algorithms

MDPI

Article

A Detailed Interpretation of an Improved Artificial Lemming Algorithm and Its Application in Engineering Problems

Xuemei Zhu [1], Yaqi Yang [2] and Jianjun Dong [2,*]

[1] Experimental and Practical Training Teaching Management Department, West Anhui University, Lu'an 237012, China
[2] School of Electronic Information and Artificial Intelligence, West Anhui University, Lu'an 237012, China
* Correspondence: 03260168@wxc.edu.cn

Abstract

This paper proposes an improved artificial lemming algorithm (IMALA), a metaheuristic optimization technique based on biological principles, and explores its application in solving engineering optimization problems. To overcome the limitations of the ALA, IMALA introduces three key improvements: population initialization based on cubic chaos mapping to improve diversity, a dynamic differential mutation strategy to achieve adaptive exploration and exploitation balance, and a stochastic adversarial learning mechanism using beta distribution to accelerate convergence and improve robustness. Under the reported iteration-based setting, IMALA achieved favorable average ranks of 1.69 and 1.58 in the nine-algorithm comparisons on CEC2017 and CEC2022, respectively, although it did not obtain the best result on every benchmark function. For practical verification, IMALA is applied to three engineering problems: a step-cone pulley with minimization weight, power loss reduction in a hydrostatic thrust bearing, and piston lever design with minimization oil volume optimization. Across the three engineering optimization problems, IMALA demonstrates competitive solution quality and relatively stable performance under the adopted experimental settings. Under the same iteration limit, the fast convergence and stability of this algorithm make it a reliable tool for solving complex, high-dimensional and multimodal engineering problems, and it has significant improvements over existing methods both in theory and practical applications.

Keywords: improved artificial lemming algorithm; metaheuristic optimization; chaotic mapping; opposition-based learning; engineering design optimization

Check for updates

Academic Editor: Thomas Hanne

Received: 13 October 2026
Revised: 8 November 2026
Accepted: 9 November 2026
Published: 10 November 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

The objective of optimization problems is to find the optimal solution that minimizes or maximizes the objective function while satisfying the constraints [1]. With the rapid development of artificial intelligence technology, complex optimization problems continue to emerge in various research fields [2], spanning from engineering design and medical image analysis to feature selection and high-dimensional numerical benchmarks. Traditional optimization methods rely on gradient information, making it difficult for them to handle high-dimensional, nonlinear, and non-convex complex optimization problems [3,4]. In contrast, metaheuristic algorithms have gradually become ideal tools for solving complex optimization problems due to their advantages of not requiring gradient information, simple structure, and strong adaptability [5,6].

As an effective global optimization tool, metaheuristic algorithms have attracted widespread attention in recent years [7]. Modern intelligent optimization algorithms have

revolutionized problem-solving across engineering, finance, and scientific domains [8]. This is evidenced by a surge of novel and improved algorithms. For instance, the Multi-strategies Improved Coati Optimization Algorithm (MSICOA) [9] integrates nonlinear inertia weight and mechanisms from other algorithms to overcome premature convergence. Similarly, the Adaptive Dynamic Crayfish Algorithm (AD-COA-L) [10] employs chaotic initialization and a local escape operator to enhance its performance on global high-dimensional problems. Beyond nature-inspired paradigms, physics-based and human-inspired concepts are also explored. The lightweight differential evolution variant Cε-LDE [11] combines Lévy flight with an advanced constraint-handling method to effectively solve constrained optimization problems. In data science, algorithms like the Halton-based enhanced adaptive dung beetle algorithm (AQDBO) [12] demonstrate excellent performance in feature selection, improving classification models by efficiently selecting the most relevant features from high-dimensional datasets. Engineering design also benefits greatly, as shown by the Improved Golden Jackal Optimization (LGJO) [13], which successfully solves constrained industrial applications.

Among the most influential bio-inspired approaches, the Whale Optimization Algorithm (WOA) [14] stands out for its bubble-net hunting mechanism, achieving an exceptional balance between exploration and exploitation [15]. The Dragonfly Algorithm (DA) [16] mimics swarm dynamic behaviors, demonstrating superior performance in multi-objective problems. Equally noteworthy is the Harris Hawks Optimization (HHO) [17], which simulates cooperative predation strategies with adaptive nonlinear energy regulation. Physics-inspired methods like the Multi-Verse Optimizer (MVO) [18] leverage cosmological principles, while the Equilibrium Optimizer (EO) [19] innovatively models mass balance dynamics. The Slime Mould Algorithm (SMA) [20] has gained prominence for its unique oscillation-based search mechanism, particularly in medical image processing [21]. Despite their widespread success, existing metaheuristic algorithms, as seen in the aforementioned studies, often exhibit notable performance deficiencies including early convergence tendencies, exploration–exploitation imbalances, and deteriorating optimization reliability in high-dimensional, multimodal search spaces [22,23].

Recent advances in metaheuristics highlight the success of hybrid and multi-strategy approaches in overcoming algorithmic limitations. For example, Xiao et al. [24] developed IHAOAVOA, combining Aquila Optimizer's exploration with African Vultures' exploitation, enhanced by COBL and FDB strategies, to achieve superior performance in global optimization and engineering design. Xiao et al. [25] proposed MSAO, a Snow Ablation Optimizer enhanced with good point set initialization and dynamic lens opposition-based learning, which significantly improved convergence accuracy and scalability on CEC benchmarks. Fuzzy rule-based systems enhance optimization algorithms by dynamically adjusting parameters and handling uncertainties through linguistic variables, improving adaptability and performance in complex problems like energy management and robotic path planning [26]. Similarly, this work on the artificial lemming algorithm (ALA) [27] adopts innovative strategies to enhance its search capabilities and applicability in complex engineering problems.

In this context, the relatively new Artificial Lemming Algorithm (ALA) [27] has shown promising performance. Its design, inspired by lemming behaviors such as long-distance migration and burrowing, offers a simple structure and an inherent balance between exploration and exploitation. The algorithm's potential has been quickly recognized, leading to several recent improvements targeting its specific weaknesses. For instance, Xie et al. [28] proposed a multi-strategy improved ALA (MsIALA) incorporating cubic chaotic mapping and double adaptive t-distribution perturbation to solve complex 3D UAV route planning problems. Similarly, Tan et al. [29] developed an enhanced ALA using a collaborative Bern-

stein polynomial and chaotic mapping initialization strategy to address global optimization and cloud task-scheduling problems more effectively.

This work selects the Artificial Lemming Algorithm (ALA) as the foundation for improvement, building upon the momentum of these recent studies. While the aforementioned improvements have demonstrated success in specific applications like UAV path planning and cloud scheduling, opportunities remain to further enhance ALA's core robustness, convergence speed, and applicability to a wider range of standard and engineering benchmarks. The proposed Improved Artificial Lemming Algorithm (IMALA) in this paper introduces a distinct set of strategies—chaotic initialization, a differential mutation strategy, and Beta distribution opposition-based learning—to address these challenges from a different perspective, aiming to contribute a more versatile and powerful variant to the growing ALA family.

Although ALA has demonstrated promising optimization performance, several potential limitations remain. First, conventional random initialization may lead to insufficient population diversity and uneven coverage of the search space. Second, the search process may experience stagnation when individuals concentrate around promising solutions, limiting further improvement. Third, population diversity may decrease during the iterative search process, reducing the ability to explore alternative regions. To address potential limitations of the ALA, we propose an improved ALA (IMALA) that integrates three established strategies at different stages of the search process. The contribution of this study lies in combining these strategies within the ALA framework and quantitatively evaluating whether their integration improves performance over the original ALA, rather than in proposing the individual strategies themselves.

- Chaotic initialization using cubic map: Population initialization based on the cubic chaotic map is designed to improve initial population diversity, reduce the risk of early clustering, and provide broader coverage of the search space [30].
- Differential current-to-best mutation strategy: Trial solutions are generated by combining the current solution, the current best solution, and a difference vector between population members, with the aim of reducing stagnation during local exploitation [31].
- Beta distribution opposition-based learning: Alternative candidate solutions are generated within bounds determined by the current population to mitigate the loss of diversity during the search and enhance exploration [32].

To systematically investigate the effectiveness and applicability of the proposed IMALA, this study focuses on three key research questions: how the proposed improvement strategies individually and collectively influence the optimization performance of the original ALA; how IMALA compares with other representative optimization algorithms in terms of solution accuracy, convergence behavior, and stability on the CEC2017 and CEC2022 benchmark functions; and to what extent IMALA can effectively solve constrained engineering optimization problems, including step-cone pulley design, hydrostatic thrust bearing design, and piston lever design.

The remainder of this paper is organized as follows. First, the fundamental principles and mathematical formulation of the original ALA are introduced. Next, the proposed IMALA and its three improvement strategies are described, together with the pseudocode, flowchart, and ablation analysis. Subsequently, the experimental results on the CEC2017 and CEC2022 benchmark functions are presented and discussed. The proposed algorithm is then applied to three engineering optimization problems, including step-cone pulley design, hydrostatic thrust bearing design, and piston lever design. Finally, the conclusions, limitations, and directions for future research are presented.

2. Artificial Lemming Algorithm

The artificial lemming algorithm (ALA) [27] is a bio-inspired metaheuristic optimization technique designed to solve complex optimization problems. It is based on the behaviors of lemmings, particularly their long-distance migration, burrow digging, foraging for food, and evading predators [33]. These behaviors are mathematically modeled to guide the search for optimal solutions. The migration and burrowing behaviors focus on exploration, allowing the algorithm to explore large search spaces, while foraging and predator evasion emphasize exploitation, refining solutions in promising areas. ALA also incorporates an energy-decreasing mechanism to dynamically balance exploration and exploitation during the optimization process [34]. This adaptive approach, combined with techniques like Brownian motion and Lévy flight, helps the algorithm avoid local optima and effectively converge towards global solutions. The working process and mathematical model of the algorithm are as follows.

2.1. Initialization

The ALA is a population-based optimization method [33]. By initializing the positions of all search agents, generating an N (population size) rows and Dim (dimensional) columns matrix within the upper and lower limits of the problem, and taking the optimal position of each round as the current approximate solution in the iteration. The population is initialized using the following formula [35]:

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

where X_i represents the position vector of the i-th individual ($i = 1, 2,..., \mathrm{N}$), lb is the lower bound and ub is the upper bound vectors of the search space; $rand$ $(0,\ 1)$ generates any numbers between 0 and 1.

2.2. Core Search Strategies

The intensity parameter for the Brownian motion model is as follows:

$$E = 2 \cdot \ln\left(\frac{1}{r_1}\right) \cdot \theta \tag{2}$$

$$\theta = 2 \cdot \arctan\left(1 - \frac{t}{T}\right) \tag{3}$$

where $r_1 \sim rand\ (0, 1)$ is a uniformly distributed random number, θ is time-varying parameter, t is the current iteration number, and T is the maximum number of iterations.

2.3. Position Update Formulas

(1) Exploration Phase (E > 1):
When $\xi < 0.3$,

$$X_{i,j}^{new} = P_j + F \cdot RB_{i,j} \cdot \left[r_3 \cdot \left(P_j - X_{i,j}\right) + (1 - r_3) \cdot \left(X_{i,j} - X_{k,j}\right)\right] \tag{4}$$

where:

- $X_{i,j}^{new}$ is the new position of the i-th individual in the j-th dimension.
- P_j is the value of the current best solution in the j-th dimension.
- F is a scaling factor.
- $RB_{i,j}$ is a random number based on Brownian motion for the i-th individual in the j-th dimension.
- $r_3 \sim$ rand $(-1, 1)$ is a random number.
- $X_{i,j}$ is the current position of the i-th individual in the j-th dimension.

- $X_{k,j}$ is the value of a randomly selected k-th individual ($k \neq i$) in the j-th dimension.

When $\xi \geq 0.3$,

$$X_{i,j}^{new} = X_{i,j} + F \cdot r_4 \cdot (1 + \sin(0.5 \cdot t)) \cdot \left(P_j - X_{k,j}\right) \tag{5}$$

where $r_4 \sim rand\ (0, 1)$ is a random number.

(2) Exploitation phase ($E \leq 1$):

When $r_5 < 0.5$ (Spiral Search),

$$X_{i,j}^{new} = P_j + F \cdot X_{i,j} \cdot spiral \cdot r_6 \tag{6}$$

where spiral is a spiral path control value. $r_6 \sim rand\ (0, 1)$ is a random number.

$$spiral = radius \cdot (\sin(2\pi r_7) + \cos(2\pi r_7)) \tag{7}$$

where radius is the radius of the spiral. $r_7 \sim rand\ (0, 1)$ is a random number.

$$radius = \sqrt{\sum_{j=1}^{\dim} (P_j - X_{i,j})^2} \tag{8}$$

where $r_5 \geq 0.5$ (Lévy Flight).

$$X_{i,j}^{new} = P_j + F \cdot G \cdot Lévy_j \cdot \left(P_j - X_{i,j}\right) \tag{9}$$

where G is a guidance factor. $Lévy_j$ is the *Lvy* flight step in the j-th dimension.

$$G = 2 \cdot sign(r_8 - 0.5) \cdot \left(1 - \frac{t}{T}\right) \tag{10}$$

where $r_8 \sim rand\ (0, 1)$ is random number.

The ALA simulates four unique lemming behaviors and integrates Brownian motion and Lévy flight for global exploration and local exploitation, allowing the algorithm to avoid local optima and enhance convergence accuracy. In addition, ALA has an energy reduction mechanism that dynamically adjusts the balance between exploration and exploitation as iterations proceed. Meanwhile, the ALA faces challenges such as premature convergence, high computational complexity, and difficulty solving high-dimensional problems. These problems will be solved by introducing improved strategies, namely the improved artificial lemming algorithm (IMALA).

3. IMALA

In response to the limitations of ALA, three strategies were proposed. First, a population initialization method based on cubic chaotic mapping; second, a dynamic mutation strategy was designed to effectively balance the global and local search capabilities of the algorithm; and third, an opposition-based learning mechanism was introduced, which enhanced the algorithm's global exploration ability, accelerated its optimization speed, and improved its robustness and adaptability.

3.1. Cubic Chaos Mapping-Based Population Initialization

To improve population diversity in the proposed algorithm, chaotic mapping is employed for the initialization of the population [36]. The cubic chaotic map, as a discrete-time dynamical system based on cubic polynomial nonlinearity, demonstrates superior performance in generating initial populations compared to conventional random methods. As an advanced member of the chaotic map family, it produces pseudorandom sequences through iterative nonlinear transformations that exhibit three fundamental chaotic characteristics: ergodicity, aperiodicity, and extreme sensitivity to initial conditions [37].

The cubic map is defined by the recurrence relation

$$x_{n+1} = a \cdot x_n\left(1 - x_n^2\right) \tag{11}$$

where:

- x_n is the value at iteration n (with $x_1 = 0.3$ in the implementation), the values x_n remain bounded in $[-1, 1]$ for this parameter setting.
- a is the control parameter (set to 2.595). For $a = 2.595$, the system exhibits chaotic dynamics with sensitive dependence on initial conditions. The cubic term x_n^3 distinguishes it from the logistic map.
- The term $(1 - x_n^2)$ introduces nonlinearity.

The cubic map's third-order polynomial form provides significant advantages over simpler chaotic maps (e.g., logistic or tent maps) in terms of chaotic behavior complexity. Its mathematical formulation enables:

- More sophisticated nonlinear dynamics that prevent premature clustering of initial solutions;
- Enhanced uniform distribution properties across the entire solution space;
- Improved trajectory coverage that facilitates better exploration of potential optimal regions.

To generate the initial population within the search space, the cubic chaotic sequence in $[-1, 1]$ is linearly mapped to the corresponding variable bounds $[lb, ub]$. Specifically, each component of the chaotic sequence is transformed using the following equation:

$$X_{i,j} = lb_j + \frac{z_{i,j} + 1}{2}\left(ub_j - lb_j\right) \tag{12}$$

where $z_{i,j}$ represents the chaotic sequence value associated with the j-th dimension of the i-th individual, and lb_j and ub_j denote the corresponding lower and upper bounds, respectively.

These characteristics make the cubic map particularly effective for population initialization in metaheuristic algorithms. When mapped to the solution space, the generated sequences create initial populations with:

- Broader spatial distribution;
- Reduced correlation between individuals;
- Superior coverage of boundary regions.

The resulting diversity enhancement directly contributes to the algorithm's global search capability, as demonstrated in our comparative studies with standard initialization methods. Furthermore, the cubic map's computational efficiency (requiring only basic arithmetic operations per iteration) ensures this initialization approach adds negligible computational overhead while providing substantial optimization performance improvements.

3.2. Differential Current-to-Best Mutation Strategy

In order to balance the development capabilities of the ALA, this paper designs a dynamic hybrid mutation strategy. This mutation strategy accelerates convergence while maintaining population diversity through elite-guided differential mutation (Equation (13)).

$$X_i(t+1) = X_i(t) + F \times (X_{best} - X_i(t)) + F \times (X_{r1}(t) - X_{r2}(t)) \tag{13}$$

where $X_i(t)$ is the current individual, X_{best} is the current best solution, $X_{r1}(t)$, $X_{r2}(t)$ are two distinct randomly selected individuals ($r_1 \neq r_2$), and $F \sim rand\ (0, 1)$ is the random scaling factor.

The mutation strategy employs a randomly generated scaling factor F, sampled from a uniform distribution $U(0, 1)$. By combining the current best solution with differential information from randomly selected individuals, the strategy introduces stochastic variation

into the search process, which may help maintain population diversity and improve optimization performance. During initial optimization phases, heightened mutation amplitudes facilitate comprehensive global exploration, while progressively diminishing variation rates during later stages enable precise local refinement. This approach yields three principal advantages: (1) enhanced escape capability from local optima during early exploration, (2) smooth transition between global and local search modes, and (3) improved convergence precision through gradual search space focusing.

3.3. Stochastic Opposition-Based Learning Strategy

The proposed strategy employs a stochastic opposition-based learning mechanism using Beta distributions, which dynamically adapts to the population's diversity [38]. The process can be mathematically decomposed into several key steps.

3.3.1. Dynamic Boundary Calculation

First, the strategy establishes dynamic boundaries based on the current population distribution:

$$\boldsymbol{d}_{max} = \max(\boldsymbol{X}, [\,], 1), \quad \boldsymbol{d}_{min} = \min(\boldsymbol{X}, [\,], 1) \tag{14}$$

where X is the population matrix of size $N \times \dim$.

Population diversity measurement:

The strategy uses normalized pairwise distances to quantify population diversity:

$$d_{i,j} = \frac{\|\mathbf{X}^{(i)} - \mathbf{X}^{(j)}\|_{\mathbf{D}^{-1}}}{\sqrt{\dim}}, \quad \boldsymbol{D} = diag(\boldsymbol{d}_{max} - \boldsymbol{d}_{min}) \tag{15}$$

where $\|\cdot\|_{\mathbf{D}^{-1}}$ denotes the weighted Euclidean norm. The characteristic diversity **CD** for each individual is

$$CD_i = \min_{j \neq i} d_{i,j} \tag{16}$$

The population's normalized diversity metric is then

$$\overline{CD} = \frac{1}{N} \sum_{i=1}^{N} CD_i \tag{17}$$

3.3.2. Beta Distribution Parameterization

The strategy employs two distinct modes of operation based on a random selection.

- Concave Mode (Exploration Emphasis): When $rand\ (0, 1) < 0.5$, use

$$spread = \left(\frac{1}{\sqrt{\overline{CD}}} \right)^{1+\mathcal{N}(0,0.5)} \tag{18}$$

For each individual $\mathbf{X}^{(i)}$, calculate the mode position:

$$\mathbf{m} = \frac{\left(\mathbf{d}_{\max} + \mathbf{d}_{\min} - \mathbf{X}^{(i)}\right) - \mathbf{d}_{\min}}{\mathbf{d}_{\max} - \mathbf{d}_{\min}} \tag{19}$$

The Beta distribution parameters are then determined by

$$(\alpha, \beta) = \begin{cases} (spread \cdot \frac{(spread-2)m_k+1}{spread(1-m_k)}, spread) & if\ m_k < 0.5 \\ (spread, spread \cdot \frac{2-spread}{spread} + \frac{spread-1}{spreadm_k}) & otherwise \end{cases} \tag{20}$$

In Equation (20), m_k denotes the k-th component of the normalized position vector m defined in Equations (19) and (22), where $k = 1, 2, \ldots, \dim$. The parameter spread is the distribution spread parameter calculated using Equation (18) or Equation (21), depending on the selected search mode. The Beta distribution parameters α and β are calculated for each dimension.

- Convex Mode (Exploitation Emphasis): Otherwise, we use

$$spread = 0.1\sqrt{\overline{CD}} + 0.9 \tag{21}$$

with the mode position calculated as

$$\mathbf{m} = \frac{\mathbf{X}^{(i)} - \mathbf{d}_{\min}}{\mathbf{d}_{\max} - \mathbf{d}_{\min}} \tag{22}$$

The Beta parameters are computed identically to the concave case.

3.3.3. Opposition Point Generation

New candidate solutions are generated by

$$\mathbf{X}_{opp}^{(i)} = (\mathbf{d}_{\max} - \mathbf{d}_{\min}) \cdot Beta(\alpha, \beta) + \mathbf{d}_{\min} \tag{23}$$

where Beta (α, β) denotes a vector of independent Beta-distributed random variables.

The algorithm converts deterministic opposition learning into controlled stochasticity using a Beta distribution, resulting in adaptive search ranges through dynamic boundaries, multimodal optimization capability via a concave strategy, fast convergence through a convex strategy, and enhanced robustness through boundary control and greedy selection.

The flowchart (Figure 1) and pseudo-code (Algorithm 1) of IMALA illustrate its structured optimization process: The algorithm begins by initializing the population using cubic chaotic mapping (Equation (11)) to ensure diversity, followed by fitness evaluation. Each iteration dynamically adjusts the search behavior through an energy coefficient (E) (Equation (2)), which triggers either exploration (Equations (4) and (5)), simulating lemming migration/digging) or exploitation (Equations (6)–(9), mimicking foraging/predator evasion). Subsequently, a differential current-to-best mutation (Equation (13)) refines solutions, while stochastic opposition-based learning (Equation (23)) with Beta distributions escapes local optima. The loop continues until reaching maximum iterations (T_{max}), returning the best solution. This hybrid approach balances global exploration and local exploitation through bio-inspired mechanisms and adaptive stochastic strategies.

Computational Complexity Analysis: Let N, D, T, and C_f denote the population size, dimension, maximum number of iterations, and cost of one objective function evaluation, respectively. The original ALA initializes N individuals in $O(ND)$ operations and performs N fitness evaluations at initialization. Each subsequent iteration requires $O(ND)$ operations for position updating and N objective function evaluations, giving a total time complexity of $O(ND + NC_f + T(ND + NC_f))$. IMALA retains the $O(ND)$ initialization and position-update costs; cubic chaotic initialization and current-to-best differential mutation do not increase their asymptotic order. The Beta-distribution-based opposition learning module, however, calculates pairwise population distances, introducing $O(N^2D)$ operations per iteration, and requires two additional objective function evaluations per individual. Thus, the current implementation evaluates $3N$ candidates per iteration, with overall time complexity $O(ND + NC_f + T(N^2D + ND + 3NC_f))$ and $O(ND)$ auxiliary space at the population level. The extra distance calculations and fitness evaluations increase IMALA's computational cost relative to ALA.

Algorithm 1 IMALA

Require: pop: Population size.
dim: Number of dimensions.
ub, *lb*: Upper bound and lower bound for dim.
T: Maximum number of iterations.
fobj: Objective function.
Ensure: *Xb*: The best solution [39].
Gb: The best fitness value.

Initialization phase:
Generate chaotic sequence using Cubic map, scale to search space bounds and evaluate initial population by Equation (11)
for t = 1 to *T*max **do**
    Calculate energy factor by Equation (2)
    for each individual **do**
        if $E > 1$ **then**
            When $\xi \geq 0.3$: Long-distance migration by Equation (4)
            When $\xi \geq 0.3$: Digging holes by Equation (5)
        else
            When $r_5 < 0.5$: Foraging by Equation (6)
            When $r_5 \geq 0.5$: Predator evasion by Equation (9)
        end if
        Apply differential current-to-best mutation strategy to update individual positions by Equation (13)
    end for
    Apply boundary constraints and evaluate new solutions
    Apply stochastic opposition-based learning by Equation (23)
    Evaluate and update population
    Update *Xb*
end for
return *Xb* and *Gb*.

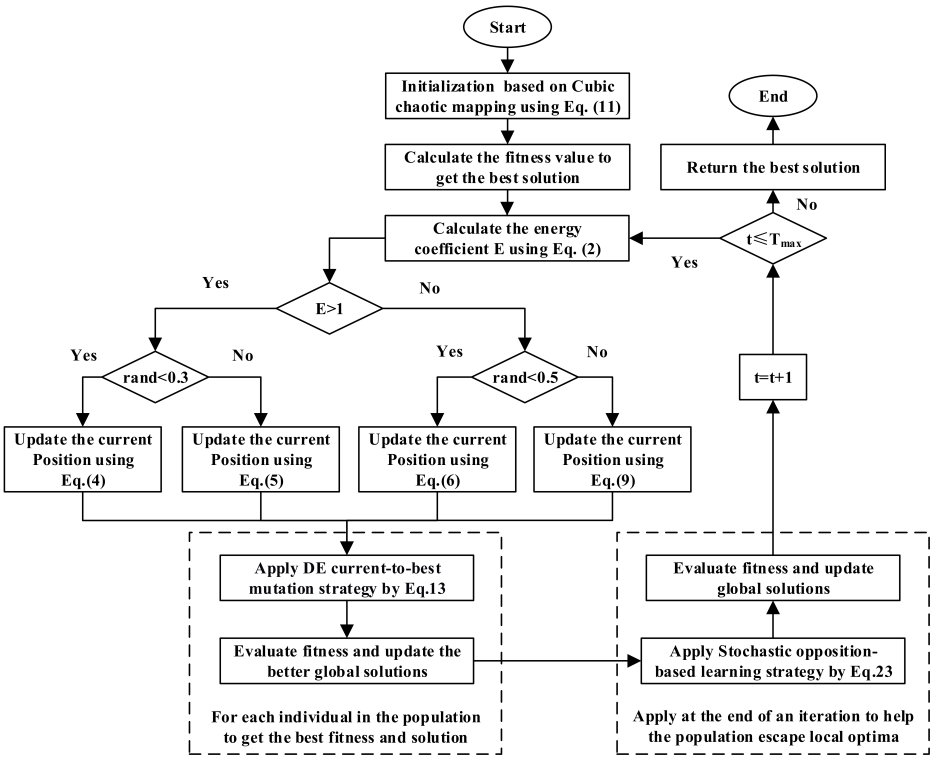
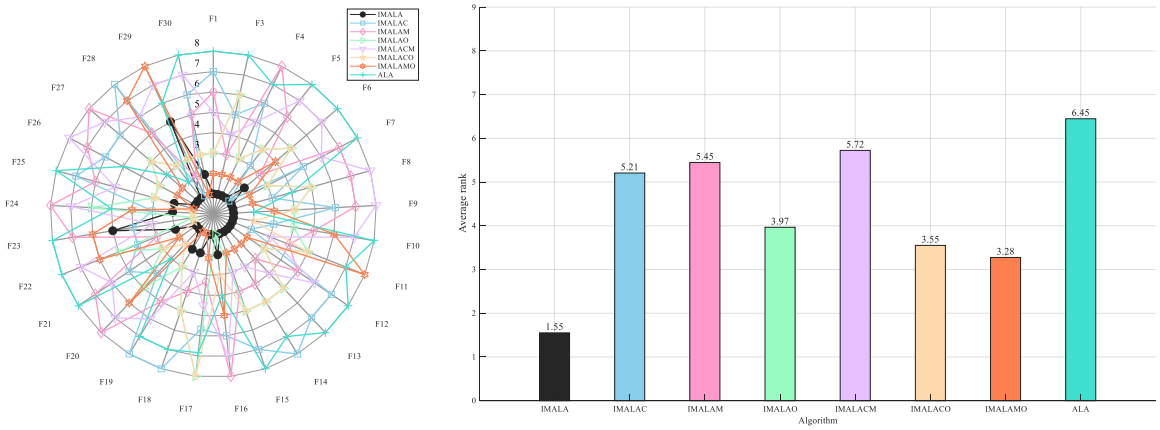


Figure 1. The flowchart of IMALA.

3.4. Ablation Study of Imala

To comprehensively evaluate the contributions of the three proposed improvement strategies, an extended ablation study was conducted on the CEC2017 benchmark functions using eight algorithm configurations: the original ALA, IMALAC (cubic chaotic initialization), IMALAM (differential current-to-best mutation), IMALAO (Beta-distribution-based stochastic opposition learning), IMALACM (cubic initialization and differential mutation), IMALACO (cubic initialization and opposition learning), IMALAMO (differential mutation and opposition learning), and the complete IMALA incorporating all three strategies. Radar charts and average-rank diagrams were employed to visualize the performance differences among these configurations.

As shown in Figure 2a, the performance of different variants varies across benchmark functions, indicating that the effectiveness of the improvement strategies depends on the characteristics of the optimization problems. Figure 2b shows that the complete IMALA achieves the lowest average rank of 1.55, compared with 6.45 for the original ALA. Among the individual strategies, IMALAO achieves the best average rank of 3.97, followed by IMALAC (5.21) and IMALAM (5.45), suggesting that Beta-distribution-based opposition learning provides the most pronounced individual improvement under the evaluated settings.



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

Figure 2. The ablation study of IMALA for CEC2017. (**a**) The radar chart. (**b**) The average rank chart.

Among the pairwise combinations, IMALAMO achieves the best average rank of 3.28, followed by IMALACO (3.55), while IMALACM obtains a relatively higher average rank of 5.72. Notably, the performance of IMALACM is weaker than that of IMALAC and IMALAM individually, indicating that combining two strategies does not necessarily guarantee additional improvements. Furthermore, removing cubic initialization, differential mutation, or opposition learning from the complete IMALA results in average ranks of 3.28, 3.55, and 5.72, respectively. These comparisons suggest that all three strategies contribute to the overall performance of IMALA, with the largest ranking difference observed when opposition learning is removed.

From an algorithmic perspective, cubic chaotic initialization aims to improve initial population diversity, differential mutation introduces additional search directions to support solution refinement, and Beta-distribution-based opposition learning generates alternative candidate solutions to facilitate exploration of promising regions. The superior overall ranking of the complete IMALA suggests that its performance benefits from the combined application of these complementary mechanisms rather than relying solely on a single strategy. Nevertheless, their contributions vary across algorithm configurations and

benchmark functions, and the observed ranking advantages should not be interpreted as universal superiority.

4. Experimental Analysis

4.1. Experimental Setting

To comprehensively evaluate the optimization performance of IMALA, eight representative algorithms were selected for comparison, including the original ALA, DE [40], FATA [41], SFOA [42], BKA [43], HHO [44], WUTP [45], and CFOA [46]. ALA provides the baseline for assessing the proposed modifications, DE serves as a classical evolutionary benchmark, and the remaining algorithms represent a range of nature-inspired search mechanisms. The experiments were conducted in MATLAB R2024a on a Windows 11 computer with an Intel® Core™ Ultra 9 processor. Each algorithm used a population of 30 and was run for a maximum of 300 iterations in each of 30 independent trials. This common iteration limit was chosen to compare solution quality, convergence behavior, and variability under the same number of iterations. Because the algorithms can make different numbers of objective-function calls within an iteration, the comparison did not impose an equal function-evaluation budget. The results should therefore be interpreted under this iteration-based protocol and do not establish an advantage at equal evaluation cost. The problem dimensions were 100 for CEC2017 and 10 for CEC2022.

4.2. Results and Analysis of Cec2017

The test functions CEC2017(F_1–F_{30}) are categorized into different kinds of functions to evaluate algorithm performance across varying landscape complexities, with optimal value (min), standard deviation (std), and average value (avg) presented in Tables 1 and 2.

The comprehensive experimental results presented in Tables 1 and 2 demonstrate the comparative performance of IMALA against eight algorithms across 30 benchmark functions from CEC test suites. Based on the data analysis (Tables 1 and 2), the following conclusions can be drawn.

- Unimodal Functions (F_1–F_3): All algorithms exhibit high performance on F_1 (global minimum), IMALA exhibits superior convergence precision, particularly evident in F_1 where it achieves minimum values of 5.253×10^9, outperforming all competitors by at least one order of magnitude. The consistently small standard deviations (3.056×10^9 for F_1) indicate remarkable algorithmic stability. For F_3, BKA and IMALA yield competitive minima (2.484×10^5 and 2.114×10^5, respectively), but DE shows the largest average error (8.881×10^5), indicating poor convergence consistency.
- Multimodal Challenges (F_4–F_{10}): In complex landscapes with multiple local optima, IMALA maintains competitive performance, notably in F_4 (min: 1.241×10^3) and F_7 (min: 2.066×10^3), where its hybrid mechanisms demonstrate effective escape from local traps. On F_6 (low-dimensional multimodal), most algorithms achieve similar minima.
- Hybrid/Composition Functions (F_{11}–F_{30}): For high-dimensional problems (100D in CEC2017), IMALA shows particular advantages in composition functions like F_{29} (min: 7.200×10^3 vs. CFOA's 8.912×10^4) and F_{30} (min: 1.571×10^6 vs. CFOA's 2.296×10^{10}), suggesting favorable performance on the tested 100-dimensional composition functions. The algorithm's adaptive mutation strategy demonstrates promising effectiveness in heterogeneous fitness landscapes, as reflected by its competitive performance across the F_{11}–F_{20} benchmark functions.

Table 1. Results of CEC2017 (Part 1).

F	Results	IMALA	ALA	DE	FATA	SFOA	BKA	HHO	WUTP	CFOA
F_1	min	5.253×10^{9}	3.014×10^{10}	4.052×10^{10}	1.380×10^{11}	1.298×10^{11}	1.437×10^{11}	7.013×10^{10}	2.459×10^{10}	2.584×10^{11}
	std	3.056×10^{9}	1.081×10^{10}	5.330×10^{9}	9.447×10^{9}	1.590×10^{10}	3.005×10^{10}	1.030×10^{10}	9.913×10^{9}	3.749×10^{10}
	avg	9.983×10^{9}	5.214×10^{10}	5.002×10^{10}	1.609×10^{11}	1.574×10^{11}	1.767×10^{11}	9.555×10^{10}	4.293×10^{10}	3.301×10^{11}
F_3	min	2.114×10^{5}	2.819×10^{5}	6.874×10^{5}	3.092×10^{5}	2.725×10^{5}	2.484×10^{5}	3.369×10^{5}	8.350×10^{5}	5.640×10^{5}
	std	4.219×10^{4}	3.619×10^{4}	8.378×10^{4}	7.654×10^{4}	9.006×10^{4}	3.846×10^{4}	1.432×10^{5}	1.236×10^{5}	3.704×10^{5}
	avg	2.720×10^{5}	3.668×10^{5}	8.881×10^{5}	3.670×10^{5}	3.620×10^{5}	2.985×10^{5}	3.936×10^{5}	1.106×10^{6}	1.010×10^{6}
F_4	min	1.241×10^{3}	4.422×10^{3}	6.439×10^{3}	2.103×10^{4}	2.132×10^{4}	1.739×10^{4}	1.332×10^{4}	3.063×10^{3}	7.479×10^{4}
	std	4.140×10^{2}	1.507×10^{3}	1.390×10^{3}	3.774×10^{3}	6.732×10^{3}	2.285×10^{4}	4.074×10^{3}	1.147×10^{3}	1.970×10^{4}
	avg	2.029×10^{3}	7.288×10^{3}	9.420×10^{3}	2.766×10^{4}	3.384×10^{4}	3.651×10^{4}	2.150×10^{4}	4.742×10^{3}	1.086×10^{5}
F_5	min	1.227×10^{3}	1.384×10^{3}	1.749×10^{3}	1.852×10^{3}	1.927×10^{3}	1.496×10^{3}	1.629×10^{3}	1.519×10^{3}	2.064×10^{3}
	std	1.462×10^{2}	1.079×10^{2}	3.873×10^{1}	4.886×10^{1}	6.558×10^{1}	1.675×10^{2}	6.557×10^{1}	5.646×10^{1}	1.283×10^{2}
	avg	1.465×10^{3}	1.592×10^{3}	1.821×10^{3}	1.957×10^{3}	2.063×10^{3}	1.651×10^{3}	1.734×10^{3}	1.638×10^{3}	2.266×10^{3}
F_6	min	6.466×10^{2}	6.552×10^{2}	6.494×10^{2}	6.978×10^{2}	6.926×10^{2}	6.723×10^{2}	6.831×10^{2}	6.360×10^{2}	7.012×10^{2}
	std	8.342×10^{0}	8.420×10^{0}	2.935×10^{0}	3.519×10^{0}	6.707×10^{0}	9.459×10^{0}	3.283×10^{0}	5.290×10^{0}	9.590×10^{0}
	avg	6.692×10^{2}	6.711×10^{2}	6.550×10^{2}	7.054×10^{2}	7.077×10^{2}	6.840×10^{2}	6.932×10^{2}	6.453×10^{2}	7.182×10^{2}
F_7	min	2.066×10^{3}	2.623×10^{3}	3.088×10^{3}	3.363×10^{3}	3.309×10^{3}	3.237×10^{3}	3.613×10^{3}	2.418×10^{3}	4.522×10^{3}
	std	1.616×10^{2}	2.269×10^{2}	1.366×10^{2}	1.391×10^{2}	1.744×10^{2}	1.571×10^{2}	1.177×10^{2}	1.638×10^{2}	9.268×10^{2}
	avg	2.386×10^{3}	2.932×10^{3}	3.293×10^{3}	3.652×10^{3}	3.660×10^{3}	3.462×10^{3}	3.836×10^{3}	2.679×10^{3}	5.961×10^{3}
F_8	min	1.495×10^{3}	1.708×10^{3}	1.998×10^{3}	2.329×10^{3}	2.307×10^{3}	1.872×10^{3}	1.989×10^{3}	1.875×10^{3}	2.509×10^{3}
	std	1.339×10^{2}	1.077×10^{2}	4.592×10^{1}	5.723×10^{1}	9.599×10^{1}	1.697×10^{2}	8.279×10^{1}	3.944×10^{1}	1.474×10^{2}
	avg	1.808×10^{3}	1.887×10^{3}	2.102×10^{3}	2.459×10^{3}	2.467×10^{3}	2.086×10^{3}	2.194×10^{3}	1.945×10^{3}	2.741×10^{3}
F_9	min	3.162×10^{4}	3.682×10^{4}	7.690×10^{4}	6.694×10^{4}	7.919×10^{4}	3.102×10^{4}	5.759×10^{4}	1.834×10^{4}	8.149×10^{4}
	std	7.368×10^{3}	1.254×10^{4}	1.058×10^{4}	5.175×10^{3}	5.595×10^{3}	1.090×10^{4}	5.486×10^{3}	6.910×10^{3}	2.147×10^{4}
	avg	4.952×10^{4}	5.786×10^{4}	9.880×10^{4}	7.785×10^{4}	8.863×10^{4}	4.261×10^{4}	7.027×10^{4}	3.038×10^{4}	1.237×10^{5}
F_{10}	min	2.802×10^{4}	2.064×10^{4}	3.062×10^{4}	3.135×10^{4}	2.987×10^{4}	1.820×10^{4}	2.246×10^{4}	3.168×10^{4}	3.198×10^{4}
	std	1.103×10^{3}	2.267×10^{3}	6.524×10^{2}	5.765×10^{2}	1.054×10^{3}	3.387×10^{3}	1.702×10^{3}	6.875×10^{2}	1.105×10^{3}
	avg	3.124×10^{4}	2.911×10^{4}	3.233×10^{4}	3.264×10^{4}	3.283×10^{4}	2.217×10^{4}	2.580×10^{4}	3.356×10^{4}	3.431×10^{4}
F_{11}	min	1.459×10^{4}	7.266×10^{4}	2.185×10^{5}	8.272×10^{4}	6.000×10^{4}	3.838×10^{4}	1.282×10^{5}	2.263×10^{5}	3.139×10^{5}
	std	9.197×10^{3}	2.461×10^{4}	4.449×10^{4}	2.328×10^{4}	3.689×10^{4}	3.483×10^{4}	3.862×10^{4}	1.072×10^{5}	2.317×10^{5}
	avg	3.138×10^{4}	1.180×10^{5}	3.353×10^{5}	1.236×10^{5}	1.176×10^{5}	9.175×10^{4}	2.136×10^{5}	4.723×10^{5}	6.096×10^{5}
F_{12}	min	1.933×10^{8}	3.307×10^{9}	1.127×10^{10}	5.878×10^{10}	3.060×10^{10}	3.891×10^{10}	2.356×10^{10}	2.545×10^{9}	1.378×10^{11}
	std	2.330×10^{8}	1.466×10^{9}	1.925×10^{9}	9.307×10^{9}	1.399×10^{10}	3.988×10^{10}	6.663×10^{9}	2.108×10^{9}	2.968×10^{10}
	avg	6.384×10^{8}	5.475×10^{9}	1.517×10^{10}	7.463×10^{10}	5.414×10^{10}	8.352×10^{10}	3.731×10^{10}	5.544×10^{9}	1.942×10^{11}

Table 1. *Cont.*

F	Results	IMALA	ALA	DE	FATA	SFOA	BKA	HHO	WUTP	CFOA
F_{13}	min	8.945×10^{4}	2.145×10^{7}	2.986×10^{7}	9.281×10^{9}	2.783×10^{9}	3.257×10^{9}	1.788×10^{9}	2.560×10^{7}	2.898×10^{10}
	std	1.602×10^{5}	6.819×10^{7}	6.522×10^{7}	1.926×10^{9}	6.794×10^{9}	5.391×10^{9}	1.268×10^{9}	1.246×10^{8}	6.383×10^{9}
	avg	3.017×10^{5}	7.945×10^{7}	1.110×10^{8}	1.177×10^{10}	9.976×10^{9}	1.074×10^{10}	3.420×10^{9}	1.151×10^{8}	4.392×10^{10}
F_{14}	min	4.644×10^{5}	3.104×10^{6}	3.097×10^{7}	1.040×10^{7}	3.225×10^{5}	1.382×10^{6}	6.399×10^{6}	1.489×10^{7}	2.590×10^{7}
	std	1.869×10^{6}	4.307×10^{6}	2.071×10^{7}	9.173×10^{6}	2.969×10^{6}	1.249×10^{7}	5.107×10^{6}	2.904×10^{7}	1.549×10^{8}
	avg	2.252×10^{6}	8.221×10^{6}	6.206×10^{7}	2.615×10^{7}	3.700×10^{6}	8.231×10^{6}	1.415×10^{7}	6.442×10^{7}	2.039×10^{8}
F_{15}	min	1.031×10^{4}	2.388×10^{5}	6.237×10^{6}	2.188×10^{9}	1.931×10^{8}	1.181×10^{8}	7.284×10^{7}	1.283×10^{6}	1.190×10^{10}
	std	1.346×10^{4}	1.906×10^{7}	2.614×10^{7}	8.298×10^{8}	5.282×10^{9}	3.276×10^{9}	4.087×10^{8}	6.482×10^{6}	7.260×10^{9}
	avg	2.920×10^{4}	5.277×10^{6}	3.515×10^{7}	3.452×10^{9}	4.330×10^{9}	3.162×10^{9}	4.951×10^{8}	8.181×10^{6}	2.361×10^{10}

Table 2. Results of CEC2017 (Part 2).

F	Results	IMALA	ALA	DE	FATA	SFOA	BKA	HHO	WUTP	CFOA
F_{16}	min	6.173×10^{3}	7.875×10^{3}	1.093×10^{4}	1.265×10^{4}	1.204×10^{4}	8.900×10^{3}	9.258×10^{3}	1.016×10^{4}	1.791×10^{4}
	std	1.309×10^{3}	9.144×10^{2}	4.751×10^{2}	1.683×10^{3}	1.316×10^{3}	2.756×10^{3}	1.606×10^{3}	5.396×10^{2}	3.135×10^{3}
	avg	8.617×10^{3}	9.370×10^{3}	1.210×10^{4}	1.469×10^{4}	1.394×10^{4}	1.260×10^{4}	1.195×10^{4}	1.138×10^{4}	2.280×10^{4}
F_{17}	min	4.169×10^{3}	5.442×10^{3}	7.977×10^{3}	1.335×10^{4}	8.975×10^{3}	5.341×10^{3}	6.818×10^{3}	7.424×10^{3}	7.095×10^{5}
	std	8.290×10^{2}	8.164×10^{2}	3.162×10^{2}	3.032×10^{5}	5.136×10^{4}	1.150×10^{6}	8.449×10^{3}	4.528×10^{2}	1.291×10^{7}
	avg	6.444×10^{3}	6.942×10^{3}	8.680×10^{3}	1.983×10^{5}	2.641×10^{4}	2.801×10^{5}	1.613×10^{4}	8.689×10^{3}	1.066×10^{7}
F_{18}	min	9.519×10^{5}	3.686×10^{6}	6.169×10^{7}	2.070×10^{7}	8.254×10^{5}	1.565×10^{6}	4.412×10^{6}	5.158×10^{7}	7.312×10^{7}
	std	2.092×10^{6}	4.708×10^{6}	3.215×10^{7}	2.110×10^{7}	5.388×10^{6}	7.412×10^{6}	6.824×10^{6}	5.997×10^{7}	2.001×10^{8}
	avg	3.021×10^{6}	9.402×10^{6}	1.131×10^{8}	4.594×10^{7}	9.055×10^{6}	6.074×10^{6}	1.443×10^{7}	1.396×10^{8}	2.992×10^{8}
F_{19}	min	8.162×10^{3}	8.437×10^{5}	2.201×10^{7}	1.473×10^{9}	2.392×10^{8}	9.673×10^{7}	3.621×10^{7}	1.834×10^{6}	1.390×10^{10}
	std	5.721×10^{4}	4.700×10^{6}	6.122×10^{7}	5.996×10^{8}	4.779×10^{9}	6.602×10^{9}	4.642×10^{8}	5.313×10^{7}	4.999×10^{9}
	avg	5.938×10^{4}	6.251×10^{6}	7.829×10^{7}	2.761×10^{9}	3.621×10^{9}	4.707×10^{9}	6.220×10^{8}	2.188×10^{7}	2.263×10^{10}
F_{20}	min	6.100×10^{3}	5.776×10^{3}	7.506×10^{3}	7.423×10^{3}	7.158×10^{3}	4.466×10^{3}	5.112×10^{3}	7.503×10^{3}	7.538×10^{3}
	std	5.512×10^{2}	6.582×10^{2}	2.455×10^{2}	2.471×10^{2}	4.177×10^{2}	6.003×10^{2}	6.332×10^{2}	3.106×10^{2}	4.878×10^{2}
	avg	7.207×10^{3}	7.035×10^{3}	8.063×10^{3}	7.921×10^{3}	8.117×10^{3}	5.753×10^{3}	6.324×10^{3}	8.121×10^{3}	8.526×10^{3}
F_{21}	min	3.016×10^{3}	3.237×10^{3}	3.558×10^{3}	3.942×10^{3}	4.069×10^{3}	3.804×10^{3}	4.108×10^{3}	3.373×10^{3}	4.470×10^{3}
	std	1.616×10^{2}	1.097×10^{2}	3.474×10^{1}	1.123×10^{2}	1.022×10^{2}	3.203×10^{2}	1.861×10^{2}	6.327×10^{1}	1.914×10^{2}
	avg	3.346×10^{3}	3.477×10^{3}	3.640×10^{3}	4.140×10^{3}	4.284×10^{3}	4.289×10^{3}	4.454×10^{3}	3.533×10^{3}	4.853×10^{3}

Table 2. *Cont.*

F	Results	IMALA	ALA	DE	FATA	SFOA	BKA	HHO	WUTP	CFOA
F_{22}	min	2.967×10^{4}	2.546×10^{4}	3.333×10^{4}	3.335×10^{4}	3.328×10^{4}	2.182×10^{4}	2.682×10^{4}	3.403×10^{4}	3.369×10^{4}
	std	1.182×10^{3}	2.275×10^{3}	5.423×10^{2}	6.082×10^{2}	7.767×10^{2}	3.372×10^{3}	1.073×10^{3}	5.829×10^{2}	1.475×10^{3}
	avg	3.330×10^{4}	3.072×10^{4}	3.463×10^{4}	3.481×10^{4}	3.504×10^{4}	2.556×10^{4}	2.874×10^{4}	3.551×10^{4}	3.654×10^{4}
F_{23}	min	3.582×10^{3}	3.706×10^{3}	3.872×10^{3}	5.200×10^{3}	4.509×10^{3}	4.675×10^{3}	5.292×10^{3}	3.842×10^{3}	5.215×10^{3}
	std	1.695×10^{2}	1.251×10^{2}	2.822×10^{1}	1.876×10^{2}	2.037×10^{2}	3.239×10^{2}	4.890×10^{2}	7.531×10^{1}	3.680×10^{2}
	avg	3.948×10^{3}	3.907×10^{3}	3.923×10^{3}	5.545×10^{3}	4.940×10^{3}	5.214×10^{3}	5.920×10^{3}	4.008×10^{3}	6.057×10^{3}
F_{24}	min	4.122×10^{3}	4.381×10^{3}	4.409×10^{3}	7.177×10^{3}	5.473×10^{3}	6.211×10^{3}	7.468×10^{3}	4.372×10^{3}	7.377×10^{3}
	std	2.562×10^{2}	2.204×10^{2}	3.312×10^{1}	5.471×10^{2}	5.414×10^{2}	1.003×10^{3}	7.702×10^{2}	1.224×10^{2}	1.135×10^{3}
	avg	4.770×10^{3}	4.748×10^{3}	4.468×10^{3}	8.610×10^{3}	6.164×10^{3}	7.170×10^{3}	8.592×10^{3}	4.531×10^{3}	9.564×10^{3}
F_{25}	min	4.223×10^{3}	6.101×10^{3}	1.294×10^{4}	1.001×10^{4}	1.257×10^{4}	9.921×10^{3}	8.115×10^{3}	6.838×10^{3}	2.508×10^{4}
	std	3.127×10^{2}	9.794×10^{2}	1.689×10^{3}	1.510×10^{3}	1.637×10^{3}	5.480×10^{3}	8.403×10^{2}	1.104×10^{3}	7.389×10^{3}
	avg	4.721×10^{3}	8.239×10^{3}	1.587×10^{4}	1.335×10^{4}	1.599×10^{4}	1.797×10^{4}	9.931×10^{3}	8.360×10^{3}	3.909×10^{4}
F_{26}	min	1.456×10^{4}	1.678×10^{4}	1.804×10^{4}	2.931×10^{4}	2.622×10^{4}	3.211×10^{4}	3.194×10^{4}	1.742×10^{4}	4.972×10^{4}
	std	1.648×10^{3}	1.829×10^{3}	3.297×10^{2}	2.694×10^{3}	4.045×10^{3}	5.316×10^{3}	1.687×10^{3}	9.985×10^{2}	5.480×10^{3}
	avg	1.719×10^{4}	1.963×10^{4}	1.862×10^{4}	3.576×10^{4}	3.445×10^{4}	3.830×10^{4}	3.501×10^{4}	1.885×10^{4}	5.856×10^{4}
F_{27}	min	3.612×10^{3}	3.660×10^{3}	4.591×10^{3}	8.435×10^{3}	4.118×10^{3}	5.160×10^{3}	5.840×10^{3}	3.567×10^{3}	1.005×10^{4}
	std	2.545×10^{2}	1.404×10^{2}	1.864×10^{2}	7.765×10^{2}	4.800×10^{2}	1.283×10^{3}	1.318×10^{3}	1.082×10^{2}	1.501×10^{3}
	avg	3.915×10^{3}	3.910×10^{3}	4.857×10^{3}	9.934×10^{3}	4.776×10^{3}	6.295×10^{3}	7.728×10^{3}	3.747×10^{3}	1.218×10^{4}
F_{28}	min	4.256×10^{3}	9.767×10^{3}	1.759×10^{4}	1.542×10^{4}	1.604×10^{4}	1.693×10^{4}	1.021×10^{4}	1.435×10^{4}	3.106×10^{4}
	std	5.921×10^{2}	4.076×10^{3}	6.024×10^{2}	1.060×10^{3}	2.384×10^{3}	3.657×10^{3}	1.060×10^{3}	9.613×10^{2}	5.963×10^{3}
	avg	5.507×10^{3}	1.483×10^{4}	1.896×10^{4}	1.796×10^{4}	2.008×10^{4}	2.084×10^{4}	1.298×10^{4}	1.715×10^{4}	4.153×10^{4}
F_{29}	min	7.200×10^{3}	8.216×10^{3}	1.096×10^{4}	1.579×10^{4}	1.332×10^{4}	1.111×10^{4}	1.223×10^{4}	1.049×10^{4}	8.912×10^{4}
	std	1.150×10^{3}	8.652×10^{2}	6.303×10^{2}	1.931×10^{4}	4.417×10^{4}	1.152×10^{5}	2.962×10^{3}	4.910×10^{2}	9.198×10^{5}
	avg	9.551×10^{3}	9.831×10^{3}	1.204×10^{4}	2.975×10^{4}	3.319×10^{4}	4.759×10^{4}	1.639×10^{4}	1.142×10^{4}	9.695×10^{5}
F_{30}	min	1.571×10^{6}	1.425×10^{7}	4.311×10^{7}	6.622×10^{9}	8.488×10^{8}	9.685×10^{8}	1.640×10^{9}	2.854×10^{7}	2.296×10^{10}
	std	2.070×10^{6}	2.494×10^{7}	2.537×10^{7}	1.699×10^{9}	5.510×10^{9}	4.725×10^{9}	1.884×10^{9}	4.558×10^{7}	8.839×10^{9}
	avg	4.113×10^{6}	4.371×10^{7}	9.596×10^{7}	9.802×10^{9}	6.801×10^{9}	8.747×10^{9}	4.045×10^{9}	7.361×10^{7}	4.127×10^{10}

The narrow standard deviation ranges (e.g., F_{16} std: 1.309×10^3 for IMALA vs. 3.135×10^3 for CFOA) across 30 independent runs confirm the IMALA's reliability. In most test cases, IMALA's average performance ranks in the top three among compared algorithms.

Statistical procedure: Tables 1 and 2 present the final fitness statistics for the 29 reported CEC2017 functions (F_1 and F_3–F_{30}). Table 3 reports the original (unadjusted) two-sided Wilcoxon rank-sum p-values for IMALA versus each of eight baselines. To control multiplicity, Holm's step-down procedure was applied separately to the 29 function-wise p-values for each baseline, with a family-wise significance level of 0.05. The $W/T/L$ rows were determined after this correction: W denotes a statistically significant comparison with a lower mean fitness for IMALA, L denotes a statistically significant comparison with a higher mean fitness, and T denotes no significant difference after adjustment. A non-significant result does not establish equivalence.

After Holm correction, the 232 function-level comparisons on CEC2017 comprise 205 statistically significant wins, 12 non-significant comparisons, and 15 statistically significant losses for IMALA. Against the original ALA, IMALA achieves 18/9/2, indicating improvements on many, but not all, functions. The corresponding counts are 26/1/2 against DE, 29/0/0 against FATA, 28/1/0 against SFOA, 25/0/4 against BKA, 26/0/3 against HHO, 24/1/4 against WUTP, and 29/0/0 against CFOA. These results support favorable performance under the evaluated settings while also showing that the advantages are problem-dependent. The $W/T/L$ counts are Holm-adjusted; the individual p-values displayed in Table 3 remain unadjusted.

A function-category analysis further reveals differences in the effectiveness of IMALA across optimization landscapes. Compared with ALA, IMALA achieves $W/T/L$ results of 2/0/0 on unimodal functions, 4/2/1 on multimodal functions, 7/3/0 on hybrid functions, and 5/4/1 on composition functions. The favorable results on hybrid functions suggest that IMALA is competitive in handling problems involving multiple interacting search components. However, the statistically significant losses against ALA on F_{10} and F_{22}, together with the competitive performance of BKA, HHO, and WUTP on certain functions, indicate that its relative advantages remain problem-dependent. These differences may reflect variations in how the algorithms respond to different fitness landscapes.

Overall, the statistical results support the effectiveness of IMALA under the adopted experimental settings while highlighting the limitations of its performance advantages. The combination of the three improvement strategies may contribute to the observed optimization performance, as suggested by the extended ablation study. Nevertheless, statistical significance alone does not establish the magnitude of improvement or universal superiority. Therefore, the $W/T/L$ results should be interpreted together with the average fitness values, convergence behavior, and ablation findings to provide a more comprehensive assessment of IMALA.

Table 3. Unadjusted Wilcoxon rank-sum p-values and Holm-adjusted $W/T/L$ statistics for CEC2017.

F	ALA	DE	FATA	SFOA	BKA	HHO	WUTP	CFOA
F_1	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}
F_3	2.439×10^{-9}	3.020×10^{-11}	3.825×10^{-9}	7.043×10^{-7}	4.226×10^{-3}	4.616×10^{-10}	3.020×10^{-11}	3.020×10^{-11}
F_4	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}
F_5	1.302×10^{-3}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	7.659×10^{-5}	1.174×10^{-9}	4.118×10^{-6}	3.020×10^{-11}
F_6	4.733×10^{-1}	9.260×10^{-9}	3.020×10^{-11}	3.020×10^{-11}	1.202×10^{-8}	3.020×10^{-11}	9.919×10^{-11}	3.020×10^{-11}
F_7	9.919×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	2.028×10^{-7}	3.020×10^{-11}
F_8	2.813×10^{-2}	3.338×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	9.260×10^{-9}	3.690×10^{-11}	6.356×10^{-5}	3.020×10^{-11}
F_9	3.671×10^{-3}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	2.278×10^{-5}	1.206×10^{-10}	4.200×10^{-10}	3.020×10^{-11}
F_{10}	3.835×10^{-6}	3.592×10^{-5}	2.572×10^{-7}	1.194×10^{-6}	3.497×10^{-9}	4.077×10^{-11}	3.159×10^{-10}	1.329×10^{-10}
F_{11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	8.153×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}
F_{12}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}
F_{13}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}
F_{14}	1.070×10^{-9}	3.020×10^{-11}	3.020×10^{-11}	8.236×10^{-2}	8.841×10^{-7}	6.696×10^{-11}	3.020×10^{-11}	3.020×10^{-11}
F_{15}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}
F_{16}	3.387×10^{-2}	4.077×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	1.857×10^{-9}	9.756×10^{-10}	8.993×10^{-11}	3.020×10^{-11}
F_{17}	7.483×10^{-2}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	2.034×10^{-9}	8.153×10^{-11}	3.690×10^{-11}	3.020×10^{-11}
F_{18}	1.429×10^{-8}	3.020×10^{-11}	3.020×10^{-11}	5.859×10^{-6}	4.427×10^{-3}	1.957×10^{-10}	3.020×10^{-11}	3.020×10^{-11}
F_{19}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}
F_{20}	2.772×10^{-1}	2.669×10^{-9}	2.377×10^{-7}	3.081×10^{-8}	1.287×10^{-9}	4.118×10^{-6}	3.825×10^{-9}	4.200×10^{-10}
F_{21}	1.236×10^{-3}	1.094×10^{-10}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	4.113×10^{-7}	3.020×10^{-11}
F_{22}	1.861×10^{-6}	7.043×10^{-7}	7.088×10^{-8}	1.429×10^{-8}	3.197×10^{-9}	5.494×10^{-11}	1.613×10^{-10}	2.439×10^{-9}
F_{23}	2.905×10^{-1}	4.643×10^{-1}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	1.224×10^{-1}	3.020×10^{-11}
F_{24}	6.204×10^{-1}	3.965×10^{-8}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	1.748×10^{-5}	3.020×10^{-11}
F_{25}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}
F_{26}	8.883×10^{-6}	4.084×10^{-5}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.157×10^{-5}	3.020×10^{-11}
F_{27}	5.011×10^{-1}	1.464×10^{-10}	3.020×10^{-11}	8.101×10^{-10}	3.020×10^{-11}	3.020×10^{-11}	6.549×10^{-4}	3.020×10^{-11}
F_{28}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}
F_{29}	3.255×10^{-1}	1.094×10^{-10}	3.020×10^{-11}	3.020×10^{-11}	4.504×10^{-11}	3.020×10^{-11}	3.825×10^{-9}	3.020×10^{-11}
F_{30}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	3.020×10^{-11}
W	18	26	29	28	25	26	24	29
T	9	1	0	1	0	0	1	0
L	2	2	0	0	4	3	4	0

These figures (Figures 3 and 4) display the average fitness over iterations for multiple algorithms (IMALA, ALA, DE, FATA, SFOA, BKA, HHO, WUTP, CFOA) across different test functions (F_1–F_{30}). Some algorithms, like IMALA in several functions (e.g., F_1, F_3), show rapid initial convergence. Their average fitness values drop sharply in the early iterations, indicating efficient exploration in the initial stages. For example, in the F_1 plot, IMALA's curve descends steeply compared to other algorithms, suggesting it quickly moves towards better solutions. Other algorithms, such as CFOA in many functions, exhibit slower convergence. Their curves decline more gradually, taking more iterations to reach relatively lower fitness values. This implies that they may have difficulty in achieving lower fitness values within the displayed iteration range. For relatively simple functions like F_1 and F_3, most algorithms can achieve decent convergence within a reasonable number of iterations. However, there are still differences in the speed and final accuracy of convergence. In more complex functions (e.g., F_{20}, F_{30}), the performance gap between algorithms becomes more pronounced. Some algorithms struggle to improve their fitness significantly, while others continue to show steady progress, highlighting the varying capabilities of algorithms in handling complex optimization landscapes.

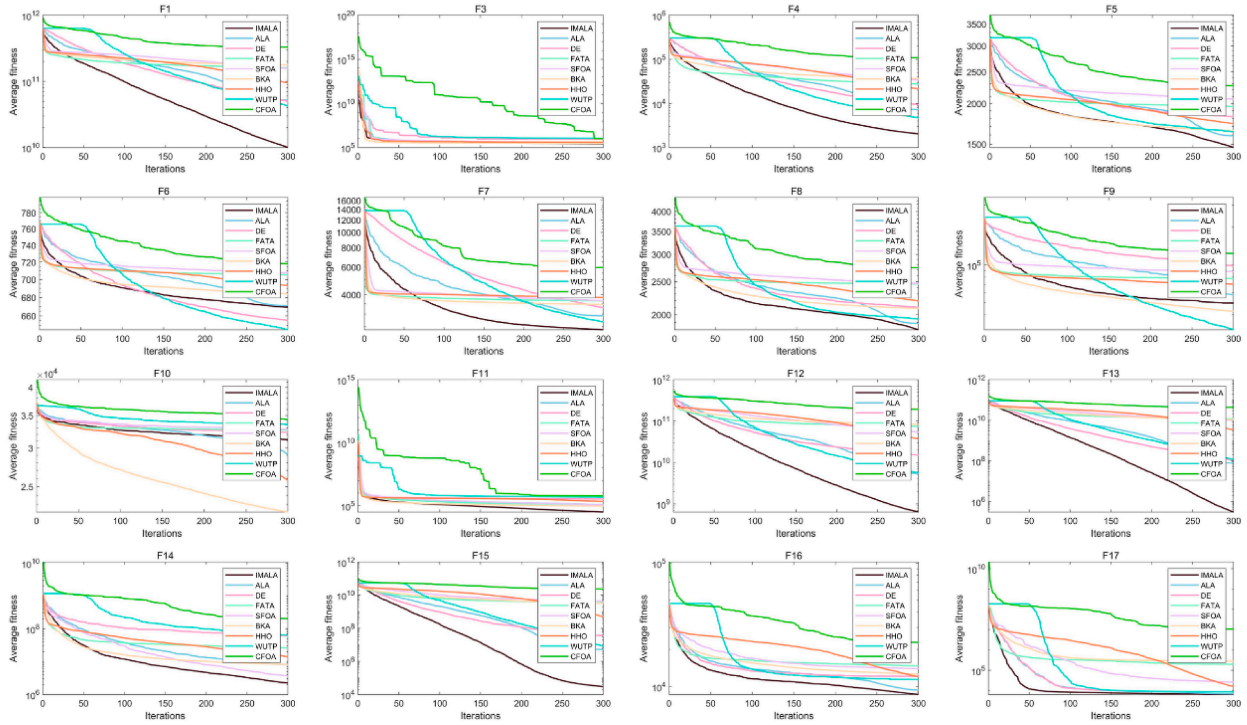


Figure 3. Convergence analysis results of the CEC2017 (Part 1).

The box plot graphs are presented as box-plots in Figures 5 and 6 show the distribution of fitness values for each algorithm across different test functions. In some functions (e.g., F_1), algorithms like IMALA have a relatively small inter-quartile range (IQR), indicating that the fitness values obtained in multiple runs are more consistent. The median is also relatively low, suggesting good overall performance. CFOA, on the other hand, often shows a large IQR in many functions, meaning there is a wide variation in the fitness values obtained in different runs. This implies instability in its performance, with some runs producing very good results and others much worse. Some algorithms have outliers in certain functions. For example, in the F3 box-plot, there are outliers for a few algorithms, which can indicate that in some runs, these algorithms deviated significantly from their typical performance. This could be due to issues like getting trapped in local optima or

having random fluctuations in the search process. Different functions exhibit different patterns of algorithm performance.

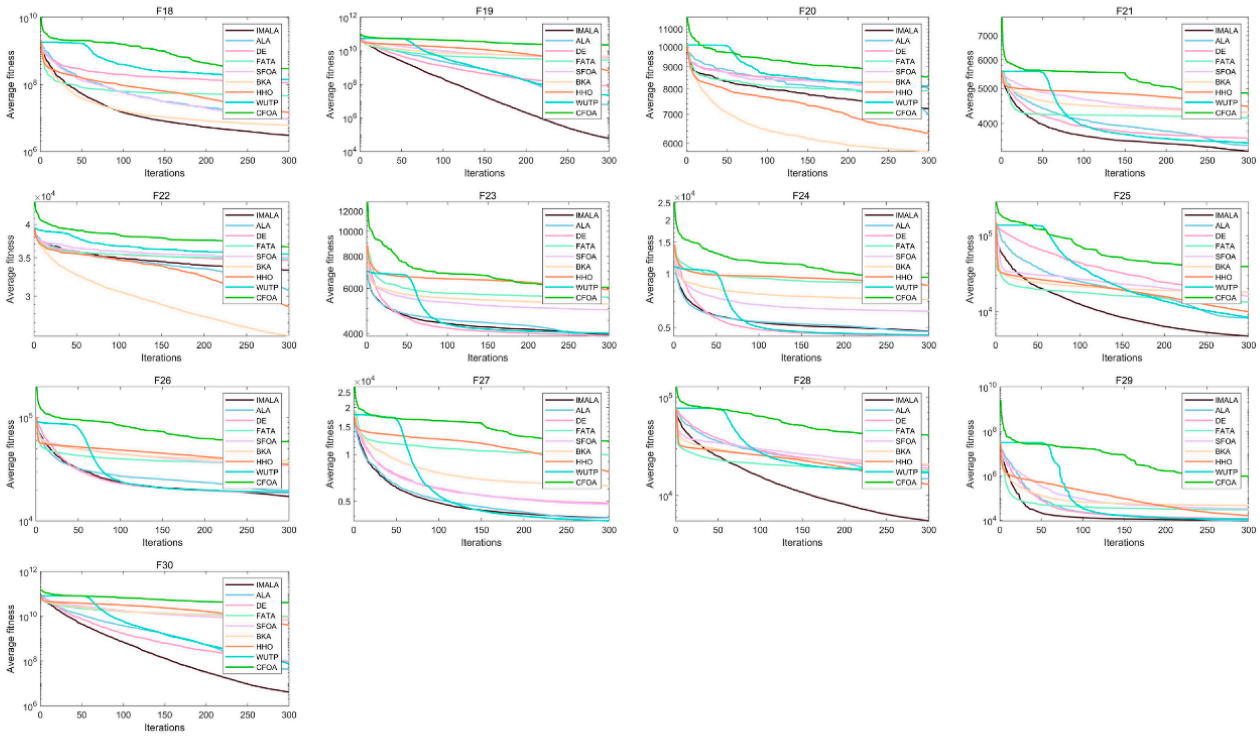


Figure 4. Convergence analysis results of the CEC2017 (Part 2).

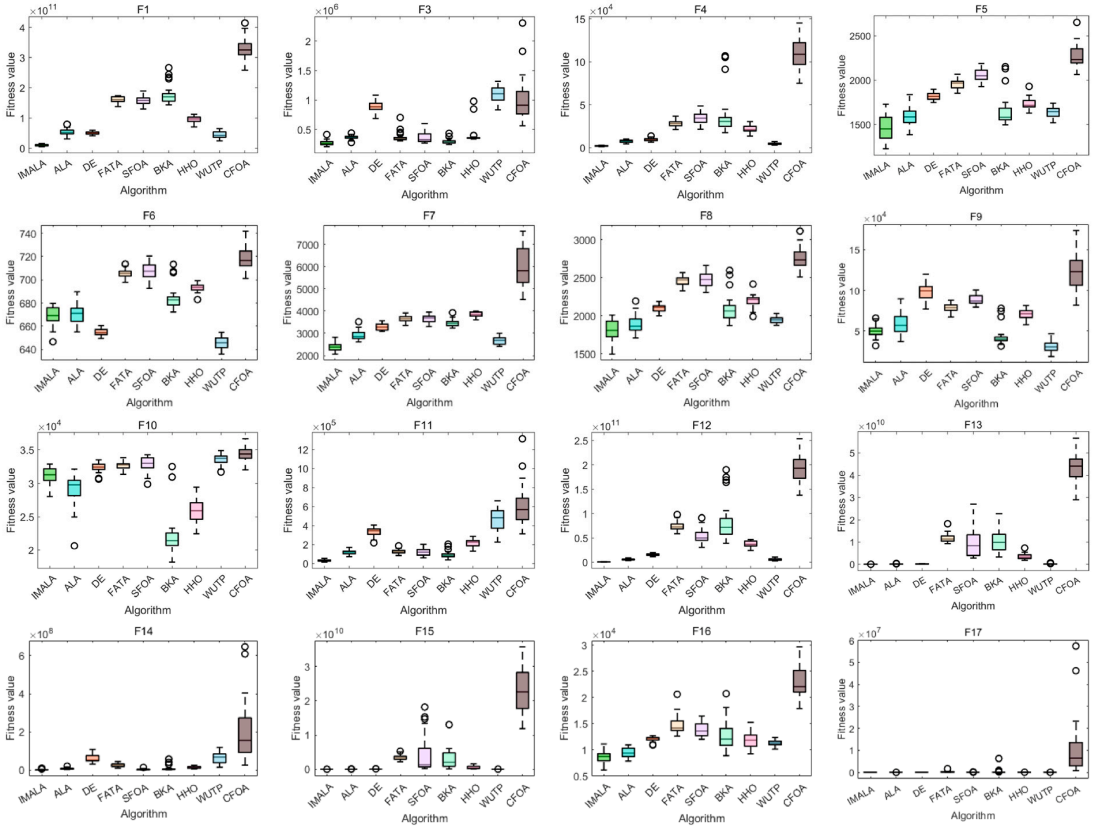


Figure 5. Box plots of fitness values of CEC2017 (Part 1).

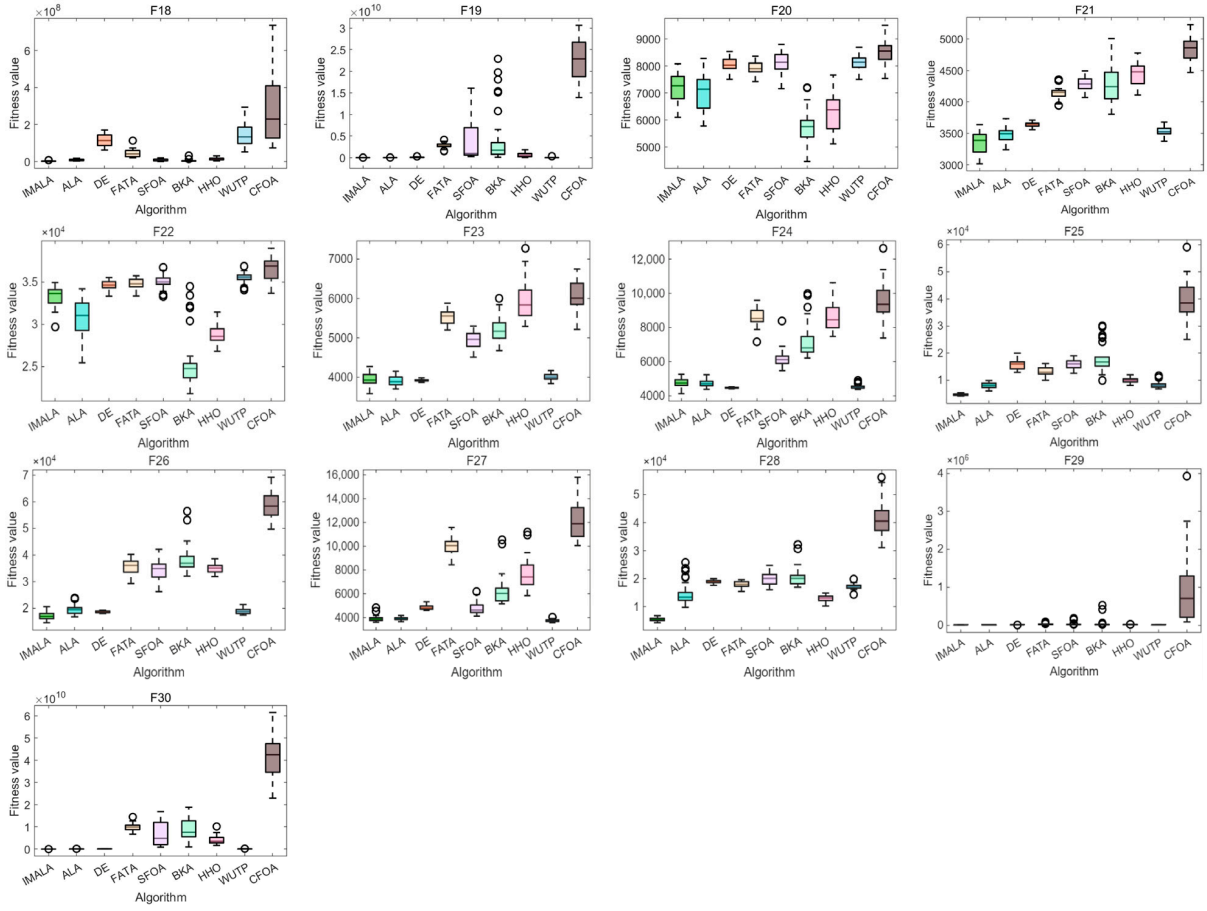


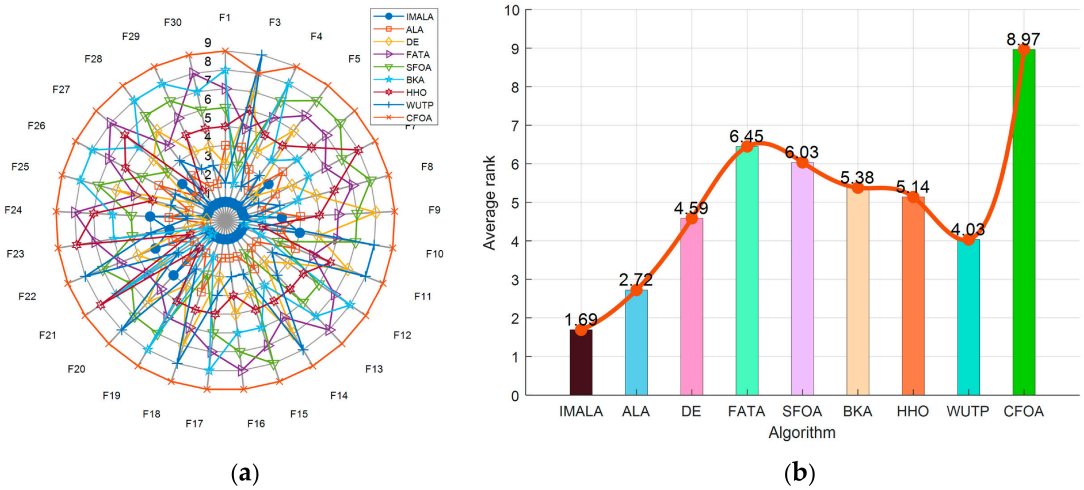
Figure 6. Box plots of fitness values of CEC2017 (Part 2).

A multi-dimensional radar chart (Figure 7a) shows the performance of different algorithms (IMALA, ALA, DE, FATA, SFOA, BKA, HHO, WUTP, CFOA) on different test functions (F_1–F_{30}). Each axis represents a test function, and the lines in different colors represent different algorithms. The farther the line is from the center, the better the performance of the corresponding algorithm on the corresponding test function. The IMALA is relatively close to the center on multiple test functions, indicating that its overall performance is relatively good and stable and excellent in most test functions. The bar chart (Figure 7b) shows the high and low average rankings of each algorithm, and the line chart connects the average rankings of each algorithm. The x-axis represents the algorithm names, and the y-axis represents the average ranking. The average ranking of IMALA is the lowest at 1.69, indicating that among all the algorithms' comprehensive comparisons, IMALA has the best overall performance. The average ranking of ALA is 2.72, and its performance is also quite good. The average ranking of CFOA is the highest at 8.97, meaning that among these algorithms, CFOA has the relatively poorest overall performance.

Overall, these two graphs provide different perspectives on the performance of each algorithm across multiple test functions. IMALA demonstrates outstanding overall performance, while CFOA lags behind.

Runtime Comparison: To assess the computational overhead of IMALA, its execution time was compared with that of the original ALA across 29 CEC2017 benchmark functions at 100 dimensions. The average execution times of IMALA and ALA were 0.04094 s and 0.01068 s, respectively, indicating that IMALA required approximately 3.83 times the runtime of ALA. IMALA exhibited longer execution times on all 29 functions, reflecting the additional computational costs introduced by the improvement strategies, particu-

larly the pairwise-distance calculations and extra objective function evaluations in the opposition-based learning module. These findings highlight a trade-off between optimization performance and computational cost. Since the experiments were conducted under equal iteration limits rather than equal function evaluation budgets, the runtime results should be interpreted within the adopted experimental settings rather than as a direct measure of computational efficiency under equal evaluation budgets. Detailed runtime results are provided in Table A2 of Appendix B.



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

Figure 7. Performance ranking chart of CEC2017. (**a**) The radar map; (**b**) the rank map.

4.3. Results and Analysis of Cec2022

Based on the results in Table 4, the nine algorithms exhibit different performance characteristics across the CEC2022 benchmark functions. Although IMALA obtains competitive results, it does not achieve the lowest minimum objective value on every function. Specifically, on F_2 and F_4, BKA achieves minimum fitness values of 4.000×10^2 and 8.050×10^2, respectively, which are lower than the corresponding values of IMALA (4.001×10^2 and 8.060×10^2). On F_{10}, DE obtains a lower minimum fitness value (2.469×10^3) than IMALA (2.500×10^3). However, IMALA achieves lower mean fitness values than BKA on F_2 (4.060×10^2 versus 4.132×10^2) and F_4 (8.144×10^2 versus 8.194×10^2), and a slightly lower mean value than DE on F_{10} (2.513×10^3 versus 2.517×10^3). These findings highlight the distinction between the best solution obtained in an individual run and the mean performance across 30 independent runs. Accordingly, the advantages of IMALA should be interpreted in relation to the particular performance metric and test function rather than as universal superiority. Overall, IMALA demonstrates competitive optimization performance under the adopted experimental settings, while BKA and DE achieve better minimum values on certain functions.

Statistical procedure: Table 5 reports the original (unadjusted) two-sided Wilcoxon rank-sum p-values for the 12 CEC2022 functions. Holm's step-down correction was applied separately across the 12 function-wise comparisons for each baseline at a family-wise significance level of 0.05, using the same $W/T/L$ procedure as in Table 3. On F_9, the reported mean fitness values of IMALA, DE, SFOA, and WUTP are identical at the displayed precision although the corresponding Wilcoxon p-values are significant. As the unrounded means are unavailable, these three comparisons are conservatively counted in T solely because the win/loss direction cannot be verified; they must not be interpreted as non-significant results or evidence of equivalence.

Table 4. Results of CEC2022.

F	Results	IMALA	ALA	DE	FATA	SFOA	BKA	HHO	WUTP	CFOA
	min	3.000×10^{2}	3.057×10^{2}	3.045×10^{3}	7.162×10^{2}	3.014×10^{2}	3.047×10^{2}	1.157×10^{3}	3.149×10^{3}	1.080×10^{4}
F_1	std	1.716×10^{-4}	6.882×10^{1}	3.941×10^{3}	2.391×10^{3}	1.677×10^{1}	1.946×10^{3}	1.175×10^{3}	4.330×10^{3}	9.767×10^{3}
	avg	3.000×10^{2}	3.657×10^{2}	1.055×10^{4}	3.452×10^{3}	3.136×10^{2}	1.202×10^{3}	3.394×10^{3}	1.065×10^{4}	2.546×10^{4}
	min	4.001×10^{2}	4.042×10^{2}	4.087×10^{2}	4.120×10^{2}	4.042×10^{2}	4.000×10^{2}	4.029×10^{2}	4.067×10^{2}	5.297×10^{2}
F_2	std	2.639×10^{0}	1.465×10^{0}	1.398×10^{0}	2.382×10^{1}	1.733×10^{0}	2.012×10^{1}	7.668×10^{1}	8.286×10^{-1}	4.601×10^{2}
	avg	4.060×10^{2}	4.075×10^{2}	4.102×10^{2}	4.576×10^{2}	4.068×10^{2}	4.132×10^{2}	4.848×10^{2}	4.079×10^{2}	9.579×10^{2}
	min	6.000×10^{2}	6.000×10^{2}	6.000×10^{2}	6.170×10^{2}	6.010×10^{2}	6.071×10^{2}	6.174×10^{2}	6.001×10^{2}	6.320×10^{2}
F_3	std	4.634×10^{-2}	4.024×10^{-1}	3.423×10^{-3}	1.018×10^{1}	1.504×10^{0}	1.117×10^{1}	9.318×10^{0}	1.511×10^{0}	1.603×10^{1}
	avg	6.001×10^{2}	6.004×10^{2}	6.000×10^{2}	6.345×10^{2}	6.033×10^{2}	6.282×10^{2}	6.436×10^{2}	6.006×10^{2}	6.510×10^{2}
	min	8.060×10^{2}	8.099×10^{2}	8.245×10^{2}	8.284×10^{2}	8.219×10^{2}	8.050×10^{2}	8.112×10^{2}	8.284×10^{2}	8.265×10^{2}
F_4	std	6.230×10^{0}	7.834×10^{0}	6.892×10^{0}	6.016×10^{0}	1.125×10^{1}	6.793×10^{0}	6.095×10^{0}	6.019×10^{0}	1.897×10^{1}
	avg	8.144×10^{2}	8.221×10^{2}	8.387×10^{2}	8.382×10^{2}	8.432×10^{2}	8.194×10^{2}	8.247×10^{2}	8.409×10^{2}	8.605×10^{2}
	min	9.000×10^{2}	9.003×10^{2}	9.032×10^{2}	9.263×10^{2}	9.026×10^{2}	9.428×10^{2}	1.092×10^{3}	9.000×10^{2}	1.183×10^{3}
F_5	std	2.735×10^{-1}	1.249×10^{1}	2.096×10^{1}	2.103×10^{2}	8.398×10^{1}	1.147×10^{2}	1.803×10^{2}	9.339×10^{-2}	5.097×10^{2}
	avg	9.002×10^{2}	9.090×10^{2}	9.223×10^{2}	1.164×10^{3}	9.536×10^{2}	1.141×10^{3}	1.448×10^{3}	9.001×10^{2}	1.736×10^{3}
	min	1.806×10^{3}	1.926×10^{3}	1.029×10^{4}	3.886×10^{4}	1.861×10^{3}	1.884×10^{3}	1.878×10^{3}	6.031×10^{3}	9.984×10^{4}
F_6	std	2.942×10^{1}	1.719×10^{3}	1.846×10^{4}	5.242×10^{5}	6.287×10^{1}	1.608×10^{3}	4.926×10^{3}	4.261×10^{5}	4.993×10^{7}
	avg	1.844×10^{3}	3.096×10^{3}	3.083×10^{4}	4.480×10^{5}	1.955×10^{3}	3.334×10^{3}	8.133×10^{3}	2.107×10^{5}	2.960×10^{7}
	min	2.008×10^{3}	2.003×10^{3}	2.006×10^{3}	2.050×10^{3}	2.024×10^{3}	2.023×10^{3}	2.040×10^{3}	2.022×10^{3}	2.037×10^{3}
F_7	std	6.469×10^{0}	5.212×10^{0}	4.781×10^{0}	2.046×10^{1}	4.638×10^{0}	1.609×10^{1}	2.573×10^{1}	9.664×10^{0}	3.013×10^{1}
	avg	2.025×10^{3}	2.022×10^{3}	2.016×10^{3}	2.081×10^{3}	2.031×10^{3}	2.046×10^{3}	2.089×10^{3}	2.036×10^{3}	2.095×10^{3}
	min	2.213×10^{3}	2.208×10^{3}	2.215×10^{3}	2.228×10^{3}	2.220×10^{3}	2.210×10^{3}	2.222×10^{3}	2.224×10^{3}	2.236×10^{3}
F_8	std	4.356×10^{0}	4.720×10^{0}	3.126×10^{0}	2.659×10^{0}	2.561×10^{0}	4.915×10^{0}	1.173×10^{1}	3.500×10^{0}	8.633×10^{1}
	avg	2.224×10^{3}	2.222×10^{3}	2.222×10^{3}	2.233×10^{3}	2.227×10^{3}	2.226×10^{3}	2.234×10^{3}	2.230×10^{3}	2.302×10^{3}
	min	2.529×10^{3}	2.529×10^{3}	2.529×10^{3}	2.545×10^{3}	2.529×10^{3}	2.529×10^{3}	2.539×10^{3}	2.529×10^{3}	2.657×10^{3}
F_9	std	9.818×10^{-10}	3.285×10^{1}	3.271×10^{-3}	3.359×10^{1}	7.384×10^{-5}	3.221×10^{1}	5.531×10^{1}	1.243×10^{-1}	7.077×10^{1}
	avg	2.529×10^{3}	2.537×10^{3}	2.529×10^{3}	2.597×10^{3}	2.529×10^{3}	2.544×10^{3}	2.634×10^{3}	2.529×10^{3}	2.755×10^{3}
	min	2.500×10^{3}	2.500×10^{3}	2.469×10^{3}	2.504×10^{3}	2.500×10^{3}	2.500×10^{3}	2.501×10^{3}	2.500×10^{3}	2.508×10^{3}
F_{10}	std	3.712×10^{1}	4.625×10^{1}	4.188×10^{1}	7.662×10^{1}	4.050×10^{1}	1.873×10^{2}	6.822×10^{1}	3.819×10^{2}	4.266×10^{2}
	avg	2.513×10^{3}	2.523×10^{3}	2.517×10^{3}	2.612×10^{3}	2.514×10^{3}	2.622×10^{3}	2.602×10^{3}	2.719×10^{3}	2.751×10^{3}
	min	2.600×10^{3}	2.600×10^{3}	2.764×10^{3}	2.783×10^{3}	2.619×10^{3}	2.606×10^{3}	2.685×10^{3}	2.662×10^{3}	5.507×10^{3}
F_{11}	std	1.400×10^{2}	1.309×10^{2}	3.066×10^{1}	3.771×10^{2}	9.040×10^{1}	1.970×10^{2}	1.429×10^{2}	8.801×10^{1}	8.540×10^{3}
	avg	2.803×10^{3}	2.804×10^{3}	2.892×10^{3}	3.166×10^{3}	2.703×10^{3}	2.854×10^{3}	2.873×10^{3}	2.863×10^{3}	1.609×10^{4}
	min	2.859×10^{3}	2.859×10^{3}	2.860×10^{3}	2.865×10^{3}	2.859×10^{3}	2.862×10^{3}	2.869×10^{3}	2.859×10^{3}	2.891×10^{3}
F_{12}	std	2.220×10^{0}	3.407×10^{0}	9.821×10^{-1}	6.412×10^{0}	2.038×10^{0}	4.359×10^{0}	6.553×10^{1}	2.323×10^{0}	7.273×10^{1}
	avg	2.863×10^{3}	2.864×10^{3}	2.862×10^{3}	2.872×10^{3}	2.862×10^{3}	2.867×10^{3}	2.948×10^{3}	2.863×10^{3}	2.962×10^{3}

Table 5. Unadjusted Wilcoxon rank-sum p-values and Holm-adjusted W/T/L statistics for CEC2022.

F	ALA	DE	FATA	SFOA	BKA	HHO	WUTP	CFOA
F_1	6.796×10^{-8}	6.796×10^{-8}	6.796×10^{-8}	6.796×10^{-8}	6.796×10^{-8}	6.796×10^{-8}	6.796×10^{-8}	6.796×10^{-8}
F_2	1.436×10^{-2}	6.917×10^{-7}	6.796×10^{-8}	6.787×10^{-2}	1.075×10^{-1}	8.597×10^{-6}	9.786×10^{-3}	6.796×10^{-8}
F_3	2.596×10^{-5}	1.803×10^{-6}	6.796×10^{-8}	6.796×10^{-8}	6.796×10^{-8}	6.796×10^{-8}	1.047×10^{-6}	6.796×10^{-8}
F_4	1.782×10^{-3}	1.235×10^{-7}	6.796×10^{-8}	1.235×10^{-7}	1.332×10^{-2}	4.680×10^{-5}	6.796×10^{-8}	1.065×10^{-7}
F_5	2.960×10^{-7}	6.796×10^{-8}	6.796×10^{-8}	6.796×10^{-8}	6.796×10^{-8}	6.796×10^{-8}	3.235×10^{-1}	6.796×10^{-8}
F_6	6.796×10^{-8}	6.796×10^{-8}	6.796×10^{-8}	2.563×10^{-7}	9.173×10^{-8}	9.173×10^{-8}	6.796×10^{-8}	6.796×10^{-8}
F_7	5.560×10^{-3}	5.874×10^{-6}	6.796×10^{-8}	7.579×10^{-4}	2.302×10^{-5}	6.796×10^{-8}	1.610×10^{-4}	7.898×10^{-8}
F_8	3.605×10^{-2}	9.045×10^{-3}	1.235×10^{-7}	3.336×10^{-3}	2.853×10^{-1}	1.444×10^{-4}	7.406×10^{-5}	6.796×10^{-8}
F_9	1.376×10^{-5}	6.786×10^{-8}	6.786×10^{-8}	6.786×10^{-8}	6.786×10^{-8}	6.786×10^{-8}	6.786×10^{-8}	6.786×10^{-8}
F_{10}	9.892×10^{-1}	5.091×10^{-4}	5.227×10^{-7}	1.294×10^{-4}	1.610×10^{-4}	3.939×10^{-7}	4.166×10^{-5}	5.227×10^{-7}
F_{11}	1.548×10^{-2}	2.690×10^{-6}	9.786×10^{-3}	1.719×10^{-1}	8.585×10^{-2}	3.152×10^{-2}	4.155×10^{-4}	6.796×10^{-8}
F_{12}	8.181×10^{-1}	9.620×10^{-2}	3.416×10^{-7}	1.017×10^{-1}	1.481×10^{-3}	6.796×10^{-8}	8.392×10^{-1}	6.796×10^{-8}
W	6	7	12	8	8	12	9	12
T	5	2	0	4	4	0	3	0
L	1	3	0	0	0	0	0	0

The Holm-adjusted $W/T/L$ statistics for CEC2022 comprise 74 wins, 18 comparisons without an established significant direction, and 4 losses across 96 function-level comparisons. Of the 18 entries classified as T, 15 are non-significant after Holm correction and three concern F_9, where significant p-values are accompanied by identical rounded mean fitness values. IMALA achieves 12/0/0 against FATA, HHO, and CFOA; however, its 6/5/1 result against ALA and 7/2/3 against DE reveal that the benefits are not uniform. Compared with the CEC2017 outcomes, the lower share of wins on CEC2022 suggests greater dependence on the evaluated test problems and settings. These counts describe performance under the current experimental protocol and do not establish universal superiority.

Friedman test across algorithms: Using the mean fitness values reported in Tables 1, 2 and 4 to rank the nine algorithms within each function, the Friedman test yielded $\chi^2(8) = 135.0161$ ($p = 2.576 \times 10^{-25}$) for CEC2017 (29 functions) and $\chi^2(8) = 63.6020$ ($p = 9.115 \times 10^{-11}$) for CEC2022 (12 functions). Both results reject the null hypothesis of identical overall ranks at $a = 0.05$. The Friedman test establishes an overall rank difference, not significant superiority in every pairwise comparison; the Holm-adjusted Wilcoxon $W/T/L$ statistics provide complementary pairwise evidence. Because the available mean values are rounded, these Friedman statistics are estimates calculated at the published precision and should be verified against the unrounded results.

Figure 8 compares the mean fitness trajectories of IMALA and eight competing algorithms on the CEC2022 functions F_1–F_{12} over 300 iterations. IMALA showed a pronounced decrease in fitness during the early iterations on several functions and attained comparatively low final fitness values. On F_1 and F_3, several algorithms also achieved substantial fitness reductions, although their trajectories and final values differed. Across the other functions, the separation between curves varied, with some algorithms improving more gradually or plateauing. Overall, IMALA showed favorable convergence in terms of iteration count on several functions, but its advantage depended on the test function. Because the algorithms may require different numbers of objective-function evaluations per iteration, Figure 8 describes convergence over iterations rather than performance under an equal evaluation budget.

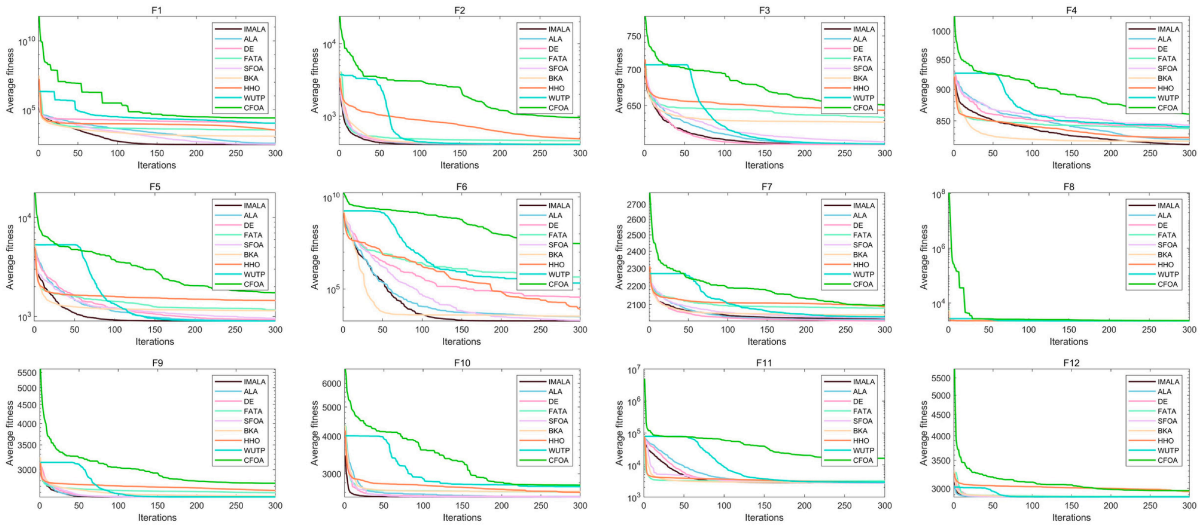


Figure 8. Convergence analysis results of the CEC2022.

The distribution of fitness values of the above nine algorithms under the F_1–F_{12} test functions shows in box plots (box plot results) (Figure 9). In many test functions, IMALA has a compact box and a lower median, indicating a concentrated distribution of fitness values and good stability, and the ability to output a relatively optimal result stably; CFOA's box is generally wider, and there are many outliers, indicating large fluctuations in fitness values and poor stability, and unstable algorithm performance. Different algorithms under the same test function have differences in box positions, lengths, and the number of outliers, reflecting the different stability of algorithm performance and the degree of result dispersion.

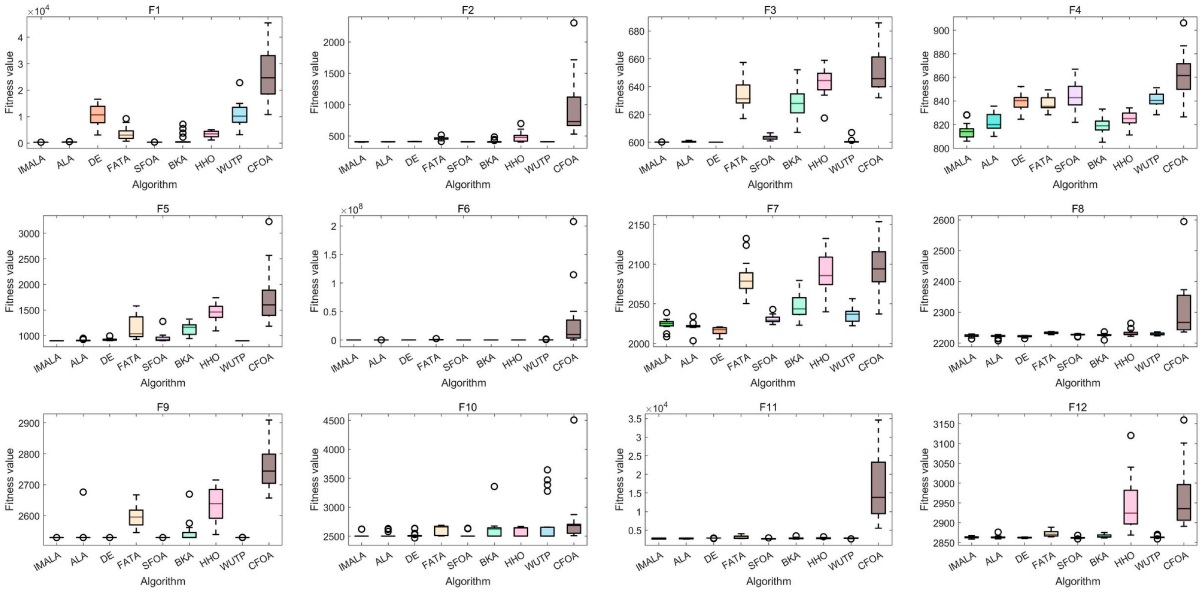
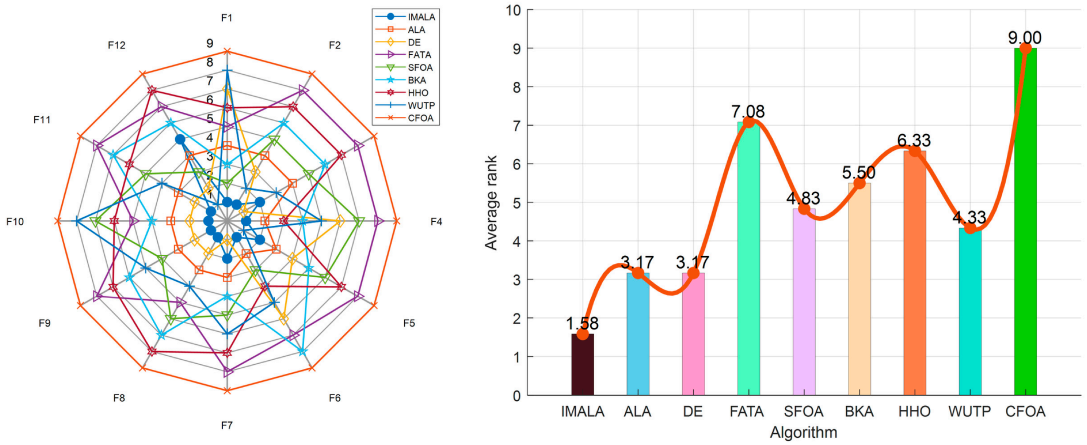


Figure 9. Box plots of fitness values of CEC2022.

The performance ranking chart of CEC2022 shows in Figure 10. IMALA is close to the center in most dimensions, indicating stable and good overall performance in different test functions; CFOA is far from the center in some dimensions, indicating poor performance in certain test functions and large fluctuations in performance. In the bar–line mixed graph, the IMALA has the best overall performance with the lowest average ranking (1.58); CFOA has the worst overall performance with the highest average ranking (9.00). These two

graphs comprehensively reflect the performance superiority and overall ranking of each algorithm under multiple test functions from different perspectives.



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

Figure 10. Performance ranking chart of CEC2022. (**a**) The radar map; (**b**) the rank map.

In the CEC2017 and CEC2022 experiments, IMALA achieved favorable solution quality across the nine compared algorithms, with average ranks of 1.69 and 1.58, respectively. Its iteration-based convergence curves show a faster early decrease in objective value than some competitors on selected functions, including F_1 and F_3. The box plots also show relatively compact fitness distributions on several functions, suggesting more consistent outcomes across independent runs. These observations are consistent with the intended roles of chaotic initialization, differential mutation, and Beta-distribution-based opposition learning. Overall, the results suggest that IMALA is a promising option for the benchmark problems tested under the reported conditions.

5. Solving Engineering Problems by Imala

IMALA was applied to three constrained engineering design problems: step-cone pulleys, hydrostatic thrust bearings, and piston levers. The engineering objective function was evaluated after handling the variable bounds and nonlinear constraints as described below.

For the IMALA implementation, each candidate variable outside its prescribed bounds was clipped to the nearest bound. The engineering APP incorporated nonlinear constraints into a penalized fitness $F(x) = f(x) + 10^{15}\,[\Sigma_i \max(0, g_i(x))^2 + \Sigma_j h_j(x)^2]$, where $g_i(x) \le 0$ and $h_j(x) = 0$. Candidate solutions were compared using F, the original objective f and constraint residuals are reported separately in Tables A3–A5 in Appendix C.

Feasibility was checked independently against the variable bounds and $g_i \le 0$. For the step-cone pulley equality constraints, $|h_j| \le 10^{-4}$ was used as the feasibility tolerance; the APP penalty itself uses the unthresholded squared residuals. The piston lever DE solution has $g_4 = +1.51 \times 10^{-14}$ and is identified separately as a numerical-boundary case.

5.1. Step-Cone Pulley Problem

This study addresses the weight minimization of a four-stage conical pulley, formulated as a constrained nonlinear optimization problem with five design variables [47]:

- Four continuous variables representing the diameter of each pulley stage (d_1, d_2, d_3, d_4).
- One variable defining the pulley width (w).

The objective function is subject to 11 nonlinear constraints to ensure a transmission power output of 0.75 horsepower. The mathematical formulation can be expressed as follows:

Minimize

$$f(\vec{x}) = \frac{\pi\rho\omega}{4}\left[d_1^2\left(11 + \left(\frac{N_1}{N}\right)^2\right) + d_2^2\left(1 + \left(\frac{N_2}{N}\right)^2\right) + d_3^2\left(1 + \left(\frac{N_3}{N}\right)^2\right) + d_4^2\left(1 + \left(\frac{N_4}{N}\right)^2\right)\right] \tag{24}$$

subject to

$$h_1(\vec{x})\& = C_1 - C_2 = 0 \tag{25}$$

$$h_2(\vec{x})\& = C_1 - C_3 = 0 \tag{26}$$

$$h_3(\vec{x})\& = C_1 - C_4 = 0 \tag{27}$$

The inequality constraints are

$$\begin{aligned} &g_i(x) = 2 - R_i \le 0, \qquad i = 1,\ 2,\ 3,\ 4 \\ &g_{i=+4}(x) = (0.75 \times 745.6998) - P_i \le 0, \quad i = 1,\ 2,\ 3,\ 4 \end{aligned} \tag{28}$$

where

$$C_i = \frac{\pi d_i}{2}\left(1 + \frac{N_i}{N}\right) + \frac{\left(\frac{N_i}{N} - 1\right)^2 d_i^2}{4a} + 2a,\ i = 1,2,3,4 \tag{29}$$

$$R_i = \exp\left(\mu\left(\pi - 2\sin^{-1}\left(\frac{\left(\frac{N_i}{N} - 1\right)d_i}{2a}\right)\right)\right),\ i = 1,2,3,4 \tag{30}$$

$$P_i = st\omega\left(1 - \frac{1}{R_i}R_i\right)\frac{\pi d_i N_i}{60},\ i = 1,2,3,4 \tag{31}$$

$$t = 8 \times 10^{-3}\ \text{m},\ s = 1.75 \times 10^6\ \text{Pa}, \mu = 0.35,\ \rho = 7200\ \text{kg/m}^3, a = 3\ \text{m}. \tag{32}$$

The comparative analysis of step-cone pulley design optimization results demonstrates IMALA's superior performance across all evaluation metrics. The parameter optimization table (Table 6) reveals IMALA achieves the most compact pulley design with the smallest diameters ($d_1 = 38.9068$ to $d_4 = 85.5795$) and near-optimal width ($w = 89.3019$), while the performance metrics table (Table 7) confirms its exceptional solution quality with the lowest minimum weight (1.64×10^1), smallest standard deviation (1.85×10^{-1}), and best average (1.66×10^1) and median (1.66×10^1) values. This outstanding consistency is particularly evident when compared to other algorithms like CFOA, which shows extreme performance fluctuations (std = 6.66×10^{12}) and substantially worse solutions (avg = 3.79×10^{12}). IMALA's design parameters and performance metrics collectively indicate it provides both the most efficient pulley configuration and the most reliable optimization process among the tested algorithms.

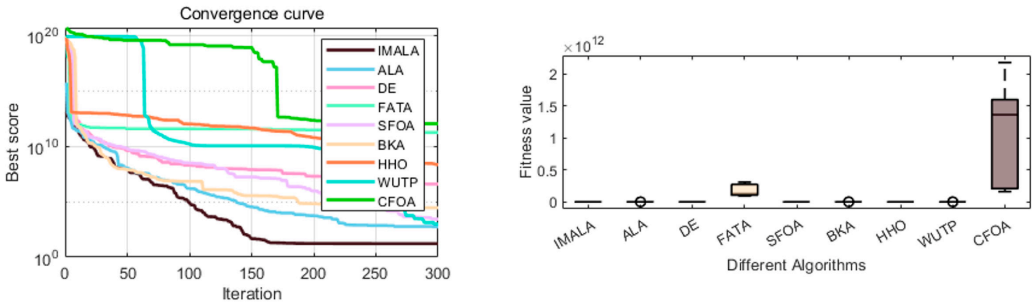
Table 6. Step-cone pulley design optimization parameter results.

Variable	IMALA	ALA	DE	FATA	SFOA	BKA	HHO	WUTP	CFOA
d_1	38.9068	39.9813	40.9169	42.5359	40.8714	39.6664	40.6473	40.9168	40.9060
d_2	53.5373	55.0170	56.3056	60.0000	56.2428	54.5834	55.9380	56.3054	57.1107
d_3	71.3774	73.3501	75.0671	78.5275	74.9842	72.7720	74.5681	75.0676	74.7480
d_4	85.5795	87.9426	90.0000	90.0000	89.9001	87.2502	89.4074	90.0000	89.9881
w	89.3019	87.8798	84.5071	90.0000	90.0000	89.5779	89.4032	84.5835	89.7744
Best F	16.3899	24.0682	2630.2669	9.35×10^{10}	30.6692	17.0765	3.41×10^5	17.1750	9.33×10^9
rank	1	4	6	9	5	2	7	3	8

Table 7. Step-cone pulley design optimization results.

Alg.	IMALA	ALA	DE	FATA	SFOA	BKA	HHO	WUTP	CFOA
min	1.64×10^{1}	2.41×10^{1}	2.63×10^{3}	9.35×10^{10}	3.07×10^{1}	1.71×10^{1}	3.41×10^{5}	1.72×10^{1}	9.33×10^{9}
worst	1.69×10^{1}	3.06×10^{2}	1.16×10^{7}	2.82×10^{11}	1.09×10^{4}	5.42×10^{10}	4.56×10^{8}	2.44×10^{6}	2.10×10^{13}
std	1.85×10^{-1}	1.13×10^{2}	3.70×10^{6}	5.87×10^{10}	3.33×10^{3}	2.22×10^{10}	1.73×10^{8}	7.71×10^{5}	6.66×10^{12}
avg	1.66×10^{1}	9.41×10^{1}	2.41×10^{6}	1.64×10^{11}	1.65×10^{3}	1.05×10^{10}	1.02×10^{8}	2.46×10^{5}	3.79×10^{12}
median	1.66×10^{1}	3.41×10^{1}	9.50×10^{5}	1.45×10^{11}	3.52×10^{2}	1.78×10^{1}	1.22×10^{7}	1.83×10^{1}	6.58×10^{11}

The convergence curve graph (Figure 11a) shows the changes in the optimal scores of nine algorithms such as IMALA and ALA over the number of iterations. The optimal score of the IMALA algorithm decreases significantly and rapidly, and can reach a lower level within a relatively small number of iterations, demonstrating its strong convergence ability, while other algorithms such as CFOA, etc., show a relatively slower decrease in score, which remains at a relatively high level even after the same number of iterations. The box plot graph (Figure 11b) presents the distribution of fitness values of each algorithm. The IMALA is compact and located on the left side box, indicating that its fitness values are concentrated and overall lower, with better stability and optimization effect; the box of the CFOA algorithm is scattered and located on the right side, suggesting that its fitness values fluctuate greatly, and its stability is poor.



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

Figure 11. Convergence curve and box plot of the step-cone pulley design optimization results. (**a**) Convergence curve; (**b**) box plot of fitness values.

5.2. Design Issues of Hydrostatic Thrust Bearings

The design optimization aims to minimize power loss in a hydrostatic thrust bearing system through optimal selection of four key design parameters [48,49]:

- Oil viscosity (μ);
- Bearing radius (R);
- Lubricant flow rate (Q);
- Recess radius (R_0).

The mathematical formulation is expressed as follows:
Minimize

$$f(x) = \frac{\left(\frac{QP_0}{0.7} + E_f\right)}{12} \tag{33}$$

subject to

$$g_1(x) = 101000 - W \leq 0, \tag{34}$$

$$g_2(x) = P_0 - 1000 \leq 0, \tag{35}$$

$$g_3(x) = \Delta T - 50 \leq 0, \tag{36}$$

$$g_4(x) = 0.001 - h \leq 0, \tag{37}$$

$$g_5(x) = R_0 - R \le 0, \tag{38}$$

$$g_6(x) = \frac{0.0307}{386.4P_0}\left(\frac{Q}{2\pi Rh}\right) - 0.001 \le 0, \tag{39}$$

$$g_7(x) = \frac{W}{\pi\left(R^2 - R_0^2\right) + 10^{-5}} - 5000 \le 0 \tag{40}$$

where

$$W = \frac{\pi P_0\left(R^2 - R_0^2\right)}{2\ln\left(\frac{R}{R_0}\right) - 10^{-5}}, \tag{41}$$

$$P_0 = \frac{6\mu Q}{\pi h^3}\ln\left(\frac{R}{R_0}\right), \tag{42}$$

$$E_f = 9336Q \times Q\gamma C\Delta T, \tag{43}$$

$$\Delta T = 2\left(10^P - 560\right), \tag{44}$$

$$P = \frac{\log_{10}[\log_{10}(8.122 \times 10^6\mu + 0.8)] - 10.04}{-3.55}, \tag{45}$$

$$h = \left(\frac{2\pi \times N}{60}\right)^2 \frac{2n\mu}{E_f}\left(\frac{R^4}{4} - \frac{R_0^4}{4}\right) - 10^{-5} \tag{46}$$

The constants used in Equations (39) and (43)–(46) are $\gamma = 0.0307$, $g = 386.4$, $C = 0.5$, and $N = 750$. The vector order used by the implementation is $[R, R_0, \mu, Q]$.

Constraints:

$$1 \le R,\ R_0,\ Q \le 16,\ 1 \times 10^{-6} \le \mu \le 16 \times 10^{-6} \tag{47}$$

The results for the hydrostatic thrust bearing design demonstrate IMALA's superior performance in both solution quality and stability. As shown in the parameter Table 8, IMALA achieves the most balanced design configuration with optimal viscosity ($\mu = 5.4693 \times 10^{-6}$), smallest bearing radius ($R = 6.0425$), and moderate flow rate ($Q = 2.4003$), and recess radius ($R_0 = 3.1838$), and rank 1. The performance metrics further confirm IMALA's advantages, yielding the lowest minimum power loss (1.66×10^3) and most stable results (std = 1.64×10^2), outperforming all other algorithms (Table 9). In contrast, CFOA exhibits extreme instability with power losses spanning 19 orders of magnitude (9.50×10^{12} to 1.96×10^{19}) and an average 16 orders of magnitude worse than IMALA (4.99×10^{18} vs. 1.90×10^3). While BKA shows competitive minimum values (1.73×10^3), its higher standard deviation (1.14×10^3) suggests less reliability compared to IMALA's consistently superior and stable performance across all evaluation metrics.

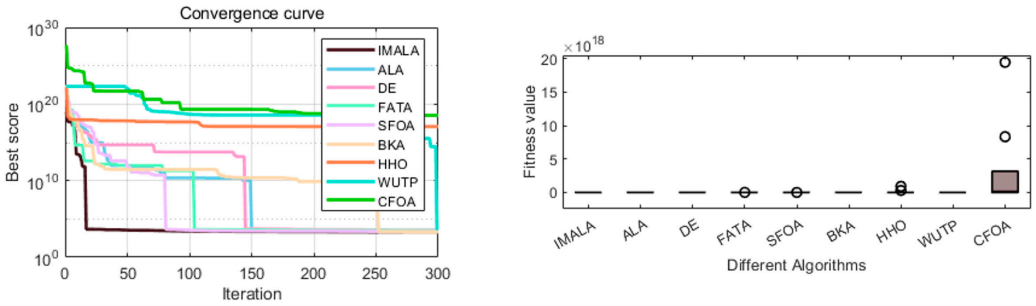
Table 8. Hydrostatic thrust bearing design optimization parameter results.

Variable	IMALA	ALA	DE	FATA	SFOA	BKA	HHO	WUTP	CFOA
R	6.0425	7.0059	6.6805	8.1062	6.1053	5.9890	9.1321	8.0277	9.8451
R_0	5.4836	6.4893	5.6637	7.6326	5.4322	5.4189	8.3472	7.5871	9.9425
μ ($\times 10^{-6}$)	5.4693	5.7445	5.5946	6.8066	6.4008	6.1360	6.1673	8.7714	6.2274
Q	2.4003	3.4680	3.9004	6.7508	3.6219	2.9797	8.2879	15.2212	5.0085
Best F	1660.3513	2119.5396	2555.4685	2924.7371	1941.4997	1731.0582	4202.5987	2973.9659	9.50×10^{12}
rank	1	4	5	6	3	2	8	7	9

Table 9. Hydrostatic thrust bearing design optimization results.

Alg.	IMALA	ALA	DE	FATA	SFOA	BKA	HHO	WUTP	CFOA
min	1.66×10^3	2.12×10^3	2.56×10^3	2.92×10^3	1.94×10^3	1.73×10^3	4.20×10^3	2.97×10^3	9.50×10^{12}
worst	2.25×10^3	1.44×10^9	5.06×10^3	3.52×10^3	2.78×10^3	5.45×10^3	2.24×10^{18}	9.61×10^{18}	1.96×10^{19}
std	1.64×10^2	4.54×10^8	7.83×10^2	2.05×10^2	2.63×10^2	1.14×10^3	7.21×10^{17}	3.04×10^{18}	6.03×10^{18}
avg	1.90×10^3	1.44×10^8	3.75×10^3	3.18×10^3	2.50×10^3	2.76×10^3	2.97×10^{17}	9.61×10^{17}	4.99×10^{18}
median	1.87×10^3	2.64×10^3	3.58×10^3	3.14×10^3	2.59×10^3	2.28×10^3	1.17×10^9	3.19×10^3	4.43×10^{18}

In the convergence curve graph (Figure 12a), the IMALA algorithm still exhibits outstanding convergence performance. In the early iterations, the optimal score decreases rapidly, and it can stabilize at a lower value after a relatively small number of iterations. In contrast, other algorithms in this part have a slower decrease in optimal score during the iteration process, and even after a large number of iterations, they have not achieved a good convergence effect. The box plot test graph (Figure 12b) shows that the fitness values of the IMALA algorithm are concentrated in the lower area, with few outliers, reflecting its stable and reliable results; the CFOA algorithm not only has a box position on the right side, but also has large outliers, indicating that its performance in fitness value is unstable, and the algorithm effect fluctuates greatly.



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

Figure 12. Convergence curve and box plot test graph of the hydrostatic thrust bearings optimization results. (**a**) The Convergence curve; (**b**) box plot of fitness values.

5.3. Piston Lever Design Issues

The main purpose of this problem is to locate the piston components—Piston height ($H = x_1$), Piston diameter ($B = x_2$), Connecting rod diameter ($D = x_3$), and Eccentricity ($X = x_4$)—by minimizing the amount of oil when the piston lever rises from the mathematical equation to 45°. The engineering optimization aims to determine four key geometric parameters of the piston system and minimize the oil volume Voil required when the piston lever rotates from initial position to 45° [50]:

$$\min f(X) = \frac{1}{4}\pi x_3^2(L_2 - L_1) \tag{48}$$

where oil volume calculation must consider the coupling effect between cylinder volume change and connecting rod trajectory.

According to mechanical design requirements, the following physical constraints must be satisfied:

Inequality constraints:

$$g_1(X) = QL\cos\theta - R \times F \le 0, \tag{49}$$

$$g_2(X) = Q(L - x_4) - M_{\max} \le 0, \tag{50}$$

$$g_3(X) = 1.2(L_2 - L_1) - L_1 \le 0, \tag{51}$$

$$g_4(X)\& = \frac{x_3}{2} - x_2 \le 0 \tag{52}$$

$$R = \frac{|-x_4(x_4\sin\theta + x_1) + x_1(x_2 - x_4\cos\theta)|}{\sqrt{(x_4 - x_2)^2 + x_1^2}}, \tag{53}$$

$$F = \frac{\pi P x_3^2}{4}, \tag{54}$$

$$L_1 = \sqrt{(x_4 - x_2)^2 + x_1^2}, \tag{55}$$

$$L_2 = \sqrt{(x_4 \sin\theta + x_1)^2 + (x_2 - x_4 \cos\theta)^2} \tag{56}$$

with the following constant parameters,

$$\theta = 45°, \tag{57}$$

$$Q = 10,000\,\text{lbs}, \tag{58}$$

$$L = 240\,\text{in}, \tag{59}$$

$$M_{\max} = 1.8 \times 10^6\,\text{lb} \cdot \text{in}, \tag{60}$$

$$P = 1500\,\text{psi} \tag{61}$$

and variable bounds:

$$0.05 \le x_1,\ x_2,\ x_4 \le 500,\ 0.05\& \le x_3 \le 120 \tag{62}$$

The two tables present the optimization results and performance metrics for piston lever design. Table 10 shows that IMALA, DE, and WUTP achieve identical optimal solutions (Voil = 1.0574), whereas CFOA exhibits severe deviations (Voil = 361.1270), indicating convergence failure. Table 11 compares the optimization performance of nine algorithms on the piston lever design problem. IMALA, DE, and WUTP achieve nearly identical objective values of 1.057, with extremely small standard deviations, indicating high accuracy and repeatability. SFOA also exhibits stable performance, with an average value of 1.058. Although ALA obtains a similar minimum and median value, its higher average (17.70) and standard deviation (52.63) reveal considerable fluctuations across runs. Similarly, FATA and BKA exhibit substantial differences between their mean and median values, suggesting occasional poor solutions. HHO and CFOA show comparatively weaker performance, with CFOA recording the highest average value (741.2). Overall, IMALA demonstrates reliable optimization performance and improved stability over ALA, while achieving solution quality comparable to DE and WUTP.

Table 10. Piston lever design optimization parameter results.

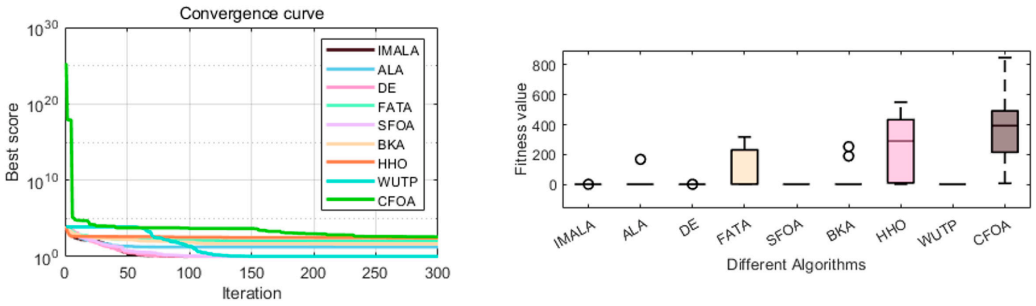
Variable	IMALA	ALA	DE	FATA	SFOA	BKA	HHO	WUTP	CFOA
x_1 (H)	0.05	0.05	0.05	0.0541	0.05	0.05	0.05	0.05	278.7879
x_2 (B)	1.0081	1.0081	1.0081	1.0530	1.0081	1.0082	1.3442	1.0081	219.4284
x_3 (D)	2.0162	2.0162	2.0162	2.0421	2.0162	2.0164	2.6353	2.0162	1.3583
x_4 (X)	500	500	500	500	500	500	291.8567	500	487.5250
Voil	1.0574	1.0574	1.0574	1.1375	1.0574	1.0576	2.3495	1.0574	361.1270
Rank	1	3	1	5	2	4	6	1	7

Table 11. Piston lever design optimization results.

Alg.	IMALA	ALA	DE	FATA	SFOA	BKA	HHO	WUTP	CFOA
min	1.057×10^{0}	1.057×10^{0}	1.057×10^{0}	1.138×10^{0}	1.057×10^{0}	1.058×10^{0}	2.349×10^{0}	1.057×10^{0}	3.611×10^{2}
worst	1.057×10^{0}	1.675×10^{2}	1.057×10^{0}	3.181×10^{2}	1.058×10^{0}	2.538×10^{2}	5.507×10^{2}	1.057×10^{0}	1.169×10^{3}
std	3.934×10^{-9}	5.263×10^{1}	4.307×10^{-9}	1.380×10^{2}	1.135×10^{-4}	9.421×10^{1}	1.999×10^{2}	1.734×10^{-7}	3.129×10^{2}
avg	1.057×10^{0}	1.770×10^{1}	1.057×10^{0}	1.066×10^{2}	1.058×10^{0}	4.516×10^{1}	2.579×10^{2}	1.057×10^{0}	7.412×10^{2}
median	1.057×10^{0}	1.057×10^{0}	1.057×10^{0}	2.395×10^{0}	1.058×10^{0}	1.059×10^{0}	2.909×10^{2}	1.057×10^{0}	7.952×10^{2}

The figure (Figure 13) presents a comparative analysis of convergence curves for various optimization algorithms in solving the piston lever design problem. The curve trends demonstrate that IMALA significantly outperforms other algorithms, with its fitness value rapidly decreasing to near zero, indicating superior global search capability and convergence

precision in Figure 13a. In contrast, CFOA consistently maintains high fitness values, suggesting susceptibility to local optima or convergence failure. Intermediate algorithms like ALA and DE show moderate performance but fail to match IMALA's optimization efficacy. Collectively, the figure visually confirms IMALA's dominance for this problem, while exposing the need for algorithmic improvements in CFOA and similar methods to prevent premature convergence or inefficient search behavior.



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

Figure 13. Convergence curve and box plot of the piston lever design optimization results. (**a**) Convergence curve; (**b**) box plot of fitness values.

The three engineering design problems provide a practical basis for assessing the performance of IMALA in constrained optimization. Feasibility verification based on the recorded design vectors indicates that the best solutions obtained by IMALA satisfy all inequality constraints, as well as the prescribed equality-constraint tolerance for the step-cone pulley problem. In contrast, some competing solutions exhibit constraint violations, highlighting the importance of interpreting penalized fitness values alongside constraint residuals. For the piston lever problem, IMALA and several competing algorithms achieve comparable best fitness values within the reported numerical precision, suggesting similar solution quality for this particular application. Furthermore, the convergence curves and fitness distributions provide additional insights into optimization behavior and run-to-run variability. Overall, these findings demonstrate the applicability of IMALA to the three investigated engineering problems, while indicating that its relative advantages remain dependent on the problem characteristics and experimental conditions.

6. Conclusions and Prospects

This paper elaborates in detail on the IMALA and verifies its outstanding performance in optimization problems through a series of experiments. IMALA performs exceptionally well when solving complex optimization tasks, and it has significant advantages over the traditional ALA and seven other comparison algorithms (DE, FATA, SFOA, BKA, HHO, WUTP, CFOA).

In terms of algorithmic performance, IMALA enhances its optimization capability by integrating three strategies: cubic chaotic initialization, differential mutation, and beta-distribution-based random opposition learning. Experimental results on the CEC2017 and CEC2022 benchmark suites validate that IMALA delivers competitive performance in convergence accuracy, convergence behavior, and stability across four categories of test functions, namely unimodal, multimodal, hybrid, and composition functions, with corresponding average ranks of 1.69 and 1.58, respectively. Ablation experiments further demonstrate that the full IMALA achieves the optimal average rank of 1.55, while the original ALA records an average rank of 6.45. This finding suggests that the overall performance superiority of the algorithm stems from the synergistic effects of the three improvement strategies. Specifically, cubic chaotic initialization is devised to boost the

diversity of the initial population; differential mutation facilitates further refinement of solutions by introducing additional search variations; beta-distribution-based opposition learning fosters exploration of underexplored regions in the search space via generating new candidate solutions. Nevertheless, performance discrepancies across individual strategies and their diverse combinations indicate that the contribution of each strategy is problem-dependent and varies with specific optimization tasks. The Wilcoxon rank-sum test provides statistical evidence of performance differences in certain comparisons, while the convergence curves and box plots further illustrate the optimization behavior and variability of the results.

In terms of engineering applications, IMALA has been successfully applied to three practical engineering optimization problems: the design of step-cone pulleys, the design of hydrostatic thrust bearings, and the design of piston lever issues. In the design of a step-cone pulley, IMALA achieved the most compact design, resulting in the minimum weight and the best performance indicators. Compared with other algorithms, its stability and optimization effect are extremely outstanding. In the optimization of the design of hydrostatic thrust bearings, IMALA also demonstrated excellent performance. Not only did it achieve the lowest power loss, but the result stability was far superior to that of other algorithms, such as CFOA, which showed significant instability in this problem. In the design of the piston lever problem, IMALA has the same ranking as DE and WUTP and is superior to other algorithms.

Despite these promising results, IMALA still has several limitations that warrant further investigation.

- Generalisability of the Results: Although IMALA demonstrates competitive performance on CEC2017, CEC2022, and three engineering optimization problems, its advantages may vary across different problem types, dimensions, and experimental settings. These results do not establish universal superiority, and further validation on broader benchmarks and practical applications is needed to assess its generalizability.
- Computational Complexity: The pairwise-distance calculations introduced by the Beta-distribution-based opposition learning strategy result in an additional $O(N^2D)$ computational cost per iteration. This overhead may affect the computational efficiency of IMALA, particularly for larger populations and higher-dimensional optimization problems. Reducing this computational burden will be considered in future work.
- Parameter Sensitivity: The performance of IMALA relies on parameters within its new components (e.g., the mutation strategy and the Beta distribution). Although robust to a degree, the lack of a fully adaptive parameter-tuning mechanism may limit its robustness across an even wider range of problem types.
- Theoretical Foundation: Like many metaheuristics, a rigorous theoretical convergence analysis of IMALA remains an open challenge. Providing solid mathematical guarantees for its convergence would strengthen its theoretical foundation.

To address these limitations, the following directions are proposed for future work:

- Algorithmic Efficiency: Future research will focus on developing a more lightweight variant of IMALA, potentially by simplifying strategies or employing surrogate models, to reduce computational cost without compromising performance.
- Adaptive Mechanism: Integrating an adaptive or fuzzy logic-based system to dynamically control the algorithm's parameters based on the search state is a promising path to enhance its robustness and self-adaptation.
- Theoretical Analysis: A more in-depth theoretical study on the convergence behavior of IMALA will be conducted to establish a solid mathematical foundation.

- Application Expansion: Applying IMALA to more complex and domain-specific problems, such as large-scale neural network training, complex scheduling, and aerospace engineering design, will further test and validate its practicality.
- Hybridization: Exploring the hybridization of IMALA with other optimization paradigms, such as local search techniques or quantum-inspired computing, could lead to more powerful solvers for challenging optimization scenarios.
- Multi-objective Optimization: Although IMALA currently focuses on single-objective optimization, it has potential for multi-objective applications. Future work will incorporate Pareto dominance, non-dominated sorting, and diversity preservation mechanisms to address conflicting objectives in engineering optimization.

In terms of application expansion, IMALA can be attempted to be applied in more complex engineering fields, such as aircraft design optimization in aerospace, processing parameter optimization for complex components in mechanical manufacturing, and grid planning and scheduling optimization in power systems. At the same time, it is possible to explore the mixed use of IMALA with other algorithms to leverage the advantages of different algorithms and solve some more challenging large-scale, high-dimensional, and multi-constraint practical optimization problems.

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

Funding: This research was funded by 2026 High-Level Talents Research Start-up Fund Project of West Anhui University: WGKQ2026008; High-Level Talents Research Start-up Fund Project of West Anhui University: WGKQ2024011

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

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

Abbreviations

The following abbreviations are used in this manuscript:

ALA	Artificial Lemming Algorithm
IMALA	Improved Artificial Lemming Algorithm
CEC	Congress on Evolutionary Computation
DE	Differential Evolution
FATA	Fata Morgana Algorithm
SFOA	Starfish optimization algorithm
BKA	Black-Winged Kite Algorithm
HHO	Harris Hawks Optimization
WUTP	Water Uptake and Transport in Plants
CFOA	Catch fish optimization algorithm
OBL	Opposition-Based Learning

Appendix A. Objective Function Evaluation Accounting

Table A1. Implementation-derived objective function evaluation accounting (N = 30; 300 full update cycles).

Algorithm	FEs per Complete Cycle	FEs for 300 Cycles, Including Initialization
IMALA	$3N = 90$	$N + 3NT = 27{,}030$
ALA	$N = 30$	$N + NT = 9030$
DE	$N = 30$	$N + NT = 9030$
FATA	$N = 30$	$NT = 9000$
SFOA	$N = 30$	$N + NT = 9030$
BKA	$2N = 60$	$N + 2NT = 18{,}030$
HHO	$N + 2a_t + 2b_t$ (30–150)	$9000 + 2\sum(a_t + b_t)$; range 9000–45,000
WUTP	$N = 30$	$N + NT = 9030$
CFOA	$N = 30$	$NT = 9000$

Note: For HHO, at denotes the number of hawks entering the rapid-dive comparison in iteration t, and b_t $(0 \le b_t \le a_t \le N)$ denotes those requiring the second comparison; each comparison uses two objective function calls.

Appendix B. CEC2017 Runtime Comparison

Table A2. Recorded mean execution time of IMALA and ALA on the 29 CEC2017 benchmark functions (100 dimensions).

Function	IMALA	ALA
F_1	0.022387	0.004282
F_3	0.018560	0.003294
F_4	0.024184	0.004459
F_5	0.029134	0.006266
F_6	0.040359	0.009854
F_7	0.029267	0.005939
F_8	0.028354	0.005730
F_9	0.029066	0.006202
F_{10}	0.030487	0.007488
F_{11}	0.029360	0.005067
F_{12}	0.026995	0.005695
F_{13}	0.026746	0.005962
F_{14}	0.033129	0.007226
F_{15}	0.027815	0.005724
F_{16}	0.030093	0.005611
F_{17}	0.036094	0.007108
F_{18}	0.026929	0.005865
F_{19}	0.087804	0.025911
F_{20}	0.039207	0.008070
F_{21}	0.047284	0.013220
F_{22}	0.041755	0.013735
F_{23}	0.058436	0.016263
F_{24}	0.065882	0.018163
F_{25}	0.059087	0.015753
F_{26}	0.055120	0.019932
F_{27}	0.060813	0.018076
F_{28}	0.053663	0.015431
F_{29}	0.050420	0.013779
F_{30}	0.078781	0.029744
Mean (29)	0.040938	0.010684

Appendix C. Constraint Verification for the Engineering Designs

Tables A3–A5 report the saved positions associated with the recorded best fitness and their residuals. The displayed design variables in Tables 6, 8 and 10 are rounded; residuals below were calculated before rounding. Vector orders are $[d_1, d_2, d_3, d_4, w]$ in mm, $[R, R_0, \mu, Q]$, and $[H, B, D, X]$, respectively; "No" denotes violation under $g_i \leq 0$ and $|h_j| \leq 10^{-4}$; "Boundary" identifies the piston DE residual of $+1.51 \times 10^{-14}$.

Table A3. Step-cone pulley objective, penalized fitness and constraint residuals.

Algorithm	f/F	Residuals in Index Order	Status
IMALA	16.3779/16.3899	x: 38.9067695224851, 53.5372661379598, 71.3773922826382, 85.5795048559005, 89.3019317111791 g_1–g_8: −0.9873, −0.997483, −1.00999, −1.02002, −711.47, −491.676, −220.765, −2.80012 h_1–h_3: 2.01172×10^{-9}, 5.61197×10^{-10}, 2.77965×10^{-9}	Yes
ALA	17.0199/24.0682	x: 39.9813241167195, 55.0170279244528, 73.3501198970931, 87.9426403375081, 87.8798070815787 g_1–g_8: −0.986872, −0.997335, −1.01019, −1.02049, −725.679, −503.499, −229.585, −9.16706 h_1–h_3: 6.03007×10^{-8}, -1.4833×10^{-8}, 5.64986×10^{-8}	Yes
DE	17.1418/2630.27	x: 40.9168552005924, 56.3055833684541, 75.067112034385, 90, 84.5071265126515 g_1–g_8: −0.986499, −0.997206, −1.01036, −1.02091, −705.198, −486.625, −217.089, −0.177132 h_1–h_3: -5.77531×10^{-7}, 1.50264×10^{-6}, 1.47159×10^{-7}	Yes
FATA	$19.6292/9.35185 \times 10^{10}$	x: 42.5359342548996, 60, 78.5275412384603, 90, 90 g_1–g_8: −0.985854, −0.996837, −1.01071, −1.02091, −840.523, −627.621, −305.716, −36.5409 h_1–h_3: −0.00526015, −0.00131258, 0.0080079	No
SFOA	18.2155/30.6692	x: 40.8713982658901, 56.2428467483687, 74.9842139495902, 89.9000729306394, 90 g_1–g_8: −0.986517, −0.997212, −1.01035, −1.02089, −785.895, −553.368, −266.637, −35.8774 h_1–h_3: -9.63977×10^{-8}, -1.23413×10^{-9}, 5.62113×10^{-8}	Yes
BKA	17.0765/17.0765	x: 39.6664315222386, 54.5833971301874, 72.7720109796616, 87.2501564220857, 89.577866551537 g_1–g_8: −0.986997, −0.997378, −1.01013, −1.02035, −740.219, −515.504, −238.482, −15.5751 h_1–h_3: -3.66107×10^{-12}, -9.82068×10^{-11}, -2.78453×10^{-11}	Yes
HHO	17.8968/340,952	x: 40.6473121717466, 55.9379676846296, 74.5681230523054, 89.4073534395328, 89.4032029124035 g_1–g_8: −0.986606, −0.997243, −1.01031, −1.02079, −769.669, −540.004, −256.602, −28.6811 h_1–h_3: -1.34824×10^{-5}, 1.26151×10^{-5}, -1.33712×10^{-7}	Yes
WUTP	17.1573/17.175	x: 40.9168229835095, 56.305377687622, 75.067609492298, 89.9999954834655, 84.5835470220451 g_1–g_8: −0.986499, −0.997206, −1.01036, −1.02091, −706.34, −487.567, −217.796, −0.683021 h_1–h_3: 1.765×10^{-9}, 3.24451×10^{-9}, -2.01993×10^{-9}	Yes

Table A3. *Cont.*

Algorithm	f/F	Residuals in Index Order	Status
CFOA	18.2927/9.32914 × 10^9	x: 40.9060314690263, 57.110735802653, 74.7480018769171, 89.9881309197322, 89.7743543918005 g_1–g_8: −0.986503, −0.997126, −1.01033, −1.02091, −783.657, −567.688, −261.968, −34.9685 h_1–h_3: −0.00294554, 0.000807595, −2.66914 × 10^{-5}	No

Table A4. Hydrostatic thrust bearing objective, penalized fitness and constraint residuals.

Algorithm	f/F	Residuals in Index Order	Status
IMALA	1660.35/1660.35	x: 6.04245836327956, 5.4835962187671, 5.46930381385806 × 10^{-6}, 2.40025716605941 g_1–g_7: −59.4221, −30.7819, −1.73829, −0.000359111, −0.558862, −0.000996187, −6.08256	Yes
ALA	2119.54/2119.54	x: 7.00591926841556, 6.48931856694423, 5.74449800991556 × 10^{-6}, 3.46803261131186 g_1–g_7: −74.4935, −293.119, −5.87011, −0.0006035, −0.516601, −0.000994478, −385.167	Yes
DE	2555.47/2555.47	x: 6.68053177416287, 5.66367113380034, 5.59457910463824 × 10^{-6}, 3.90040167174101 g_1–g_7: −995.216, −145.872, −3.65223, −0.00100473, −1.01686, −0.000995688, −2413.54	Yes
FATA	2924.74/2924.74	x: 8.1062244319038, 7.63256778307103, 6.80662937982064 × 10^{-6}, 6.75075962978612 g_1–g_7: −15,246.6, −402.406, −19.6789, −0.00106781, −0.473657, −0.000991478, −36.4198	Yes
SFOA	1941.5/1941.5	x: 6.10534921843843, 5.43220610615318, 6.40083178395053 × 10^{-6}, 3.62189047469209 g_1–g_7: −39.5801, −32.5459, −14.7578, −0.000748584, −0.673143, −0.000995566, −858.857	Yes
BKA	1731.06/1731.06	x: 5.98904743164472, 5.41893719774633, 6.13600248323455 × 10^{-6}, 2.97966128729206 g_1–g_7: −1138.44, −9.86697 × 10^{-8}, −11.3215, −0.000517278, −0.57011, −0.000995854, −1.13548	Yes
HHO	4202.6/4202.6	x: 9.13214469309109, 8.34722919612753, 6.1673074676281 × 10^{-6}, 8.28791575159264 g_1–g_7: −5.13683 × 10^{-9}, −578.863, −11.7377, −0.00175156, −0.784915, −0.000990096, −2656.73	Yes
WUTP	2973.97/2973.97	x: 8.0277165602737, 7.58710010372005, 8.77140227719016 × 10^{-6}, 15.2211970608451 g_1–g_7: −3148.11, −456.091, −39.0614, −0.00198003, −0.440616, −0.000985208, −181.591	Yes
CFOA	2797.1/9.49936 × 10^{12}	x: 9.84508163976728, 9.94252859906439, 6.22744052690514 × 10^{-6}, 5.00854008569189 g_1–g_7: −185,396, −67.7458, −12.5295, 0.00185697, 0.097447, −0.00100805, −52,277.7	No

Table A5. Piston lever objective, penalized fitness and constraint residuals.

Algorithm	f/F	Residuals in Index Order	Status
IMALA	1.05739/1.05739	x: 0.0500000000000005, 1.00811351737858, 2.01622703472348, 499.999999999505 g_1–g_4: -1.13447×10^{-5}, -4.4×10^{6}, −498.594, -1.68401×10^{-11}	Yes
ALA	1.05739/1.05739	x: 0.0500000000179372, 1.0081135174639, 2.0162270349277, 500 g_1–g_4: −0.000357242, -4.4×10^{6}, −498.594, -4.996×10^{-14}	Yes
DE	1.05739/1.05739	x: 0.05, 1.00811351735789, 2.01622703471581, 500 g_1–g_4: -3.93484×10^{-8}, -4.4×10^{6}, −498.594, 1.5099×10^{-14}	Boundary
FATA	1.13751/1.13751	x: 0.0541369009086048, 1.05295919757773, 2.04212472740747, 500 g_1–g_4: −44,067.3, -4.4×10^{6}, −498.53, −0.0318968	Yes
SFOA	1.05741/1.05741	x: 0.05, 1.00812108361309, 2.01623493533075, 500 g_1–g_4: −13.3256, -4.4×10^{6}, −498.594, -3.61595×10^{-6}	Yes
BKA	1.05759/1.05759	x: 0.05, 1.00817830948713, 2.01635660078346, 500 g_1–g_4: −218.339, -4.4×10^{6}, −498.594, -9.0954×10^{-9}	Yes
HHO	2.34946/2.34946	x: 0.05, 1.34419579110811, 2.63534960349866, 291.856719291219 g_1–g_4: -8.9407×10^{-8}, -2.31857×10^{6}, −289.996, −0.026521	Yes
WUTP	1.05739/1.05739	x: 0.05, 1.00811355674756, 2.01622706430821, 500 g_1–g_4: −0.0499498, -4.4×10^{6}, −498.594, -2.45935×10^{-8}	Yes
CFOA	361.127/361.127	x: 278.787904092891, 219.428411965036, 1.35832996976767, 487.525007028783 g_1–g_4: −207,609, -4.27525×10^{6}, −87.7318, −218.749	Yes

References

1. Pereira, J.L.J.; Oliver, G.A.; Francisco, M.B.; Cunha, J.S.S.; Gomes, G.F. A review of multi-objective optimization: Methods and algorithms in mechanical engineering problems. *Arch. Comput. Methods Eng.* **2022**, *29*, 2285–2308. [CrossRef]
2. Tang, J.; Liu, G.; Pan, Q. A review on representative swarm intelligence algorithms for solving optimization problems: Applications and trends. *IEEE/CAA J. Autom. Sin.* **2021**, *8*, 1627–1643. [CrossRef]
3. Modu, B.; Abdullah, M.P.; Bukar, A.L.; Hamza, M.F. A systematic review of hybrid renewable energy systems with hydrogen storage: Sizing, optimization, and energy management strategy. *Int. J. Hydrogen Energy* **2023**, *48*, 38354–38373. [CrossRef]
4. Chen, Y.; Chi, Y. Harnessing structures in big data via guaranteed low-rank matrix estimation: Recent theory and fast algorithms via convex and nonconvex optimization. *IEEE Signal Proc. Mag.* **2018**, *35*, 14–31. [CrossRef]
5. Abd Elaziz, M.; Dahou, A.; Abualigah, L.; Yu, L.; Alshinwan, M.; Khasawneh, A.M.; Lu, S. Advanced metaheuristic optimization techniques in applications of deep neural networks: A review. *Neural Comput. Appl.* **2021**, *33*, 14079–14099. [CrossRef]
6. Guo, Z.; Liu, G.; Jiang, F. Chinese Pangolin Optimizer: A novel bio-inspired metaheuristic for solving optimization problems. *J. Supercomput.* **2025**, *81*, 517. [CrossRef]
7. Houssein, E.H.; Saeed, M.K.; Hu, G.; Al-Sayed, M.M. Metaheuristics for solving global and engineering optimization problems: Review, applications, open issues and challenges. *Arch. Comput. Methods Eng.* **2024**, *31*, 4485–4519. [CrossRef]
8. Selvarajan, S. A comprehensive study on modern optimization techniques for engineering applications. *Artif. Intell. Rev.* **2024**, *57*, 194. [CrossRef]
9. Li, C.; Wang, Z.; Yu, J.; Abdel-Salam, M.; Houssein, E.H.; Zhong, R. Multi-strategies improved coati optimization algorithm and performance analysis. *Knowl. Inf. Syst.* **2025**, *67*, 11095–11159. [CrossRef]
10. Elhosseny, M.; Abdel-Salam, M.; El-Hasnony, I.M. Adaptive dynamic crayfish algorithm with multi-enhanced strategy for global high-dimensional optimization and real-engineering problems. *Sci. Rep.* **2025**, *15*, 10656. [CrossRef] [PubMed]

11. Wang, W.C.; Xu, L.; Chau, K.W.; Liu, C.J.; Ma, Q.; Xu, D.M. Cε-LDE: A lightweight variant of differential evolution algorithm with combined ε constrained method and L'evy flight for constrained optimization problems. *Expert. Syst. Appl.* **2023**, *211*, 118644. [CrossRef]
12. Abdel-Salam, M.; Oliva, D.; P'erez-Cisneros, M.; El-Hasnony, I.M. Advanced feature selection approach with Halton-based enhanced adaptive dung beetle algorithm. *Clust. Comput.* **2025**, *28*, 595. [CrossRef]
13. Wang, J.; Wang, W.C.; Chau, K.W.; Qiu, L.; Hu, X.X.; Zang, H.F.; Xu, D.-M. An Improved Golden Jackal Optimization Algorithm Based on Multi-strategy Mixing for Solving Engineering Optimization Problems. *J. Bionic Eng.* **2024**, *21*, 1092–1115. [CrossRef]
14. Rana, N.; Latiff, M.S.A.; Abdulhamid, S.M.; Chiroma, H. Whale optimization algorithm: A systematic review of contemporary applications, modifications and developments. *Neural Comput. Appl.* **2020**, *32*, 16245–16277. [CrossRef]
15. Jia, C.; He, L.; Liu, D.; Fu, S. Path planning and engineering problems of 3D UAV based on adaptive coati optimization algorithm. *Sci. Rep.* **2024**, *14*, 30717. [CrossRef] [PubMed]
16. Meraihi, Y.; Ramdane-Cherif, A.; Acheli, D.; Mahseur, M. Dragonfly algorithm: A comprehensive review and applications. *Neural Comput. Appl.* **2020**, *32*, 16625–16646. [CrossRef]
17. Heidari, A.A.; Mirjalili, S.; Faris, H.; Aljarah, I.; Mafarja, M.; Chen, H. Harris hawks optimization: Algorithm and applications. *Future Gener. Comp. Sy* **2019**, *97*, 849–872. [CrossRef]
18. Abualigah, L. Multi-verse optimizer algorithm: A comprehensive survey of its results, variants, and applications. *Neural Comput. Appl.* **2020**, *32*, 12381–12401. [CrossRef]
19. Faramarzi, A.; Heidarinejad, M.; Stephens, B.; Mirjalili, S. Equilibrium optimizer: A novel optimization algorithm. *Knowl.-Based Syst.* **2020**, *191*, 105190. [CrossRef]
20. Li, S.; Chen, H.; Wang, M.; Heidari, A.A.; Mirjalili, S. Slime mould algorithm: A new method for stochastic optimization. *Future Gener. Comp. Sy* **2020**, *111*, 300–323. [CrossRef]
21. Conteh, I.M.; Salawudeen, A.T.; Abdulsalami, A.O.; Du, Q. An Improved Triangular Mutated Slime Mould Algorithm for Developing an Optimized Frequency Control of a Bi-zonal Islanded Microgrid Power System. *Results Eng.* **2025**, *25*, 104306. [CrossRef]
22. Turgut, O.E.; Turgut, M.S.; Kırtepe, E. A systematic review of the emerging metaheuristic algorithms on solving complex optimization problems. *Neural Comput. Appl.* **2023**, *35*, 14275–14378. [CrossRef]
23. Hamad, Q.S.; Saleh, S.A.M.; Suandi, S.A.; Samma, H.; Hamad, Y.S.; Hussien, A.G. A Review of Enhancing Sine Cosine Algorithm: Common Approaches for Improved Metaheuristic Algorithms. *Arch. Comput. Methods Eng.* **2024**, *32*, 2549–2606. [CrossRef]
24. Xiao, Y.; Guo, Y.; Cui, H.; Wang, Y.; Li, J.; Zhang, Y. IHAOAVOA: An improved hybrid aquila optimizer and African vulture's optimization algorithm for global optimization problems. *Math. Biosci. Eng.* **2022**, *19*, 10963–11017. [CrossRef] [PubMed]
25. Xiao, Y.; Cui, H.; Hussien, A.G.; Hashim, F.A. MSAO: A multi-strategy boosted snow ablation optimizer for global optimization and real-world engineering applications. *Adv. Eng. Inform.* **2024**, *61*, 102464. [CrossRef]
26. Liu, Z.; Xu, S.; Zhang, B.; Guo, S. Voltage tracking and regulation of vehicle PEMFC system under low load condition based on fuzzy LQG hybrid strategy. *ISA Trans.* **2025**, *165*, 510–523. [CrossRef] [PubMed]
27. Xiao, Y.; Cui, H.; Khurma, R.A.; Castillo, P.A. Artificial lemming algorithm: A novel bionic meta-heuristic technique for solving real-world engineering optimization problems. *Artif. Intell. Rev.* **2025**, *58*, 84. [CrossRef]
28. Xie, Y.; Sun, Z.; Yuan, K.; Sun, Z. 3D UAV Route Optimization in Complex Environments Using an Enhanced Artificial Lemming Algorithm. *Symmetry* **2025**, *17*, 946. [CrossRef]
29. Tan, Y.; Wang, J.; Wang, B. An Improved Bionic Artificial Lemming Algorithm for Global Optimization and Cloud Task-Scheduling Problems. *Biomimetics* **2025**, *10*, 572. [CrossRef] [PubMed]
30. Yang, J.; Liu, Z.; Zhang, X.; Hu, G. Elite chaotic manta ray algorithm integrated with chaotic initialization and opposition-based learning. *Mathematics* **2022**, *10*, 2960. [CrossRef]
31. Mohamed, A.W.; Hadi, A.A.; Mohamed, A.K. Differential evolution mutations: Taxonomy, comparison and convergence analysis. *IEEE Access* **2021**, *9*, 68629–68662. [CrossRef]
32. Choi, T.J. A rotationally invariant stochastic opposition-based learning using a beta distribution in differential evolution. *Expert. Syst. Appl.* **2023**, *231*, 120658. [CrossRef]
33. Qaraad, M.; Rahrmann, E.; Guinovart, D. Optimizing miRNA-Based Breast Cancer Subtyping with AHALA: A Multi-Stage Classification Approach. *bioRxiv* **2025**. [CrossRef]
34. Fakhouri, H.N.; Hamad, F.; Ishtaiwi, A.; Hudaib, A.; Halalsheh, N.; Fakhouri, S.N. Advancing Engineering Solutions with Protozoa-Based Differential Evolution: A Hybrid Optimization Approach. *Automation* **2025**, *6*, 13. [CrossRef]
35. Lu, Z.; You, Z.; Xia, B. Time optimal trajectory planning of robotic arm based on improved sand cat swarm optimization algorithm. *Appl. Intell.* **2025**, *55*, 323. [CrossRef]
36. Wang, Y.; Zhang, X.; Yu, D.J.; Bai, Y.J.; Du, J.P.; Tian, Z.T. Tent Chaotic Map and Population Classification Evolution Strategy-Based Dragonfly Algorithm for Global Optimization. *Math. Probl. Eng.* **2022**, *2022*, 2508414. [CrossRef]

37. Sun, L.; Liang, H.; Ding, W.; Xu, J.; Chang, B. CMEFS: Chaotic mapping-based mayfly optimization with fuzzy entropy for feature selection. *Appl. Intell.* **2024**, *54*, 7397–7417. [CrossRef]
38. Li, J.; Gao, L.; Li, X. Multi-operator opposition-based learning with the neighborhood structure for numerical optimization problems and its applications. *Swarm Evol. Comput.* **2024**, *84*, 101457. [CrossRef]
39. Zhu, X.; Zhang, J.; Jia, C.; Liu, Y.; Fu, M. A Hybrid Black-Winged Kite Algorithm with PSO and Differential Mutation for Superior Global Optimization and Engineering Applications. *Biomimetics* **2025**, *10*, 236. [CrossRef] [PubMed]
40. Ahmad, M.F.; Isa, N.A.M.; Lim, W.H.; Ang, K.M. Differential evolution: A recent review based on state-of-the-art works. *Alex. Eng. J.* **2022**, *61*, 3831–3872. [CrossRef]
41. Qi, A.; Zhao, D.; Heidari, A.A.; Liu, L.; Chen, Y.; Chen, H. FATA: An efficient optimization method based on geophysics. *Neurocomputing* **2024**, *607*, 128289. [CrossRef]
42. Zhong, C.; Li, G.; Meng, Z.; Li, H.; Yildiz, A.R.; Mirjalili, S. Starfish optimization algorithm (SFOA): A bio-inspired metaheuristic algorithm for global optimization compared with 100 optimizers. *Neural Comput. Appl.* **2025**, *37*, 3641–3683. [CrossRef]
43. Wang, J.; Wang, W.C.; Hu, X.X.; Qiu, L.; Zang, H.F. Black-winged kite algorithm: A nature-inspired meta-heuristic for solving benchmark functions and engineering problems. *Artif. Intell. Rev.* **2024**, *57*, 98. [CrossRef]
44. Shehab, M.; Mashal, I.; Momani, Z.; Shambour, M.K.Y.; AL-Badareen, A.; Al-Dabet, S.; Bataina, N.; Alsoud, A.R.; Abualigah, L. Harris hawk's optimization algorithm: Variants and applications. *Arch. Comput. Methods Eng.* **2022**, *29*, 5579–5603. [CrossRef]
45. Braik, M.; Al-Hiary, H. A novel meta-heuristic optimization algorithm inspired by water uptake and transport in plants. *Neural Comput. Appl.* **2025**, *37*, 13643–13724. [CrossRef]
46. El-Kenawy, E.S.M.; Alhussan, A.A.; Khafaga, D.S.; Alharbi, A.H.; Alzakari, S.A.; Abdelhamid, A.A.; Ibrahim, A.; Eid, M.M. Comment Feedback Optimization Algorithm (CFOA): A Feedback-Driven Framework for Robust and Adaptive Optimization. *J. Intell. Syst. Internet Things* **2025**, *16*, 123–141. [CrossRef]
47. Wang, R.B.; Hu, R.B.; Geng, F.D.; Xu, L.; Chu, S.C.; Pan, J.S.; Meng, Z.-Y.; Mirjalili, S. The Animated Oat Optimization Algorithm: A nature-inspired metaheuristic for engineering optimization and a case study on Wireless Sensor Networks. *Knowl.-Based Syst.* **2025**, *318*, 113589. [CrossRef]
48. Liu, J.; Li, W.; Li, Y. LWMEO: An efficient equilibrium optimizer for complex functions and engineering design problems. *Expert. Syst. Appl.* **2022**, *198*, 116828. [CrossRef]
49. Zhang, Z.; Gao, Y.; Liu, Y.; Zuo, W. A hybrid biogeography-based optimization algorithm to solve high-dimensional optimization problems and real-world engineering problems. *Appl. Soft Comput.* **2023**, *144*, 110514. [CrossRef]
50. Bayzidi, H.; Talatahari, S.; Saraee, M.; Lamarche, C.P. Social network search for solving engineering optimization problems. *Comput. Intell. Neurosci.* **2021**, *2021*, 8548639. [CrossRef] [PubMed]

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.