 14 cm, and columns with thicknesses (t) outside the range 0.2–0.8 cm are not available in the market. The material properties of the column are defined as follows: yield stress $\sigma_y = 500$ kgf/cm^2, modulus of elasticity $E = 0.85 \times 10^6$ kgf/cm^2, and density $\rho = 0.0025$ kgf/cm^3. The length of the column is $L = 250$ cm, and the design variable vector is defined as $\underset{X}{\rightarrow} = (x_1, x_2) = (d, t)$. The mathematical model can be expressed as follows:

$$\begin{aligned}
&\text{Minimize:} && f(\vec{X}) = 9.82dt + 2d,\\
&\text{Subject to:} && g_1(\vec{X}) = \frac{P}{\pi dt\sigma_y} - 1 \le 0,\\
& && g_2(\vec{X}) = \frac{8PL^2}{\pi^3 Edt(d^2+t^2)} - 1 \le 0,\\
& && g_3(\vec{X}) = \frac{2.0}{t} - 1 \le 0,\\
& && g_4(\vec{X}) = \frac{d}{14} - 1 \le 0,\\
& && g_5(\vec{X}) = \frac{0.2}{t} - 1 \le 0,\\
& && g_6(\vec{X}) = \frac{t}{0.8} - 1 \le 0,\\
&\text{Parameter range:} && 2 \le d \le 14,\ 0.2 \le t \le 0.8
\end{aligned}$$

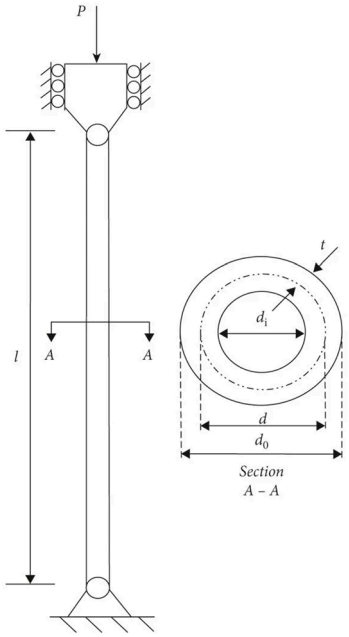


Figure 9. TCD diagram.

The optimization results for the TCD problem are presented in Table 15, where ECMPO ranks first among all compared algorithms.

Table 15. Comparison results of TCD problem.

Algorithm	Optimal Design Variables		Aver	Std	Best	Worst	Rank
	d	t					
ECMPO	5.4512	0.2920	26.5313	0.0000	26.5313	26.5313	1
GWO	5.4625	0.2916	26.9652	0.3123	26.5652	27.8069	10
VOA	5.4514	0.2928	27.0033	0.3239	26.5775	27.9265	11
HOA	5.4512	0.2920	26.6401	0.2283	26.5318	27.5530	6
WaOA	5.4475	0.2925	26.8025	0.2257	26.5453	27.3539	8
ASEO	5.4500	0.2921	26.5354	0.7743	27.4471	29.3286	12
BSLO	5.4512	0.2920	26.5360	0.0059	26.5314	26.5591	5
EPO	5.4512	0.2920	26.5326	0.0014	26.5313	26.5367	4
SRA	5.4528	0.2921	26.9523	0.5914	26.5459	29.1646	9
CEO	5.4480	0.2925	26.7937	0.3532	26.5429	28.0607	7
CGBPO	5.4512	0.2920	26.5326	0.0016	26.5313	26.5374	3
EMEPO	5.4512	0.2920	26.5313	0.0000	26.5313	26.5313	1

4.4. Cantilever Beam Design (CBD)

The objective of the Cantilever Beam Design (CBD) problem is to minimize the weight of the beam subject to a set of constraints. The schematic diagram of the CBD problem is shown in Figure 10. In this model, x_i (I = 1, 2, 3, 4, 5) denotes the side length (or cross-sectional dimension) of the *i*-th beam segment. Each variable represents the geometric size of an individual hollow square section along the beam. The mathematical model of the CBD problem can be defined as follows:

Consider: $\vec{x} = \left[x_1 \;\; x_2 \;\; x_3 \;\; x_4 \;\; x_5\right]$

Minimize: $f(\vec{x}) = 0.6224(x_1 + x_2 + x_3 + x_4 + x_5)$,

Subject to: $g(\vec{x}) = \frac{61}{x_1^3} + \frac{27}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \leqslant 0$,

Parameter range: $0.01 \leqslant x_1, x_2, x_3, x_4, x_5 \leqslant 100$.

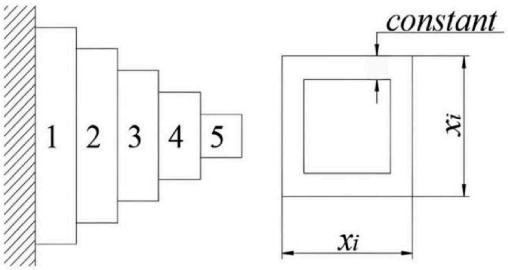


Figure 10. CBD diagram.

The comparison results for the CBD problem are presented in Table 16, where ECMPO outperforms all eleven competing algorithms.

Table 16. Comparison results of CBD problem.

Algorithm	Optimal Design Variables					Aver	Std	Best	Worst	Rank
	x_1	x_2	x_3	x_4	x_5					
ECMPO	5.9427	4.8799	4.4726	3.4837	2.1273	1.3500	0.0002	1.3500	1.3012	1
GWO	5.4583	5.0262	4.4281	3.9043	2.3021	1.3436	0.0021	1.3405	1.3495	5
VOA	5.9706	41.7708	2.9816	20.2295	22.8784	8.2376	1.6362	5.4115	11.5122	12
HOA	5.9982	3.7567	4.8968	5.2099	8.9111	1.5335	1.0346	1.3400	7.1049	9
WaOA	5.9367	7.2992	7.8937	5.7653	1.2070	1.9426	0.3488	1.4411	2.9371	10
ASEO	5.5998	7.6838	14.3510	6.3825	5.1991	2.5252	0.7384	1.3884	3.8386	11
BSLO	5.6290	10.8814	3.9290	3.4737	1.8499	1.3439	0.0017	1.3412	1.3578	6
EPO	5.8977	4.8835	4.5052	3.4751	2.1464	1.3408	0.0005	1.3402	1.3421	4
SRA	5.6747	5.6379	4.7617	3.6495	2.4151	1.4083	0.0300	1.3444	1.4697	8
CEO	5.8289	5.3495	4.4908	3.6945	2.2381	1.3604	0.0098	1.3453	1.3826	7
CGBPO	5.9907	5.3088	4.4970	3.4968	2.1528	1.3401	0.0002	1.3400	1.3404	2
EMEPO	5.7995	5.2959	4.5133	3.5030	2.1553	1.3402	0.0002	1.3400	1.3402	3

4.5. Gear Train Design (GTD)

The Gear Train Design (GTD) problem is a classical constrained integer optimization problem in mechanical engineering. Its objective is to minimize the gear ratio, defined as the ratio between the angular velocity of the output shaft and that of the input shaft. The design variables consist of the numbers of teeth on gears A (x_1), B (x_2), C (x_3), and D (x_4). The structure of the gear train is illustrated in Figure 11, and its mathematical model can be expressed as follows:

Consider: $\vec{x} = \left[x_1 \;\; x_2 \;\; x_3 \;\; x_4\right] = \left[n_A \;\; n_B \;\; n_C \;\; n_D\right]$,

Minimize: $f(\vec{x}) = \left(\frac{1}{6.931} - \frac{x_1 x_2}{x_3 x_4}\right)^2$,

Parameter range: $12 \leqslant x_1, x_2, x_3, x_4 \leqslant 60$. And all are integers.

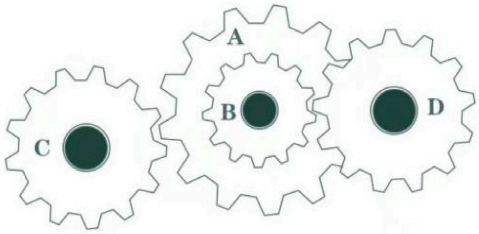


Figure 11. GTD diagram.

Specifically, continuous candidate solutions are transformed into integer-valued solutions using a rounding-based mapping strategy as follows:

$$x_i^{int} = round(x_i), i = 1, 2, 3, 4$$

where x_i^{int} denotes the integer-valued decision variable.

After rounding, a boundary control mechanism is applied to ensure feasibility within the predefined range [12, 60]. This guarantees that all generated solutions satisfy the problem constraints. The rounding-based discretization strategy is widely adopted in continuous metaheuristic algorithms for solving discrete and mixed-integer optimization problems, as it provides a simple and effective bridge between continuous search spaces and integer-constrained decision variables [48,49]. Importantly, this transformation is applied only as a post-processing step and does not alter the internal search mechanisms of any compared algorithms, ensuring a fair and consistent evaluation framework [50].

The experimental results are presented in Table 17, where ECMPO outperforms all competing algorithms.

Table 17. Comparison results of GTD problem.

Algorithm	Optimal Design Variables				Aver	Std	Best	Worst	Rank
	x_1	x_2	x_3	x_4					
ECMPO	19	16	49	43	4.1568×10^{-9}	6.6309×10^{-9}	2.7009×10^{-12}	2.7265×10^{-8}	1
GWO	19	16	43	49	1.2069×10^{-8}	3.3968×10^{-8}	2.7009×10^{-12}	1.7613×10^{-7}	4
VOA	16	19	43	49	2.6123×10^{-4}	8.1403×10^{-4}	2.7009×10^{-12}	3.1048×10^{-3}	10
HOA	16	19	49	43	4.3878×10^{-9}	7.2905×10^{-9}	2.7009×10^{-12}	2.7265×10^{-8}	2
WaOA	16	19	49	43	6.2821×10^{-9}	9.7390×10^{-9}	2.7009×10^{-12}	2.8578×10^{-8}	3
ASEO	13	31	57	49	1.6593×10^{-7}	2.9280×10^{-7}	9.9399×10^{-11}	1.1012×10^{-6}	8
BSLO	18	23	57	50	6.0594×10^{-3}	8.8266×10^{-3}	9.6793×10^{-7}	3.1081×10^{-2}	12
EPO	26	15	53	51	3.5356×10^{-8}	1.3838×10^{-7}	2.3078×10^{-11}	7.7786×10^{-7}	6
SRA	22	17	54	48	5.2559×10^{-5}	1.3580×10^{-4}	1.1661×10^{-10}	6.8882×10^{-4}	9
CEO	15	26	53	51	3.4593×10^{-4}	1.3161×10^{-3}	2.3078×10^{-11}	6.8726×10^{-3}	11
CGBPO	16	19	43	49	5.3005×10^{-8}	2.4706×10^{-7}	2.7009×10^{-12}	1.3811×10^{-6}	7
EMEPO	21	15	37	59	2.5015×10^{-8}	3.6460×10^{-8}	3.0676×10^{-10}	1.7491×10^{-7}	5

4.6. Reducer Design Problem (RDP)

In mechanical systems, the reducer is an essential component of gearboxes and is widely used in various applications. The schematic diagram is shown in Figure 12. In this optimization problem, the weight of the reducer is minimized subject to 11 constraints. This problem involves seven design variables, namely the face width $b(= x_1)$, module $m(= x_2)$, number of teeth of the pinion $Z(= x_3)$, length of the first shaft between bearings

$l_1(=x_4)$, length of the second shaft between bearings $l_2(=x_5)$, diameter of the first shaft $d_1(=x_6)$, and diameter of the second shaft $d_2(=x_7)$. The mathematical model of the RDP problem can be expressed as follows:

$$
\begin{aligned}
&\text{Minimize:} && f(x) = 0.7854x_1x_2^2(3.3333x_3^2 14.9334x_3 - 1.508x_1(x_6^2 + x_7^2) + \\
&&& 7.4777(x_6^3 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2),\\
&\text{Subject to:} && g_1(x) = \frac{27}{x_1x_2^2x_3} - 1 \leqslant 0,\\
&&& g_2(x) = \frac{397.5}{x_1x_2^2x_3^2} - 1 \leqslant 0,\\
&&& g_3(x) = \frac{1.93x_4^3}{x_2x_6^4x_3} - 1 \leqslant 0,\\
&&& g_4(x) = \frac{1.93x_5^3}{x_2x_7^4x_3} - 1 \leq 0,\\
&&& g_5(x) = \frac{\sqrt{(745x_4/(x_2x_3))^2 + 16.9\times10^6}}{110x_6^3} - 1 \leq 0,\\
&&& g_6(x) = \frac{\sqrt{(745x_4/(x_2x_3))^2 + 157.5\times10^6}}{85x_7^3} - 1 \leq 0,\\
&&& g_7(x) = \frac{x_2x_3}{40} - 1 \leq 0,\\
&&& g_8(x) = \frac{5x_2}{x_1} - 1 \leq 0,\\
&&& g_9(x) = \frac{x_1}{12x_2} - 1 \leq 0,\\
&&& g_{10}(x) = \frac{1.5x_6 + 1.9}{x_4} - 1 \leq 0,\\
&&& g_{11}(x) = \frac{1.1x_7 + 1.9}{x_5} - 1 \leq 0,\\
&&& 2.6 \leq x_1 \leq 3.6,\ 0.7 \leq x_2 \leq 0.8,\ 17 \leq x_3 \leq 28,\\
&\text{Parameter range:} && 7.3 \leq x_4 \leq 8.3,\ 7.8 \leq x_5 \leq 8.3,\ 2.9 \leq x_6 \leq 3.9,\\
&&& 5.0 \leq x_7 \leq 5.5
\end{aligned}
$$

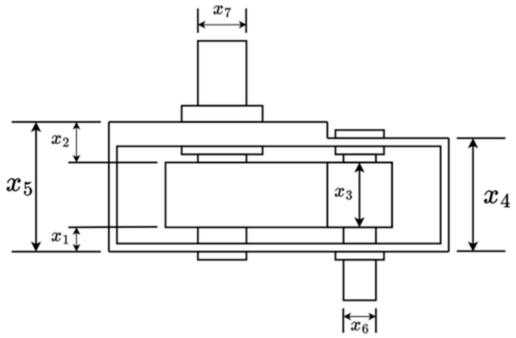


Figure 12. RDP diagram.

The optimization results of different algorithms for the RDP problem are presented in Table 18, from which it is evident that ECMPO outperforms all competing algorithms.

Based on the above experimental results, ECMPO consistently ranks first across all six selected engineering optimization problems, demonstrating its robust search capability and stable performance over diverse problem types. Furthermore, compared with other algorithms, ECMPO not only achieves high solution accuracy but also exhibits superior overall optimization performance, highlighting its effectiveness and strong potential in addressing complex engineering optimization problems.

Table 18. Comparison results of RDP problem.

Algorithm	Optimal Design Variables							Aver	Std	Best	Worst	Rank
	x_1	x_2	x_3	x_4	x_5	x_6	x_7					
ECMPO	3.4976	0.7000	17.0000	7.3000	7.8000	3.3501	5.2857	2995.8535	1.6753	2995.5424	3004.8754	1
GWO	3.5058	0.7000	17.0000	7.3000	7.8000	3.3505	5.2845	3517.3617	756.3543	2998.8868	5356.5735	8
VOA	3.4979	0.7000	17.0018	7.3001	7.8002	3.3500	5.2856	3056.3749	66.6517	2995.9004	3210.6366	6
HOA	3.4976	0.7000	17.0000	7.3000	7.8000	3.3501	5.2857	2999.0682	4.2863	2995.5424	3010.8936	2
WaOA	3.4960	0.7000	17.0217	7.3000	7.9282	3.3789	5.2846	3033.2241	13.6549	3010.0921	3060.9277	4
ASEO	3.5395	0.7000	17.0000	8.2949	8.1116	3.3519	5.3009	3109.4775	74.0879	3037.0912	3402.2915	7
BSLO	3.5856	0.7000	17.1809	7.9990	8.2592	3.5244	5.4271	4887.0337	1859.8685	3217.6354	12869.7511	10
EPO	3.4989	0.7000	17.0094	7.7127	8.1373	3.3510	5.2855	3561.7943	511.7961	3008.6839	4760.6211	9
SRA	3.4957	0.7000	17.0000	7.6691	7.8254	3.3718	5.2860	3035.4335	26.4377	3005.2673	3116.4731	5
CEO	3.5032	0.7015	19.4697	7.8182	7.8666	3.6459	5.3737	5035.3681	1615.6244	3607.2795	12056.3797	12
CGBPO	3.5894	0.7059	17.0000	7.3000	7.9847	3.3646	5.4310	4908.9087	2001.4637	3160.8434	9965.3460	11
EMEPO	3.4976	0.7000	17.0000	7.3089	7.8009	3.3501	5.2856	3004.4296	3.9550	2995.6534	3014.3896	3

5. Conclusions and Future Work

The PO algorithm, as a recently developed metaheuristic method, improves population diversity by simulating parrot behaviors; however, it is still prone to premature convergence and local optima stagnation. To address these issues, this paper proposes an enhanced variant, termed ECMPO, by incorporating an elite learning strategy, Cauchy mutation, and a multi-scale multi-directional search mechanism within a unified framework.

Specifically, the elite learning strategy is introduced prior to the parrot behavioral phases to accelerate convergence by guiding individuals toward high-quality regions. After the behavioral update, a conditional search mechanism is employed, in which elite individuals perform multi-scale multi-directional search to refine local exploitation, while non-elite individuals undergo Cauchy mutation to enhance global exploration. Through this hybrid design, ECMPO effectively achieves a dynamic balance between exploration and exploitation.

Extensive experiments are conducted on the CEC2017 benchmark suite to investigate the effects of the proposed strategies. Further validation is performed on the CEC2020, CEC2021, and CEC2022 test suites, as well as six classical engineering optimization problems, to comprehensively assess the performance of the proposed ECMPO algorithm. The experimental results demonstrate that ECMPO consistently outperforms the original PO algorithm and eleven state-of-the-art metaheuristic algorithms in terms of solution accuracy, convergence behavior, and robustness. In particular, ECMPO ranks first across all six engineering problems, highlighting its strong competitiveness in solving real-world complex optimization tasks.

Despite its promising performance, ECMPO still has certain limitations. First, as a population-based metaheuristic algorithm, its performance may be influenced by the initialization of the population, and different initial distributions may lead to variations in convergence behavior. Second, like most stochastic optimization methods, ECMPO may be affected by noise or uncertainty in objective function evaluations, which could potentially impact its stability in noisy environments. In addition, the integration of multiple enhancement strategies increases the algorithmic complexity, which may reduce computational efficiency in large-scale or real-time applications.

Future work will focus on addressing these limitations by developing more robust initialization strategies, improving noise-resistance capability, and simplifying the algorithmic structure to reduce computational overhead. Moreover, adaptive parameter control mech-

anisms and hybrid frameworks combining ECMPO with other intelligent optimization methods will be further investigated. The application of ECMPO to more complex tasks, such as high-dimensional optimization and hyperparameter tuning in machine learning models, will also be explored.

Author Contributions: Conceptualization, Y.Y. and M.Q.; methodology, M.L.; software, Y.Y. and M.L.; validation, Y.G., L.Z. and B.D.; formal analysis, M.L.; investigation, Y.Y.; resources, M.Q.; data curation, Y.Y.; writing—original draft preparation, Y.Y. and M.L.; writing—review and editing, M.Q. and Y.G.; visualization, L.Z.; project administration, Y.G.; funding acquisition, M.Q. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Program of China (14th Five-Year Plan), grant number 2021YFC3002100, and Impact of Moral Injury on the Mental Health of Emergency Rescue Personnel (Project No. ZY20260201).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

PO	Parrot Optimization
ECMPO	Elite Cauchy Multi-scale Multi-directional Parrot Optimization
GWO	Grey Wolf Optimizer
VOA	Vulture Optimization Algorithm
HOA	Hippopotamus Optimization Algorithm
WaOA	Walrus Optimization Algorithm
ASEO	Arctic Sea Eagle Optimization
BSLO	Blood Sucker Leech Optimization
EPO	Emperor Penguin Optimizer
CGBPO	Chaotic–Gaussian Barycenter Parrot Optimization
PVD	Pressure Vessel Design
GROP	Robot Gripper Optimization
TCD	Tubular Column Design
CBD	Cantilever Beam Design
GTD	Gear Train Design
RDP	Reducer Design Problem

References

1. Zhao, F.; Zhu, B.; Wang, L.; Xu, T.; Zhu, N.; Jonrinaldi, J. An offline learning co-evolutionary algorithm with problem-specific knowledge. *Swarm Evol. Comput.* **2022**, *75*, 101148. [CrossRef]
2. Panigrahy, D.; Samal, P. A novel hybrid eso-de-who algorithm for solving real-engineering optimization problems. *Int. J. Syst. Assur. Eng. Manag.* **2025**, *16*, 254–309. [CrossRef]
3. Zhan, Z.H.; Li, J.Y.; Kwong, S.; Zhang, J. Learning-aided evolution for optimization. *IEEE Trans. Evol. Comput.* **2022**, *27*, 1794–1808. [CrossRef]
4. Dehghani, M.; Hubalovsky, S.; Trojovsky, P. Northern Goshawk Optimization: A New Swarm-Based Algorithm for Solving Optimization Problems. *IEEE Access* **2021**, *9*, 162059–162080. [CrossRef]
5. Bencherqui, A.; Tahiri, M.A.; Karmouni, H.; Alfidi, M.; El Afou, Y.; Qjidaa, H.; Sayyouri, M. Chaos-Enhanced Archimede Algorithm for Global Optimization of Real-World Engineering Problems and Signal Feature Extraction. *Processes* **2024**, *12*, 406. [CrossRef]
6. Wang, S.; Li, Y.; Li, Y. Heterogeneous Genetic Learning and Comprehensive Learning Strategy Particle Swarm Optimizer. *Algorithms* **2025**, *18*, 755. [CrossRef]

7. Wei, L.; Zhong, H. Optimal Scheduling of Microgrids Based on a Two-Population Cooperative Search Mechanism. *Biomimetics* **2025**, *10*, 665. [CrossRef] [PubMed]
8. Lv, W.; He, Y.; Yang, Y.; Ma, X.; Chen, J.; Zhang, Y. Improving the Dung Beetle Optimizer with Multiple Strategies: An Application to Complex Engineering Problems. *Biomimetics* **2025**, *10*, 717. [CrossRef] [PubMed]
9. Khoshaba, F.S.; Kareem, S.W.; Hawezi, R.S. Swallow Search Algorithm (SWSO): A Swarm Intelligence Optimization Approach Inspired by Swallow Bird Behavior. *Computers* **2025**, *14*, 345. [CrossRef]
10. Li, G.; Zhang, T.; Tsai, C.-Y.; Yao, L.; Lu, Y.; Tang, J. Review of the Metaheuristic Algorithms in Applications: Visual Analysis Based on Bibliometrics. *Expert Syst. Appl.* **2024**, *255*, 124857. [CrossRef]
11. Dehghani, M.; Montazeri, Z.; Trojovská, E.; Trojovský, P. Coati Optimization Algorithm: A New Bio-Inspired Metaheuristic Algorithm for Solving Optimization Problems. *Knowl.-Based Syst.* **2023**, *259*, 110011. [CrossRef]
12. Zhao, W.; Wang, L.; Zhang, Z.; Mirjalili, S.; Khodadadi, N.; Ge, Q. Quadratic interpolation optimization (QIO): A new optimization algorithm based on generalized quadratic interpolation and its applications to real-world engineering problems. *Comput. Methods Appl. Mech. Eng.* **2023**, *417*, 116446. [CrossRef]
13. Eker, E.; Kayri, M.; Ekinci, S.; Izci, D. Comparison of Swarm-Based Metaheuristic and Gradient Descent-Based Algorithms in Artificial Neural Network Training. *Adv. Distrib. Comput. Artif. Intell. J.* **2023**, *12*, e29969. [CrossRef]
14. Shan, W.; Hu, H.; Cai, Z.; Chen, H.; Liu, H.; Wang, M.; Teng, Y. Multi-Strategies Boosted Mutative Crow Search Algorithm for Global Tasks: Cases of Continuous and Discrete Optimization. *J. Bionic Eng.* **2022**, *19*, 1830–1849. [CrossRef]
15. Lian, J.; Hui, G.; Ma, L.; Zhu, T.; Wu, X.; Heidari, A.A.; Chen, Y.; Chen, H. Parrot Optimizer: Algorithm and Applications to Medical Problems. *Comput. Biol. Med.* **2024**, *172*, 108064. [CrossRef]
16. Houssein, E.H.; Emam, M.M.; Alomoush, W.; Samee, N.A.; Jamjoom, M.M.; Zhong, R.; Dhal, K.G. An Efficient Improved Parrot Optimizer for Bladder Cancer Classification. *Comput. Biol. Med.* **2024**, *181*, 109080. [CrossRef] [PubMed]
17. Abdel-Salam, M.; Alomari, S.A.; Yang, J.; Lee, S.; Saleem, K.; Smerat, A.; Snasel, V.; Abualigah, L. Harnessing Dynamic Turbulent Dynamics in Parrot Optimization Algorithm for Complex High-Dimensional Engineering Problems. *Comput. Methods Appl. Mech. Eng.* **2025**, *440*, 117908. [CrossRef]
18. Huang, G.; Wei, J.; Yuan, Y.; Huang, H.; Chen, H. Parrot Optimization Algorithm for Improved Multi-Strategy Fusion for Feature Optimization of Data in Medical and Industrial Field. *Swarm Evol. Comput.* **2025**, *95*, 101908. [CrossRef]
19. Shaheen, M.A.M.; Hasanien, H.M.; Ullah, Z.; Alferidi, A.; Alsolami, M.; Lami, B. Hybrid Osprey-Parrot Optimization Algorithm for Enhanced Parameter Identification of Supercapacitor Model. *J. Energy Storage* **2025**, *137*, 118604. [CrossRef]
20. Li, F.; Dai, C.; Hussien, A.G.; Zheng, R. IPO: An Improved Parrot Optimizer for Global Optimization and Multilayer Perceptron Classification Problems. *Biomimetics* **2025**, *10*, 358. [CrossRef]
21. Sheykhi, E.; Yilmaz, M. Enhancing Electric Vehicle Battery Charging Efficiency Using an Improved Parrot Optimizer and Photovoltaic Systems. *Energies* **2025**, *18*, 3808. [CrossRef]
22. Yang, Y.; Fu, M.; Zhou, X.; Jia, C.; Wei, P. A Multi-Strategy Parrot Optimization Algorithm and Its Application. *Biomimetics* **2025**, *10*, 153. [CrossRef]
23. Wu, B.; Luo, J. A Novel Improved Binary Optimization Algorithm and Its Application in FS Problems. *Mathematics* **2025**, *13*, 675. [CrossRef]
24. Li, H.; Govindarajan, V.; Ang, T.F.; Shaikh, Z.A.; Ksibi, A.; Chen, Y.-L.; Ku, C.S.; Leong, M.C.; Shabaruddin, F.H.; Wan Ishak, W.Z.; et al. MSPO: A Machine Learning Hyperparameter Optimization Method for Enhanced Breast Cancer Image Classification. *Digit. Health* **2025**, *11*, 20552076251361603. [CrossRef]
25. Lei, W.; Huang, J.; Gu, Y. Multistrategy Chaotic Parrot Optimization Algorithm for Engineering Application Research. *J. Mech. Sci. Technol.* **2025**, *39*, 6625–6639. [CrossRef]
26. Huang, K.; Li, W.; Fang, H.; Wu, X.; Wang, L.; Peng, H. IPORF: A Combined Improved Parrot Optimizer Algorithm and Random Forest for Fault Diagnosis in AUV. *Ocean Eng.* **2024**, *313*, 119665. [CrossRef]
27. Soundradevi, G.; Jawahar, R.; Dani Abraham, J.; Radhika, A.; Lawrence Sahaya Sundar, A. Hybrid Parrot Optimizer and Dynamic Weighted Hypergraph Convolutional Network for Fast Electric Vehicle Charging Stations with Renewable Energy. *Iran. J. Sci. Technol. Trans. Electr. Eng.* **2025**, *50*, 497–513. [CrossRef]
28. Saad, M.R.; Emam, M.M.; Houssein, E.H. An Efficient Multi-Objective Parrot Optimizer for Global and Engineering Optimization Problems. *Sci. Rep.* **2025**, *15*, 5126. [CrossRef] [PubMed]
29. Yan, Z.; Cai, S.; Hou, S.; Yan, J. Adaptive Quantum Multi-Objective Parrot Optimizer for Task Allocation of Multi-UUV Systems in Underwater Search. *Ocean Eng.* **2024**, *314*, 119760. [CrossRef]
30. Zhang, Y.H.; Wang, J.S.; Xing, Y.X.; Sun, Y.F.; Zhang, S.W.; Bai, X.L. Multi-Objective Parrot Optimizer with Improved Lévy Flight and Adaptive Elliptical Segmentation—Based Screening Mechanism for Layout Optimization of Wireless Sensor Networks. *J. Netw. Comput. Appl.* **2026**, *247*, 104413. [CrossRef]
31. Chakraborty, P.; Sharma, S.; Saha, A.K. Convergence Analysis of Butterfly Optimization Algorithm. *Soft Comput.* **2023**, *27*, 7245–7257. [CrossRef]

32. Chen, C.; Cao, L.; Chen, Y.; Chen, B.; Yue, Y. A Comprehensive Survey of Convergence Analysis of Beetle Antennae Search Algorithm and Its Applications. *Artif. Intell. Rev.* **2024**, *57*, 141. [CrossRef]
33. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
34. Abdollahzadeh, B.; Gharehchopogh, F.S.; Mirjalili, S. African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems. *Comput. Ind. Eng.* **2021**, *158*, 107408. [CrossRef]
35. Amiri, M.H.; Mehrabi Hashjin, N.; Montazeri, M.; Mirjalili, S.; Khodadadi, N. Hippopotamus optimization algorithm: A novel nature-inspired optimization algorithm. *Sci. Rep.* **2024**, *14*, 5032. [CrossRef]
36. Trojovský, P.; Dehghani, M. A new bio-inspired metaheuristic algorithm for solving optimization problems based on walruses behavior. *Sci. Rep.* **2023**, *13*, 8775. [CrossRef]
37. Wang, W.; Tian, W.; Xu, D.; Zang, H. Arctic Puffin Optimization: A Bio-Inspired Metaheuristic Algorithm for Solving Engineering Design Optimization. *Adv. Eng. Softw.* **2024**, *195*, 103694. [CrossRef]
38. Bai, J.; Nguyen-Xuan, H.; Atroshchenko, E.; Kosec, G.; Wang, L.; Wahab, M.A. Blood-sucking leech optimizer. *Adv. Eng. Softw.* **2024**, *195*, 103696. [CrossRef]
39. Dhiman, G.; Kumar, V. Emperor penguin optimizer: A bio-inspired algorithm for engineering problems. *Knowl.-Based Syst.* **2018**, *159*, 20–50. [CrossRef]
40. Hussein, N.K.; Qaraad, M.; El Najjar, A.M.; Farag, M.A.; Elhosseini, M.A.; Mirjalili, S.; Guinovart, D. Schrödinger Optimizer: A Quantum Duality-Driven Metaheuristic for Stochastic Optimization and Engineering Challenges. *Knowl.-Based Syst.* **2025**, *328*, 114273. [CrossRef]
41. Dong, Y.; Zhang, S.; Zhang, H.; Zhou, X.; Jiang, J. Chaotic Evolution Optimization: A Novel Metaheuristic Algorithm Inspired by Chaotic Dynamics. *Chaos Solitons Fractals* **2025**, *192*, 116049. [CrossRef]
42. Kannan, B.K.; Kramer, S.N. An Augmented Lagrange Multiplier Based Method for Mixed Integer Discrete Continuous Optimization and Its Applications to Mechanical Design. *J. Mech. Des.* **1994**, *116*, 405–411. [CrossRef]
43. Datta, R.; Pradhan, S.; Bhattacharya, B. Analysis and Design Optimization of a Robotic Gripper Using Multiobjective Genetic Algorithm. *IEEE Trans. Syst. Man Cybern Syst.* **2016**, *46*, 16–26. [CrossRef]
44. Rao, S.S. *Engineering Optimization: Theory and Practice*; John Wiley & Sons: Hoboken, NJ, USA, 2019.
45. Chickermane, H.; Gea, H.C. Structural optimization using a new local approximation method. *Int. J. Numer. Meth. Engng.* **1996**, *39*, 829–846. [CrossRef]
46. Abderazek, H.; Sait, S.M.; Yildiz, A.R. Optimal Design of Planetary Gear Train for Automotive Transmissions Using Advanced Meta-Heuristics. *Int. J. Veh. Des.* **2019**, *80*, 121–136. [CrossRef]
47. Guan, Z.; Ren, C.; Niu, J.; Wang, P.; Shang, Y. Great Wall Construction Algorithm: A Novel Meta-Heuristic Algorithm for Engineering Problems. *Expert Syst. Appl.* **2023**, *233*, 120905. [CrossRef]
48. Neumann, C.; Stein, O. Feasible Rounding Approaches for Equality Constrained Mixed-Integer Optimization Problems. *Optimization* **2023**, *72*, 581–606. [CrossRef]
49. Hübner, R.; Schöbel, A. When Is Rounding Allowed in Integer Nonlinear Optimization? *Eur. J. Oper. Res.* **2014**, *237*, 404–410. [CrossRef]
50. Burer, S.; Letchford, A.N. Non-convex mixed-integer nonlinear programming: A survey. *Surv. Oper. Res. Manag. Sci.* **2012**, *17*, 97–106. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.