Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Research on UAV Path Planning Using Multi-Strategy Enhancements to Escape Algorithm

Jin Wu

wujin1026@126.com

Xi'an University of Posts and Telecommunications

Linlin Guo

Xi'an University of Posts and Telecommunications

Shangzhao Dai

Xi'an University of Posts and Telecommunications

Research Article

Keywords: escape algorithm, improved escape algorithm, meta-heuristic algorithm, 3D UAV path planning

Posted Date: November 25th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11785846/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Research on UAV Path Planning Using Multi-Strategy Enhancements to Escape Algorithm

Jin Wu[1] · Linlin Guo[1] · Shangzhao Dai[1]

[1] School of Electronic Engineering, Xi′ an University of Posts and Telecommunications, Xi′ an 710121, People′ s Republic of China

Abstract

Three-dimensional UAV path planning constitutes a high-dimensional, multi-constraint complex optimization problem, for which conventional methods are prone to converge to local optima. Although metaheuristic algorithms have been widely adopted, the Escape Algorithm (ESC) still suffers from several limitations: insufficient initial population quality, fixed thresholds that hinder dynamic balance between exploration and exploitation phases, and a lack of local geometric awareness during the exploitation phase. To address these issues, this study proposes a multi-strategy enhanced Escape Algorithm (IESC), which integrates Piecewise chaotic mapping initialization, lens-based inverse learning, a nonlinear attenuation-vibration hybrid factor, elite-guided gradient simulation acceleration, and an experience exchange strategy. These integrated components aim to improve population quality, enhance diversity, achieve a dynamic equilibrium between exploration and exploitation, and provide directional search information. Experimental results on the CEC2020 benchmark dataset demonstrate that IESC outperforms mainstream algorithms—including ESC, PSO, DE, GWO, MVO, GJO, SWO, and APO—in terms of both convergence accuracy and stability; further validation via the Wilcoxon rank-sum test confirms the statistical significance of these performance advantages. Simulation results for three-dimensional UAV path planning indicate that the proposed algorithm can generate high-quality trajectories characterized by smooth transitions, clear obstacle avoidance, and optimal feasibility, thereby validating its effectiveness and practical applicability in complex engineering optimization scenarios.

Keywords: escape algorithm; improved escape algorithm; meta-heuristic algorithm; 3D UAV path planning

1.Introduction

With the development of emerging applications such as 3D UAV path planning, complex optimization problems have garnered increasing attention. Such problems typically require the search for optimal solutions under multiple constraints; UAV path planning is a prime example. It not only involves planning an optimal flight trajectory in three-dimensional space but also must satisfy multiple physical constraints — including flight range, altitude, yaw angle, and climb angle — characterized by high dimensionality and strong coupling. Traditional methods, such as the Newton method and the steepest descent method, rely heavily on the continuity and differentiability of the objective function; when applied to high-dimensional, multi-constraint problems, these methods are prone to converge to local optima, making it difficult to meet practical engineering requirements.

Meta-heuristic algorithms have emerged as vital tools for tackling complex optimization problems, owing to their advantages—such as independence from precise mathematical models and strong adaptability to high-dimensional nonlinear problems. Based on their inspiration, these algorithms can be categorized into evolutionary algorithms, swarm intelligence algorithms, physics-based algorithms, and human behavior-based algorithms; among these, human behavior-based algorithms have garnered considerable attention for their ability to simulate decision-making processes and social interaction mechanisms. Existing algorithms—including genetic algorithms, differential evolution, particle swarm optimization, and grey wolf optimization—have already found applications across a wide range of problem domains. However, the "no free lunch" theorem demonstrates that no single algorithm can be universally effective for all problems, which continues to drive the exploration of new algorithmic approaches.

The Escape Algorithm (ESC)[2] is a meta-heuristic algorithm based on human behavior, recently proposed and inspired by crowd evacuation dynamics. It partitions the population into three categories—calm, conformist, and panicked—and achieves a balance between exploration and development through the coordinated search of these three states. However, ESC still suffers from several limitations: suboptimal quality of random initialization, fixed thresholds that struggle to dynamically balance exploration and development, and a lack of local geometric awareness during the development phase; consequently, its convergence efficiency diminishes when applied to high-dimensional, complex problems.

To address the aforementioned limitations, this paper proposes an improved Escape Algorithm (IESC) that integrates piecewise chaotic mapping initialization, lens-based inverse learning, a nonlinear attenuation-vibration hybrid factor, elite-guided gradient simulation acceleration, and an experience exchange strategy, thereby enhancing its global search capability, convergence accuracy, and robustness. Experiments conducted on the CEC2020 benchmark functions and a three-dimensional UAV path planning problem validate its comprehensive optimization performance and engineering applicability.

The main contributions of this study are summarized as follows:

(1) Proposed a multi-strategy enhanced Escape Algorithm (IESC). The Piecewise chaotic mapping and lens-based inverse learning strategies are introduced to generate a more uniformly distributed initial population and expand the effective search space, significantly improving the population diversity and global exploration capability.

(2) Designed a unique nonlinear attenuation-oscillation hybrid factor (NDOHF). This factor integrates temporal decay, population diversity feedback, and periodic oscillation, providing an adaptive and flexible threshold conversion mechanism to dynamically balance the exploration and exploitation phases, effectively avoiding premature convergence.

(3) Introduced an elite-guided gradient acceleration strategy and an Experience Exchange Strategy (EES). The gradient simulation acceleration provides directional search information for local exploitation, while the three-stage EES enhances the algorithm's ability to escape local optima by utilizing historical search experience.

(4) Conducted comprehensive benchmark experiments and analyses. The

performance of IESC is rigorously evaluated on 10 benchmark functions from CEC2020. Quantitative, qualitative, and statistical (Wilcoxon rank-sum test) analyses, along with exploration/exploitation ratio tracking, demonstrate the superior convergence accuracy and stability of IESC compared to other mainstream algorithms.

(5) Applied IESC to a complex 3D UAV path planning problem. By constructing a cost function with multiple constraints, simulation results verify that IESC can generate high-quality, smooth, and safe flight trajectories, demonstrating its practical applicability in real-world complex engineering optimization.

The remainder of this study is organized as follows. Section 2 introduces the basic principles of the original ESC. Section 3 details the mathematical modeling and specific improvement strategies of IESC. Section 4 provides performance comparisons of IESC on the CEC2020 benchmark functions. Section 5 presents the environmental modeling and results for 3D UAV path planning. Section 6 summarizes this work along with a future outlook.

2.Escape Algorithm

The Escape Algorithm (ESC) is a meta-heuristic optimization algorithm proposed by Ouyang et al. in 2024, inspired by three behavioral patterns exhibited by individuals during emergency evacuations: the calm individuals, who rationally seek optimal solutions; the conformist individuals, who follow and converge; and the panicked individuals, who exhibit random perturbations. ESC partitions the population into three categories—calm, conformist, and panicked—each of which plays distinct roles during the exploration phase: guiding, bridging, and perturbing, respectively; together, these roles preserve population diversity and facilitate convergence toward the global optimum. In the development phase, all individuals transition to the calm category, undergoing fine-grained search by converging toward the elite individuals, thereby simulating the process by which the population gradually converges to the optimal exit.

2.1 Exploration Phase

When $t \leq T/2$ (where t denotes the current iteration count and T denotes the maximum number of iterations of the algorithm), the algorithm is in the exploration phase. At this stage, the population is sorted according to its fitness values and then partitioned proportionally into three groups: a calm group accounting for 15%, a herd-following group accounting for 35%, and a panic group accounting for 50%. This distribution reflects the actual scenario in real evacuation situations—where a few individuals remain calm, some individuals follow the crowd, and the majority fall into panic.

The Calm Group achieves rational search by converging toward the group's average decision center C_j, ensuring that the population converges toward a consensus region that represents the collective decision of the group. The search formula of the Calm Group is given by Eq.(1):

$$x_{i,j}^{new} = x_{i,j} + m_1 \times (\omega_1 \times (C_j - x_{i,j}) + v_{c,j}) \times P(t) \quad (1)$$

In Eq.(1), C_j denotes the average value of all calm individuals within that

dimension, representing the centroid of the calm population in the j-th dimension. m_1 is determined by a Bernoulli distribution, in which the probabilities of taking the values 0 or 1 are equal; this models the scenario where, during emergencies, crowd congestion leads to the failure to update certain dimensions. ω_1 is the adaptive learning weight.

In Eq.(1), the random movement of calm individuals is described by $v_{c,j}$, with the specific formula defined in Eq.(2). *P(t)* denotes the overall panic index of the population, with the specific formula defined in Eq. (3).

$$v_{c,j} = R_{c,j} - x_{i,j} + \varepsilon_j \tag{2}$$

$$P(t) = \cos(\frac{\pi t}{6T}) \tag{3}$$

In Eq.(2), $R_{c,j}$ this denotes a randomly generated position within the boundary of the calm zone.

The conformity group refers to individuals who lack their own decision-making capability during the escape process; their positions are updated under the influence of the two other groups, as described in Eq. (4).

$$x_{i,j}^{new} = x_{i,j} + m_1 \times (\omega_1 \times (C_j - x_{i,j}) + m_2 \times \omega_2 \times (x_{p,j} - x_{i,j}) + v_{h,j} \times P(t)) \tag{4}$$

In Eq.(4), $x_{p,j}$ is an individual randomly selected from the panic-stricken group and represents the potential direction of random escape for the panic group. ω_2 is another adaptive Levy weight used for the conformity group, which is applied to simulate the step size in the exploration stage.

Individuals in the panic group exhibit irregular patterns when exploring the solution space, as their group behavior is jointly influenced by a latent exit mechanism and the random directions of other individuals; their update formula is given by Eq. (5).

$$\begin{aligned} x_{i,j}^{new} &= x_{i,j} + m_1 \times (\omega_1 \times (E_j - x_{i,j}) + \\ & m_2 \times \omega_2 \times (x_{rand,j} - x_{i,j}) + v_{p,j} \times P(t)) \end{aligned} \tag{5}$$

After population initialization, the algorithm employs a fitness function $f_i = f(x_i)$ to evaluate the fitness of each individual, then sorts the population in descending order of fitness. The top-ranked individuals are then stored in an elite pool; these elite individuals represent the optimal potential solutions identified by the population—namely, the exit point.

In Eq.(5), E_j denotes an individual randomly selected from the elite pool, representing a potential exit point induced by panic; $x_{rand,j}$ denotes an individual randomly selected from the population, representing a random position formed by the movement of panicked individuals. The definition of the vector $v_{p,j}$ is given by Eq. (6), $R_{p,j} = r_{\min,j}^{p} + r_{i,j} \times (r_{\max,j}^{p} - r_{\min,j}^{p})$ is a randomly generated position within the boundary of the panicked population, $r_{\max,j}^{p}$ and $r_{\min,j}^{p}$ respectively represent the maximum and minimum values of the j-th dimension for all individuals in the panic

group.

$$v_{p,j} = R_{p,j} - x_{i,j} + \varepsilon_j \tag{6}$$

2.2 Development Phase

When the number of iterations exceeds half of the maximum number of iterations, the algorithm enters the development phase, where all individuals are uniformly treated as a "calm population," and the search focus shifts from global exploration to local fine-grained refinement. Each individual gradually optimizes its position by approaching members of the elite pool, while also considering randomly selected individuals from the population to prevent search stagnation. This process simulates the behavior of a crowd in the late stages of evacuation—gradually converging toward a pre-identified exit—thereby enabling a smooth transition of the algorithm from extensive exploration to refined development. The position update in this phase is given by Eq. (7).

$$x_{i,j}^{new} = x_{i,j} + m_1 \times \omega_1 \times (E_j - x_{i,j}) + m_2 \times \omega_2 \times (x_{rand,j} - x_{i,j}) \tag{7}$$

In Eq.(7), the $x_{i,j}$ term denotes the position of the i-th individual in the j-th dimension. E_j represents the position of a member in the elite pool, which both symbolizes a potential safe exit and signifies one of the best solutions identified to date. $x_{rand,j}$ denotes the position of an individual randomly selected from the population. By gradually aligning the individual's position with those of both the elite pool members and the randomly selected individual, the individual can, through population-based simulation, progressively converge toward the identified optimal exit.

3. Improvement of the Escape Algorithm

To address the limitations of the Escape Algorithm (ESC)—such as low initial population quality, imbalanced exploration and development performance, susceptibility to local optima, and limited capability in high-dimensional search—this study proposes a Multi-Strategy Enhanced Escape Algorithm (IESC). The specific enhancements include: employing a piecewise chaotic mapping for initialization to improve population quality and convergence speed; introducing lens-based inverse learning to enhance global search capability and local escape performance; designing a nonlinear attenuation-oscillation hybrid factor to dynamically balance exploration and development phases; incorporating an elite-guided gradient simulation acceleration strategy during the development phase to boost local development accuracy; and integrating an experience exchange strategy to strengthen optimization capability and coordinate exploration with development. Through the synergistic application of these multiple strategies, IESC achieves comprehensive improvements in the algorithm's overall performance.

3.1 Piecewise chaotic mapping initialization

To improve the quality of the initial population in the ESC algorithm, this paper introduces a piecewise chaotic map [3] for population initialization. Chaotic sequences

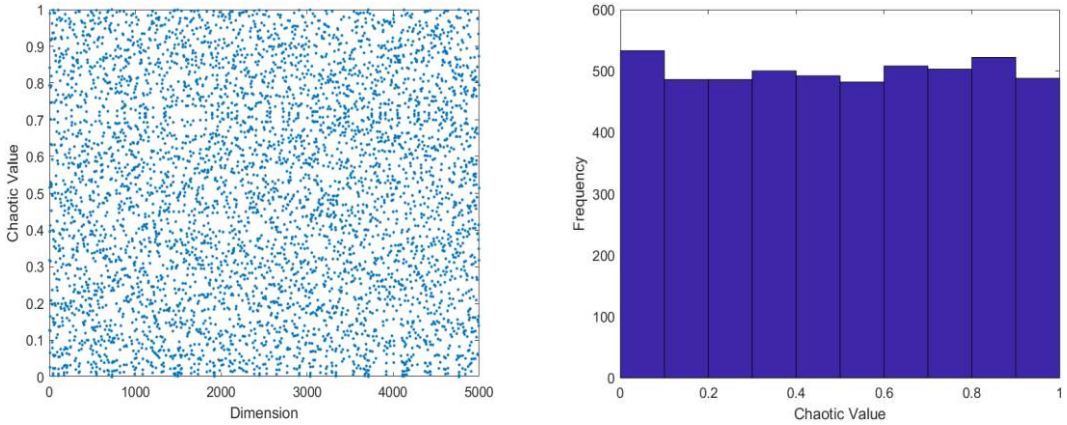
exhibit excellent ergodicity and randomness, enabling the generation of initial solutions with more uniform distributions, thereby enhancing population diversity and accelerating convergence.

The piecewise chaotic map is a classified map capable of generating chaotic sequences with uniformly distributed internal components and excellent ergodic properties within a given interval. Its mathematical formulation is given by Eq. (8).

$$X_{n+1}=\begin{cases}\dfrac{X_n}{d} & 0\le X_n\le d\\ \dfrac{X_n-d}{0.5-d} & d\le X_n\le 0.5\\ \dfrac{1-d-X_n}{0.5-d} & 0.5\le X_n\le 1-d\\ \dfrac{1-X_n}{d} & 1-d\le X_n\le 1\end{cases} \tag{8}$$

In Eq.(8), the parameter d serves as the control variable for the segmentation range, determining the boundaries of the four linear intervals to ensure that these intervals do not overlap, with X_n and d taking values in the range [0,1]. Through parameter sensitivity analysis and comparative experiments, it was found that when d=0.3, the generated chaotic sequence exhibits optimal performance in terms of uniformity and ergodicity; therefore, in IESC, the value of d is consistently set to 0.3.

The piecewise chaotic map is illustrated in Fig.1. This map exhibits superior performance in terms of uniformity and diversity, and yields a population distribution that is more uniformly distributed and possesses higher diversity upon initialization. Consequently, this study employs the piecewise chaotic map for the initialization of the ESC.



(a) Piecewise mapping of diversity distribution plot

(b) Piecewise mapping equalization frequency distribution plot

Fig.1 Piecewise chaotic mapping distribution plot

3.2 Reverse Learning Strategy for Lenses

To enhance global search capability and avoid becoming trapped in a local optimum, this paper introduces a lens-based inverse learning strategy [5]. This strategy

expands the search space and increases population diversity by generating inverse solutions; furthermore, by dynamically adjusting the imaging contraction coefficient, it flexibly controls the generation intensity of inverse solutions, thereby effectively improving the algorithm's convergence performance and robustness in complex optimization problems.

A schematic diagram of reverse lens imaging is shown in Fig.2. The regions [a, b] denote the imaging range of the lens, while the y-axis indicates the position of the convex lens. If a particle P has a height h, its projection onto the x-axis is denoted as X_p, the reverse image point obtained by particle P passing through the convex lens on the y-axis is denoted as p*, with a height of h*, and its projection onto the x-axis is denoted as X_{p^*}. The quantities u, v, and f represent, respectively, the object distance, image distance, and focal length of the lens imaging.

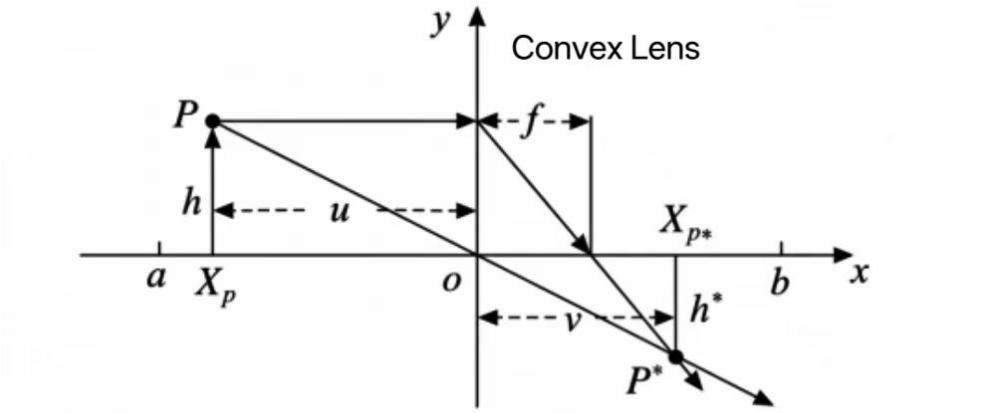


Fig.2 Schematic diagram of the lens imaging inverse learning model

The principle of lens imaging is illustrated by Eqs.(9) and (10).

$$\frac{1}{u}+\frac{1}{v}=\frac{1}{f} \tag{9}$$

$$\frac{(a+b)/2-X_p}{X_{p^*}-(a+b)/2}=\frac{h}{h^*} \tag{10}$$

Let $k=h^*/h$ denote the imaging contraction coefficient; then, the projected image points of the lens can be expressed as given in Eq. (11).

$$X_{p^*}=\frac{a+b}{2}+\frac{a+b}{2k}-\frac{X_p}{k} \tag{11}$$

When k=1, Eq. (11) can be simplified $X_{p^*}=a+b-X_p$ to X_{p^*} the standard inverse learning solution. By adjusting the imaging contraction coefficient k, the generation position of the inverse solution can be dynamically tuned to explore different directions; therefore, this paper adopts an adaptive directional learning solution space, where k is defined as given in Eq. (12).

$$k(t) = (1 + \sqrt{\frac{t}{T}})^{10} \tag{12}$$

As shown in Eq. (12), the parameter k varies slowly in the early iterations, enabling the algorithm to conduct extensive exploration during the exploration phase; in contrast, k varies more rapidly in the later iterations, allowing the algorithm to focus more on high-value areas during the development phase.

The lens-based inverse learning strategy associates the projection positions of the origin and the image point with the population states before and after each iteration, using the search interval [a, b] as the generation range for inverse solutions. This strategy dynamically adjusts the distribution range of inverse solutions—enhancing population diversity in the early stages of iteration and facilitating localized fine-grained search in later stages—thereby effectively balancing the algorithm's global exploration and local development capabilities.

3.3 Nonlinear Attenuation–Oscillation Hybrid Factor

ESC employs a fixed threshold to delineate the exploration and development phases; however, this rigid strategy exhibits significant limitations: if local convergence occurs prematurely in the early stages, the algorithm may continue exploration, leading to resource wastage; conversely, if it is forced to transition to development before exhaustive exploration has been completed in later stages, it may miss out on superior regions. To address these issues, this paper introduces the Nonlinear Decay-Oscillation Hybrid Factor (NDOHF), which integrates temporal decay, population diversity feedback, and periodic oscillation. This approach enables the algorithm to dynamically and adaptively balance exploration and development based on its current search state, thereby avoiding suboptimal behavior caused by fixed thresholds, and ultimately enhancing both search efficiency and solution quality. The underlying principle is illustrated in Eq. (13).

$$a(t) = \max(a_{\min}, (1 - \frac{t}{T})^{\gamma}) + \eta \cdot \sin(\frac{\pi t}{\tau}) \tag{13}$$

$$\gamma = 2 \cdot (1 - \frac{1}{D} \sum_{d=1}^{D} \frac{\sigma_d}{u_d - l_d}) \tag{14}$$

In Eq.(14), t denotes the current iteration count, and T represents the maximum number η of iterations, set to 500; α is the oscillation amplitude parameter, set to 0.1; τ is the oscillation period, set to T/5; and α_{min} denotes the minimum value of the mixing factor, set to 0.2. In Eq.(14), the adaptive decay coefficient varies with the population diversity, where σ_d denotes the standard deviation of the population in the d-th dimension, and U_d and L_d denote the upper and lower bounds of the d-th dimension, respectively, for d = 1, 2,..., D.

As demonstrated by Eqs. (13) and (14), the NDOHF factor achieves a smooth transition between exploration and exploitation through nonlinear attenuation: when population diversity is high, the attenuation coefficient approaches zero, thereby sustaining global exploration; when the population becomes concentrated, the attenuation coefficient approaches 2, accelerating local exploitation. The periodic

oscillatory term provides the algorithm with multiple opportunities to escape local optima, effectively preventing premature convergence. The adaptive mechanism enables the algorithm to autonomously adjust its parameters based on the search state, reducing reliance on preset thresholds and thereby enhancing robustness and generalizability.

3.4 Elite-Guided Graded Simulation Acceleration Strategy

In its development phase, the ESC algorithm relies solely on differential updates between elite and random individuals, lacking awareness of local geometric features, which results in insufficient convergence efficiency. To address this limitation, this paper proposes a gradient simulation acceleration strategy[8], which leverages differential approximations of gradient directions among elite individuals to provide directional search information during the development phase. This strategy is based on the gradient descent principle but does not require the objective function to be differentiable; instead, it infers the optimization direction from the historical positions of elite individuals, enabling the algorithm to perform more intelligent localized fine-grained searches. The underlying principles are illustrated in Eqs.(15)-(19).

$$G_{elite} = \frac{(X_1 - X_2) + (X_2 - X_3)}{\left\|(X_1 - X_2) + (X_2 - X_3)\right\| + \varepsilon} \tag{15}$$

Eq.(15) represents the formula for computing X_1 X_2 X_3 the gradient direction of the elite individuals. The variables $\mathcal{E}$ denote the position vectors of the top three elite individuals in the current population by fitness ranking. To prevent numerical instability caused by a zero denominator, a regularization term of approximately 2.22 $\times 10^{-16}$ is employed. Eq.(15) approximates the gradient direction of the objective function by computing the difference vectors between the elite individuals.

$$s = 0.5 \cdot [1 + \tanh(10(\frac{\overline{\sigma}}{\Delta} - 0.2))] \tag{16}$$

In Eq. (16), s denotes the gradient descent step-size control factor, which dynamically adjusts the contribution weight of the gradient term based on population diversity. $\overline{\sigma}$ is the average of the standard deviations across all dimensions is calculated to quantify the dispersion of individuals within each dimension, thereby reflecting the overall diversity level of the population. Δ is the search space range is defined as the maximum value across all dimensions. When population diversity is high (>0.2), the gradient term is fully activated, accelerating convergence; when population diversity is low (<0.2), the gradient term is suppressed to prevent excessive clustering. In this strategy, a Levy distribution weight WLevy is introduced to inject random perturbations into the gradient direction.

To enable the algorithm to conduct large-scale searches in the early stages of the optimization process and smaller-scale refinements in later stages, a time-varying decay coefficient c(t) is incorporated into the strategy, ensuring that the influence of the gradient term diminishes with each iteration, as illustrated in Eq.(17).

$$c(t) = 0.1 \cdot (1 - \frac{t}{T}) \tag{17}$$

In Equation (17), t denotes the current iteration count, and T denotes the

maximum number of iterations. To prevent invalid gradient updates caused by numerical errors, a conditional check is incorporated into the gradient update process, as illustrated in Eq. (18).

$$V_g = \begin{cases} s \cdot g_{elite} \square\ w_{Levy}, & \left\| s \cdot g_{elite} \square\ w_{Levy} \right\| \geq 10^{-5} \\ 0, & otherwise \end{cases} \tag{18}$$

The elite-guided gradient simulation acceleration strategy is implemented by integrating it with the ESC development phase formulation (7), thereby endowing the algorithm with greater purposefulness during its development phase and enhancing its search precision; the formulation for the IESC development phase is given by Eq.(19).

$$\begin{aligned} x_{i,j}^{new} &= x_{i,j} + m_1 \times \omega_1 \times (E_j - x_{i,j}) + \\ & m_2 \times \omega_2 \times (x_{rand,j} - x_{i,j}) + V_g \cdot c(t) \end{aligned} \tag{19}$$

3.5 Experience Exchange Strategy

To enhance the optimization capability of the ESC algorithm and improve the balance between exploration and exploitation, this paper introduces the Experience Exchange Strategy (EES). Inspired by the human societal mechanism of experience sharing, this strategy treats the population's historical search information as accumulative experience; by dynamically adjusting the utilization of this experience in a phased manner, it enhances both the diversity and convergence properties of the algorithm.

EES progresses through three distinct phases—experience scarcity, experience crossover, and experience sharing—according to the advancement of the search process. This enables the algorithm to adaptively adjust the intensity of its experience reference and its information interaction mechanisms across different search stages, thereby overcoming the limitations of traditional algorithms, which typically employ a single information interaction paradigm. In the initial phase of search, when the population possesses limited experience, EES selects only two individual agents for moderate crossover, thereby preserving the exploratory capability of the original algorithm while preventing premature convergence. In the middle phase, as the population accumulates sufficient experience, EES increases the number of individual agents to three, broadening the scope of information interaction and assisting agents trapped in local optima to escape such constraints. In the final phase, when the population concentrates in dominant regions, EES shifts its experience utilization strategy from crossover-based exploration to averaging-based refinement, thereby enhancing the precision of local development. Through these three phases, and by employing a greedy selection mechanism to retain superior solutions, EES collaboratively achieves an adaptive transition from extensive exploration to refined development.

This strategy dynamically aligns the intensity of experience utilization with the search phase, rather than relying on fixed threshold switching; by synergistically adjusting the number of experienced agents and their utilization modes, it balances global exploration with local exploitation; embedded as an independent preprocessing module in each iteration cycle, it enhances search efficiency and solution accuracy without compromising the original algorithmic framework.

3.6 Flowchart of the improved escape algorithm

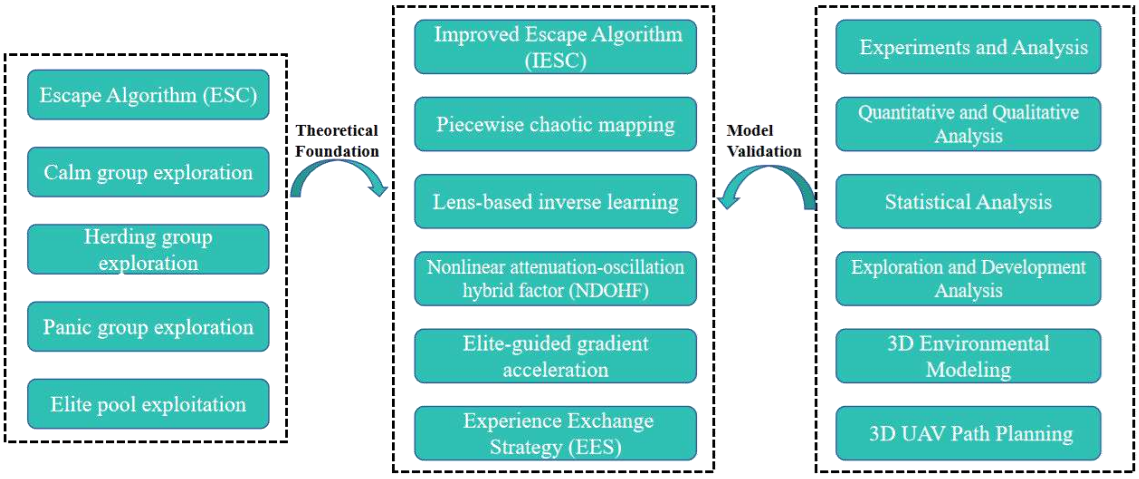


Fig.3 Conceptual Framework of IESC

The overall framework of the proposed IESC algorithm is illustrated in Fig.3. This schematic diagram provides a macroscopic overview of the research, which is primarily divided into three interconnected modules: the theoretical foundation of the ESC, the core improvement strategies of IESC, and the subsequent experiments and analysis.

On the left side, the theoretical foundation establishes the basic mechanics of ESC, including calm, herding, and panic group exploration, as well as elite pool exploitation. The middle module highlights the five key strategies integrated into IESC to overcome existing limitations: Piecewise chaotic mapping, lens-based inverse learning, the nonlinear attenuation-oscillation hybrid factor (NDOHF), elite-guided gradient acceleration, and the Experience Exchange Strategy (EES). The right module encompasses the model validation, including quantitative, qualitative, statistical, and exploration/development analyses, finally culminating in 3D environmental modeling and 3D UAV path planning.

To further elucidate the operational mechanism of the IESC core strategies shown in the middle module of Fig.3, the detailed algorithm flowchart is illustrated in Fig.4. This flowchart clearly depicts the specific logical structure, execution sequence, input-output relationships, and the interaction mechanisms among various modules.

The algorithm optimization process consists of three primary stages: In the first stage, the escaping population is iteratively updated based on the EES strategy, and a greedy selection strategy is employed to determine the update targets; In the second stage, leveraging the principles of the escaping algorithm, a hybrid factor and a gradient acceleration mechanism are introduced for exploration and development, again utilizing the greedy selection strategy to retain superior solutions; In the third stage, a lens-based inverse learning strategy is applied to enhance local search precision, followed by the use of the greedy selection strategy to filter out high-quality solutions. The algorithm terminates upon reaching the predefined maximum number of iterations, and the current optimal solution is output as the final optimization result.

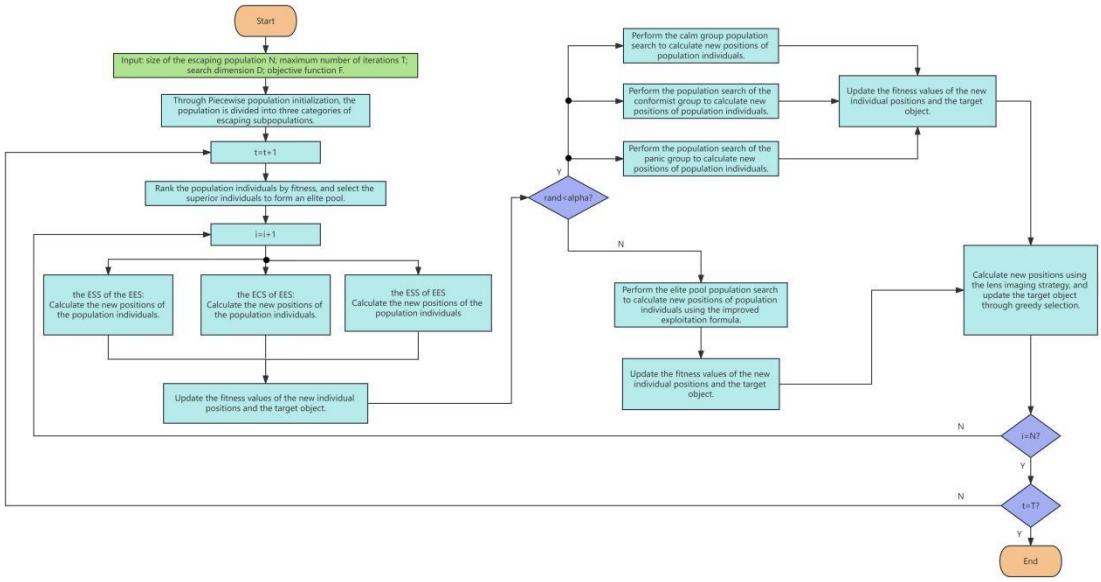


Fig.4 Detailed Flowchart of the IESC Algorithm

4. Benchmark Function Experiments and Result Analysis

To validate the effectiveness and optimization performance of the algorithm, this section employs ten standard test functions from CEC2020 for performance evaluation, covering four types: single-modal, multi-modal, hybrid, and composite functions. Single-modal functions are used to assess the algorithm's local development and convergence capabilities; multi-modal functions, which contain numerous local optima, evaluate the algorithm's global exploration capability and its ability to escape local optima; whereas hybrid and composite functions, whose structures are designed for this purpose, provide a comprehensive assessment of the algorithm's ability to balance exploration and exploitation. Detailed information about each test function is presented in Table 1, which includes key parameters such as function dimensionality (D), search range (range), and theoretical optimal value (f_{min}).

Table 1 CEC2020 Test Function Set

type	function name	range	*D*	f_{min}
Single-modal function	F1: Shifted and Rotated Bend Cigar Function	[-100,100]	10	100
Multimodal function	F2: Shifted and Rotated Schwefel′ s Function	[-100,100]	10	1100
	F3: Shifted and Rotated Lunacek bi-Rastrigin Function	[-100,100]	10	700
	F4: Expanded Rosenbrock′ s plus Griewangk′ s Function	[-100,100]	10	1900
blending function	F5: Hybrid Function 1 (N=3)	[-100,100]	10	1700
	F6: Hybrid Function 2 (N=4)	[-100,100]	10	1600
	F7: Hybrid Function 3 (N=5)	[-100,100]	10	2100

Table 1(continued) CEC2020 Test Function Set

type	function name	range	*D*	f_{min}
compound function	F8: Compositon Function 1 (N=3)	[-100,100]	10	2200
	F9: Compositon Function 2 (N=4)	[-100,100]	10	2400
	F10: Compositon Function 3 (N=5)	[-100,100]	10	2500

This paper conducts a comprehensive evaluation of the performance of the proposed IESC by comparing it with ESC and several advanced optimization algorithms. The comparison algorithms include the classical PSO and DE algorithms, the GWO and MVO algorithms—widely used in recent years—as well as the recently proposed GJO, SWO, and APO algorithms. Across all 10 benchmark functions, the performance of IESC was assessed from two perspectives: convergence accuracy (average and optimal values) and stability (standard deviation). Statistical significance analysis was performed using the Wilcoxon rank-sum test to scientifically validate the effectiveness of IESC. The parameters of all comparison algorithms were set according to the values recommended in the original literature; detailed settings are provided in Table 2.

Table 2 Parameter Variable Settings

Algorithm Name	Parameter Name	Numeric value
PSO	Constants c1, c2	2
	Inertia weight ω	$\omega_{\max}=0.9, \omega_{\min}=0.4$
	maximum speedVmax	6
DE	zoom factor*F*	0.5
	Cross-probability *CR*	0.3
GJO	Escape energy *E*	[1.5,0]
	Parameter c3	1.5
GWO	Parametera	[2,0]
MVO	Maximum probability P1	1
	Minimum probability P2	0.2
SWO	Weight factor TR	0.3
	Crossing ratio CR	0.2
APO	Parameter *F*	0.5
	Parameter *C*	0.5

Regarding experimental setup, the population size was uniformly set to 30, and the maximum number of iterations was set to 1,000. Given the stochastic nature of metaheuristic algorithms, each algorithm was run independently 20 times for each test to obtain statistically meaningful results. All simulation experiments were conducted on the Windows 10 operating system, using the hardware platform Intel(R) Core(TM) i5-9300H CPU (@2.4 GHz) and the software environment MATLAB R2023b.

4.1 Quantitative Analysis of CEC2020

Table 1 presents the basic information of the CEC2020 test functions. In this section, we computed the best value, mean, and standard deviation from 20

independent runs of each test function, with the best values highlighted in bold. The best value reflects the highest search accuracy achievable by the algorithm, while the mean and standard deviation are used to comprehensively evaluate the overall performance and robustness of the algorithm. For dimension D=10, the statistical results obtained by each algorithm after 20 independent runs on the 10 test functions are shown in Table 3. As can be seen from Table 3, IESC outperforms all competing algorithms across all three evaluation metrics for most test functions in the CEC2020 benchmark; only on the F9 function does its mean value lag behind that of the APO algorithm, yet it still surpasses the other competing algorithms. This demonstrates that IESC possesses excellent global exploration and local development capabilities, enabling it to consistently converge to high-quality solutions. On the F2, F8, and F9 functions, although IESC did not achieve the best standard deviation, it still ranked among the top performers across all algorithms, further validating the algorithm's robustness and comprehensive performance advantages across diverse optimization problem types.

Table 3 Test results of IESC and the comparison algorithm on the benchmark functions (CEC2020, F1–F10)

Fn		IESC	ESC	PSO	GWO	GJO	SWO	APO	DE	MVO
F1	Best	**1.00E+02**	1.08E+02	**1.00E+02**	4.80E+03	1.93E+05	2.67E+09	1.21E+02	1.02E+02	2.00E+03
	Mean	**1.00E+02**	4.58E+02	1.94E+03	4.30E+07	1.43E+08	8.18E+09	7.13E+02	1.08E+02	7.61E+03
	Std	**1.21E-02**	3.04E+02	2.76E+03	1.83E+08	2.06E+08	2.73E+09	8.19E+02	4.80E+00	3.95E+03
F2	Best	**1.10E+03**	1.15E+03	1.22E+03	1.16E+03	1.54E+03	2.62E+03	1.11E+03	1.38E+03	1.26E+03
	Mean	**1.32E+03**	1.42E+03	1.86E+03	1.85E+03	2.02E+03	3.07E+03	1.40E+03	1.83E+03	1.72E+03
	Std	1.70E+02	1.72E+02	3.39E+02	2.56E+02	3.18E+02	1.85E+02	1.93E+02	**1.65E+02**	2.26E+02
F3	Best	**7.11E+02**	7.15E+02	7.19E+02	7.14E+02	7.30E+02	9.06E+02	7.22E+02	7.21E+02	7.18E+02
	Mean	**7.15E+02**	7.18E+02	7.29E+02	7.27E+02	7.66E+02	1.00E+03	7.48E+02	7.18E+02	7.28E+02
	Std	**1.77E+00**	1.94E+00	7.62E+00	7.14E+00	2.02E+01	5.65E+01	4.91E+00	3.17E+00	7.81E+00
F4	Best	**1.90E+03**	**1.90E+03**	**1.90E+03**	**1.90E+03**	**1.90E+03**	4.31E+03	**1.90E+03**	**1.90E+03**	**1.90E+03**
	Mean	**1.90E+03**	**1.90E+03**	**1.90E+03**	**1.90E+03**	2.40E+03	8.62E+04	**1.90E+03**	**1.90E+03**	**1.90E+03**
	Std	**2.41E-01**	2.67E-01	4.95E-01	1.56E+00	2.15E+03	8.92E+04	5.26E-01	3.72E-01	5.82E-01
F5	Best	**1.73E+03**	2.22E+03	2.03E+03	2.72E+03	2.58E+03	6.81E+04	1.74E+03	3.49E+03	2.59E+03
	Mean	**1.78E+03**	6.62E+03	4.45E+03	4.30E+04	2.17E+05	1.73E+06	1.80E+03	7.21E+03	6.92E+03
	Std	**1.65E+02**	6.34E+03	2.32E+03	1.07E+05	2.80E+05	1.09E+06	1.82E+02	4.08E+03	4.42E+03
F6	Best	**1.60E+03**	**1.60E+03**	**1.60E+03**	**1.60E+03**	1.65E+03	1.81E+03	**1.60E+03**	**1.60E+03**	**1.60E+03**
	Mean	**1.60E+03**	**1.60E+03**	1.80E+03	1.74E+03	1.81E+03	2.31E+03	1.61E+03	1.61E+03	1.72E+03
	Std	**2.83E+00**	3.73E+00	1.06E+02	1.27E+02	1.02E+02	1.64E+02	2.61E+01	1.21E+01	9.67E+01
F7	Best	**2.10E+03**	2.12E+03	2.18E+03	3.26E+03	6.20E+03	1.49E+04	**2.10E+03**	2.12E+03	2.85E+03
	Mean	**2.11E+03**	2.57E+03	3.50E+03	8.91E+03	1.47E+06	3.85E+05	2.12E+03	2.15E+03	6.56E+03
	Std	**5.84E+00**	7.88E+02	1.48E+03	5.69E+03	1.59E+06	3.96E+05	7.87E+00	2.40E+02	4.33E+03
F8	Best	**2.23E+03**	2.30E+03	**2.23E+03**	2.30E+03	2.31E+03	2.53E+04	**2.23E+03**	2.24E+03	2.30E+03
	Mean	**2.28E+03**	2.30E+03	2.32E+03	2.32E+03	2.41E+03	3.11E+04	2.29E+03	2.29E+03	2.33E+03
	Std	1.40E+01	**5.44E-01**	1.03E+02	5.39E+01	2.99E+02	2.82E+02	3.28E+01	1.69E+01	9.83E+01
F9	Best	**2.50E+03**	2.73E+03	**2.50E+03**	2.73E+03	2.54E+03	2.78E+03	**2.50E+03**	2.62E+03	**2.50E+03**
	Mean	2.58E+03	2.74E+03	2.74E+03	2.75E+03	2.75E+03	2.87E+04	**2.52E+03**	2.75E+03	2.72E+03
	Std	1.02E+01	**2.22E+00**	1.34E+02	1.16E+01	7.08E+01	3.61E+01	7.01E+01	3.02E+01	7.46E+01
F10	Best	**2.90E+03**	2.94E+03	**2.90E+03**	**2.90E+03**	**2.90E+03**	3.05E+03	**2.90E+03**	**2.90E+03**	**2.90E+03**
	Mean	**2.90E+03**	2.95E+03	2.92E+03	2.94E+03	2.97E+03	3.44E+03	2.92E+03	2.91E+03	2.92E+03
	Std	**2.53E+00**	3.49E+00	2.28E+01	1.44E+01	6.33E+01	1.96E+02	2.28E+01	1.15E+01	2.31E+01

4.2 Statistical Analysis of CEC2020

Table 4 Rank-sum test results of IESC and the comparison algorithm on the benchmark functions (CEC2020, F1–F10)

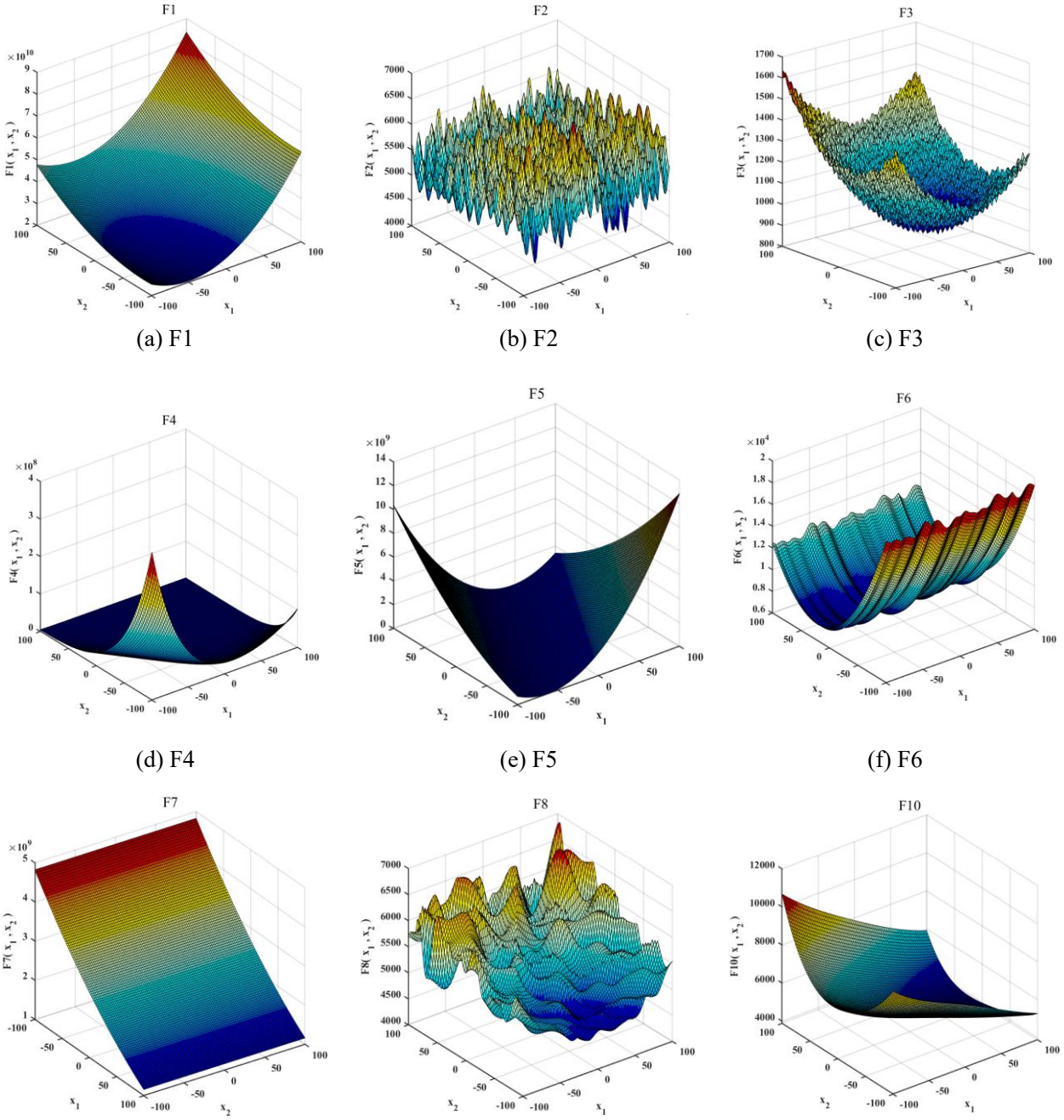
Fn	ESC	DE	PSO	GWO	SWO	GJO	APO	MVO
F1	6.80E-08	6.92E-07	6.80E-08	6.80E-08	6.80E-08	6.80E-08	6.80E-08	6.80E-08
F2	5.26E-05	2.29E-02	3.60E-02	9.09E-03	6.80E-08	5.26E-05	6.36E-02	9.09E-03
F3	6.00E-03	2.22E-04	1.33E-02	6.80E-08	6.80E-08	1.79E-02	1.20E-06	1.50E-03
F4	2.73E-01	1.81E-03	3.37E-01	9.17E-08	6.80E-08	2.00E-04	9.75E-06	3.94E-02
F5	6.80E-08	6.80E-08	6.80E-08	6.80E-08	6.80E-08	2.36E-06	6.80E-08	6.80E-08
F6	6.87E-04	1.05E-06	9.13E-07	6.80E-08	6.80E-08	3.07E-06	9.13E-07	4.60E-04
F7	2.22E-07	6.80E-08	6.80E-08	6.80E-08	6.80E-08	1.79E-01	5.23E-07	6.80E-08
F8	1.80E-02	3.07E-06	1.43E-07	6.80E-08	6.80E-08	1.64E-02	6.22E-04	6.80E-08
F9	2.22E-04	2.10E-03	1.25E-05	6.92E-07	6.80E-08	1.87E-06	1.66E-07	1.16E-04
F10	1.06E-07	4.39E-01	5.23E-07	1.92E-07	6.80E-08	2.18E-02	2.06E-06	3.05E-04

Based on the results of the Wilcoxon rank-sum test presented in Table 4, this study further validates, from a statistical perspective, the performance differences between IESC and the various comparison algorithms on the CEC2020 test functions. This test determines whether there is a statistically significant difference between the optimization results of the two groups by calculating a p-value, with a threshold of $p < 0.05$ commonly adopted to indicate a significant difference. As shown in the table, the p-values for IESC compared to the vast majority of the comparison algorithms are all below 0.05, indicating that IESC is statistically significantly superior to these algorithms. These findings support the earlier conclusion that IESC exhibits advantages in convergence accuracy, robustness, and overall performance.

4.3 Qualitative Analysis of CEC2020

Figs.5-8 provide a qualitative analysis of the optimization process and performance of IESC on the CEC2020 benchmark functions from four perspectives: Fig.5 shows the three-dimensional morphology of the test functions; Fig.6 displays the historical distribution of all search positions across 1,000 iterations; Fig.7 illustrates the iterative variation of the population's average fitness; and Fig.8 presents a comparative analysis of the average convergence curves of IESC against other advanced algorithms.

Analysis of the search history distribution reveals that the algorithm initially explores a wide range of regions, gradually focuses on promising areas, and ultimately conducts intensive exploration near the optimal solution, thereby achieving an effective balance between exploration and exploitation. The average fitness of the population exhibits a monotonically decreasing trend, confirming the stability of the algorithm's convergence. Comparative convergence curves demonstrate that IESC achieves rapid convergence and high precision on most test functions, effectively escapes local optima, and approaches the global optimum with fewer iterations. These results collectively highlight IESC's superior performance in terms of exploration capability, exploitation efficiency, convergence speed, and solution accuracy.

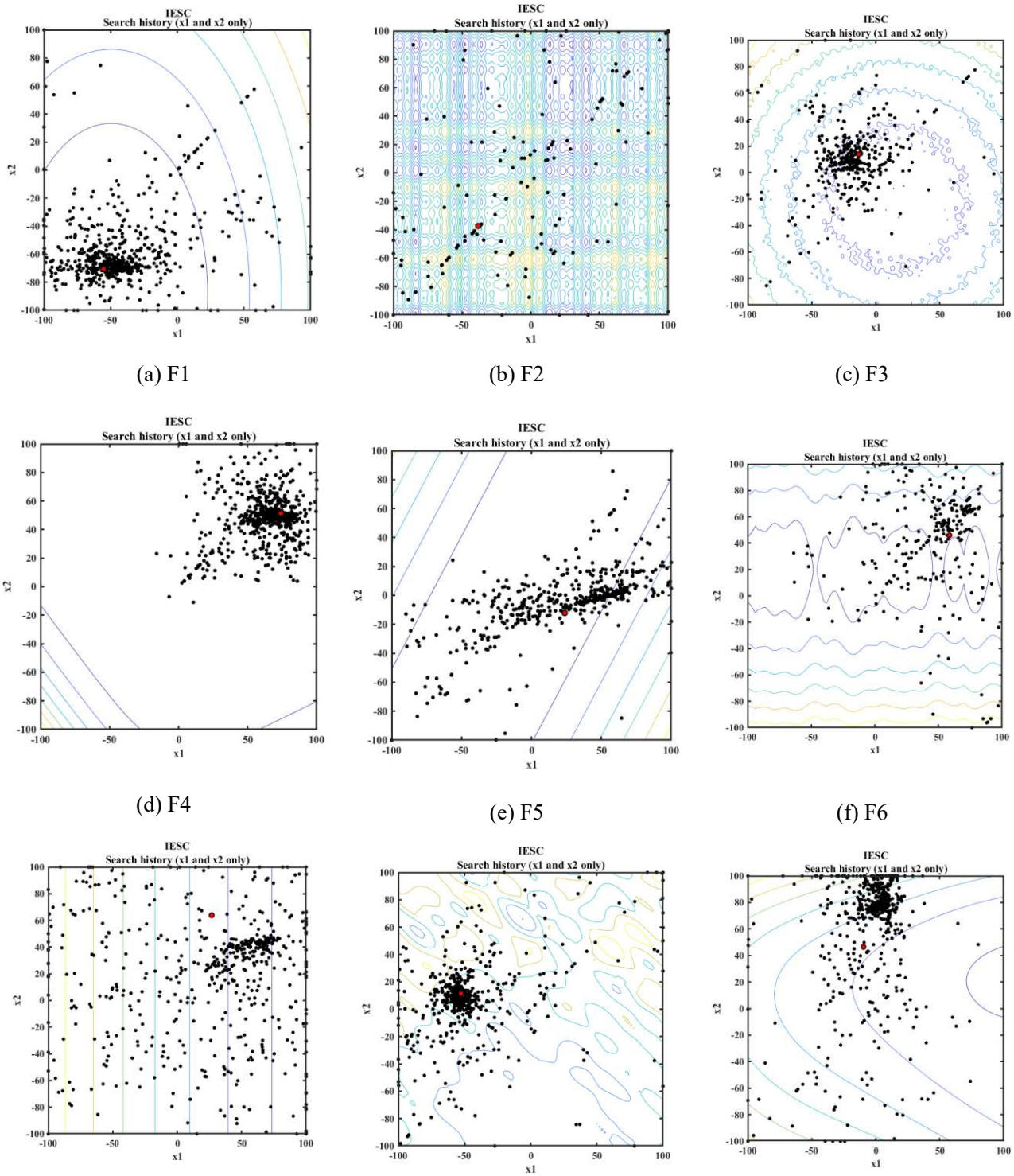


(a) F1 (b) F2 (c) F3

(d) F4 (e) F5 (f) F6

(g) F7 (h) F8 (i) F10

Fig.5 Three-dimensional image of the IESC CEC2020 test function

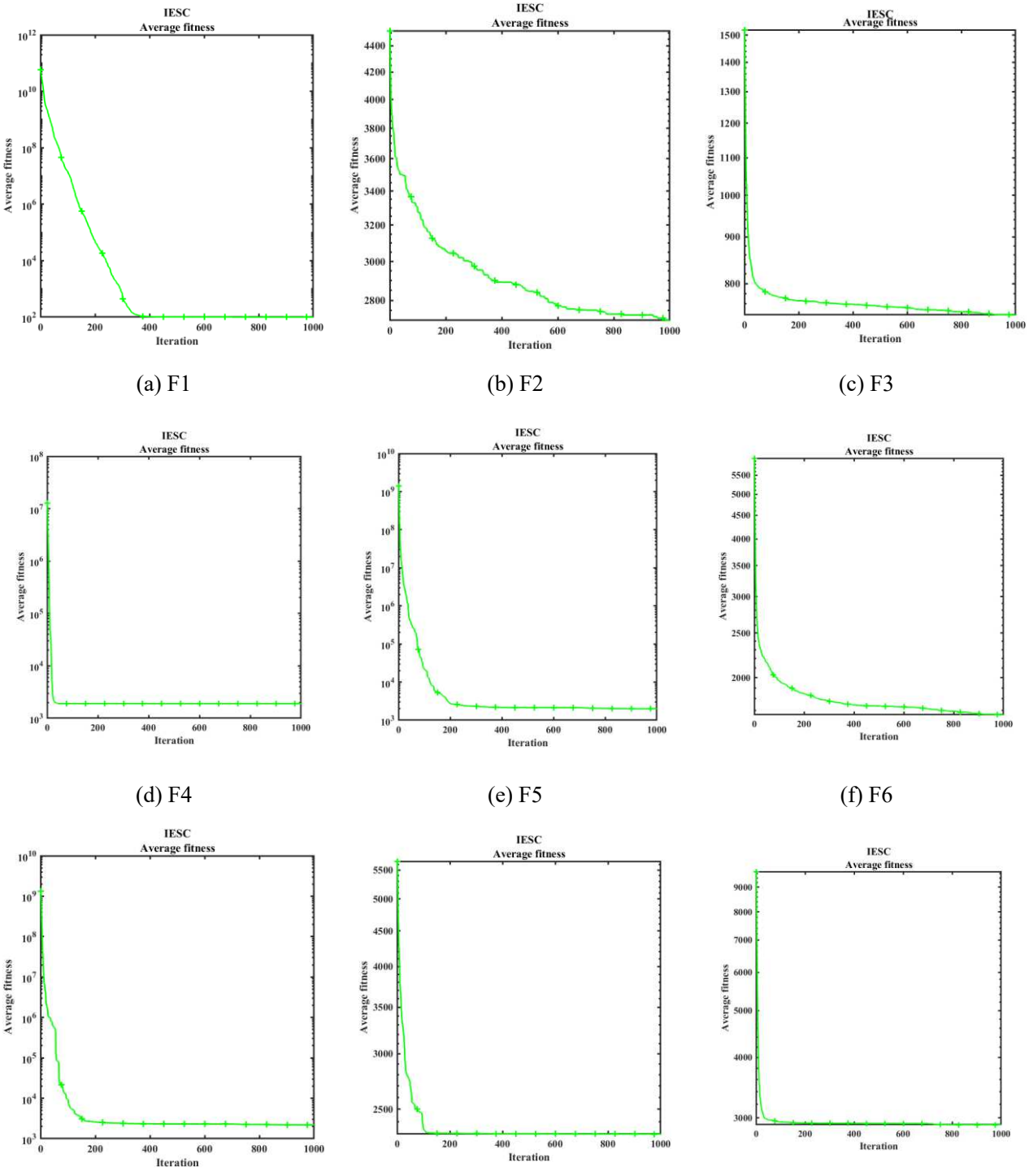


(a) F1 (b) F2 (c) F3

(d) F4 (e) F5 (f) F6

(g) F7 (h) F8 (i) F10

Fig.6 Historical plot of the IESC CEC2020 test function search results

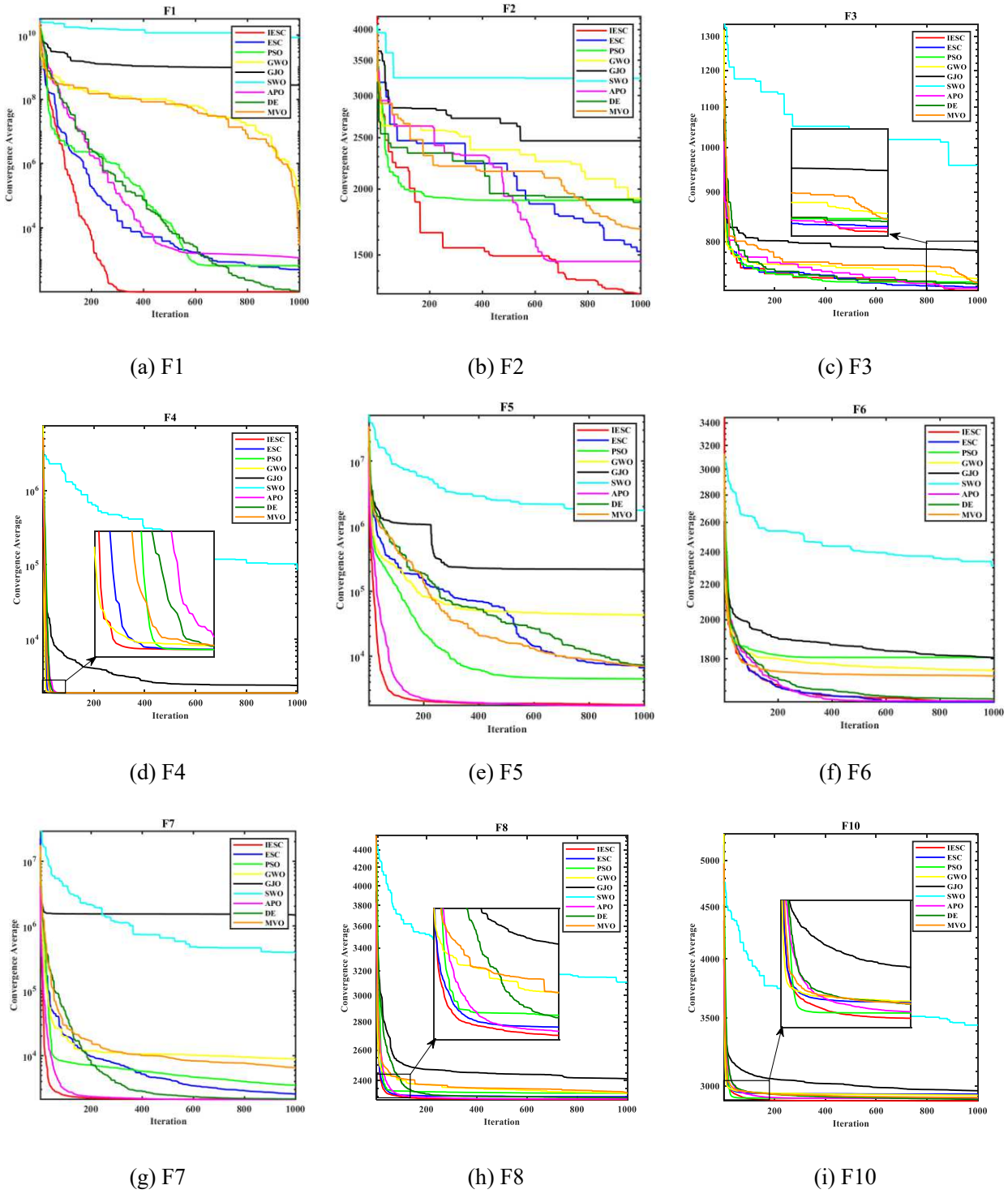


(a) F1 (b) F2 (c) F3

(d) F4 (e) F5 (f) F6

(g)F7 (h) F8 (i) F10

Fig.7 Average fitness evolution curve of the test function population for IESC CEC2020



(a) F1 (b) F2 (c) F3

(d) F4 (e) F5 (f) F6

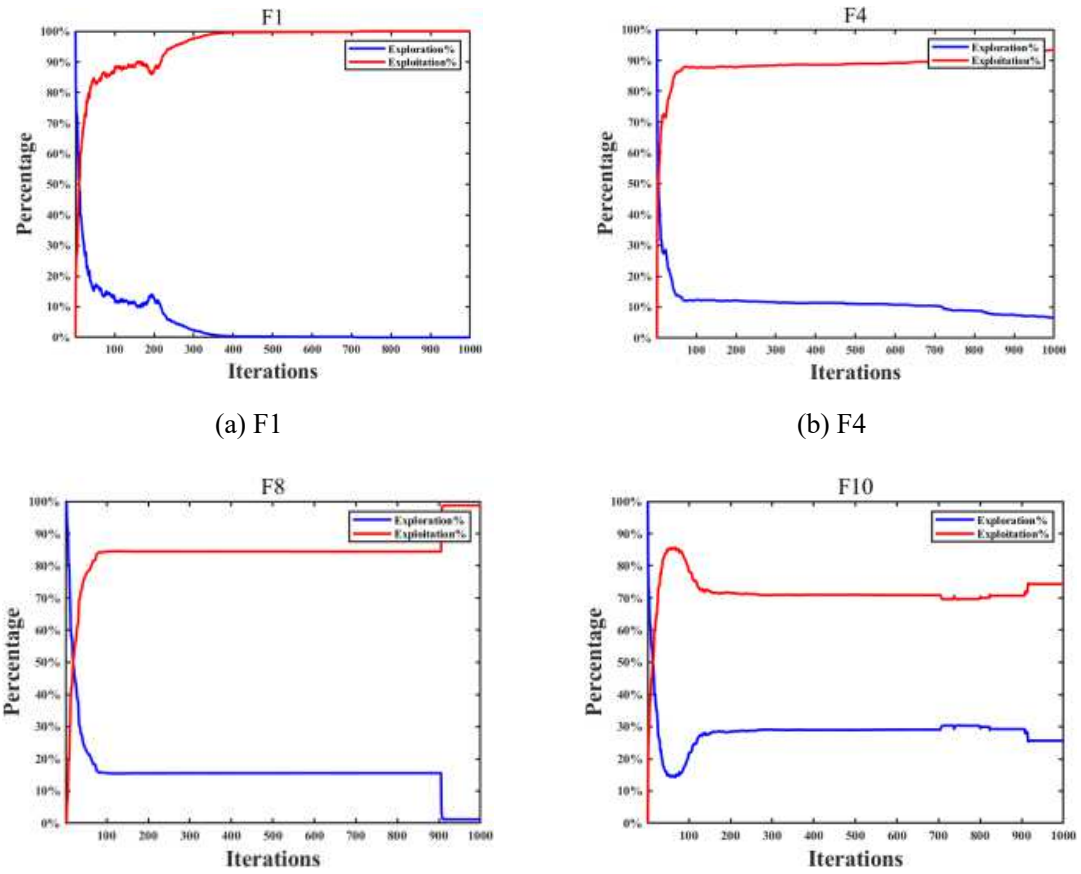
(g) F7 (h) F8 (i) F10

Fig.8 Comparison of average convergence curves for the IESC CEC2020 test functions

4.4 Exploration and Development Analysis

In metaheuristic algorithms, the appropriate coordination between global exploration and local exploitation is crucial for determining optimization performance, while the distribution differences of population individuals across various dimensions of the solution space can serve as a criterion for evaluating the search phase. Fig.9 illustrates the variation in the exploration-to-exploitation ratio of IESC across selected test functions of CEC2020 with the number of iterations. As shown in the figure, during the initial iterations, the algorithm predominantly focuses on exploration,

covering a broad search region; as iterations progress, the exploitation ratio gradually increases and eventually surpasses the exploration ratio, driving individuals toward the optimal region. This dynamic transition demonstrates that IESC can adaptively adjust its search strategy—ensuring thorough exploration in the early stages to avoid premature convergence, while shifting toward refined search in later stages to enhance solution accuracy—thereby exhibiting robust capabilities in both global optimization and local refinement.



(a) F1 (b) F4

(c) F8 (d) F10

Fig.9 IESC CEC2020 exploration-to-development ratio curve

5.Application of IESC to 3D UAV Path Planning

This section addresses the three-dimensional path planning problem for drones [11] by constructing a drone cost function that integrates optimality criteria with multiple constraints, thereby transforming the path planning problem into a constrained optimization problem.

5.1 Environmental Modeling and Path Planning Constraints

The main optimization objective of UAV path planning is to minimize the total length, and the flight path is set as a series of waypoint sequences by the ground control station. Each waypoint corresponds to a coordinate node in the search map, denoted as $P_{i,j} = (x_{i,j}, y_{i,j}, z_{i,j})$. By calculating the Euclidean distance between adjacent nodes, the total length of the entire path containing n waypoints can be obtained from Eq.(20).

$$F_1(X_i)=\sum_{j=1}^{n-1}\left\|\overrightarrow{P_{i,j}P_{i,j+1}}\right\| \tag{20}$$

To ensure the safe flight of the UAV, it is essential to effectively mitigate potential threats within the mission airspace. In this paper, all threats are modeled as cylinders, as illustrated in Fig.10; each threat *k* is defined by its center coordinates C_k and its radius R_k. For a given path segment $\overrightarrow{P_{i,j}P_{i,j+1}}$, its threat cost depends on the distance between that segment and the threat center point. Additionally, the UAV's diameter D and the safety margin S of the collision zone are taken into account, and the corresponding threat costs are given by Eqs. (21) and (22), respectively.

$$F_2(X_i)=\sum_{j=1}^{n-1}\sum_{k=1}^{K}T_k(\overrightarrow{P_{i,j}P_{i,j+1}}) \tag{21}$$

$$T_k(\overrightarrow{P_{i,j}P_{i,j+1}})=\begin{cases}0, & if\ d_k>S+D+R_k\\ (S+D+R_k)-d_k, & if\ D+R_k<d_k\le S+D+R_k\\ \infty, & if\ d_k\le D+R_k\end{cases} \tag{22}$$

The size of a drone is determined by its diameter S, while the safety distance D is influenced by factors such as the specific application scenario, operational conditions, and positioning accuracy. In a static environment where positioning signals are reliable, S can be set to several tens of meters; however, in environments with dynamic obstacles or poor positioning signals, S may need to be extended to several hundred meters to ensure safety.

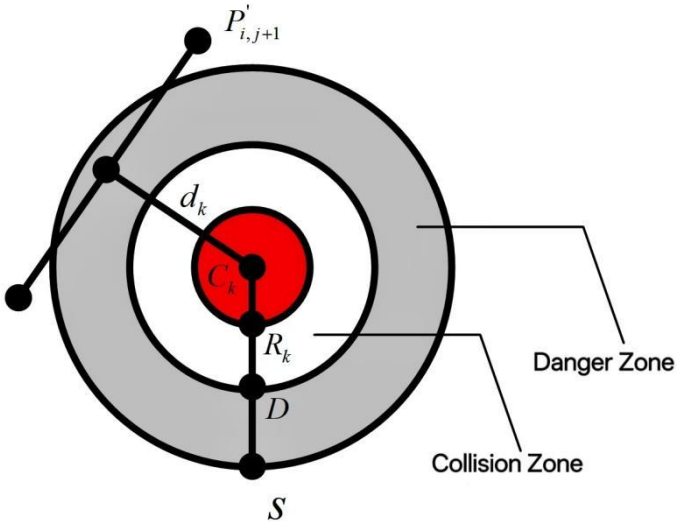


Fig.10 Drone Threat Costs

The flight altitude is typically constrained between a minimum altitude h_{min} and a maximum altitude h_{max}. To ensure the flight safety of the drone, constraints on the flight altitude must be imposed. Let $h_{i,j}$ denote the flight altitude relative to the ground, as illustrated in Fig.11; the altitude cost $H_{i,j}$ for the waypoint $P_{i,j}$ is then calculated according to Eq. (23).

$$H_{i,j}=\begin{cases}\left|h_{i,j}-\dfrac{(h_{\max}+h_{\min})}{2}\right|, & if\ h_{\min}\le h_{i,j}\le h_{\max}\\ \infty, & otherwise\end{cases} \tag{23}$$

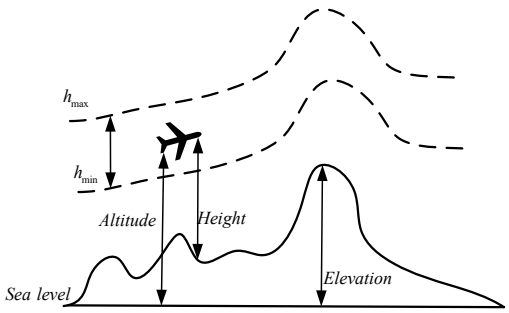


Fig.11 Drone altitude cost

Eq.(23) ensures that the flight altitude remains close to the average altitude, while imposing penalties for values that exceed this range. By summing the altitude costs across all waypoints, the total altitude cost is obtained, as shown in Eq. (24).

$$F_3(X_i) = \sum_{j=1}^{n-1} H_{i,j} \tag{24}$$

The smoothness cost is used to evaluate the rates of turning and climbing, which is crucial for generating feasible trajectories that comply with UAV dynamics. As shown in Fig.12, the turning angle $\phi_{i,j}$ is defined as the angle between the projected vectors of two consecutive path segments on the horizontal plane *Oxy*. Let $\vec{k}$ denote the unit vector along the z-axis; the projected vectors are given by Eq. (25):

$$\overrightarrow{P'_{i,j}P'_{i,j+1}} = \vec{k} \times (\overrightarrow{P_{i,j}P_{i,j+1}} \times \vec{k}) \tag{25}$$

The calculation $\phi_{i,j}$ formula for the turning angle is given by Eq. (26).

$$\phi_{i,j} = arctan\left(\frac{\left\|\overrightarrow{P'_{i,j}P'_{i,j+1}} \times \overrightarrow{P'_{i,j+1}P'_{i,j+2}}\right\|}{\overrightarrow{P'_{i,j}P'_{i,j+1}} \cdot \overrightarrow{P'_{i,j+1}P'_{i,j+2}}}\right) \tag{26}$$

The climb angle $\psi_{i,j}$ is defined as the angle between a path segment $\overrightarrow{P_{i,j}P_{i,j+1}}$ and its projection onto the horizontal plane $\overrightarrow{P'_{i,j}P'_{i,j+1}}$; the calculation formula is given by Eq.(27).

$$\phi_{i,j} = arctan\left(\frac{z_{i,j+1} - z_{i,j}}{\left\|\overrightarrow{P'_{i,j}P'_{i,j+1}}\right\|}\right) \tag{27}$$

The smooth cost F4 is jointly determined by the variations in the turn angle $\phi_{i,j}$ and the climb angle $\psi_{i,j}$, as shown in Eq.(28).

$$F_4(X_i) = a_1 \sum_{j=1}^{n-1} \phi_{i,j} + a_2 \sum_{j=1}^{n-1} \left|\psi_{i,j} - \psi_{i,j-1}\right| \tag{28}$$

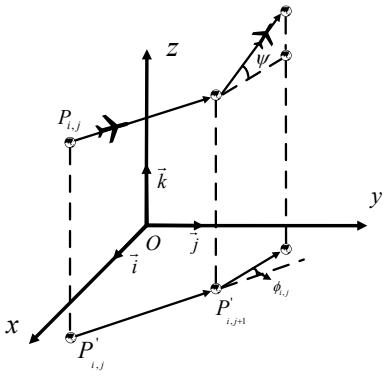


Fig.12 Turning slope angle calculation diagram

In Eq.(28), a_1 and a_2 are the penalty factors for the turning angle and the climb angle, respectively. By integrating constraints on path length, safety, altitude, and smoothness, the total cost function for the UAV path X_i is defined as shown in Eq. (29).

$$F(X_i) = \sum_{m=1}^{4} b_m F_m(X_i) \tag{29}$$

In Eq.(29), b_m denotes the weight coefficient, and F1 to F4 correspond to path length cost, threat cost, altitude cost, and smoothing cost, respectively. The decision variables X_i consist of n flight points located within the flight space.

In this experiment, the weight coefficients b_1, b_2, b_3, and b_4 were set to 5, 1, 10, and 1, respectively. The starting point was located at (0, 0, 0), and the endpoint was located at (100,100,100). Fig.13 illustrates the three-dimensional environment model of this experiment.

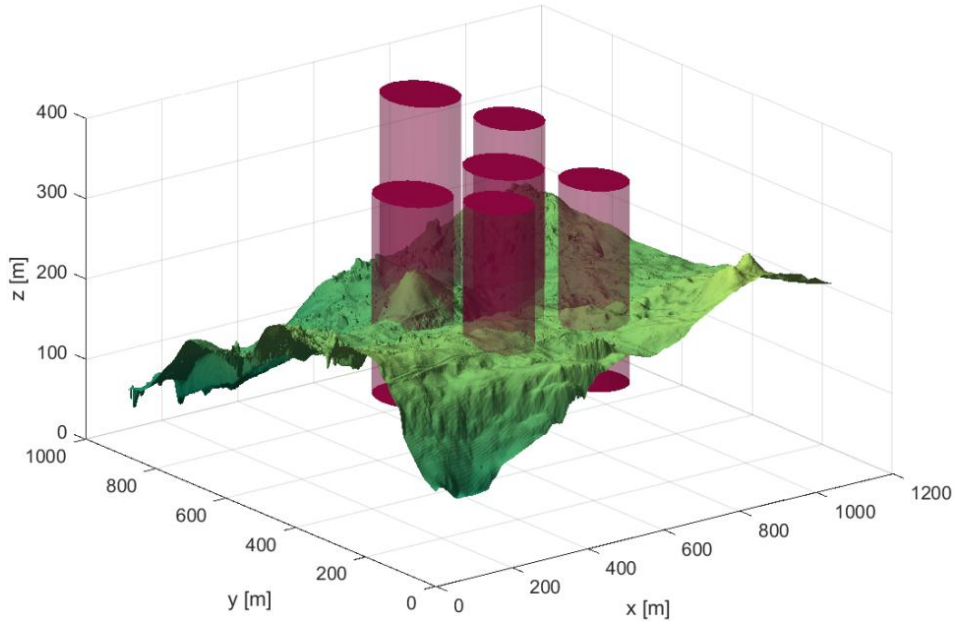


Fig.13 Three-dimensional environment modeling diagram

5.2 Simulation Analysis and Experimental Results

Table 5 Optimal Cost of IESC and the Benchmark Algorithm in 3D Path Planning

algorithm	optimal value	average value	standard error	crest value	median
IESC	**7.8075E+03**	**7.9254E+03**	**66.7402**	**8.1012E+03**	**7.9215E+03**
ESC	8.0681E+03	8.4593E+03	398.0809	9.6367E+03	8.2944E+03
GJO	9.4810E+03	1.8000E+11	3.6935E+11	9.0000E+11	1.7975E+04
GWO	7.8409E+03	9.0000E+10	2.7701E+11	9.0000E+11	8.5907E+03
DE	1.2961E+04	1.4253E+04	809.4592	1.6037E+04	1.4092E+04
APO	7.8410E+03	7.9537E+03	85.7703	8.1431E+03	7.9295E+03
MVO	8.4747E+03	9.3148E+03	559.1446	1.0246E+04	9.3366E+03
PSO	1.5318E+04	6.3000E+11	4.2315E+11	9.0000E+11	9.0000E+11
SWO	1.5216E+04	1.3500E+12	1.1119E+12	2.7000E+12	1.8000E+12

This experiment compared nine algorithms—IESC, ESC, SWO, PSO, MVO, GWO, GJO, DE, and APO—using uniform parameters: population size of 30, maximum number of iterations set to 500, and 20 independent runs per scenario. The results are presented in Table 5. The data indicate that IESC achieves the best overall performance, with an optimal cost of 7.8075E+03, outperforming all other compared algorithms; its average cost of 7.9254E+03 is the lowest among all algorithms, with a standard deviation of only 66.7402, reflecting minimal variability; its median cost of 7.9215E+03 closely matches its mean value, and its maximum cost of 8.1012E+03 also remains relatively low, demonstrating IESC's robustness and consistency. Although GWO and APO achieve optimal solutions in individual runs that are comparable to those of IESC, their average performance and robustness are inferior to those of IESC.All compared algorithms were able to generate feasible paths. As shown in Fig.14, the IESC trajectory exhibits smooth transitions, clear obstacle avoidance, and high overall rationality, effectively balancing path smoothness, safety, and feasibility. This demonstrates its robust path planning capability in complex three-dimensional environments.

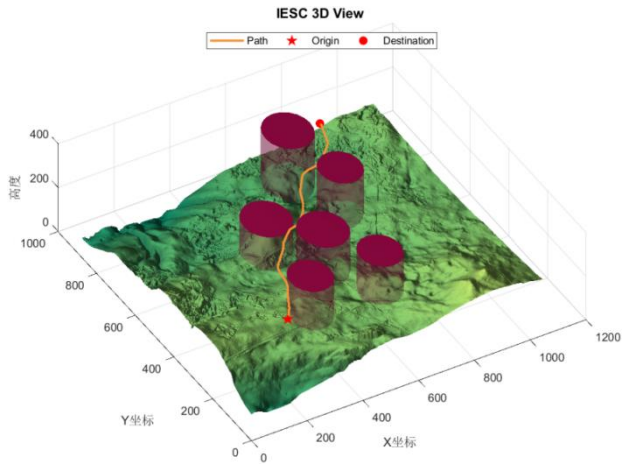
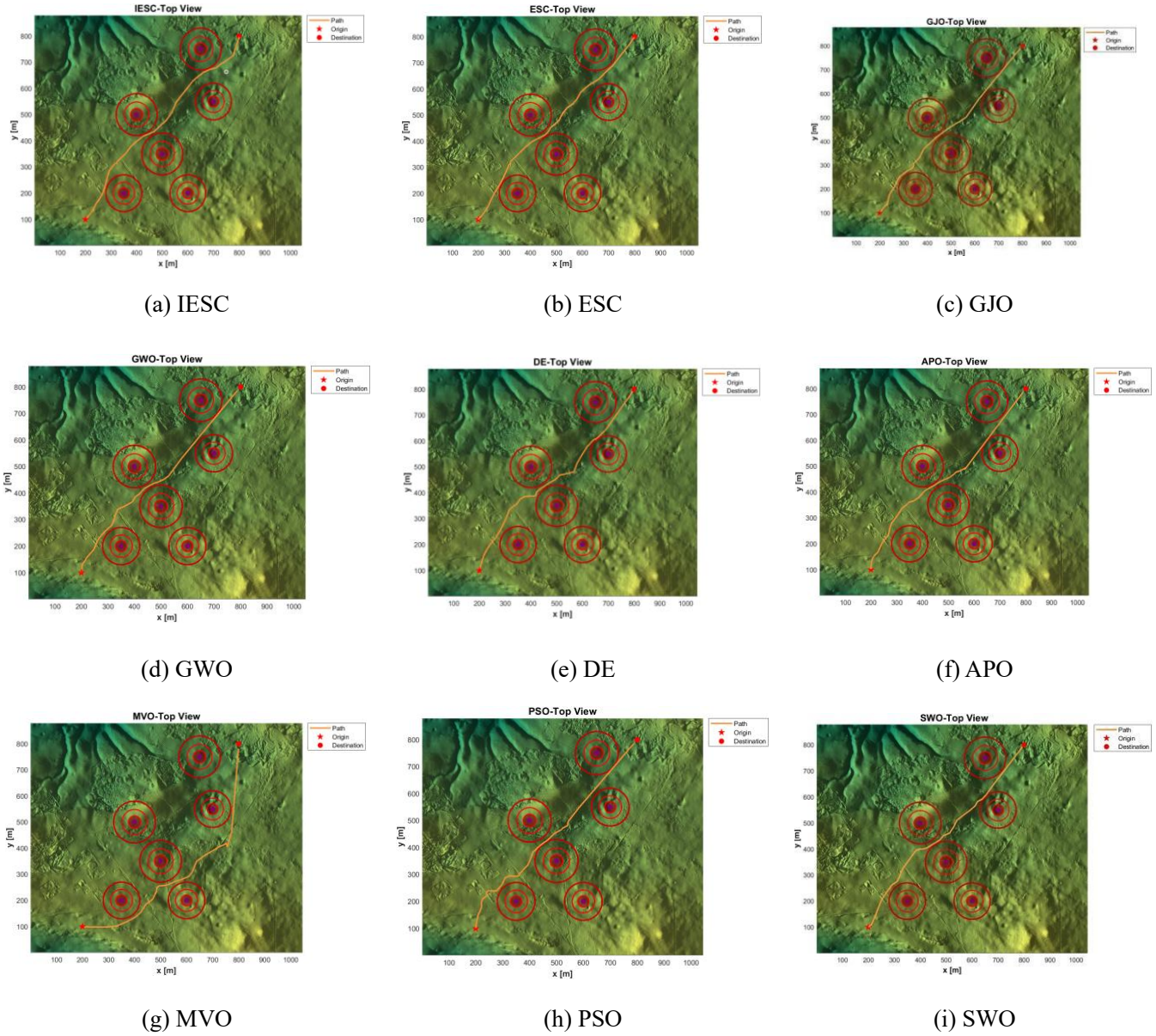


Fig.14 3D view of the optimal path of the IESC algorithm

Fig.15 shows that, in the top view, the IESC path smoothly navigates around obstacles with minimal turns, a continuous trajectory, and no redundant backtracking; compared to those of other algorithms, it is shorter and smoother, demonstrating an optimal balance between planar obstacle avoidance and path efficiency.



(a) IESC (b) ESC (c) GJO

(d) GWO (e) DE (f) APO

(g) MVO (h) PSO (i) SWO

Fig.15 Top-view of the optimal paths for different comparison algorithms

Fig.16 presents a comparative plot of the convergence curves of the optimal path results obtained by different algorithms.

Fig.16 shows that the convergence curve of IESC rapidly decreases and stabilizes within approximately 100 iterations, exhibiting fast and stable convergence without significant oscillations or premature plateauing. Compared to other algorithms, IESC can quickly approach the high-quality solution region in the early stages of iteration and then refine its solution in the later stages, demonstrating the effectiveness of its multi-strategy approach in enhancing both global exploration and local refinement capabilities while avoiding getting trapped in local optima.

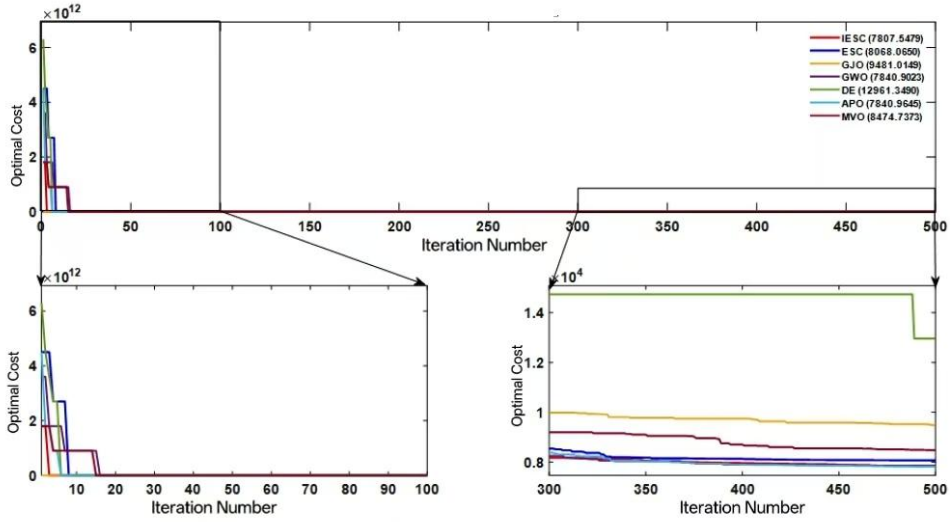


Fig.16 Comparison of convergence curves for different comparison algorithms

6. Summary and Outlook

6.1 Summary

This paper addresses the limitations of the escape algorithm in three-dimensional UAV path planning—such as low initial population quality, imbalanced exploration and development phases, susceptibility to local optima, and inadequate performance in high-dimensional spaces — by proposing a multi-strategy enhanced escape algorithm (IESC). After elaborating on the fundamental principles and limitations of the ESC algorithm, this work integrates five improvement strategies: (1) introducing a piecewise chaotic mapping for initialization to enhance population quality; (2) employing lens-based inverse learning to augment diversity; (3) designing a nonlinear attenuation-vibration hybrid factor to achieve a dynamic balance between exploration and development; (4) incorporating an elite-guided gradient simulation acceleration strategy to provide directional information for local search; and (5) introducing an experience exchange strategy with three-stage dynamic adjustments to improve convergence. The synergistic application of these strategies significantly enhances the algorithm's global search capability and convergence accuracy. Experimental results on the CEC2020 benchmark functions demonstrate that IESC outperforms mainstream algorithms—including ESC, PSO, DE, GWO, MVO, GJO, SWO, and APO—on the majority of test functions; the Wilcoxon test confirms the statistical significance of these improvements, while variations in the exploration-to-development phase ratio validate its adaptive balancing capability. Simulation results for three-dimensional UAV path planning further show that IESC surpasses the comparative algorithms in both optimal cost and stability, capable of generating high-quality trajectories characterized by smooth transitions, clear obstacle avoidance, and optimal feasibility, thereby confirming its effectiveness in solving complex engineering optimization problems.

6.2 Outlook

Future research may focus on the following directions: developing online replanning capabilities for dynamic environments by integrating real-time sensing

data; integrating IESC into actual UAV flight control systems for physical validation; and extending these algorithms to other engineering optimization problems—such as robot path planning and autonomous driving—to further test their generalizability.

Acknowledgements Supported by Natural Science Foundation of Shaanxi Province (No.2025JC-YBMS-736, 2025JC-YBMS-343), and the Shaanxi Province Key Research and Development Project (No.2025CY-YBXM-061).

Author contribution J. W. and L. G. completed the algorithmic principles while writing the main parts of the manuscript. Also, J. W. and L. G. completed the experiments and computed the data for the manuscript. S. D. checked the format of the manuscript. All authors reviewed the manuscript.

Data availability No datasets were generated or analyzed during the current study.

Declarations

Conflict of interest The authors declare no competing interests

References

[1] Deng Yun. Application of a Yuan Heuristic Intelligent Optimization Algorithm in Path Planning for UAV-Based Forest Disaster Relief [D]. Hubei: Hubei University of Technology, 2024.

[2] Ouyang K, Fu S, Chen Y, et al. Escape: an optimization method based on crowd evacuation behaviors[J]. Artificial Intelligence Review, 2024, 58(1): 19.

[3] Xue Yang, Ni Dabin, Lu Qiuhong, et al. Path planning for mobile robots based on the PGWO algorithm [J]. Control and Decision, 2025,40(4):1395–1401.

[4] Lu Sha, Xu Lizhuo, Wang Wenjing. An arithmetic optimization algorithm based on lens imaging inverse learning [J]. Journal of Nanning Normal University (Natural Science Edition), 2025,42(3):89–99.

[5] Long Wen, Wu Tiebin, Tang Mingzhu, et al. A gray wolf optimization algorithm based on lens imaging learning strategy [J]. Journal of Automation, 2020,46(10):2148–2164.

[6] He Fengling, Liu Baohong, Ding Xiaoxue, et al. An improved African vulture optimization algorithm based on chaotic lens imaging inverse learning [J]. Journal of Kunming University, 2026,48(3):65–80.

[7] Shen Yue, Chen Limin, Wang Yidi. An optimization algorithm integrating differential evolution and lens imaging for black-winged kites [J]. Journal of Mudanjiang Normal University (Natural Science Edition), 2025(4):9–13.

[8] Fan Youmin. Adaptive Parameter Setting and Global Optimization Enhancement of the Gradient Descent Method [D]. Lanzhou: Northwest Normal University, 2024.

[9] Li Duoxiao, Zhao Jian. An adaptive control algorithm for spotted kingfishers integrating right-angle walking and two-stage escape [J]. Journal of Liaoning University of Science and Technology, 2026,49(2):117–126.

[10] Ma Pengge, Zhang Yan, Yang Hai, et al. Multi-drone 3D path planning based on the GDT-PSO algorithm [J/OL]. Electronics, Optics and Control, 2026. https://link.cnki.net/urlid/41.1227.tn.20260819.1455.014.

[11] Liu Jiangting, Zhu Shunkang, Gu Qiuyi, et al. A 3D path planning algorithm for UAVs based on the integration of ISPSO and ACO [J]. Radio Engineering, 2025,55(4):866–876.

[12] Huang, C. Improved PSO Algorithm for 3D Environmental Path Planning of Drones [J]. Surveying, Mapping and Spatial Geographic Information, 2026,49(7):136–139.

[13] Zhao Yuheng, Liang Xiaolei, Zhang Dongmei, et al. An improved swan optimization algorithm based on multi-strategy fusion [J]. Computer Engineering & Design, 2026,47(1):55–63.

[14] Liu Yunping, Zhu Huiru, Fang Weihua. Improved Grey Wolf Algorithm for UAV Path Planning [J]. Electronics, Optics and Control, 2023,30(7):1-7.

[15] Chen Liwei, Ma Zehua, Wang Tong, et al. Path planning for unmanned aerial vehicles based on an improved Harris eagle algorithm [J]. Applied Science & Technology, 2024,51(2):17–23,30.

[16] Wang Haiqun, Song Guozhang, Ge Chao. Three-dimensional path planning for UAVs based on an improved Centroidal Ant Colony Algorithm [J]. Electronics, Optics and Control, 2024,31(11):55–61,82.

[17] Zhao Y, Li X. CCESC: a crisscross-enhanced escape algorithm for global and reservoir production optimization[J]. Biomimetics, 2025, 10(8): 529.