algorithms

MDPI

Article

Adaptive Elite Differential Gold Rush Optimizer for Three-Dimensional UAV Path Planning in Complex Mountainous Environments

Fan Yang * and Lixin Lyu

School of Information and Artificial Intelligence, Anhui Business College, Wuhu 241002, China; lixin_lyu@abc.edu.cn
* Correspondence: yangfan@abc.edu.cn

Abstract

To improve the reliability and path quality of three-dimensional UAV path planning in complex mountainous environments, this paper proposes an Adaptive Elite Differential Gold Rush Optimizer (AEDGRO). The main novelty of AEDGRO lies in the coordinated integration of three enhancement mechanisms into the original Gold Rush Optimizer: chaotic good-point initialization for improving initial population coverage, adaptive elite differential mining for strengthening exploitation around promising regions, and stagnation-aware Gaussian–Cauchy mutation for escaping local optima. A UAV path-planning model is constructed by considering path length, altitude fluctuation, trajectory smoothness, terrain collision avoidance, threat-region avoidance, and UAV safety clearance. The experimental results on the IEEE CEC2017 benchmark suite show that AEDGRO obtains the best Friedman average ranking of 1.63, outperforming the original GRO with a ranking of 4.80. In the UAV path-planning experiments, AEDGRO achieves the lowest mean fitness value of 235.69 and the smallest standard deviation of 7.55, indicating better path quality and stronger robustness than the compared algorithms. The generated trajectories are smoother and can effectively avoid mountainous terrain and threat regions. These results demonstrate that AEDGRO has clear advantages in global optimization accuracy, convergence stability, and UAV path-planning applicability.

Keywords: UAV path planning; Gold Rush Optimizer; adaptive elite differential mining; chaotic good-point initialization; Gaussian–Cauchy mutation

1. Introduction

With the rapid development of artificial intelligence, unmanned systems, and complex engineering optimization technologies, efficiently obtaining optimal solutions in high-dimensional, nonlinear, multi-constrained, and complex search spaces has become an important research topic in the field of optimization. Particularly in practical applications such as UAV path planning [1], robot navigation [2], energy scheduling [3], wireless sensor network deployment [4], and engineering parameter optimization [5], traditional mathematical optimization methods often struggle to effectively solve complex optimization problems because they strongly depend on the continuity, differentiability, and convexity of objective functions [6]. Therefore, swarm intelligence-based metaheuristic optimization algorithms have gradually become important approaches for solving complex optimization problems due to their gradient-free characteristics, flexible structures, and strong global search capability.

Check for updates

Academic Editor: Xiao-Zhi Gao

Received: 18 May 2026
Revised: 3 June 2026
Accepted: 7 June 2026
Published: 10 June 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.

In recent years, a large number of swarm intelligence optimization algorithms have been proposed and widely applied in various optimization scenarios. For example, Particle Swarm Optimization (PSO) [7] simulates the foraging behavior of bird flocks to achieve cooperative population search and possesses advantages such as simple structure and fast convergence speed. Differential Evolution (DE) [8] enhances global search capability through differential mutation mechanisms and exhibits strong competitiveness in continuous optimization problems. Grey Wolf Optimizer (GWO) [9] achieves a balance between exploration and exploitation by simulating the hierarchical hunting mechanism of grey wolf packs. In addition, an increasing number of novel swarm intelligence algorithms have been proposed, such as the Swarm Intelligence-based Baboon Optimization Algorithm (BOA) [10] was developed by mimicking the dominance hierarchy, cooperative foraging behavior, and adaptive stress regulation observed in baboon communities; Trip Road Optimization (TRO) [11], inspired by human travel decision-making behavior and composed of three core mechanisms: route exploration for global search, social influence for convergence guidance, and adaptive adjustment for refined exploitation; Star-Nosed Mole Optimizer (SNMO) [12], which simulates the foraging behavior of star-nosed moles; Enzyme Action Optimizer (EAO) [13], inspired by adaptive enzyme mechanisms in biological systems; Crocodile Ambush Optimization Algorithm (CAOA) [14], motivated by the energy-saving and ambush hunting behaviors of crocodiles; Bounty Hunter Optimizer (BHO) [15], inspired by the "thorough investigation," "rough search," and "hunter reassignment" behaviors of bounty hunters; and Gray Langurs Optimizer (GLO) [16], proposed based on the group behavior and social hierarchy of gray langurs. These algorithms further improve the ability to solve complex problems by introducing novel biological behavior mechanisms or cooperative population strategies. Furthermore, several recently proposed metaheuristic algorithms have demonstrated promising optimization performance, including the Dhole Optimization Algorithm (DOA) [17], Secretary Bird Optimization Algorithm (SBOA) [18], Hannibal Barca optimizer (HBO) [19], and Gold Rush Optimizer (GRO) [20], Gold Rush Optimizer (AGRO) [21], Quantum-Enhanced Gold Rush Optimizer (QHGRO) [22]. Nevertheless, although these Swarm Intelligence algorithms have demonstrated remarkable effectiveness across a wide range of applications, many existing approaches still encounter several limitations when addressing complicated high-dimensional multimodal optimization tasks, including weakened population diversity, slow convergence behavior, and a strong tendency to become trapped in local optimal regions.

As one of the representative complex constrained optimization problems, UAV path planning has attracted extensive attention in recent years. During tasks such as reconnaissance, logistics transportation, and disaster monitoring, UAVs are required to plan safe, smooth, and low-cost flight paths in complex environments [23]. Particularly in three-dimensional mountainous environments, path planning must not only avoid complex terrains and threat regions but also simultaneously consider multiple objectives, including flight distance, trajectory smoothness, and flight stability [24]. Therefore, this problem exhibits strong characteristics of high dimensionality, nonlinearity, and multiple constraints. Traditional path search methods, such as the Dijkstra algorithm [25], A* algorithm [26], and artificial potential field method [27], can achieve satisfactory planning results in relatively simple environments. For example, Lazy Chaos A* introduces chaos-inspired semi-known exploration behavior to improve the path search efficiency of a 3R robot arm [28]. In addition, comparative studies on real-time path-planning methods for planar manipulators with obstacle avoidance have shown that graph-search and sampling-based methods are efficient for structured environments [29]. However, these methods are usually more suitable for discrete or semi-known search spaces, while three-dimensional UAV path planning

in mountainous environments requires continuous waypoint optimization under terrain, threat, smoothness, and altitude constraints.

In addition to general-purpose benchmark optimization, metaheuristic algorithms have also been widely applied to control and robotic systems. For example, the Whale Optimization Algorithm was employed for finite-time control of wing-rock motion in delta wing aircraft, demonstrating that swarm intelligence can effectively tune nonlinear control parameters [30]. The Bee Algorithm was applied to the control design of two-link robot arm systems, where optimization-based tuning improved the control performance under nonlinear robotic dynamics [31]. Similarly, the Butterfly Optimization Algorithm was combined with optimal linear active disturbance rejection control to reject wing-rock motion in delta wing aircraft [32]. These studies confirm the effectiveness of metaheuristic optimizers in nonlinear engineering systems, especially for controller tuning and trajectory-related optimization problems. However, most of these studies mainly focus on control parameter optimization, while fewer works investigate how to enhance the optimizer itself for three-dimensional UAV path planning under terrain and threat constraints.

Gold Rush Optimizer (GRO) [20] is a recently proposed swarm intelligence optimization algorithm inspired by the behavior of miners searching for gold mines during the gold rush era. The proposed method maintains an effective trade-off between broad search capability and local refinement performance by employing a migration-based search mechanism, mining, and cooperative search behaviors and has demonstrated competitive optimization performance on several optimization problems. Nevertheless, the original GRO still suffers from several limitations. For instance, random initialization may lead to uneven population distribution, thereby reducing the efficiency of early-stage exploration. As the iteration process proceeds, population diversity gradually decreases, making the algorithm prone to premature convergence to local optima. In addition, its local exploitation capability remains insufficient, resulting in relatively slow convergence on complex multimodal problems [33–35]. Therefore, further improving the population diversity, convergence efficiency, and local optimum escaping capability of GRO has become an important research issue.

To address the above problems, this paper proposes an Adaptive Elite Differential Gold Rush Optimizer (AEDGRO). First, a Chaotic Good-Point Initialization strategy (CGPI) is introduced, which combines good-point sets and chaotic mapping to improve the uniformity and randomness of the initial population, thereby enhancing early-stage global exploration capability. Second, an Adaptive Elite Differential Mining strategy (AEDM) is incorporated into the original mining stage to improve local exploitation capability and convergence speed through elite guidance and differential perturbation mechanisms. Finally, a Stagnation-Aware Gaussian–Cauchy Mutation strategy (SGCM) is proposed, which dynamically introduces Gaussian mutation and Cauchy long-jump mechanisms when the algorithm falls into stagnation, thereby enhancing population diversity and helping the algorithm escape from local optima. Through the synergistic cooperation of these three strategies, AEDGRO can establish a more effective dynamic balance between global exploration and local exploitation.

To verify the effectiveness of the proposed algorithm, extensive experiments are conducted on the IEEE CEC2017 benchmark functions and three-dimensional UAV path-planning problems, and comparisons are performed with several classical and advanced swarm intelligence optimization algorithms. The experimental results demonstrate that AEDGRO exhibits significant advantages in terms of optimization accuracy, convergence performance, stability, and robustness. Moreover, it can effectively solve complex high-dimensional optimization problems and demonstrates strong application potential in practical UAV path-planning tasks.

The primary innovations and major findings of this study are summarized as follows:

(1) A new Adaptive Elite Differential Gold Rush Optimizer (AEDGRO) is proposed for complex continuous optimization and three-dimensional UAV path planning. Compared with the original GRO, AEDGRO improves the search process from three key aspects: initialization quality, local exploitation capability, and stagnation-escaping ability.

(2) Three synergistic improvement strategies are designed and integrated into GRO. The chaotic good-point initialization strategy improves the spatial distribution quality of the initial population; the adaptive elite differential mining strategy enhances convergence efficiency and local search accuracy; and the stagnation-aware Gaussian–Cauchy mutation mechanism maintains population diversity and helps the algorithm escape from local optima.

(3) The proposed AEDGRO is comprehensively validated on the IEEE CEC2017 benchmark suite and a three-dimensional UAV path-planning problem. The experimental results demonstrate that AEDGRO achieves better optimization accuracy, convergence stability, and path quality than the compared algorithms, confirming its effectiveness for complex engineering optimization tasks.

The remainder of this paper is organized as follows: Section 2 establishes the UAV path-planning model and constructs the UAV path-planning framework based on AEDGRO. Section 3 introduces the proposed AEDGRO algorithm in detail, including the original GRO algorithm and the three proposed improvement strategies. Section 4 presents the experimental setup and provides experimental validation and performance analysis on both UAV path-planning problems and the CEC2017 benchmark functions. Finally, Section 5 presents the overall conclusions of this study and outlines several potential directions for future investigation.

2. UAV Path-Planning Model and Optimization Framework

2.1. UAV Path Representation

To effectively solve the three-dimensional unmanned aerial vehicle (UAV) path-planning problem in complex mountainous environments, the flight trajectory is represented by a sequence of intermediate control nodes distributed between the start point and the destination point [36,37]. The optimization objective is to determine a feasible and smooth trajectory while satisfying terrain and threat avoidance constraints. Assume that the start point and end point of the UAV are respectively defined as $P_s = (x_s, y_s, z_s)$ and $P_e = (x_e, y_e, z_e)$, where x, y, and z denote the spatial coordinates in the three-dimensional environment [38,39].

To construct the flight path, N intermediate waypoints are introduced between the start and end points. Each waypoint is represented as follows [38,39]:

$$P_i = (x_i, y_i, z_i), i = 1, 2, \ldots, N \tag{1}$$

Thus, the complete path can be expressed as follows:

$$\mathcal{P} = \{P_s, P_1, P_2, \ldots, P_N, P_e\} \tag{2}$$

In this study, the x-coordinates of the intermediate nodes are uniformly distributed between the start and end points to guarantee the continuity and forward progression of the trajectory, while the y- and z-coordinates are optimized by the proposed algorithm. Therefore, the decision vector of the optimization problem is formulated as follows:

$$X = [y_1, y_2, \ldots, y_N, z_1, z_2, \ldots, z_N] \tag{3}$$

where the dimension of the search space is $D = 2N$.

After generating the intermediate nodes, cubic spline interpolation is employed to construct a smooth continuous trajectory. Compared with piecewise linear interpolation, cubic spline interpolation can significantly improve path smoothness and reduce abrupt turning behaviors, thereby enhancing the maneuverability and safety of the UAV.

The interpolated trajectory is represented as follows:

$$\Gamma(t) = (X(t), Y(t), Z(t)), t \in [0, 1] \tag{4}$$

where $X(t)$, $Y(t)$, and $Z(t)$ denote the interpolated spatial coordinates of the UAV trajectory.

In this study, the complete UAV path is represented by the start point, the destination point, and n_w intermediate waypoints. The x-coordinates of the intermediate waypoints are uniformly distributed between the start and destination points to ensure forward flight progression $x_i = x_s + \frac{i}{n_w+1}(x_t - x_s), i = 1, 2, \ldots, n_w$, where x_s and x_t are the x-coordinates of the start and target points, respectively. The y- and z-coordinates of the intermediate waypoints are optimized by AEDGRO. Therefore, each candidate solution is encoded as $X = [y_1, z_1, y_2, z_2, \ldots, y_{n_w}, z_{n_w}]$ and the dimension of the optimization problem is ($D = 2n_w$). In the UAV path-planning experiment, ($n_w = 2$) is used, resulting in a four-dimensional decision vector. After the intermediate waypoints are generated, cubic spline interpolation is applied to connect the start point, intermediate waypoints, and destination point into a continuous smooth trajectory.

2.2. Terrain and Threat Modeling

To simulate realistic flight environments, a three-dimensional mountainous terrain model is constructed. The terrain elevation is generated by combining sinusoidal fluctuation functions and multiple Gaussian mountain peaks, enabling the environment to contain complex topographic variations.

The terrain elevation function is defined as follows:

$$H(x, y) = max(H_s(x, y), H_m(x, y)) \tag{5}$$

where $H_s(x, y)$ represents the sinusoidal terrain fluctuation component and $H_m(x, y)$ denotes the mountain peak component.

The sinusoidal terrain model is formulated as follows [38,39]:

$$\begin{pmatrix} H_s(x, y) = \sin(y + a) + b\sin(x) + c\cos\big(d\big(x^2 + y^2\big)\big) + \\ e\cos(y) + f\sin(f(x^2 + y^2)) + g\cos(y) \end{pmatrix} \tag{6}$$

where a, b, c, d, e, f, g are terrain control parameters.

To further increase environmental complexity, multiple mountain peaks are introduced using Gaussian functions:

$$H_m(x, y) = \sum_{i=1}^{M} h_i \exp\left(-\frac{(x - A_{x,i})^2}{a_i^2} - \frac{(y - A_{y,i})^2}{b_i^2} \right) \tag{7}$$

where a_i denotes the number of mountain peaks; $(A_{x,i}, A_{y,i})$ represents the center position of the i-th mountain peak; h_i is the peak height; a_i and b_i control the slope steepness along the x- and y-directions, respectively.

In addition to terrain obstacles, threat regions are introduced to simulate dangerous flight zones such as radar detection areas or restricted regions. Each threat region is modeled as a circular forbidden area characterized by its center coordinate and influence radius.

Suppose the center of the k-th threat area is $T_k = (x_k, y_k)$ and its radius is R_k.

Then, the UAV trajectory is considered unsafe if:

$$\sqrt{(x - x_k)^2 + (y - y_k)^2} < R_k \tag{8}$$

For terrain collision avoidance, the UAV altitude must always remain above the terrain surface:

$$Z(t) \geq H(X(t), Y(t)) \tag{9}$$

If either the terrain collision constraint or the threat-region constraint is violated, the generated path is regarded as infeasible and a large penalty value is assigned to the objective function.

2.3. UAV Safety Clearance and Simplified Dynamic Constraints

To improve the practical feasibility of the generated trajectory, the physical size of the UAV and basic motion constraints are considered in the revised path-planning model. The UAV is approximated as a sphere with an equivalent safety radius (r_u). Therefore, the terrain and threat avoidance constraints are expanded by a safety clearance margin. For terrain avoidance, the altitude of each trajectory point should satisfy:

$$z(s) \geq H(x(s), y(s)) + r_u + h_s \tag{10}$$

where $H(x, y)$ denotes the terrain elevation, r_u is the equivalent UAV radius, and h_s is the additional safety altitude margin.

For threat-region avoidance, the horizontal distance between the UAV and the center of the k-th threat region should satisfy:

$$\sqrt{(x(s) - x_k)^2 + (y(s) - y_k)^2} \geq R_k + r_u + d_s \tag{11}$$

where R_k is the original threat radius and d_s is the safety distance margin.

In addition, to avoid dynamically infeasible trajectories, two simplified motion constraints are introduced, including the maximum turning angle and maximum climb angle:

$$\theta_i \leq \theta_{max}, \ \ \gamma_i \leq \gamma_{max} \tag{12}$$

where θ_i denotes the turning angle between two adjacent path segments and γ_i denotes the climb angle. If these constraints are violated, a penalty term is added to the objective function. Although a full nonlinear UAV dynamic model is not considered in this study, these simplified constraints can improve the maneuverability and safety of the planned trajectory.

2.4. UAV Path Cost Function

To comprehensively evaluate the quality of the generated UAV trajectory, a multi-objective path evaluation model is established. The objective function simultaneously considers flight distance, altitude variation, and trajectory smoothness.

The overall fitness function is formulated as follows:

$$F = w_1 F_1 + w_2 F_2 + w_3 F_3 \tag{13}$$

where F_1 denotes the path length cost; F_2 represents the altitude fluctuation cost; F_3 corresponds to the path smoothness cost; w_1, w_2 and w_3 are weighting coefficients satisfying:

$$w_1 + w_2 + w_3 = 1 \tag{14}$$

In this study, the weights are empirically set as $w_1 = 0.5, \ w_2 = 0.3, w_3 = 0.2$.

(1) Path Length Cost

The total flight distance of the UAV is calculated using the Euclidean distance between adjacent interpolated trajectory points:

$$F_1 = \sum_{i=1}^{L-1} \sqrt{(X_{i+1} - X_i)^2 + (Y_{i+1} - Y_i)^2 + (Z_{i+1} - Z_i)^2} \tag{15}$$

where L denotes the number of interpolated trajectory points.

A shorter path generally leads to lower energy consumption and shorter mission execution time.

(2) Altitude Fluctuation Cost

To reduce excessive altitude oscillations during flight, the altitude variation term is defined as follows:

$$F_2 = \sum_{i=1}^{L} \left| Z_i - \overline{Z} \right| \tag{16}$$

where $\overline{Z}$ is the average altitude of the trajectory.

This term helps maintain stable flight altitude and improves flight safety.

(3) Path Smoothness Cost

To avoid sharp turning behaviors, the turning smoothness of the trajectory is evaluated using the cosine similarity between adjacent path segments.

Suppose two adjacent trajectory vectors are defined as $\mathbf{v}_i = (\Delta x_i, \Delta y_i, \Delta z_i)$ and $\mathbf{v}_{i+1} = (\Delta x_{i+1}, \Delta y_{i+1}, \Delta z_{i+1})$.

Then, the turning evaluation coefficient is computed as follows:

$$C_i = \frac{\mathbf{v}_i \cdot \mathbf{v}_{i+1}}{\| \mathbf{v}_i \| \| \mathbf{v}_{i+1} \|} \tag{17}$$

The smoothness cost is formulated as follows:

$$F_3 = \sum_{i=1}^{L-2} (\cos \phi - C_i) \tag{18}$$

where ϕ is the expected turning angle threshold.

A smaller smoothness cost indicates a smoother trajectory with fewer abrupt direction changes.

This study focuses on the high-level path-planning problem rather than low-level nonlinear flight control. Therefore, the UAV is modeled as a point-mass path-following vehicle with safety clearance, turning angle, climb angle, and obstacle avoidance constraints. This modeling approach is commonly used in optimization-based path planning because it separates global trajectory generation from low-level attitude and dynamics control. The planned path is then tracked by a simple waypoint-based controller. Although the full nonlinear UAV dynamic model is not included, the added geometric and maneuverability constraints can ensure that the generated trajectory is smoother and more feasible than a purely geometric path. The integration of a complete nonlinear UAV dynamic model will be considered in future work.

2.5. Optimization Framework Based on AEDGRO

Based on the established UAV path-planning model, the proposed Adaptive Elite Differential Gold Rush Optimizer (AEDGRO) is employed to search for the optimal trajectory in the continuous solution space.

First, the UAV path is encoded into a population of candidate solutions, where each individual represents a complete set of intermediate waypoints. During the optimization process, the population continuously evolves through adaptive exploration and exploitation mechanisms.

The overall optimization procedure of the proposed framework can be summarized as follows:

Step 1: Initialize the population within the feasible search space.

Step 2: Construct the UAV trajectory using waypoint encoding and cubic spline interpolation.

Step 3: Evaluate path feasibility according to terrain collision and threat avoidance constraints.

Step 4: Compute the path fitness using the comprehensive cost model.

Step 5: Update the population positions using the proposed AEDGRO strategies.

Step 6: Continuously update the global best solution until the maximum iteration number is reached.

Step 7: Output the optimal collision-free UAV trajectory.

To provide a clearer description of the proposed framework, the general block diagram of AEDGRO-based UAV path planning is presented in Figure 1. The framework consists of five main modules: environment modeling, path encoding, AEDGRO-based waypoint optimization, path reconstruction and feasibility evaluation, and final trajectory output. In the feasibility evaluation module, both terrain collision and threat-region violation are checked. If a candidate trajectory violates any constraint, the penalty assignment module is activated and connected to the fitness evaluation module. Otherwise, the trajectory cost is directly calculated based on path length, altitude variation, and smoothness.

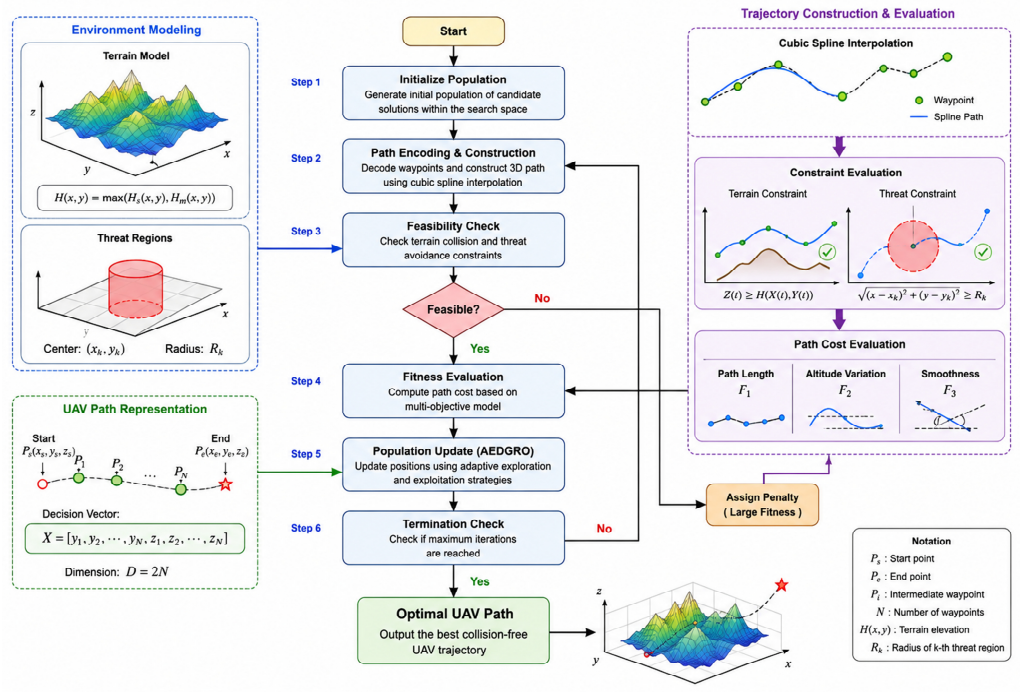


Figure 1. Overall framework of the proposed AEDGRO-based UAV path-planning model.

3. Adaptive Elite Differential Gold Rush Optimizer

3.1. Review of GRO

The Gold Rush Optimizer (GRO) [20] is a population-based metaheuristic optimization algorithm proposed by Kamran Zolfi, which is inspired by the historical gold rush phenomenon and the cooperative gold prospecting behaviors of miners. The algorithm mainly simulates three key prospecting mechanisms during the gold rush era, including migration toward rich gold mines, local mining around promising areas, and collaboration among prospectors. Through the interaction of these three behaviors, GRO achieves a dynamic balance between global exploration and local exploitation.

In GRO, each gold prospector represents a candidate solution in the search space. The population of prospectors is initialized randomly within the problem boundaries. The position matrix of the population can be represented as follows [20]:

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,D} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,D} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N,1} & x_{N,2} & \cdots & x_{N,D} \end{bmatrix} \tag{19}$$

where N denotes the population size, D represents the dimensionality of the optimization problem, and $x_{i,j}$ denotes the j-th dimension of the i-th prospector. The fitness value of each prospector is evaluated by the objective function $f(\cdot)$. The best solution found during the search process is denoted as X_{best}, which represents the currently discovered richest gold mine.

The first search behavior in GRO is the migration mechanism. During the gold rush era, prospectors migrated toward regions where richer gold mines were discovered. In GRO, this behavior is modeled by guiding individuals toward the current global best solution. The migration operator is mathematically formulated as follows [20]:

$$\begin{pmatrix} D_1 = C_1 \cdot X_{best} - X_i \\ X_i^{t+1} = X_i^t + A_1 \cdot D_1 \end{pmatrix} \tag{20}$$

where X_i denotes the current individual, X_{best} represents the current global best solution, and D_1 denotes the migration direction vector. The coefficient vectors A_1 and C_1 are defined as follows [20]:

$$\begin{pmatrix} A_1 = 1 + l_1(r_1 - 0.5) \\ C_1 = 2r_2 \end{pmatrix} \tag{21}$$

where r_1 and r_2 are uniformly distributed random vectors in $[0, 1]$, and l_1 is the convergence control parameter. This migration mechanism allows prospectors to gradually approach promising regions while maintaining certain exploration capability.

The second behavior is the mining mechanism, which simulates the local gold mining process around promising locations. Each prospector performs local exploitation around randomly selected prospectors to search for potentially richer areas. The mathematical model of the mining operator is given as follows [20]:

$$\begin{pmatrix} D_2 = X_i - X_r \\ X_i^{t+1} = X_r + A_2 \cdot D_2 \end{pmatrix} \tag{22}$$

where X_r denotes a randomly selected individual and A_2 is defined as follows:

$$A_2 = 2l_2 r_1 - l_2 \tag{23}$$

Here, l_2 is another adaptive convergence parameter used to gradually reduce the search range during iterations. When $|A_2| < 1$, the mining process focuses more on local exploitation, whereas larger values enhance global exploration capability.

The third search behavior is the collaboration mechanism. During gold prospecting, miners often cooperated to share mining information and improve searching efficiency. GRO models this collaborative behavior through information interaction among individuals. The collaboration operator is expressed as follows [20]:

$$\begin{pmatrix} D_3 = X_{g2} - X_{g1} \\ X_i^{t+1} = X_i + r_1 \cdot D_3 \end{pmatrix} \tag{24}$$

where $g1$ and $g2$ denote two randomly selected prospectors. This strategy dynamically adjusts the search direction based on the positional difference between two individuals, thereby enhancing the exploration capability of the algorithm.

After generating new candidate positions, GRO adopts a greedy selection mechanism to determine whether the new solution should replace the current solution. For minimization problems, the update rule is defined as follows [20]:

$$X_i^{t+1} = X_{new}, \text{ if } f(X_{new}) < f(X_i) \tag{25}$$

Otherwise, the original solution is retained. Meanwhile, boundary control is employed to ensure that all individuals remain within the predefined search space.

Overall, GRO achieves optimization through the collaborative interaction of migration, mining, and cooperation behaviors. The migration operator strengthens convergence toward the global best region, the mining operator enhances local exploitation capability, and the collaboration operator improves global exploration and population diversity. Although GRO demonstrates competitive performance on various benchmark functions and engineering optimization problems, the original algorithm still suffers from several limitations, including insufficient initial population diversity, relatively slow convergence speed, and susceptibility to local optima stagnation in high-dimensional multimodal optimization problems. Therefore, to further improve the optimization capability of GRO, this paper proposes the Adaptive Elite Differential Gold Rush Optimizer (AEDGRO), which introduces three enhanced strategies to strengthen population diversity, convergence speed, and global optimization performance.

3.2. Chaotic Good-Point Initialization Strategy (CGPI)

In the original GRO algorithm, the initial population is generated completely randomly within the search boundaries. Although this initialization mechanism is simple and easy to implement, the generated individuals are often unevenly distributed in the search space, which may lead to insufficient population diversity and weak global exploration capability during the early optimization stage. Particularly for high-dimensional multimodal optimization problems, poor initialization quality can easily cause premature convergence and deteriorate the convergence speed of the algorithm.

To address this issue, a Chaotic Good-Point Initialization (CGPI) strategy is introduced in this paper. The proposed strategy combines the advantages of good-point set theory and chaotic mapping to simultaneously enhance the uniformity and randomness of the initial population distribution.

First, the good-point set is utilized to generate uniformly distributed initial individuals. Compared with traditional random initialization, the good-point mechanism can effectively reduce population aggregation and improve the coverage capability of individuals throughout the search space. The mathematical model of the good-point initialization is expressed as follows [40]:

$$g_{i,j} = \mathrm{mod}\left(i \cdot r^j, 1\right) \tag{26}$$

where $i = 1, 2, \ldots, N, j = 1, 2, \ldots, D, N$ denotes the population size, D represents the dimensionality of the problem, and r is defined as $r = \frac{\sqrt{5}-1}{2}$.

Although the good-point set improves population uniformity, its deterministic nature may reduce population randomness. Therefore, logistic chaotic mapping is further introduced to enhance ergodicity and stochasticity. The chaotic sequence is generated as follows:

$$c_{t+1} = 4c_t(1 - c_t) \tag{27}$$

where $c_t \in (0, 1)$ denotes the chaotic variable at iteration t.

Subsequently, the good-point population and chaotic population are combined to construct the final initialized population:

$$X = 0.5G + 0.5C \tag{28}$$

where G and C denote the good-point population and chaotic population, respectively.

Finally, the individuals are mapped into the search space according to:

$$X_i = lb + X_i \cdot (ub - lb) \tag{29}$$

As show in Figure 2, through the combination of good-point distribution and chaotic perturbation, the proposed CGPI strategy significantly improves the initial population diversity and spatial coverage capability. As a result, the algorithm can explore promising regions more effectively during the early optimization stage, thereby accelerating convergence speed and improving global search performance.

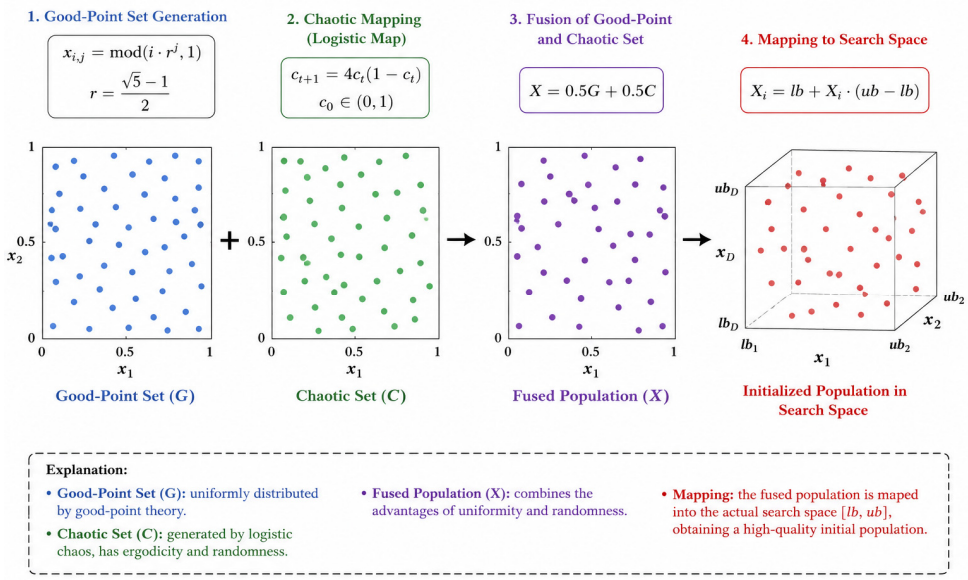


Figure 2. Schematic illustration of the Chaotic Good-Point Initialization Strategy (CGPI).

3.3. Adaptive Elite Differential Mining Strategy (AEDM)

Although the migration and mining mechanisms of the original GRO algorithm provide a certain balance between exploration and exploitation, the search process still heavily depends on random interactions among individuals. As the optimization progresses, the population diversity gradually decreases, causing the algorithm to suffer from slow convergence speed and insufficient exploitation capability in complex multimodal optimization problems.

To further strengthen the local exploitation capability and convergence efficiency of GRO, an Adaptive Elite Differential Mining (AEDM) strategy is proposed in this paper. Inspired by differential evolution and elite collaborative learning mechanisms, AEDM introduces adaptive differential perturbation and elite-guided search into the original mining phase.

First, the elite individuals with better fitness values are selected from the population. The elite mean solution is calculated as follows:

$$E_{mean} = \frac{1}{N_e} \sum_{i=1}^{N_e} X_i \tag{30}$$

where N_e denotes the number of elite individuals and E_{mean} represents the elite mean position.

Then, three random individuals X_{r1}, X_{r2}, and X_{r3} are selected from the population to construct the differential mutation vector. The adaptive mutation operator is formulated as follows [41]:

$$V_i = X_i + F(X_{best} - X_i) + F(X_{r1} - X_{r2}) + \lambda(E_{mean} - X_{r3}) \tag{31}$$

where X_i denotes the current individual; X_{best} represents the global best solution; F denotes the adaptive scaling factor; $\lambda = 0.3rand$ denotes the elite guidance coefficient.

The adaptive scaling factor is dynamically adjusted according to the iteration process:

$$F = 0.9 - 0.5\frac{t}{T} \tag{32}$$

where t and T denote the current iteration number and maximum iteration number, respectively.

To further balance exploration and exploitation, the crossover probability is adaptively updated as follows:

$$CR = 0.2 + 0.7\frac{t}{T} \tag{33}$$

The trial vector is generated by:

$$U_{i,j} = \begin{cases} V_{i,j}, & rand < CR \\ X_{i,j}, & otherwise \end{cases} \tag{34}$$

As show in Figure 3, in the early optimization stage, larger mutation factors maintain strong global exploration capability, while in the later stage, smaller perturbations gradually enhance local exploitation precision. Meanwhile, the elite guidance mechanism effectively utilizes high-quality population information to accelerate convergence toward promising regions.

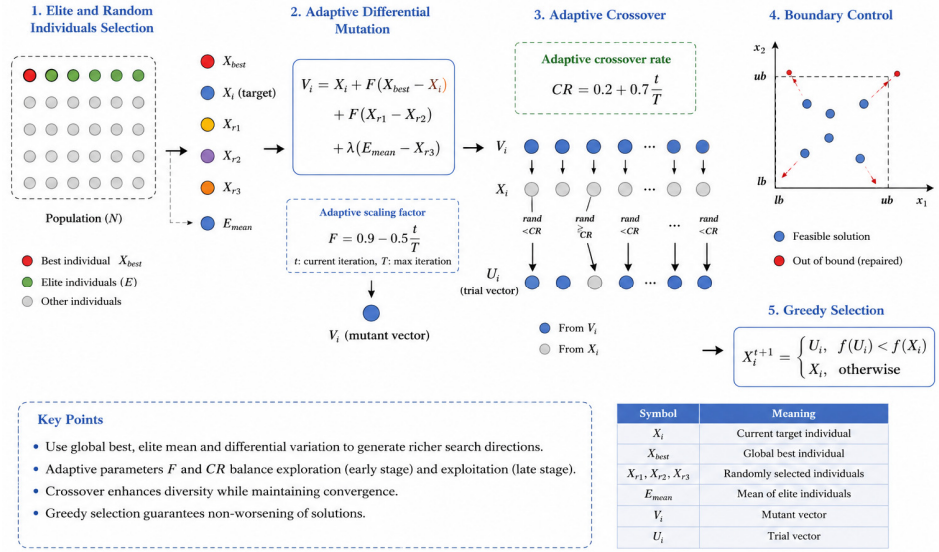


Figure 3. Schematic illustration of the Adaptive Elite Differential Mining Strategy (AEDM).

Consequently, the proposed AEDM strategy significantly enhances convergence speed, optimization accuracy, and exploitation capability, while effectively maintaining population diversity during the optimization process.

3.4. Stagnation-Aware Gaussian–Cauchy Mutation Strategy (SGCM)

Although the proposed CGPI and AEDM strategies considerably improve the exploration and exploitation abilities of GRO, the population may still experience stagnation

when solving highly multimodal optimization problems. As the population diversity gradually decreases during the later optimization stage, the algorithm may become trapped in local optima, resulting in premature convergence and degraded optimization accuracy.

To overcome this limitation, a Stagnation-Aware Gaussian–Cauchy Mutation (SGCM) strategy is introduced. The proposed strategy dynamically activates mutation perturbation when the algorithm enters a stagnation state, thereby enhancing population diversity and escaping local optima.

First, the stagnation condition is defined as follows:

$$\left| f_{best}^{t} - f_{best}^{t-1} \right| < \varepsilon \tag{35}$$

where f_{best}^{t} and f_{best}^{t-1} denote the best fitness values of two consecutive iterations, and ε is a very small threshold value.

When the stagnation condition is satisfied continuously for several iterations, Gaussian mutation or Cauchy mutation is probabilistically selected to perturb the population.

The Gaussian mutation operator is expressed as follows [42]:

$$X_{new} = X_{best} + \alpha \cdot N(0,1) \cdot (ub - lb) \tag{36}$$

The Cauchy mutation operator is formulated as follows:

$$X_{new} = X_{best} + \alpha \cdot C(0,1) \cdot (ub - lb) \tag{37}$$

where $N(0,1)$ denotes the Gaussian distribution; $C(0,1)$ denotes the Cauchy distribution; α represents the adaptive mutation scale factor. The adaptive mutation coefficient is dynamically adjusted as follows:

$$\alpha = 0.1\left(1 - \frac{t}{T}\right)^2 + 0.001 \tag{38}$$

As show in Figure 4, the Gaussian mutation mainly enhances local exploitation capability through fine-grained perturbation, while the Cauchy mutation introduces long-distance jumps to help the algorithm escape local optima. By combining these two mutation operators adaptively, the SGCM strategy effectively improves the population diversity and robustness of AEDGRO during the later optimization stage.

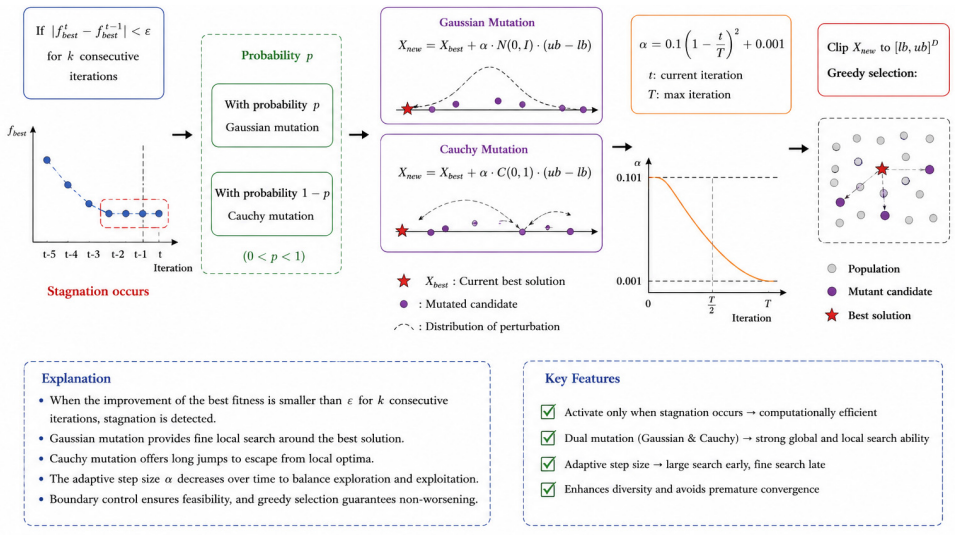


Figure 4. Schematic illustration of the Stagnation-Aware Gaussian–Cauchy Mutation Strategy (SGCM).

Therefore, the proposed SGCM strategy can effectively alleviate premature convergence and significantly improve the global optimization capability of AEDGRO on complex multimodal benchmark functions.

3.5. Overall Evolution Mechanism of AEDGRO

The overall optimization process of the proposed AEDGRO algorithm is illustrated in Algorithm 1. First, the Chaotic Good-Point Initialization Strategy (CGPI) is employed to initialize the population, which improves the diversity and distribution uniformity of individuals in the search space. After population initialization, the fitness values of all individuals are calculated, and the current global best solution is determined.

During the iterative optimization process, the original migration, mining, and collaboration mechanisms of GRO are executed to balance global exploration and local exploitation. Meanwhile, the proposed Adaptive Elite Differential Mining Strategy (AEDM) is incorporated into the mining phase to further enhance convergence speed and exploitation capability through elite guidance and adaptive differential mutation. In addition, when the algorithm falls into stagnation, the Stagnation-Aware Gaussian–Cauchy Mutation Strategy (SGCM) is activated to increase population diversity and help the algorithm escape local optima.

After generating new candidate solutions, greedy selection and boundary control are adopted to update the population and retain superior individuals. The iterative optimization process continues until the maximum number of iterations is reached, and the optimal solution obtained by AEDGRO is finally returned. By integrating the CGPI, AEDM, and SGCM strategies, AEDGRO achieves better convergence performance, stronger global search capability, and higher optimization accuracy than the original GRO algorithm.

Algorithm 1. Pseudocode of AEDGRO

Input: N, T, dim, lb, ub, and objective function $f(\cdot)$.
Output: Best fitness value f_{best}, and best solution X_{best}.
Initialize population using CGPI strategy.
Evaluate population fitness.
Determine global best solution.
while $t < T$ **do**
  for each individual X_i **do**
    Perform original GRO migration/mining/collaboration update.
    Apply AEDM strategy.
    Generate mutation vector V_i.
    Generate trial vector U_i.
    Boundary control
    Greedy selection
  end for
  Check stagnation condition.
  If stagnation occurs, apply SGCM strategy.
  Update global best solution.
end while
Return the global best solution X_{best} and its fitness f_{best}.

4. Experimental Design

4.1. Benchmark Configuration

To comprehensively evaluate the optimization capability and practical applicability of the proposed AEDGRO algorithm, both benchmark function experiments and UAV

path-planning experiments were conducted in this study. The benchmark experiments were performed using the IEEE CEC2017 test suite [43], which contains various unimodal, multimodal, hybrid, and composition functions. These benchmark functions are widely adopted to evaluate the convergence performance, exploration capability, exploitation ability, and robustness of optimization algorithms.

In the benchmark experiments, the dimensionality of all test functions was set to 30 dimensions. For all comparative approaches, the number of search agents was consistently initialized as 30, while the termination criterion was defined as 1000 iterative cycles. In order to minimize stochastic interference and improve the credibility of the experimental evaluation, every method was repeatedly performed 30 independent runs on each benchmark problem. The optimization performance was quantitatively assessed by the mean objective value and the corresponding standard deviation obtained from all runs, and the most competitive outcomes in each case were emphasized using bold formatting.

For the UAV path-planning experiments, a three-dimensional mountainous terrain environment was established using a terrain elevation model combined with Gaussian mountain peaks and threat regions. The map size was set to 200×200, and the start point and destination point of the UAV were defined as (0,0,0) and (200,200,30), respectively. Circular threat regions were introduced to simulate dangerous flight areas that the UAV must avoid during the path-planning process.

In the UAV path-planning task, the number of intermediate waypoints between the start and end points was set to 2, resulting in a decision vector dimension of 4. Cubic spline interpolation was employed to construct smooth continuous trajectories. The maximum number of iterations was set to 1000, the population size was fixed at 30, and each algorithm was independently executed 30 times to ensure fairness and stability of the experimental results.

All numerical simulations were performed on an iMac (Retina 5K, 27-inch, 2020; Apple Inc., Cupertino, CA, USA) equipped with a 3.6 GHz 10-core Intel® Core™ i9 processor and 128 GB DDR4 RAM running macOS Monterey. The entire experimental implementation and algorithmic verification processes were carried out using MATLAB R2024b as the computational environment.

4.2. Comparative Algorithms

To objectively evaluate the optimization performance of the proposed AEDGRO algorithm, several well-known metaheuristic optimization algorithms were selected for comparison in both benchmark function optimization and UAV path-planning tasks. The selected algorithms include classical population-based optimization methods, recently proposed swarm intelligence algorithms, and several GRO-based variants.

The compared algorithms are as follows:

- Particle Swarm Optimization (PSO) [7].
- Differential Evolution (DE) [8].
- Grey Wolf Optimizer (GWO) [9].
- Dhole Optimization Algorithm (DOA) [17].
- Secretary Bird Optimization Algorithm (SBOA) [18].
- Hannibal Barca optimizer (HBO) [19].
- Gold Rush Optimizer (GRO) [20].
- Gold Rush Optimizer (AGRO) [21].
- Quantum-Enhanced Gold Rush Optimizer (QHGRO) [22].

Among these algorithms, PSO, DE, and GWO are representative classical optimization algorithms widely used in engineering optimization problems due to their simple structures and strong optimization capabilities. DOA, SBOA, and HBO are recently proposed swarm

intelligence algorithms that demonstrate competitive performance on complex optimization tasks. GRO is the original baseline algorithm of the proposed method, while AGRO and QHGRO are improved variants of GRO designed to enhance convergence behavior and population diversity.

For fairness, the parameter settings of all compared algorithms were selected according to their original references. The configuration details of all benchmark algorithms are presented in Table 1. Moreover, all algorithms were executed under the same experimental conditions, including identical population size, maximum iteration number, search boundaries, and independent run times. The same objective function and UAV path-planning environment were used for all compared algorithms to ensure the objectivity and reliability of the experimental results.

Table 1. Parameter configurations used in the compared algorithms.

Optimization Method	Control Variables	Assigned Settings
PSO	c_1, c_2, w	2, 2, 0.8
DE	p_{cr}, F	0.8, 0.8
GWO	a	[0, 2]
DOA	$C1, C3, \mu, K$	1, 3, 25, 0.5
SBOA	CF, K, RB	$[0,1], \{1, 2\}, [0,1]$
HBO	$Coef$	0.94
GRO	m	$[0,1]$
AGRO	m	$[0,1]$
QHGRO	m	$[0,1]$
AEDGRO		

Through comparisons with both classical and advanced optimization algorithms, the effectiveness and superiority of AEDGRO in terms of convergence speed, optimization accuracy, population diversity, and UAV path-planning capability can be comprehensively validated.

4.3. UAV Path-Planning Experiments

4.3.1. Experimental Environment Setup

To further verify the practical applicability of the proposed AEDGRO algorithm, UAV path-planning experiments were conducted in a three-dimensional mountainous environment. The terrain model was generated by combining sinusoidal fluctuation functions and Gaussian mountain peaks, enabling the environment to contain complex topographic variations and obstacle distributions. The size of the map was set to 200×200, and the terrain surface was constructed using the proposed elevation model described in Section 2.

To simulate dangerous flight environments, circular threat regions were introduced into the terrain. The center position of the threat region was set to (50,140), and the corresponding threat radius was fixed at 30. During the optimization process, the UAV trajectory was required to avoid all threat regions while remaining above the terrain surface to ensure flight safety.

The start point and end point of the UAV were defined as $P_s = (0, 0, 0)$ and $P_e = (200, 200, 30)$ respectively. Two intermediate waypoints were inserted between the start point and destination point, resulting in a decision vector dimension of 4. Cubic spline interpolation was employed to construct smooth continuous trajectories between the waypoints.

For all competing methods, the number of individuals in the population was consistently initialized to 30, and the iterative termination threshold was specified as 1000 generations. To alleviate the impact of stochastic factors on the experimental outcomes, each

algorithm was performed through 30 separate independent trials. The UAV path quality was evaluated using the proposed fitness model, which simultaneously considers path length, altitude variation, and trajectory smoothness.

4.3.2. Analysis of UAV Experiment Results

The three-dimensional trajectories and top-view projections of the UAV flight paths generated by different algorithms are presented in Figures 5 and 6, respectively. The corresponding fitness convergence curves are illustrated in Figure 7, while the statistical results are summarized in Table 2.

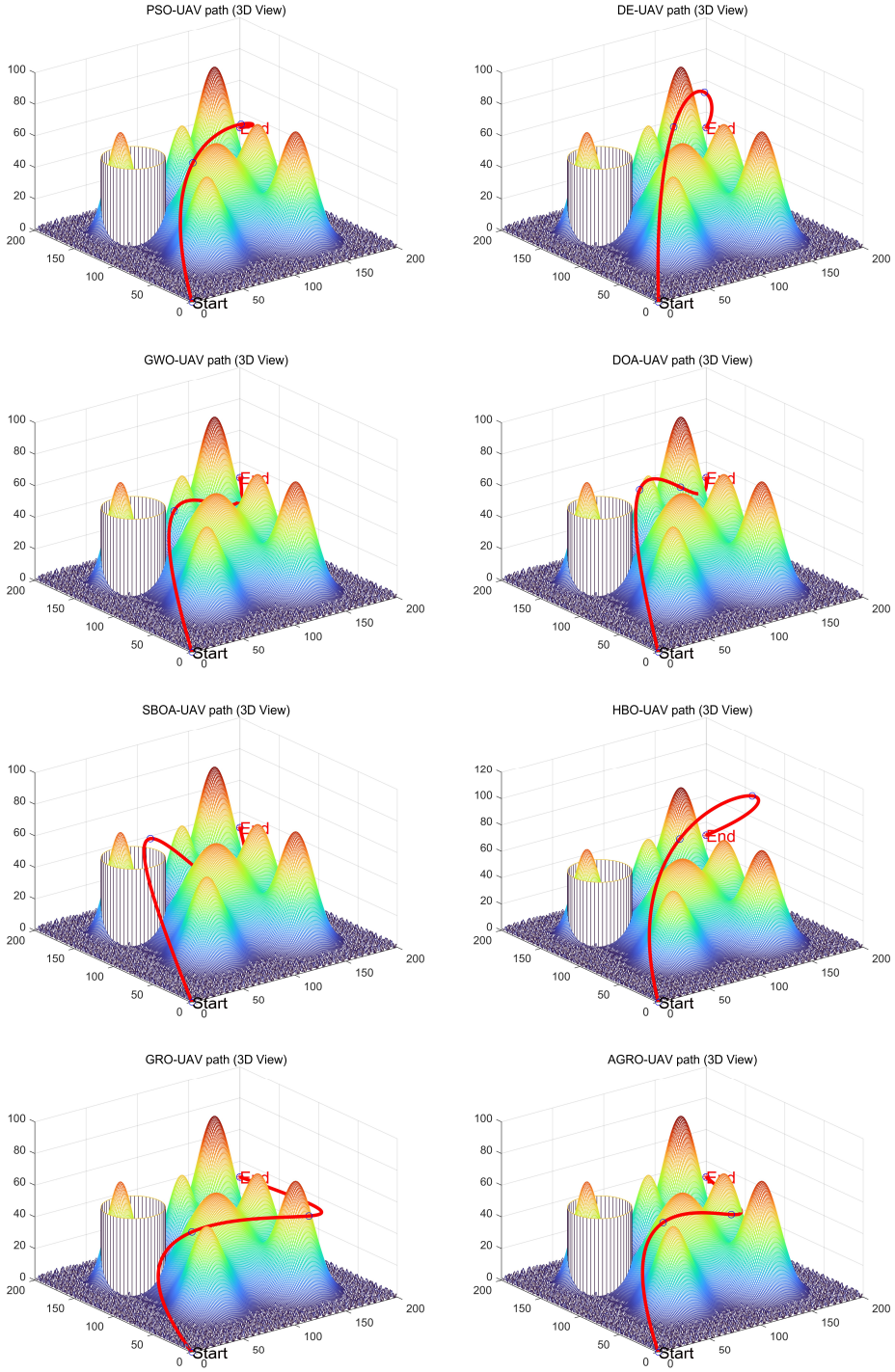


Figure 5. *Cont.*

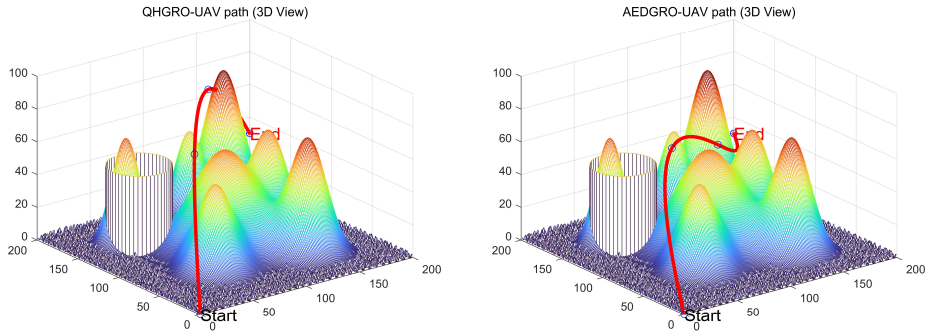


Figure 5. Three-dimensional UAV trajectory generated by different algorithms in the mountainous environment.

Table 2. Statistical performance comparison of different optimization algorithms in UAV path planning.

Algorithm	Mean	Std	Best	Worst	Median	Run Time	Friedman	Friedman Rank
PSO	286.41	63.66	231.11	401.47	231.40	53.56	4.70	4
DE	255.19	34.34	232.19	353.87	242.42	108.09	5.60	7
GWO	363.12	57.41	233.29	394.40	392.43	50.00	8.10	9
DOA	342.55	76.39	231.10	403.64	392.99	97.50	7.10	8
SBOA	310.79	75.70	231.10	392.32	347.29	112.10	5.25	6
HBO	366.74	82.98	231.10	489.80	395.86	123.74	8.10	10
GRO	254.72	47.66	231.12	348.38	231.49	65.63	4.20	3
AGRO	265.04	52.04	231.10	347.16	231.10	294.02	4.15	2
QHGRO	288.75	73.77	231.10	403.17	231.21	67.17	4.95	5
AEDGRO	235.69	7.55	231.09	260.41	231.77	65.28	2.85	1

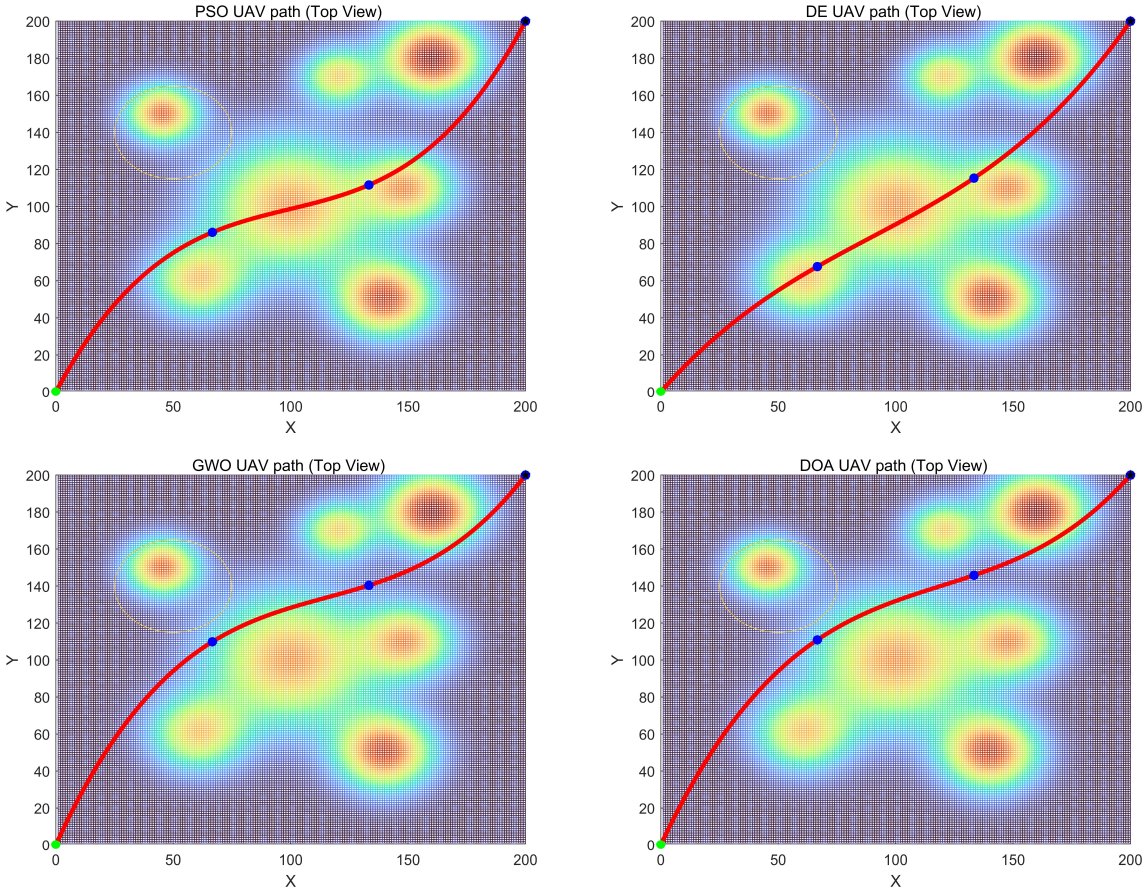


Figure 6. *Cont.*

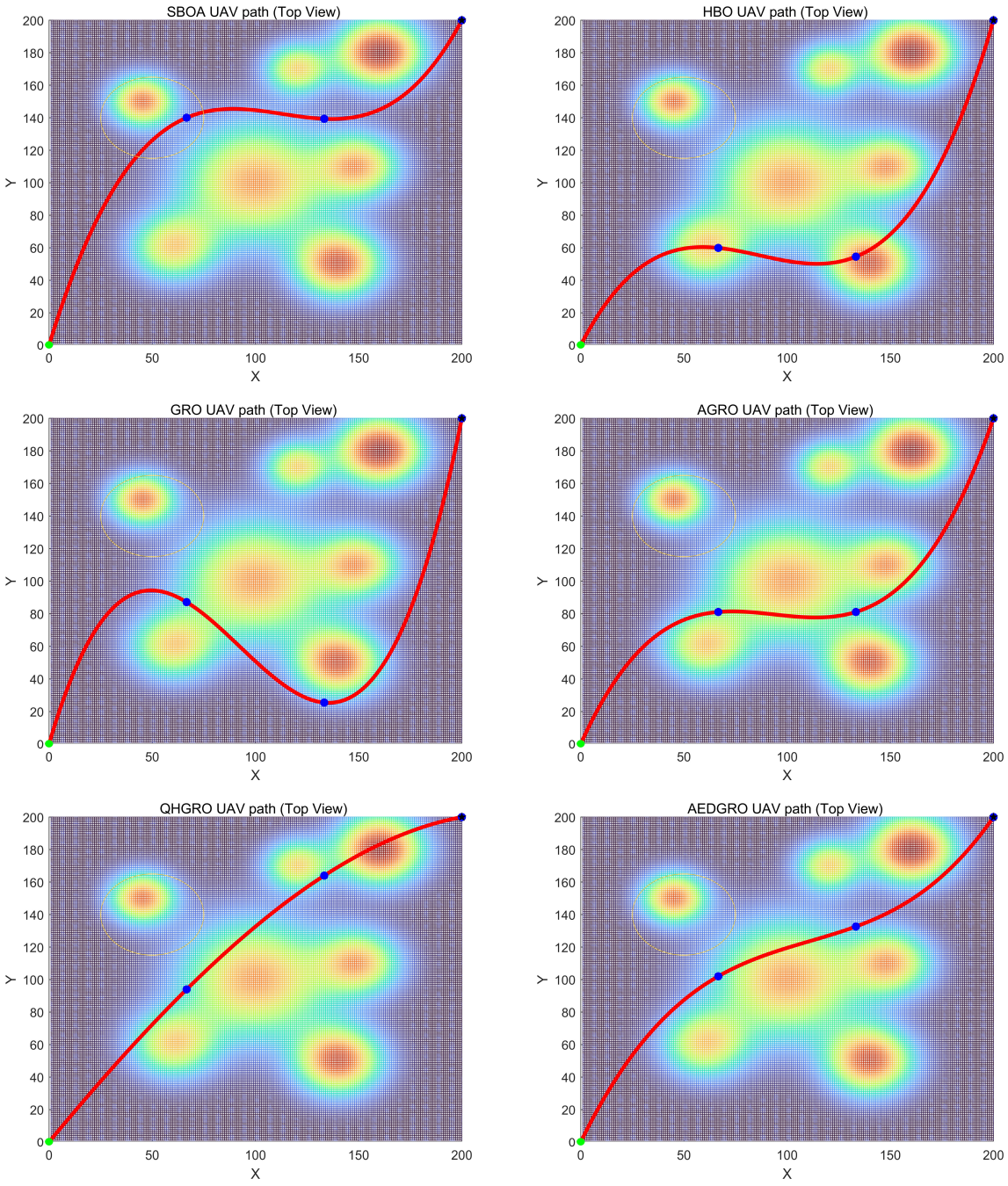


Figure 6. Top-view projection of the UAV trajectory generated by different algorithms.

Figure 5 presents the three-dimensional flight paths obtained by various optimization approaches within the complex mountainous terrain scenario for Unmanned Aerial Vehicle navigation. As shown in the figure, the experimental environment contains multiple mountain peaks with varying elevations, while several cylindrical threat regions are also introduced, resulting in a highly undulating terrain structure. For UAV path planning, the algorithm must not only identify a feasible route between the starting point and the destination, but also avoid collisions with mountainous terrain and threat regions. As illustrated in Figure 5, all compared methods are able to construct valid navigation routes that successfully connect the initial position to the designated destination point; however, noticeable differences exist in the resulting path shapes. Some algorithms produce trajectories with obvious detours or local oscillations near mountainous areas, indicating

limited search stability and path adjustment capability in complex terrains. In contrast, the trajectory generated by AEDGRO is generally smoother, allowing the UAV to naturally bypass high-altitude regions and threat areas while maintaining a continuous and stable flight trend. This demonstrates that AEDGRO possesses superior terrain adaptability in three-dimensional environments and can achieve more desirable path configurations while ensuring flight safety.

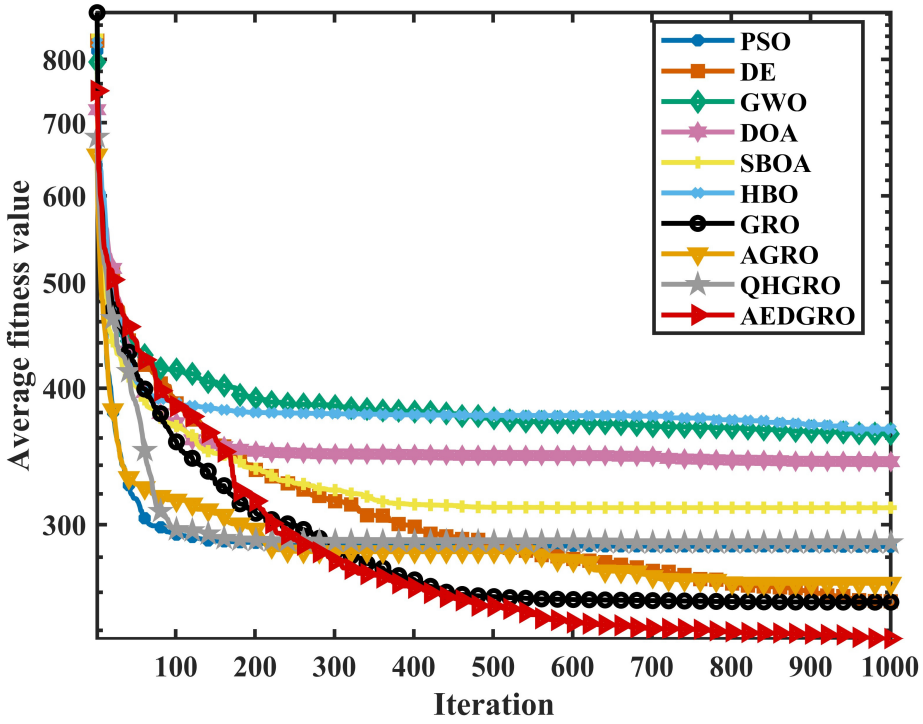


Figure 7. Average convergence curves of different optimization algorithms in the UAV path-planning problem.

Further observation of Figure 5 reveals that the path generated by AEDGRO does not exhibit abrupt turns or excessive altitude fluctuations when approaching mountainous regions or threat areas. Instead, obstacle avoidance is accomplished through relatively smooth trajectory transitions. Such path characteristics are particularly important for practical UAV operations because excessive turning maneuvers or sharp altitude changes may increase flight control difficulty and energy consumption. The trajectory generated by AEDGRO effectively balances flight safety and path continuity, indicating that the proposed method achieves a favorable trade-off among path length, altitude variation, and turning smoothness.

Figure 6 presents the two-dimensional top-view projections of the paths generated by different algorithms. Compared with the three-dimensional visualization, the top-view projection more clearly reflects the horizontal distribution of the UAV trajectories and the relative positional relationship between the paths and the threat regions. As shown in Figure 6, the threat regions are distributed on one side of the search space, imposing constraints on the horizontal flight routes of the UAV. Some algorithms exhibit significant path deviations or redundant detours when approaching the threat regions, implying that part of the path economy is sacrificed during the obstacle avoidance process. By contrast, the path generated by AEDGRO successfully avoids the threat regions while maintaining relatively continuous heading variations, without obvious backtracking or unnecessary detours. These results indicate that AEDGRO is capable of not only satisfying terrain safety

requirements in the three-dimensional altitude direction but also effectively accomplishing threat avoidance in the two-dimensional horizontal plane.

In addition, Figure 6 shows that the trajectory generated by AEDGRO progresses steadily from the starting point toward the target point with a relatively compact path distribution, suggesting that the algorithm can effectively control path deviation during the search process. Since cubic spline interpolation is adopted in this study to construct continuous trajectories, the positions of intermediate nodes directly influence the smoothness and feasibility of the final flight path. AEDGRO is able to obtain a more reasonable distribution of intermediate nodes, allowing the interpolated trajectory to maintain strong continuity and further improving the smoothness of the flight path. This also demonstrates the superior search accuracy of AEDGRO in node optimization.

Figure 7 presents the mean convergence profiles of the compared optimization methods when applied to the path-planning problem for unmanned aerial vehicle navigation. It can be observed that all algorithms rapidly reduce the fitness value during the early stages of iteration, indicating that they possess a certain degree of initial search capability. However, significant differences emerge in the later stages of convergence. Algorithms such as GWO, DOA, and HBO tend to stabilize prematurely, with only limited subsequent improvement in fitness values, implying a certain degree of search stagnation in this path-planning task. GRO and AGRO exhibit relatively better convergence performance, although their final convergence levels remain inferior to that of AEDGRO. In comparison, AEDGRO maintains a stable downward convergence trend throughout the entire optimization process and ultimately achieves the lowest fitness value, indicating that the proposed algorithm retains the ability to continuously improve path quality during the later optimization stages.

Table 2 reports the statistical results of different algorithms over 30 independent runs. As shown in the table, AEDGRO achieves the lowest average fitness value of 235.69 among all compared algorithms, indicating that it can obtain superior overall path costs. Meanwhile, the standard deviation of AEDGRO is 7.55, which is significantly lower than those of PSO, DE, GWO, DOA, SBOA, HBO, GRO, AGRO, and QHGRO, demonstrating stronger stability and robustness across multiple independent runs. In terms of the best result, AEDGRO achieves a Best value of 231.09, which is slightly better than those of the competing algorithms. Regarding the worst-case performance, the Worst value of AEDGRO is 260.41, which remains considerably lower than those of most comparison methods, indicating that AEDGRO can still maintain satisfactory path-planning quality even under relatively unfavorable optimization conditions. Furthermore, AEDGRO achieves the best Friedman ranking value of 2.85, ranking first among all compared algorithms, which further confirms its superior overall optimization performance.

In terms of computational time, AEDGRO requires 65.28 s, which is comparable to GRO (65.63 s) and QHGRO (67.17 s), while being substantially lower than AGRO (294.02 s). Although the computational cost of AEDGRO is marginally greater than that of GWO (50.00 s) and PSO (53.56 s), its remarkable superiority in optimization quality, particularly with respect to the mean fitness results, demonstrates that the additional computational overhead is reasonable and worthwhile, standard deviation, worst-case performance, and Friedman ranking, the additional computational cost remains within a reasonable range. Therefore, AEDGRO achieves a favorable balance among path quality, convergence performance, stability, and computational efficiency.

Overall, AEDGRO demonstrates excellent comprehensive optimization performance in the UAV path-planning experiments. This can be mainly attributed to the synergistic cooperation of the three proposed improvement strategies. Specifically, the Chaotic Good-Point Initialization Strategy (CGPI) combines good-point sets with chaotic mapping to

improve the distribution uniformity and spatial coverage capability of the initial population, enabling the algorithm to more thoroughly explore feasible path regions during the early optimization stage. The Adaptive Elite Differential Mining Strategy (AEDM) employs elite guidance and adaptive differential perturbation mechanisms to enhance local exploitation capability and convergence efficiency in promising path regions. Meanwhile, the Stagnation-Aware Gaussian–Cauchy Mutation Strategy (SGCM) dynamically introduces Gaussian perturbations and Cauchy long-jump mechanisms when the optimization process stagnates, thereby enhancing population diversity and reducing the risk of being trapped in local optima. Consequently, through the collaborative integration of these three strategies, AEDGRO achieves a more effective balance between global exploration and local exploitation, enabling it to generate more stable, safer, and smoother UAV flight paths in complex mountainous environments.

4.4. Benchmark Function Validation

To further evaluate the global optimization capability of the proposed AEDGRO algorithm, experiments were conducted on the IEEE CEC2017 benchmark suite [43]. The CEC2017 benchmark functions contain various unimodal, multimodal, hybrid, and composition optimization problems, which provide a comprehensive platform for evaluating the convergence performance, exploration capability, exploitation ability, and robustness of optimization algorithms.

During the experimental evaluation, all benchmark problems were implemented in a 30-dimensional search space. For consistency across all comparative methods, the swarm size was initialized to 30 individuals, while the maximum evolutionary process was limited to 1000 iterations. To minimize stochastic disturbances and maintain an equitable comparison framework, each optimization algorithm was repeatedly conducted 30 independent times on every benchmark function. The obtained results were statistically assessed using the mean objective outcome and the corresponding standard deviation, and the most competitive values in each category were distinguished through bold formatting.

4.4.1. Quantitative Comparison Results

To comprehensively assess the effectiveness of the proposed AEDGRO approach in handling challenging numerical optimization tasks, a series of experiments were performed on the 30-dimensional CEC2017 Benchmark Suite test set. This benchmark collection contains multiple categories of functions, including unimodal, multimodal, hybrid, and composition problems, thereby providing a diversified evaluation framework for examining the global search performance, local refinement capability, convergence reliability, and overall robustness of optimization algorithms under different optimization scenarios. Therefore, this benchmark suite provides a more comprehensive assessment of the overall optimization performance of AEDGRO. The experimental results are presented in Table 3 and Figures 8–11.

Table 3 reports the statistical results of different algorithms on the 30-dimensional CEC2017 benchmark functions. It can be observed that AEDGRO achieves competitive optimization performance on most benchmark functions, particularly on complex multimodal, hybrid, and composition functions. Specifically, AEDGRO obtains the best average results among all compared algorithms on functions F2, F7, F9, F12, F16, F17, F20, F23, F24, F26, F28, and F29, demonstrating its strong global exploration capability and effective exploitation ability when handling complex nonlinear search spaces.

Table 3. Statistical comparison results of different algorithms on the 30-dimensional CEC2017 benchmark functions.

Function	Metric	PSO	DE	GWO	DOA	SBOA	HBO	GRO	AGRO	QHGRO	AEDGRO
F1	Ave	8.1891×10^{8}	1.2749×10^{5}	2.6164×10^{9}	4.0632×10^{3}	5.9916×10^{3}	2.3711×10^{8}	4.5666×10^{7}	2.9587×10^{3}	5.8689×10^{3}	3.9452×10^{3}
	Std	4.6618×10^{8}	5.1187×10^{4}	1.6185×10^{9}	5.5210×10^{3}	6.1726×10^{3}	1.7230×10^{8}	3.9874×10^{7}	4.0640×10^{3}	6.5756×10^{3}	3.8204×10^{3}
F2	Ave	7.7755×10^{30}	7.6808×10^{24}	2.5455×10^{32}	7.9180×10^{17}	1.1846×10^{13}	1.8424×10^{27}	6.8612×10^{26}	3.5967×10^{17}	1.0590×10^{13}	2.3334×10^{12}
	Std	2.6913×10^{31}	1.7945×10^{25}	6.9968×10^{32}	4.3298×10^{18}	2.4992×10^{13}	8.0474×10^{27}	2.2113×10^{27}	1.8636×10^{18}	2.4040×10^{13}	5.8077×10^{12}
F3	Ave	2.6577×10^{4}	1.1165×10^{5}	5.0682×10^{4}	4.6867×10^{4}	1.0792×10^{4}	4.0517×10^{4}	4.2889×10^{4}	4.8809×10^{3}	1.6030×10^{4}	8.0616×10^{3}
	Std	9.6228×10^{3}	1.5379×10^{4}	1.2648×10^{4}	1.8313×10^{4}	4.8746×10^{3}	1.0895×10^{4}	8.6668×10^{3}	3.0446×10^{3}	6.8894×10^{3}	2.8821×10^{3}
F4	Ave	7.3412×10^{2}	5.6969×10^{2}	6.3999×10^{2}	4.8234×10^{2}	4.8993×10^{2}	5.7415×10^{2}	5.3371×10^{2}	4.8032×10^{2}	4.9251×10^{2}	4.9642×10^{2}
	Std	2.8804×10^{2}	1.2442×10	1.4759×10^{2}	2.6642×10	4.5357×10	4.5051×10	2.2727×10	9.9132×10^{0}	2.5901×10	2.3908×10
F5	Ave	7.0252×10^{2}	6.5615×10^{2}	6.1555×10^{2}	7.2217×10^{2}	5.8032×10^{2}	6.4314×10^{2}	5.8843×10^{2}	5.6146×10^{2}	6.2648×10^{2}	5.5497×10^{2}
	Std	3.3564×10	1.2305×10	3.3821×10	6.6203×10	2.7843×10	2.3837×10	1.9452×10	1.8347×10	3.0671×10	1.5291×10
F6	Ave	6.1711×10^{2}	6.0024×10^{2}	6.1232×10^{2}	6.3552×10^{2}	6.0251×10^{2}	6.2007×10^{2}	6.0980×10^{2}	6.0306×10^{2}	6.1523×10^{2}	6.0009×10^{2}
	Std	6.3273×10^{0}	4.4157×10^{-2}	4.5040×10^{0}	1.6887×10	1.9419×10^{0}	6.9944×10^{0}	3.3257×10^{0}	2.7177×10^{0}	7.0362×10^{0}	1.3333×10^{-1}
F7	Ave	9.7030×10^{2}	8.9929×10^{2}	9.0869×10^{2}	1.0919×10^{3}	8.4122×10^{2}	9.5596×10^{2}	8.6450×10^{2}	8.1283×10^{2}	9.6374×10^{2}	7.8360×10^{2}
	Std	3.0856×10	1.9948×10	5.1425×10	1.0115×10^{2}	3.5311×10	7.0812×10	6.5877×10	2.1818×10	5.4348×10	1.3877×10
F8	Ave	9.9015×10^{2}	9.5993×10^{2}	8.9696×10^{2}	9.7865×10^{2}	8.7841×10^{2}	9.3728×10^{2}	8.8909×10^{2}	8.6059×10^{2}	9.2101×10^{2}	8.5951×10^{2}
	Std	2.7151×10	1.5222×10	1.9593×10	4.9243×10	1.6472×10	2.9444×10	1.7419×10	1.1752×10	3.0025×10	1.2622×10
F9	Ave	1.6254×10^{3}	2.2606×10^{3}	2.0187×10^{3}	4.0044×10^{3}	1.1220×10^{3}	4.2392×10^{3}	1.5791×10^{3}	1.0855×10^{3}	3.4797×10^{3}	9.1561×10^{2}
	Std	1.2116×10^{3}	3.5053×10^{2}	6.5558×10^{2}	1.6722×10^{3}	2.4183×10^{2}	1.2227×10^{3}	3.9947×10^{2}	1.3848×10^{2}	1.0587×10^{3}	1.6301×10
F10	Ave	7.1377×10^{3}	6.2768×10^{3}	4.9442×10^{3}	7.8076×10^{3}	3.8996×10^{3}	4.7579×10^{3}	4.6651×10^{3}	5.7616×10^{3}	5.0143×10^{3}	3.9692×10^{3}
	Std	7.3674×10^{2}	3.2318×10^{2}	1.1584×10^{3}	7.7092×10^{2}	7.7266×10^{2}	6.9397×10^{2}	6.2405×10^{2}	6.1025×10^{2}	6.4140×10^{2}	5.5474×10^{2}
F11	Ave	1.4296×10^{3}	1.8430×10^{3}	2.3386×10^{3}	1.2927×10^{3}	1.2025×10^{3}	1.4420×10^{3}	1.2888×10^{3}	1.2016×10^{3}	1.1967×10^{3}	1.1712×10^{3}
	Std	4.9205×10	3.5393×10^{2}	1.0911×10^{3}	6.9404×10	4.8707×10	1.1501×10^{2}	5.2032×10	3.1728×10	3.6893×10	3.5429×10
F12	Ave	6.1443×10^{7}	1.1001×10^{7}	7.8451×10^{7}	3.9870×10^{5}	6.0917×10^{5}	1.7674×10^{7}	1.4308×10^{6}	1.1857×10^{6}	7.4666×10^{5}	1.8754×10^{5}
	Std	9.3339×10^{7}	2.5153×10^{6}	1.1852×10^{8}	2.9526×10^{5}	5.3388×10^{5}	2.9280×10^{7}	1.2636×10^{6}	1.0182×10^{6}	7.0192×10^{5}	1.6234×10^{5}
F13	Ave	2.0535×10^{6}	1.4088×10^{6}	9.6413×10^{6}	2.5134×10^{4}	2.7446×10^{4}	1.8642×10^{5}	2.7617×10^{4}	4.5603×10^{3}	2.0480×10^{4}	1.5848×10^{4}
	Std	1.6030×10^{6}	5.4568×10^{5}	2.6241×10^{7}	2.5513×10^{4}	2.2017×10^{4}	8.1707×10^{5}	1.6653×10^{4}	3.4385×10^{3}	1.7736×10^{4}	1.4395×10^{4}
F14	Ave	6.6959×10^{4}	3.4253×10^{5}	4.3356×10^{5}	2.0943×10^{4}	2.1015×10^{4}	4.2016×10^{5}	3.1290×10^{4}	2.6276×10^{3}	7.7893×10^{4}	3.7381×10^{3}
	Std	6.4788×10^{4}	2.4941×10^{5}	5.2491×10^{5}	2.2447×10^{4}	2.1053×10^{4}	4.4333×10^{5}	3.2769×10^{4}	2.1421×10^{3}	6.6743×10^{4}	2.0492×10^{3}
F15	Ave	5.5440×10^{4}	4.0889×10^{5}	9.2337×10^{5}	6.6442×10^{3}	1.0614×10^{4}	1.0790×10^{4}	5.7941×10^{3}	3.8881×10^{3}	1.1063×10^{4}	5.2871×10^{3}
	Std	3.1420×10^{4}	2.0798×10^{5}	1.5456×10^{6}	6.5243×10^{3}	9.9178×10^{3}	1.0831×10^{4}	3.6180×10^{3}	2.3825×10^{3}	1.0376×10^{4}	5.1098×10^{3}
F16	Ave	2.7957×10^{3}	2.7075×10^{3}	2.6054×10^{3}	2.9978×10^{3}	2.2325×10^{3}	2.7502×10^{3}	2.3812×10^{3}	2.3527×10^{3}	2.7404×10^{3}	2.1922×10^{3}
	Std	2.8588×10^{2}	1.7867×10^{2}	3.2185×10^{2}	3.9376×10^{2}	2.7925×10^{2}	3.2986×10^{2}	2.4597×10^{2}	2.2861×10^{2}	2.8270×10^{2}	2.4226×10^{2}
F17	Ave	2.0957×10^{3}	2.0978×10^{3}	2.0487×10^{3}	2.3711×10^{3}	1.9458×10^{3}	2.3535×10^{3}	1.8824×10^{3}	1.9133×10^{3}	2.1315×10^{3}	1.8272×10^{3}
	Std	1.4781×10^{2}	1.0660×10^{2}	1.5984×10^{2}	2.0688×10^{2}	1.0181×10^{2}	2.7930×10^{2}	1.0422×10^{2}	1.1887×10^{2}	1.6461×10^{2}	8.2897×10
F18	Ave	1.8219×10^{6}	1.0476×10^{6}	1.7398×10^{6}	2.2822×10^{5}	3.1246×10^{5}	2.9325×10^{6}	3.3718×10^{5}	6.6653×10^{4}	8.6687×10^{5}	1.9950×10^{5}
	Std	1.4915×10^{6}	6.0463×10^{5}	1.8986×10^{6}	2.6471×10^{5}	3.7763×10^{5}	2.6666×10^{6}	3.6937×10^{5}	9.8756×10^{4}	9.1074×10^{5}	1.7425×10^{5}
F19	Ave	8.9830×10^{4}	2.5759×10^{5}	5.4314×10^{6}	8.8432×10^{3}	9.4069×10^{3}	2.4099×10^{4}	6.8167×10^{3}	4.7948×10^{3}	9.1321×10^{3}	7.9840×10^{3}
	Std	1.4900×10^{5}	1.5731×10^{5}	2.1645×10^{7}	6.8986×10^{3}	1.0745×10^{4}	6.0805×10^{4}	6.3700×10^{3}	2.5156×10^{3}	9.1605×10^{3}	7.7080×10^{3}
F20	Ave	2.4816×10^{3}	2.3624×10^{3}	2.4676×10^{3}	2.7034×10^{3}	2.2428×10^{3}	2.5358×10^{3}	2.2963×10^{3}	2.3447×10^{3}	2.5388×10^{3}	2.1712×10^{3}
	Std	1.6749×10^{2}	8.5107×10	1.6756×10^{2}	1.7862×10^{2}	1.1584×10^{2}	1.6327×10^{2}	7.4175×10	1.2734×10^{2}	2.2706×10^{2}	7.1382×10

Table 3. *Cont.*

Function	Metric	PSO	DE	GWO	DOA	SBOA	HBO	GRO	AGRO	QHGRO	AEDGRO
F21	Ave	2.4928×10^{3}	2.4587×10^{3}	2.3969×10^{3}	2.5005×10^{3}	2.3574×10^{3}	2.4486×10^{3}	2.3751×10^{3}	2.3574×10^{3}	2.4079×10^{3}	2.3493×10^{3}
	Std	3.5020×10	9.8615×10^{0}	2.0376×10	4.4866×10	1.6692×10	3.1584×10	3.3939×10	1.9548×10	3.7174×10	1.3785×10
F22	Ave	4.9336×10^{3}	2.9645×10^{3}	4.6021×10^{3}	5.0162×10^{3}	2.4425×10^{3}	4.2739×10^{3}	2.3450×10^{3}	2.3043×10^{3}	3.7849×10^{3}	2.6003×10^{3}
	Std	3.0105×10^{3}	1.2606×10^{2}	1.7151×10^{3}	3.3243×10^{3}	7.7358×10^{2}	2.0954×10^{3}	2.5719×10	1.6972×10	2.0468×10^{3}	9.1608×10^{2}
F23	Ave	2.9109×10^{3}	2.8032×10^{3}	2.7807×10^{3}	2.8773×10^{3}	2.7166×10^{3}	2.8098×10^{3}	2.7368×10^{3}	2.7156×10^{3}	2.7808×10^{3}	2.6955×10^{3}
	Std	6.9456×10	1.4420×10	4.6364×10	6.5955×10	1.8171×10	5.1591×10	2.7451×10	1.8434×10	3.7816×10	1.4120×10
F24	Ave	3.0730×10^{3}	3.0167×10^{3}	2.9474×10^{3}	3.0152×10^{3}	2.8902×10^{3}	2.9857×10^{3}	2.9025×10^{3}	2.8788×10^{3}	2.9478×10^{3}	2.8694×10^{3}
	Std	6.2385×10	1.3695×10	6.3798×10	8.0974×10	1.8202×10	4.3958×10	2.4900×10	1.7236×10	3.6692×10	1.2662×10
F25	Ave	2.9692×10^{3}	2.9214×10^{3}	3.0184×10^{3}	2.8963×10^{3}	2.9003×10^{3}	2.9595×10^{3}	2.9431×10^{3}	2.8880×10^{3}	2.8951×10^{3}	2.8900×10^{3}
	Std	6.7148×10	1.1813×10	6.7450×10	1.7669×10	2.0194×10	2.8970×10	1.9489×10	1.6522×10	1.7109×10	1.0381×10
F26	Ave	5.2955×10^{3}	4.7272×10^{3}	4.9020×10^{3}	4.8659×10^{3}	3.9677×10^{3}	4.8043×10^{3}	4.1401×10^{3}	4.3146×10^{3}	5.1638×10^{3}	3.7143×10^{3}
	Std	1.0157×10^{3}	5.1688×10^{2}	4.8549×10^{2}	1.5636×10^{3}	7.8061×10^{2}	1.3416×10^{3}	7.5355×10^{2}	7.5719×10^{2}	8.1063×10^{2}	6.1931×10^{2}
F27	Ave	3.2663×10^{3}	3.2469×10^{3}	3.2823×10^{3}	3.2487×10^{3}	3.2165×10^{3}	3.2580×10^{3}	3.2524×10^{3}	3.1910×10^{3}	3.2516×10^{3}	3.2119×10^{3}
	Std	4.4719×10	6.8002×10^{0}	3.9827×10	2.4271×10	1.2697×10	2.0021×10	1.5314×10	1.2909×10	2.9505×10	7.7915×10^{0}
F28	Ave	3.4074×10^{3}	3.3014×10^{3}	3.4389×10^{3}	3.2263×10^{3}	3.2250×10^{3}	3.3497×10^{3}	3.3080×10^{3}	3.2329×10^{3}	3.2167×10^{3}	3.2140×10^{3}
	Std	1.9555×10^{2}	1.2889×10	7.9669×10	2.5336×10	2.6091×10	5.4520×10	4.3442×10	2.2460×10	1.9237×10	1.4728×10
F29	Ave	4.0482×10^{3}	3.8126×10^{3}	3.9280×10^{3}	4.0091×10^{3}	3.5791×10^{3}	3.9808×10^{3}	3.7946×10^{3}	3.5736×10^{3}	3.8830×10^{3}	3.5223×10^{3}
	Std	1.8635×10^{2}	1.0816×10^{2}	1.9905×10^{2}	2.4972×10^{2}	1.8645×10^{2}	2.6288×10^{2}	1.4713×10^{2}	2.3359×10^{2}	2.5149×10^{2}	1.3123×10^{2}
F30	Ave	1.1307×10^{6}	4.5154×10^{5}	9.5294×10^{6}	1.2713×10^{4}	1.3977×10^{4}	1.0948×10^{5}	4.2883×10^{4}	5.3187×10^{3}	9.7006×10^{3}	1.0541×10^{4}
	Std	7.7487×10^{5}	2.0107×10^{5}	7.0043×10^{6}	4.1829×10^{3}	8.7502×10^{3}	2.0122×10^{5}	3.3798×10^{4}	2.5259×10^{3}	3.2879×10^{3}	3.2448×10^{3}

For several highly challenging functions, such as F2, F12, and F30, the performance differences among algorithms are particularly significant. Algorithms such as PSO, GWO, and HBO exhibit relatively large average fitness values on these functions, indicating that they are more likely to suffer from local stagnation or insufficient search precision in high-dimensional complex search spaces. In contrast, AEDGRO achieves lower fitness values while maintaining relatively small standard deviations. For example, on F9, AEDGRO achieves an average value of 9.1561×10^2 with a standard deviation of only 1.6301×10^1, which is significantly better than those of the compared algorithms. This demonstrates that AEDGRO can not only obtain superior solutions but also maintain strong optimization stability.

Furthermore, Table 3 shows that AEDGRO generally achieves relatively small standard deviation values on most benchmark functions, indicating strong robustness over multiple independent runs. In comparison, although some algorithms may obtain competitive results on several functions, their relatively large standard deviations suggest insufficient optimization stability. For instance, DOA, HBO, and QHGRO exhibit noticeable fluctuations on certain functions, whereas AEDGRO maintains stable optimization performance while preserving superior solution quality.

Figure 8 presents the mean convergence trajectories of the compared optimization methods on several representative functions from the 30-dimensional CEC2017 Benchmark Suite. As shown in the figure, all algorithms exhibit a certain degree of fitness reduction during the early stages of iteration; however, significant differences emerge in their later-stage convergence performance. AEDGRO maintains a continuous downward convergence trend on most benchmark functions and eventually reaches lower fitness values. By contrast, some algorithms gradually converge prematurely during the middle and late optimization stages, with only marginal subsequent improvement, indicating a tendency to become trapped in local optimal regions. Particularly on complex multimodal and composition functions, the convergence curves of AEDGRO are generally smoother, suggesting that the proposed algorithm can effectively balance global exploration and local exploitation during the optimization process.

Further observation of Figure 8 reveals that AEDGRO typically achieves rapid fitness reduction during the early optimization stage, indicating that the Chaotic Good-Point Initialization Strategy (CGPI) improves the distribution quality of the initial population and enables the algorithm to identify promising search regions more efficiently. During the later optimization stage, AEDGRO still maintains a certain degree of convergence capability, demonstrating that the Adaptive Elite Differential Mining Strategy (AEDM) and the Stagnation-Aware Gaussian–Cauchy Mutation Strategy (SGCM) effectively enhance local exploitation ability while reducing the risk of premature convergence to local optima.

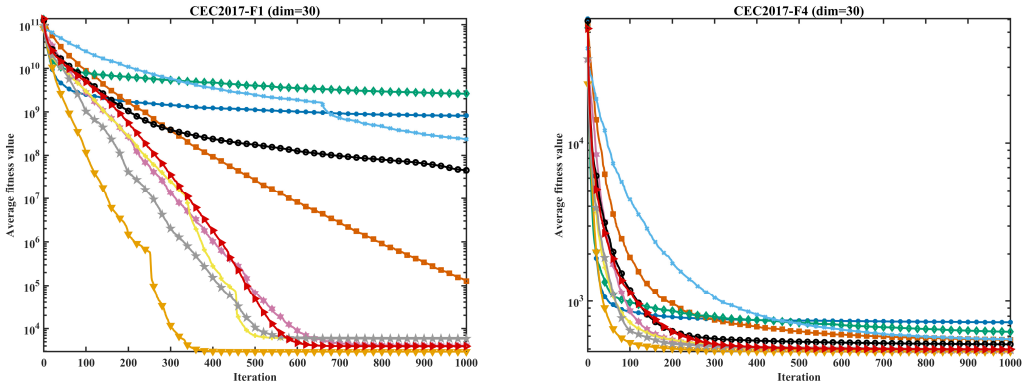


Figure 8. *Cont.*

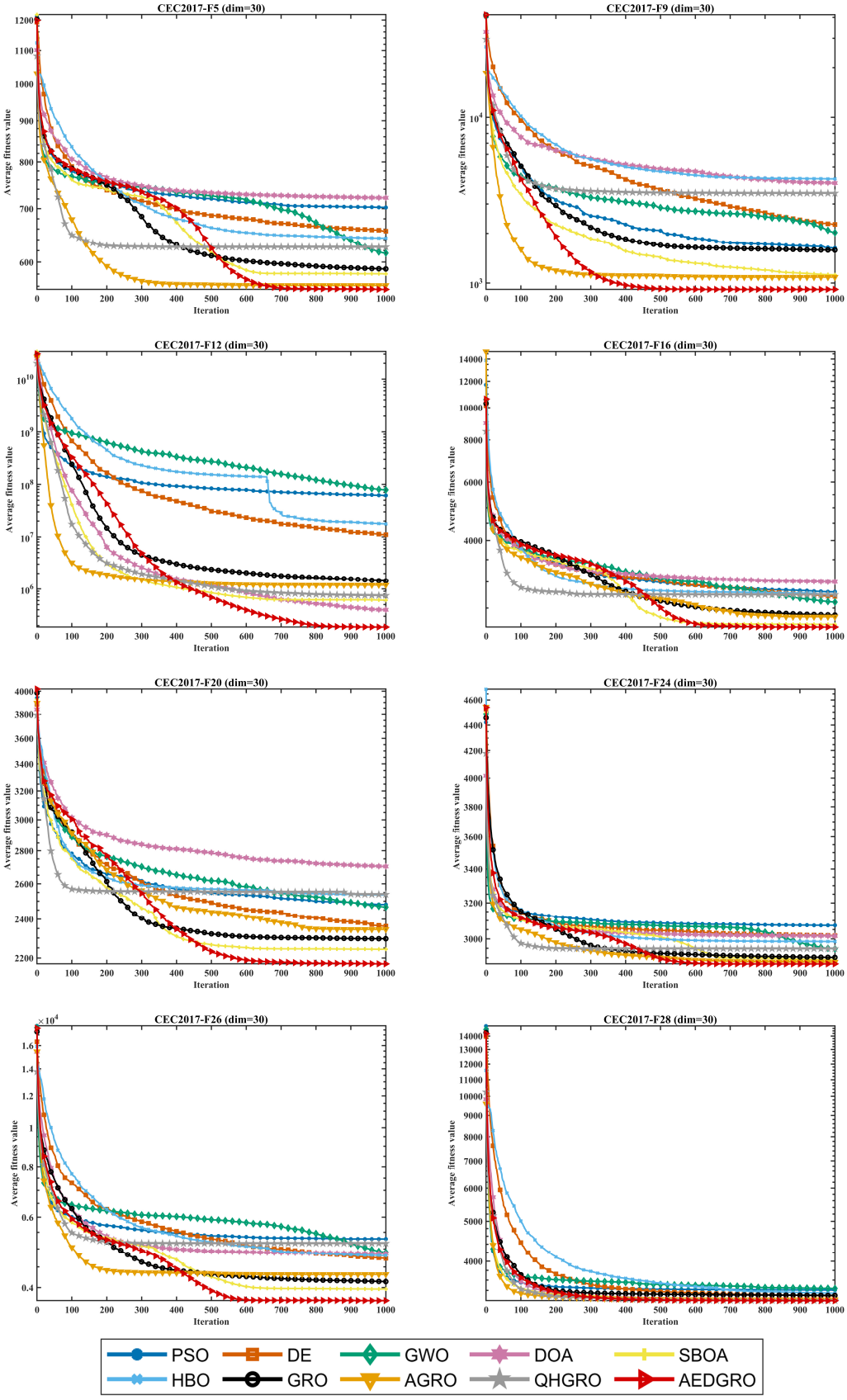


Figure 8. Convergence curves of different algorithms on the 30-dimensional CEC2017 benchmark function.

Figure 9 presents the violin plots of different algorithms on representative benchmark functions. These plots provide a more intuitive visualization of the result distributions

obtained from multiple independent runs. The distribution of AEDGRO is generally more concentrated, with relatively narrow violin widths, indicating strong stability and consistency across repeated runs. In contrast, some algorithms exhibit wider violin distributions and even noticeable long-tail phenomena, implying larger performance fluctuations and stronger sensitivity to randomness. In addition, the median positions of AEDGRO are generally lower than those of the comparison algorithms, further demonstrating that AEDGRO can consistently obtain superior solutions in most runs rather than relying on a few occasional high-quality results.

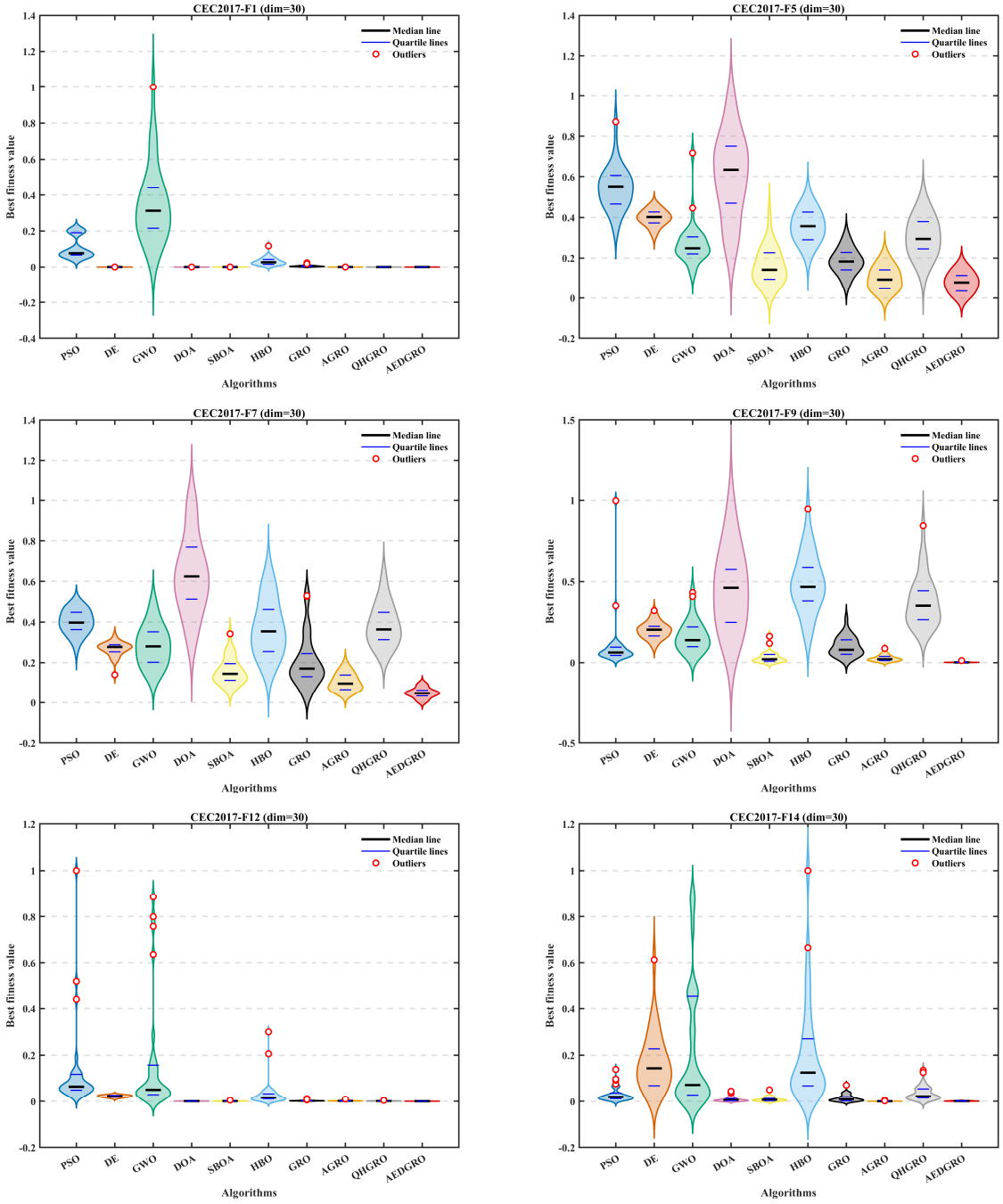


Figure 9. *Cont.*

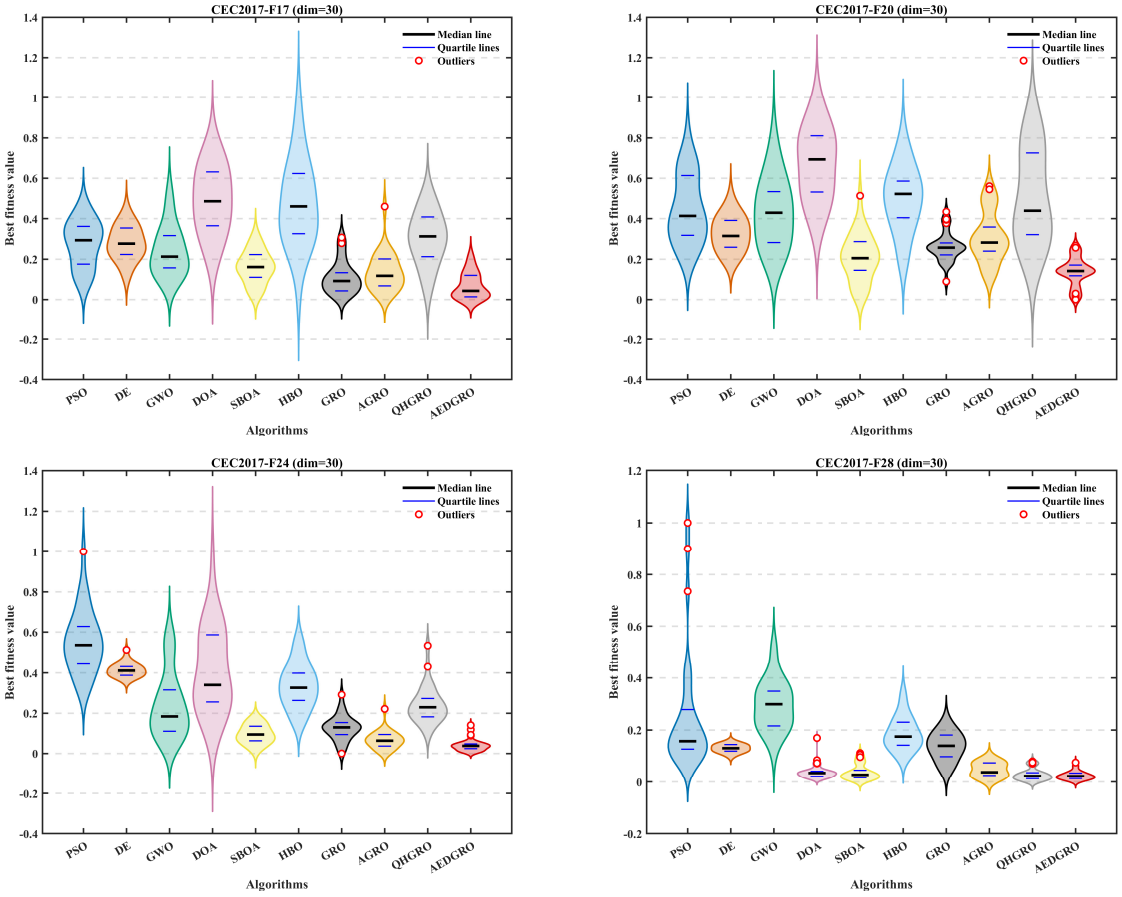


Figure 9. Violin plot of fitness distributions for different algorithms on the 30-dimensional CEC2017 benchmark function.

Figure 10 presents the parallel coordinate ranking results of different algorithms on the 30-dimensional CEC2017 benchmark functions. This figure provides a more intuitive visualization of the overall ranking variations of different algorithms across all benchmark functions. As shown in the figure, AEDGRO generally maintains relatively low ranking positions on most test functions, while its ranking curve exhibits comparatively small fluctuations, indicating that the proposed algorithm can maintain stable optimization performance across different types of benchmark problems. In particular, on several complex multimodal, hybrid, and composition functions, AEDGRO achieves significantly better rankings than most competing algorithms, demonstrating its strong global exploration capability and local exploitation ability in complex search spaces.

In contrast, the ranking curves of PSO, GWO, DOA, and HBO fluctuate considerably, with obvious variations across different benchmark functions, indicating that these algorithms are highly dependent on problem characteristics and may suffer from performance degradation on certain complex functions. Among them, HBO and GWO frequently appear in relatively lower ranking regions on multiple functions, suggesting insufficient stability when handling high-dimensional complex optimization problems. AGRO, GRO, and QHGRO achieve comparatively competitive overall rankings, indicating that the GRO-based improvement mechanisms can effectively enhance optimization performance. Nevertheless, compared with these GRO variants, the ranking curve of AEDGRO is generally smoother and remains in more favorable ranking positions on most benchmark functions, further demonstrating that the three proposed improvement strategies can effectively improve the comprehensive optimization capability and problem adaptability of the algorithm.

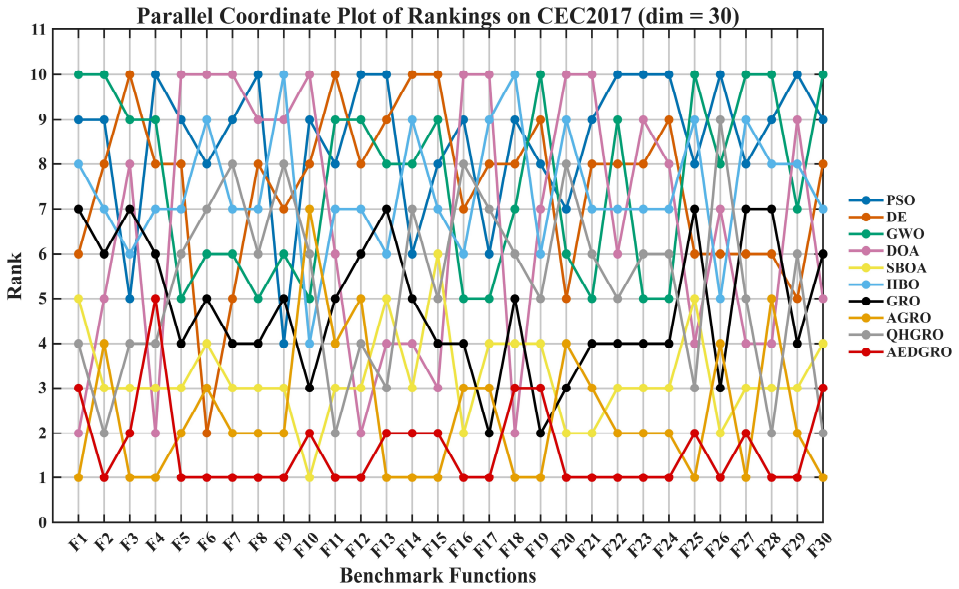


Figure 10. Ranking distribution of different algorithms on each function of the CEC2017 benchmark suite (A smaller rank indicates better algorithmic performance).

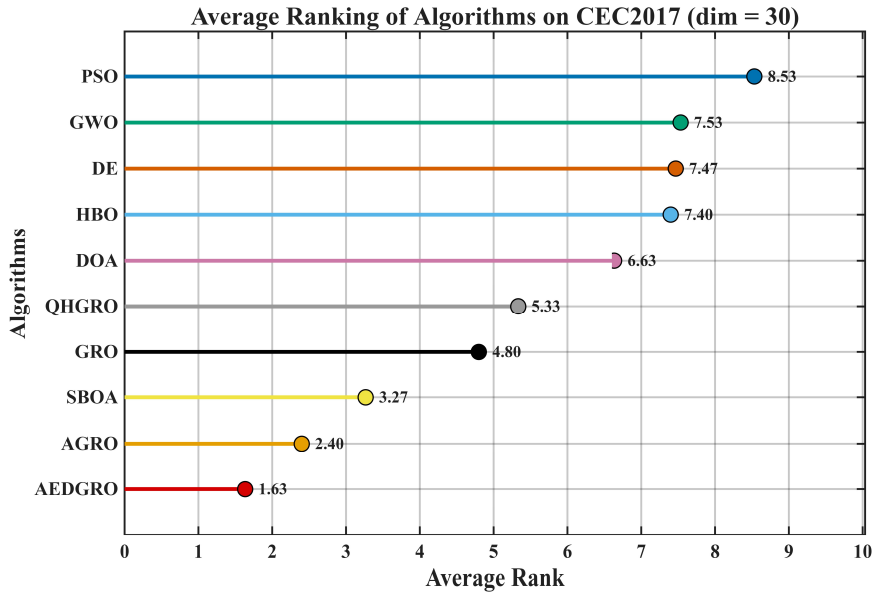


Figure 11. Lollipop chart of Friedman average rankings for different algorithms on the CEC2017 benchmark suite.

Furthermore, Figure 10 shows that AEDGRO consistently maintains top-ranking positions across the majority of the CEC2017 benchmark functions. Specifically, the rankings of AEDGRO are predominantly concentrated within the top three positions, and it achieves the first rank on a considerable number of test functions. Unlike several competing algorithms whose rankings fluctuate significantly between high and low positions depending on the problem characteristics, AEDGRO exhibits only minor ranking variations throughout the benchmark suite. This observation indicates that the proposed algorithm not only excels on a few individual functions but also delivers consistently superior performance across unimodal, multimodal, hybrid, and composition functions. The concentration of rankings within the top three positions further demonstrates the strong optimization capability, robustness, and adaptability of AEDGRO when dealing with diverse and complex optimization landscapes. Therefore, Figure 10 provides additional evidence that the proposed

CGPI, AEDM, and SGCM strategies effectively enhance both the exploration and exploitation abilities of the original GRO, enabling AEDGRO to achieve stable and competitive performance across a wide range of optimization problems.

Figure 11 presents the Friedman average ranking results of different algorithms on the 30-dimensional CEC2017 benchmark functions. The Friedman average ranking provides a comprehensive evaluation of the overall optimization performance of algorithms across all benchmark functions, where a smaller average ranking value indicates better overall performance. As shown in the figure, AEDGRO achieves the lowest Friedman average ranking value of 1.63, ranking first among all compared algorithms, which demonstrates its superior comprehensive optimization capability on the entire CEC2017 benchmark suite.

Further observation of Figure 11 reveals that AGRO achieves a Friedman average ranking of 2.40, ranking second, while SBOA obtains a ranking value of 3.27, ranking third. In comparison, GRO achieves a Friedman average ranking of 4.80, representing a moderate overall performance level. Compared with the original GRO, the average ranking of AEDGRO is reduced from 4.80 to 1.63, indicating that the proposed improvement strategies can significantly enhance the overall optimization performance of the algorithm. Moreover, AEDGRO further outperforms AGRO by achieving a lower average ranking value, demonstrating that the proposed CGPI, AEDM, and SGCM strategies provide more substantial advantages in enhancing population diversity, improving local exploitation capability, and reducing the risk of premature convergence to local optima.

In addition, QHGRO achieves a Friedman average ranking of 5.33, while DOA obtains 6.63. The Friedman average rankings of HBO, DE, and GWO are 7.40, 7.47, and 7.53, respectively, whereas PSO exhibits the highest average ranking value of 8.53. These results indicate that traditional algorithms and several comparison methods are significantly inferior to AEDGRO in handling complex high-dimensional optimization problems, suggesting that these algorithms are more likely to suffer from search stagnation or insufficient convergence accuracy when solving complex multimodal, hybrid, and composition functions.

From the perspective of the overall ranking distribution, AEDGRO maintains a noticeable advantage over the other compared algorithms, particularly when compared with PSO, GWO, DE, and HBO, where the superiority in Friedman average ranking becomes even more significant. This demonstrates that AEDGRO can not only achieve competitive results on several individual benchmark functions but also maintain consistently stable and superior optimization performance across the entire benchmark suite. Therefore, Figure 11 further verifies that AEDGRO possesses strong global exploration capability, local exploitation ability, and overall robustness when solving complex high-dimensional optimization problems.

Overall, the experimental results presented in Table 3 and Figures 8–11 demonstrate that AEDGRO exhibits strong comprehensive optimization capability on the 30-dimensional CEC2017 benchmark functions. The proposed algorithm not only achieves superior fitness values but also demonstrates strong convergence performance, optimization stability, and robustness. This can be mainly attributed to the synergistic cooperation of the CGPI, AEDM, and SGCM strategies, which enable AEDGRO to achieve a more effective balance between global exploration and local exploitation, thereby delivering superior optimization performance on complex high-dimensional optimization problems.

4.4.2. Statistical Analysis Based on the Wilcoxon Rank-Sum Method

To provide a more rigorous assessment of the performance discrepancies between AEDGRO and the competing optimization approaches, the Wilcoxon rank-sum statistical test was adopted to analyze the experimental outcomes obtained from the 30-dimensional CEC2017 Benchmark Suite. As a commonly utilized nonparametric hypothesis testing

technique, the Wilcoxon rank-sum method is capable of effectively identifying whether the observed differences between two optimization algorithms possess statistical significance. In the present investigation, the significance threshold was specified as 0.05 [44,45]. A p-value lower than 0.05 demonstrates that the performance gap between AEDGRO and the corresponding comparative method is statistically meaningful rather than caused by random fluctuations. The detailed statistical results are summarized in Table 4.

Table 4 presents the statistical outcomes of the Wilcoxon rank-sum comparisons conducted between AEDGRO and the remaining competing approaches across all benchmark problems. It can be clearly observed that the majority of the obtained p-values are far below the predefined significance threshold of 0.05, demonstrating that the optimization advantages achieved by AEDGRO on most test functions are statistically reliable and cannot be attributed merely to stochastic variations or accidental effects.

Further observation of the $(+/=/-)$ statistical results at the bottom of Table 4 reveals that AEDGRO achieves "30/0/0" against PSO, DE, and GWO across all 30 benchmark functions, indicating that AEDGRO outperforms these three algorithms on every benchmark function without any ties or inferior results. This demonstrates that traditional algorithms exhibit substantially weaker overall performance than AEDGRO on complex high-dimensional optimization problems, further confirming the superior global exploration capability and convergence accuracy of AEDGRO.

For HBO, AEDGRO achieves a "29/0/1" result, indicating that AEDGRO outperforms HBO on 29 benchmark functions and fails to outperform it on only one function. Compared with GRO, AEDGRO achieves a "27/0/3" result, demonstrating that AEDGRO performs better than the original GRO on the vast majority of benchmark functions. These results indicate that the proposed improvement strategies can effectively enhance the overall optimization performance of GRO, thereby further improving its convergence capability and stability in complex search spaces.

In addition, AEDGRO achieves "23/0/7" and "22/0/8" results against AGRO and QHGRO, respectively, indicating that although these improved GRO variants exhibit competitive performance on several benchmark functions, AEDGRO still maintains clear superiority on most test functions. Particularly on complex multimodal, hybrid, and composition functions, AEDGRO can more consistently obtain superior optimization results, demonstrating that the proposed CGPI, AEDM, and SGCM strategies can further enhance the algorithm's capability to maintain population diversity and strengthen local exploitation performance.

From the distribution of the p-values, it can be observed that the p-values corresponding to most benchmark functions reach the order of 10^{-11} or even smaller, particularly for functions such as F2, F6, F7, F9, F12, F20, F23, and F24. This indicates extremely significant differences between AEDGRO and most comparison algorithms. Meanwhile, on several functions such as F19, F28, and F30, the p-values between AEDGRO and certain comparison algorithms are relatively larger, suggesting that the performance gaps among algorithms on these functions are comparatively smaller. Nevertheless, from the overall statistical results, AEDGRO still maintains clear superiority on the vast majority of benchmark functions.

In summary, the Wilcoxon rank-sum statistical results listed in Table 4 provide additional evidence supporting the effectiveness of AEDGRO on the 30-dimensional CEC2017 Benchmark Suite problems. The proposed algorithm not only produces more competitive optimization outcomes on the majority of benchmark functions, but also demonstrates statistically convincing performance advantages over the compared approaches. These findings further confirm that the introduced enhancement mechanisms can substantially strengthen the optimization accuracy, convergence reliability, and overall robustness of the algorithm.

Table 4. Wilcoxon rank-sum statistical analysis results between AEDGRO and various comparative methods on the 30-dimensional CEC2017 Benchmark Suite test functions.

Function	PSO	DE	GWO	DOA	SBOA	HBO	GRO	AGRO	QHGRO
F1	3.01986×10^{-11}	3.01986×10^{-11}	3.01986×10^{-11}	6.73495×10^{-1}	1.05470×10^{-1}	3.01986×10^{-11}	3.01986×10^{-11}	2.00949×10^{-1}	4.55297×10^{-1}
F2	3.01986×10^{-11}	3.01986×10^{-11}	3.01986×10^{-11}	1.60621×10^{-6}	1.95268×10^{-3}	3.01986×10^{-11}	3.01986×10^{-11}	9.51394×10^{-6}	1.08690×10^{-1}
F3	4.97517×10^{-11}	3.01986×10^{-11}	3.01986×10^{-11}	3.01986×10^{-11}	2.92054×10^{-2}	3.01986×10^{-11}	3.01986×10^{-11}	1.10577×10^{-4}	5.18568×10^{-7}
F4	2.22727×10^{-9}	3.01986×10^{-11}	1.09367×10^{-10}	2.70863×10^{-2}	4.64273×10^{-1}	3.82016×10^{-10}	7.11859×10^{-9}	5.97056×10^{-5}	3.04177×10^{-1}
F5	3.01986×10^{-11}	3.01986×10^{-11}	1.09367×10^{-10}	3.01986×10^{-11}	1.49316×10^{-4}	3.01986×10^{-11}	4.31061×10^{-8}	2.28230×10^{-1}	7.38908×10^{-11}
F6	3.01986×10^{-11}	4.99795×10^{-9}	3.01986×10^{-11}	3.01986×10^{-11}	4.50432×10^{-11}	3.01986×10^{-11}	3.01986×10^{-11}	5.49405×10^{-11}	3.01986×10^{-11}
F7	3.01986×10^{-11}	3.01986×10^{-11}	3.01986×10^{-11}	3.01986×10^{-11}	2.60985×10^{-10}	3.01986×10^{-11}	2.22727×10^{-9}	2.02829×10^{-7}	3.01986×10^{-11}
F8	3.01986×10^{-11}	3.01986×10^{-11}	5.07231×10^{-10}	3.01986×10^{-11}	1.99628×10^{-5}	4.07716×10^{-11}	2.83145×10^{-8}	6.84323×10^{-1}	4.97517×10^{-11}
F9	3.01986×10^{-11}	3.01986×10^{-11}	3.01986×10^{-11}	3.01986×10^{-11}	9.75550×10^{-10}	3.01986×10^{-11}	3.01986×10^{-11}	1.46431×10^{-10}	3.01986×10^{-11}
F10	3.01986×10^{-11}	3.01986×10^{-11}	1.42984×10^{-5}	3.01986×10^{-11}	8.88303×10^{-1}	3.83067×10^{-5}	1.58461×10^{-4}	2.15440×10^{-10}	2.19589×10^{-7}
F11	3.01986×10^{-11}	3.01986×10^{-11}	3.01986×10^{-11}	4.18250×10^{-9}	7.95900×10^{-3}	3.01986×10^{-11}	1.46431×10^{-10}	1.00353×10^{-3}	6.66888×10^{-3}
F12	3.01986×10^{-11}	3.01986×10^{-11}	3.01986×10^{-11}	1.17376×10^{-3}	4.35308×10^{-5}	3.01986×10^{-11}	1.20567×10^{-10}	3.49711×10^{-9}	1.63506×10^{-5}
F13	3.01986×10^{-11}	3.01986×10^{-11}	8.99341×10^{-11}	2.17017×10^{-1}	3.26509×10^{-2}	1.83680×10^{-2}	5.08422×10^{-3}	7.28836×10^{-3}	2.28230×10^{-1}
F14	1.20567×10^{-10}	3.01986×10^{-11}	5.49405×10^{-11}	1.02773×10^{-6}	4.80107×10^{-7}	3.01986×10^{-11}	2.19589×10^{-7}	3.33861×10^{-3}	1.20567×10^{-10}
F15	6.69552×10^{-11}	3.01986×10^{-11}	4.07716×10^{-11}	3.71077×10^{-1}	1.27321×10^{-2}	3.33861×10^{-3}	1.18817×10^{-1}	7.84460×10^{-1}	3.51366×10^{-2}
F16	6.72195×10^{-10}	1.69472×10^{-9}	1.38525×10^{-6}	1.07018×10^{-9}	6.73495×10^{-1}	3.35195×10^{-8}	1.12278×10^{-2}	1.17107×10^{-2}	2.01522×10^{-8}
F17	4.99795×10^{-9}	2.60985×10^{-10}	4.31061×10^{-8}	3.68973×10^{-11}	1.74791×10^{-5}	2.37147×10^{-10}	1.76490×10^{-2}	1.11426×10^{-3}	3.19674×10^{-9}
F18	6.12104×10^{-10}	1.07018×10^{-9}	4.80107×10^{-7}	8.07275×10^{-1}	1.41278×10^{-1}	3.49711×10^{-9}	1.02326×10^{-1}	2.49131×10^{-6}	2.59736×10^{-5}
F19	1.28704×10^{-9}	3.01986×10^{-11}	5.49405×10^{-11}	3.25527×10^{-1}	7.17189×10^{-1}	3.63222×10^{-1}	7.50587×10^{-1}	2.33989×10^{-1}	4.37641×10^{-1}
F20	1.46431×10^{-10}	5.57265×10^{-10}	3.15889×10^{-10}	3.01986×10^{-11}	5.08422×10^{-3}	1.61323×10^{-10}	8.35200×10^{-8}	2.19589×10^{-7}	1.61323×10^{-10}
F21	3.01986×10^{-11}	3.01986×10^{-11}	2.60985×10^{-10}	3.01986×10^{-11}	7.24456×10^{-2}	3.01986×10^{-11}	1.06657×10^{-7}	1.25970×10^{-1}	2.37147×10^{-10}
F22	6.51827×10^{-9}	1.06657×10^{-7}	2.66947×10^{-9}	4.80107×10^{-7}	3.50117×10^{-3}	3.82489×10^{-9}	1.06657×10^{-7}	4.35835×10^{-2}	1.38525×10^{-6}
F23	3.01986×10^{-11}	3.01986×10^{-11}	4.07716×10^{-11}	3.01986×10^{-11}	3.83067×10^{-5}	3.01986×10^{-11}	4.68563×10^{-8}	2.77257×10^{-5}	4.97517×10^{-11}
F24	3.01986×10^{-11}	3.01986×10^{-11}	8.10136×10^{-10}	4.97517×10^{-11}	7.73869×10^{-6}	3.01986×10^{-11}	3.52006×10^{-7}	8.68437×10^{-3}	6.06576×10^{-11}
F25	1.20567×10^{-10}	2.03380×10^{-9}	3.01986×10^{-11}	3.64389×10^{-2}	2.49939×10^{-3}	6.06576×10^{-11}	1.20567×10^{-10}	1.44122×10^{-2}	5.39510×10^{-1}
F26	2.67842×10^{-6}	9.53321×10^{-7}	1.54652×10^{-9}	1.37033×10^{-3}	1.91124×10^{-2}	3.50117×10^{-3}	1.76490×10^{-2}	2.38848×10^{-4}	2.92155×10^{-9}
F27	8.99341×10^{-11}	3.33839×10^{-11}	6.69552×10^{-11}	1.20567×10^{-10}	7.01266×10^{-2}	3.68973×10^{-11}	6.06576×10^{-11}	8.10136×10^{-10}	8.89099×10^{-10}
F28	3.33839×10^{-11}	3.01986×10^{-11}	3.01986×10^{-11}	8.31461×10^{-3}	1.18817×10^{-1}	3.01986×10^{-11}	1.09367×10^{-10}	8.56412×10^{-4}	8.53382×10^{-1}
F29	1.20567×10^{-10}	7.11859×10^{-9}	1.69472×10^{-9}	6.72195×10^{-10}	5.10598×10^{-1}	3.49711×10^{-9}	3.64589×10^{-8}	4.91783×10^{-1}	2.83145×10^{-8}
F30	3.01986×10^{-11}	3.01986×10^{-11}	3.01986×10^{-11}	5.55457×10^{-2}	1.33454×10^{-1}	2.60151×10^{-8}	1.54652×10^{-9}	1.10234×10^{-8}	2.77189×10^{-1}
(+/=/−)	(30/0/0)	(30/0/0)	(30/0/0)	(24/0/6)	(19/0/11)	(29/0/1)	(27/0/3)	(23/0/7)	(22/0/8)

4.4.3. Computational Time Analysis

To further evaluate the computational efficiency of the proposed AEDGRO algorithm, this section records the total runtime of different algorithms on the 30-dimensional CEC2017 benchmark functions. The corresponding results are presented in Figure 12. Runtime can directly reflect the computational cost required during the optimization process and therefore serves as an important indicator for evaluating the practical applicability of optimization algorithms in complex real-world problems.

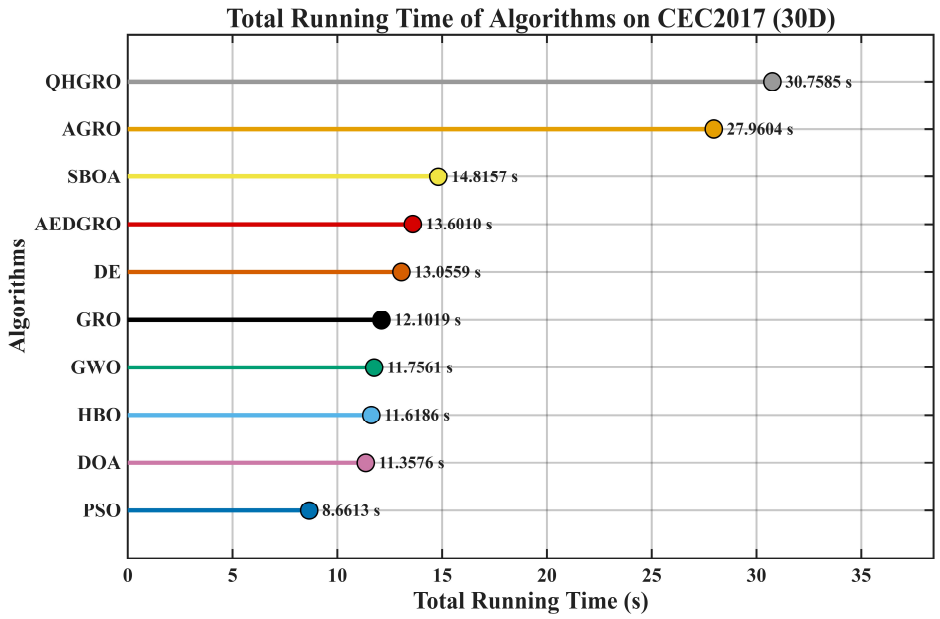


Figure 12. Runtime comparison results of various optimization algorithms on the 30-dimensional CEC2017 Benchmark Suite functions.

As shown in Figure 12, noticeable differences exist in the runtime among different algorithms, which are mainly related to algorithmic structural complexity, population update mechanisms, and additional search strategies. Among all compared algorithms, PSO achieves the shortest total runtime of only 8.6613 s, indicating that its update mechanism is relatively simple and computational complexity is relatively low, thereby providing high execution efficiency. The runtimes of DOA, HBO, and GWO are 11.3576 s, 11.6186 s, and 11.7561 s, respectively, also demonstrating relatively fast computational speeds. In comparison, GRO requires a total runtime of 12.1019 s, which is slightly higher than those of the aforementioned algorithms, mainly due to its more sophisticated population search mechanism.

The total runtime of AEDGRO is 13.6010 s, which is slightly higher than those of GRO, GWO, and PSO, but still remains within a reasonable range overall. Compared with the original GRO, AEDGRO increases the runtime by only approximately 1.5 s, indicating that although the proposed improvement strategies introduce additional computational overhead, they do not significantly reduce the execution efficiency of the algorithm. This is mainly because the Chaotic Good-Point Initialization Strategy (CGPI), Adaptive Elite Differential Mining Strategy (AEDM), and Stagnation-Aware Gaussian–Cauchy Mutation Strategy (SGCM) improve the search capability of the algorithm without introducing excessively complicated additional computations, thereby maintaining the overall computational cost at a relatively low level.

Further observation of Figure 12 reveals that the runtimes of AGRO and QHGRO are significantly higher than those of the other algorithms. Specifically, AGRO requires a total runtime of 27.9604 s, while QHGRO exhibits the highest runtime of 30.7585 s. In comparison,

the runtime of AEDGRO is substantially lower than those of these two improved GRO variants, indicating that AEDGRO can more effectively control computational cost while simultaneously improving optimization performance. Therefore, from the perspective of computational efficiency, AEDGRO achieves a favorable balance between optimization performance and computational overhead.

In addition, when combined with the experimental results presented in Table 3 and Figures 8–11, it can be observed that although the runtime of AEDGRO is slightly higher than those of several traditional algorithms, it exhibits clear advantages in terms of convergence performance, optimization accuracy, stability, and Friedman average ranking. Therefore, the additional computational cost introduced by AEDGRO is both reasonable and acceptable. Overall, Figure 12 demonstrates that AEDGRO can significantly enhance comprehensive optimization performance while maintaining relatively high computational efficiency, indicating strong potential for practical engineering applications.

Although several algorithms require less computational time than AEDGRO, computational efficiency alone cannot fully reflect the practical value of a UAV path-planning algorithm. The main goal of this study is to obtain a safe, smooth, and low-cost path while maintaining acceptable computational cost. As shown in the experimental results, AEDGRO achieves the best path fitness, the smallest standard deviation, and the best Friedman ranking. Its runtime is only slightly higher than that of some simple algorithms, such as PSO and GWO, but it is comparable to GRO and QHGRO and much lower than AGRO. Therefore, the additional computational cost introduced by CGPI, AEDM, and SGCM is acceptable because it brings significant improvements in solution quality, robustness, and path stability. This indicates that AEDGRO achieves a better trade-off between computational cost and optimization performance.

5. Conclusions and Future Work

This paper proposed an Adaptive Elite Differential Gold Rush Optimizer for three-dimensional UAV path planning in complex mountainous environments. Compared with the original GRO, AEDGRO introduces three targeted mechanisms: chaotic good-point initialization, adaptive elite differential mining, and stagnation-aware Gaussian–Cauchy mutation. The CEC2017 benchmark results show that AEDGRO obtains the best Friedman average ranking of 1.63, clearly outperforming GRO with a ranking of 4.80. In the UAV path-planning task, AEDGRO achieves the lowest mean fitness value of 235.69 and the smallest standard deviation of 7.55 among all compared algorithms. Compared with GRO, the mean fitness value is reduced from 254.72 to 235.69, indicating an improvement of approximately 7.47%. Compared with AGRO, AEDGRO also greatly reduces the computational time from 294.02 s to 65.28 s while obtaining a better path cost. These quantitative results demonstrate that AEDGRO improves not only optimization accuracy but also path-planning stability and computational practicality.

Despite these advantages, the proposed method still has several limitations. First, the current UAV path-planning model mainly focuses on static mountainous environments and predefined threat regions. Dynamic obstacles and unknown obstacles are only discussed from an extension perspective and have not yet been fully verified in real-time experiments. Second, the UAV dynamics are simplified by using safety clearance, turning angle, and climb angle constraints, while a complete nonlinear UAV dynamic model is not included. Third, the proposed AEDGRO is validated in numerical simulation, and real UAV experiments or hardware-in-the-loop tests remain necessary. Finally, although AEDGRO achieves a good balance between path quality and computational cost, the additional strategies still introduce a certain computational overhead compared with some simpler optimizers.

Future work will focus on four directions. First, AEDGRO will be extended to dynamic and unknown environments by integrating online replanning, obstacle prediction, and sensor-based map updating. Second, nonlinear UAV dynamics and more advanced trajectory-tracking controllers will be incorporated to improve real flight feasibility. Third, hardware-in-the-loop simulation and real UAV platform experiments will be conducted to evaluate real-time applicability. Fourth, multi-objective and distributed versions of AEDGRO will be developed to simultaneously optimize path length, energy consumption, risk, flight time, and coordination among multiple UAVs.

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

Funding: This research was funded by the Outstanding Young Teacher Cultivation Project of the Department of Education of Anhui Province (Grant No. YQYB2024169), the Key Research Projects in Natural Sciences for Colleges and Universities of Anhui Province (Grant No. 2024AH050529), and the Key Teaching Research Project of Anhui Business College (Grant No. ZL2024003).

Data Availability Statement: All data generated or analyzed during this study are included in this published article.

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

References

1. Yu, M.; Yang, H.; Wei, X.; Fu, S.; Xu, J.; Zhang, J. Carbon Footprint Efficiency Ratio: A Unified Indicator for Green Evaluation of Evolutionary Algorithms. *IEEE Trans. Emerg. Top. Comput. Intell.* **2026**, 1–14. [CrossRef]
2. Jiang, X.; Kuroiwa, T.; Cao, Y.; Sun, L.; Zhang, H.; Kawaguchi, T.; Hashimoto, S. Enhanced Pure Pursuit Path Tracking Algorithm for Mobile Robots Optimized by NSGA-II with High-Precision GNSS Navigation. *Sensors* **2025**, *25*, 745. [CrossRef]
3. Fu, K.; Guo, Y.; Qiu, W. Optimization of Microgrid Scheduling Based on Adaptive Collaborative Secretary Bird Optimization Algorithm. *Mathematics* **2026**, *14*, 622. [CrossRef]
4. Liu, Y.; Yuan, B.; Yuan, S. Improved Secretary Bird Optimization Algorithm Based on Financial Investment Strategy for Global Optimization and Real Application Problems. *Symmetry* **2026**, *18*, 688. [CrossRef]
5. Yu, M.; Zhang, J.; Yu, L.; Yang, H.; Lu, H.; Wei, X.; Xu, J. Alzheimer's disease brain image segmentation using multi-feature fusion in 3D Rényi entropy model and quantum hybrid optimization. *Artif. Intell. Rev.* **2026**, *59*, 95. [CrossRef]
6. Yu, M.; Xu, J.; Liang, W.; Qiu, Y.; Bao, S.; Tang, L. Improved multi-strategy adaptive Grey Wolf Optimization for practical engineering applications and high-dimensional problem solving. *Artif. Intell. Rev.* **2024**, *57*, 277. [CrossRef]
7. Kennedy, J.; Eberhart, R. Particle swarm optimization. In *Proceedings of the ICNN'95-International Conference on Neural Networks*; IEEE: Piscataway, NJ, USA, 1995; Volume 4, pp. 1942–1948.
8. Storn, R.; Price, K. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **1997**, *11*, 341–359.
9. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
10. Deng, B. Baboon optimization algorithm: A novel nature-inspired metaheuristic algorithm for optimization problems. *Ain Shams Eng. J.* **2026**, *17*, 104178. [CrossRef]
11. Sofi Karim, A.O.; Rahman, C.M. Trip road optimization algorithm for optimizing engineering problems. *Neural Comput. Appl.* **2026**, *38*, 227. [CrossRef]
12. Eltamaly, A.M.; Rabie, A.H. A star-nosed mole optimizer (SNMO): A novel optimization algorithm. *Comput. Electr. Eng.* **2026**, *135*, 111179. [CrossRef]
13. Rodan, A.; Al-Tamimi, A.-K.; Al-Alnemer, L.; Mirjalili, S.; Tiňo, P. Enzyme action optimizer: A novel bio-inspired optimization algorithm. *J. Supercomput.* **2025**, *81*, 686. [CrossRef]
14. Xu, X. Crocodile Ambush Optimization Algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems. *Array* **2025**, *28*, 100529. [CrossRef]
15. Yu, M.; Yang, H.; Zhang, J.; Ouyang, K.; Fu, S.; Tan, P.; Jiang, F.; Xu, J. Bounty hunter optimizer: A novel metaheuristic with an application to multi-UAV mobile edge computing and path planning. *Knowl.-Based Syst.* **2026**, *341*, 115836. [CrossRef]

16. Barshandeh, S.; Khodadadi, N.; Abdollahzadeh, B.; Mohammadzadeh, A.; El-Kenawy, E.-S.M.; Eid, M.M.; Mosalam, K.M. Gray langurs optimizer: A multi-group bio-inspired optimization algorithm. *Artif. Intell. Rev.* **2026**, *59*, 166. [CrossRef]
17. Mohammed, B.O.; Aghdasi, H.S.; Salehpour, P. Dhole optimization algorithm: A new metaheuristic algorithm for solving optimization problems. *Clust. Comput.* **2025**, *28*, 430. [CrossRef]
18. Fu, Y.; Liu, D.; Chen, J.; He, L. Secretary bird optimization algorithm: A new metaheuristic for solving global optimization problems. *Artif. Intell. Rev.* **2024**, *57*, 123. [CrossRef]
19. Ouertani, M.W.; Manita, G.; Korbaa, O. Hannibal Barca optimizer: The power of the pincer movement for global optimization and multilevel image thresholding. *Clust. Comput.* **2025**, *28*, 482. [CrossRef]
20. Zolf, K. Gold rush optimizer: A new population-based metaheuristic algorithm. *Oper. Res. Decis.* **2023**, *33*, 113–150. [CrossRef]
21. Lyu, L.; Kong, G.; Yang, F.; Li, L.; He, J. Augmented Gold Rush Optimizer is used for engineering optimization design problems and UAV path planning. *IEEE Access* **2024**, *12*, 134304–134339. [CrossRef]
22. Yu, M.; Lu, H.; Wang, D.; Du, J.; Kong, D.; Xu, X.; Xu, J. Quantum-enhanced gold rush Optimizer for multi-threshold segmentation of lupus nephritis pathological images. *Displays* **2026**, *92*, 103334. [CrossRef]
23. Yu, M.; Du, J.; Xu, X.; Xu, J.; Jiang, F.; Fu, S.; Zhang, J.; Liang, A. A multi-strategy enhanced Dung Beetle Optimization for real-world engineering problems and UAV path planning. *Alex. Eng. J.* **2025**, *118*, 406–434. [CrossRef]
24. Xue, G.; Zhang, E.; An, G.; Du, J.; Yin, X.; Zhou, P.; Zhang, X. UAV-Based Coverage Path Planning for Unmanned Agricultural Vehicles. *Sensors* **2026**, *26*, 927. [CrossRef]
25. Haam, S.; Kim, H.; Yoo, M.; Song, W.S. Optimal additional evacuation route analysis in deep underground station structures using Dijkstra's algorithm. *Dev. Built Environ.* **2025**, *24*, 100784. [CrossRef]
26. Lin, S.; Fan, X.; Xie, Z.; Wu, Y.; Yang, H. Self-adaptive search algorithm for path planning based on the A* algorithm. *Sci. Rep.* **2025**, *15*, 44901. [CrossRef]
27. Zhang, H.; Wei, C.; He, Y. Research on multi-vehicle formation control based on improved artificial potential field method. *Proc. Inst. Mech. Eng. Part D J. Automob. Eng.* **2024**, *239*, 4108–4119. [CrossRef]
28. Flaieh, E.H.; Humaidi, A.J. Lazy Chaos A*: A novel chaos-inspired algorithm for efficient 3R robot arm path planning based on semi-known exploration behavior. *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* **2026**, *240*, 3628–3649. [CrossRef]
29. Muhammed, M.L.; Humaidi, A.J.; Flaieh, E.H. Towards Comparison and Real Time Implementation of Path Planning Methods for 2R Planar Manipulator with Obstacles Avoidance. *Math. Model. Eng. Probl.* **2022**, *9*, 379–389. [CrossRef]
30. Al-qassar, A.; Al-Obaidi, A.S.; Falah, A.; Humaidi, A.; Nasser, A.; Alkhayyat, A.; Ibraheem, I.; Ijost, I. Finite-Time Control of Wing-Rock Motion for Delta Wing Aircraft Based on Whale-Optimization Algorithm. *Indones. J. Sci. Technol.* **2021**, *6*, 441–456. [CrossRef]
31. Kadhim, R.; Kadhim, M.; Al-Khazraji, H.; Humaidi, A. Bee Algorithm Based Control Design for Two-links Robot Arm Systems. *IIUM Eng. J.* **2024**, *25*, 367–380. [CrossRef]
32. Abdul-Kareem, A.I.; Hasan, A.F.; Al-Qassar, A.A.; Humaidi, A.J.; Hassan, R.F.; Ibraheem, I.K.; Azar, A.T. Rejection of wing-rock motion in delta wing aircrafts based on optimal LADRC schemes with butterfly optimization algorithm. *J. Eng. Sci. Technol.* **2022**, *17*, 2476–2495.
33. Zhang, Y.; Adegboye, O.R.; Feda, A.K.; Agyekum, E.B.; Kumar, P. Dynamic gold rush optimizer: Fusing worker adaptation and salp navigation mechanism for enhanced search. *Sci. Rep.* **2025**, *15*, 15779. [CrossRef]
34. Nie, Y.; Liu, Y.; Zhao, W.; Li, J. Optimizing Linear Antenna Arrays with The Chaotic Gold Rush Optimizer. In *Proceedings of the 2024 IEEE 14th International Conference on Electronics Information and Emergency Communication (ICEIEC)*; IEEE: Piscataway, NJ, USA, 2024; pp. 31–34. [CrossRef]
35. Panagiotakis, C. AGRO: An Adaptive Gold Rush Optimizer with Dynamic Strategy Selection. *Algorithms* **2026**, *19*, 192. [CrossRef]
36. Zhang, S.; Wang, L. Multi-UAV cooperative path planning based on chaotic grey wolf optimiser. *Aeronaut. J.* **2025**, *130*, 741–761. [CrossRef]
37. Huang, S.; Fu, Y.; Liu, D.; Chen, J.; He, L.; Zhang, J. DLC-NGO: An enhanced northern goshawk optimization for 3D UAV path planning in complex environments with multiple threats. *Clust. Comput.* **2025**, *28*, 677. [CrossRef]
38. Li, Z.; Zheng, B.; Fu, Y.; Liu, X. Adaptive walrus optimization algorithm for unmanned aerial vehicle path planning and engineering optimization problems. *Clust. Comput.* **2025**, *28*, 555. [CrossRef]
39. Fu, S.; Li, K.; Huang, H.; Ma, C.; Fan, Q.; Zhu, Y. Red-billed blue magpie optimizer: A novel metaheuristic algorithm for 2D/3D UAV path planning and engineering design problems. *Artif. Intell. Rev.* **2024**, *57*, 134. [CrossRef]
40. Li, Y.; Ni, Z.; Jin, F.; Li, J.; Li, F. Research on Clustering Method of Improved Glowworm Algorithm Based on Good-Point Set. *Math. Probl. Eng.* **2018**, *2018*, 8724084. [CrossRef]
41. Zhang, J.; Sanderson, A.C. JADE: Adaptive differential evolution with optional external archive. *IEEE Trans. Evol. Comput.* **2009**, *13*, 945–958. [CrossRef]
42. Chen, X.; Shen, A. Self-adaptive differential evolution with Gaussian–Cauchy mutation for large-scale CHP economic dispatch problem. *Neural Comput. Appl.* **2022**, *34*, 11769–11787. [CrossRef]

43. Wu, G.; Mallipeddi, R.; Suganthan, P. Problem Definitions and Evaluation Criteria for the CEC 2017 Competition and Special Session on Constrained Single Objective Real-Parameter Optimization. 2016, pp. 1–18. Available online: https://www.researchgate.net/publication/317228117_Problem_Definitions_and_Evaluation_Criteria_for_the_CEC_2017_Competition_and_Special_Session_on_Constrained_Single_Objective_Real-Parameter_Optimization (accessed on 6 June 2026).
44. Zhang, Y.; Yang, X. Multi-Strategy Enhanced Connected Banking System Optimizer for Global Optimization and Corporate Bankruptcy Forecasting. *Mathematics* **2026**, *14*, 618. [CrossRef]
45. Zhang, Y.; Lu, J.; Yang, X. MESETO: A Multi-Strategy Enhanced Stock Exchange Trading Optimization Algorithm for Global Optimization and Economic Dispatch. *Mathematics* **2026**, *14*, 981. [CrossRef]

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