scientific reports

Check for updates

OPEN

Research on multi-UAV autonomous obstacle avoidance algorithm integrating improved dynamic window approach and ORCA

Xucheng Chang[1,3], Jingyu Wang[2,3]✉, Kang Li[2,3], Xinhui Zhang[2,3] & Qian Tang[2,3]

To address the issue that traditional UAV obstacle-avoidance algorithms had low efficiency in unknown and complex environments, an improved DWA (Dynamic Window Approach) fusion algorithm was proposed. Regarding the lack of a global perspective in the DWA algorithm, a bidirectional search strategy was introduced to enhance the global value of the planned trajectory. Confronted with the difficulty of balancing calculation speed and accuracy in the DWA algorithm, a dynamic time step adjusted according to the environment was designed to weigh the computational efficiency. Aiming at the poor environmental adaptability of the DWA algorithm, a trajectory evaluation function with variable weights was put forward to improve environmental fitness. To boost the inter-UAV obstacle-avoidance ability in the multi-UAV collaborative mode, the improved DWA algorithm was integrated with the Optimal Reciprocal Collision Avoidance (ORCA) method. Simulation experiments were conducted to verify the effectiveness of the proposed improved fusion algorithm. Compared with the conventional DWA algorithm, the proposed method achieved a 27.90% reduction in UAV flight path length, a 17.01% decrease in mission completion time, and a 21.5% reduction in iteration counts. These significant performance improvements demonstrated its practical value for engineering applications of multi-UAV autonomous obstacle-avoidance technology.

Keywords UAVs, Improved DWA algorithm, ORCA algorithm, Autonomous obstacle avoidance

With the rapid advancement of intelligent system technologies, unmanned aerial vehicle (UAV) technology had achieved significant maturity and demonstrated tremendous application potential across various domains including agriculture, civilian applications, and military operations. Within this context, researchers had initiated explorations into cooperative multi-UAV working modes, seeking to expand mission scopes and enhance overall system performance through collaborative operations. Autonomous obstacle avoidance technology for multiple UAVs had consequently emerged as a prominent research focus[1–5]. Nevertheless, achieving efficient autonomous obstacle avoidance for UAVs operating in complex, unknown environments remained a critical challenge requiring urgent resolution[6–9].

UAV obstacle avoidance algorithms represented one of the core technologies in autonomous UAV navigation systems, primarily categorized into global and local obstacle avoidance approaches. Global obstacle avoidance algorithms such as A* and Dijkstra relied on complete environmental information for path planning. While these algorithms could identify globally optimal paths, they typically demanded substantial computational resources and exhibited poor real-time performance, rendering them unsuitable for unknown or complex environments. In contrast, local obstacle avoidance algorithms including the Dynamic Window Approach (DWA) and Artificial Potential Field (APF) operated independently of global environmental information. These methods could rapidly adapt to environmental changes while maintaining high computational efficiency, making them particularly suitable for scenarios with stringent real-time requirements. However, such local algorithms demonstrated limited search capabilities and were prone to converging on locally optimal solutions, thereby constraining their effectiveness in unknown and complex environments[8,10–12].

[1]School of Automation, Zhengzhou University of Aeronautics, Zhengzhou 450046, China. [2]School of Aerospace Engineering, Zhengzhou University of Aeronautics, Zhengzhou 450046, China. [3]Xucheng Chang, Jingyu Wang, Kang Li, Xinhui Zhang and Qian Tang contributed equally to this work. ✉email: jywang0802@163.com

When operating in unknown and complex environments, UAVs encountered multiple challenges including difficulties in acquiring global information, unpredictable dynamic obstacles, and intricate multi-UAV interactions. These challenges rendered traditional global obstacle avoidance algorithms—or even hybrid global–local approaches—inadequate in meeting mission requirements for real-time performance and computational efficiency, often resulting in suboptimal effectiveness[3,7,10,11,13]. The Dynamic Window Approach (DWA) algorithm, as a prevalent obstacle avoidance method in UAV applications, had emerged as a primary solution for enhancing autonomous avoidance capabilities due to its demonstrated advantages in local path planning and rapid response. Subsequent improvements and optimizations to the DWA algorithm were consequently positioned to address these critical obstacle avoidance challenges in unknown, complex environments[14–16].

Regarding obstacle avoidance algorithms for UAVs in unknown and complex environments, literature[17] proposed a strategy that integrated global path planning and local path planning. This strategy effectively overcame the limitations of traditional algorithms in complex dynamic environments by combining the A* algorithm and DWA algorithm, and introducing key nodes of the global path and an adaptive window mechanism. However, this improved algorithm based on the A* algorithm had a strong dependence on environmental information and may have been difficult to apply in completely unknown environments. In contrast, the improved method proposed in this study, based on the Dynamic Window Approach (DWA) algorithm, fully leveraged DWA's advantages in real-time performance and prior environment knowledge independence, making it more suitable for applications in unknown and complex environments. Literature[18] addressed the limitations of the traditional Rapidly exploring Random Tree (RRT) algorithm in dynamic obstacle avoidance by proposing an improved fusion algorithm of RRT and DWA. This algorithm improved the efficiency and safety of path planning by introducing strategies such as goal-biased random sampling, adaptive step sizes, and adaptive border node filtering. However, because the RRT algorithm incurred high computational costs when dealing with complex environments, fusing it with the DWA algorithm may have further increased the computational burden. In contrast, the proposed method incorporated an improved bidirectional search strategy, which not only reduced computational load but also enhanced the global optimality of planned trajectories, thereby significantly improving the algorithm's overall performance. Literature[19] proposed a fusion algorithm that combined the Vector Field Histogram Plus (VFH+) algorithm with an improved DWA algorithm, incorporating a variable weight factor from the A* algorithm, thereby enhancing local path planning capabilities and significantly optimizing obstacle avoidance effects in UAV formation flight. However, the improved cooperative obstacle avoidance control algorithm had deficiencies in rapid response and may have struggled to effectively address obstacle avoidance challenges posed by high-speed moving obstacles. To address this issue, the adaptive evaluation function introduced in this paper dynamically adjusted weight coefficients in real time, enabling better adaptation to varying environments and UAV states. This significantly enhanced its responsiveness, particularly in effectively tackling the challenge of avoiding high-speed moving obstacles. By leveraging the inherent responsiveness of the DWA algorithm, the enhanced adaptive DWA algorithm demonstrated superior environmental adaptability and state awareness, enabling more effective high-speed obstacle avoidance and excelling in rapid response scenarios involving fast-moving obstacles.

The Optimal Reciprocal Collision Avoidance (ORCA) algorithm was a distributed obstacle avoidance algorithm designed for multiple agents in dynamic local environments. It utilized mathematical and geometric models to handle obstacle avoidance problems through each agent's perception of the local environment, without requiring direct communication between agents. With fast execution speeds, it was suitable for scenarios involving collaborative collision avoidance between multiple agents. Literature[20] proposed a method for optimal speed vector decision-making for ships based on the ORCA algorithm. This method enabled each intelligent ship to independently and autonomously perform distributed collaborative collision avoidance. It achieved distributed collaborative collision avoidance for intelligent ships and provided a useful reference for obstacle avoidance in multi-UAV scenarios. Given the strengths of the ORCA algorithm in multi-agent collision avoidance, this paper proposed a fusion of the DWA and ORCA algorithms. Building upon the strengths of Optimal Reciprocal Collision Avoidance (ORCA) in multi-agent collision avoidance, this study proposed a novel hybrid approach that integrated the Dynamic Window Approach (DWA) with ORCA. This synergistic integration effectively compensated for DWA's inherent limitations in inter-drone obstacle avoidance, thereby offering an innovative solution for multi-UAV cooperative operations.

This study developed an improved DWA-ORCA algorithm to address the limitations of conventional UAV obstacle avoidance approaches in complex unknown environments with multiple UAVs. The research comprised two key components: first, comprehensive enhancements to single-UAV algorithm performance through a bidirectional search strategy that expanded path planning perspectives and improved trajectory quality, an environmentally adaptive time-step mechanism that optimized the speed-accuracy tradeoff, and a dynamic weighted evaluation function that enabled real-time environmental adaptation; second, an innovative integration of the improved DWA algorithm with ORCA for multi-UAV coordination, combining DWA's efficient local planning with ORCA's robust multi-agent collision avoidance capabilities. These synergistic advancements not only overcame traditional DWA's limitations in global optimization, computational efficiency and environmental adaptability, but also significantly improved path planning performance in unknown environments while establishing an effective solution for multi-UAV autonomous obstacle avoidance that enhanced both system performance and mission execution efficiency in complex operational scenarios.

UAV kinematics model

Model building hypothesis

Kinematic modeling of UAVs served as the theoretical foundation for the design of path-planning algorithms. In this chapter, based on the dynamic characteristics of quadrotor UAVs and under reasonable simplifications, a

kinematic model was established to provide mathematical support for the subsequent improvement of the DWA algorithm. The kinematic model of the quadrotor UAV was simplified based on the following assumptions:

(1) The low-speed flight assumption held that the flight Mach number remained below 0.3 (Ma < 0.3), which permitted the Euler angle rates to approximate the body angular velocities while disregarding the coupling effects of rotor dynamics and gyroscopic moments.
(2) The rigid body assumption maintained that the UAV's mass was uniformly distributed and that its inertia tensor matrix exhibited a diagonalized form.
(3) The gravitational field assumption treated the gravitational acceleration as constant and neglected the effects of Earth's curvature and Coriolis force.

Coordinate system definition

(1) World Coordinate Frame OE - XEYEZE: The East-North-Up (ENU) geographic coordinate system was adopted, where the X-axis points east, the Y-axis pointed north, and the Z-axis pointed vertically upward. Based on the assumptions, the established kinematic model for the UAV is illustrated in Fig. 1,the three-dimensional angular velocity componented in the world coordinate frame were expressed as shown in Eq. (1),

$$\omega b = [\omega\psi, \omega\theta, \omega\phi] \tag{1}$$

In the equation: $\omega\psi, \omega\theta, \omega\phi$ represented the angular velocities of yaw, pitch, and roll, respectively.

(2) Body Coordinate Frame OB - XBYBZB: The origin was located at the UAV's center of mass, with the X -axis aligned with the longitudinal axis pointing forward, the Y-axis aligned with the transverse axis pointing to the right, and the Z-axis following the right-hand rule. The three-dimensional velocity components in the body coordinate frame were expressed as shown in Eq. (2),

$$v_B = [vx, vy, vz] \tag{2}$$

In the equation: x, y, z represented the linear velocities of the UAV along the x, y, z-axes, respectively.

Kinematical equation

(1) Linear motion equations.

Based on the coordinate transformation relationship, the mapping from body-frame velocities to world-frame velocities was achieved through a rotation matrix,

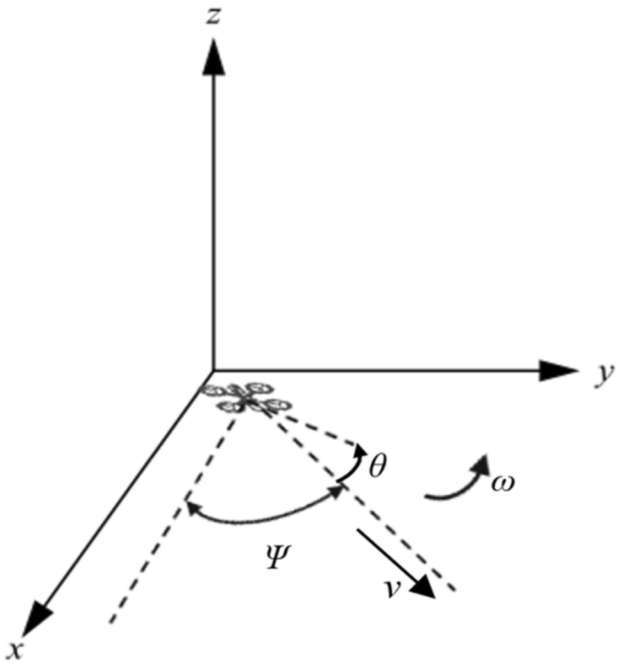


Fig. 1. Illustration of the drone kinematics model.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = R(\psi, \theta) \begin{bmatrix} vx \\ vy \\ vz \end{bmatrix} \tag{3}$$

where the rotation matrix $R(\psi, \theta)$ was defined as,

$$R(\psi, \theta) = \begin{bmatrix} \cos\psi\cos\theta & -\sin\psi & \cos\psi\sin\theta \\ \sin\psi\cos\theta & \cos\psi & \sin\psi\sin\theta \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \tag{4}$$

Assuming the UAV was flying at low speed and without roll motion $\phi = 0$, the lateral velocity $Vy = 0$ could be simplified as,

$$\begin{cases} \dot{x} = vz\cos\psi\cos\theta \\ \dot{y} = vx\sin\psi\cos\theta \\ \dot{z} = vz \end{cases} \tag{5}$$

(2) Angular Velocity Equations.

Under the assumption of low speed and small angles ($\theta \ll 1$), higher-order nonlinear terms could be neglected, and the derivatives of the Euler angles could be approximately linearized with respect to the body angular velocities.

$$\begin{cases} \dot{\psi} = \omega_{\psi} \\ \dot{\theta} = \omega_{\theta} \end{cases} \tag{6}$$

Kinematical equation

Based on the Euler method, the continuous motion equations were discretized. Through the combination of Eqs. (5) and (6), the trajectory recursion formula within the time step $\triangle$ t was obtained.

$$\begin{cases} x_{k+1} = x_k + v_k\cos\psi_k\cos\theta_k.\Delta t \\ y_{k+1} = y_k + v_k\sin\psi_k\cos\theta_k.\Delta t \\ z_{k+1} = z_k + v_z.\Delta t \\ \psi_{k+1} = \psi_k + \omega_{\psi}.\Delta t \\ \theta_{k+1} = \theta_k + \omega_{\theta}.\Delta t \end{cases} \tag{7}$$

Horizontal Motion: The forward velocity vk was projected onto the horizontal plane through $\cos\theta k$ and then decomposed into eastward and northward components through $\cos\psi_k / \sin\psi_k$.

Vertical Motion: The vertical velocity v_z was directly accumulated to achieve altitude control.

Attitude Update: The Euler angles were updated by integrating the angular velocities, ensuring smooth changes in heading and pitch.

To ensure the feasibility of the generated trajectory, a curvature constraint was introduced.

$$\kappa = \frac{\left\|\boldsymbol{\omega}^{\mathrm{B}} \times \boldsymbol{v}^{\mathrm{B}}\right\|}{\left\|\boldsymbol{v}^{\mathrm{B}}\right\|^3} \leq \frac{1}{Rmin} \tag{8}$$

In the equation: $Rmin$ represented the minimum turning radius of the UAV.

Dynamic window approach and its improvements

Dynamic window approach algorithm

The Dynamic Window Approach (DWA) was a commonly used local obstacle avoidance algorithm in the field of UAV path planning. Based on the UAV's dynamic characteristics and mission environment constraints, the algorithm constructed a velocity-constrained space. Within this space, the algorithm sampled velocities and generated a series of feasible predicted trajectories by incorporating the UAV's kinematic equations. Finally, the algorithm evaluated these predicted trajectories using a predefined cost function to select and execute the optimal trajectory in the current space[21]. The specific steps were as follows,

(1) Velocity-Constrained Space.

During UAV operation, its resultant velocity, resultant angular velocity (v), resultant acceleration (a), and resultant angular acceleration $(\dot{\omega})$ were constrained by factors such as UAV performance limits $Vp(v, \omega)$, power constraints $Vd(\dot{v}, \dot{\omega})$, and obstacle avoidance requirements $Vo(v, \omega)$. The specific relationships were given by Eqs. (9)–(11)[22],

$$Vp(v,\boldsymbol{\omega}) = \begin{cases} v \in [v\min, v\max] \\ \omega\theta \in [\omega\theta, \min, \omega\theta, \max] \\ \omega\psi \in [\omega\psi, \min, \omega\psi, \max] \end{cases} \tag{9}$$

$$Vd(v,\boldsymbol{\omega}) = \begin{cases} v \in [vc - \dot{v}\max\Delta t, vc + \dot{v}\max\Delta t] \\ \omega\psi \in [\omega\psi, c - \dot{\omega}\psi, \max\Delta t, \omega\psi, c + \dot{\omega}\psi, \max\Delta t] \\ \omega\theta \in [\omega\theta, c - \dot{\omega}\theta, \max\Delta t, \omega\theta, c + \dot{\omega}\theta, \max\Delta t] \end{cases} \tag{10}$$

$$Vo(v,\boldsymbol{\omega}) = \{((v,\omega\psi,\omega\theta))\,|dist((v,\omega\psi,\omega\theta) > safe_dist\} \tag{11}$$

In the equations: $v\min$, $v\max$ denoted the UAV's minimum and maximum velocities respectively; $\dot{v}\min$, $\dot{v}\max$ represented the UAV's minimum and maximum resultant accelerations respectively; $\omega_{\theta,\min}, \omega_{\psi,\min}$ and $\omega_{\theta,\max}, \omega_{\psi,\max}$ indicated the minimum and maximum angular velocities for pitch angle and yaw angle respectively; $\dot{\omega}_{\theta,\min}, \dot{\omega}_{\psi,\min}$ and $\dot{\omega}_{\theta,\max}, \dot{\omega}_{\psi,\max}$ specified the minimum and maximum angular accelerations for pitch angle and yaw angle respectively; $\mathrm{safe_dist}$ was defined as the minimum clearance distance between the UAV and nearby obstacles when operating in state (v_0, ω_0).

Consequently, the velocity-constrained space during UAV operation was mathematically formulated as Eq. (12),

$$Vr(v,\boldsymbol{\omega}) = Vp(v,\boldsymbol{\omega}) \cap Vd(v,\boldsymbol{\omega}) \cap Vo(v,\boldsymbol{\omega}) \tag{12}$$

(2) Hierarchical Sampling Optimization.

The conventional DWA algorithm was originally designed for 2D path planning. When extending this 2D algorithm to three-dimensional space, the dimensionality of the search space increased significantly. In the 3D velocity-constrained space, maintaining the traditional uniform sampling approach would have substantially increased computational load, thereby compromising the real-time advantage of the DWA algorithm. To address this challenge, our study introduced a hierarchical sampling strategy that decomposed the 3D velocity space into horizontal and vertical motion layers.

Horizontal Motion Layer Sampling: In the horizontal motion layer, uniform sampling was applied to the resultant linear velocity $\boldsymbol{v}$ and yaw angular velocity ω_ψ, generating candidate velocity pairs $(\boldsymbol{v}_{\mathrm{window}}, \omega_{\psi\,\mathrm{window}})$.

$$v_{window} = [\max(0, v_{cur} - a_v\Delta t), \min(v_{\max}, v_{cur} + a_{\omega_\psi}\Delta t)] \tag{13}$$

$$\omega_{\psi,\mathrm{window}} = [\max(-\omega_{\psi,\max}, \omega_{\psi,\mathrm{cur}} - a_{\omega_z}\Delta t), \min(\omega_{\psi,\max}, \omega_{\psi,\mathrm{cur}} + a_{\omega_\psi}\Delta t)] \tag{14}$$

In the equations: v_{cur} and $\omega_{\psi,\mathrm{cur}}$ represented the current linear velocity and yaw angular velocity, respectively; av and $a\omega\psi$ denoted the current linear acceleration and yaw angular acceleration, respectively.

Vertical Motion Layer Sampling: For each candidate horizontal velocity pair $(\boldsymbol{v}, \omega_\psi)$, sparse sampling was applied exclusively to the pitch angular velocity ω_θ in the vertical motion layer, effectively reducing the search space dimensionality.

$$\omega_{\theta,\mathrm{window}} = [\max(-\omega_{\theta,\max}, \omega_{\theta,\mathrm{cur}} - a_{\omega_\theta}\Delta t), \min(\omega_{\theta,\max}, \omega_{\theta,\mathrm{cur}} + a_{\omega_\theta}\Delta t)] \tag{15}$$

In the equation: $\omega_{\theta,\mathrm{cur}}$ represented the current pitch angular velocity; a_{ω_θ} denoted the pitch angular acceleration.

(3) Trajectory Prediction.

The UAV's resultant velocity was decomposed along the three axes x, y, z. Considering velocity changes occur within an extremely small time interval dt, the motion was approximated as uniform linear motion. Under this assumption, velocity sampling was performed on the velocity-constrained space $Vr(vt, \boldsymbol{\omega}t)$ at time t, and combined with the UAV trajectory recursion formula in Eq. (7), the predicted trajectory of the UAV at time $(t + dt)$ was obtained.

$$\boldsymbol{x}(t + dt) = \boldsymbol{x}(t) + \Delta t.\mathrm{f}(\boldsymbol{x}(t), u(t)) \tag{16}$$

In the equation: $\boldsymbol{x}(t) = [x, y, z, \psi, \theta, \omega_\psi, \omega_\theta]^T$ was the UAV state vector, $\boldsymbol{u}(t) = [\boldsymbol{v}, \omega_\psi, \omega_\theta]^T$ was the control input vector, was the kinematic model, yielding Eq. (17),

$$\begin{cases} \dot{x} = v\cos\theta\cos\psi \\ \dot{y} = v\sin\theta\cos\psi \\ \dot{z} = v\sin\psi \end{cases} \tag{17}$$

In the equation:

$$\begin{cases} \dot{\theta} = \omega_\theta \\ \dot{\psi} = \omega_\psi \end{cases} \tag{18}$$

Based on the above equations, the predicted 3D trajectory space at time was obtained, as shown in Fig. 2, where $\boldsymbol{v}, \omega_\psi, \omega_\theta$ represented the constraints on $(\dot{\boldsymbol{v}}_a, \dot{\boldsymbol{v}}_b, \dot{\boldsymbol{\omega}}_{\psi,a}, \dot{\boldsymbol{\omega}}_{\psi,b}, \dot{\boldsymbol{\omega}}_{\theta,a}, \dot{\boldsymbol{\omega}}_{\theta,b}) \subset \boldsymbol{v}_r(v, \boldsymbol{\omega}_\psi, \boldsymbol{\omega}_\theta)$ on the UAV's selectable resultant acceleration, yaw angular velocity, and pitch angular velocity, respectively, within the trajectory window at time t_i.

(4) Performance evaluation.

The trajectory evaluation was conducted using the cost functions specified in Eqs. (19), (21), and (22) to quantitatively assess the predicted trajectories derived from Eq. (17). After normalizing the scores of each sub-cost function, they were summed and compared to select the trajectory with optimal overall performance. The corresponding parameters $\boldsymbol{v}$ and $\boldsymbol{\omega}$ for this optimal state were subsequently output.

(A) The heading evaluation sub-function $\mathrm{head}(v_i, \omega_{\psi,i}, \omega_{\theta,i})$ quantified the angular deviation between the UAV's current orientation and the target point, ensuring precise trajectory tracking accuracy.

$$\mathrm{heading}(v_i, \omega_{\psi,i}, \omega_{\theta,i}) = 1 - \frac{|\Delta\theta_\psi| + |\Delta\theta_\theta|}{\pi} \tag{19}$$

In the equation:

$$\begin{cases} \Delta\theta\mathrm{goal},\psi = \theta\psi - \theta\psi, i \\ \Delta\theta\mathrm{goal},\theta = \theta\theta - \theta\theta, i \end{cases} \tag{20}$$

In the equation: $\theta\mathrm{goal}, \psi, \theta\mathrm{goal}, \theta$ represented the target azimuth angle and elevation angle respectively, and $\theta\psi, i, \theta\theta, i$ denoted the azimuth angle and elevation angle at the trajectory endpoint.

(B) Obstacle evaluation sub-function $\mathrm{dist}(vi, \omega\psi, i, \omega\theta, i)$ assessed the distance between the UAV and the nearest obstacle to ensure path safety.

$$\mathrm{dist}(vi, \omega\psi, i, \omega\theta, i) = \begin{cases} \min(D_{io})_{0\le i\le n}, if\ \min(D_{io})_{0\le i\le n} < D_{safe} \\ D_{safe}, others \end{cases} \tag{21}$$

In the equation: $\min(D_{io})_{0\le i\le n}$ is the minimum Euclidean distance from the trajectory endpoint to all obstacles, and D_{safe} is the obstacle avoidance safety distance.

(C) Velocity ν_i Evaluation Sub-function evaluated the current velocity magnitude to ensure trajectory efficiency.

$$\mathrm{velocity}(vi) = (|vi|) \tag{22}$$

$$\mathrm{f}(vi, \omega\psi, i, \omega\theta, i) = \alpha 1 \frac{heading(vi, \omega\psi, i, \omega\theta, i)}{\sum_i^n heading(vi, \omega\psi, i, \omega\theta, i)} + \alpha 2 \frac{\mathrm{dist}(vi, \omega\psi, i, \omega\theta, i)}{\sum_i^n \mathrm{dist}(vi, \omega\psi, i, \omega\theta, i)} + \alpha 3 \frac{\mathrm{velocity}(vi)}{\sum_i^n \mathrm{velocity}(vi)} \tag{23}$$

In the equation: $\alpha 1, \alpha 2, \alpha 3$ represent the weighting coefficients for the heading evaluation sub-function, obstacle distance evaluation sub-function, and velocity evaluation sub-function, respectively[23].

The "Dual-Target Point" bidirectional search strategy

The bidirectional search strategy represented an efficient optimization approach for path planning, which significantly improved search efficiency by simultaneously conducting trajectory searches from both the starting and ending points, thereby reducing the number of search nodes. While the Dynamic Window Approach (DWA)

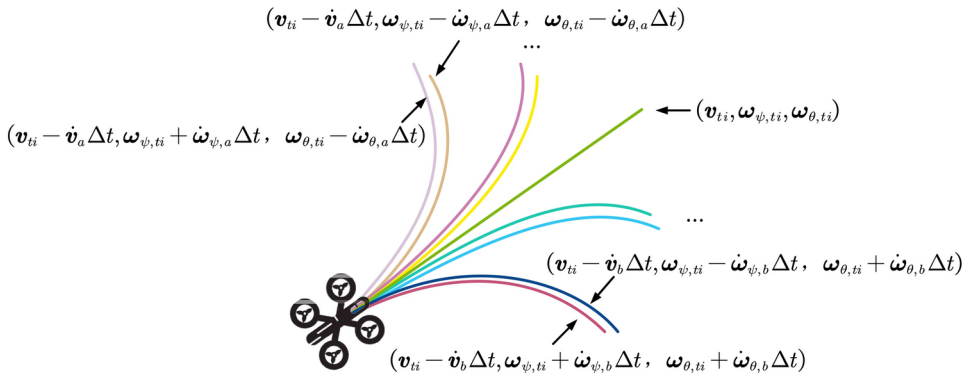


Fig. 2. Predicted trajectory space at time.

algorithm had been widely adopted due to its rapid response and low computational overhead, it exhibited inherent limitations in global path planning. To address this deficiency, a bidirectional search strategy was incorporated into the DWA algorithm, enhancing the global value of planned paths through expanded search perspectives while further improving algorithmic efficiency.

In conventional bidirectional search approaches, the local path planning algorithms' focus on obtaining locally optimal solutions often led to difficulties in effectively converging paths from both directions. This study proposed an innovative "dual-target point" bidirectional search strategy to resolve this issue. Within this framework, both the mission's starting and ending points were treated as departure points while simultaneously serving as each other's target points, ensuring the search process converged to a uniquely determined planned path. The flowchart of the bidirectional DWA algorithm was presented in Fig. 3, while the conceptual diagram of the dual-target point search was illustrated in Fig. 4.

The implementation of the dual-target point search strategy yielded the pseudocode for the bidirectional DWA algorithm as presented in Table 1.

PSO-based optimal time step search

The DWA algorithm performed obstacle avoidance and path planning through a dynamic window. The configuration of this dynamic window, which was influenced by factors such as UAV hardware constraints, mission environmental limitations, and algorithmic time steps, constituted a critical determinant of algorithm performance. Conventional DWA algorithms typically employed fixed time steps predetermined based on UAV specifications and environmental data. This approach resulted in insufficient adaptability to local environmental variations, relatively slow response times, and challenges in achieving an optimal balance between computational speed and accuracy. Given that hardware constraints and environmental limitations were generally fixed and non-adjustable, modifying the time step emerged as the most effective means to optimize dynamic window

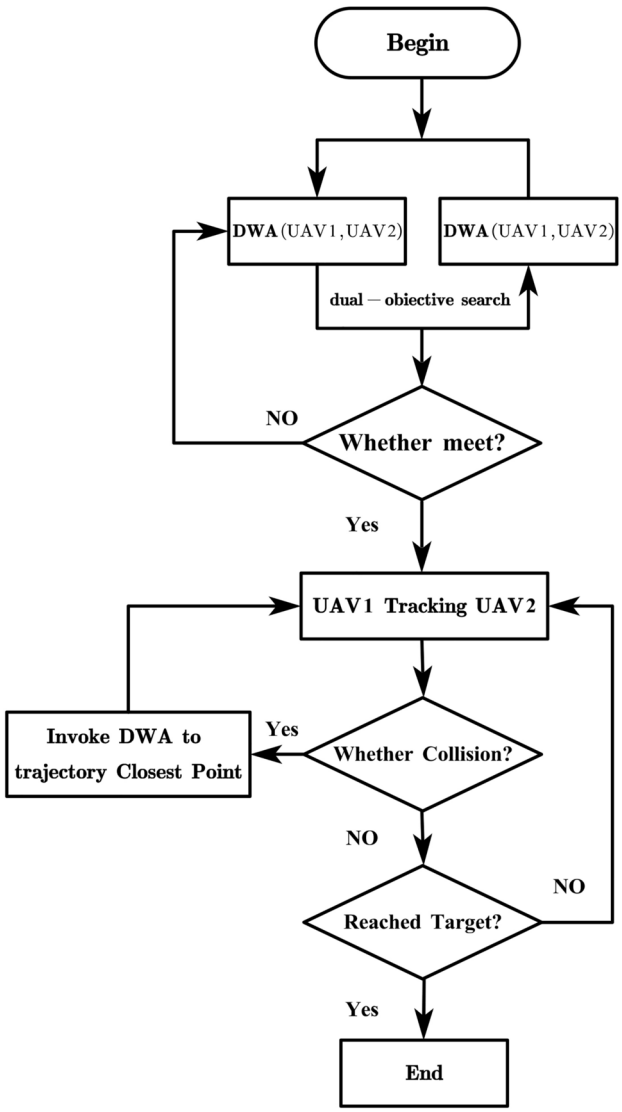


Fig. 3. Flowchart of bidirectional DWA algorithm.

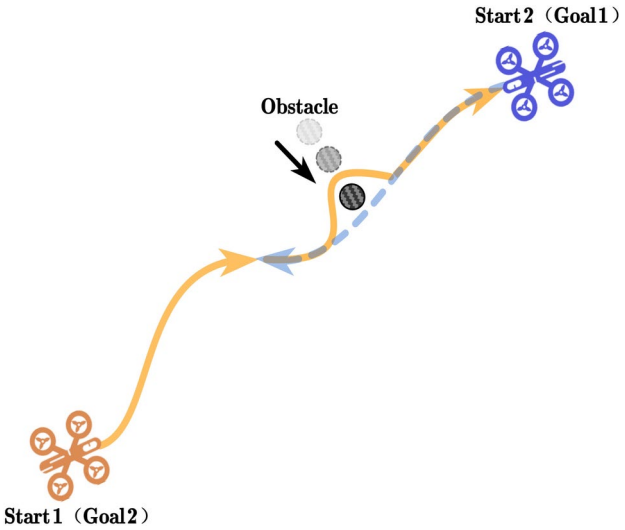


Fig. 4. Concept map for dual target search.

Input: Start pose s = (x_s, y_s, z_s, ψ_s, θ_s), Goal g = (x_g, y_g, z_g, ψ_g, θ_g), Obstacle set O
Output: Optimal trajectory τ*
1: Initialize dual agents:
$x_1 \leftarrow$ [x_s, y_s, z_s, ψ_s, θ_s]^T // Forward agent
$x_2 \leftarrow$ [x_g, y_g, z_g, ψ_g+π, θ_g-π]^T // Backward agent
2: while $\|x_1[1:3] - x_2[1:3]\|_2 > \varepsilon$ do
3: // Forward search
4: (v_h, ω_ψ) ~ Φ_$h(x_1)$ // Horizontal sampling (Eq.6)
5: ω_θ ← {ω_θ | ω_θ ∈ [θ-Δθ, θ+Δθ], N=3} // Vertical sparse sampling
6: $\tau_1 \leftarrow$ argmax{αG_h(·) + βG_d(·)} // Eq.12-13
7: // Backward search
8: (v'_h, ω'_ψ) ~ Φ_$h(x_2)$
9: ω'_θ ← {ω'_θ | ω'_θ ∈ [θ'-Δθ, θ'+Δθ], N=3}
10: $\tau_2 \leftarrow$ argmax{αG'_h(·) + βG'_d(·)}
11: // State propagation
12: $x_1 \leftarrow$ f(x_1, τ_1.controls) // Kinematic update (Eq.7)
13: $x_2 \leftarrow$ f(x_2, τ_2.controls)
14: $\tau^* \leftarrow \tau_1 \cup \tau_2$ // Trajectory stitching
15: return τ*

Table 1. The pseudo-code for the bidirectional DWA algorithm.

configuration and enhance the overall performance of the DWA algorithm[24]. Within the framework of local algorithms, this study proposed a methodology that adopted the principle of approximating global optima through local optimization. Specifically, we introduced a Particle Swarm Optimization (PSO)-based approach to determine offline optimal time steps by analyzing environmental conditions and UAV states, subsequently verified through Response Surface Methodology (RSM). The operational environment information (including the number of obstacles within detection range and minimum obstacle spacing) and UAV state parameters (velocity and attitude under fixed UAV type specifications) underwent decoupling processing. Compared to direct global search fitting, this separated search-and-fitting approach reduced dimensionality and computational load more effectively, better aligning with the real-time requirements of the DWA algorithm. Considering the algorithm's

application in unknown complex environments, the dynamic time step $\mathrm{f}(d\min, \boldsymbol{v}, \boldsymbol{\omega})$ was formulated as a function. The specific implementation proceeded as follows,

(A) Parameter Initialization: The UAV performance parameters, obstacle distribution patterns, fixed time steps, and UAV dynamic constraints were configured.
(B) Training Experiments and Data Collection: Using Latin Hypercube Sampling (LHS), we generated a multi-factor experimental matrix where each experimental condition consisted of parameter triplets $(d\min, \boldsymbol{v}, \boldsymbol{\omega}) \in R^3$. Extensive training experiments were conducted, during which the UAV's flight trajectories were recorded and relevant data were systematically collected and stored.
(C) Trajectory Discretization: The trajectories obtained from simulation experiments underwent discretization processing. Each simulated trajectory was segmented into n discrete elements using fixed unit time steps, as illustrated in Fig. 5. This process constructed a training dataset incorporating environmental and state characteristics, with the schematic representation shown in the same figure.
(D) Optimal Time Step Search Within Micro-Elements: The particle swarm parameters were configured under consistent initial conditions. Specifically, for parameter $(d\min, \boldsymbol{v}, \boldsymbol{\omega}) \in R^3$, trajectory elements with identical values of $d\min$,$\boldsymbol{v}$, and $\boldsymbol{\omega}$ were combined to form search intervals. The PSO algorithm was then integrated with the DWA method to perform parallel searches for the optimal time step within each element, based on three experimental conditions. During the search process, the evaluation function and fitness function were set as shown in Eqs. (24) and (25) respectively.

The DWA algorithm evaluation function was:

$$\mathrm{f}(vi, \omega\psi, i, \omega\theta, i) = 0.2 \times \frac{heading(vi, \omega\psi, i, \omega\theta, i)}{\sum_{i}^{n} \mathrm{heading}(vi, \omega\psi, i, \omega\theta, i)} + 0.4 \times \frac{\mathrm{dist}(vi, \omega\psi, i, \omega\theta, i)}{\sum_{i}^{n} \mathrm{dist}(vi, \omega\psi, i, \omega\theta, i)} \tag{24}$$

The PSO algorithm fitness function was:

$$\mathrm{F}(dt) = 0.5 \times \frac{1}{T(dt) - T\min} + 0.5 \times \frac{S(dt) - S\min}{S\max - S\min} \tag{25}$$

In the equation: $T\min$ represented the minimum completion time, while $S\max$, $S\min$ denoted the maximum and minimum trajectory smoothness metrics, respectively. The dynamic time-step mechanism was designed to balance computational efficiency against trajectory quality; consequently, the velocity sub-evaluation function was omitted from the DWA algorithm during training experiments. The PSO fitness function incorporated both computation time and trajectory smoothness components, with detailed parameter specifications available in references[25,26].

(E) Data Fitting and Dynamic Time-Step Curve Generation: Multiple simulation experiments were conducted under various environmental conditions with fixed time-step settings. Through multivariate conditional optimization, we obtained dataset dt of optimal time steps, which was then combined with parameter set

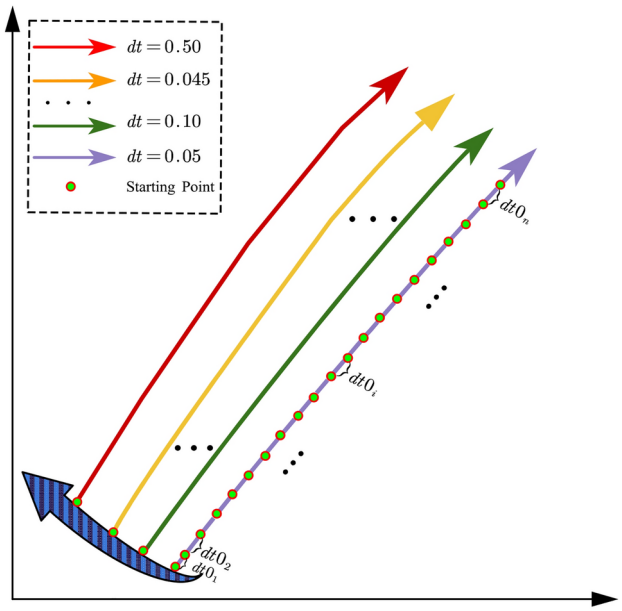


Fig. 5. The schematic diagram for micro-elementalization of multiple trajectories.

$\boldsymbol{\omega} - dt$ for data fitting. This process generated individual optimal time-step curves under single-dimensional constraints. With the UAV radius set to 0.5 m, three characteristic fitting curves were produced as shown in Fig. 6.

Analysis of Fig. 5a revealed that in the curve of minimum obstacle spacing versus optimal time step: when condition $dmin \in (0, 0.5)$ occurred, obstacle avoidance took higher priority in the algorithm's operation, resulting in a positive correlation between the optimal time step and parameter $d_{\min}$. However, when condition $dmin = 0.5$ was met (i.e., when $dmin$ equaled the UAV's radius), the trend of the fitted curve changed significantly. Specifically, under condition $dmin \in (0.5,1)$, the algorithm began to comprehensively consider multiple performance factors including obstacle avoidance, velocity, and trajectory smoothness, leading to a nonlinear relationship between the optimal time step and parameter $d_{\min}$. Analysis of Fig. 5b and c revealed nonlinear relationships in the fitted curves between the UAV's linear velocity ν and optimal time step dt, where both parameters $v - dt$ and $\omega - dt$ exhibited nonlinear characteristics constrained by computational efficiency and heading angle variations. When velocity was relatively low (condition $v \in (0, 0.17)$), the algorithm prioritized trajectory coverage efficiency, resulting in a positive correlation for parameter $v - dt$. As velocity increased (condition $v \in (0.17,0.4)$), trajectory smoothness became the limiting factor for time step growth. Under condition $v \in (0.4,0.6)$, the optimal step length achieved a balance between search efficiency and trajectory smoothness. Within range $\omega \in (-0.3, -0.1) \cap (0.2,0.3)$, the optimal time steps remained stable with minor fluctuations around 0.05, while in range $\omega \in (-0.1,0.2)$, parameter ω showed symmetry about $\omega = 0$, where smaller absolute values of ω enabled better target tracking and permitted larger time steps. These observed trends in the optimal time step fitting curves were theoretically consistent with the principles of the DWA algorithm and reflected practical operational requirements.

(F) Multivariable Coupled Response Surface Modeling and Interaction Mechanism Analysis: When considering three-dimensional parameter space $(d\min, v, \boldsymbol{\omega})$, exhaustive search methods suffered from exponential growth in computational complexity, while pure numerical optimization failed to reveal the underlying interaction mechanisms between parameters. To address these limitations, Response Surface Methodology (RSM) was introduced to construct explicit parameter-response relationship models. This approach enabled the establishment of global approximation models using limited experimental data, which effectively analyzed multi-factor coupling mechanisms and achieved efficient yet accurate offline decision-making with low memory requirements.

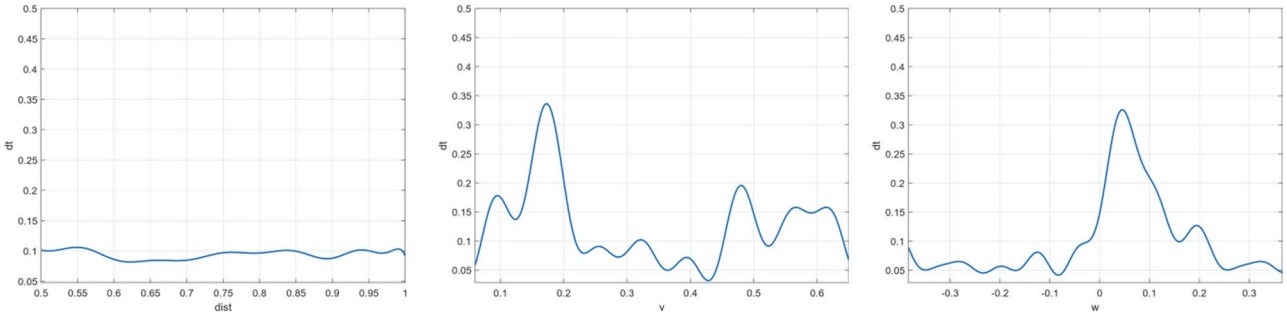
The RSM model was constructed using optimized sample sets $\left\{x^{(k)}, dt_{opt}^{(k)}\right\}_{k=1}^{m}$ obtained through parallel PSO searches,

$$d\hat{t} = \beta 0 + \sum_{i=1}^{3} \beta i \hat{x} i + \sum_{i=1}^{3} \beta i i \hat{x} i^2 + \sum_{i<j} \beta i j \hat{x} i \hat{x} j \tag{26}$$

Based on parameter set $dt = \boldsymbol{X}\boldsymbol{\beta}$, an optimal time-step RSM model was successfully developed, where parameter $\boldsymbol{X} = [1 dv\omega d^2 v^2 \omega^2 dvd\omega v\omega]$ was incorporated to achieve the fitted result $\beta = [0.096\ -0.075\ 0.07 - 0.188\ 0.0770.061\ -0.189\ -0.110\ 0.0350.350]$. The resulting spatial function of dt obtained through this fitting process is visually presented in Fig. 7.

Design of state-information-based adaptive weighting function

The conventional Dynamic Window Approach (DWA) algorithm evaluated feasible trajectories by assigning fixed weight coefficients to factors such as heading angle, obstacle distance, and velocity, enabling selection of relatively optimal trajectories in the configuration space. However, this static weighting mechanism significantly compromised environmental adaptability, resulting in suboptimal performance in complex, dynamic



a. $d_{\min} - dt$ b. $v - dt$ c. $\omega - dt$

Fig. 6. Fitting curve for optimal time step.

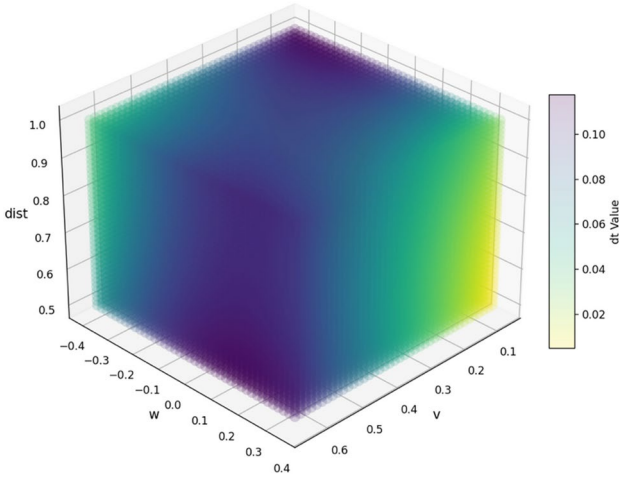


Fig. 7. Fitting curve for optimal time step.

environments. To address this limitation, this study developed an adaptive trajectory evaluation function that dynamically adjusted weight coefficients based on real-time environmental conditions and UAV state information, thereby optimizing performance across diverse mission scenarios and operational states.

(1) Adaptive heading angle evaluation.

The conventional heading angle evaluation method primarily focused on the deviation between the UAV's actual heading/pitch angles and the target angles. However, in complex environments, this approach was observed to cause abrupt changes in both heading and pitch angles, which adversely affected trajectory smoothness. To address this limitation, the study introduced an adaptive heading angle weight adjustment coefficient designed to mitigate angular variations. This innovation effectively reduced safety risks while simultaneously improving trajectory smoothness. The adaptive weighting coefficients were mathematically defined as follows in Eqs. (27) and (28) respectively,

$$\text{heading1}(vi, \omega\psi, i, \omega\theta, i) = 1 - \frac{|\Delta\psi|}{\pi} \tag{27}$$

$$\text{heading2}(vi, \omega\psi, i, \omega\theta, i) = 1 - \frac{|\Delta\theta|}{\pi} \tag{28}$$

$$k11 = \frac{1}{1 + |\Delta\psi|} \tag{29}$$

$$k12 = \frac{1}{1 + |\Delta\theta|} \tag{30}$$

In the equations: $\Delta\psi$ and $\Delta\theta$ denoted the variation in yaw angle and pitch angle between consecutive time steps, respectively.

(2) Adaptive obstacle distance evaluation.

The conventional Dynamic Window Approach (DWA) algorithm evaluated obstacles primarily based on the nearest obstacle distance using fixed weight coefficients. However, since obstacle avoidance priorities could vary under different conditions, these fixed weights often failed to meet actual mission requirements across various scenarios. To address this limitation, the study proposed a dynamic weight coefficient method that incorporated both the number of obstacles within the risk range and the minimum obstacle distance, enabling more precise performance optimization. The adaptive obstacle distance weighting coefficient was defined in Eq. (31).

$$k2 = \frac{N + 1}{1 + d\min} \tag{31}$$

In the equation: N represented the number of obstacles within the defined risk radius $2R$, while $d\min$ denoted the minimum obstacle distance from the UAV. Based on Eqs. (30) and (31), the adaptive trajectory evaluation function was set as shown in Eq. (32),

$$\text{f}(vi, \boldsymbol{\omega} i) = \alpha 1 \frac{k11 heading1(vi, \boldsymbol{\omega} i) + k12 heading2(vi, \boldsymbol{\omega} i)}{\sum_{i}^{n} heading(vi, \boldsymbol{\omega} i)} + k2\alpha 2 \frac{\text{dist}(vi, \boldsymbol{\omega} i)}{\sum_{i}^{n} \text{dist}(vi, \boldsymbol{\omega} i)} + \alpha 3 \frac{\text{velocity}(vi)}{\sum_{i}^{n} \text{velocity}(vi)} \tag{32}$$

In summary, this study implemented both mechanistic innovations and performance improvements to the DWA algorithm. The introduction of a bidirectional search strategy enhanced the global optimization capability and search efficiency of path planning. For performance enhancement, the implementation of dynamic time-stepping and adaptive evaluation functions simultaneously improved trajectory quality, accelerated search speed, and increased environmental adaptability. These improvements addressed distinct algorithmic performance aspects while remaining theoretically compatible. The integration of these three enhanced methods yielded a comprehensively improved DWA algorithm, demonstrating superior overall performance and enabling reliable operation in unknown complex environments.

DWA-ORCA fusion strategy

DWA-ORCA fusion

The ORCA algorithm was a widely used distributed collision avoidance approach for multi-agent systems. This method constructed velocity obstacle profiles based on agents' velocity information to establish collision-free velocity spaces within time horizon τ for inter-agent obstacle avoidance. Meanwhile, the DWA algorithm searched for optimal paths within unit time step dt. Both algorithms operated in discrete time intervals and maintained theoretically compatible principles. When synchronized to identical execution time intervals, they exhibited inherent fusion feasibility[27].

In terms of performance characteristics, the DWA algorithm demonstrated superior computational speed and efficiency, though its performance in dynamic obstacle avoidance and inter-agent collision prevention remained limited. In contrast, ORCA excelled in swarm avoidance applications despite higher computational complexity. The fusion of these algorithms combined DWA's rapid velocity search capability with ORCA's strengths in dynamic inter-agent collision avoidance. Furthermore, preliminary velocity pre-screening through DWA significantly reduced ORCA's computational load, thereby maintaining high computational efficiency while improving avoidance reliability and safety.

The improved DWA-ORCA algorithm implemented a dual-layer decision architecture: the DWA layer generated candidate velocities satisfying both kinematic constraints and target-oriented motion, while the ORCA layer enforced multi-agent collision avoidance constraints to produce final safe velocities. This relationship was formally expressed in Eq. (33).

$$\mathbf{u}_{\text{final}} = \text{ORCA}(R(\psi_k, \theta_k) \cdot \mathbf{v}_{\text{DWA}}) \tag{33}$$

In the equation: $\mathbf{v}_{\text{DWA}}$ represented the pre-selected velocity set v^{pref} filtered by the DWA algorithm, and $R(\psi_k, \theta_k)$ denoted the transformation matrix at time step k obtained from Eq. (4), which converted the control input $(v, \omega_\psi, \omega_\theta)$ in the body coordinate frame to the ORCA world coordinate frame control input (v_x, v_y, v_z).

The workflow of the fusion algorithm was illustrated in Fig. 8.

3D DWA-ORCA fusion framework

The fusion algorithm proposed in this study adopted a hierarchical decision-making architecture, which achieved efficient obstacle avoidance for multiple UAVs in unknown complex environments through coordinated optimization between the DWA layer and ORCA layer. The specific workflow was as follows,

(1) DWA Layer (Trajectory Pre-selection Layer): Generated a set of three-dimensional candidate control inputs that satisfied kinematic constraints.

$$\mathbf{u}_{\text{DWA}} = \left\{ (v, \boldsymbol{\omega}) \in \mathbb{R}^3 \;\middle|\; \begin{array}{c} v \in [v_{\min}, v_{\max}] \\ \omega_z \in [\dot{\theta}_{\min}, \dot{\theta}_{\max}] \\ \omega_y \in [\dot{\phi}_{\min}, \dot{\phi}_{\max}] \\ \sqrt{(x(t)-x_{obs})^2 + (y(t)-y_{obs})^2 + (z(t)-z_{obs})^2} \geq r_{safe} \end{array} \right\} \tag{34}$$

In the equation: v represented the resultant velocity of the UAV, with ω_ψ and ω_θ denoting the yaw angular velocity and pitch angular velocity respectively.

(2) ORCA Layer (Collision Avoidance Correction Layer): The ORCA layer performs filtering of candidate velocities through 3D velocity obstacle constraints to generate a set of safe control commands that satisfy multi-UAV cooperative collision avoidance requirements. The operational principle of the ORCA algorithm is illustrated in Fig. 9.

For UAV A and obstacle UAV B, their three-dimensional velocity obstacle domain $\mathbf{VO}^{\tau}_{A|B}$ was defined as,

$$\mathbf{VO}^{\tau}_{A|B} = \left\{ \mathbf{v}_A \in \mathbb{R}^3 | \exists t \in [0, \tau], \parallel (\mathbf{p}_A - \mathbf{p}_B) + (\mathbf{v}_A - \mathbf{v}_B)t \parallel < r_A + r_B \right\} \Big\} \tag{35}$$

In the equation: $\mathbf{p}_A, \mathbf{p}_B$ represented the position vectors of UAV A and B, $\mathbf{v}_A, \mathbf{v}_B$ denoted the velocity vectors of UAV A and B, and r_A, r_B corresponded to the safety radii of UAV A and B.

(A) Computation of ORCA constraint plane.

The tangent plane approximation of the VO boundary generated a half-space constraint, forming the inequality constraint:

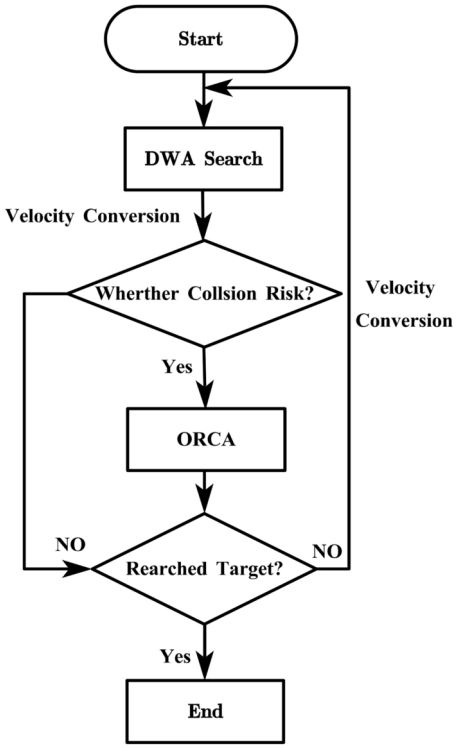


Fig. 8. The flowchart of the fusion algorithm.

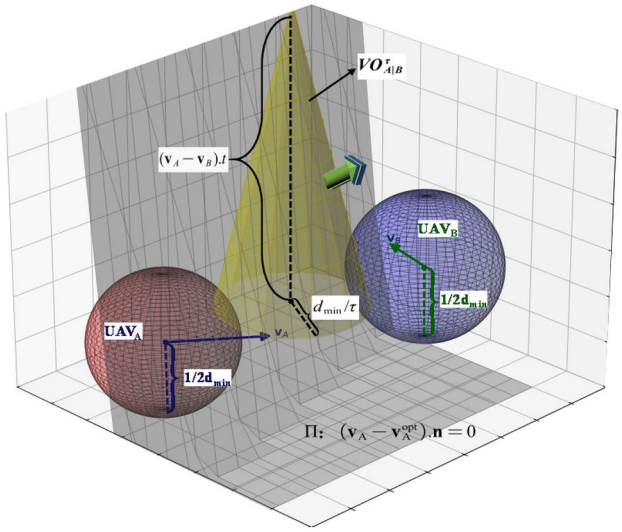


Fig. 9. Geometric Solution of Velocity Obstacle Domain in ORCA Algorithm.

$$\Pi: \quad (\mathbf{v}_{\mathrm{A}}-\mathbf{v}_{\mathrm{A}}^{\mathrm{opt}}).\mathbf{n} \geq \frac{d_{\min}}{2\tau} \tag{36}$$

(B) Calculation of the ORCA constraint plane's normal vector and optimal reciprocal velocity.

The normal vector n of the ORCA constraint plane was the key parameter ensuring coordinated obstacle avoidance with minimal velocity adjustments for UAVs. Through position prediction and directional orthogonalization processing, the tangent plane normal vector was derived as,

$$n = \frac{(\mathbf{p}_B-\mathbf{p}_A)\times(\mathbf{v}_B-\mathbf{v}_A)}{\|(\mathbf{p}_B-\mathbf{p}_A)\times(\mathbf{v}_B-\mathbf{v}_A)\|} \tag{37}$$

The optimal reciprocal velocity was obtained by minimizing the sum of squared velocity adjustments for both avoiding agents using Lagrange's method, yielding,

$$\mathbf{v}_A^{\text{opt}} = \mathbf{v}_A^{\text{pref}} + \omega \left(\frac{\mathbf{p}_A - \mathbf{p}_B}{\tau} - (\mathbf{v}_B - \mathbf{v}_A) \right) \tag{38}$$

In the equation: ω represented the responsibility coefficient, typically set to $\omega = 1/2$ to ensure symmetric adjustments and minimal total cost between both agents.

$$\boldsymbol{v}* = \boldsymbol{v}^{pref} - \frac{\left(v^{pref} \cdot n - \frac{d_{\min}}{2\tau}\right)}{\|n\|^2} n \tag{39}$$

For UAV A to reach the boundary of the ORCA constraint plane and exit the velocity obstacle region, it needed to attain velocity v_A^{opt}. This required a vector correction to its current velocity v_A^{cur}, where the minimum velocity projection correction was defined as,

$$\Delta \mathbf{v}_{\mathrm{A}_{\min}} = \mathbf{v}_{\mathrm{A}}^{new} - \mathbf{v}_{\mathrm{A}} = -\frac{\left((\mathbf{v}_B - \mathbf{v}_A) - \mathbf{v}_A^{\text{opt}}\right) \cdot \boldsymbol{n}}{\|n\|^2} \boldsymbol{n} \tag{40}$$

(C) Inverse Transformation (ORCA Control Input → DWA Control Input).

The velocity correction vector in the world coordinate system was converted to body-frame control inputs through pseudo-inverse computation,

$$\Delta \mathbf{u}_{\min} = J^{\dagger} \Delta \mathbf{v}_{\mathrm{A}_{\min}} \tag{41}$$

The Jacobian matrix was defined as,

$$J = \begin{bmatrix} \cos\psi\cos\theta & -v\sin\psi\cos\theta \cdot \Delta t & -v\cos\psi\sin\theta \cdot \Delta t \\ \sin\psi\cos\theta & v\cos\psi\cos\theta \cdot \Delta t & -v\sin\psi\sin\theta \cdot \Delta t \\ \sin\theta & 0 & v\cos\theta \cdot \Delta t \end{bmatrix} \tag{42}$$

(D) Optimal Control Input Selection.

After completing the generation of candidate control inputs in the DWA layer and velocity corrections in the ORCA layer, the algorithm selected the optimal control input $\Delta \mathbf{u}_{\min}$ by balancing the path planning capability of DWA and the inter-agent obstacle avoidance capability of ORCA through a multi-objective optimization function based on the minimum velocity projection correction $\mathbf{u}*$. The relevant formulation was expressed in Eq. (43),

$$u^* = \arg \min_{u \in V_{safe}} (\| u - u_{\mathrm{DWA}}^* \| + \| \Delta \boldsymbol{v} \|) \tag{43}$$

In the equation: u_{DWA}^* represented the original optimal control input from the DWA layer; $\| u - u_{\mathrm{DWA}}^* \|$ quantified the deviation degree of the control input from the DWA-planned path; $\| \Delta \boldsymbol{v} \|$ denoted the velocity adjustment induced by ORCA corrections.

DWA-ORCA fusion algorithm pseudocode

Improved fusion algorithm design

In multi-UAV cooperative missions, inter-vehicle collision avoidance presented one of the key challenges. To enhance inter-vehicle obstacle avoidance performance in multi-agent cooperative modes, this study developed an improved DWA-ORCA fusion strategy based on both the comprehensive DWA algorithm improvements from Chapter 2 and the DWA-ORCA fusion approach presented in section "Dynamic window approach algorithm". This strategy established an efficient autonomous obstacle avoidance algorithm suitable for multiple UAVs operating in unknown complex environments (Table 2).

The design architecture of the improved DWA-ORCA fusion algorithm was shown in Fig. 10. This architecture integrated the enhanced DWA algorithm as its foundation with ORCA's distributed collision avoidance mechanism, achieving efficient autonomous obstacle avoidance for multiple UAVs in unknown complex environments. Specifically, the improved DWA algorithm generated preliminary flight trajectories, while the ORCA algorithm further optimized and adjusted these trajectories to maintain safe inter-UAV distances and prevent collisions.

Simulation analysis

To verify the effectiveness of the DWA improvement method, DWA-ORCA fusion strategy, and improved DWA-ORCA fusion algorithm, this study conducted comparative simulation experiments in MATLAB 2019b, evaluating four improvement approaches alongside the improved DWA-ORCA fusion algorithm. The test subject was a small quadrotor UAV, selected for its high maneuverability and strong adaptability—characteristics that make it widely applicable in logistics delivery, agricultural monitoring, aerial photography, and particularly

Input: Current state x=(x,y,z,ψ,θ), Goal g, Obstacles O, Other UAVs U
Output: Safe control command u*=[v, ω_ψ, ω_θ]
// DWA Layer: Velocity Space Generation
V_dyn ← Calculate dynamic velocity window based on kinematic constraints
for each (v, ω_ψ, ω_θ) in V_dyn do
  τ ← Predict trajectory using kinematic model with time step Δt
  G_heading ← Evaluate heading alignment with target direction
  G_obs ← Evaluate obstacle proximity score
  if G_obs exceeds safety threshold then retain candidate (u_candidate, τ)
// ORCA Layer: Multi-agent Collision Avoidance
for each u_candidate in candidate set do
  R = rotation_matrix(ψ, θ)
  for each neighboring UAV_j in U do
    Compute velocity obstacle region VO for UAV pair
    if current velocity lies within VO then
      Calculate normal vector to VO boundary
      Project velocity to safe region boundary
  Convert projected 3D velocity to control commands [v, ω_ψ, ω_θ]
// Optimal Control Selection
u* ← Select control minimizing deviation from candidate set
return u*

Table 2. Pseudo-code for the DWA-ORCA Algorithm.

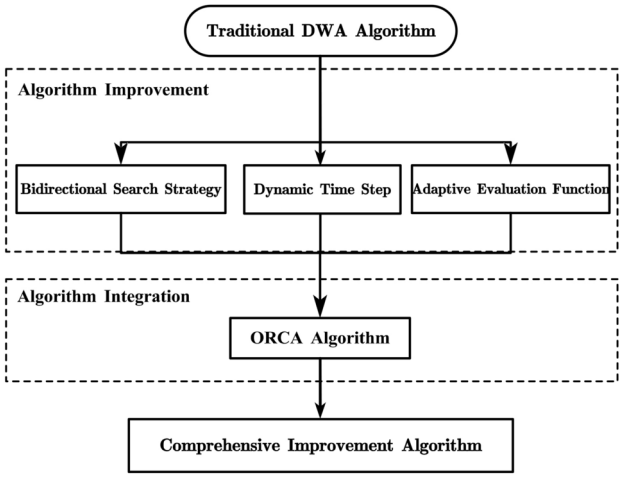


Fig. 10. The design framework of an improved fusion algorithm based on DWA.

suitable for precision operations in complex environments. The UAV parameters were configured according to references[8] and[11], as detailed in Table 3. Algorithm parameters were set as shown in Table 4. For the weighting coefficients—Heading Weight, Avoidance Weight, and Velocity Weight—typical ranges were (0.01 ~ 0.12), (0.1 ~ 0.3), and (0.1 ~ 0.5) respectively. Based on established DWA algorithm parameter settings for UAV

Parameter	Value	Unit
Flight acceleration	2	(m s−2)
Heading angular acceleration	10	(m s−1)
Max-flight velocity	0.7	(m s−1)
Max-Yaw angular acceleration	3	(rad s−1)
Rotor spacing	0.5	(m)

Table 3. Operating parameters of unmanned aerial vehicles.

Parameter	Value	Unit
Obstacle radius	0.5	(m)
Default time step	0.1	(s)
Velocity resolution	5	(m s−1)
Heading resolution	6	(rad s−1)
Avoidan safety radius	1	(m)
ORCA algorithm range	5	(m)
Heading weight	0.05	/
Avoidance weight	0.2	/
Velocity weight	0.1	/

Table 4. Algorithm parameters.

path planning simulations from reference[28], and considering both UAV specifications and obstacle scenarios, Heading Weight and Avoidance Weight were set to intermediate values within their ranges (0.05 and 0.2). Given the UAV's relatively low operating speed during simulations, Velocity Weight was assigned the lower bound value of 0.1.

Validation of improved DWA algorithm

At the single-UAV experimental level, for systematic evaluation of the performance optimization effects of the improved Dynamic Window Approach (DWA) algorithm, this study established an experimental framework using random sampling methods. The experimental environment incorporated randomly generated obstacle distributions, where initial positions were uniformly sampled within a spherical region centered at (0,0,0) (unit: meters) with a radius of 0.5 m, and target positions were similarly sampled within a spherical region centered at (10,10,10) under identical spatial constraints. This design ensured spatial uniformity of test samples across scenarios. Through repeated randomized trials, 10 valid experimental datasets were acquired. Statistical analysis based on the Law of Large Numbers was conducted to compute average performance metrics for single-UAV operations during individual task executions.

The bidirectional search strategy

Comparative simulation experiments were conducted between the conventional DWA algorithm and the bidirectional DWA algorithm to verify the effectiveness of the bidirectional search improvement. Figure 11 presented the comparative simulation results of both algorithms, while Table 5 provided the average performance metrics from sampled experiments evaluating the bidirectional search enhancement.

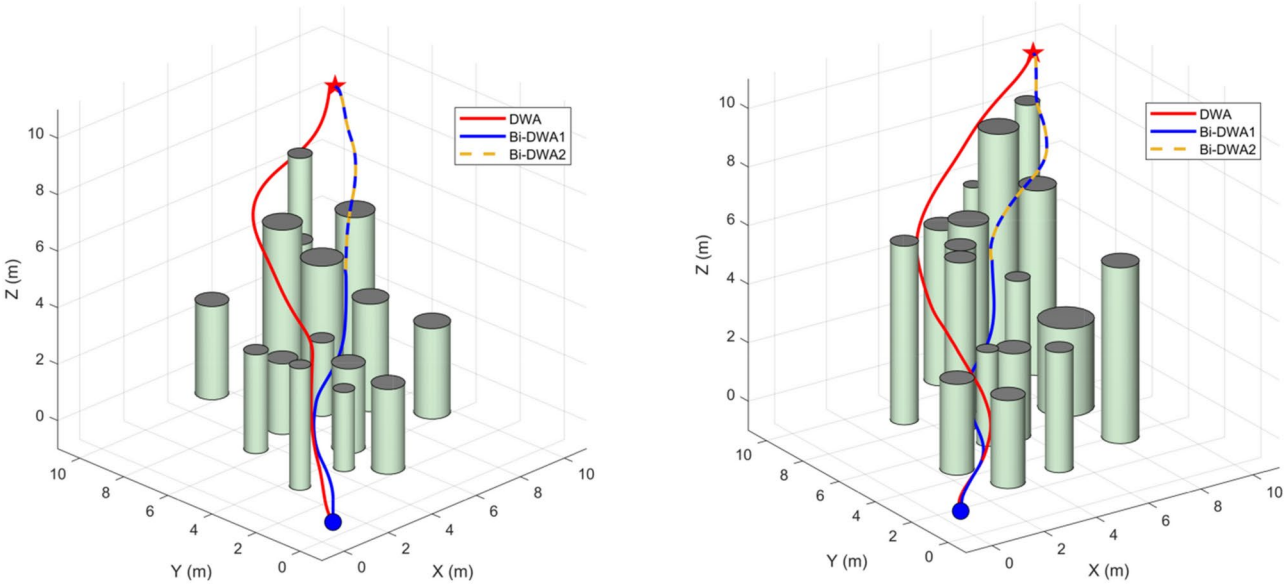
The comparison between the conventional DWA algorithm and the bidirectional DWA algorithm showed that the introduction of the bidirectional search strategy significantly improved the global value of the planned paths. Compared to the traditional algorithm, the improved version achieved an average reduction of 3.03 m in path length, 8.57 s in mission time, and 115.2 iterations.

The dynamic time step

To verify the improvement effect of the dynamic time step strategy, the optimal time step curve was applied to the DWA algorithm for simulation experiments. The heading angle variations under different time steps were shown in Fig. 12, where the red curve represented the heading angle change corresponding to the dynamic time step setting.

The comparative performance of the algorithm under different time steps was shown in Table 6.

The simulation analysis comparing dynamic and fixed time steps in the DWA algorithm revealed that when the time step was set to 0.05 s, the heading angle variation was minimized, but at the cost of significantly increased mission completion time and iteration counts. As the time step increased, the number of iterations decreased, but this exacerbated heading angle changes and prolonged mission duration. Consequently, fixed time steps failed to ensure balanced algorithm performance. In contrast, the dynamic time step approach not only met computational requirements for iteration efficiency but also maintained optimal mission completion time and heading angle stability, resulting in substantial overall performance improvement. The dynamic time step enhancement strategy significantly boosted the algorithm's comprehensive performance.



(a) simulation scenario 1 (b) simulation scenario 2

Fig. 11. Simulation example diagram of bidirectional search strategy comparison.

Algorithm	Traditional DWA algorithm	Bidirectional DWA algorithm
Average distance (m)	24.65	21.62
Average time (s)	56.78	48.21
Average iterations	510.70	340.00

Table 5. Comparison of average performance of bidirectional improvement algorithms.

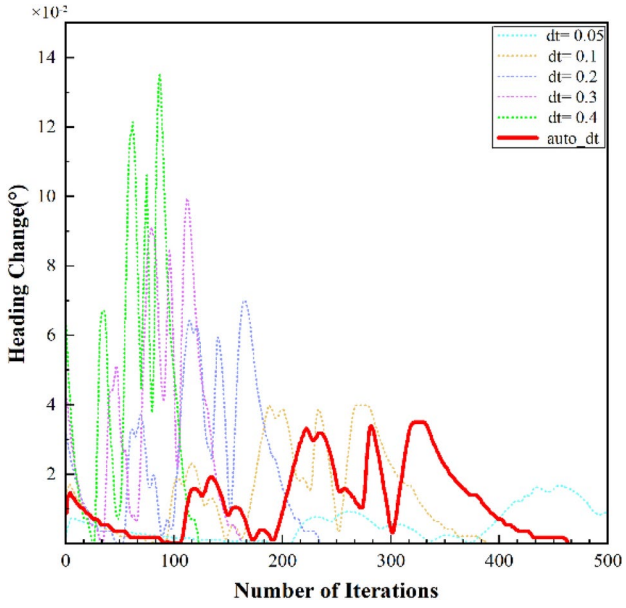
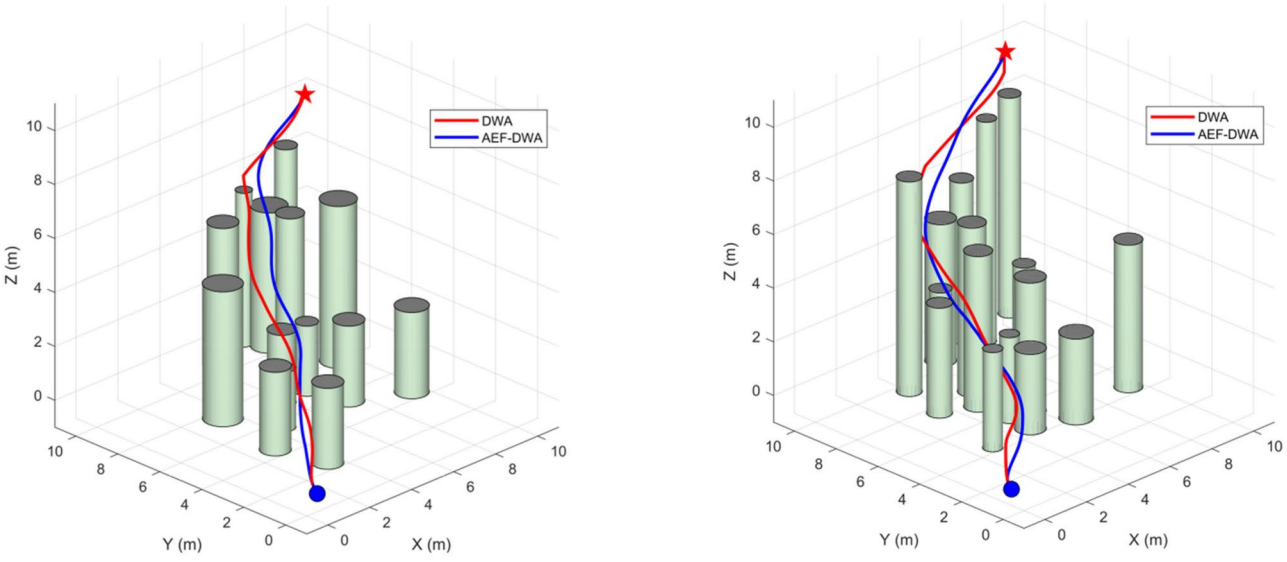


Fig. 12. Changes in heading angle at different time steps.

Time step (dt)	Total time (s)	Iterations count	Average heading change (°)
0.05	40.54	793	0.007
0.1	39.32	387	0.019
0.2	39.51	139	0.038
0.3	42.37	102	0.044
0.4	43.24	81	0.059
Auto-dt	38.61	479	0.017

Table 6. Performance comparison of improvements with dynamic time steps.



(a) simulation scenario 1 (b) simulation scenario 2

Fig. 13. Adaptive evaluation function comparative simulation experiment.

Algorithm	Traditional DWA	Adaptive evaluation DWA
Average distance (m)	21.08	20.06
Minimum obstacle distance (m)	0.47	0.38
Average minimum obstacle distance (m)	0.65	0.27

Table 7. Performance comparison of improvements with dynamic time steps.

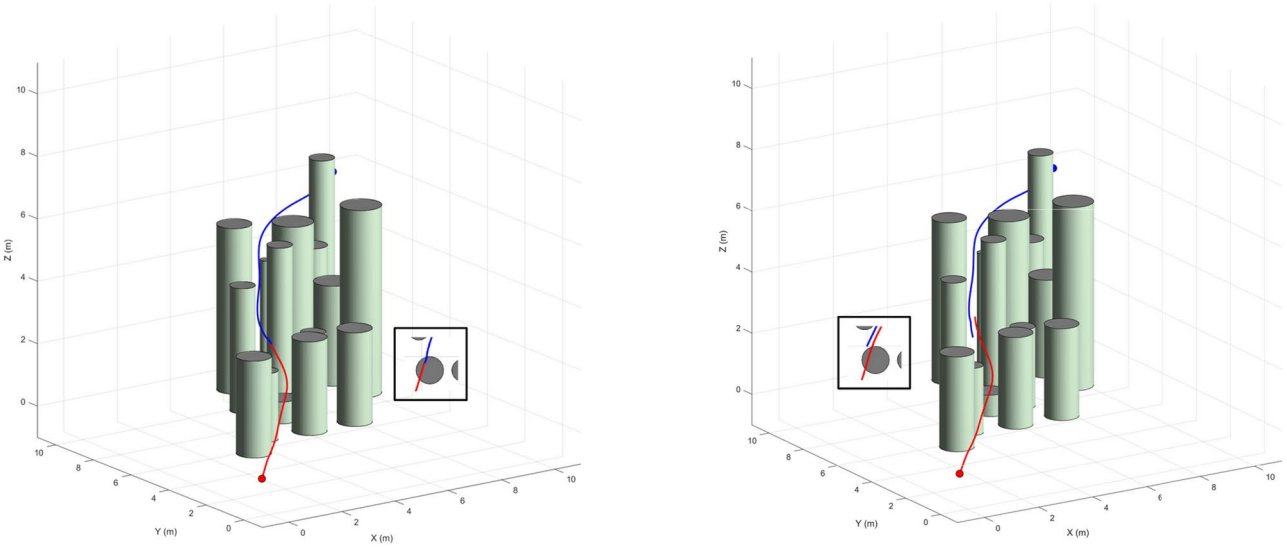
The adaptive evaluation function

To verify the improvement effect of the adaptive evaluation DWA algorithm, comparative simulation experiments were conducted between the conventional DWA algorithm and the adaptive evaluation DWA algorithm. Figure 13 showed the comparative simulation examples of both algorithms, while Table 7 presented the average performance comparison from sampled experiments.

The simulation comparison between the adaptive evaluation DWA algorithm and the conventional DWA algorithm in Table 7 showed that the adaptive evaluation DWA algorithm significantly improved environmental adaptability. Compared with the traditional algorithm, the improved algorithm reduced the average path length by 1.02 m, decreased the minimum obstacle spacing by 0.09 m, and reduced the average minimum obstacle spacing by 0.40 m.

Validation of DWA-ORCA fusion algorithm

The DWA-ORCA fusion strategy was validated for inter-UAV collision avoidance performance through 10 experimental trials conducted under the same randomized simulation conditions described in section "DWA-ORCA fusion". The collision avoidance success rates of both algorithms were statistically compared. Figure 14 showed comparative simulation examples between the conventional DWA algorithm and the DWA-ORCA



(a) simulation scenario 1 (b) simulation scenario 2

Fig. 14. Comparative simulation example diagram of DWA-ORCA fusion algorithm.

Algorithm	Traditional DWA	DWA-ORCA
Avoidance count	6	10
Collision count	4	0
Success rate	60%	100%

Table 8. Comparison of Inter-UAV collision avoidance results using DWA-ORCA fusion algorithm.

fusion algorithm, while Table 8 presented the inter-UAV avoidance performance comparison between the two approaches.

The comparison of obstacle avoidance results in Table 8 demonstrated that the DWA-ORCA algorithm could more effectively predict and execute inter-UAV avoidance maneuvers based on real-time operational states of the UAVs. Compared to the conventional DWA algorithm, the DWA-ORCA fusion algorithm achieved a 40% improvement in inter-UAV avoidance success rate.

Validation of improved DWA-ORCA fusion algorithm

To further validate the effectiveness of the enhanced DWA-ORCA hybrid algorithm, we conducted simulation experiments using two typical multi-UAV operational scenarios: terrain search and convergence operations, as specified in [34]. Comparative analyses were performed between the baseline DWA algorithm and the improved DWA-ORCA fusion algorithm, with the Artificial Potential Field (APF) method serving as the comparative baseline. In randomly generated obstacle environments, each mission type was executed five times, resulting in 10 experimental groups. Representative simulation results are illustrated in Figs. 15 and 16, while average performance metrics under both mission conditions and comprehensive evaluations are quantitatively presented in Tables 9, 10, and 11 respectively.

Aggregation Task: Three initial positions were randomly generated within the map, while three target points were randomly distributed within a spherical region centered at (5,5,5) (unit: meters) with a radius of 1 m. Three UAVs executed the aggregation task to converge toward the designated targets.

Target Search Task: To ensure sufficient path length for comprehensive performance evaluation, three UAV departure points were uniformly sampled within spherical regions centered at (0,0,0), (2,0,0), and (10,0,0) (unit: meters), each with radius $r = 0.5$ m. Correspondingly, three target points were generated within spherical regions at (10,10,0), (0,10,0), and (6,10,0) under identical spatial constraints. The UAVs executed the search task to navigate between these geographically distributed waypoints.

Experimental results demonstrate that the conventional Dynamic Window Approach (DWA) algorithm exhibits superior path planning stability compared to the Artificial Potential Field (APF) method. The APF algorithm, operating through potential field mechanisms, demonstrates susceptibility to local minima entrapment, resulting in path oscillations and frequent trajectory adjustments that become particularly pronounced in narrow passages and dense obstacle environments. These instability phenomena may elevate

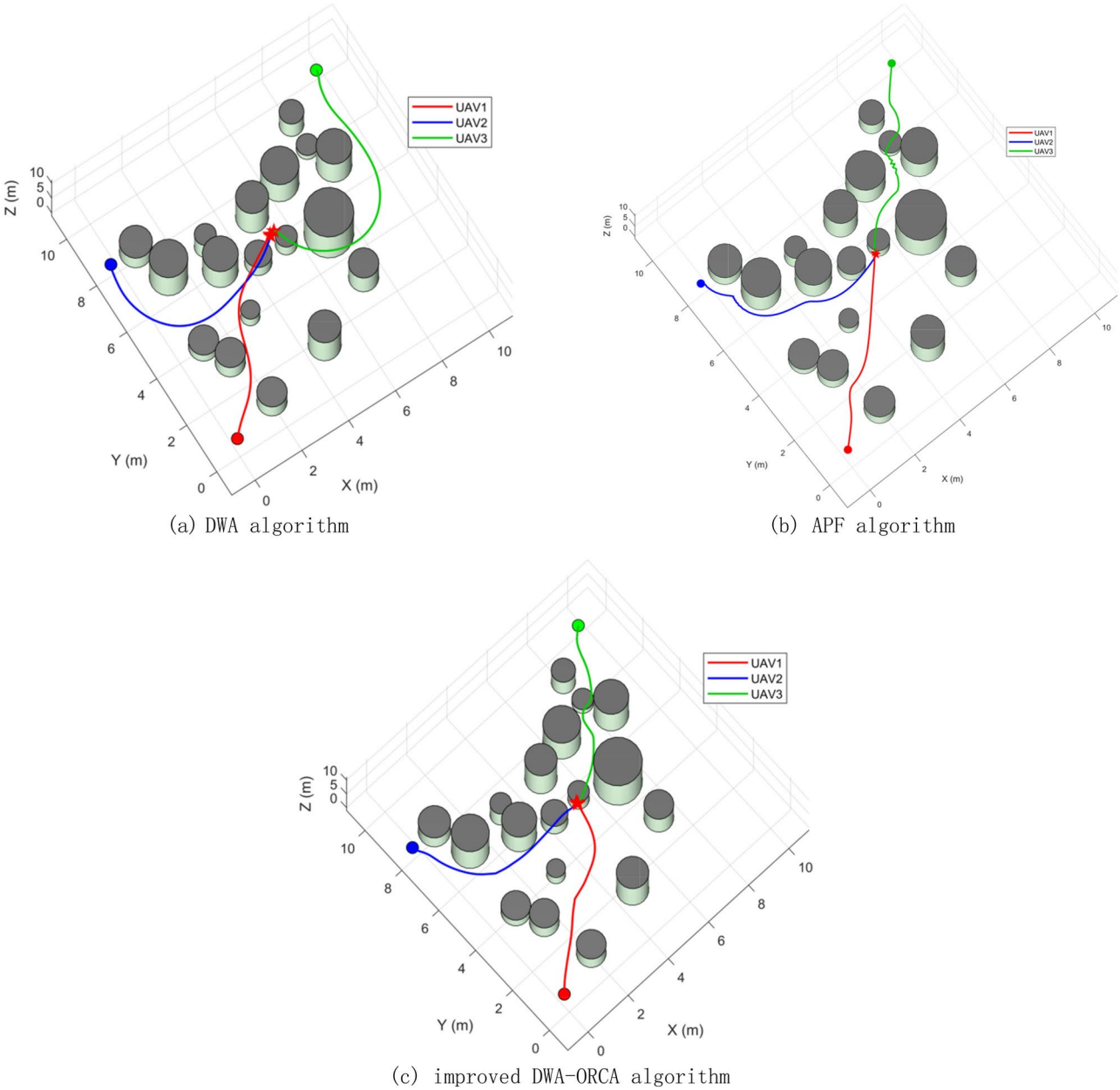


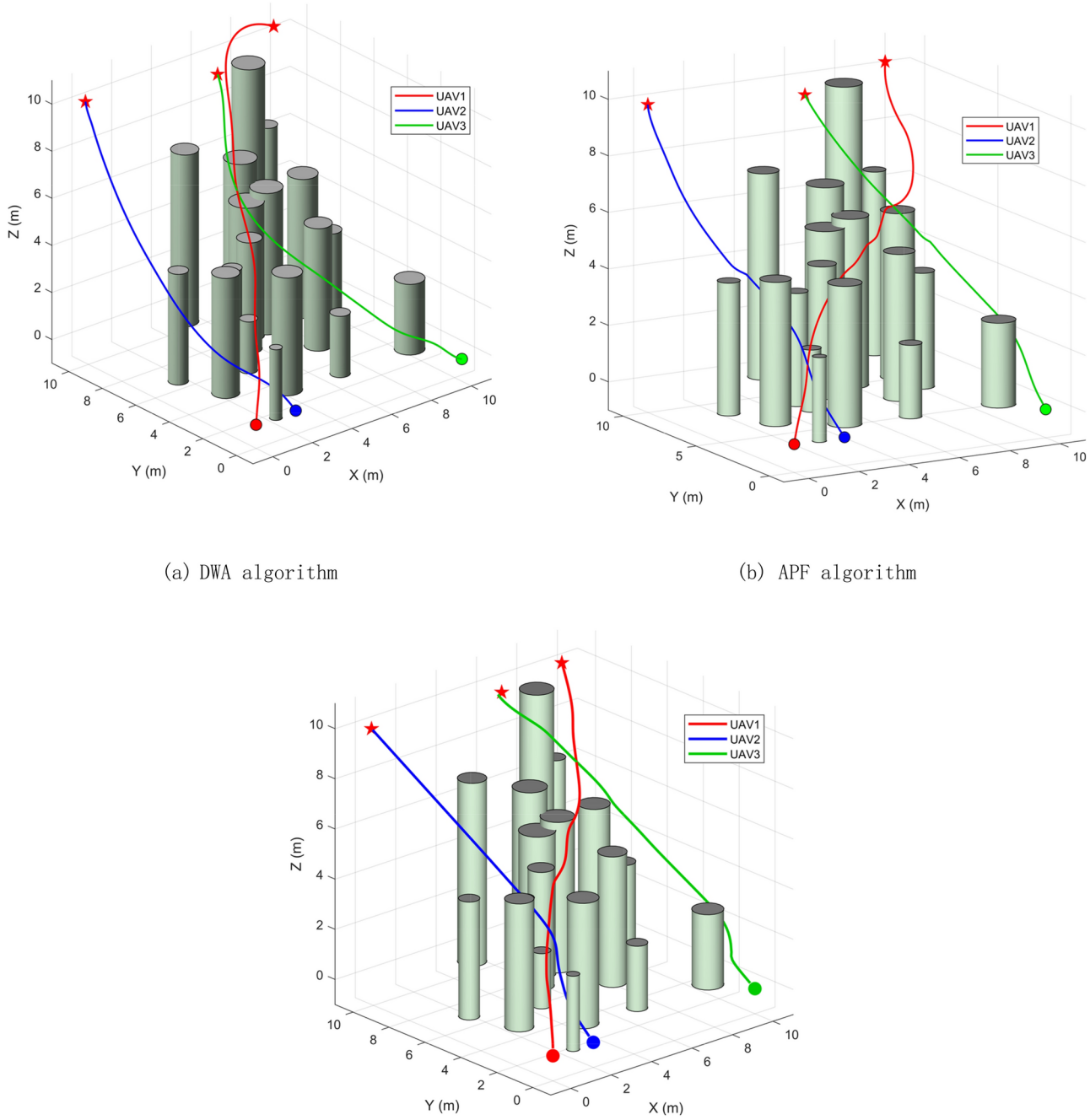
Fig. 15. Convergence task simulation experiment example.

collision risks while compromising path planning efficiency. Notably, the APF algorithm achieves merely 10% obstacle avoidance success rate in complex environments, significantly lower than DWA's 20% performance.

In contrast, the DWA algorithm employs dynamic window-based velocity and orientation adjustments to circumvent the local minima problem inherent in APF approaches, generating smoother and more stable trajectories. However, DWA still faces computational complexity challenges in obstacle-dense environments, with each planning cycle consuming 286 ms of computation time, 58 MB memory allocation, and 23% CPU utilization, thereby constraining real-time performance.

The enhanced DWA-ORCA hybrid algorithm maintains DWA's path stability while significantly improving planning efficiency and obstacle avoidance capabilities. Key enhancements include:

(1) Path Quality: Optimization of global cost functions reduces average path length by 27.9% (from 19.91 to 14.35 m) while achieving 100% obstacle avoidance success rate. Notably, the improved algorithm employs hierarchical decision-making architecture and bidirectional search strategy to substantially reduce computational redundancy. The DWA pre-screening layer compresses candidate velocity space significantly, while a conditional triggering mechanism activates ORCA-based collision avoidance only when inter-robot distances fall below 2 m. This design eliminates continuous global computation, maintaining lower computational load than standard DWA.



(a) DWA algorithm

(b) APF algorithm

(c) improved DWA-ORCA algorithm

Fig. 16. Search and localization task simulation experiment example.

Task	Algorithm	DWA	APF	Improved DWA-ORCA
Convergence	Average Distance(m)	16.91	14.23	11.92
	Average Time(s)	37.33	34.62	30.49
	Average Iterations	432.63	442.68	331.25
	Collision Occurred?	Yes	Yes	No

Table 9. Comparison of average performance of different algorithms for convergence simulation T.

Task	Algorithm	DWA	APF	Improved DWA-ORCA
Search and localization	Average distance (m)	22.91	19.27	16.77
	Average time (s)	49.93	46.20	41.92
	Average iterations	578.60	583.40	462.10
	Collision occurred?	Yes	Yes	No

Table 10. Comparison of average performance of different algorithms for search and localization simulation tasks.

Algorithm	DWA	APF	Improved DWA-ORCA
Average distance (m)	19.91	16.75	14.35
Average time (s)	43.63	40.41	36.21
Average iterations	505.60	513.05	396.70
Avoidance success	20%	10%	100%
Memory usage (MB)	58	32	41
Average planning time per instance (ms)	56.46	36.95	48.75
CPU load (%)	23	16	18

Table 11. Comparison of average performance of algorithms in comprehensive experiments.

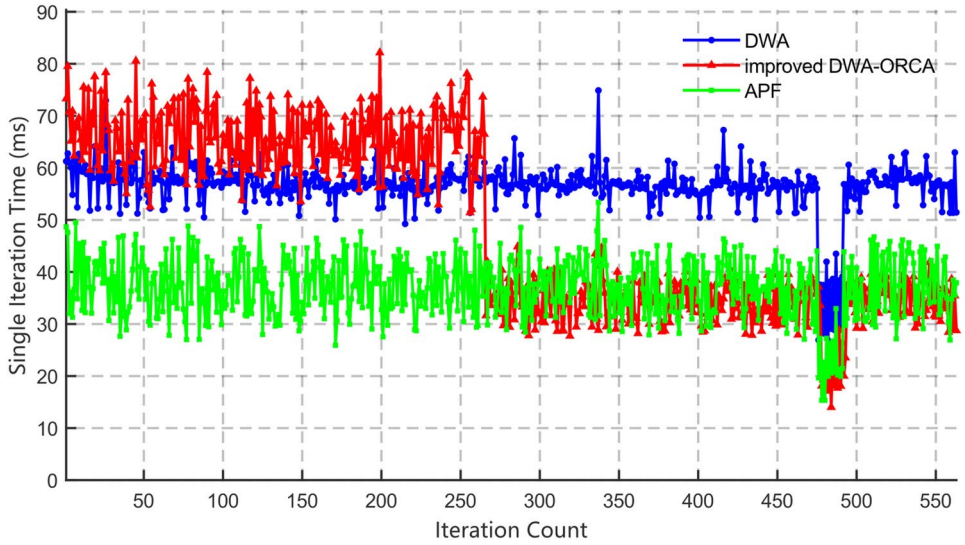


Fig. 17. Algorithm iteration time curve chart.

(2) Resource Efficiency: As shown in the algorithm iteration time curve chart (Fig. 17), despite the longer initial iteration time, the average single—cycle planning time of the improved DWA-ORCA fusion algorithm drops to 48.75 ms once the bidirectional search mechanism is triggered. This represents a 13.6% improvement over the traditional DWA (56.46 ms) and is well below the 200—ms PX4 flight control threshold. Meanwhile, memory consumption decreases from 58 to 41 MB (a 29.3% reduction) and CPU load falls from 23 to 18%. Although planning time is 31.9% higher than that of APF (36.95 ms), this is justified as the fusion algorithm delivers much better success rates and path stability, enhancing autonomous navigation in complex environments.

Conclusion

This study presents an enhanced DWA-ORCA fusion algorithm developed to address the challenge of autonomous obstacle avoidance for multiple drones operating in unknown and complex environments. The algorithm incorporates a bidirectional search strategy, which significantly improves path planning efficiency by broadening the search horizon and effectively reducing the average path length and task completion time. The dynamic time step design, optimized through a PSO-RSM methodology, adaptively adjusts computational resource allocation in response to environmental complexity. This ensures high trajectory smoothness while

satisfying the real-time requirements of the PX4 flight control system. Furthermore, an adaptive evaluation function is utilized to dynamically balance heading and obstacle avoidance priorities, resulting in a substantial enhancement of obstacle avoidance performance in dense obstacle scenarios, including a reduction in the minimum safety distance. The hierarchical DWA-ORCA fusion strategy also considerably improves inter-drone obstacle avoidance by integrating local planning efficiency with robust collision avoidance capabilities. Collectively, these advancements ensure that the algorithm provides a reliable technical solution for the autonomous navigation of multiple drones in complex and unfamiliar settings.

Future Outlook: The improved DWA-ORCA fusion algorithm proposed in this study demonstrates good scalability in applications involving complex environments and larger-scale UAV formations. By employing a hierarchical architecture, the algorithm decouples global path planning from local obstacle avoidance, reducing system coupling and enhancing scalability. The dynamic time step design adjusts computational resource allocation based on environmental complexity, avoiding performance bottlenecks caused by fixed time steps. Additionally, a distributed computing architecture supports multi-agent collaboration, reducing communication overhead through a sparse communication network and ensuring efficient operation of large-scale formations. Future research will further optimize communication efficiency and computational resource allocation, explore collaboration with heterogeneous systems, and enable multimodal mission execution, driving the application of the algorithm in larger-scale formations and complex environments.

Data availability

The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Received: 12 February 2025; Accepted: 16 April 2025
Published online: 26 April 2025

References

1. Yusuf, S. M. et al. Towards autonomous multi-UAVs surveillance mission: a study of Nigerian telecommunication masts surveillance. In *Proceedings of 2021 1st International Conference on Multidisciplinary Engineering and Applied Science (ICMEAS). Abuja, Nigeria: IEEE* 1–6 (2021).
2. Tejani, G. G., Bhensdadia, V. H. & Bureerat, S. Examination of three meta-heuristic algorithms for optimal design of planar steel frames. *Adv. Comput. Design* **1**(1), 79–90 (2016).
3. Sun, G. et al. V2V routing in a VANET based on the autoregressive integrated moving average model. *IEEE Trans. Veh. Technol.* **68**(4), 3666–3676 (2019).
4. Alhussen, A. et al. Early attention-deficit/hyperactivity disorder (ADHD) with NeuroDCT-ICA and rhinofish optimization (RFO) algorithm based optimized ADHD-AttentionNet. *Sci. Rep.* **15**, 1–12 (2025).
5. Mohamed, A. A., Al-Saleh, A., Sharma, S. K. & Tejani, G. G. Zero-day exploits detection with adaptive WavePCA-Autoencoder (AWPA) adaptive hybrid exploit detection network (AHEDNet). *Sci. Rep.* **15**, 1–15 (2025).
6. XiaoWei, Fu., Zhi, C. Y. & Di, W. Obstacle avoidance and collision avoidance of UAV swarm based on improved VFH algorithm and information sharing strategy. *Comput. Ind. Eng.* **186**, 109761 (2023).
7. Sun, G. et al. Bus-trajectory-based street-centric routing for message delivery in urban vehicular ad hoc networks. *IEEE Trans. Veh. Technol.* **67**(8), 7550–7563 (2018).
8. Çelik, E. et al. Reconfigured single- and double-diode models for improved modelling of solar cells/modules. *Sci. Rep.* **15**, 1–18 (2025).
9. Mohamed, A. A., Al-Saleh, A., Sharma, S. K. & Tejani, G. G. Zero-day exploits detection with adaptive WavePCA-Autoencoder (AWPA) adaptive hybrid exploit detection netwoArk (AHEDNet). *Sci. Rep.* **15**, 1–15 (2025).
10. Tejani, G. G., Mashru, N., Patel, P., Sharma, S. K. & Çelik, E. Application of the 2-archive multi-objective cuckoo search algorithm for structure optimization. *Sci. Rep.* **14**, 1–20 (2024).
11. Sun, G., Wang, Y., Yu, H. & Guizani, M. Proportional fairness-aware task scheduling in space-air-ground integrated networks. *IEEE Trans. Serv. Comput.* **17**(5), 1–14 (2024).
12. Liang, C., Liu, L. & Liu, C. Multi-UAV autonomous collision avoidance based on PPO-GIC algorithm with CNN–LSTM fusion network. *Neural Netw.* **162**, 21–33 (2023).
13. Ma, N., Meng, J., Luo, J. & Liu, Q. Optimization of thermal-fluid-structure coupling for variable-span inflatable wings considering case correlation. *Aerosp. Sci. Technol.* **150**, 109448 (2024).
14. Wu, H. et al. A novel fuzzy control path planning algorithm for intelligent ship based on scale factors. *J. Supercomput.* **80**, 202–225 (2024).
15. Wang, X., Cheng, M., Zhang, S. & Gong, H. Multi-UAV cooperative obstacle avoidance of 3D vector field histogram plus and dynamic window approach. *Drones* **7**, 504 (2023).
16. Xiong, B. et al. UAV path planning based on improved A and DWA algorithms. *Int. J. Aerospace Eng.* **2021**, 56 (2021).
17. Wang, S., Li, G. & Liu, B. Research on unmanned vehicle path planning based on the fusion of an improved rapidly exploring random tree algorithm and an improved dynamic window approach algorithm. *World Electr. Veh. J.* **15**, 292 (2024).
18. Pamarthi, V. & Richa, A. Enhancing UAV navigation with dynamic programming and hybrid probabilistic route mapping: an improved dynamic window approach. *Int. J. Inf. Technol.* **16**(2), 1023–1032 (2023).
19. Song, F. F. Research on collaborative collision avoidance decision-making method for multi-intelligent ships based on ORCA algorithm. In *Dalian: Dalian Maritime University* (2023) (in Chinese).
20. Gong, X. et al. A local path planning algorithm for robots based on improved DWA. *Electronics* **13**, 2965 (2024).
21. Fox, D., Burgard, W. & Thruns, S. The dynamic window approach to collision avoidance. *IEEE Robot. Autom. Mag.* **4**(1), 23–33 (1997).
22. Wu, B. et al. Dynamic path planning for Forklift AGV based on smoothing A* and improved DWA hybrid algorithm (2024).
23. Liu, H. & Zhang, Y. ASL-DWA: an improved A-star algorithm for indoor cleaning robots. *IEEE Access* **10**, 99498–99515 (2022).
24. Sang, W., Yue, Y., Zhai, K. & Lin, M. Research on AGV path planning integrating an improved A* algorithm and DWA algorithm. *Appl. Sci.* **14**, 7551 (2024).
25. Jain, M. et al. An overview of variants and advancements of PSO algorithm. *Appl. Sci.* **12**, 8392 (2022).
26. van den Berg, J., Lin, M. C. & Manocha, G. Optimal reciprocal collision avoidance for real-time multi-agent navigation. In *IEEE, Proceedings of 2011 IEEE International Conference on Robotics and Automation (ICRA). IEEE* 388–395 (2011).
27. Li, Y. Research on three-dimensional dynamic path planning method for UAVs based on RRT-DWA algorithm. In *University of Electronic Science and Technology of China* (2024) (in Chinese).

28. Shakeri, R. et al. Design challenges of multi-UAV systems in cyber-physical applications: a comprehensive survey and future directions. *IEEE Commun. Surv. Tutor.* **21**(4), 3340–3385 (2019).

Author contributions

X.C. and J.Y.W. wrote the original draft and made subsequent revisions. K.L., Q.T., and X.Z. contributed to the writing of the original draft as well as reviewing and editing the final manuscript. All authors approved the final version of the manuscript.

Funding

This study was supported by Henan Provincial Science and Technology Research Project (No.242102220053/232102220035) and the Scientific Research Team Plan of Zhengzhou University of Aeronautics (24ZHTD01003).

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to J.W.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by-nc-nd/4.0/.

© The Author(s) 2025