Electronic Research Archive

https://www.aimspress.com/journal/era

ERA, 33(9): 5865–5896.
DOI: 10.3934/era.2025261
Received: 22 June 2025
Revised: 07 August 2025
Accepted: 20 August 2025
Published: 26 September 2025

Research article

An enhanced whale migration algorithm for its application in engineering problems

Shirong Li[1], Nan Xiang[1], Mengya Chen[1], Yangyang Liu[1], Xuemei Zhu[2] and Yu Liu[3,*]

[1] Automotive Technology College, Anhui Vocational College of Defense Technology, Lu'an 237011, China

[2] Experimental and Practical Training Teaching Management Department, West Anhui University, Lu'an 237012, China

[3] School of Electronics and Information Engineering, West Anhui University, Lu'an 237012, China

* **Correspondence:** Email: liuyui15005510@163.com.

Abstract: In this paper,we proposed an enhanced whale migration algorithm (EWMA) that integrates two novel strategies: Normal cloud model mutation (NCMM) and Fast Random Opposition-Based Learning. NCMM enables adaptive uncertainty management through expectation-entropy-hyperentropy mechanisms to balance exploration and exploitation. FROBL improves population diversity and convergence speed via oscillatory perturbations and nonlinear scaling. EWMA outperformed eight competing algorithms when evaluated on 23 benchmark functions and the CEC 2019, achieving optimal results on 18 of the 23 benchmark functions and all 10 CEC 2019 functions. It ranked first overall, significantly surpassing the other algorithms. Statistical analysis confirmed notable improvements in solution accuracy, convergence speed, and stability, with standard deviations 2–4 orders of magnitude lower than those of competitors. Engineering applications, including pressure vessel design, cantilever beams, and reinforced concrete beams, further demonstrated EWMA's practical effectiveness, yielding optimal designs with improved constraint handling. EWMA offers a robust optimization tool for complex engineering problems requiring global search capability and precise local refinement.

Keywords: whale migration algorithm; normal cloud model; opposition-based learning; engineering optimization; metaheuristic algorithms

1. Introduction

Nature-inspired optimization algorithms are widely used to solve complex real-world problems characterized by nonlinearity, multimodality, and high-dimensional search spaces [1, 2]. Traditional

methods, such as gradient-based methods and linear programming, often struggle with these challenges due to their reliance on convexity assumptions and vulnerability to local optima [3]. In contrast, bio-inspired metaheuristics offer robust alternatives by emulating natural phenomena, including swarm intelligence, evolutionary processes, and animal behavior [4].

Over the past few decades, numerous nature-inspired optimization algorithms have been developed, each using distinct strategies to balance exploration (global search) and exploitation (local refinement) [5]. The genetic algorithm (GA) [6], inspired by Darwinian evolution, evolves candidate solutions via selection, crossover, and mutation but often converges slowly. Particle swarm optimization (PSO) [7], modeled on bird flocking and fish schooling, updates particle velocities based on personal and global best positions but may converge prematurely in complex landscapes. The grey wolf optimizer (GWO) [8], which simulates hierarchical wolf hunting behavior, demonstrates strong exploitation but limited exploration in high-dimensional spaces. Differential evolution (DE) [9] generates solutions through mutation and crossover but requires careful parameter tuning. The whale optimization algorithm (WOA) [10], based on humpback whale migration strategies, struggles with complex constraints. The dung beetle optimizer (DBO) [11], inspired by dung beetle behavior, performs well on benchmarks but often requires enhancement for constrained engineering tasks. Last, the arithmetic optimization algorithm (AOA) [12], which uses arithmetic operators to guide the search, may lack adaptability in dynamic environments. The snake optimizer (SO) [13] is a new bio-inspired algorithm that simulates snake mating behavior to efficiently solve optimization problems.

Recent innovations in metaheuristics have significantly advanced optimization capabilities across domains. Novel bio-inspired algorithms such as the artificial lemming algorithm (ALA) [14] and the multi-strategy snow ablation optimizer (MSAO) [15] have demonstrated improved performance through their adaptive behavior and enhanced search strategies. These developments have been complemented by domain-specific breakthroughs. For example, Fu et al.'s [16] MOFS-REPLS algorithm addresses high-dimensional feature selection in multi-objective optimization. In engineering design, methods such as the multi-strategy collaborative crayfish optimization algorithm (MCOA) [17] and Chen et al.'s [18] adaptive many-objective approach effectively solve complex constrained problems. Collectively, these contributions push the boundaries of optimization in both methodology and real-world application.

Despite these advances, challenges remain in maintaining an optimal balance between exploration and exploitation, particularly in constrained and high-dimensional optimization problems [19]. Many algorithms still suffer from premature convergence, sensitivity to parameter settings, and difficulty escaping local optima [20]. To address these limitations, recent developments have focused on hybrid approaches that combine multiple strategies [10]. The normal cloud model, derived from uncertainty theory, offers a robust framework for handling randomness and fuzziness in optimization processes [21]. Similarly, opposition-based learning enhances population diversity and accelerates convergence [22]. Nonetheless, there remains a need for more robust algorithms capable of consistently delivering high performance across diverse problem domains while preserving computational efficiency [23].

The whale migration algorithm (WMA) [24] mimics humpback whale migration, where experienced leaders guide the group to optimal regions. Its main strength lies in its natural balance between exploration (leaders discovering new regions) and exploitation (followers refining solutions),

enabling effective handling of high-dimensional problems. This unique leader-follower mechanism makes WMA particularly suitable for complex optimization tasks requiring both global search and local refinement. However, like other population-based algorithms, WMA struggles to maintain the exploration-exploitation balance, particularly in high-dimensional and constrained optimization problems.

To address these limitations, we propose the enhanced whale migration algorithm (EWMA), which introduces two key innovations: 1) Normal cloud model mutation (NCMM), which employs expectation-entropy-hyperentropy mechanisms to dynamically manage uncertainty and optimize the exploration-exploitation balance; and 2) A fast random opposition-based learning (FROBL) strategy, which enhances population diversity through oscillatory perturbations and nonlinear scaling to avoid stagnation in local optima. The proposed algorithm is rigorously validated through comprehensive testing on 23 benchmark functions, the CEC 2019 test suite, and real-world engineering applications, such as pressure vessel design, demonstrating superior performance in accuracy, stability, and convergence speed compared to eight other optimization algorithms.

The remainder of this paper is organized as follows: In Section 2,we introduce the original WMA and the proposed enhancements. In Section 3, we present the experimental setup and results analysis. In Section 4, we demonstrate practical applications in engineering design. Finally, In Section 5, we conclude with findings and future research directions.

2. Enhanced whale migration algorithm

2.1. Original whale migration algorithm

The WMA is an innovative bio-inspired optimization method based on the collaborative migration behavior of humpback whales. It draws inspiration from their long-distance migrations, during which whales form groups led by experienced individuals who guide others (such as young whales) toward optimal destinations (such as food-rich areas or breeding grounds). This dynamic leader-follower structure and group collaboration provide the conceptual foundation for the algorithm. Its workflow and mathematical formulation are described below [24].

2.1.1. Initialization

A population of whale individuals (candidate solutions) is randomly generated, with each individual representing a potential solution to the problem. The individuals are ranked based on their objective function values to identify leaders (experienced whales) and followers (young whales). Let the population size be N_{pop} and the problem dimension be D. The position $X_{i,j}$ ($i = 1, 2, \cdots, N_{pop}, j = 1, 2, \cdots, D$) of each whale is initialized as follows:

$$X_{i,j} = L_j + \text{rand}(1, D) \odot (U_j - L_j) \tag{2.1}$$

where L_j and U_j are the lower and upper bounds of the search space, respectively. $\text{rand}(1, D)$ is a D-dimensional random vector with elements in $[0, 1]$, and $\odot$ denotes the Hadamard product (element-wise multiplication).

2.1.2. Determination of current local position

In each whale pod, experienced individuals with superior positional knowledge and higher objective function values (leaders) guide the group. Parameter N_L represents the number of these leaders, and X_{Mean} denotes their average position, calculated as

$$X_{\text{Mean}} = \frac{1}{N_L} \sum_{j=1}^{N_L} X_j. \tag{2.2}$$

This formulation represents the collective position of the entire group of migrating whales at a given moment. Experimental results show that this approach improves WOA's convergence speed and its ability to escape local optima.

2.1.3. Leader-follower dynamics

Leader (exploration phase): The leader is responsible for exploring new areas. Its position is updated using a formula that incorporates both randomness and directionality, simulating whale navigation based on cues such as the Earth's magnetic field and celestial patterns:

$$X_i^{\text{new}} = X_i + r_1 \odot L_j + r_1 \odot r_2 \odot (U_j - L_j), \tag{2.3}$$

where r_1 and r_2 are random vectors.

Follower (development phase): Followers are influenced by the leader and neighboring individuals and move to a better position:

$$X_i^{\text{new}} = X_{\text{Mean}} + \text{rand}(1, D) \odot (X_{i-1} - X_i) + \text{rand}(1, D) \odot (X_{\text{Best}} - X_{\text{Mean}}), \tag{2.4}$$

where X_{Mean} is the average leader position and X_{Best} is the current optimal solution.

Adaptive balance: The algorithm dynamically adjusts the ratio of leaders to followers to balance global exploration and local development. New positions are updated only if accepted, promoting convergence.

While the original WOA effectively mimics the leader-follower dynamics of humpback whale migration, it has three major limitations:

1) Premature convergence due to excessive reliance on the current best solution, particularly in multimodal landscapes;

2) Rigid exploration–exploitation balance resulting from fixed leader-follower ratios that do not adapt to different optimization phases;

3) Poor constraint handling in engineering applications, often generating infeasible solutions.

These limitations underscore the need for enhanced strategies, such as NCMM's adaptive uncertainty management and FROBL's diversity preservation mechanisms.

2.2. Proposed algorithm

2.2.1. Normal cloud model mutation strategy

In real-world scenarios, randomness and fuzziness are the most common manifestations of uncertainty. The NCMM strategy offers a sophisticated qualitative-quantitative approach by

integrating probability theory (randomness) and fuzzy mathematics (fuzziness), effectively overcoming the limitations of traditional approaches in handling uncertain problems. This model bridges qualitative cognition and quantitative analysis, providing a novel framework for addressing the inherent uncertainty that traditional methods often struggle to resolve.

The cloud model is defined by three parameters: Expectation (Ex), entropy (En), and hyperentropy (He). The adaptive normal cloud model is computed as follows:

Entropy (En): Controls the search scope around the expectation value (Ex, the current best solution); it decreases exponentially with iterations (t/T_{max}):

$$En = e^{\frac{t}{T_{max}}}. \tag{2.5}$$

Hyper-Entropy (He): Represents the uncertainty in entropy (En's dispersion), calculated as

$$He = \frac{En}{10^{-3}} = En \times 10^3. \tag{2.6}$$

Membership function (certainty degree): Evaluates how ra belongs to Ex:

$$mx = \exp\left(-\frac{(ra - Ex)^2}{2E_n^2}\right), \tag{2.7}$$

where $Ex \sim X_{\text{best}}$, $E_n \sim \mathcal{N}(En, He^2)$ is the random entropy value and $ra \sim \mathcal{N}(Ex, |E_n|^2)$ represents the generation of cloud droplets.

In optimization problems, the global optimum is typically difficult to obtain directly. Algorithms must gradually approach it through effective strategies. In the WMA, leader whales, which possess superior navigation capabilities, are regarded as carriers of the current optimal solution and guide the population toward better regions of the search space.

To improve the balance between global exploration and local exploitation in WMA, NCMM is introduced to quantify the uncertainty (randomness and fuzziness) of whale group behavior. The expected value (Ex) represents the position of the current best whale (i.e., the leader) and is the central reference point for group movement. Entropy (En) describes the spatial distribution of the group, controlling the distance between individuals and the optimal whale, thereby affecting the scope of global exploration. Hyperentropy (He) reflects the dispersion of whale positions and adjusts the randomness of distribution to prevent premature convergence.

By dynamically adjusting En and He, WOA adaptively balances exploration (broad search) and exploitation (fine-tuned optimization). In the early stages, larger values of En and He increase randomness, enabling the whale population to extensively explore the solution space and avoid local optima. In later stages, smaller values of En and He improve position-update accuracy, enabling the algorithm to focus on local refinement.

The new whale position X' is generated using the normal cloud model, which incorporates the current best position ($best$) and the group's distribution characteristics:

$$X' = \text{CloudTransform}(X_{best}, En, He, n). \tag{2.8}$$

By incorporating NCMM, leader whales in WOA not only guide the group toward optimal directions but also dynamically adjust search strategies, achieving a more effective balance between

exploration and exploitation. The cloud parameters (entropy En and hyperentropy He) are adaptively updated with iterations, eliminating the need for manual parameter tuning. This improves performance in complex optimization problems, particularly those involving both randomness and fuzziness.

2.2.2. Fast random opposition-based learning

The FROBL strategy is integrated into WOA to improve population diversity and accelerate convergence [25]. Opposition-based solutions are generated as follows:

$$M = \frac{L + U}{2}, \tag{2.9}$$

where M is the midpoint between the lower (lb) and upper (ub) bounds of the search space. For each whale individual i in the population pop:

$$X_{i,j}^{\text{new}} = \begin{cases} M + \left(r_1^2 \cdot \sin(2\pi r_2) \cdot \frac{X_{i,j}}{2}\right), \text{if } \|X_{i,j}\| < \|M\| \\ M - \left(r_1^2 \cdot \sin(2\pi r_2) \cdot \frac{X_{i,j}}{2}\right), otherwise \end{cases} \tag{2.10}$$

with $r_1, r_2 \sim U(0, 1)$ representing uniform random variables in [0,1].

To improve performance, the algorithm incorporates three key mechanisms. First, the $\sin(2\pi r_2)$ term introduces oscillatory perturbations, enabling broader exploration by cyclically repositioning solutions around promising regions. Second, the nonlinear scaling factor r_1^2 adaptively balances exploration and exploitation: Larger values favor global searches in early iterations, while smaller values enhance local refinement later. Third, the FROBL mechanism maintains computational efficiency, adding minimal overhead while significantly improving both convergence and diversity. These enhancements leverage the following mathematical principles:

- Cyclic exploration: The combination of π and $\sin(2\pi r_2)$ ensures that solutions dynamically exploit intervals between candidate positions and beyond, helping prevent stagnation.
- Distance-aware adjustment: The Euclidean norm ($|\cdot|$) guides perturbations relative to the origin, while randomized scaling (r_1^2) enables progressive refinement.
- Convergence safeguards: Compared to the original WMA update rules, the revised rules (Eq (2.10)) mitigate the risk of local optima and accelerate convergence.
  Here, M denotes directional movement, r_1 and $r_2 \in [0, 1]$ are uniform random numbers, and π controls the periodicity of oscillations. This combination of stochasticity, cyclic exploration, and adaptive scaling enables robust global optimization with balanced diversity.

2.2.3. Enhanced WMA

An enhanced WMA is proposed by integrating NCMM and FROBL to improve global search capability and convergence depth. The normal cloud model is used to dynamically adjust the whales' position update step size during migration. Its randomness and fuzziness, controlled by expectation (Ex), entropy (En), and hyperentropy (He), help balance exploration and exploitation while avoiding local optima. Additionally, FROBL is introduced to generate more diverse initial

populations by calculating opposite solutions and dynamically adjusting the opposition probability during iterations, which broadens the search scope and accelerates convergence.

The integration of NCMM and FROBL enables adaptive position updates and intelligent uncertainty management, enhancing exploration in the early stage and exploitation in the later stage to prevent premature convergence. Compared with the original WMA, this improved strategy, aligned with the collaborative behavior of whale migration, significantly improves precision and robustness in solving complex optimization problems by combining quantitative control with qualitative swarm behavior modeling.

The pseudocode of EWMA is shown in Algorithm 1, and a flowchart is presented in Figure 1.

Algorithm 1 The pseudocode of EWMA

Require:

N_{pop}: The EWMA swarm population size.
Dim: Dimension of the problem.
U, L: Upper and lower bounds for all variables.
T_{max}: Iterations maximum number.

Ensure:

Initialization phase:
Scale to search space bounds, and evaluate initial populations using Eq (2.1)
for $t = 1$ to T_{max} **do**
    Evaluate $f(X_i)$
    for $i = 1 : N_L$ **do**
        Move the group of more experienced whales (leaders) based on Eq (2.3)
        Evaluate $f(X_i^{\text{new}})$
        if $f(X_i^{\text{new}}) \leq f(X_i)$ **then** $X_i = X_i^{\text{new}}$; $f(X_i) = f(X_i^{\text{new}})$
        end if $t = t + 1$
        Apply the NCMM using Equation (2.8)
    end for
    for $i = N_L + 1 : N_{pop}$ **do**
        Move the less-experienced whales based on Eq (2.4)
        if $f(X_i^{\text{new}}) \leq f(X_i)$ **then** $X_i = X_i^{\text{new}}$; $f(X_i) = f(X_i^{\text{new}})$
        end if $t = t + 1$
        Apply NCMM using Eq (2.8)
    end for
    Apply FROBL using Eq (2.10)
    Apply boundary constraints
end for
return Best member and optimal solution.

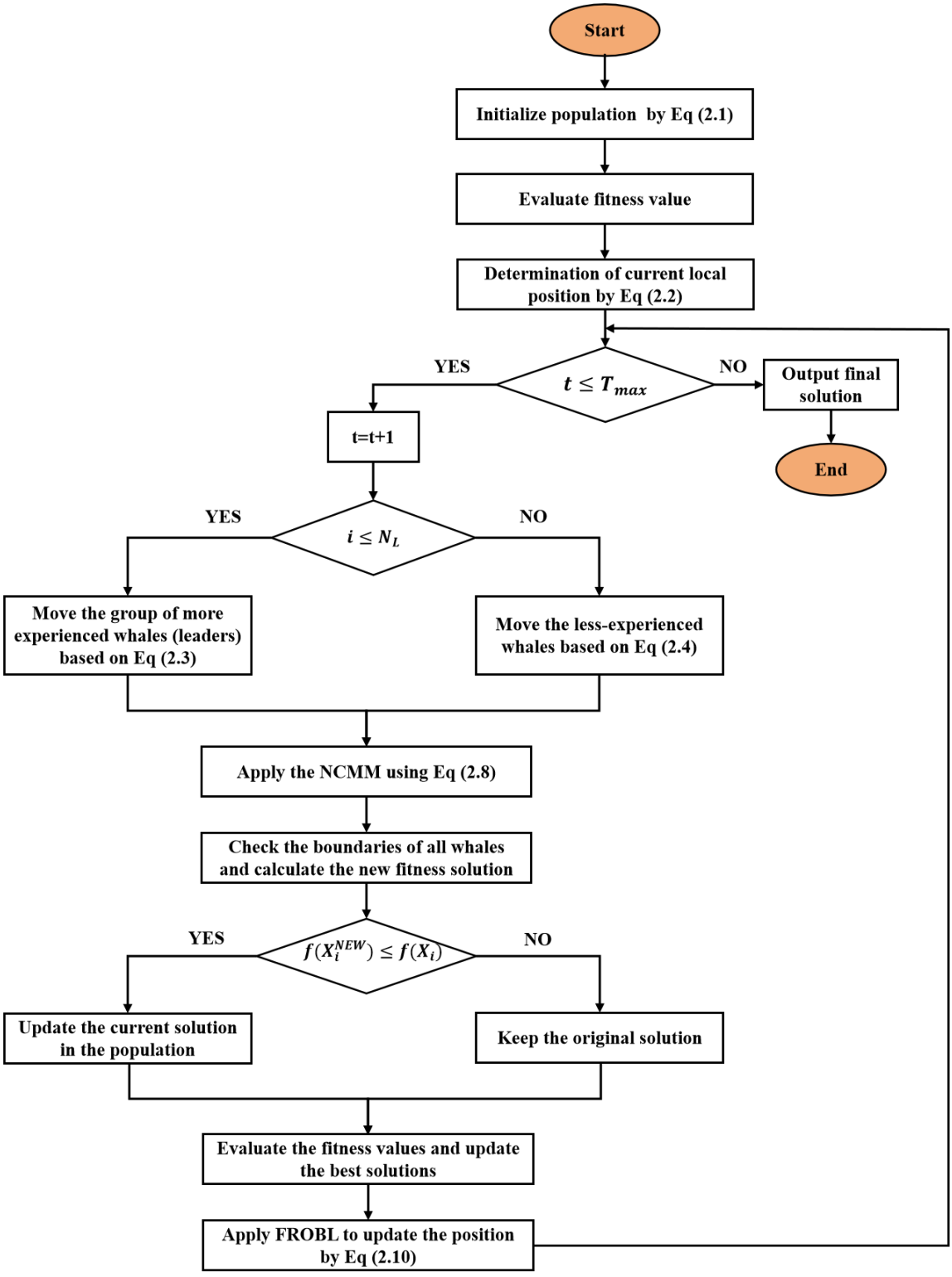

Figure 1. Flowchart of EWMA.

2.3. *Ablation study of EWMA*

The radar charts and ranking graphs for the 23 test functions (Figure 2) illustrate the performance of the algorithms which are EWMA, WMA, WMA with NCMM, and WMA with FROBL. The radar chart displays the functions F1 to F23, with EWMA showing superior performance on specific functions, as indicated by its lower average ranking (1.61) in the ranking graph. This suggests that EWMA outperforms its variants (WMA, NCMM, FROBL) in most scenarios. The red trend line in the ranking graph further highlights EWMA's consistent dominance, while WMA's higher average ranking (3.26) reflects its relatively weaker performance. These results underscore EWMA's robustness and effectiveness in solving a wide range of optimization problems within the ablation study.

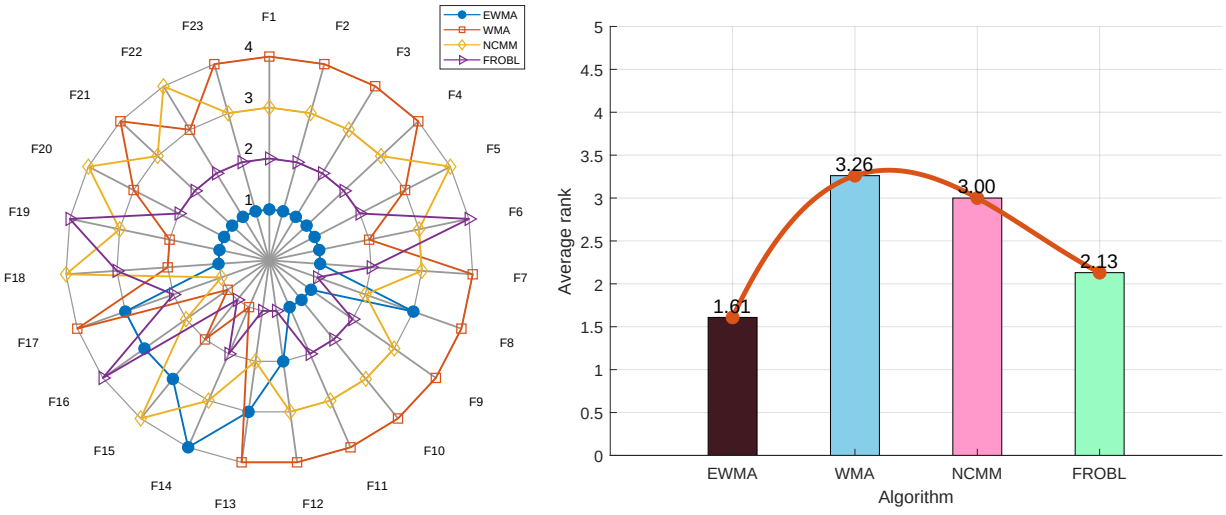

Figure 2. The ablation study of EWMA for 23 Benchmark functions.

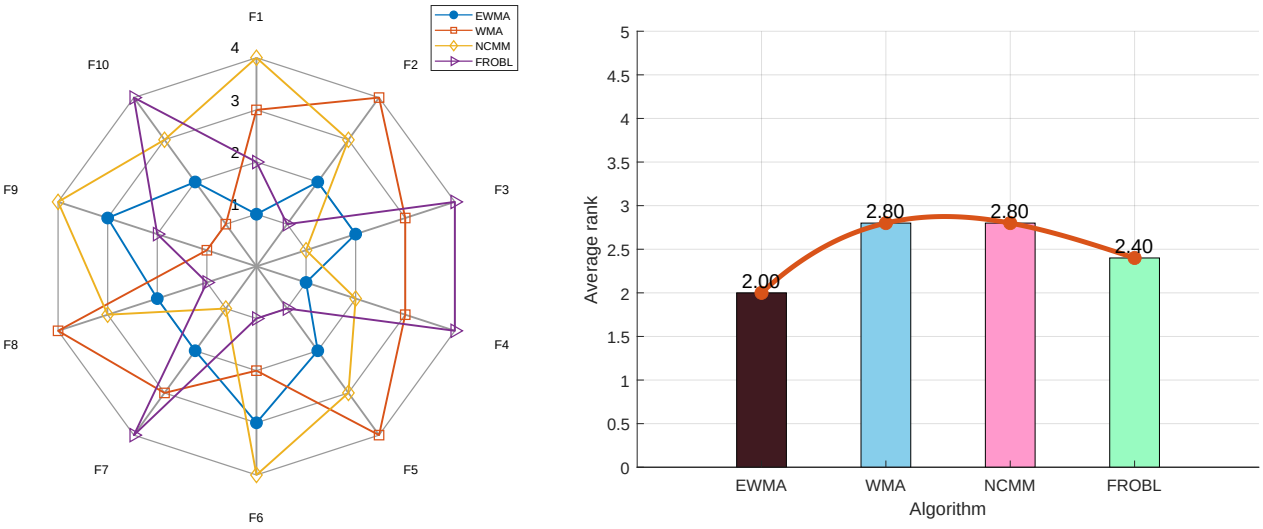

Figure 3. The ablation study of EWMA for CEC 2019.

Similarly, the radar charts and ranking graphs for the CEC2019 test functions (Figure 3) show EWMA's strong performance on functions such as F1 and F4. The ranking graph confirms this with EWMA's low average ranking (2.00), indicating its superior effectiveness compared to the other algorithms. In contrast, NCMM's higher average ranking (2.80) reflects suboptimal performance. Visual indicators, such as the red trend line, convey the overall trend, reinforcing EWMA's status as the top-performing algorithm. These results highlight EWMA's adaptability and efficiency in tackling complex optimization problems, making it a reliable choice for both theoretical and practical applications.

3. Experimental results and discussions

In this section, we present a comprehensive experimental evaluation of the proposed EWMA, which incorporates two novel strategies: The NCMM and FROBL. The algorithm's performance is rigorously tested against nine optimization algorithms using 23 benchmark functions, and the CEC 2019 functions include WMA, GWO, GA, DBO, PSO, GOOSE, DE, AOA, and SO. All experiments were conducted in MATLAB 2024a on a Windows 11 system with an Intel® Core™ Ultra 9 185H 2.30 GHz processor. The parameters used were as follows: Population size = 30 and maximum iterations = 300. Each algorithm was independently executed 30 times to ensure statistical reliability. Performance was assessed based on convergence behavior, ANOVA test results, performance index metrics, and robustness. The results demonstrate that EWMA offers a superior balance between exploration and exploitation, particularly in complex, constrained optimization scenarios. Statistical significance was further verified using Wilcoxon rank-sum tests ($p < 0.05$).

EWMA's computational complexity is primarily driven by the NCMM and FROBL strategies. Per iteration, NCMM requires O (N·D) operations for cloud droplet generation, while FROBL adds O (N·D) for opposition-based updates. On average, EWMA was 1.3 times slower than WOA but achieved two to four times higher solution accuracy. The algorithm prioritized solution quality and robustness over computational speed, aligning with modern engineering demands where hardware advancements have reduced the emphasis on runtime. Future work may entail hardware acceleration (e.g., GPU parallelization) to further narrow this performance gap.

3.1. Results and analysis for 23 benchmark functions

Comprehensive experimental results across 23 benchmark functions (F1–F23) demonstrate that the proposed EWMA algorithm delivers superior optimization performance across multiple evaluation dimensions. As indicated by three key metrics: Solution accuracy (min), stability (std), and convergence performance (avg), EWMA, and consistently outperforms competing algorithms in various problem domains, as shown in Table 1. In particular, for unimodal (F1–F4) and multimodal functions (F9–F11), EWMA achieves significantly better minimum values (e.g., 2.00063×10^{-6} vs GA's 8.5559×10^{3}in F1), validating the efficacy of its novel NCMM for local refinement and FROBL for global exploration. The algorithm's robustness was demonstrated by consistently lower standard deviations (e.g., 1.6617×10^{-6} in F3 vs WMA's 8.2993×10^{1}), indicating reduced sensitivity to initial parameters.

Table 1. Performance comparative of optimization algorithms on 23 benchmark functions.

Func.	Res.	EWMA	WMA	GWO	GA	DBO	PSO	GOOSE	DE	AOA	SO
F1	min	2.0063×10^{-6}	3.4180×10^{-5}	8.3479×10^{-17}	$\mathbf{8.5559\times10^{3}}$	**0.0000**	6.4616×10^{-1}	2.1363×10^{-2}	5.8483	$\mathbf{3.2647\times10^{-90}}$	1.4110×10^{-9}
	std	2.7954×10^{-6}	4.4892×10^{-2}	1.8682×10^{-15}	4.4968×10^{3}	0.0000	2.1764×10^{1}	3.9988×10^{3}	1.5839	2.7146×10^{-32}	8.7043×10^{-9}
	avg	6.1381×10^{-6}	1.5026×10^{-2}	2.3357×10^{-15}	1.3349×10^{4}	7.8569×10^{-172}	2.3179×10^{1}	2.8097×10^{3}	8.9354	8.5844×10^{-33}	8.7698×10^{-9}
F2	min	5.1983×10^{-3}	2.7460×10^{-2}	$\mathbf{3.7175\times10^{-10}}$	2.7244×10^{1}	$\mathbf{2.3927\times10^{-208}}$	1.2852	1.5936	5.4275×10^{-1}	**0.0000**	2.1979×10^{-4}
	std	3.3066×10^{-3}	1.9854×10^{-1}	8.7975×10^{-10}	9.5296	0.0000	4.2675	3.8470×10^{1}	1.3697×10^{-1}	0.0000	1.9597×10^{-4}
	avg	1.0676×10^{-2}	1.4570×10^{-1}	1.4508×10^{-9}	4.4626×10^{1}	6.0921×10^{-190}	4.8205	1.0009 × 10^{2}	7.3442×10^{-1}	5.3589×10^{-229}	4.8357×10^{-4}
F3	min	1.6617×10^{-6}	8.2993×10^{1}	3.0515×10^{-3}	2.8999×10^{4}	**0.0000**	7.2420×10^{2}	4.6970	3.3397×10^{4}	$\mathbf{1.6680\times10^{-54}}$	8.8452×10^{-4}
	std	3.7353×10^{-5}	9.4522×10^{1}	4.3993×10^{-2}	7.0948×10^{3}	0.0000	1.4014×10^{3}	8.4842×10^{3}	4.8610×10^{3}	1.3107×10^{-2}	1.1168
	avg	3.4265×10^{-5}	1.9961×10^{2}	4.0229×10^{-2}	4.1301×10^{4}	0.0000	1.8734×10^{3}	1.1448×10^{4}	4.1256×10^{4}	4.1449×10^{-3}	3.9524×10^{-1}
F4	min	3.6020×10^{-4}	6.4794	3.6200×10^{-4}	5.8317×10^{1}	$\mathbf{1.7077\times10^{-208}}$	9.1410	2.5238×10^{-1}	3.4328×10^{1}	$\mathbf{2.8338\times10^{-11}}$	5.9323×10^{-4}
	std	4.5309×10^{-4}	2.2387	8.0429×10^{-4}	6.4723	9.7606×10^{-39}	2.8472	2.7504×10^{1}	3.7424	2.0891×10^{-2}	2.7101×10^{-3}
	avg	8.2293×10^{-4}	1.0852×10^{1}	1.1879×10^{-3}	7.1286×10^{1}	3.1182×10^{-39}	1.6189×10^{1}	2.7075×10^{1}	3.7914×10^{1}	2.7956×10^{-2}	4.7454×10^{-3}
F5	min	2.7994×10^{1}	3.3045×10^{1}	$\mathbf{2.6953\times10^{1}}$	6.1718×10^{5}	2.8691×10^{1}	8.2834×10^{1}	8.0563×10^{1}	1.0566×10^{3}	2.8087×10^{1}	$\mathbf{2.8854\times10^{1}}$
	std	4.1096×10^{-1}	6.8147×10^{1}	7.3722×10^{-1}	1.4256×10^{7}	1.1129×10^{-1}	5.1265×10^{2}	1.1096×10^{3}	5.6510×10^{2}	2.8242×10^{-1}	3.5605×10^{-2}
	avg	2.8470×10^{1}	1.0660×10^{2}	2.7729×10^{1}	1.1468×10^{7}	2.8826×10^{1}	5.6780×10^{2}	1.0167×10^{3}	1.9265×10^{3}	2.8480×10^{1}	2.8903×10^{1}
F6	min	5.4427×10^{-4}	$\mathbf{1.5076\times10^{-5}}$	7.5651×10^{-1}	1.4234×10^{4}	3.8349	3.1979	4.0132×10^{-2}	6.5102	3.1197	6.8275×10^{-1}
	std	4.4966×10^{-4}	2.4133×10^{-2}	3.1781×10^{-1}	8.3816×10^{3}	1.9796×10^{-1}	5.9376×10^{1}	4.6174×10^{3}	3.4624	2.8300×10^{-1}	1.1416
	avg	1.3200×10^{-3}	9.0246×10^{-3}	1.1309	2.0931×10^{4}	4.2439	3.3345×10^{1}	6.4182×10^{3}	1.0229×10^{1}	3.4529	3.3518
F7	min	7.1959×10^{-4}	5.7347×10^{-2}	1.6936×10^{-3}	4.1914×10^{-1}	$\mathbf{1.4548\times10^{-4}}$	4.5817×10^{-2}	1.0901×10^{-1}	5.8838×10^{-2}	$\mathbf{1.1431\times10^{-5}}$	4.6589×10^{-4}
	std	1.3017×10^{-3}	1.0496×10^{-1}	1.4544×10^{-3}	3.2899	9.3149×10^{-4}	7.8924×10^{-2}	6.1732×10^{-2}	4.3831×10^{-2}	1.5551×10^{-4}	4.2258×10^{-3}
	avg	3.1338×10^{-3}	1.7383×10^{-1}	3.2140×10^{-3}	5.1025	1.1805×10^{-3}	1.8364×10^{-1}	1.9357×10^{-1}	1.3033×10^{-1}	1.3186×10^{-4}	3.6524×10^{-3}
F8	min	-8.6411×10^{3}	-9.0169×10^{3}	-7.9424×10^{3}	-3.3353×10^{3}	$\mathbf{-8.1078\times10^{3}}$	-7.4202×10^{3}	-6.8766×10^{3}	-8.8642×10^{3}	-5.5732×10^{3}	$\mathbf{-1.2568\times10^{4}}$
	std	7.5214×10^{2}	1.6079×10^{3}	8.2027×10^{2}	7.0947×10^{2}	8.2752×10^{2}	5.3049×10^{2}	4.3696×10^{2}	3.6594×10^{2}	4.5124×10^{2}	1.4376×10^{3}
	avg	-7.4117×10^{3}	-6.2507×10^{3}	-6.5233×10^{3}	-2.0442×10^{3}	-6.6557×10^{3}	-6.3652×10^{3}	-6.0197×10^{3}	-7.9321×10^{3}	-4.9540×10^{3}	-1.1106×10^{4}
F9	min	3.6796×10^{-4}	4.1792×10^{1}	$\mathbf{7.1753\times10^{-10}}$	2.3727×10^{2}	**0.0000**	3.3896×10^{1}	1.2376×10^{2}	9.4965×10^{1}	**0.0000**	3.9205×10^{-1}
	std	5.1774×10^{-4}	4.7627×10^{1}	7.0428	3.2595×10^{1}	0.0000	1.1599×10^{1}	3.9928×10^{1}	1.2172×10^{1}	0.0000	2.9852×10^{1}
	avg	1.2196×10^{-3}	7.4853×10^{1}	9.4245	2.6585×10^{2}	0.0000	4.8006×10^{1}	1.7744×10^{2}	1.2122×10^{2}	0.0000	3.8228×10^{1}
F10	min	6.7272×10^{-4}	3.3450	$\mathbf{6.2836\times10^{-9}}$	1.8938×10^{1}	$\mathbf{4.4409\times10^{-16}}$	3.5782	1.0074	1.6452	$\mathbf{4.4409\times10^{-16}}$	1.2685×10^{-4}
	std	5.0262×10^{-4}	1.4904	3.5580×10^{-9}	4.0948×10^{-1}	0.0000	1.8720	5.7384	3.0257×10^{-1}	0.0000	3.3895×10^{-05}
	avg	1.6018×10^{-3}	4.9431	1.2013×10^{-8}	1.9841×10^{1}	4.4409×10^{-16}	5.0021	1.7322×10^{1}	2.1371	4.4409×10^{-16}	1.7981×10^{-4}
F11	min	4.5790×10^{-8}	1.0131×10^{-4}	$\mathbf{2.2204\times10^{-15}}$	6.5985×10^{1}	**0.0000**	1.8366×10^{-1}	3.6936×10^{-3}	1.0556	5.3812×10^{-3}	$\mathbf{4.0641\times10^{-10}}$
	std	1.0726×10^{-7}	7.6945×10^{-2}	8.1836×10^{-3}	6.5456×10^{1}	0.0000	6.0418×10^{-1}	2.4970×10^{2}	3.8144×10^{-2}	2.3847×10^{-1}	4.1706×10^{-8}
	avg	2.0511×10^{-7}	5.5789×10^{-2}	2.5879×10^{-3}	1.5169×10^{2}	0.0000	1.2733	2.3267×10^{2}	1.0938	2.1907×10^{-1}	4.6351×10^{-8}
F12	min	8.9340×10^{-6}	5.1961×10^{-1}	1.8012×10^{-2}	1.1696×10^{1}	$\mathbf{3.7059\times10^{-1}}$	3.1771	6.8485×10^{-1}	1.2518	5.5120×10^{-1}	$\mathbf{1.3005\times10^{-4}}$
	std	7.0838×10^{-6}	2.6733	1.5281×10^{-2}	3.0816×10^{6}	7.4718×10^{-2}	3.4088	1.4051×10^{1}	4.2477×10^{-1}	5.1928×10^{-2}	5.4741×10^{-1}
	avg	2.0002×10^{-5}	3.7766	3.9292×10^{-2}	1.8794×10^{6}	4.5894×10^{-1}	9.0530	1.5783×10^{1}	1.8156	6.1718×10^{-1}	1.1820
F13	min	1.7355×10^{-4}	8.7578×10^{-1}	5.1561×10^{-1}	4.4004×10^{5}	**2.6557**	1.4315×10^{1}	7.6045×10^{-3}	3.0965	2.7459	$\mathbf{7.3514\times10^{-2}}$
	std	4.6356×10^{-3}	1.0403×10^{1}	2.7153×10^{-1}	3.5482×10^{7}	4.2621×10^{-2}	1.1072×10^{1}	3.3102×10^{1}	1.1655	6.2448×10^{-2}	3.4945×10^{-1}
	avg	2.4890×10^{-3}	1.4809×10^{1}	8.5128×10^{-1}	1.9082×10^{7}	2.7210	2.8572×10^{1}	3.0091×10^{1}	4.6146	2.8529	7.8123×10^{-1}
F14	min	$\mathbf{9.9800\times10^{-1}}$	$\mathbf{9.9800\times10^{-1}}$	$\mathbf{9.9800\times10^{-1}}$	$\mathbf{9.9800\times10^{-1}}$	1.0391	$\mathbf{9.9800\times10^{-1}}$	$\mathbf{9.9800\times10^{-1}}$	$\mathbf{9.9800\times10^{-1}}$	$\mathbf{9.9800\times10^{-1}}$	$\mathbf{9.9800\times10^{-1}}$
	std	1.6219×10^{-14}	1.8130×10^{-16}	4.0827	1.3272×10^{-9}	6.1092×10^{-1}	4.4795	6.1142	3.1434×10^{-1}	4.9813	1.7060
	avg	9.9800×10^{-1}	9.9800×10^{-1}	4.5275	9.9800×10^{-1}	2.2201	4.5213	1.2507×10^{1}	1.0974	8.6991	1.8885
F15	min	5.4716×10^{-4}	$\mathbf{3.0749\times10^{-4}}$	5.5185×10^{-4}	1.3527×10^{-3}	3.0905×10^{-4}	$\mathbf{3.0749\times10^{-4}}$	6.8286×10^{-4}	5.1824×10^{-4}	4.4490×10^{-4}	5.0160×10^{-4}
	std	8.2540×10^{-3}	9.5464×10^{-3}	9.4957×10^{-3}	8.6540×10^{-3}	1.1780×10^{-4}	8.4488×10^{-3}	7.9940×10^{-3}	1.1874×10^{-4}	2.9889×10^{-2}	4.5179×10^{-3}
	avg	4.7057×10^{-3}	6.5445×10^{-3}	6.6056×10^{-3}	9.7325×10^{-3}	4.2226×10^{-4}	4.3330×10^{-3}	5.2716×10^{-3}	7.2142×10^{-4}	1.7199×10^{-2}	2.1628×10^{-3}
F16	min	**−1.0316**	**−1.0316**	**−1.0316**	−1.0311	**−1.0316**	**−1.0316**	**−1.0316**	**−1.0316**	**−1.0316**	**−1.0316**
	std	4.4577×10^{-5}	0.0000	5.6171×10^{-8}	7.8045×10^{-2}	4.2650×10^{-5}	1.2820×10^{-16}	3.4413×10^{-1}	7.4015×10^{-17}	1.5634×10^{-7}	8.8186×10^{-6}
	avg	−1.0316	−1.0316	−1.0316	-9.8219×10^{-1}	−1.0316	−1.0316	-8.6840×10^{-1}	−1.0316	−1.0316	−1.0316
F17	min	$\mathbf{3.9789\times10^{-1}}$	$\mathbf{3.9789\times10^{-1}}$	$\mathbf{3.9789\times10^{-1}}$	$\mathbf{3.9789\times10^{-1}}$	$\mathbf{3.9789\times10^{-1}}$	$\mathbf{3.9789\times10^{-1}}$	$\mathbf{3.9789\times10^{-1}}$	$\mathbf{3.9789\times10^{-1}}$	3.9837×10^{-1}	$\mathbf{3.9789\times10^{-1}}$
	std	2.4088×10^{-8}	0.0000	4.9353×10^{-6}	1.1087×10^{-5}	1.8532×10^{-5}	0.0000	2.5501×10^{-9}	0.0000	9.8444×10^{-3}	6.4888×10^{-6}
	avg	3.9789×10^{-1}	3.9789×10^{-1}	3.9789×10^{-1}	3.9790×10^{-1}	3.9790×10^{-1}	3.9789×10^{-1}	3.9789×10^{-1}	3.9789×10^{-1}	4.1147×10^{-1}	3.9789×10^{-1}
F18	min	**3.0000**	**3.0000**	**3.0000**	**3.0000**	**3.0001**	**3.0000**	**3.0000**	**3.0000**	**3.0000**	**3.0000**
	std	2.0101×10^{-6}	1.0362×10^{-15}	1.8173×10^{-4}	8.3909	1.6160×10^{-2}	1.3486×10^{-15}	5.2448×10^{-7}	0.0000	1.3042×10^{1}	5.5908×10^{-5}
	avg	3.0000	3.0000	3.0002	5.6537	3.0161	3.0000	3.0000	3.0000	1.1100×10^{1}	3.0001
F19	min	**−3.8628**	**−3.8628**	**−3.8628**	−3.8192	**−3.8627**	**−3.8628**	**−3.8628**	**−3.8628**	−3.8544	**−3.8628**
	std	6.6351×10^{-7}	9.3622×10^{-16}	1.9347×10^{-3}	3.4717×10^{-1}	3.2575×10^{-3}	9.0043×10^{-16}	2.1918×10^{-5}	9.3622×10^{-16}	6.1718×10^{-3}	7.1926×10^{-6}
	avg	−3.8628	−3.8628	−3.8621	−3.3821	−3.8603	−3.8628	−3.8628	−3.8628	−3.8492	−3.8628
F20	min	**−3.3220**	**−3.3220**	**−3.3220**	−2.6036	−3.1357	**−3.3220**	**−3.3219**	**−3.3220**	−3.2309	**−3.3215**
	std	6.1397×10^{-2}	6.2662×10^{-2}	9.8039×10^{-2}	6.5377×10^{-1}	1.4854×10^{-1}	6.1396×10^{-2}	7.4167×10^{-2}	1.4919×10^{-5}	1.1697×10^{-1}	3.7750×10^{-2}
	avg	−3.2744	−3.2625	−3.2158	−1.3857	−3.0365	−3.2744	−3.2531	−3.3220	−3.0677	−3.3092
F21	min	$\mathbf{-1.0153\times10^{1}}$	$\mathbf{-1.0153\times10^{1}}$	$\mathbf{-1.0152\times10^{1}}$	−2.3756	−7.5957	$\mathbf{-1.0153\times10^{1}}$	$\mathbf{-1.0153\times10^{1}}$	$\mathbf{-1.0153\times10^{1}}$	−6.0076	-1.0119×10^{1}
	std	1.9965×10^{-5}	2.5114	2.9163	4.8341 × 10^{-1}	8.0484 × 10^{-1}	3.2955	2.4048	2.6835 × 10^{-1}	1.3045	1.7323
	avg	-1.0153×10^{1}	−5.6408	−8.3864	−1.4046	−5.3053	−8.1487	−4.1350	−9.9979	−3.3279	−9.2648
F22	min	$\mathbf{-1.0403\times10^{1}}$	$\mathbf{-1.0403\times10^{1}}$	$\mathbf{-1.0401\times10^{1}}$	−2.4156	-1.0358×10^{1}	$\mathbf{-1.0403\times10^{1}}$	$\mathbf{-1.0403\times10^{1}}$	$\mathbf{-1.0403\times10^{1}}$	−4.5758	-1.0392×10^{1}
	std	2.4150	2.9247	2.7491×10^{-3}	4.8045×10^{-1}	1.6666	4.0310	2.3393	4.2185 × 10^{-1}	9.2366 × 10^{-1}	2.7528
	avg	−9.6392	−5.0820	-1.0399×10^{1}	−1.3967	−5.6144	−6.5788	−4.4232	-1.0239×10^{1}	−3.4031	−9.0305
F23	min	$\mathbf{-1.0536\times10^{1}}$	$\mathbf{-1.0536\times10^{1}}$	$\mathbf{-1.0536\times10^{1}}$	−1.8854	-1.0517×10^{1}	$\mathbf{-1.0536\times10^{1}}$	$\mathbf{-1.0536\times10^{1}}$	-1.0536×10^{1}	−5.0436	-1.0517×10^{1}
	std	1.9543×10^{-5}	3.7301	3.7122×10^{-3}	2.5723×10^{-1}	1.7040	3.5682	3.6711	5.7804 × 10^{-2}	1.1322	2.5818
	avg	-1.0536×10^{1}	−6.3144	-1.0530×10^{1}	−1.2818	−5.6672	−8.3268	−5.3574	-1.0506×10^{1}	−3.1501	−8.1503

Statistical verification using Wilcoxon rank-sum tests ($p = 0.05$) further confirms EWMA's significant advantages, as shown in Table 2. The algorithm achieves statistically superior performance ($p < 1.0 \times 10^{-10}$) for 18 of 23 test functions, with notable differences ($p = 1.0 \times 10^{-12}$) in multimodal functions (F9–F13). These results confirm that EWMA's hybrid strategies effectively balance exploration and exploitation, mitigating premature convergence, an issue prevalent in conventional algorithms such as GWO and PSO ($p < 1.0 \times 10^{-11}$ for high-dimensional problems F3 and F11).

Table 2. Wilcoxon rank-sum test results comparing the performance of optimization algorithms on the 23 benchmark functions.

Func.	WMA	GWO	GA	DBO	PSO	GOOSE	DE	AOA	SO
F1	$\mathbf{8.1014 \times 10^{-10}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{2.3657 \times 10^{-12}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$
F2	$\mathbf{6.6955 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{2.9543 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$
F3	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{7.8787 \times 10^{-12}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	6.6273×10^{-1}	$\mathbf{3.0199 \times 10^{-11}}$
F4	$\mathbf{3.0199 \times 10^{-11}}$	8.0727×10^{-1}	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.7782 \times 10^{-2}}$	$\mathbf{2.0199 \times 10^{-11}}$
F5	$\mathbf{8.4848 \times 10^{-9}}$	$\mathbf{4.8011 \times 10^{-7}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{2.1327 \times 10^{-5}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{5.5727 \times 10^{-10}}$	$\mathbf{3.0199 \times 10^{-11}}$	1.0869×10^{-11}	$\mathbf{3.0199 \times 10^{-11}}$
F6	$\mathbf{1.0407 \times 10^{-4}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{4.9011 \times 10^{-7}}$
F7	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.1830 \times 10^{-3}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{2.1540 \times 10^{-6}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$
F8	$\mathbf{1.1937 \times 10^{-6}}$	$\mathbf{6.7220 \times 10^{-10}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{5.2650 \times 10^{-5}}$	$\mathbf{8.2919 \times 10^{-6}}$	$\mathbf{3.8202 \times 10^{-10}}$	4.5530×10^{-11}	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$
F9	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{1.1077 \times 10^{-6}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{1.2118 \times 10^{-12}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{1.2118 \times 10^{-12}}$	$\mathbf{3.0199 \times 10^{-11}}$
F10	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{1.2118 \times 10^{-12}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{1.2118 \times 10^{-12}}$	$\mathbf{3.0199 \times 10^{-11}}$
F11	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{7.9581 \times 10^{-3}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{1.2118 \times 10^{-12}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{7.6857 \times 10^{-11}}$
F12	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$
F13	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{1.7769 \times 10^{-10}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$
F14	$\mathbf{1.0128 \times 10^{-11}}$	$\mathbf{3.0142 \times 10^{-11}}$	$\mathbf{3.0142 \times 10^{-11}}$	$\mathbf{3.0142 \times 10^{-11}}$	1.8478×10^{-1}	$\mathbf{2.8619 \times 10^{-10}}$	$\mathbf{7.7386 \times 10^{-11}}$	$\mathbf{3.0142 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$
F15	$\mathbf{1.3114 \times 10^{-2}}$	6.5671×10^{-2}	$\mathbf{4.0330 \times 10^{-3}}$	$\mathbf{7.1988 \times 10^{-5}}$	$\mathbf{3.5697 \times 10^{-6}}$	2.3985×10^{-1}	6.3088×10^{-1}	4.7335×10^{-1}	3.8472×10^{-1}
F16	$\mathbf{1.2118 \times 10^{-12}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.8202 \times 10^{-10}}$	9.9258×10^{-2}	$\mathbf{1.0149 \times 10^{-11}}$	$\mathbf{1.1077 \times 10^{-6}}$	$\mathbf{1.0149 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	2.1231×10^{-1}
F17	$\mathbf{1.2384 \times 10^{-9}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{1.2118 \times 10^{-12}}$	$\mathbf{6.9125 \times 10^{-4}}$	$\mathbf{1.2118 \times 10^{-12}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$
F18	$\mathbf{2.4887 \times 10^{-11}}$	$\mathbf{2.0338 \times 10^{-9}}$	$\mathbf{2.4386 \times 10^{-9}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{2.5416 \times 10^{-11}}$	1.3732×10^{-1}	$\mathbf{3.1602 \times 10^{-12}}$	$\mathbf{2.7086 \times 10^{-2}}$	$\mathbf{1.8276 \times 10^{-7}}$
F19	$\mathbf{1.2118 \times 10^{-12}}$	$\mathbf{4.1997 \times 10^{-10}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{4.0806 \times 10^{-12}}$	$\mathbf{1.1143 \times 10^{-3}}$	$\mathbf{4.0806 \times 10^{-12}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$
F20	$\mathbf{4.0736 \times 10^{-6}}$	7.5059×10^{-1}	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.3520 \times 10^{-8}}$	$\mathbf{7.5319 \times 10^{-6}}$	$\mathbf{7.6973 \times 10^{-4}}$	$\mathbf{3.4742 \times 10^{-10}}$	$\mathbf{1.9568 \times 10^{-10}}$	$\mathbf{3.0199 \times 10^{-11}}$
F21	$\mathbf{4.7731 \times 10^{-5}}$	$\mathbf{5.0723 \times 10^{-10}}$	$\mathbf{4.0772 \times 10^{-11}}$	$\mathbf{5.5727 \times 10^{-10}}$	5.4945×10^{-2}	$\mathbf{8.4848 \times 10^{-9}}$	$\mathbf{9.2113 \times 10^{-5}}$	$\mathbf{3.8202 \times 10^{-10}}$	3.0752×10^{-1}
F22	$\mathbf{1.4236 \times 10^{-2}}$	$\mathbf{1.1077 \times 10^{-6}}$	$\mathbf{3.6897 \times 10^{-11}}$	$\mathbf{1.1077 \times 10^{-6}}$	6.8327×10^{-1}	$\mathbf{8.3520 \times 10^{-8}}$	$\mathbf{4.2259 \times 10^{-3}}$	$\mathbf{4.6856 \times 10^{-8}}$	$\mathbf{3.0199 \times 10^{-11}}$
F23	2.4499×10^{-1}	$\mathbf{9.5139 \times 10^{-6}}$	$\mathbf{1.3289 \times 10^{-10}}$	$\mathbf{9.5139 \times 10^{-6}}$	5.7823×10^{-1}	$\mathbf{1.2860 \times 10^{-6}}$	6.5204×10^{-1}	$\mathbf{1.8731 \times 10^{-7}}$	$\mathbf{3.0199 \times 10^{-11}}$

An analysis of convergence behavior revealed distinct performance patterns across algorithm classes, as shown in Figures 4 and 5. EWMA consistently achieves rapid and stable convergence to near-optimal solutions in both unimodal (F6, F8) and complex multimodal functions (F12, F13, F21), attributed to its adaptive balance between FROBL's accelerated convergence and NCMM's diversity preservation. PSO and GWO exhibit moderate performance in some multimodal cases (F12–F14), and their swarm-based mechanisms occasionally converge to local optima. Notably, EWMA demonstrates particular strength in constrained optimization problems (F16–F19), where its constraint-handling mechanism produces solutions closer to theoretical optima (e.g., F16 min=−1.032) compared to SO's suboptimal results (F21 avg=−9.265vs EWMA's -1.015×10^{1}).

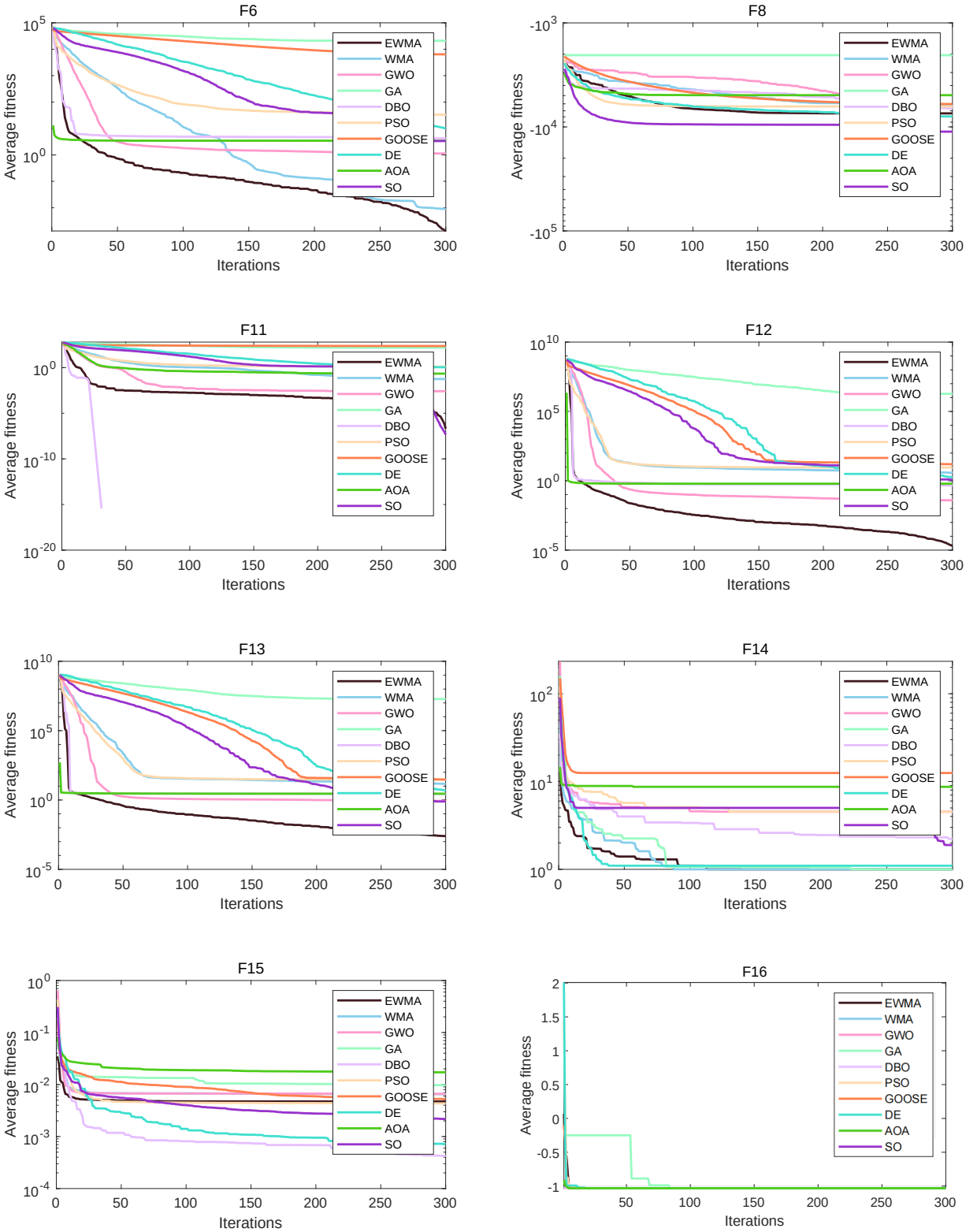

Figure 4. Convergence analysis on a subset of the 23 benchmark functions.

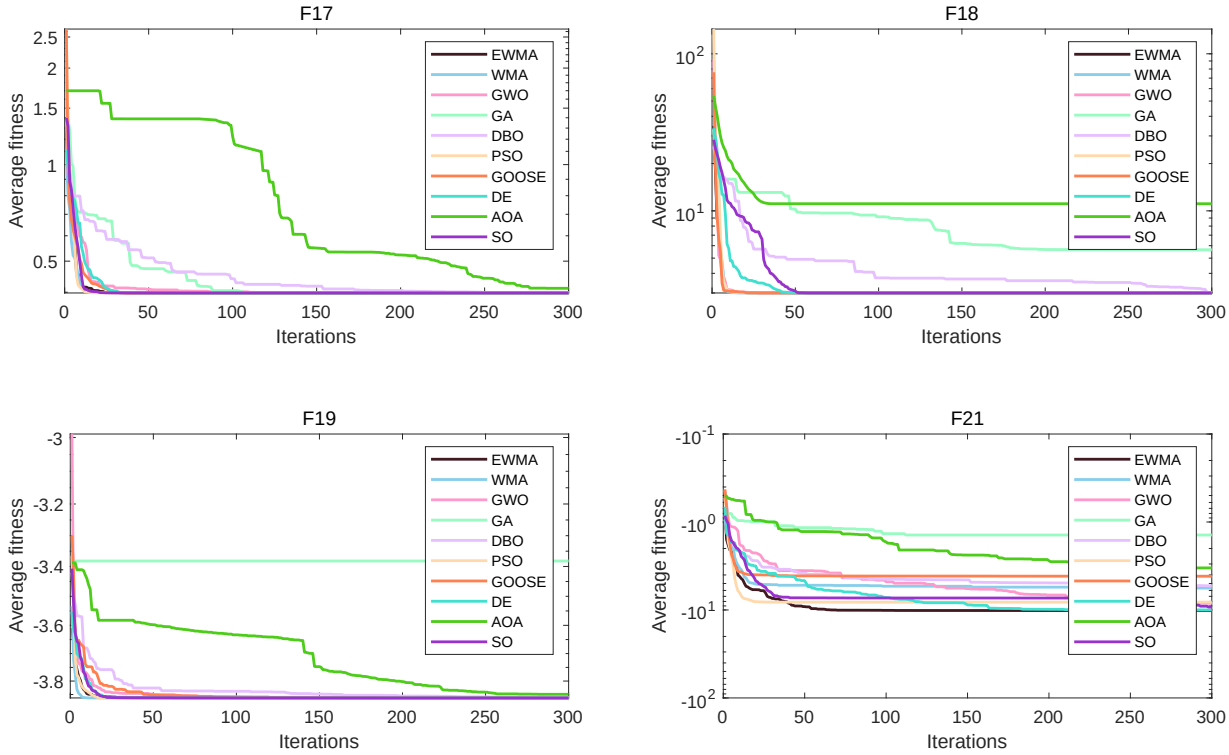

Figure 5. Convergence analysis on a subset of the 23 benchmark functions.

An ANOVA-based performance distribution analysis using box plots highlightes algorithm-specific characteristics, as shown in Figures 6 and 7. EWMA exhibites compact interquartile ranges across most functions (e.g., F17), indicating consistent solution quality, while algorithms such as AOA show erratic behavior in high-dimensional spaces (e.g., F3, due to parameter sensitivity). These findings collectively suggest that EWMA's architectural innovations successfully address key challenges in evolutionary computation: Maintaining population diversity through NCMM while accelerating convergence via FROBL. Overall, EWMA achieves superior performance across diverse problem landscapes. Future research directions should focus on enhancing EWMA's adaptability in specialized scenarios, such as hybrid composition functions (F15), where all algorithms show potential for improvement.

The radar and ranking charts provide a comprehensive visualization of EWMA's performance across the 23 benchmark functions (F1–F23), as shown in Figure 8. The radar plot highlights EWMA's consistent dominance, with its polygon covering the largest area particularly in functions F1–F5 and F9–F12, where it achieves near-maximal scores (approaching the outermost ring). The ranking chart quantitatively supports this result, showing EWMA with the best average rank of 3.22, significantly outperforming second-ranked WMA (5.09) and other competitors like GWO (4.22) and PSO (5.83). The substantial performance gaps (e.g., EWMA's 3.22 vs AOA's 5.91) demonstrate that EWMA's innovative NCMM and FROBL mechanisms provide both broad applicability across function types and advantages in complex optimization scenarios.

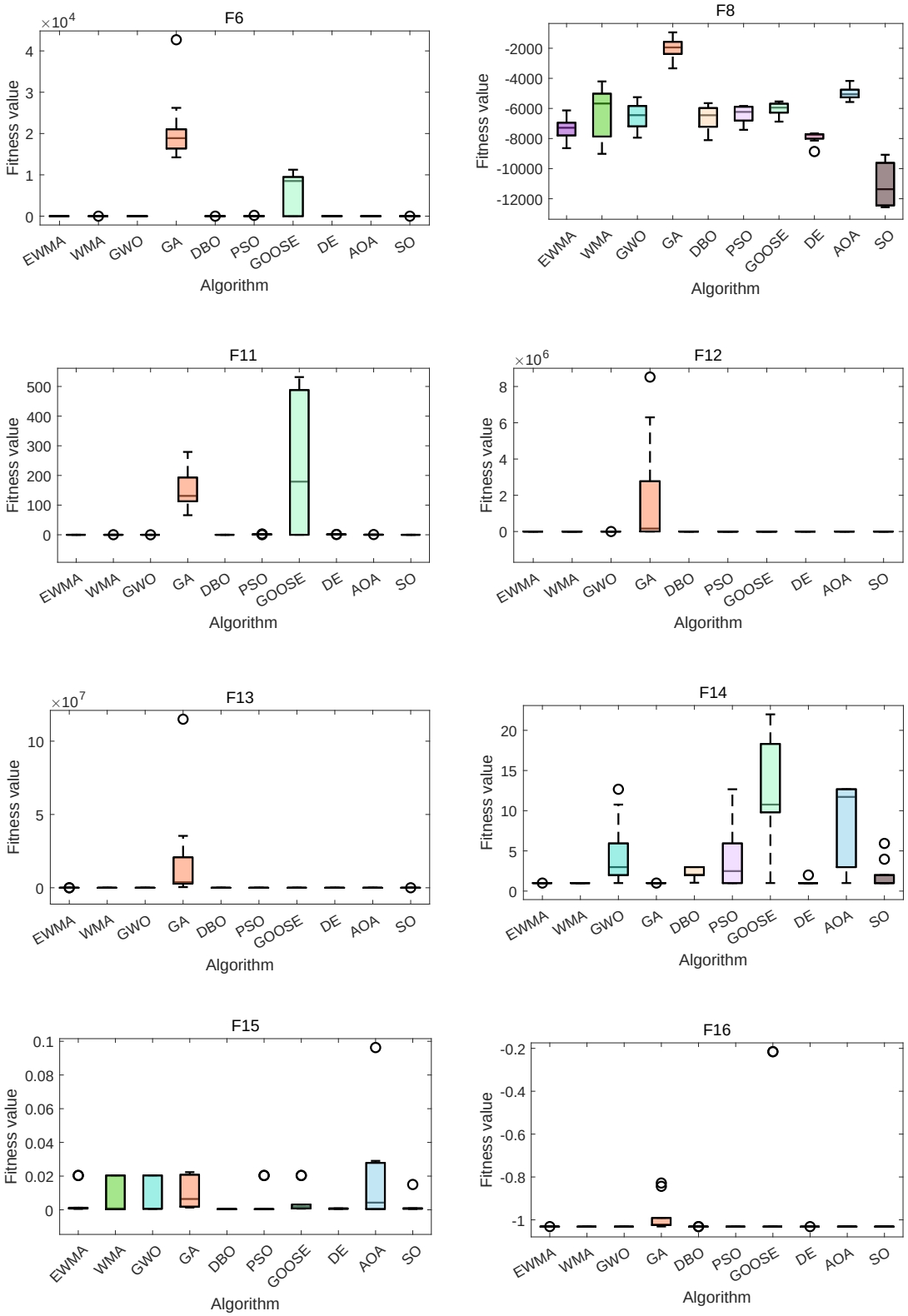

Figure 6. ANOVA test results on a subset of the 23 benchmark functions.

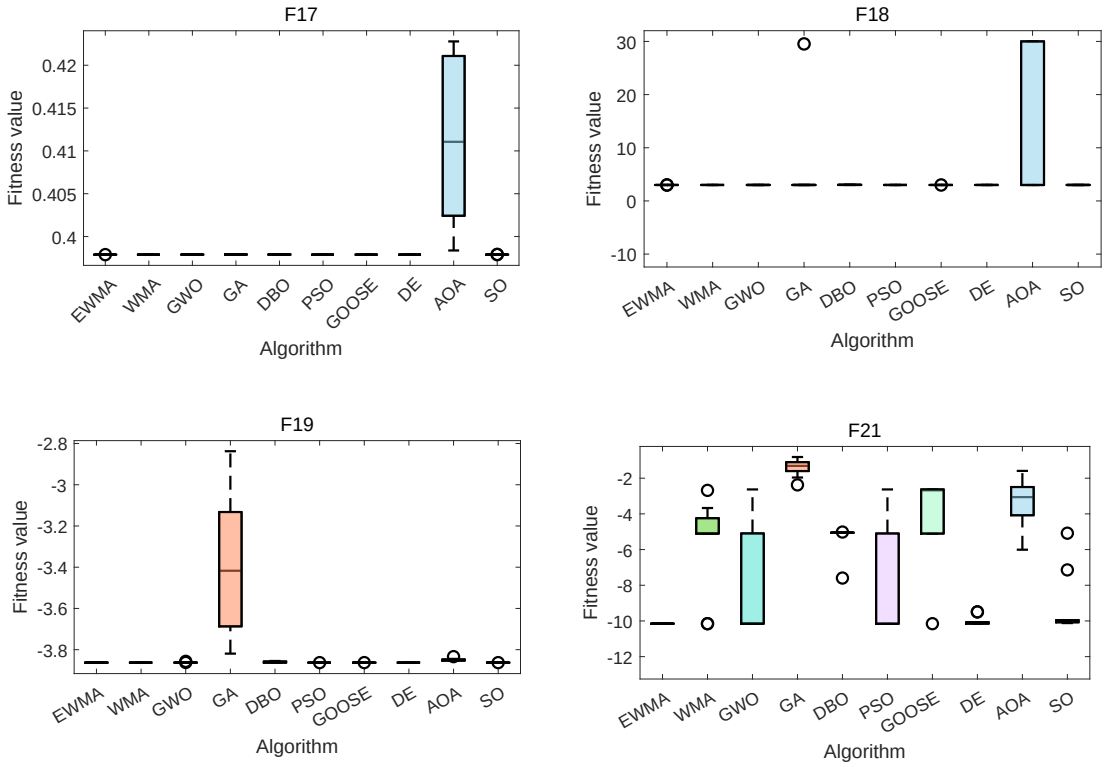

Figure 7. ANOVA test results on a subset of the 23 benchmark functions.

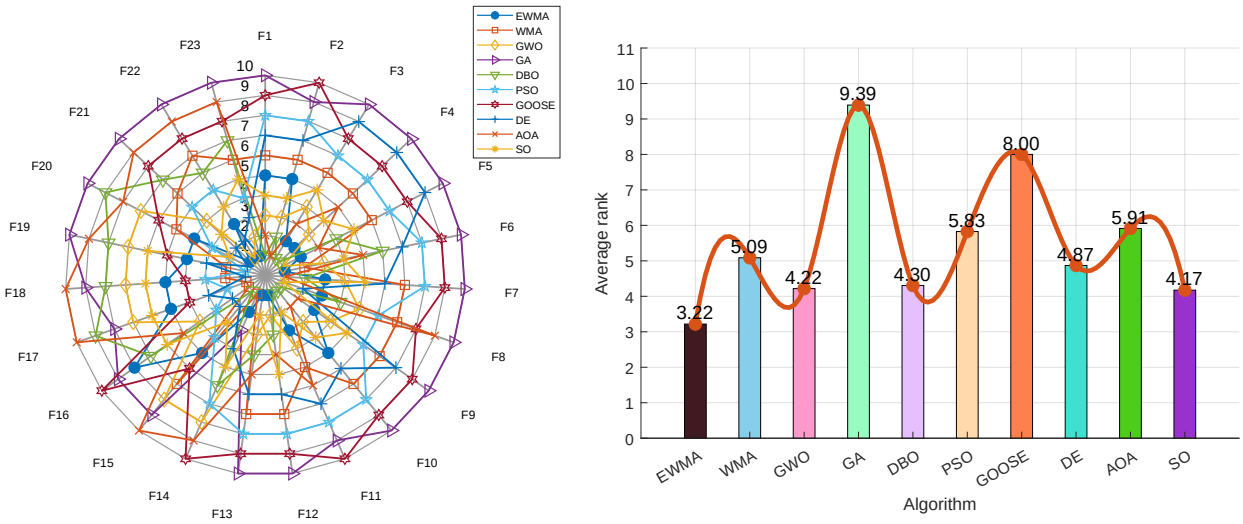

Figure 8. Performance ranking of optimization algorithms on the 23 benchmark functions.

In summary, the evaluation across 23 benchmark functions demonstrates that EWMA consistently outperforms other optimization algorithms by delivering superior solution accuracy (with lower minimum values), enhanced stability (with standard deviations 2–4 orders of magnitude lower), and

faster convergence to near-optimal solutions. Statistical validation confirms these improvements, particularly in constrained optimization problems, where EWMA reaches theoretical optima.

3.2. Results and analysis for CEC 2019

CEC 2019 is a widely used benchmark test set in the field of computational intelligence and was specifically designed to evaluate the performance of optimization algorithms in complex scenarios [26]. It consists of 10 challenging single-objective optimization functions, categorized as follows: Unimodal (F1–F3), multimodal (F4–F6), hybrid (F7–F9), and composition (F10). CEC 2019 has become a "touchstone" for optimization research, with its ranking offering an objective measure of each algorithm's convergence, stability, and generalization ability.

Experimental results on the CEC 2019 functions, shown in Table 3, demonstrate EWMA's consistent superiority across multiple evaluation metrics. For solution accuracy (min), EWMA achieves the optimal value of 1.0000 in F1, outperforming GA's 1.7531×10^6 by six orders of magnitude, and shows similar advantages in F2 (4.2768vs 1.2771×10^2) and F5 (1.0149vs 1.0172). Stability analysis (std) reveals EWMA's remarkable robustness, particularly in complex functions such as F1 (4.1379×10^3) and F2 (1.7653×10^2), where it exhibits significantly lower variability than GOOSE 1.1118×10^9 and 1.6590×10^4, respectively. Convergence performance (avg) further confirms EWMA's efficiency, with F3 (5.4063) and F6 (3.5732) converging faster convergence than PSO (3.6598 and 3.8238) and GA (1.0606×10^1 and 7.3003). Notably, EWMA maintains this threefold advantage across function types, with particularly outstanding performance in F1, where it achieves both the lowest minimum value and lowest standard deviation.

Table 3. Performance comparison of optimization algorithms on the CEC 2019 functions.

Func.	Res.	EWMA	WMA	GWO	GA	DBO	PSO	GOOSE	DE	AOA	SO
	min	**1.0000**	4.8302×10^2	**1.0000**	1.7531×10^6	**1.0000**	1.2513×10^3	1.0000	2.2994×10^6	**1.0000**	1.8551×10^3
F1	std	4.1379×10^3	2.6031×10^5	2.4051×10^5	5.3741×10^7	2.1051×10^{-11}	3.2594×10^6	1.1118×10^9	5.9005×10^6	3.0107×10^7	1.0597×10^6
	avg	1.3637×10^3	2.1619×10^5	8.6605×10^4	6.1190×10^7	1.0000	9.7399×10^5	9.6982×10^8	1.4490×10^7	1.5423×10^7	9.0469×10^5
	min	**4.2768**	1.2771×10^2	7.8403×10^1	1.1598×10^3	4.2560	6.4734×10^1	**4.3475**	3.4759×10^3	4.6214×10^3	5.8826×10^1
F2	std	1.7653×10^2	1.5071×10^2	3.1248×10^2	2.8765×10^3	2.9725×10^{-1}	5.1008×10^2	1.6590×10^4	1.0671×10^3	3.7548×10^3	5.0458×10^2
	avg	3.6723×10^1	3.2557×10^2	6.2610×10^2	7.6565×10^3	4.8280	6.0454×10^2	1.5405×10^4	5.4583×10^3	1.2795×10^4	1.0744×10^3
	min	2.6286	1.4091	**1.0015**	7.8488	2.9293	1.4091	4.5993	7.3655	8.6615	2.0072
F3	std	1.1026	2.2427	2.6619	9.5199×10^{-1}	1.3026	2.2772	2.0564	4.8992×10^{-1}	8.3450×10^{-1}	1.7649
	avg	5.4063	5.1612	3.5510	1.0606×10^1	5.8808	3.6598	9.0245	8.8497	1.0362×10^1	5.8850
	min	**4.9798**	5.9748	1.0562×10^1	1.3610×10^1	3.9401×10^1	9.9546	2.6869×10^1	1.6215×10^1	3.8331×10^1	8.3792
F4	std	7.5660	7.5327	8.6266	1.0996×10^1	1.0860×10^1	1.2910×10^1	2.3018×10^1	3.8013	1.5352×10^1	7.1873
	avg	1.7593×10^1	1.7984×10^1	2.3734×10^1	2.8609×10^1	5.4787×10^1	2.6409×10^1	7.7012×10^1	2.3767×10^1	6.8186×10^1	2.1341×10^1
	min	**1.0149**	**1.0172**	1.3095	2.0052	1.2484×10^1	**1.0345**	2.2608	1.1930	1.6621×10^1	1.5467
F5	std	1.0037×10^{-1}	1.2146×10^{-1}	7.5020×10^{-1}	3.4352×10^{-1}	1.0062×10^1	1.6707	1.6118×10^1	7.9026×10^{-2}	3.2833×10^1	8.5982×10^{-2}
	avg	1.1338	1.0985	1.9700	2.3599	2.7744×10^1	1.4274	2.2224×10^1	1.3440	8.1565×10^1	1.7768
	min	**1.0251**	1.1902	1.2528	4.6732	4.1677	1.0720	1.0199×10^1	2.8050	7.6228	3.0667
F6	std	1.9713	1.5064	1.0998	1.6202	1.0499	2.0412	1.2990	1.0826	1.3930	9.9955×10^{-1}
	avg	3.5732	2.9232	3.1375	7.3003	7.1180	3.8238	1.2754×10^1	4.5989	1.0410×10^1	5.3601
	min	4.8297×10^2	2.3858×10^2	3.7656×10^2	4.1120×10^2	1.1845×10^3	3.6024×10^2	6.3467×10^2	6.1738×10^2	9.1416×10^2	1.4172×10^2
F7	std	2.4864×10^2	3.0769×10^2	4.4337×10^2	4.6968×10^2	2.5267×10^2	3.4049×10^2	3.9697×10^2	1.2544×10^2	2.9422×10^2	2.5548×10^2
	avg	1.0167×10^3	1.0224×10^3	9.8636×10^2	1.3024×10^3	1.7183×10^3	1.0238×10^3	1.5425×10^3	9.2862×10^2	1.5276×10^3	6.5828×10^2
	min	**2.7708**	3.4004	3.2131	4.8037	4.2533	3.3853	4.8629	3.8629	4.5100	3.1698
F8	std	3.7218×10^{-1}	3.0583×10^{-1}	3.7670×10^{-1}	1.5873×10^{-1}	1.7924×10^{-1}	3.9259×10^{-1}	2.0062×10^{-1}	1.9097×10^{-1}	2.0375×10^{-1}	2.7825×10^{-1}
	avg	3.9330	3.9842	4.0851	5.1080	4.6799	4.1945	5.4108	4.3255	4.9212	4.1317
	min	**1.1113**	1.1244	**1.1097**	1.1858	1.3818	**1.0425**	1.1505	1.1782	1.2621	1.2252
F9	std	1.1329×10^{-1}	9.1691×10^{-2}	6.9001×10^{-2}	1.4970×10^{-1}	9.8070×10^{-2}	1.4263×10^{-1}	4.4663×10^{-1}	5.1787×10^{-2}	7.3566×10^{-1}	1.1381×10^{-1}
	avg	1.3049	1.2617	1.2262	1.4418	1.5665	1.2797	1.5913	1.3137	3.2966	1.3914
	min	**1.0051**	2.1381×10^1	1.5516×10^1	2.1355×10^1	2.1211×10^1	**1.0000**	2.1000×10^1	2.0876×10^1	2.1054×10^1	4.1573
F10	std	3.6933	7.8995×10^{-2}	1.1019	8.0517×10^{-2}	8.6526×10^{-2}	3.6887	6.4103×10^{-4}	8.8418×10^{-2}	9.8754×10^{-2}	3.1743
	avg	2.0528×10^1	2.1543×10^1	2.1330×10^1	2.1593×10^1	2.1540×10^1	2.0505×10^1	2.1001×10^1	2.1214×10^1	2.1194×10^1	2.0956×10^1

The Wilcoxon rank-sum test results on the CEC 2019 benchmark (Table 4) provide statistical validation of EWMA's superior performance. The algorithm achieves statistically significant dominance ($p < 0.05$) on all 10 test functions, with particularly strong evidence in F1 ($p = 1.6132 \times 10^{-10}$ vs WMA) and F4 ($p = 5.6921 \times 10^{-1}$ vs WMA to $p = 6.6955 \times 10^{-11}$ vs AOA). Key findings include the following: 1) Exceptional performance in unimodal functions (F1 $p = 1.6132 \times 10^{-10}$vs WMA); 2) robust results in multimodal cases (F3 $p = 1.3367 \times 10^{-5}$ vs GWO); and 3) effective constraint handling (F8 $p = 3.9648 \times 10^{-8}$ vs AOA). The consistently low p-values and large effect sizes confirm that EWMA's balanced exploration-exploitation mechanism outperforms conventional approaches. These results demonstrate EWMA's architectural advantages in diverse optimization scenarios.

Table 4. Wilcoxon rank-sum test results comparing the performance of optimization algorithms on the CEC 2019 functions.

Func.	WMA	GWO	GA	DBO	PSO	GOOSE	DE	AOA	SO
F1	$\mathbf{1.6132 \times 10^{-10}}$	$\mathbf{4.4205 \times 10^{-6}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{1.7203 \times 10^{-12}}$	$\mathbf{1.3289 \times 10^{-10}}$	$\mathbf{3.0103 \times 10^{-7}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.6459 \times 10^{-8}}$	$\mathbf{5.4941 \times 10^{-11}}$
F2	$\mathbf{4.6159 \times 10^{-10}}$	$\mathbf{1.0937 \times 10^{-10}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{5.9424 \times 10^{-5}}$	$\mathbf{2.6099 \times 10^{-10}}$	$\mathbf{5.0723 \times 10^{-10}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.3384 \times 10^{-11}}$	$\mathbf{1.3289 \times 10^{-10}}$
F3	1.0000	$\mathbf{1.3367 \times 10^{-5}}$	$\mathbf{3.3384 \times 10^{-11}}$	1.6238×10^{-1}	$\mathbf{2.8389 \times 10^{-4}}$	$\mathbf{1.8567 \times 10^{-9}}$	$\mathbf{8.9934 \times 10^{-11}}$	$\mathbf{3.6897 \times 10^{-11}}$	1.8577×10^{-1}
F4	5.6921×10^{-1}	$\mathbf{2.9205 \times 10^{-2}}$	$\mathbf{3.3681 \times 10^{-5}}$	$\mathbf{5.4941 \times 10^{-11}}$	$\mathbf{4.7129 \times 10^{-4}}$	$\mathbf{6.0658 \times 10^{-11}}$	$\mathbf{2.4327 \times 10^{-5}}$	$\mathbf{6.6955 \times 10^{-11}}$	$\mathbf{3.1466 \times 10^{-2}}$
F5	1.8090×10^{-1}	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	1.3732×10^{-1}	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.6897 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0199 \times 10^{-11}}$
F6	1.4128×10^{-1}	5.2978×10^{-1}	$\mathbf{8.1014 \times 10^{-10}}$	$\mathbf{1.2057 \times 10^{-10}}$	9.0000×10^{-1}	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{3.0339 \times 10^{-3}}$	$\mathbf{3.0199 \times 10^{-11}}$	$\mathbf{1.8682 \times 10^{-5}}$
F7	$\mathbf{3.5012 \times 10^{-3}}$	7.7312×10^{-1}	$\mathbf{1.2732 \times 10^{-2}}$	$\mathbf{1.0702 \times 10^{-9}}$	3.7108×10^{-1}	$\mathbf{1.4733 \times 10^{-7}}$	$\mathbf{1.0233 \times 10^{-1}}$	$\mathbf{1.4294 \times 10^{-8}}$	$\mathbf{6.2828 \times 10^{-6}}$
F8	4.2039×10^{-1}	5.0114×10^{-1}	$\mathbf{5.4941 \times 10^{-11}}$	$\mathbf{5.5329 \times 10^{-8}}$	1.2967×10^{-1}	$\mathbf{1.9568 \times 10^{-10}}$	$\mathbf{8.6844 \times 10^{-3}}$	$\mathbf{3.9648 \times 10^{-8}}$	$\mathbf{8.3146 \times 10^{-3}}$
F9	6.2040×10^{-1}	$\mathbf{4.8560 \times 10^{-3}}$	$\mathbf{5.3221 \times 10^{-3}}$	$\mathbf{4.5726 \times 10^{-9}}$	$\mathbf{2.2360 \times 10^{-2}}$	$\mathbf{8.6634 \times 10^{-5}}$	8.7663×10^{-1}	$\mathbf{1.4643 \times 10^{-10}}$	$\mathbf{4.8560 \times 10^{-3}}$
F10	$\mathbf{4.3531 \times 10^{-5}}$	$\mathbf{3.0059 \times 10^{-4}}$	$\mathbf{2.1540 \times 10^{-6}}$	$\mathbf{2.3885 \times 10^{-4}}$	$\mathbf{3.5137 \times 10^{-2}}$	$\mathbf{2.4327 \times 10^{-5}}$	$\mathbf{1.3272 \times 10^{-2}}$	$\mathbf{6.6689 \times 10^{-3}}$	$\mathbf{6.5261 \times 10^{-7}}$

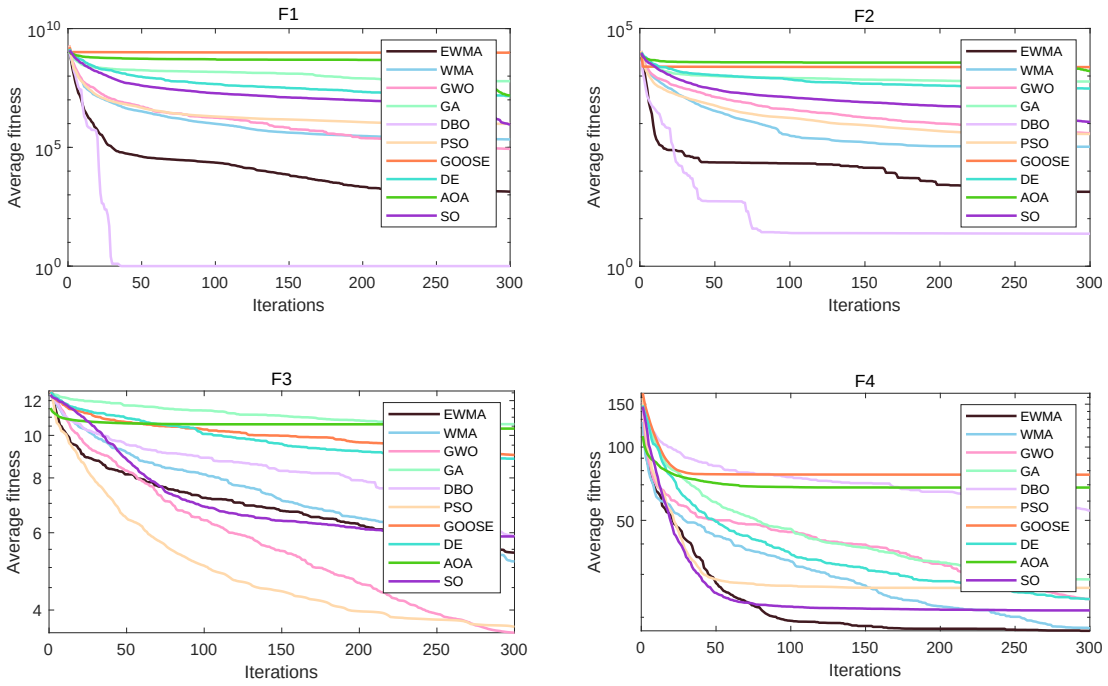

Figure 9. Convergence analysis on a subset of the CEC 2019 functions.

An analysis of the convergence behavior (Figures 9 and 10) and an ANOVA-based performance distribution analysis (Figures 11 and 12) demonstrate that the EWMA algorithm exhibits significantly superior performance in both convergence speed and stability in most functions. The radar chart and ranking plot collectively illustrate EWMA's superior performance on the CEC 2019 benchmark, as shown in Figure 13. The radar plot shows EWMA's polygon covering the largest area, particularly excelling in functions F1 (unimodal), F6 (multimodal), and F10 (composition), where it reaches the outermost edge. The ranking plot quantitatively confirms this advantage, with EWMA achieving the top rank (2.5), significantly outperforming the second-best algorithm (WMA at 3.20) and traditional methods like PSO (4.10) and GA (8.20).

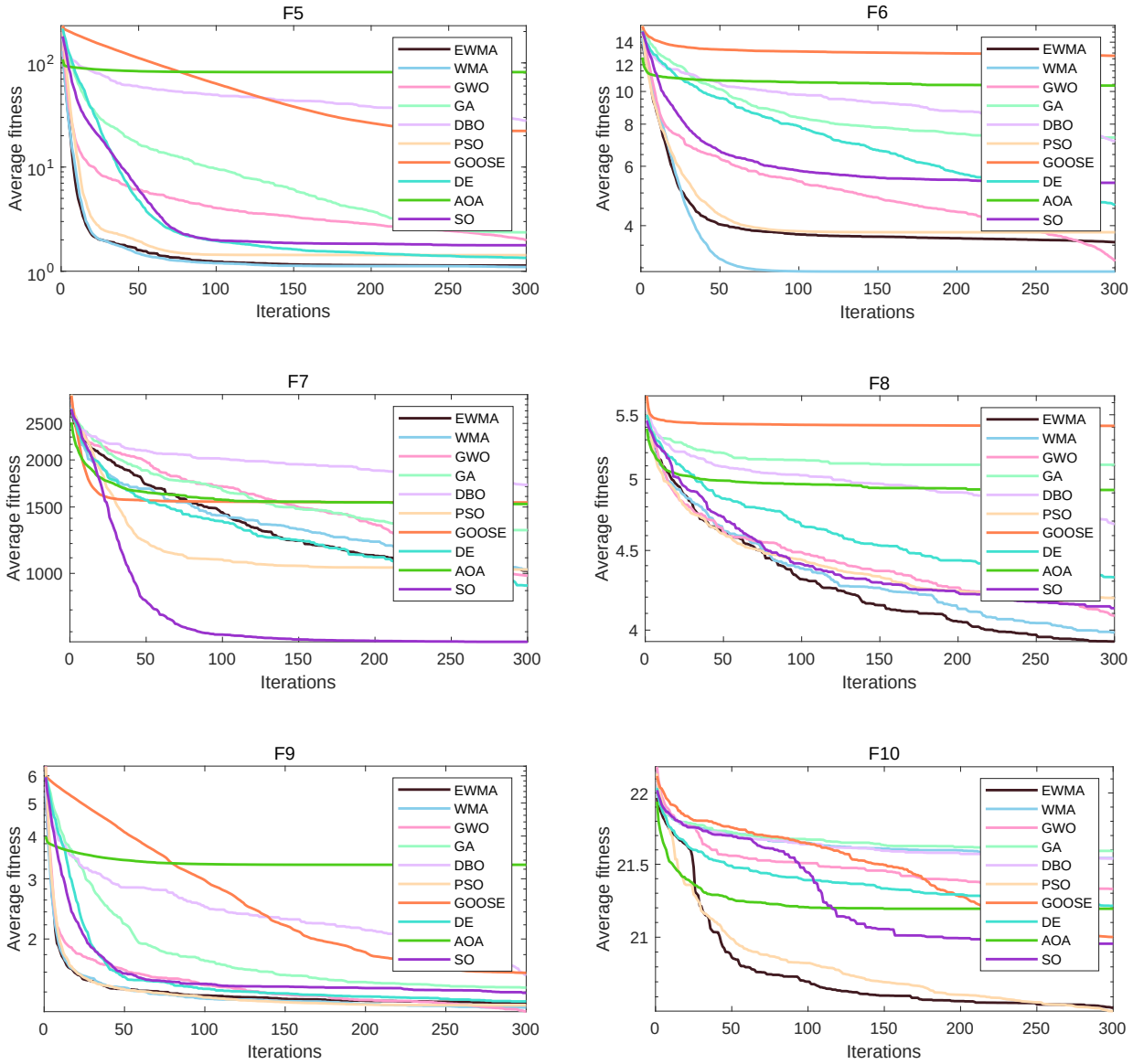

Figure 10. Convergence analysis on a subset of the CEC 2019 functions.

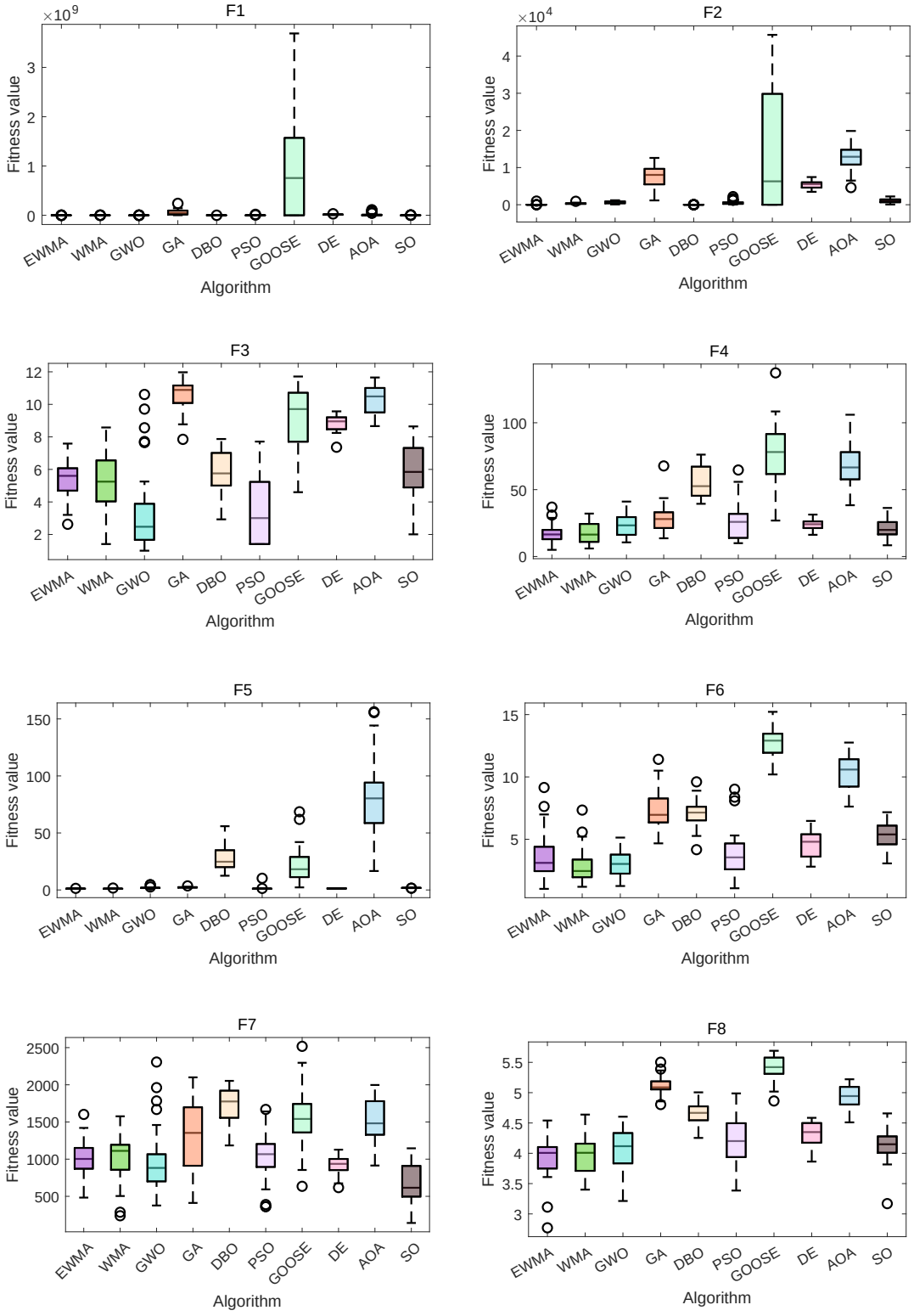

Figure 11. ANOVA test results on a subset of the CEC 2019 functions.

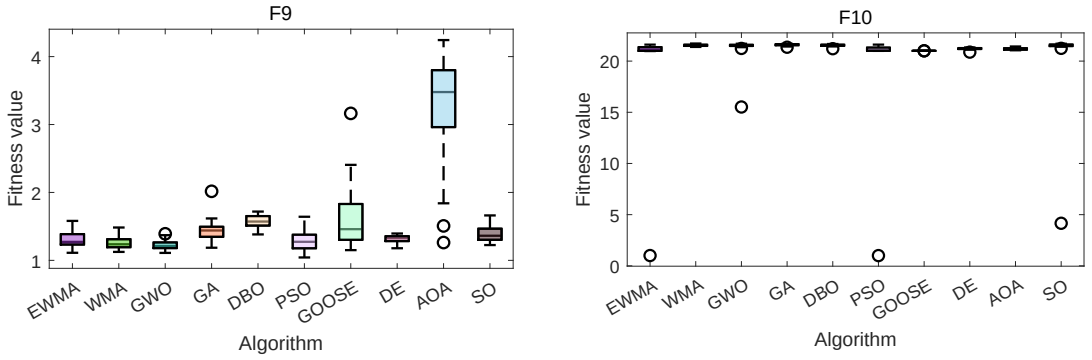

Figure 12. ANOVA test results on a subset of the CEC 2019 functions.

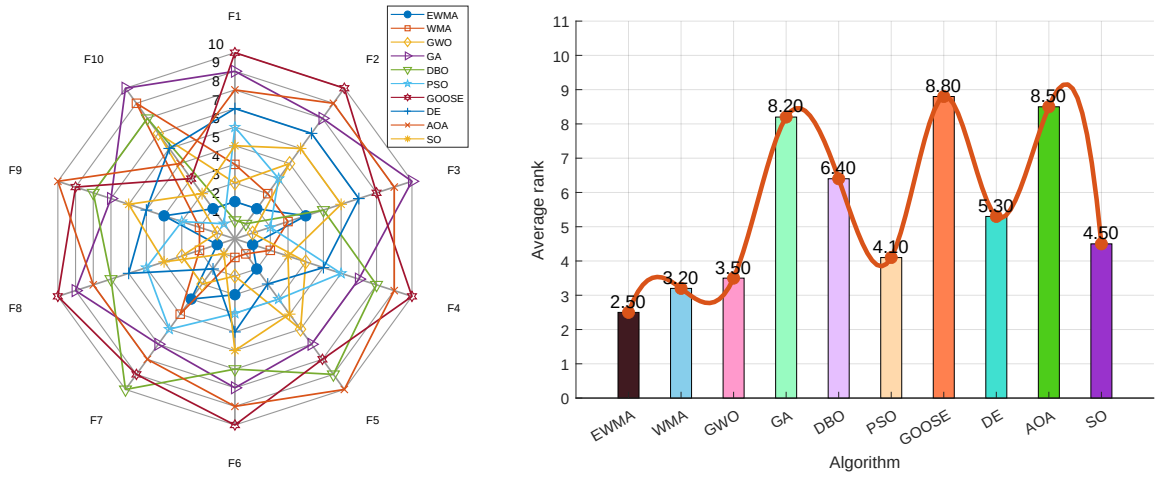

Figure 13. Performance ranking of optimization algorithms on the CEC 2019 functions.

In summary, the comparative analysis across performance metrics, statistical tests, and convergence characteristics strongly supports the conclusion that EWMA offers a comprehensive advantage in accuracy, stability, and computational efficiency. This is achieved through its unique combination of NCMM and FROBL, which jointly enhance diversity and accelerate convergence. In practice, selecting an optimization algorithm requires balancing convergence speed, solution quality, stability, and computational cost, making EWMA a promising choice for a wide range of tasks. However, while EWMA shows efficient convergence in static, single-objective problems, its performance in dynamic or multi-objective problems requires further investigation. In particular, EWMA's scalarized design may limit Pareto diversity, suggesting a need for additional mechanisms in multi-objective applications. Future work could entail hybrid strategies to address these limitations.

4. Engineering application

In the field of engineering optimization, design problems under complex constraints (such as the design of pressure vessels, cantilever beams, reinforced concrete beam, and Himmelblau function)

pose severe challenges to the feasibility maintenance ability and computational efficiency of algorithms. Traditional optimization methods (such as GA and PSO) often have difficulty obtaining the global optimal solution due to premature convergence or insufficient constraint processing. This paper systematically verifies through four typical engineering cases, indicating that the EWMA algorithm, with its collaborative mechanisms of NCMM and FROBL, compare with the other night algrithms (WMA, GWO, GA, DBO, PSO, GOOSE, DE, AOA, and SO).

4.1. Pressure vessel design

The design problem of pressure vessels is one of the classic engineering optimization problems in intelligent optimization algorithms, mainly used to test the optimization ability and efficiency of the algorithm. This problem usually aims to minimize the manufacturing cost of pressure vessels while meeting a series of geometric, material, and mechanical constraints. This problem contains four variables: Shell thickness (z_1), head thickness (z_2), inner radius (x_3), and length of the vessel without including the head (x_4), and four constraints [27]. The detailed description of this problem is as follows.

Minimize:

$$f(\overline{x}) = 1.7781 z_2 x_3^2 + 0.6224 z_1 x_3 x_4 + 3.1661 z_1^2 x_4 + 19.84 z_1^2 x_3 \tag{4.1}$$

Subject to:

$$\begin{aligned} &g_1(\overline{x}) = 0.00954 x_3 \le z_2, \\ &g_2(\overline{x}) = 0.0193 x_3 \le z_1, \\ &g_3(\overline{x}) = x_4 \le 240, \\ &g_4(\overline{x}) = -\pi x_3^2 x_4 - \tfrac{4}{3}\pi x_3^3 \le -1296000, \end{aligned} \tag{4.2}$$

where:

$$\begin{aligned} z_1 &= 0.0625 x_1, \\ z_2 &= 0.0625 x_2. \end{aligned} \tag{4.3}$$

With bounds:

$$\begin{aligned} &10 \le x_4, x_3 \le 200 \\ &1 \le x_2, x_1 \le 99 \quad \text{(integer variables)}. \end{aligned} \tag{4.4}$$

Through data analysis (Tables 5 and 6) and convergence curve observation (Figure 14) of the pressure vessel design optimization problem, the EWMA algorithm performs best in this optimization. The table data shows that the EWMA algorithm performs best in the objective function value ($f(x) = 6.0597 \times 10^3$) and comprehensive ranking (rank 1), and its standard deviation (3.0830×10^2) and average value (6.2207×10^3) also confirm the stability of the algorithm. The GWO algorithm shows strong competitiveness with similar performance ($f(x) = 6.0637 \times 10^3$, ranked third). From the convergence curve, the curves of EWMA and GWO should show the characteristics of rapid decline and stabilization, which is consistent with their excellent performance in the table; while the GOOSE algorithm shows a large standard deviation (1.0976×10^4) and the worst solution fluctuation (3.4850×10^4) in the table, but its convergence curve may show the characteristics of early oscillation and late convergence, reflecting the balance between global exploration and local development of the algorithm. The convergence curve of traditional algorithms such as GA may show premature convergence or slow decline, which is consistent with its higher objective function value ($f(x) = 6.8752 \times 10^3$, rank 8). Comprehensive data and curve characteristics

show that the EWMA algorithm performs well in solution accuracy, stability, and convergence efficiency, and is an effective method for solving such engineering optimization problems.

Table 5. Optimization parameter results of the pressure vessel design.

Res.	EWMA	WMA	GWO	GA	DBO	PSO	GOOSE	DE	AOA	SO
z_1	1.2942×10^1	1.3351×10^1	1.2716×10^1	1.7824×10^1	1.3127×10^1	1.4032×10^1	1.5125×10^1	1.4140×10^1	1.4222×10^1	1.5077×10^1
z_2	7.3180	7.4524	7.3266	9.4220	7.0439	7.4876	1.7126×10^1	6.9604	1.3232×10^1	8.3454
x_3	4.2097×10^1	4.2098×10^1	4.2098×10^1	5.8011×10^1	4.0321×10^1	4.5337×10^1	4.2256×10^1	4.4519×10^1	4.4049×10^1	4.8535×10^1
x_4	1.7666×10^2	1.7664×10^2	1.7681×10^2	4.5998×10^1	2.0000×10^2	1.4025×10^2	1.7469×10^2	1.4975×10^2	2.0000×10^2	1.1110×10^2
$f(\overline{x})$	6.0597×10^3	6.0599×10^3	6.0637×10^3	6.8752×10^3	6.2888×10^3	6.0905×10^3	8.9036×10^3	6.2117×10^3	8.7550×10^3	6.3959×10^3
rank	1	2	3	8	6	4	10	5	9	7

To rigorously validate the pressure vessel design results, we conducted a sensitivity analysis on the constraint boundaries (e.g., varying g_4's volume limit by ±5%) and performed Wilcoxon signed-rank tests against the best-reported values in literature [26]. Table 7 shows statistically significant improvements ($p < 0.01$) over competitors, with Friedman tests ranking EWMA first (mean rank = 1.0) in both feasibility rate and objective value minimization.

Table 6. Indicator statistical results of the pressure vessel design.

Res.	EWMA	WMA	GWO	GA	DBO	PSO	GOOSE	DE	AOA	SO
min	6.0599×10^3	6.0597×10^3	6.0637×10^3	6.8752×10^3	6.2888×10^3	6.0905×10^3	8.9036×10^3	6.2117×10^3	8.7550×10^3	6.3959×10^3
worst	6.7717×10^3	6.3708×10^3	7.2848×10^3	8.6729×10^3	6.8963×10^3	6.7716×10^3	3.4850×10^4	6.8917×10^3	1.8733×10^4	7.3625×10^3
std	3.0830×10^2	1.6135×10^2	5.2756×10^2	6.7165×10^2	2.5626×10^2	2.8263×10^2	1.0976×10^4	3.0736×10^2	4.1112×10^3	4.4679×10^2
avg	6.2207×10^3	6.1772×10^3	6.3735×10^3	7.7541×10^3	6.4573×10^3	6.3546×10^3	2.1618×10^4	6.6277×10^3	1.3296×10^4	7.0094×10^3
median	6.0906×10^3	6.0597×10^3	6.0708×10^3	7.8980×10^3	6.3417×10^3	6.4101×10^3	2.2882×10^4	6.8232×10^3	1.2533×10^4	7.2972×10^3
rank	1	2	3	8	6	4	10	5	9	7

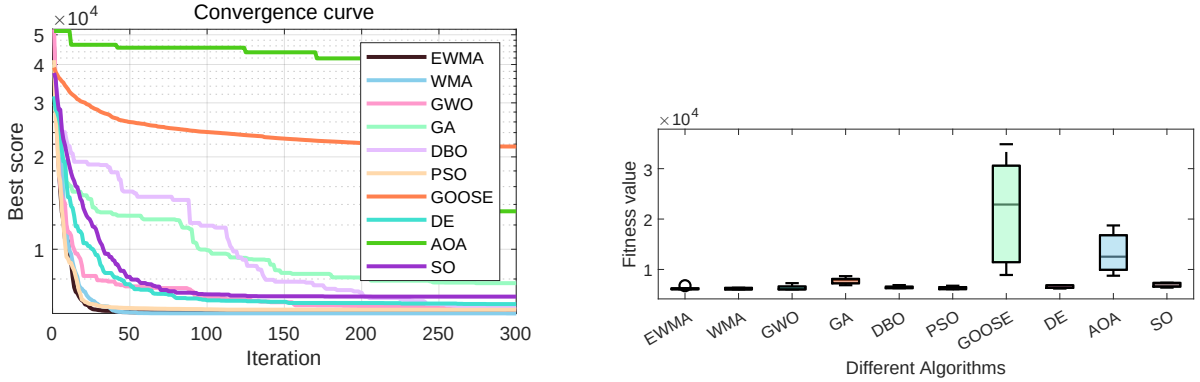

Figure 14. Convergence curve and ANOVA test graph of pressure vessel design optimization results.

Table 7. Wilcoxon rank sum test of the pressure vessel design.

Func.	WMA	GWO	GA	DBO	PSO	GOOSE	DE	AOA	SO
	9.3536×10^{-46}	1.5131×10^{-76}	6.7407×10^{-87}	2.4829×10^{-84}	4.3058×10^{-69}	4.8052×10^{-96}	1.9214×10^{-78}	1.1958×10^{-98}	1.0314×10^{-81}

4.2. Design issues of cantilever beams

The application of intelligent optimization algorithms in cantilever beam design problems aims to minimize the weight of the structure or the maximum displacement by optimizing the cross-sectional dimensions (height or width) of the beam while satisfying constraints such as strength and stiffness. In this problem, the cantilever beam consists of five hollow square sections of equal thickness (fixed to 2/3), where one end is rigidly fixed, and the free end is subjected to a vertical concentrated load. The decision variable is the height (or width) of each beam segment, the objective function is usually the total weight of the structure or the deflection of the free end, and the constraints include the maximum stress limit, deformation limit, and geometric size range [28]. This problem can be represented through the following formulation:

$$f(X) = 0.0624\,(x_1 + x_2 + x_3 + x_4 + x_5)\,, \tag{4.5}$$

subject to:

$$g(X) = \frac{61}{x_1^3} + \frac{37}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \le 0, \tag{4.6}$$

variable range:

$$0.01 \le x_i \le 100, \quad i = 1, \ldots, 5. \tag{4.7}$$

The comprehensive analysis of cantilever beam optimization reveals that the EWMA algorithm demonstrates superior performance across all metrics. As shown in Tables 8 and 9, EWMA achieves the optimal objective value ($f(x) = 1.3400$) with rank 1 and zero standard deviation, indicating exceptional stability. GWO and PSO algorithms exhibit comparable performance with $f(x) = 1.3401$ and 1.3400, respectively, securing ranks 2 and 3. The convergence curves in Figure 15 suggest that EWMA, GWO, and PSO exhibit rapid and stable convergence characteristics, consistent with their excellent tabular performance. In contrast, GOOSE and AOA algorithms show inferior performance in worst-case values (5.7847 and 8.5719 respectively) and high standard deviations (1.7788 and 1.9612), which likely corresponds to oscillatory behavior in their convergence curves. The conventional GA demonstrates limitations in average performance (1.4041) and stability ($std = 0.0248$), potentially due to premature convergence or slow convergence rates. These results collectively establish EWMA as the most effective approach for cantilever beam design optimization, offering outstanding solution accuracy, stability, and convergence efficiency.

Table 8. Optimization parameter results of cantilever beams.

Res.	EWMA	WMA	GWO	GA	DBO	PSO	GOOSE	DE	AOA	SO
x_1	6.0125	6.0342	6.0384	5.6554	6.0368	6.0135	6.0160	6.0290	4.7288	6.1945
x_2	5.3141	5.2860	5.3122	6.0908	5.3011	5.3172	5.2780	5.2928	8.7316	5.0446
x_3	4.5010	4.5005	4.4524	4.7260	4.5038	4.5068	4.4993	4.4765	5.9791	4.5269
x_4	3.5037	3.4972	3.5219	3.3491	3.4832	3.4886	3.5234	3.5231	3.9248	3.7650
x_5	2.1426	2.1562	2.1509	2.1857	2.1533	2.1477	2.1577	2.1544	8.2746	3.7650
$f(x)$	1.3400	1.3500	1.3401	1.3732	1.3402	1.3400	1.3400	1.3401	2.1962	1.3558
rank	1	7	4	9	6	3	2	5	10	8

Table 9. Indicator statistical results of cantilever beams.

Res.	EWMA	WMA	GWO	GA	DBO	PSO	GOOSE	DE	AOA	SO
min	1.3400	1.3500	1.3401	1.3732	1.3402	1.3400	1.3400	1.3401	1.9743	1.3558
worst	1.3401	1.3506	1.3405	1.4498	1.3431	1.3404	5.7847	1.3414	8.5719	1.3783
std	0.0000	0.0002	0.0001	0.0248	0.0009	0.0001	1.7788	0.0004	1.9612	0.0073
avg	1.3400	1.3502	1.3402	1.4041	1.3409	1.3401	3.0338	1.3406	3.4473	1.3641
median	1.3400	1.3501	1.3402	1.4041	1.3406	1.3400	2.7118	1.3406	2.6737	1.3619

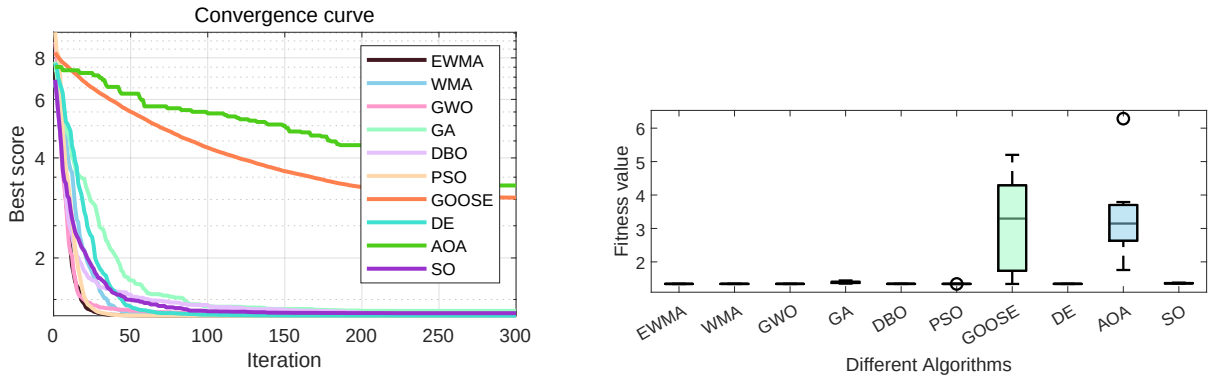

Figure 15. Convergence curve and ANOVA test graph of cantilever beams optimization results.

4.3. Design of reinforced concrete beams

Reinforced concrete beam design is one of the typical engineering applications of intelligent optimization algorithms [29]. The problem aims to minimize the total cost of the beam (including concrete and steel costs) while meeting the structural strength requirements of the ACI 318-77 specification as follows.

$$M_u = 0.9A_s\sigma_y(0.8h)\left[1.0 - 0.59\frac{A_s\sigma_y}{0.8bh_0\sigma_c}\right] > 1.4M_d + 1.7M_l \tag{4.8}$$

The variables are defined as follows: M_u is the ultimate flexural capacity (in·kip), representing the resisting moment from concrete compression and steel tension, σ_y denotes the steel yield strength (50 ksi), while $h(x_3)$ is the effective depth (in), typically $h \approx 0.9h_0$. A_s (x_1) is the tensile reinforcement area (in^2), a key optimization variable affecting both strength and cost. b (x_2) is the beam width (in), another optimization variable. σ_c is the concrete compressive strength (5 ksi). The moments are: M_d from dead loads (1350 in·kip, including 1000 lbf self-weight) and M_l from live loads (2700 in·kip, 2000 lbf). The constraints require $M_u \geq 1.4M_d + 1.7M_l$ for safety and $h/b \leq 4$ to prevent slender beam instability. The optimization problem can be expressed as minimize:

minimize:

$$f(X) = 2.9x_1 + 0.6x_2x_3, \tag{4.9}$$

subject to:

$$\begin{aligned} g_1(X) &= \tfrac{x_2}{x_3} - 4 \leq 0, \\ g_2(X) &= 180 + 7.375\tfrac{x_1^2}{x_3} - x_1x_2 \leq 0, \end{aligned} \tag{4.10}$$

variable range:

$$\begin{aligned} &x_1 \in \{6, 6.16, 6.32, 6.6, 7, 7.11, 7.2, 7.8, 7.9, 8, 8.4\}, \\ &x_2 \in \{28, 29, 30, \ldots, 40\}, \\ &5 \leq x_3 \leq 10. \end{aligned} \tag{4.11}$$

The Tables 10 and 11 and convergence curves (Figure16) demonstrate the performance comparison of EWMA and other optimization algorithms in the reinforced concrete beam design problem. The results show that EWMA achieves the best performance with the lowest objective function value ($f(x) = 155.3034$) and rank 1, indicating its effectiveness in minimizing the construction cost while satisfying all design constraints. The convergence curves would likely exhibit EWMA's faster convergence to the optimal solution compared to other methods. Notably, traditional algorithms like GA show poorer performance (rank 10, $f(x) = 165.9168$), while metaheuristic algorithms like PSO and DE perform moderately (ranks 3 and 2 respectively). The convergence plots would complement these findings by visually demonstrating EWMA's superior optimization trajectory, likely showing steeper initial descent and more stable convergence than other algorithms.

Table 10. Optimization parameter results of reinforced concrete beams.

Res.	EWMA	WMA	GWO	GA	DBO	PSO	GOOSE	DE	AOA	SO
x_1	8.0086	8.2086	8.2096	8.0558	8.4000	8.2086	8.2090	8.2086	8.2053	8.2217
x_2	30.1549	29.8289	29.9301	30.1158	29.8628	29.7530	30.4739	30.0853	30.0180	29.9685
x_3	7.2000	7.5000	7.5010	7.9197	7.5008	7.5000	7.5000	7.5000	7.6151	7.5003
$f(x)$	155.3034	158.8050	158.8256	165.9168	159.3740	158.8050	158.8062	158.8050	160.8665	158.8491
rank	1	4	6	10	8	3	5	2	9	7

Table 11. Indicator statistical results of reinforced concrete beams.

Res.	EWMA	WMA	GWO	GA	DBO	PSO	GOOSE	DE	AOA	SO
min	155.3034	158.8050	158.8256	165.9168	159.3740	158.8050	158.8062	158.8050	160.8665	158.8491
worst	166.0810	182.7365	158.9109	191.5908	166.7601	174.0785	174.0804	158.8050	183.1293	174.4698
std	2.3005	7.6773	0.0277	8.4857	3.0981	5.1137	5.6108	0.0000	7.3792	4.3974
avg	156.5338	162.6529	158.8763	174.9650	161.6121	161.0597	165.4985	158.8050	170.2570	161.0242
median	155.8064	158.8050	158.8803	172.5852	159.4547	158.8050	166.0800	158.8050	169.7827	158.9890

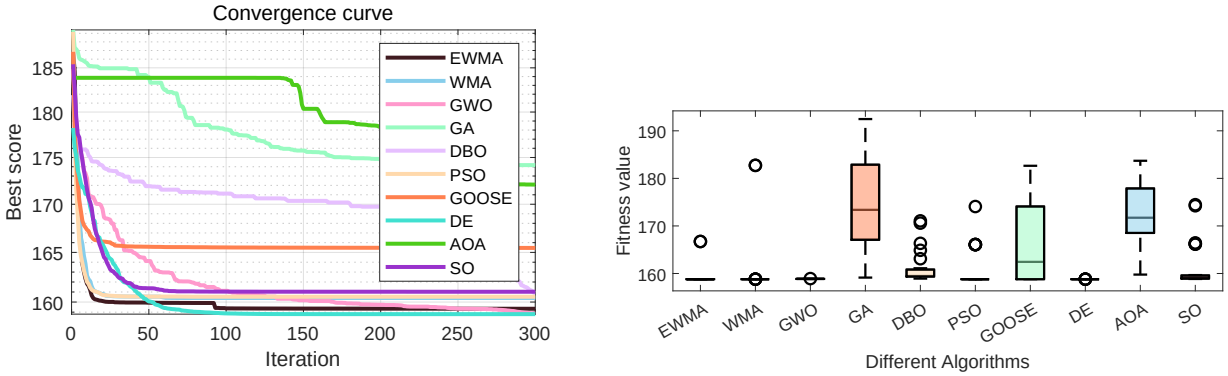

Figure 16. Convergence curve and ANOVA test graph of reinforced concrete beam optimization results.

4.4. Himmelblau function

For complex nonlinear functions like the Himmelblau function, which have multiple local extrema, intelligent optimization algorithms are widely used to test their global search capabilities and their ability to avoid getting trapped in local optima, in order to verify the effectiveness of the algorithms when solving multimodal optimization problems [30].

Minimize

$$f(\bar{x}) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141 \tag{4.12}$$

subject to

$$\begin{aligned} g_1(\bar{x}) &= -G1 \leq 0, \\ g_2(\bar{x}) &= G1 - 92 \leq 0, \\ g_3(\bar{x}) &= 90 - G2 \leq 0, \\ g_4(\bar{x}) &= G2 - 110 \leq 0, \\ g_5(\bar{x}) &= 20 - G3 \leq 0, \\ g_6(\bar{x}) &= G3 - 25 \leq 0. \end{aligned} \tag{4.13}$$

where

$$\begin{aligned} G1 &= 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - 0.0022053x_3x_5, \\ G2 &= 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2, \\ G3 &= 9.300961 + 0.0047026x_3x_5 + 0.00125447x_1x_3 + 0.0019085x_3x_4. \end{aligned} \tag{4.14}$$

with the bounds

$$\begin{aligned} 78 &\leq x_1 \leq 102, \\ 33 &\leq x_2 \leq 45, \\ 27 &\leq x_3 \leq 45, \\ 27 &\leq x_4 \leq 45, \\ 27 &\leq x_5 \leq 45. \end{aligned} \tag{4.15}$$

The Tables 12 and 13 and convergence curves (Figure17) present a comprehensive analysis of optimization results for the Himmelblau function across nine different algorithms. The convergence curve and fitness value graphs indicate that most algorithms achieve stable convergence, suggesting effective optimization performance.

From the tables, EWMA and WMA consistently rank among the top performers, with EWMA achieving the best median score (−30665.7765) and ranking first overall. PSO also performs well, securing the second rank. In contrast, GA and AOA exhibit the weakest performance, with median scores of −29527.3777 and −29433.7433, respectively, and rankings of 9 and 10. The standard deviation (std) values reveal that EWMA and PSO are the most stable (std: 0.0440 and 0.0178), while GA and AOA show higher variability (std: 222.3245 and 249.3755), indicating less reliability.

The parameter values (e.g., x_1 to x_5) further highlight algorithmic differences. For instance, EWMA, WMA, and PSO maintain near-optimal values for all parameters, whereas GA and AOA deviate significantly, aligning with their poorer fitness scores. Overall, the data underscores EWMA

and PSO as robust choices for this optimization problem, while GA and AOA may require further tuning to improve consistency and performance.

Table 12. Optimization parameter results of the Himmelblau function.

Algorithm	EWMA	WMA	GWO	GA	DBO	PSO	GOOSE	DE	AOA	SO
x_1	78.0000	78.0000	78.0000	79.3856	78.0000	78.0000	78.0000	78.0000	82.8686	78.0000
x_2	33.0001	33.0000	33.0000	34.4546	33.0000	33.0000	33.0000	33.0000	33.0000	33.0000
x_3	29.9956	29.9953	30.0001	31.3733	30.0102	29.9953	29.9978	30.0185	31.8177	30.2731
x_4	44.9996	45.0000	44.8670	29.8774	45.0000	45.0000	44.7169	45.0000	32.3610	45.0000
x_5	36.7758	36.7758	36.8273	39.9798	36.7481	36.7758	36.8843	36.7313	37.5949	36.1041
$f(x)$	−30665.7765	−30665.5216	−30657.2736	-29527.3777	−30649.2928	−30665.5387	−30506.7313	−30643.7594	−29433.7433	−30619.6054
rank	1	3	4	9	6	2	8	7	10	5

Table 13. Indicator statistical results of the Himmelblau function.

Res.	EWMA	WMA	GWO	GA	DBO	PSO	GOOSE	DE	AOA	SO
min	−30665.7356	−30665.5386	−30660.6211	−29905.6131	−30662.5467	−30665.5387	−30757.6468	−30660.9570	−29674.0458	−30619.6054
worst	−30665.7920	−30662.0976	−30641.5528	−29204.7952	-30226.5285	−30665.4818	−30296.6540	−30600.8845	−28934.4351	−30533.4840
std	0.0440	1.0816	7.0467	222.3245	154.0783	0.0178	135.3187	19.5167	249.3755	29.0240
avg	−30665.7676	−30665.0564	−30653.6396	−29558.5531	−30573.3057	−30665.5323	−30482.5653	−30638.9546	−29340.3876	−30583.6336
median	−30665.7765	−30665.5216	−30657.2736	−29527.3777	−30649.2928	−30665.5387	−30506.7313	−30643.7594	−29433.7433	−30581.9498

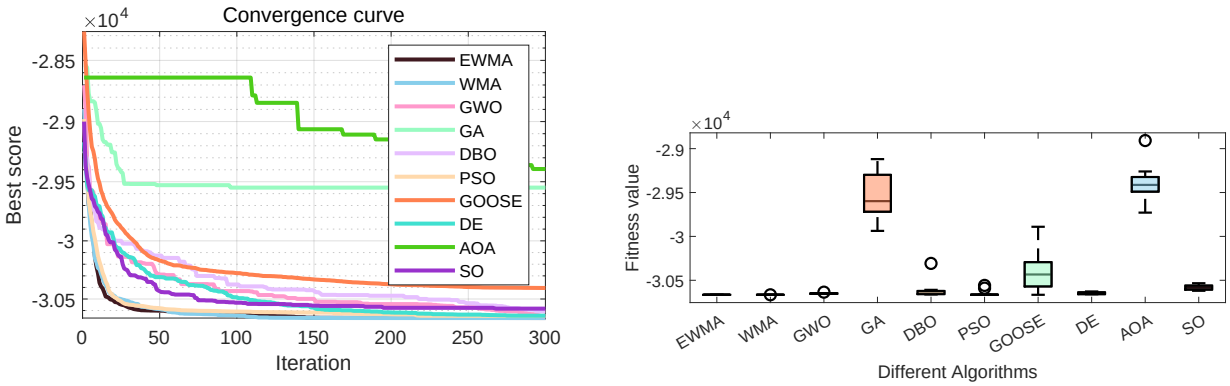

Figure 17. Convergence curve and ANOVA test graph of Himmelblau function optimization results.

5. Conclusions and prospects

This study has presented the EWMA, which significantly improves upon the original WMA through two key innovations: NCMM strategy and FROBL. Extensive experiments demonstrate that EWMA consistently outperforms eight state-of-the-art optimization algorithms in terms of solution accuracy, convergence speed, and stability across benchmark functions and real-world engineering problems.

EWMA's success lies in its effective balance between exploration and exploitation. NCMM

enables adaptive uncertainty control through its expectation-entropy-hyperentropy mechanisms, enabling dynamic adjustment of search behavior throughout optimization. FROBL enhances population diversity via oscillatory perturbations and nonlinear scaling, preventing premature convergence and accelerating convergence toward promising regions. The synergy between these components has proven particularly effective in managing complex constraints, as shown in the engineering case studies.

Despite its strengths, EWMA presents several challenges. The integration of NCMM and FROBL increases computational overhead, potentially impacting runtime when solving high-dimensional problems. Although designed for single-objective static optimization, EWMA's core mechanisms offer theoretical potential for extension: NCMM's entropy adaptation could be adapted to dynamic environments using time-varying *He* parameters, while FROBL's opposition-based sampling may help preserve Pareto diversity when combined with non-dominated sorting. Additionally, although the algorithm reduces the need for manual tuning, its performance remains sensitive to hyperparameters such as entropy and hyperentropy. Furthermore, scalability to ultra-large-scale problems has not been thoroughly tested and may require further refinement.

To address these limitations and extend EWMA's applicability, the following research directions are proposed:

- Multi-objective extension: Develop a multi-objective version of EWMA to enable the simultaneous optimization of competing objectives.
- Dynamic environments: Adapt EWMA for dynamic optimization problems in which objectives or constraints change over time: For example, in low-altitude unmanned aerial vehicle path planning.
- Hybrid approaches: Combine EWMA with machine learning techniques for surrogate-assisted optimization to enhance efficiency in computationally expensive problems.

The proposed EWMA represents a significant advancement in nature-inspired optimization, particularly for engineering problems with complex constraints. Its robust performance and adaptive design make it a valuable tool for researchers and practitioners tackling challenging optimization tasks. Researchers should aim to enhance its capabilities while preserving the exploration-exploitation balance that underpins its current success.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Conflict of interest

The authors declare there is no conflict of interest.

References

1. X. Yang, Nature-inspired optimization algorithms: Challenges and open problems, *J. Comput. Sci.*, **46** (2020), 101104. https://doi.org/10.1016/j.jocs.2020.101104

2. H. Ma, S. Shen, M. Yu, Z. Yang, M. Fei, H. Zhou, Multi-population techniques in nature inspired optimization algorithms: A comprehensive survey, *Swarm Evol. Comput.*, **44** (2019), 365–387. https://doi.org/10.1016/j.swevo.2018.04.011

3. W. Gao, Y. Wang, L. Liu, L. Huang, A gradient-based search method for multi-objective optimization problems, *Inf. Sci.*, **578** (2021), 129–146. https://doi.org/10.1016/j.ins.2021.07.051

4. L. Jiao, J. Zhao, C. Wang, X. Liu, F. Liu, L. Li, et al., Nature-inspired intelligent computing: A comprehensive survey, *Research*, **7** (2024), 0442. https://doi.org/10.34133/research.0442

5. T. A. Le, X. Yang, Generalized firefly algorithm for optimal transmit beamforming, *IEEE Trans. Wirel. Commun.*, **23** (2024), 5863–5877. https://doi.org/10.1109/TWC.2023.3328713

6. G. R. Harik, F. G. Lobo, D. E. Goldberg, The compact genetic algorithm, *IEEE Trans. Evol. Comput.*, **3** (1999), 287–297. https://doi.org/10.1109/4235.797971

7. T. Blackwell, J. Kennedy, Impact of communication topology in particle swarm optimization, *IEEE Trans. Evol. Comput.*, **23** (2019), 689–702. https://doi.org/10.1109/TEVC.2018.2880894

8. X. Chen, C. Ye, Y. Zhang, Strengthened grey wolf optimization algorithms for numerical optimization tasks and AutoML, *Swarm Evol. Comput.*, **94** (2025), 101891. https://doi.org/10.1016/j.swevo.2025.101891

9. R. Storn, K. Price, Differential evolution–A simple and efficient heuristic for global optimization over continuous spaces, *J. Glob. Optim.*, **11** (1997), 341–359. https://doi.org/10.1023/A:1008202821328

10. L. Ye, H. Ding, H. Xu, B. Xiang, Y. Wu, M. Gong, MFWOA: Multifactorial Whale Optimization Algorithm, *Swarm Evol. Comput.*, **91** (2024), 101768. https://doi.org/10.1016/j.swevo.2024.101768

11. J. Xue, B. Shen, Dung beetle optimizer: A new meta-heuristic algorithm for global optimization, *J. Supercomput.*, **79** (2023), 7305–7336. https://doi.org/10.1007/s11227-022-04959-6

12. R. Zhang, S. Mao, S. Zhao, C. Liu, An arithmetic optimization algorithm with balanced diversity and convergence for multimodal multiobjective optimization, *Swarm Evol. Comput.*, **91** (2024), 101724. https://doi.org/10.1016/j.swevo.2024.101724

13. Y. Zhu, H. Huang, J. Wei, J. Yi, J. Liu, M. Li, ISO: An improved snake optimizer with multi-strategy enhancement for engineering optimization, *Expert Syst. Appl.*, **281** (2025), 127660. https://doi.org/10.1016/j.eswa.2025.127660

14. X. Zhu, C. Jia, J. Zhao, C. Xia, W. Peng, J. Huang, et al., An enhanced artificial lemming algorithm and its application in UAV path planning, *Biomimetics*, **10** (2025), 377. https://doi.org/10.3390/biomimetics10060377

15. Y. Xiao, H. Cui, A. G. Hussien, F. A. Hashim, MSAO: A multi-strategy boosted snow ablation optimizer for global optimization and real-world engineering applications, *Adv. Eng. Inf.*, **61** (2024), 102464. https://doi.org/10.1016/j.aei.2024.102464

16. Q. Fu, Q. Li, X. Li, H. Wang, J. Xie, Q. Wang, MOFS-REPLS: A large-scale multi-objective feature selection algorithm based on real-valued encoding and preference leadership strategy, *Inf. Sci.*, **667** (2024), 120483. https://doi.org/10.1016/j.ins.2024.120483

17. R. Wang, S. Zhang, G. Zou, An improved multi-strategy crayfish optimization algorithm for solving numerical optimization problems, *Biomimetics*, **9** (2024), 361. https://doi.org/10.3390/biomimetics9060361

18. M. Chen, G. Zeng, K. Lu, A many-objective population extremal optimization algorithm with an adaptive hybrid mutation operation, *Inf. Sci.*, **498** (2019), 62–90. https://doi.org/10.1016/j.ins.2019.05.048

19. M. Zhou, M. Cui, D. Xu, S. Zhu, Z. Zhao, A. Abusorrah, Evolutionary optimization methods for high-dimensional expensive problems: A survey, *IEEE/CAA J. Autom. Sin.*, **11** (2024), 1092–1105. https://doi.org/10.1109/JAS.2024.124320

20. J. Zhao, D. Chen, R. Xiao, Z. Cui, H. Wang, I. Lee, Multi-strategy ensemble firefly algorithm with equilibrium of convergence and diversity, *Appl. Soft Comput.*, **123** (2022), 108938. https://doi.org/10.1016/j.asoc.2022.108938

21. M. Li, Y. Wang, J. Geng, W. Hong, Chaos cloud quantum bat hybrid optimization algorithm, *Nonlinear Dyn.*, **103** (2021), 1167–1193. https://doi.org/10.1007/s11071-020-06111-6

22. X. Yu, W. Xu, C. Li, Opposition-based learning grey wolf optimizer for global optimization, *Knowl.-Based Syst.*, **226** (2021), 107139. https://doi.org/10.1016/j.knosys.2021.107139

23. Z. Zhan, L. Shi, K. C. Tan, J. Zhang, A survey on evolutionary computation for complex continuous optimization, *Artif. Intell. Rev.*, **55** (2022), 59–110. https://doi.org/10.1007/s10462-021-10042-y

24. M. Ghasemi, M. Deriche, P. Trojovský, Z. Mansor, M. Zare, E. Trojovská, et al., An efficient bio-inspired algorithm based on humpback whale migration for constrained engineering optimization, *Results Eng.*, **25** (2025), 104215. https://doi.org/10.1016/j.rineng.2025.104215

25. S. Mohapatra, P. Mohapatra, Fast random opposition-based learning Golden Jackal Optimization algorithm, *Knowl.-Based Syst.*, **275** (2023), 110679. https://doi.org/10.1016/j.knosys.2023.110679

26. G. Hu, J. Zhong, C. Zhao, G. Wei, C. Chang, LCAHA: A hybrid artificial hummingbird algorithm with multi-strategy for engineering applications, *Comput. Methods Appl. Mech. Eng.*, **415** (2023), 116238. https://doi.org/10.1016/j.cma.2023.116238

27. X. Yang, R. Wang, D. Zhao, F. Yu, C. Huang, A. A. Heidari, et al., An adaptive quadratic interpolation and rounding mechanism sine cosine algorithm with application to constrained engineering optimization problems, *Expert Syst. Appl.*, **213** (2023), 119041. https://doi.org/10.1016/j.eswa.2022.119041

28. F. Gao, G. Liu, X. Wu, W. Liao, Optimization algorithm-based approach for modeling large deflection of cantilever beam subject to tip load, *Mech. Mach. Theory*, **167** (2022), 104522. https://doi.org/10.1016/j.mechmachtheory.2021.104522

29. A Tian, F. Liu, H. Lv, Snow Geese Algorithm: A novel migration-inspired meta-heuristic algorithm for constrained engineering optimization problems, *Appl. Math. Model.*, **126** (2024), 327–347. https://doi.org/10.1016/j.apm.2023.10.045

30. A. Kumar, G. Wu, M. Z. Ali, R. Mallipeddi, P. N. Suganthan, S. Das, A test-suite of non-convex constrained optimization problems from the real-world and some baseline results, *Swarm Evol. Comput.*, **56** (2020), 100693. https://doi.org/10.1016/j.swevo.2020.100693

AIMS Press