Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2026.073861

ARTICLE

Dynamic Weighted Spherical Particle Swarm Optimization for UAV Path Planning in Complex Environments

Rui Yao[1,2], Yuye Wang[1,2,*], Fei Yu[1,2,3,*], Hongrun Wu[1,2] and Zhenya Diao[1,2]

[1]College of Physics and Information Engineering, Minnan Normal University, Zhangzhou, 363000, China
[2]Key Lab of Intelligent Optimization and Information Processing, Minnan Normal University, Zhangzhou, 363000, China
[3]Key Lab of Light Field Manipulation and System Integration Applications in Fujian Province, Zhangzhou, 363000, China

*Corresponding Authors: Yuye Wang. Email: wyuye2002@mnnu.edu.cn; Fei Yu. Email: yufei@whu.edu.cn

Received: 27 September 2025; Accepted: 22 December 2025; Published: 12 March 2026

ABSTRACT: Path planning for Unmanned Aerial Vehicles (UAVs) in complex environments presents several challenges. Traditional algorithms often struggle with the complexity of high-dimensional search spaces, leading to inefficiencies. Additionally, the non-linear nature of cost functions can cause algorithms to become trapped in local optima. Furthermore, there is often a lack of adequate consideration for real-world constraints, for example, due to the necessity for obstacle avoidance or because of the restrictions of flight safety. To address the aforementioned issues, this paper proposes a dynamic weighted spherical particle swarm optimization (DW-SPSO) algorithm. The algorithm adopts a dual Sigmoid-based adaptive weight adjustment mechanism for balancing global exploration and local exploitation, as well as a lens-based opposition learning one to improve search flexibility and solution diversity. Simulation experiments on real digital elevation models demonstrate that DW-SPSO significantly outperforms recent state-of-the-art particle swarm optimization (PSO) variants in terms of path safety, smoothness, and convergence speed. The performance superiority is statistically validated by the Wilcoxon signed-rank test. The results confirm the algorithm's effectiveness in generating high-quality UAV paths under diverse threat conditions, offering a robust solution for autonomous navigation systems.

KEYWORDS: Dynamic weight adjustment; lens opposition learning; particle swarm optimization; path planning; unmanned aerial vehicles

1 Introduction

Over the past few decades, Unmanned Aerial Vehicles (UAVs) have been widely used for reconnaissance and surveillance, express delivery, and rescue purposes. A key part of the UAV navigation system is to plan the optimal flight routing. The flight route planning task involves identifying multiple flight waypoints from origin to destination while accounting for various operational constraints, including environmental factors and energy efficiency considerations [1].

Secure path planning for UAVs is highly important in improving the autonomy and intelligence of UAVs, thus attracting much attention from people. Concerning the path planning challenges for UAVs or robots, various techniques have been proposed by scholars both domestically and internationally [2,3]. For instance, the A* algorithm is capable of identifying near-optimal paths in simple, small-scale environments through heuristic cost evaluation. In complex and exponentially expanding search spaces, challenges arise in accurately estimating unknown path costs, leading to suboptimal solutions and increased computation

time [4]. In contrast, the traditional RRT with a random tree can efficiently find a feasible path by randomly sampling points in low-dimensional environments with limited knowledge, outperforming environment-dependent methods such as the Probabilistic Roadmap for single-query planning. However, due to its inherent cost-agnostic nature, RRT cannot guarantee optimality, exhibiting a low probability of achieving optimal solutions in large-scale search spaces and failing to meet practical demands for rapid, high-quality path generation [5,6].

Metaheuristic algorithms are different from classical methods in that they can solve complex combinatorial optimization problems better. Therefore, they were extensively employed in multi-UAV path-planning research. These are genetic algorithms (GA) [7] and differential evolution (DE) [8]. Algorithms for swarm intelligence include ant colony algorithm(ACO) [9], the artificial bee colony (ABC) algorithm [10], the moth flame optimization (MFO) [11], and the butterfly optimization algorithm (BOA) [12]. Further refinements of metaheuristics are evident in domains like energy systems, where a stable social learning swarm optimizer excelled in photovoltaic design optimization [13], highlighting a focus on stability and reliability—equally vital for robust UAV path planning. Particle swarm optimization (PSO) [14,15], in particular, has been extensively utilized, and numerous PSO variants have been proposed.

The PSO algorithm is a collective behavior inspired by the swarm intelligence observed in bird flocks and fish schools. PSO was selected for UAV path planning owing to its computational efficiency, rapid convergence, and adaptability to complex environments [16,17]. Unlike evolutionary algorithms that depend on mutation and crossover, PSO achieves stable convergence with lower computational cost by balancing individual and collective intelligence. Its parallel structure also facilitates real-time execution on embedded systems, ideal for UAV applications. Its effectiveness stems from two intrinsic swarm intelligence principles: cognitive coherence (individual experience) and social coherence (collective experience). In contrast to traditional evolutionary algorithms, which depend on mutation and crossover operations, PSO allows each particle to iteratively enhance solutions by dynamically balancing its own historical best performance with the swarm's global best. This unique mechanism allows PSO to converge stably toward near-optimal solutions, significantly cutting computational time compared to other nature-inspired methods. Moreover, PSO shows limited sensitivity to initial conditions and objective function variations, while adapting to complex environments through minor parametric adjustments, mainly involving an acceleration coefficient and two weighting factors. Novel analytical frameworks have advanced both the theoretical understanding and practical efficacy of metaheuristics. For instance, complex network theory has been employed to reveal how swarm connectivity influences PSO performance [18], while spherical vector-based and adaptive PSO variants have demonstrated significant improvements in path planning under threats and complex system optimization, respectively [19,20]. Owing to its inherently parallel architecture, PSO can be efficiently implemented on multi-core processors, GPUs, or distributed computing clusters, fulfilling real-time processing needs for both offline and online path planning scenarios [21]. Leveraging these benefits, the PSO algorithm is extensively utilized in UAV path planning, with several enhanced variants having been developed. Phung and Ha [22] developed a motion-encoded particle swarm optimization (MPSO) algorithm for UAV moving target search, encoding search trajectories as motion segments to preserve swarm coherence.

Observation shows that while current UAV path planning algorithms may enhance efficiency temporarily, they usually have a sluggish convergence rate and a tendency to be captured by local optima. Even more critically, they do not properly guarantee UAV operation safety. The spherical vector particle swarm optimization (SPSO) algorithm was proposed by Phung and Ha [14], which incorporates spherical vector encoding to inherently link with UAV kinematics, thereby guaranteeing path feasibility. However, conventional fixed weight approaches lack the capacity to dynamically balance exploration and exploitation, resulting in delayed response to emergent threats. Building on these insights and utilizing the flight characteristics of UAVs,

this paper presents a novel dynamic weighted spherical particle swarm optimization (DW-SPSO) algorithm. This advancement integrates real-time threat distance adaptive coefficient adjustment to optimize the trade-off between global exploration and local exploitation. Moreover, to increase route diversity under threat scenarios and enhance path safety, a lens imaging principle-based opposition learning strategy is integrated to boost the algorithm's exploratory and exploitative abilities.

The UAV path planning issue is articulated by establishing an objective function that is capable of accounting for differing needs and restrictions related to UAVs as well as to flight paths. A novel PSO algorithm with some improvements through strategic application is provided. This process leverages the UAV's configuration space to generate high-quality solutions. Using 2 real digital elevation model (DEM) maps, ChrismasTerrain and TerrainData, create both simple and complex terrain with 3 and 7 obstacles, respectively. The enhancement algorithm is compared with the original SPSO and other metaheuristic approaches. The innovations and contributions of this paper are as follows:

- Dual Sigmoids enable dynamic weight adjustment for balancing global exploration and local exploitation.
- A lens-based opposition learning strategy is integrated to boost algorithm flexibility and broaden the search range.
- Simulation experiments on both simple and complex terrains, and comparisons to other algorithms prove that the improvement of the algorithm is effective and practicable.
- The algorithm demonstrates significant advantages in UAV path-planning for complex terrains, verifying its practical utility.

The remainder of this paper is structured as follows. Section 2 formulates the threat environment model and defines the cost functions for UAV path planning. Section 3 details the proposed DW-SPSO algorithm, including the spherical vector encoding, adaptive weighting strategy, and lens-based opposition learning mechanism. Section 4 presents the experimental setup, simulation results, a comprehensive comparative analysis with state-of-the-art algorithms, and a statistical performance validation using the Wilcoxon signed-rank test. Finally, Section 5 concludes the paper by summarizing the findings and suggesting directions for future research.

2 Threat Environment Model

This study formulates the path planning problem into the following cost function, incorporating optimal criteria and relevant constraints for UAVs.

2.1 Optimal Path

Defining appropriate criteria tailored to specific operational scenarios is crucial for a UAV performing a mission. In this context, the objective is to minimize distance and utilize the flight path $X_i = (P_{i1}, P_{i2}, \ldots, P_{ij}, \ldots, P_{in})$, where n denotes the number of waypoints the UAV must visit. Each waypoint aligns with a path node on the map, with the path nodes' coordinates being $P_{ij} = (x_{ij}, y_{ij}, z_{ij})$, and for two path nodes, the Euclidean distance is $\|\overrightarrow{P_{ij}P_{i,j+1}}\|$, resulting in the cost for the path length:

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

2.2 Safety and Feasibility Constraints

UAV path planning: Completing the path planning requires ensuring the safe movement of the UAV by mitigating threats posed by obstacles. Let K denote the set of all threats, with each threat modeled as a cylinder. According to the definition illustrated in Fig. 1, C_k represents the projected center coordinate of the k obstacle on the horizontal plane, with a corresponding radius R_k. While prior studies often approximate the UAV as a dimensionless particle due to its compact size, this work explicitly incorporates the UAV's spatial occupancy. To enhance threat assessment precision. Specifically, the UAV is assigned a safety diameter D. For a path segment $\overrightarrow{P_{ij}P_{i,j+1}}$, let d_k denote the Euclidean distance from the obstacle center C_k to this segment. The associated threat cost is proportional to this distance d_k. By integrating these geometric parameters, the proposed framework enables a more rigorous quantification of collision risks during UAV navigation through threat-laden environments.

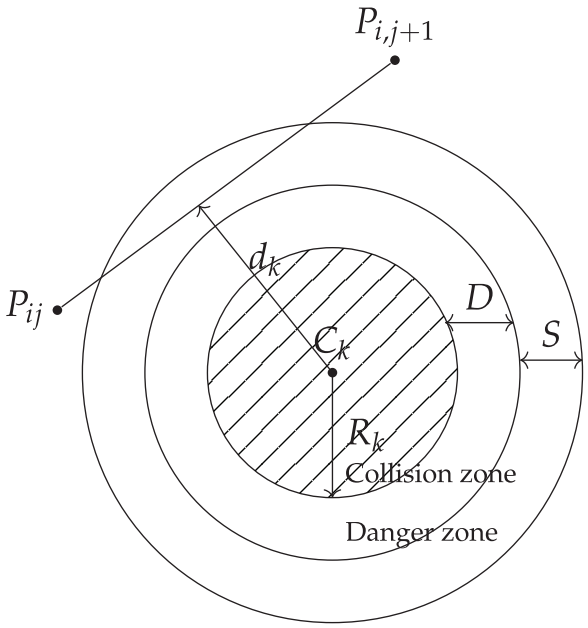


Figure 1: Determination of the threat cost

The threat cost F_2, a measure of how close the UAV is to obstacles, is computed by evaluating the distance from each waypoint P_{ij} to the collision zone within the obstacle set K, relative to the safety margin S.

$$\begin{cases} F_2(X_i) = \sum_{j=1}^{n-1} \sum_{k=1}^{K} T_k(\overrightarrow{P_{ij}P_{i,j+1}}), \\ T_k(\overrightarrow{P_{ij}P_{i,j+1}}) = \begin{cases} 0, & \text{if } d_k > S + D + R_k \\ (S + D + R_k) - d_k, & \text{if } D + R_k < d_k \le S + D + R_k \\ \infty, & \text{if } d_k \le D + R_k \end{cases} \end{cases} \tag{2}$$

The safety diameter D is determined by the physical dimensions of the UAV, while the safety margin S is influenced by various operational factors, including environmental conditions and positioning system accuracy. For example, in static environments with reliable GPS, S typically ranges from several tens of meters.

During mission execution, the UAV's flying altitude is typically constrained between predefined minimum and maximum thresholds, denoted as $h_{\min}$ and $h_{\max}$, respectively. These constraints arise from application-specific demands, such as ensuring adequate resolution and field of view for visual data acquisition in surveying or search tasks. The altitude cost associated with a waypoint P_{ij} is computed as follows:

$$H_{ij} = \begin{cases} \left| h_{ij} - \dfrac{(h_{\max} + h_{\min})}{2} \right|, & \text{if } h_{\min} \le h_{ij} \le h_{\max} \\ \infty, & \text{otherwise} \end{cases} \tag{3}$$

The altitude h_{ij} denotes the UAV's vertical distance relative to the ground, while H_{ij} is determined by Eq. (3), which ensures compliance with a set mean altitude while penalizing deviations beyond the acceptable range. As a result, the altitude cost function is expressed as:

$$F_3(X_i) = \sum_{j=1}^{n} H_{ij} \tag{4}$$

Evaluating the smoothing cost requires calculating both the turning angle and the climbing angle. As illustrated in Fig. 2 and mathematically defined in Eq. (6), the turning angle φ_{ij} is defined as the angle between the two projected path segments on the Oxy plane. Let $\vec{k}$ denote the unit vector along the z-axis. The projected vector can then be mathematically expressed as:

$$\overrightarrow{P'_{ij}P'_{i,j+1}} = \vec{k} \times (\overrightarrow{P_{ij}P_{i,j+1}} \times \vec{k}) \tag{5}$$

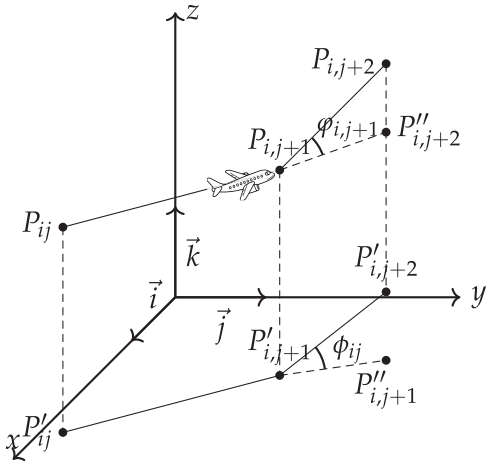


Figure 2: Turning and climbing angle calculation

Thus, the turning angle is determined by:

$$\varphi_{ij} = \arctan\left(\frac{\left\| \overrightarrow{P'_{ij}P'_{i,j+1}} \times \overrightarrow{P'_{i,j+1}P'_{i,j+2}} \right\|}{\overrightarrow{P'_{ij}P'_{i,j+1}} \cdot \overrightarrow{P'_{i,j+1}P'_{i,j+2}}} \right) \tag{6}$$

The climb angle, denoted as ψ_{ij}, is the angle between the ascent trajectory of the UAV and the horizontal plane. Specifically, the climb angle ψ_{ij} represents the angular deviation between the actual path segment $\overrightarrow{P_{ij}P_{i,j+1}}$ and its horizontal projection $\overrightarrow{P'_{ij}P'_{i,j+1}}$. The term $z_{i,j+1} - z_{ij}$ corresponds to the difference in altitude (vertical displacement) between two consecutive waypoints. Thus, the climb angle is calculated as:

$$\psi_{ij} = \arctan\left(\frac{z_{i,j+1} - z_{ij}}{\left\|\overrightarrow{P'_{ij}P'_{i,j+1}}\right\|}\right) \tag{7}$$

The smoothing cost is formulated as follows:

$$F_4(X_i) = a_1 \sum_{j=1}^{n-2} \varphi_{ij} + a_2 \sum_{j=1}^{n-1} \left|\psi_{ij} - \psi_{i,j-1}\right| \tag{8}$$

where a_1 and a_2 are the penalty coefficients for the turning angle and climbing angle, respectively.

2.3 Total Cost Function

Taking into account the optimality, safety, and feasibility constraints related to the path X_i, the overall cost function for the entire path X_i can be defined through Eqs. (1) to (8):

$$F(X_i) = \sum_{k=1}^{4} b_k F_k(X_i) \tag{9}$$

Choosing the most suitable function for UAV path planning is crucial. This study uses the optimal overall cost function in complex environments with multiple threats during UAV navigation as a suitable function. The coefficients b_k represent the weights assigned to each cost component.

3 Optimal Path

3.1 Particle Swarm Optimization Algorithm Based on Spherical Vector

The SPSO algorithm encodes each flight path as a sequence of vectors, where each vector represents the UAV's movement from one waypoint to the next [14]. These vectors are expressed in spherical coordinates and comprise three components: magnitude $\rho \in (0, \text{path length})$, elevation angle $\psi \in (-\frac{\pi}{2}, \frac{\pi}{2})$, and azimuth angle $\varphi \in (-\pi, \pi)$. Consequently, a flight path Ω_i consisting of n waypoints are represented as a spherical vector sequence with $N = n - 2$ elements:

$$\Omega_i = (\rho_{i1}, \psi_{i1}, \phi_{i1}, \rho_{i2}, \psi_{i2}, \phi_{i2}, \cdots, \rho_{iN}, \psi_{iN}, \phi_{iN}), N = n - 2 \tag{10}$$

The spherical vector $(\rho_{ij}, \psi_{ij}, \phi_{ij})$ is denoted as u_{ij}. For a swarm of M particles searching in an N-dimensional space (where N corresponds to the number of spherical vector components representing the path), the update equations of SPSO are defined as follows:

$$\Delta u_{ij}^{t+1} \leftarrow w^t \cdot \Delta u_{ij}^t + c_{\text{gpi}}(t) r_{1j}^t \left(q_{ij}^t - u_{ij}^t\right) + c_{\text{ggi}}(t) r_{2j}^t \left(q_{gj}^t - u_{ij}^t\right) \tag{11}$$

$$u_{ij}^{t+1} \leftarrow u_{ij}^t + \Delta u_{ij}^{t+1}, \ (i = 1, 2, \ldots, M; j = 1, 2, \ldots, N) \tag{12}$$

In these equations, $c_{\text{gpi}}(t)$ and $c_{\text{ggi}}(t)$ represent the cognitive and social acceleration coefficients at iteration t, respectively, whose adaptive design is detailed in Section 3.2. The terms r_{1j}^t and r_{2j}^t are uniform

random numbers within $[0,1]$ sampled for each dimension j and iteration t, introducing stochasticity to the search process. The local best and global best positions of particles are represented by the vector sets $Q_i = (q_{i1}, q_{i2}, \ldots, q_{iN})$ and $G_i = (g_{i1}, g_{i2}, \ldots, g_{iN})$, respectively. Determining Q_i and G_i requires constructing the vector flight path map Ω_i, which directly evaluates the relevant costs. The transformation from the spherical coordinate vector $u_{ij} = (\rho_{ij}, \psi_{ij}, \phi_{ij}) \in \Omega_i$ to the Cartesian waypoint $P_{ij} = (x_{ij}, y_{ij}, z_{ij}) \in X_i$ is performed as follows:

$$x_{ij} = x_{i,j-1} + \rho_{ij} \sin\psi_{ij} \cos\phi_{ij} \tag{13}$$

$$y_{ij} = y_{i,j-1} + \rho_{ij} \sin\psi_{ij} \sin\phi_{ij} \tag{14}$$

$$z_{ij} = z_{i,j-1} + \rho_{ij} \cos\psi_{ij} \tag{15}$$

Using spherical vectors in SPSO mainly relies on improvement in navigation safety through establishing mathematical correlations between the vector components (magnitude, elevation, and azimuth) and the UAV's kinematic parameters (velocity, turning angle, and climb angle). With this, particle-based solutions search through space-configuration instead of spaceCartesian, thereby increasing the possibility of finding good quality trajectories. Notably, the elevation and azimuth parameters inherently enforce constraints on steering and climb angles through their geometric definitions, substantially constraining the solution space.

3.2 Adaptive Weighting Strategy

Conventional UAV path planning employs metaheuristic algorithms with static parameters to strike a balance between exploration and exploitation. Fixed acceleration coefficients and inertia weights lack adaptability to nonlinear cost functions or dynamic threats. Similarly, standard SPSO uses static spherical vector encoding with constant social-cognitive coefficients, integrating UAV kinematics geometrically but missing the dynamic exploration-exploitation balance. This rigidity causes slow convergence and local optima trapping in complex environments due to unresponsiveness to changing solution landscapes. To address this limitation, a dual Sigmoid-based adaptive acceleration mechanism modulating cognitive and social coefficients in real-time via particle-best Euclidean distances and iteration phases. The proposed mechanism for dynamic search landscapes boosts optimization efficiency and solution quality.

The proposed mechanism adaptively adjusts the cognitive coefficient $c_{\mathrm{gpi}}(t)$ and the social coefficient $c_{\mathrm{ggi}}(t)$ for each particle i at iteration t. The adjustment is governed by the real-time Euclidean distances between the particle's current position $x_i(t)$, and its personal best position $p_i(t)$, as well as the global best position $g_i(t)$ found by the entire swarm so far. The coefficients are defined by the following Sigmoid functions, where b_1 and b_2 are constant scaling parameters that define the maximum possible values for the cognitive and social coefficients, respectively. They are determined empirically to set an upper bound for the acceleration influence. The terms $\|p_i(t) - x_i(t)\|$ and $\|g_i(t) - x_i(t)\|$ denote the Euclidean distances representing how far the particle is from its personal best solution and global best solution, respectively.

$$c_{\mathrm{gpi}}(t) = \frac{b_1}{1 + e^{-a_1(t)\cdot\|p_i(t) - x_i(t)\|}} \tag{16}$$

$$c_{\mathrm{ggi}}(t) = \frac{b_2}{1 + e^{-a_2(t)\cdot\|g_i(t) - x_i(t)\|}} \tag{17}$$

The parameters $a_1(t)$ and $a_2(t)$ are time-varying steepness parameters that control the sensitivity of the Sigmoid functions to these distances. Their values evolve throughout the optimization process to enforce distinct behaviors during different search phases. The parameter $a_1(t)$, associated with the cognitive component, increases linearly with the iteration count t. This progressive increase amplifies the sensitivity

of $c_{\mathrm{gpi}}(t)$ to the distance $\|p_i(t) - x_i(t)\|$ during the later stages of optimization. As a result, particles are increasingly encouraged to refine their search around their personal best positions, enhancing local exploitation. Conversely, $a_2(t)$, governing the social component, decreases linearly with t. This reduction boosts the responsiveness of $c_{\mathrm{ggi}}(t)$ to the global distance $\|g_i(t) - x_i(t)\|$ primarily during the early phases, promoting robust global exploration by drawing particles more strongly towards the swarm's best-found region. The opposing evolutionary trends of $a_1(t)$ and $a_2(t)$ ensure a smooth and automatic transition from a broad, exploratory search to a focused, exploitative refinement. Notably, the sum $a_1(t) + a_2(t) = 3a_{\mathrm{initial}}$ remains constant throughout the process, ensuring computational balance.

$$a_1(t) = a_{\mathrm{initial}} \cdot \left(1 + \frac{t}{t_{\max}}\right) \tag{18}$$

$$a_2(t) = a_{\mathrm{initial}} \cdot \left(2 - \frac{t}{t_{\max}}\right) \tag{19}$$

The inherent monotonic increasing property of the Sigmoid function is central to the mechanism's behavior. The acceleration coefficients increase as the corresponding distances increase. When a particle is far from a the best position, the argument of the Sigmoid becomes a large positive number, causing $c_{\mathrm{gpi}}(t)$ and $c_{\mathrm{ggi}}(t)$ to approach their maximum values b_1 and b_2. This generates strong acceleration forces, compelling the particle to move rapidly towards the promising best position to bridge the large gap. Conversely, when a particle is already close to a best position, the argument of the Sigmoid becomes small, causing the coefficients to diminish towards lower values. This allows for gentle, precise adjustments in the vicinity of the best position, preventing overshooting and facilitating fine-tuned local search. This intelligent design aligns the magnitude of velocity updates with the immediate optimization need: aggressive movement towards distant promising areas and cautious refinement within nearby regions. The intelligent design ensures vigorous exploration of vast search spaces while maintaining stable exploitation in promising regions without disrupting the convergence process.

By incorporating this dynamic state awareness and phase-dependent tuning into the acceleration coefficients, the proposed mechanism effectively addresses the rigidity of static PSO parameter configurations. The proposed mechanism offers a more adaptive, efficient, and intelligent approach to balancing exploration and exploitation in complex UAV path planning problems.

3.3 Lens-Based Opposition Learning Strategy

To overcome the limitations of the SPSO algorithm in balancing global exploration and local exploitation during UAV path planning, which often leads to premature convergence and slow optimization, this paper incorporates a lens imaging-based opposition learning strategy [23]. This method enhances the algorithm's ability to explore uncharted regions of the search space while simultaneously refining solutions in promising areas, thereby improving both population diversity and convergence speed. The core idea is inspired by the conjugate relationship between an object and its image in geometric optics, where a convex lens generates a symmetrical image of an object. By analogizing the current solution (particle position) to an object, its opposition solution (mirror point) can be dynamically generated within the solution space, facilitating a bidirectional search mechanism.

Mathematically, for a particle located at position x within the current search boundaries $[a, b]$ in a given dimension, its lens-based opposition solution x^* is calculated using the lens imaging formula:

$$u_{ij}^* = \frac{a+b}{2} + \frac{a+b}{2k} - \frac{u_{ij}}{k} \tag{20}$$

Here, $k > 0$ is a scaling factor that controls the degree of opposition. This factor is dynamically adjusted throughout the iterations to balance exploration and exploitation. Initially, a larger k value pushes the opposition solution closer to the center of the search space, encouraging exploration of broader regions. As the optimization progresses, k is linearly decreased to a smaller value, which draws the opposition solution nearer to the current solution, thus promoting local refinement around promising areas. The dynamic update of k is governed by:

$$k_t = k_{\max} - \frac{k_{\max} - k_{\min}}{t_{\max}} \cdot t \tag{21}$$

where t is the current iteration and $t_{\max}$ is the maximum number of iterations. Furthermore, the search boundaries $[a, b]$ for each dimension are not fixed but are dynamically contracted based on the distribution of the current population, focusing the search on increasingly promising regions. The process of generating an opposition solution via this strategy is conceptually illustrated in Fig. 3, which depicts the optical analogy of an object at x forming an image at x^* through a lens placed at the midpoint $o = (a + b)/2$.

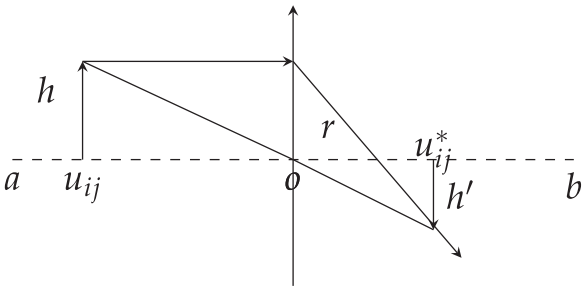


Figure 3: Oppositional learning based on lens imaging principle

The integration of this lens-based opposition learning into the SPSO framework occurs after the standard velocity and position update. For each particle, an opposition solution is generated in the spherical coordinate space using Eq. (20). The fitness of this new solution is evaluated and compared against the original particle's fitness. If the opposition solution yields a better fitness value, the opposition solution replaces the current particle in the swarm. This mechanism injects new, high-quality solutions into the population, enhancing diversity and providing an effective means to escape local optima. The strategy systematically improves the algorithm's performance by leveraging the inherent duality and symmetry of optical systems, resulting in a more robust and efficient balance between exploration and exploitation for complex UAV path planning problems.

3.4 Application of DW-SPSO in UAV Track Planning

The implementation of the DW-SPSO algorithm in the trajectory planning of UAVs incorporates spherical vector encoding, dynamic weight adjustment, and a lens-based opposition learning strategy, thereby establishing a comprehensive optimization framework. The algorithm's core task is to balance global exploration and local exploitation, enhancing solution diversity through concepts from geometric optics theory. This is an attempt to create relatively safe and smooth flight paths in dangerous threat environments. Flowchart of DW-SPSO, as delineated in Algorithm 1, shares a structural resemblance with other PSO algorithms, including parameter initialization, particle generation, and swarm evolution. Nevertheless, DW-SPSO primarily distinguishes itself from other PSO algorithms through the representation of particle positions and velocities, as well as the update equations. This unique design enables DW-SPSO to address

the path planning issue for UAVs in complex environments more effectively, ensuring their safe and efficient task execution.

Algorithm 1: DW-SPSO for UAV path planning

```
/* Initialization: */
Get search map and path planning information;
Set parameters: a_initial, b1, b2, k_max, k_min, t_max;
for each particle i in Popsize do
    Initialize Ω_i, u_ij; compute F(X_i);
end for
/* Optimization Loop: */
for gen ← 1 to t_max do
    Update sigmoid parameters a1, a2;                          /* Eqs. (18), (19)*/
    for each particle i do
        Compute individual learning factor c_ppi and social learning factor c_ggi;   /* Eqs. (16), (17)*/
        Compute velocity update;
        Update position Ω_i;
        Compute fitness F(X_i);                                /* Eq. (9)*/
    end for
    Update global best g_ij;
    for each particle i do
        Compute backward solution x*;                          /* Eq. (20) */
        Compute fitness F_new(X_i);
        if F_new(X_i) < F(X_i) then
            Replace Ω_i with x*;
        end if
    end for
end for
return Ω_i;
```

4 Experimental Simulation and Analysis

To examine the performance of the enhanced PSO algorithm, a series of computational simulations were conducted for comparative analysis and experimental verification.

4.1 Evaluation Using DEM Maps

The evaluation scenarios are based on the public 5 metre DEM of Australia, derived from a LiDAR dataset, openly provided by Australia [24]. This dataset is available under a Creative Commons Attribution 4.0 International Licence from the official portal (https://elevation.fsdf.org.au/). Two distinct terrain patches on Christmas Island (approximate extent: 105.53°E to 105.65°E, 10.42°S to 10.55°S) were selected and extended to create two benchmarking scenarios: the ChristmasTerrain model and the TerrainData model. The number and placement of threats, represented by green cylinders, vary in complexity between these scenarios. Table 1 displays the environmental parameters for the simple and complex obstacle configurations within the ChristmasTerrain model, while Table 2 provides the corresponding parameters for the TerrainData model.

Table 1: ChrismasTerrain obstacle model simulation parameters

Scenarios1			Scenarios2		
Name	**Center position**	**Radius**	**Name**	**Center position**	**Radius**
Start	(200, 100, 150)	–	Start	(200, 100, 150)	–
End	(800, 800, 150)	–	End	(800, 800, 150)	–
Obstacle 1	(400, 500, 100)	80	Obstacle 1	(500, 150, 140)	70
Obstacle 2	(400, 300, 120)	90	Obstacle 2	(650, 600, 120)	60
Obstacle 3	(500, 150, 140)	70	Obstacle 3	(700, 550, 150)	70
			Obstacle 4	(300, 750, 100)	80
			Obstacle 5	(600, 750, 50)	80
			Obstacle 6	(300, 500, 150)	60
			Obstacle 7	(700, 300, 100)	60

Table 2: TerrainData obstacle model simulation parameters

Scenarios3			Scenarios4		
Name	**Center position**	**Radius**	**Name**	**Center position**	**Radius**
Start	(10, 10, 200)	–	Start	(10, 10, 200)	–
End	(400, 400, 150)	–	End	(400, 400, 150)	–
Obstacle 1	(100, 300, 100)	30	Obstacle 1	(300, 300, 100)	30
Obstacle 2	(300, 300, 100)	20	Obstacle 2	(200, 100, 100)	20
Obstacle 3	(100, 50, 100)	30	Obstacle 3	(100, 200, 100)	30
			Obstacle 4	(300, 100, 100)	20
			Obstacle 5	(200, 50, 100)	20
			Obstacle 6	(150, 350, 100)	30
			Obstacle 7	(180, 300, 150)	80

4.2 Parameter Setting

The proposed DW-SPSO is compared with standard PSO [25] and some state-of-the-art PSO variant algorithms, including APSO [26], AWPSO [27], DSPSO [28], SPSO [14], VPPSO [29], PPSOBAS [30] and TS-CEPSO [31]. APSO adapts parameters based on the population state. AWPSO employs an S-type function-based weighting strategy. DSPSO incorporates dynamic differential mutation. SPSO utilizes spherical coordinates for path encoding. VPPSO uses a "velocity pausing" mechanism to maintain diversity. PPSOBAS hybridizes PSO with the Beetle Antennae Search algorithm. TS-CEPSO integrates chaotic maps and feasibility rules for complex constraints.

To ensure experimental fairness and comparability, all algorithms were configured under consistent conditions: a population size of 100 and a maximum of 100 iterations. The parameter settings for all compared algorithms were meticulously adopted from their respective original publications or authoritative implementations to ensure a faithful and unbiased comparison. The complete parameter configurations are summarized in Table 3.

Table 3: Parameter settings for all compared algorithms

Algorithm	Source	Parameter settings
PSO	[25]	$c_1 = 1.49445$, $c_2 = 1.49445$, $V_{\max} = 5$, $V_{\min} = -5$
APSO	[26]	Adaptive parameters: $W \in \{0.4, 0.9\}$, $c_1 \in \{1.0, 1.5, 2.0\}$, $c_2 \in \{1.0, 1.5, 2.0\}$
AWPSO	[27]	$w_{\max} = 0.9$, $w_{\min} = 0.4$, $c_1^{\text{initial}} = 2.5$, $c_2^{\text{initial}} = 0.5$, $c_1^{\text{final}} = 0.5$, $c_2^{\text{final}} = 2.5$
DSPSO	[28]	$c_1 = 2.0$, $w = 1 - (1 - 0.7) \cdot \frac{t}{T_{\max}}$, Mutation probability $P_m = 0.1$
SPSO	[14]	$w = 0.73$, $c_1 = 1.5$, $c_2 = 1.5$, $V_{\max} = 5$, $V_{\min} = -5$
VPPSO	[29]	$w(t) = \exp\left(-(2.5t/T_{\max})^{2.5}\right)$, $c_1 = 1.5$, $c_2 = 1.5$
PPSOBAS	[30]	PSO: $c_1 = 1.49445$, $c_2 = 1.49445$; BAS: $d_0 = 0.5$, $\delta = 0.1$; Tent chaotic map with $a = 0.499$
TS-CEPSO	[31]	$w_{\max} = 0.9$, $w_{\min} = 0.2$, $C_{\max} = 1.5$, $C_{\min} = 0.2$, $c_1 = C$, $c_2 = 2.5 - c_1$, $r_1 = 0.129$, $r_2 = 0.871$
DW-SPSO (Proposed)	This work	$a_{\text{initial}} = 3.1623 \times 10^{-5}$, $b_1 = 0.65$, $b_2 = 0.475$, $w_{\max} = 0.9$, $w_{\min} = 0.4$

For the proposed DW-SPSO algorithm, the key parameters of the dual Sigmoid mechanism (a_{initial}, b_1, b_2) were rigorously optimized. An initial sensitivity analysis of over 200 candidate parameter sets identified these five parameters as the most influential. Bayesian optimization was then employed to minimize the average fitness (Eq. 9) within the defined search spaces ($a_{\text{initial}} \in [10^{-6}, 10^{-3}]$; $b_1, b_2 \in [0.1, 2.0]$), converging to the optimal combination listed in Table 3. The algorithm demonstrated robustness to ±10% variations in these key parameters, with performance degradation remaining below 5%, confirming the stability of the selected configuration.

A paired sample t-test [32] with a significance level of $\alpha = 0.05$ (95% confidence) is used to assess the statistical significance of the performance differences between DW-SPSO and all other algorithms. The notation D^+ indicates that the performance of DW-SPSO is statistically better than the compared algorithm, D^- indicates the DW-SPSO performance is statistically worse, and N indicates no statistically significant difference (Not Applicable). NA means Not Applicable.

4.3 Comparison between PSO Algorithms

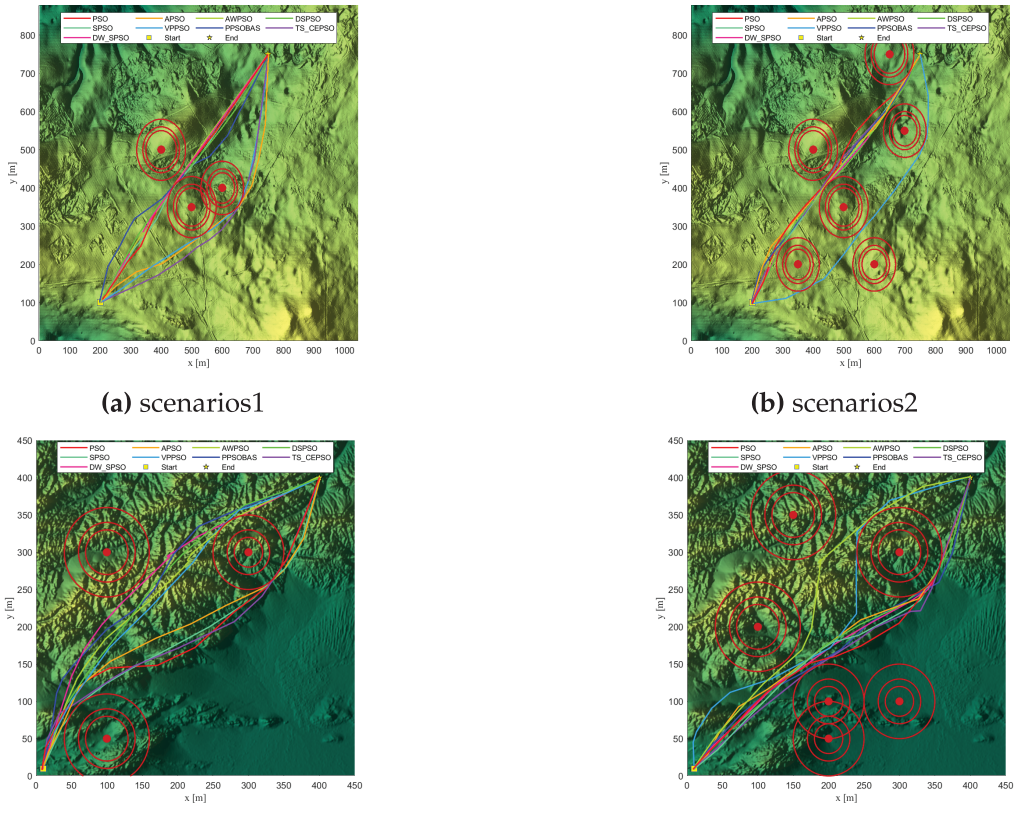
To comprehensively assess the performance of the proposed DW-SPSO algorithm. The experiments were performed on two distinct terrain models (ChristmasTerrain and TerrainData) with varying obstacle complexities. They examined the result in terms of the quality of the path, whether it converged or not, as well as in terms of significance.

To further quantify the path quality, a detailed analysis was conducted on three critical metrics derived from Eqs. (6)–(8): climb angle, safe distance, and smoothness score. As shown in Table 4, DW-SPSO achieves the best overall performance with a balanced combination of these metrics. Specifically, the algorithm maintains a moderate climb angle of 5.30°, which is significantly lower than most competitors (e.g., APSO: 8.90°, SPSO: 8.49°), indicating more stable altitude transitions. In terms of safety, DW-SPSO maintains a safe distance of 4.98 m, comparable to the best-performing algorithms, while ensuring collision avoidance. Most notably, DW-SPSO achieves the highest smoothness score of 85.02, demonstrating its superior ability to generate paths with minimal abrupt turns and altitude changes. This balanced performance across all three metrics validates the effectiveness of the spherical vector encoding and dynamic weight adjustment in producing practically feasible UAV paths.

Table 4: Quantitative comparison of path segment quality

Algorithm	Climb angle (°)	Safe distance (m)	Smoothness score
PSO [25]	6.81	4.73	81.73
APSO [26]	8.90	5.10	79.71
AWPSO [27]	8.19	4.60	84.02
DSPSO [28]	4.85	4.81	83.70
SPSO [14]	8.49	5.03	84.49
VPPSO [29]	8.23	4.91	77.12
PPSOBAS [30]	8.79	4.96	79.99
TS-CEPSO [31]	7.55	4.97	79.58
DW-SPSO (Proposed)	5.30	4.98	85.02

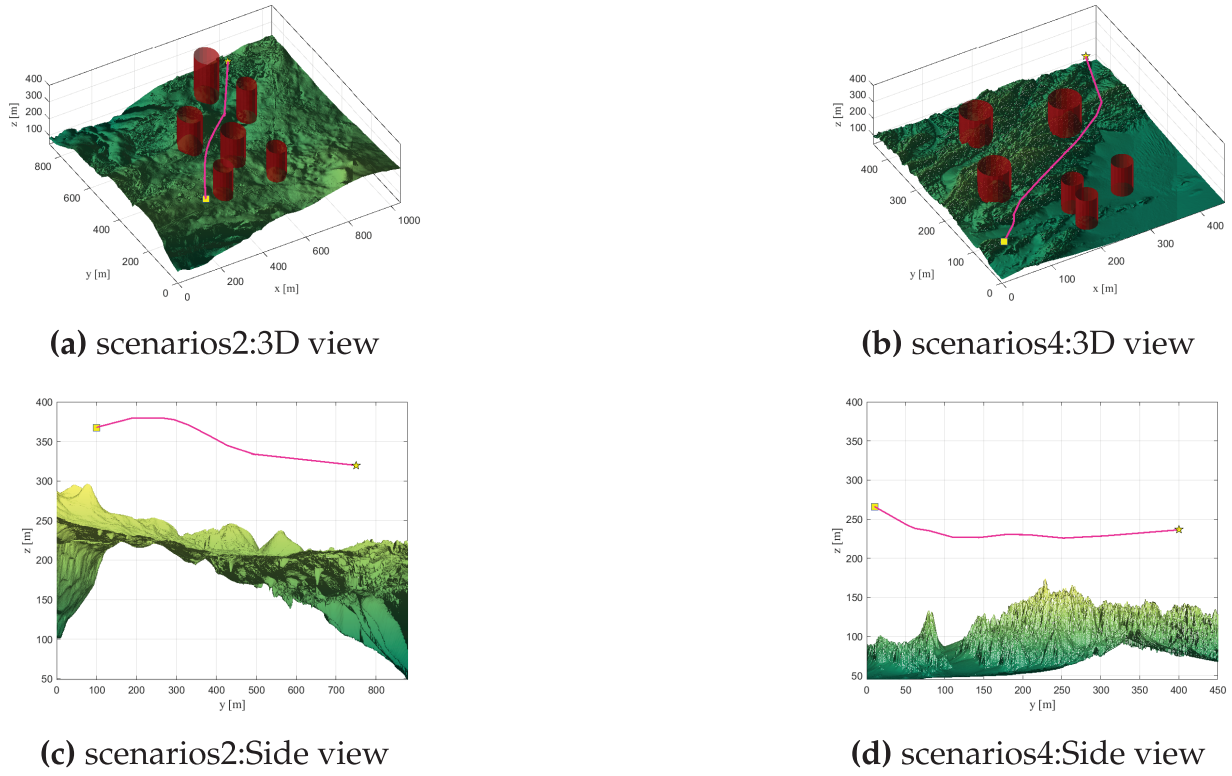
Fig. 4 shows the top-down views of the paths taken by the compared PSO algorithms for the Christmas Terrain model and TerrainData model, respectively, All algorithms successfully generated feasible paths that satisfied constraints like path length, obstacle avoidance, and smoothness. Notably, the paths produced by DW-SPSO exhibit superior adaptability to complex environments, with smoother transitions and fewer sharp turns, as highlighted in Fig. 5. This demonstrates the effectiveness of the dynamic weight adjustment mechanism and lens-based opposition learning in enhancing path quality.



(a) scenarios1 **(b)** scenarios2

(c) scenarios3 **(d)** scenarios4

Figure 4: Top view of PSO paths on model for scenarios 1 and 4



(a) scenarios2:3D view

(b) scenarios4:3D view

(c) scenarios2:Side view

(d) scenarios4:Side view

Figure 5: The planned paths generated by DW-SPSO for scenarios 2 and 4

DW-SPSO's computational efficiency was evaluated against standard PSO and SPSO. All three algorithms share theoretical time complexity $O(M \cdot N \cdot T)$ for population size M, dimensionality N, and iterations T. Empirically, DW-SPSO incurs 15%–20% extra time per iteration due to dual Sigmoid adaptation and lens opposition learning. This overhead is justified by superior convergence (Fig. 6), achieving better solutions in fewer iterations. Scalability tests (50–500 particles) confirm linear scaling, ensuring practical UAV applications. The algorithm's parallel fitness evaluation suits hardware acceleration on embedded GPUs (e.g., NVIDIA Jetson), where convergence gains offset iteration overhead for real-time replanning.

Fig. 6 shows the convergence trend of the best fitness values with respect to iteration. DW-SPSO is always faster and achieves better fitness values compared to other algorithms, indicating its rapid convergence and robust optimization capability. The dual Sigmoid-based adaptive acceleration mechanism enables DW-SPSO to balance exploration and exploitation effectively, avoiding premature convergence to local optima.

Table 5 presents a summary of the average fitness values. All variants converge similarly, with the exception of VPPSO, due to its velocity pausing mechanism. This algorithm deliberately halts velocity updates when diversity metrics fall below certain thresholds, resulting in periodic plateaus. Additionally, standard deviations and paired t-test results are provided for the ChristmasTerrain and TerrainData models, respectively. DW-SPSO demonstrates superior performance compared to most algorithms, as indicated by its lower average fitness values and smaller standard deviations, which suggest higher solution quality and stability. The t-test results (denoted as D+) confirm that the performance improvements of DW-SPSO are statistically significant in most scenarios.

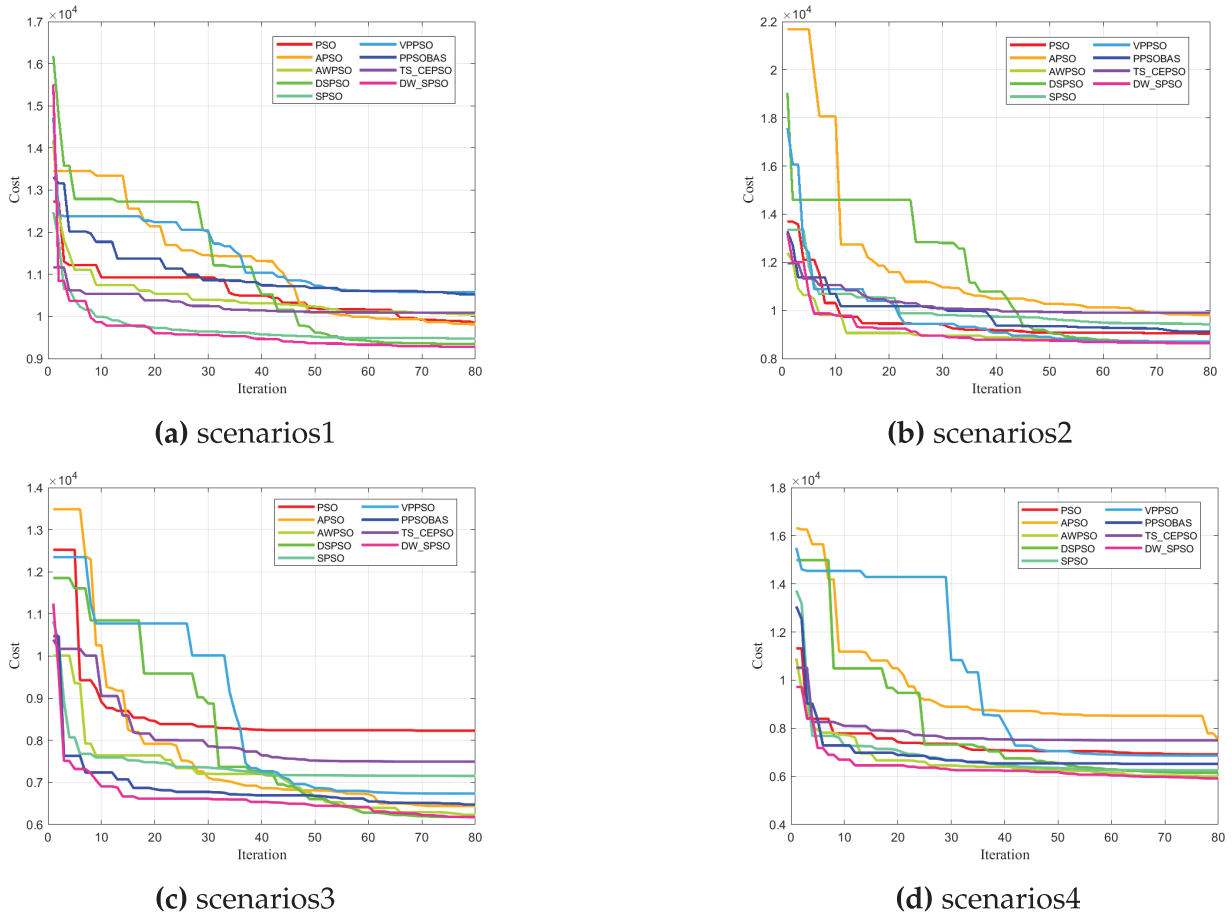


Figure 6: Best fitness of DW-SPSO and metaheuristic algorithms on the model

Table 5: Combined algorithm metrics comparison

Algorithm	Scenarios1			Scenarios2			Scenarios3			Scenarios4		
	Avg	Std	*t*-test	Avg	Std	*t*-test	Avg	Std	*t*-test	Avg	Std	*t*-test
PSO [25]	8722	1304	D+	8968	1182	D+	10787	897	D+	11402.49	891	D+
APSO [26]	8517	1950	D+	8249	2358	D+	11596	1675	D+	12527	2656	D+
AWPSO [27]	6757	1001	D-	6674	687	D-	9783	674	D+	9949	1147	D+
DSPSO [28]	8048	1996	D+	8010	2215	D+	10987	1606	D+	12498	2804	D+
SPSO [14]	7253	803	N	7175	1105	N	9913	605	D+	10240	1174	D+
VPPSO [29]	8396	2345	D+	8211	1948	D+	10458	1716	D+	11086	1051	D+
PPSOBAS [30]	7639	741	D+	7916	1132	D+	10786	770	D+	11192	1058	N
TS-CEPSO [31]	7202	578	D+	7446	489	D+	9652	147	D+	9996	153	D+
DW-SPSO (Proposed)	6709	964	D+	6740	886	D+	9684	592	D+	9941	785	D+

The Wilcoxon signed-rank test [33] results unequivocally demonstrate the superiority of the proposed DW-SPSO algorithm. As shown in Table 6, DW-SPSO achieves statistically significant improvements

($p < 0.05$) over all other state-of-the-art algorithms, including the baseline SPSO ($p = 0.03485$). The p-values for comparisons with algorithms like APSO, AWPSO, and TS-CEPSO are exceedingly small ($p < 1e - 50$), providing overwhelming evidence that the performance differences are not due to chance. This comprehensive statistical superiority validates the effectiveness of the dynamic weight adjustment and lens-based opposition learning strategies introduced in this work.

Table 6: Wilcoxon signed-rank test p-values

Algorithm	PSO	APSO	AWPSO	DSPSO	SPSO	VPPSO	PPSOBAS	TS-CEPSO	DW-SPSO
PSO	—	2.16e-68	9.82e-211	0.0116	1.17e-173	3.10e-27	6.18e-07	6.63e-219	2.81e-187
APSO	2.16e-68	—	4.83e-245	3.50e-42	1.25e-184	1.05e-11	2.50e-47	2.96e-246	3.42e-186
AWPSO	9.82e-211	4.83e-245	—	3.81e-175	2.60e-71	1.25e-180	4.24e-211	0.00020	8.33e-60
DSPSO	0.0116	3.50e-42	3.81e-175	—	1.07e-76	1.84e-25	0.06012	3.03e-155	1.60e-85
SPSO	1.17e-173	1.25e-184	2.60e-71	1.07e-76	—	9.13e-141	4.59e-211	1.05e-57	0.03485
VPPSO	3.10e-27	1.05e-11	1.25e-180	1.84e-25	9.13e-141	—	1.77e-14	8.10e-190	2.13e-135
PPSOBAS	6.18e-07	2.50e-47	4.24e-211	0.06012	4.59e-211	1.77e-14	—	5.54e-249	2.54e-200
TS-CEPSO	6.63e-219	2.96e-246	0.00020	3.03e-155	1.05e-57	8.10e-190	5.54e-249	—	2.65e-50
DW-SPSO	2.81e-187	3.42e-186	8.33e-60	1.60e-85	0.03485	2.13e-135	2.54e-200	2.65e-50	—

To provide a more robust statistical verification that is insensitive to non-normal data distributions, the Wilcoxon signed-rank test was further conducted the Wilcoxon signed-rank test on the results from 30 independent runs. The detailed p-values are comprehensively summarized in Table 6. The analysis reveals that DW-SPSO achieves statistically significant improvements ($p < 0.05$) over all other state-of-the-art algorithms across the majority of scenarios. For instance, when compared to the baseline SPSO, the superiority of DW-SPSO is statistically significant ($p = 0.03485$). Notably, the p-values for comparisons against algorithms such as APSO, AWPSO, and TS-CEPSO are exceedingly small ($p < 1e - 50$), providing overwhelming evidence that the observed performance enhancements are not attributable to random chance. This rigorous non-parametric analysis, coupled with the earlier t-test results, offers comprehensive and robust statistical evidence that firmly validates the effectiveness of the proposed dynamic weight adjustment mechanism and lens-based opposition learning strategy.

4.4 Ablation and Sensitivity Testing

To thoroughly analyze the contributions of individual components in the DW-SPSO algorithm and evaluate parameter sensitivity, systematic ablation experiments and sensitivity tests. The tests were performed in a simplified 150 × 150 environment with 6 waypoints, resulting in an 18-dimensional search space.

Ablation studies were conducted by systematically disabling key components of the proposed algorithm. As shown in Table 7 (with the best-performing configuration highlighted in bold), the complete DW-SPSO achieved the best performance with an average fitness of 2520.86. Removing the dual Sigmoid weighting mechanism resulted in a 13.9% performance degradation (fitness: 2871.76), demonstrating its crucial role in balancing exploration and exploitation. Disabling the lens opposition learning caused a 1.9% performance drop (fitness: 2569.83), confirming its contribution to solution diversity. The baseline PSO without any enhancements performed worse (fitness: 3296.40), highlighting the collective importance of both proposed mechanisms.

Table 7: Ablation study results

Configuration	Average fitness	Std	Relative degradation
Complete DW-SPSO	**2520.86**	2.23	–
w/o sigmoid weighting	2871.76	527.94	13.9%
w/o lens learning	2569.83	157.58	1.9%
Baseline PSO	3296.40	495.12	23.5%

Parameter sensitivity analysis was extended beyond the typical ±10% range to assess robustness under more extreme conditions (±60% variation). The social coefficient scaling factor b_2 proved most sensitive, with performance deviations up to 14.1%. The initial steepness parameter a_{initial} and distance scaling factor d_2 showed moderate sensitivity (5.7% deviation), while b_1 and d_1 demonstrated better stability with deviations below 6%. Notably, the algorithm maintained reasonable performance even under these extreme parameter variations, confirming its robustness for practical applications.

4.5 Hardware Implementation Considerations

To address the practical deployment of the proposed DW-SPSO algorithm, this section discusses its computational characteristics and potential integration with UAV hardware platforms. The algorithm's inherent parallelism in fitness evaluation makes it suitable for hardware acceleration. Recent studies have demonstrated that the successful implementation of PSO variants on embedded systems is commonly used in UAV autopilots. For instance, El-Metwally et al. implemented a smart decision-making framework on NVIDIA Jetson platforms, showing real-time capability for autonomous navigation tasks. Similarly, research by Alhusseini et al. [20] showcased adaptive PSO optimization on embedded GPU architectures, achieving significant speedup for complex optimization problems.

While current simulations validate DW-SPSO for offline planning, its computational structure shows strong real-time replanning potential. Spherical vector encoding reduces dimensionality vs. Cartesian representations, and adaptive mechanisms minimize redundant computations. Future work will implement DW-SPSO on embedded processors (NVIDIA Jetson Orin, Qualcomm Snapdragon Ride), leveraging parallel computing for dynamic environment performance.

5 Conclusion

This paper proposed the DW-SPSO algorithm to address key challenges in UAV path planning within complex environments. By integrating a dual Sigmoid-based adaptive weight adjustment mechanism and a lens opposition learning strategy, the algorithm significantly enhances the balance between global exploration and local exploitation. Simulation experiments demonstrate that DW-SPSO outperforms existing PSO variants in terms of path safety, smoothness, and convergence speed in complex threat scenarios. Specifically, DW-SPSO achieved an average fitness value of 6709 in Scenario 1 and 9941 in Scenario 4, representing improvements of up to 7.5% and 2.8%, respectively, compared to the baseline SPSO algorithm.

While the algorithm shows clear advantages in complex obstacle environments, there remains potential for optimizing performance balance and computational efficiency in simpler settings. Future work will focus on developing a multi-objective optimization framework that simultaneously optimizes metrics such as path length, energy consumption, and risk. Furthermore, research will expand to include dynamic obstacles and collaborative multi-UAV environments. Theoretical investigations using complex network analysis will also be pursued to better understand the swarm dynamics and convergence behavior of the proposed algorithm.

Acknowledgement: Not applicable.

Funding Statement: This work was supported by the National Natural Science Foundation of China (Grant No. 62106092), the Natural Science Foundation of Fujian Province (Grant Nos. 2024J01822, 2025J01981), and the Natural Science Foundation of Zhangzhou City (Grant No. ZZ2024J28).

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Rui Yao, Yuye Wang, Fei Yu; algorithm implementation and data curation: Rui Yao; analysis and interpretation of results: Rui Yao, Yuye Wang, Fei Yu; draft manuscript preparation: Rui Yao; review, editing, and supervision: Yuye Wang, Fei Yu, Hongrun Wu, Zhenya Diao. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The digital elevation model (DEM) data used in this study are publicly available from Geoscience Australia at https://elevation.fsdf.org.au/.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Chan Y, Ng KK, Lee C, Hsu LT, Keung K. Wind dynamic and energy-efficiency path planning for unmanned aerial vehicles in the lower-level airspace and urban air mobility context. Sustain Energy Technol Assess. 2023;57(9):103202. doi:10.1016/j.seta.2023.103202.
2. Gul F, Mir I, Abualigah L, Sumari P, Forestiero A. A consolidated review of path planning and optimization techniques: technical perspectives and future directions. Electronics. 2021;10(18):2250. doi:10.3390/electronics10182250.
3. Maboudi M, Homaei M, Song S, Malihi S, Saadatseresht M, Gerke M. A review on viewpoints and path planning for UAV-based 3-D reconstruction. IEEE J Sel Top Appl Earth Obs Remote Sens. 2023;16(8):5026–48. doi:10.1109/jstars.2023.3276427.
4. Penin B, Giordano PR, Chaumette F. Minimum-time trajectory planning under intermittent measurements. IEEE Robot Autom Lett. 2018;4(1):153–60. doi:10.1109/lra.2018.2883375.
5. Pharpatara P, Hérissé B, Bestaoui Y. 3-D trajectory planning of aerial vehicles using RRT. IEEE Trans Control Syst Technol. 2016;25(3):1116–23. doi:10.1109/tcst.2016.2582144.
6. Li Y, Wei W, Gao Y, Wang D, Fan Z. PQ-RRT*: an improved path planning algorithm for mobile robots. Expert Syst Appl. 2020;152(2):113425. doi:10.1016/j.eswa.2020.113425.
7. Sonmez A, Kocyigit E, Kugu E. Optimal path planning for UAVs using genetic algorithm. In: 2015 International Conference on Unmanned Aircraft Systems (ICUAS). Piscataway, NJ, USA: IEEE; 2015. p. 50–5. doi:10.1109/ICUAS.2015.7152274.
8. Chai X, Zheng Z, Xiao J, Yan L, Qu B, Wen P, et al. Multi-strategy fusion differential evolution algorithm for UAV path planning in complex environment. Aerosp Sci Technol. 2022;121(12):107287. doi:10.1016/j.ast.2021.107287.
9. Perez-Carabaza S, Besada-Portas E, Lopez-Orozco JA, De La Cruz JM. Ant colony optimization for multi-UAV minimum time search in uncertain domains. Appl Soft Comput. 2018;62(4):789–806. doi:10.1016/j.asoc.2017.09.009.
10. Xu C, Duan H, Liu F. Chaotic artificial bee colony approach to uninhabited combat air vehicle (UCAV) path planning. Aerosp Sci Technol. 2010;14(8):535–41. doi:10.1016/j.ast.2010.04.008.
11. Karthik K, Balasubramanian C, Praveen R. Hybrid golden Jackal and moth flame optimization algorithm based coverage path planning in heterogeneous UAV networks. Sci Rep. 2025;15(1):31054. doi:10.1038/s41598-025-15345-6.
12. Zhang Q, Barat A, Li N, Zhao R, Guo Q. Improved butterfly optimization algorithm based power cable UAV trajectory optimization algorithm. In: 2024 IEEE 7th Information Technology, Networking, Electronic and Automation

Control Conference (ITNEC). Piscataway, NJ, USA: IEEE; 2024. Vol. 7, p. 1106–11. doi:10.1109/itnec60942.2024.10733293.
13. Deng L, Liu S. Advancing photovoltaic system design: an enhanced social learning swarm optimizer with guaranteed stability. Comput Ind. 2025;164(4):104209. doi:10.1016/j.compind.2024.104209.
14. Phung MD, Ha QP. Safety-enhanced UAV path planning with spherical vector-based particle swarm optimization. Appl Soft Comput. 2021;107(2):107376. doi:10.1016/j.asoc.2021.107376.
15. Yu Z, Si Z, Li X, Wang D, Song H. A novel hybrid particle swarm optimization algorithm for path planning of UAVs. IEEE Internet Things J. 2022;9(22):22547–58. doi:10.1109/jiot.2022.3182798.
16. Sonny A, Yeduri SR, Cenkeramaddi LR. Autonomous UAV path planning using modified PSO for UAV-assisted wireless networks. IEEE Access. 2023;11:70353–67. doi:10.1109/access.2023.3293203.
17. Meng Q, Chen K, Qu Q. PPSwarm: multi-UAV path planning based on hybrid PSO in complex scenarios. Drones. 2024;8(5):192. doi:10.3390/drones8050192.
18. Deng L, Liu S. Collective dynamics of particle swarm optimization: a network science perspective. Physica A. 2025;675(6):130778. doi:10.1016/j.physa.2025.130778.
19. Liu Y, Zhang H, Zheng H, Li Q, Tian Q. A spherical vector-based adaptive evolutionary particle swarm optimization for UAV path planning under threat conditions. Sci Rep. 2025;15(1):2116. doi:10.1038/s41598-025-85912-4.
20. Alhusseini H, Abdali LM, Issa HA, Velkin VI. Adaptive particle swarm optimization based model predictive control MPPT algorithm for PV systems under partial shading conditions. Results Eng. 2025;28(9):107419. doi:10.2139/ssrn.5353112.
21. Tan Y, Ding K. A survey on GPU-based implementation of swarm intelligence algorithms. IEEE Trans Cybern. 2015;46(9):2028–41. doi:10.1109/tcyb.2015.2460261.
22. Phung MD, Ha QP. Motion-encoded particle swarm optimization for moving target search using UAVs. Appl Soft Comput. 2020;97:106705. doi:10.36227/techrxiv.13333298.
23. Yu F, Guan J, Wu H, Chen Y, Xia X. Lens imaging opposition-based learning for differential evolution with cauchy perturbation. Appl Soft Comput. 2024;152(1):111211. doi:10.1016/j.asoc.2023.111211.
24. Australia G. Digital elevation model (DEM) of Australia derived from LiDAR 5 metre grid. Canberra, ACT, Australia: Geoscience Australia; 2015.
25. Roberge V, Tarbouchi M, Labonté G. Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning. IEEE Trans Ind Inf. 2012;9(1):132–41. doi:10.1109/tii.2012.2198665.
26. Zhan ZH, Zhang J, Li Y, Chung HSH. Adaptive particle swarm optimization. IEEE Trans Syst Man Cybern B. 2009;39(6):1362–81. doi:10.1109/tsmcb.2009.2015956.
27. Liu W, Wang Z, Yuan Y, Zeng N, Hone K, Liu X. A novel sigmoid-function-based adaptive weighted particle swarm optimizer. IEEE Trans Cybern. 2019;51(2):1085–93. doi:10.1109/tcyb.2019.2925015.
28. Zhang X, Wang X, Kang Q, Cheng J. Differential mutation and novel social learning particle swarm optimization algorithm. Inf Sci. 2019;480(24):109–29. doi:10.1016/j.ins.2018.12.030.
29. Shami TM, Mirjalili S, Al-Eryani Y, Daoudi K, Izadi S, Abualigah L. Velocity pausing particle swarm optimization: a novel variant for global optimization. Neural Comput Appl. 2023;35(12):9193–223. doi:10.1007/s00521-022-08179-0.
30. Wang L, Wu D, Gao Y. A reinforced particle swarm optimization algorithm based on beetle antenna search for UAV path planning. In: 2023 42nd Chinese Control Conference (CCC). Piscataway, NJ, USA: IEEE; 2023. p. 4455–60. doi:10.23919/ccc58697.2023.10240914.
31. Shivani, Chauhan D, Rani D. A feasibility restoration particle swarm optimizer with chaotic maps for two-stage fixed-charge transportation problems. Swarm Evol Comput. 2024;91(1):101776. doi:10.1016/j.swevo.2024.101776.
32. Hsu H, Lachenbruch PA. Paired t test. Hoboken, NJ, USA: John Wiley & Sons, Inc.; 2014. doi:10.4135/9781506359182.
33. Hollander M, Wolfe DA, Chicken E. Nonparametric statistical methods. Hoboken, NJ, USA: John Wiley & Sons, Inc.; 2014.