



Research article**A hierarchical risk-aware UAV path-planning framework coupling risk-distance weighted A^* and improved dung beetle optimization****Xiao Ming Wu^{1,*}, Hao Li², Shan Xing Wang², Qi Zhang², Zhi Han Liang² and Su Yu Zhou³**¹ Library, Shandong Jiaotong University, Jinan 250357, Shandong, China² School of Information Science & Electrical Engineering, Shandong Jiaotong University, Jinan 250357, Shandong, China³ Chengdu Free Flight Aerospace Science & Technology Co., Ltd., China*** Correspondence:** Email: wxm755@163.com.

Abstract: Urban unmanned aerial vehicle (UAV) operations face significant risks arising from dense population exposure and vulnerable infrastructure. This paper presents a hierarchical risk-aware path-planning framework, termed RDA*-IDB, which explicitly couples coarse-resolution risk-aware global guidance with high-resolution local trajectory refinement. First, a voxel-based three-dimensional urban risk map is constructed by integrating fatality risk and property damage risk. At the global planning level, a risk distance weighted A^* planner, denoted as RDA*, operates on the coarse-resolution risk map to balance flight distance and risk exposure and then generate a sequence of risk-aware global waypoints. Each pair of adjacent global waypoints is subsequently mapped to a corresponding high-resolution local planning domain, within which an improved dung beetle optimizer refines the trajectory under path length, turning angle, and obstacle feasibility considerations. Tent chaotic initialization and Lévy flight perturbation are incorporated to improve population diversity and reduce premature convergence. Simulations are conducted in a $5\text{ km} \times 5\text{ km}$ urban area of Jinan, China, using the flight parameters of a DJI M210 RTK UAV. Under the considered simulation conditions, the proposed framework achieves the lowest unit-path average risk value among the evaluated methods. However, the improvement over A^* is small, and the longer generated path results in a higher accumulated path risk cost than those obtained by A^* and RRT*. The results therefore reveal a scenario-specific trade-off between average relative risk intensity per unit distance and total path length.

Keywords: unmanned aerial vehicle; hierarchical path planning; urban risk map; risk-distance weighted A^* ; improved dung beetle optimization

Mathematics Subject Classification: 90B25, 90B50

1. Introduction

UAVs have been increasingly investigated for resilient formation mission planning [1] and traffic monitoring using onboard object-detection algorithms [2]. However, the rapid increase in UAV applications has raised substantial concerns regarding operational risks, particularly when numerous UAVs operate within low-altitude urban airspace, making UAV operational safety a critical issue [3]. Consequently, generating low-risk flight paths for UAVs has emerged as an essential requirement for ensuring safe and efficient operations.

Both academia and industry have extensively researched risk assessment approaches for UAV operations. Standardized by the Joint Authorities for Rulemaking on Unmanned Systems (JARUS) [4], the Specific Operations Risk Assessment (SORA) establishes a dual-risk taxonomy: The classification of ground-related risk, the ground risk class (GRC), together with that of air-related risk, the air risk class (ARC), delivering a procedural framework for UAV risk quantification. Paielli and Erzberger from the National Aeronautics and Space Administration (NASA) [5, 6] developed an analytical model to estimate aircraft conflict probabilities for free-flight scenarios. Their approach employs multidimensional Gaussian distributions to model flight error probabilities and linear transformations within a Cartesian coordinate system, thus establishing an analytical foundation for the geometric analysis of UAV collision avoidance. Mueller and Kochenderfer from Stanford University [7] introduced an Airborne Collision Avoidance System X (ACAS-X)-based collision avoidance algorithm designed for multirotor UAVs. By casting conflict resolution within a partially observable Markov decision process (POMDP) framework, dynamic programming is used to optimize UAV evasive maneuvers. Mitici et al. [8] synthesized disparate mathematical models into a consolidated computational paradigm, enabling probabilistic estimation of aerial conflicts and collisions. This advances foundational frameworks for next-generation air traffic management (ATM).

UAV path planning approaches are essential for the safe integration of UAVs into low-altitude urban airspace. Conventional planning techniques include Dijkstra's algorithm [9], the A* algorithm, and the rapidly-exploring random tree (RRT) algorithm [10]. Intelligent optimization algorithms include particle swarm optimization (PSO) [11], the genetic algorithm (GA) [12], the coati optimization algorithm (COA) [13], the sparrow search algorithm (SSA) [14], and ant colony optimization (ACO) [15]. Recent methodological studies have also proposed a soft-thresholding aphid optimization algorithm [16] and a data-driven iterative-learning consensus-control approach for distributed-parameter multiagent systems [17]. Specifically, Peng Wei's group [18] at George Washington University applied Monte Carlo tree search algorithms to compute collision-free, on-demand UAV routes in crowded urban environments. Zeng et al. integrated sequential Latin hypercube sampling [19] with support vector machine and AdaBoost classifiers. This hybrid approach improves classifier training efficiency and helps alleviate local optimum problems in UAV path planning.

Sung et al. [20] comparatively analyzed the computational complexity of various computational intelligence approaches for offline two-dimensional (2D) and three-dimensional (3D) path planning. Hu et al. [21] proposed an event-triggered prescribed-performance control method for multiplayer nonlinear systems by integrating integral reinforcement learning with adaptive dynamic programming, enabling online control-policy learning while reducing unnecessary control updates. Meng et al. [22] investigated the maximization of the downlink sum rate in a nonorthogonal multiple access-based space-air-ground integrated internet of things (IoT) network by jointly optimizing the deployment

locations and transmit powers of high-altitude platforms and UAVs, the bandwidth allocation ratio, and communication link selection. They employed a deep Q-learning algorithm and an alternative optimization algorithm based on differential annealing to solve the resulting joint optimization problem involving both discrete and continuous variables. Chang et al. [23] developed a quantized fuzzy feedback control method for electric vehicle lateral dynamics to address signal quantization and improve closed-loop control performance. Some studies have incorporated risk cost into UAV trajectory planning. Primatesta et al. [24] integrated risk assessment results into an RRT-based planner to generate risk-aware UAV paths in urban environments. Dai et al. [25] proposed a conflict-free A* algorithm to manage flight conflicts and mitigate delays. Rudnick-Cohen et al. [26] employed an RRT-based optimization framework to minimize the third-party risk associated with UAV takeoff trajectories. Pang et al. [27] further incorporated third-party risk into an integrated path-cost assessment model, and Phung and Ha [11] applied particle swarm optimization to safety-constrained UAV path planning. More broadly, UAV path planning in voxelized urban airspace is influenced by network connectivity, environmental constraints, and heterogeneous traversal costs, consistent with the general role of topology and heterogeneity in complex network propagation processes discussed by Ji et al. [28].

Existing path-planning approaches address different aspects of this problem. Weighted A* adjusts the influence of heuristic information to balance search efficiency and solution quality [29], whereas D* is primarily designed for incremental replanning in partially known or changing environments [30]. Risk-aware graph search and sampling-based planners generally incorporate environmental risk, flight conflict, or third-party exposure into the path cost, whereas PSO-based and other swarm optimization planners directly optimize candidate waypoint sequences under multiple constraints.

These studies demonstrate that weighted heuristic search, risk-aware planning, and population-based trajectory optimization have each been investigated in previous work. However, the reviewed approaches do not explicitly couple coarse-resolution, risk-aware global guidance with high-resolution local trajectory refinement through a waypoint-to-domain mapping mechanism. Accordingly, the principal contribution of the present study lies in establishing this multiresolution hierarchical coupling for urban risk-aware UAV path planning. A focused comparison of the representative path-planning approaches and the proposed framework is provided in Table 1.

This paper presents a hierarchical planning framework with embedded risk constraints to balance UAV flight safety and route efficiency. Its principal contribution is the explicit coupling of coarse-resolution, risk-aware global guidance with high-resolution local trajectory refinement across different planning levels. Initially, the coarse global planning phase operates on a 3D urban risk map with a resolution of $100\text{ m} \times 100\text{ m} \times 20\text{ m}$, in which a risk-distance-weighted A* (RDA*) algorithm is employed to evaluate the integrated risk cost and distance cost and generate discrete waypoints. Subsequently, each pair of adjacent global waypoints is mapped to a corresponding local planning domain with a resolution of $1\text{ m} \times 1\text{ m} \times 1\text{ m}$. The two global waypoints determine the local start and goal positions as well as the spatial boundaries of the local optimization domain, thereby transferring the global risk-aware guidance to the high-resolution trajectory refinement stage. Within each local domain, the improved dung beetle optimization (IDB) algorithm incorporates tent chaotic initialization and Lévy flight perturbation to improve initial population diversity and reduce premature convergence. Ultimately, the optimized local trajectories are sequentially concatenated to form the complete UAV path. In this hierarchical framework, risk-aware guidance is provided by the first-level RDA* planner,

and the second-level IDB planner performs fine-resolution geometric trajectory refinement under path length, turning angle, and obstacle feasibility considerations.

Table 1. Focused comparison of representative path-planning approaches and the proposed framework.

Approach category	References	Principal mechanism	Difference from the proposed framework
Weighted A*	[29]	Adjusts the influence of heuristic information to balance search efficiency and solution quality.	Does not independently provide a voxel-based urban risk representation or a coarse-to-fine local trajectory-refinement mechanism.
D*	[30]	Incrementally repairs a path when the known environmental information or traversal cost changes.	Primarily addresses replanning in partially known or changing environments rather than static preflight hierarchical planning with coarse-resolution risk guidance and fine-resolution trajectory refinement.
RRT-based risk-aware planners	[24, 26]	Primatesta et al. incorporated urban risk information into an RRT-based planner, whereas Rudnick-Cohen et al. optimized UAV takeoff trajectories to reduce third-party risk.	Focus on risk-aware sampling-based planning for urban paths or takeoff trajectories but do not establish an explicit coarse-to-fine mapping from global risk-guided waypoints to local metaheuristic optimization domains.
Risk- and conflict-aware cost-based planners	[25, 27]	Dai et al. developed conflict-free four-dimensional path planning by considering airspace occupancy, whereas Pang et al. incorporated third-party risk into an integrated path-cost model for metropolitan UAV operations.	Formulate conflict or third-party risk within their path-planning costs, but do not explicitly transfer coarse-resolution, risk-aware global guidance to high-resolution local trajectory-refinement domains.
PSO-based planner	[11]	Optimizes candidate UAV waypoint sequences through population-based search under safety and obstacle constraints.	Does not use graph-search-based global risk guidance followed by fine-resolution local trajectory refinement.
Proposed RDA*-IDB framework	–	Explicitly maps adjacent coarse-resolution, risk-aware global waypoints to corresponding high-resolution local optimization domains and refines the resulting trajectories using IDB.	Integrates voxel-based urban risk representation, risk-distance global guidance, waypoint-to-domain mapping, and obstacle-constrained local trajectory refinement within a unified multiresolution framework.

The fatality risk model, property damage-risk model, tent chaotic map, and Lévy flight operator

are adopted or adapted from previous studies; the contribution of the present study lies in their problem-specific integration within the proposed hierarchical framework. At the formulation level, the RDA* cost function combines normalized accumulated and heuristic risk costs with the corresponding accumulated and heuristic distance costs, and the heuristic risk term $h_R(n)$ is estimated by sampling voxel risk values along a three-dimensional Bresenham line from the current node to the goal node. At the framework level, adjacent global waypoints define the endpoints and spatial boundaries of the corresponding high-resolution local optimization domains, within which IDB refines each trajectory under obstacle, path length, and turning angle constraints. The real-world urban spatial data provide application-oriented validation of the resulting multiresolution risk-aware planning framework.

The paper's structure is organized as follows: Section 2 details UAV path planning constraints; Section 3 develops a hierarchical scenario model for low-altitude urban airspace; Section 4 presents the integrated RDA*-IDB planning approach; Section 5 conducts case validation in realistic urban settings to demonstrate its effectiveness; and finally, Section 6 concludes the study.

2. Path-planning problem formulation and risk-based constraints

When formulating the UAV path-planning problem, the objective components and hard feasibility constraints should be clearly distinguished. In this study, the permitted flight region and obstacle avoidance requirements are treated as hard feasibility constraints because any candidate path violating either condition is regarded as infeasible. By contrast, path length, accumulated path risk, and turning angle cost are treated as objective components to be minimized at the corresponding planning levels.

At the first planning level, the RDA* algorithm searches for a feasible coarse-resolution global waypoint sequence by balancing accumulated path risk and path distance cost. At the second planning level, the IDB algorithm optimizes the intermediate waypoints between each pair of adjacent global waypoints by minimizing local path length and turning angle costs while satisfying the fine-resolution flight region and obstacle avoidance constraints. Therefore, the complete path-planning task is formulated as a sequential hierarchical constrained optimization problem.

The following subsections first define the individual objective components and feasibility constraints. The complete hierarchical path-planning problem is then summarized in a single mathematical relationship in Eq (14), which jointly specifies the global and local decision variables, objective functions, feasibility constraints, and final path construction process.

(1) Flight area constraint

Flight area constraint is used to ensure the UAV's flight trajectory remains within a designated three-dimensional space, which is expressed mathematically as Eq (1).

$$\Omega = \{(x_i, y_i, z_i) \mid x_{\min} \leq x_i \leq x_{\max}, y_{\min} \leq y_i \leq y_{\max}, z_{\min} \leq z_i \leq z_{\max}\}. \quad (1)$$

In Eq (1), Ω denotes the permitted three-dimensional flight region for UAV operations. The coordinate (x_i, y_i, z_i) represents the position of the i th waypoint P_i on the generated path. The coordinate triplets $(x_{\min}, y_{\min}, z_{\min})$ and $(x_{\max}, y_{\max}, z_{\max})$ define the lower and upper bounds of the flight region along the three coordinate axes, respectively. All waypoints on a feasible path, including the prescribed start and goal points, must lie within Ω . Any candidate waypoint located outside these coordinate bounds is regarded as infeasible and is excluded from the planning process.

(2) Path-length objective

Shorter flight trajectories directly reduce UAV operational time and energy consumption. Therefore, the path length objective is defined in Eq (2).

$$L = \sum_{i=1}^{N-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2}. \quad (2)$$

In Eq (2), L denotes the total length of the UAV flight path, and the length of each segment is calculated using the three-dimensional Euclidean distance between two adjacent waypoints (x_i, y_i, z_i) and $(x_{i+1}, y_{i+1}, z_{i+1})$. N denotes the total number of waypoints, and (x_1, y_1, z_1) and (x_N, y_N, z_N) represent the prescribed start and goal positions, respectively. The path length term is treated as an efficiency-related objective component rather than as an independent feasibility constraint.

(3) Obstacle avoidance constraint

The UAVs should be able to avoid any path intersections with all obstacles in the urban environment. Therefore, to solve this problem, an obstacle avoidance constraint model is built. In this paper, the obstacle regions in the urban environment are marked using a binary function, which is given in Eq (3).

$$f_o(x, y, z) = \begin{cases} 1, & \text{if } (x, y, z) \text{ is an obstacle,} \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

In Eq (3), a function value of 1 indicates that the corresponding airspace voxel is occupied by an obstacle and is therefore nonflyable, whereas a function value of 0 indicates free and flyable airspace. To satisfy the obstacle avoidance constraint, not only all generated waypoints but also the evaluated voxels or sampled path positions must belong to free airspace. Therefore, a candidate path is regarded as infeasible if any waypoint or path segment intersects a voxel for which the occupancy function is equal to 1.

(4) Turning angle definition

To quantify the geometric smoothness of a candidate trajectory, the turning angle between two consecutive path segments is defined in Eq (4).

$$\begin{cases} \vartheta_i = \frac{(P_i - P_{i-1}) \cdot (P_{i+1} - P_i)}{\|P_i - P_{i-1}\| \|P_{i+1} - P_i\|}, \\ \theta_i = \arccos(\vartheta_i), \quad i = 2, 3, \dots, N-1. \end{cases} \quad (4)$$

In Eq (4), θ_i represents the turning angle between the incoming and outgoing path segments at the i th internal waypoint. The quantity ϑ_i denotes the normalized inner product of the two adjacent path-segment vectors. For two nonzero adjacent vectors, the Cauchy–Schwarz inequality guarantees that $-1 \leq \vartheta_i \leq 1$, and therefore, $\theta_i \in [0, \pi]$.

Here, $P_i = (x_i, y_i, z_i)$ denotes the position vector of the i th waypoint, whereas $P_i - P_{i-1}$ and $P_{i+1} - P_i$ represent the incoming and outgoing path-segment vectors, respectively. Because a turning angle is defined only at an internal waypoint, Eq (4) is applied for $i = 2, 3, \dots, N-1$. The operator “ $\cdot$ ” denotes the dot product, and $\|\cdot\|_2$ denotes the Euclidean norm. In the second-level local optimization, the turning angles calculated using Eq (4) are incorporated into the turning-angle cost of the IDB objective function.

(5) Path risk objective

For UAV path planning, geometric feasibility alone is insufficient to characterize operational safety. Therefore, the accumulated path risk objective is introduced to quantify the risk exposure along the complete path, as defined in Eq (5).

$$R = \sum_{i=1}^{N-1} \left\{ \frac{L_i}{2} [C_u(x_i, y_i, z_i) + C_u(x_{i+1}, y_{i+1}, z_{i+1})] \right\}. \quad (5)$$

Equation (5) defines the accumulated path risk objective R . For each path segment, the corresponding risk contribution is calculated by multiplying the segment length by the mean integrated risk value at its two endpoints, which provides a trapezoidal approximation of the risk accumulated along that segment. L_i denotes the length of the segment connecting the i th and $(i + 1)$ th waypoints, and N denotes the total number of waypoints. The accumulated path risk is incorporated into the first-level global objective function and is not imposed as an independent upper-bound constraint.

Moreover, $C_u(x_i, y_i, z_i)$ represents the risk value at the grid coordinate (x_i, y_i, z_i) , integrating two fundamental risk components: fatality risk cost and property damage risk cost. The calculation of C_u is presented in Eq (6).

$$C_u = \sum_{i=1}^2 \alpha_i \omega_i C_i. \quad (6)$$

In Eq (6), C_u represents the normalized integrated risk value assigned to a single three-dimensional airspace voxel. α_1 and α_2 represent the weight coefficients for the fatality risk cost and property damage risk cost, respectively. These weight coefficients satisfy $0 < \alpha_i \leq 1$ and $\alpha_1 + \alpha_2 = 1$. ω_i denotes the normalization coefficient for each type of risk cost and is calculated as $\omega_i = 1/C_{i,max}$, where $C_{i,max}$ is the maximum value of the corresponding risk component among all three-dimensional airspace voxels in the considered urban risk map. C_i represents a specific type of risk cost, where $C_1 = C_{r,f}$ and $C_2 = C_{r,b}$.

Therefore, the fatality risk cost and property damage risk cost are normalized separately before being combined, thereby avoiding the direct addition of two risk components with different numerical scales. Because the normalized risk components and their corresponding weights are non-negative, the resulting value of C_u lies within the interval $[0, 1]$. It is interpreted as a relative risk index for comparing different airspace voxels during path planning rather than as a direct fatality probability or an absolute monetary loss value.

The detailed construction and validation of the urban risk map, including ground feature extraction, population density estimation, sheltering factor assignment, ground obstacle-layer generation, impact energy calculation, and risk map resolution conversion, were systematically presented in our previous risk-assessment study [31]. In the present study, the generated risk map is adopted as an input to the hierarchical path-planning framework rather than redeveloped as an independent risk-assessment contribution.

The fatality risk cost and property damage risk cost are derived from Eqs (7) and (8), respectively. In Eq (7), $C_{r,f}$ represents the fatality risk cost, which consists of the fatality risk for ground pedestrians $C_{r,p}$ and the fatality risk for people in ground vehicles $C_{r,v}$.

$$C_{r-f} = C_{r-p} + C_{r-v}, \quad (7)$$

$$C_{r-b} = \begin{cases} \alpha_b \psi(e^\mu), & 0 < h \leq e^\mu, \\ \alpha_b \psi(h), & h > e^\mu. \end{cases} \quad (8)$$

When a UAV fails in the air and hits ground pedestrians, the fatality risk should be evaluated from three parts. First, midair failure probability depends on the UAV's intrinsic reliability. The second is the probability of UAVs hitting ground pedestrians, which is associated with ground population distribution and population density. The third is the probability of death after a UAV hits a pedestrian, which is governed by the UAV's impact kinetic energy. Based on the analysis above, the fatality risk for ground pedestrians can be expressed as Eq (9).

$$C_{r-p} = P_{crash} \times P_{p-i} \times R_{p-f}. \quad (9)$$

In Eq (9), C_{r-p} denotes the fatality risk cost resulting from a UAV's hit to ground pedestrians. P_{crash} defines the probability of UAV in-flight failure culminating in ground impact. In this study, P_{crash} represents the baseline UAV system failure probability under nominal operating conditions. Following the parameter setting adopted by Pang et al. [27] for third-party risk assessment and UAV path optimization in metropolitan environments, P_{crash} is set to 3.42×10^{-4} per flight hour. This parameter is regarded as a spatially uniform reliability parameter for the construction of the preflight urban risk map. P_{p-i} quantifies pedestrians within urban impact zones susceptible to UAV crashes. R_{p-f} is the fatality probability for ground pedestrians after being impacted by a UAV, namely the ground pedestrian fatality rate.

Previous studies have classified the post-failure descent process of UAVs into four primary modes: ballistic descent, uncontrolled glide, parachute descent, and active fly-away [32]. Different descent modes may result in different impact locations, impact footprints, and impact severities. In this study, ballistic descent is adopted as a uniform baseline contingency scenario following an in-flight UAV failure. Therefore, the resulting fatality risk represents the conditional risk associated with the ballistic descent scenario rather than an aggregated risk covering all possible descent modes. The same descent assumption is applied to all airspace voxels and candidate flight paths, thereby providing a consistent basis for comparing their relative risk levels. Equation (10) gives the total number of ground pedestrians P_{p-i} exposed within the UAV impact area under the ballistic descent scenario.

$$P_{p-i} = aN_p. \quad (10)$$

In this equation, a is an exposure adjustment coefficient associated with the ground characteristics at the UAV crash location. N_p denotes the number of ground pedestrians within the UAV impact area. The value of N_p is determined by the population density at the corresponding ground location and the potential ground impact area of the failed UAV rather than being assigned as an independent constant. Under the adopted ballistic descent assumption, the ground impact area is estimated according to the UAV dimensions, representative pedestrian dimensions, and the descent angle. Therefore, the pedestrian exposure term reflects both the spatial variation in population density and the impact footprint of the falling UAV. The detailed calculations of the population exposure and ground impact

area were presented in our previous risk-assessment study [31]. To calculate the ground pedestrian fatality rate R_{p-f} , Eq (11) is used:

$$R_{p-f} = \frac{1}{1 + \sqrt{\frac{\alpha}{\beta}} \left(\frac{\beta}{E_k} \right)^{\frac{1}{4S_f}}}. \quad (11)$$

In Eq (11), E_k denotes the UAV kinetic energy at ground impact. It is determined by the UAV mass and the impact velocity. Under the adopted ballistic descent assumption, the impact velocity depends on the UAV failure altitude and aerodynamic parameters, including the air density, drag coefficient, and frontal area of the falling UAV. Therefore, the influence of the failure altitude is incorporated into the fatality risk calculation through the impact velocity and the resulting impact kinetic energy.

S_f is a sheltering factor satisfying $S_f \in (0, 1]$, where a larger value indicates stronger protection provided by the corresponding ground feature. α is defined as the impact energy corresponding to a fatality rate of 50% when $S_f = 0.5$. β is the threshold of impact kinetic energy that may cause ground fatalities when S_f approaches zero. The detailed calculations of the ballistic descent, impact velocity, and impact kinetic energy were presented in our previous risk-assessment study [31].

The sheltering factor S_f is used to characterize the relative protection provided by different ground features against the impact of a falling UAV. The definition of the sheltering factor and its role in the pedestrian fatality model follow previous studies on UAV third-party risk assessment and urban ground risk mapping [24, 27]. In this study, S_f is treated as a normalized ordinal parameter rather than a directly measured probability. A five-level scale, $S_f = \{0, 0.25, 0.50, 0.75, 1.00\}$, is adopted to distinguish the relative sheltering capacities of the ground feature classes extracted from the high-resolution satellite image. The assignment is based on the presence, continuity, and structural strength of potential protective elements, such as vegetation canopies, vehicles, roofs, walls, and multistory building structures. A larger value of S_f indicates a higher relative capacity to absorb impact energy or shield ground pedestrians.

Accordingly, the urban terrain is divided into five major sheltering categories comprising fifteen ground feature subclasses:

- $S_f = 0$ (approximately unprotected areas): Grassy areas, water surfaces, paved concrete surfaces, loess terrain, and construction sites with blue coverings. These areas generally lack effective structural elements capable of shielding pedestrians from a falling UAV.
- $S_f = 0.25$ (limited sheltering): Roads designed for vehicular travel. Although roads themselves provide little fixed protection, vehicles and other discontinuous roadside elements may provide limited sheltering effects.
- $S_f = 0.50$ (moderate sheltering): Vegetation with woody stems, outbuildings with blue roofs, and structures with red roof coverings. Tree canopies and lightweight roof structures can absorb part of the impact energy, but their protective capacity remains limited.
- $S_f = 0.75$ (strong sheltering): Tiled low-rise buildings in brown, gray, or red, together with industrial zones characterized by silver-gray or white-blue roofs. Their relatively continuous roofs and rigid structural components provide stronger sheltering effects.
- $S_f = 1.00$ (the highest sheltering level): Multistory buildings. Their complete roof, wall, and multilayer structural systems provide the strongest relative protection among the ground feature categories considered in this study.

The above values represent relative sheltering levels used to establish a consistent mapping between the extracted ground feature classes and the fatality risk model; they should not be interpreted as directly measured absolute protection probabilities. For the unprotected category, $S_f = 0$ is retained as a categorical label and is interpreted in Eq (11) as the limiting condition $S_f \rightarrow 0^+$ rather than as an ordinary positive parameter directly substituted into the exponent.

As indicated in Eq (7), $C_{r,v}$ represents the fatality risk of occupants in ground vehicles which could be obtained with Eq (12).

$$C_{r,v} = P_{crash} P_{v,i} R_{v,f}. \quad (12)$$

In the equation above, P_{crash} quantifies the probability of a UAV in-flight failure resulting in ground impact. $P_{v,i}$ denotes the number of ground vehicles exposed within the potential UAV impact area. It is determined by the vehicle density at the corresponding ground location and the impact area of the failed UAV. Therefore, the vehicle exposure term reflects both the spatial distribution of ground traffic and the influence of the UAV failure altitude on the impact footprint. $R_{v,f}$ denotes the occupant fatality rate resulting from a UAV impact on a ground vehicle. It is evaluated according to the impact energy associated with the relative motion between the falling UAV and the moving vehicle and therefore depends on the UAV impact velocity, vehicle speed, and failure altitude. The detailed calculation of the vehicle exposure and impact-related fatality rate was presented in our previous risk-assessment study [31].

When the risk cost of property damage is calculated using Eq (8), α_b is a weight coefficient for different building types with the value $\alpha_b \in (0, 1]$. The values are determined values assigned according to specific building categories. $\psi(h)$ belongs to a log-normal distribution function that can be obtained from Eq (13). h_b represents the urban building height, with μ and σ representing the mean and standard deviation of the log-transformed variable, respectively:

$$\psi(h_b; \mu, \sigma) = \frac{1}{h_b \sigma \sqrt{2\pi}} e^{-\frac{(\ln h_b - \mu)^2}{2\sigma^2}}. \quad (13)$$

The coefficient α_b represents the relative property damage severity associated with different building categories. Blue- and red-roofed sheds are assigned $\alpha_b = 0.25$, indicating a comparatively low level of potential economic loss in the event of a UAV impact. Tile low-rise buildings, including the brown, gray, and red variants, are assigned $\alpha_b = 0.50$, representing a moderate level of potential property damage. Industrial areas with silver-grey or white-blue roofs and high-rise buildings are assigned higher values of $\alpha_b = 0.80$ and $\alpha_b = 1.00$, respectively because impacts on these building categories may result in greater economic losses. Therefore, a larger value of α_b indicates a higher relative property damage consequence rather than a direct probability of structural failure.

The spatial distribution and vertical extent of the property damage component are determined by the footprints and heights of buildings or other ground infrastructure. For a voxel located outside the corresponding footprint or above the structure height, the property damage risk is set to zero. The building height data therefore define the vertical range of the property damage risk, and α_b distinguishes the relative property damage severity and potential economic consequences associated with different building categories. The detailed assignment and application of these building category coefficients were presented in our previous risk-assessment study [31].

2.1. Hierarchical path-planning optimization formulation

Let the first-level global path be represented by an ordered sequence of coarse-resolution, three-dimensional waypoints $\mathcal{G} = \{G_1, G_2, \dots, G_M\}$, where $G_i = (x_i^g, y_i^g, z_i^g)$ denotes the i th global waypoint, and M denotes the total number of global waypoints. For each pair of adjacent global waypoints G_j and G_{j+1} , the second-level local decision vector is defined as $X_j = [x_{j,1}, y_{j,1}, z_{j,1}, \dots, x_{j,K}, y_{j,K}, z_{j,K}]^T$, where $K = 5$ denotes the number of intermediate waypoints. The mapped local start and goal positions are fixed and are not included in the decision vector.

After decoding X_j , the local start point, the five intermediate waypoints, and the local goal point are interpolated to generate the sampled local path $\Gamma_j(X_j) = \{Q_{j,1}, Q_{j,2}, \dots, Q_{j,N_s}\}$, where $N_s = 500$, and $Q_{j,s}$ denotes the s th sampled position of the j th local path.

$$\left\{ \begin{array}{l} \widehat{\mathcal{G}} \in \arg \min_{\mathcal{G}} J_G(\mathcal{G}) = \alpha_R \omega_R R(\mathcal{G}) + \alpha_D \omega_D L(\mathcal{G}), \\ \text{s.t. } G_1 = G_{\text{start}}, \quad G_M = G_{\text{goal}}, \\ G_i \in \Omega_c, \quad f_o^c(G_i) = 0, \quad i = 1, 2, \dots, M, \\ G_{i+1} \in \mathcal{N}_{26}(G_i), \quad i = 1, 2, \dots, M-1, \\ \alpha_R + \alpha_D = 1, \quad \alpha_R \geq 0, \quad \alpha_D \geq 0, \\ \widehat{X}_j \in \arg \min_{X_j} J_{D,j}(X_j) = \omega_{\text{path}} J_{\text{path},j}(X_j) + \omega_{\text{turn}} J_{\text{turn},j}(X_j), \quad j = 1, 2, \dots, M-1, \\ \text{s.t. } L_j \leq X_j \leq U_j, \quad j = 1, 2, \dots, M-1, \\ Q_{j,1} = S_j, \quad Q_{j,N_s} = T_j, \quad j = 1, 2, \dots, M-1, \\ Q_{j,s} \in \Omega_j, \quad f_o^f(Q_{j,s}) = 0, \\ j = 1, 2, \dots, M-1, \quad s = 1, 2, \dots, N_s, \\ \omega_{\text{path}} + \omega_{\text{turn}} = 1, \quad \omega_{\text{path}} = 0.8, \quad \omega_{\text{turn}} = 0.2, \\ \widehat{\mathcal{P}} = \Gamma_1(\widehat{X}_1) \oplus \Gamma_2(\widehat{X}_2) \oplus \dots \oplus \Gamma_{M-1}(\widehat{X}_{M-1}). \end{array} \right. \quad (14)$$

In Eq (14), J_G denotes the first-level global objective function. The accumulated path risk objective $R(\mathcal{G})$ and the path length objective $L(\mathcal{G})$ are calculated using Eqs (5) and (2), respectively. The normalization factors ω_R and ω_D place the risk and distance components on comparable numerical scales, and α_R and α_D determine their relative priorities.

The global path must remain within the permitted flight region defined in Eq (1) and satisfy the obstacle avoidance condition defined by Eq (3). In addition, every pair of consecutive global waypoints must satisfy the adopted 26-neighbor connectivity relationship.

At the second planning level, $J_{\text{path},j}$ is calculated according to the path length definition in Eq (2), and $J_{\text{turn},j}$ is constructed from the turning angles defined in Eq (4). L_j and U_j denote the lower and upper bounds of the local decision vector, and S_j and T_j denote the mapped feasible local start and goal positions.

Ω_j denotes the spatial boundary of the j th fine-resolution local planning domain, and f_o^f denotes its fine-resolution obstacle occupancy function. The local domain and obstacle avoidance constraints in Eq (14) are evaluated using all $N_s = 500$ interpolated path samples rather than only the five intermediate waypoints. A candidate local path is regarded as infeasible if any sampled position falls outside the local planning domain or intersects an occupied fine-resolution voxel.

The symbol $\oplus$ in Eq (14) denotes ordered path concatenation. The complete UAV path $\widehat{\mathcal{P}}$ is obtained by sequentially connecting the optimized local path segments and removing the duplicated endpoint shared by every two adjacent local segments. Equation (14) therefore provides a single unified mathematical relationship for the complete hierarchical path-planning problem, explicitly connecting the global and local decision variables, their corresponding objective functions and feasibility constraints, and the final path construction process.

Because the Bresenham-based risk heuristic used by RDA* is not guaranteed to be admissible or consistent, RDA* is regarded as a numerical search method for obtaining an approximate feasible solution to the first-level problem in Eq (14); exact global optimality is not claimed. Similarly, IDB is a stochastic population-based method used to approximate the solutions of the second-level local optimization problems in Eq (14), and no guarantee of reaching the global optimum is asserted.

3. Hierarchical UAV path-planning scenario modeling

To support safe UAV path-planning, the normalized three-dimensional risk map is first introduced as an input to the hierarchical planning framework. The urban airspace is partitioned into rectangular three-dimensional voxels, and each voxel is assigned an integrated risk value calculated using Eq (6). The resulting risk-based airspace model is illustrated in Figure 1, where each voxel is represented by a rectangular cell with a center point C_{xyz} .

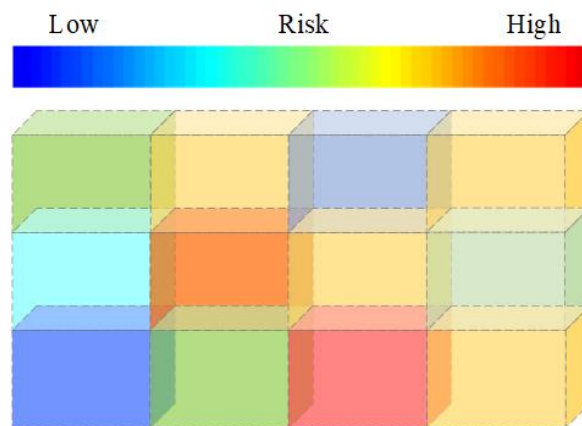


Figure 1. A risk-based airspace model.

Because the first-level global planner and the second-level local planner have different requirements for computational efficiency and spatial detail, risk maps with different voxel resolutions are employed. The current multilayer-based model provides a fine-resolution risk map with a voxel size of $1\text{ m} \times 1\text{ m} \times 1\text{ m}$. This fine-resolution map is spatially aggregated to obtain the coarse-resolution risk map required by the first-level global planner, and the detailed environmental information is retained for second-level local trajectory refinement. The corresponding resolution conversion procedure is given in Eq (15).

$$\mathbf{C}_{u,n,n,m}(i, j, k) = \frac{1}{n^2 m} \sum_{a=1}^n \sum_{b=1}^n \sum_{c=1}^m \mathbf{C}_{u,1,1,1}(ni + a, nj + b, mk + c). \quad (15)$$

Within the resulting $n \times n \times m$ dimensional risk map, $\mathbf{C}_{u,n,n,m}(i, j, k)$ corresponds to the mean risk estimate of one voxel. The three parameters i , j , and k refer to a voxel's position in the x , y , and z directions of the new map. Similarly, the other three parameters a , b , and c would iterate over all voxel locations of the original high-resolution map, which is contained within the new voxel to compute the average value of the risk cost.

3.1. Urban 3D risk map model

According to Eq (6), a location-based 3D risk map is constructed by partitioning the airspace into many voxels and quantifying the integrated risk within each of them. The integrated risk caused by UAV operations includes the fatality and property damage risk. Figure 2 illustrates the dimension of each voxel in the generated 3D risk map as $n \times n \times m$ meters.

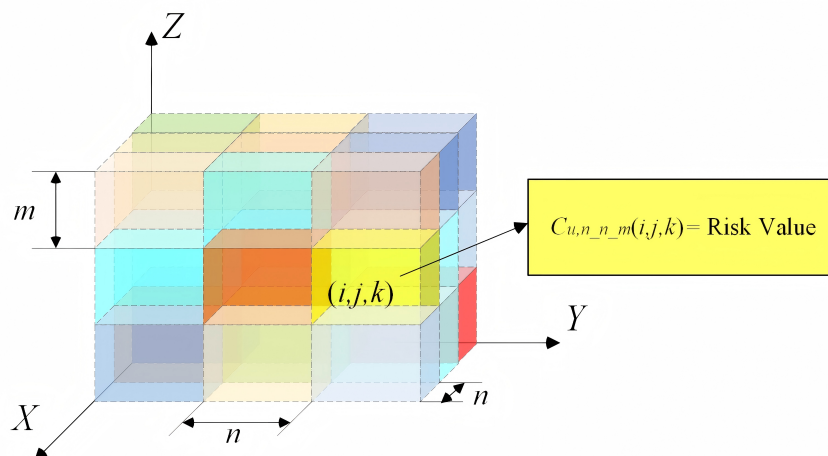


Figure 2. Representation of a 3D risk map.

Further, the risk map can be expressed as a 3D risk matrix $\mathbf{C}_{u,n,n,m}(x, y, z)$, which is given in Eq (16). This matrix is built from a lot of voxels with the same size of $n \times n \times m$. In this way, the resolution of this risk map is $N \times N \times M$.

$$\mathbf{C}_{u,n,n,m}(x, y, z) = \{C_1, C_2, \dots, C_M\}. \quad (16)$$

The 3D matrix can be sliced into M layers along the Z -axis, as stated in Eq (17). By fixing k and iterating over i and j , the corresponding 2D risk matrix $\mathbf{C}_k$ can be obtained. It can be seen that each element at (i, j, k) stands for a 2D grid and holds a risk value of $\mathbf{C}_{u,n,n,m}(i, j, k)$.

$$\mathbf{C}_k = \begin{bmatrix} C(1, 1, k) & C(1, 2, k) & \cdots & C(1, N, k) \\ C(2, 1, k) & C(2, 2, k) & \cdots & C(2, N, k) \\ \vdots & \vdots & \ddots & \vdots \\ C(N, 1, k) & C(N, 2, k) & \cdots & C(N, N, k) \end{bmatrix}, \quad k = 1, 2, \dots, M. \quad (17)$$

In this paper, to quantify risk impacts on UAV paths as well as to reduce complexities and improve the computational efficiencies of a path planning algorithm, a resolution conversion mechanism is proposed for the risk map generation based on Eq (15), as illustrated in Figure 3. In the original map, the size of each voxel is $1\text{ m} \times 1\text{ m} \times 1\text{ m}$. To obtain the specified risk constraints suited to urban UAV operations, a new risk map is defined with a voxel with the size of $100\text{ m} \times 100\text{ m} \times 20\text{ m}$. With this conversion mechanism, each converted risk voxel encompasses $100 \times 100 \times 20$ voxels of the original one.

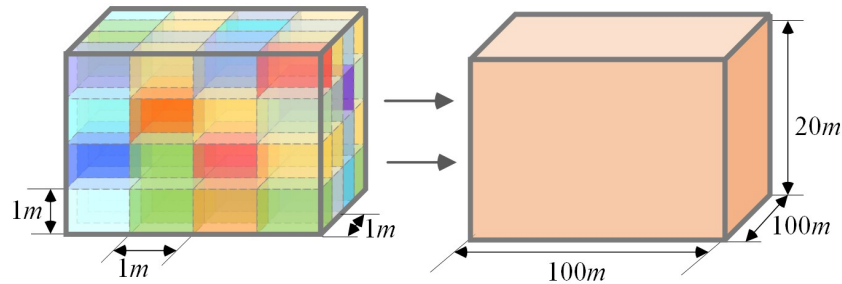


Figure 3. Risk map resolution conversion.

3.2. Hierarchical path-planning scenario model

Most of the traditional path-planning scenarios are idealized and cannot capture all the complexities of urban environments. However, searching for an optimal UAV path in a large-scale, high-resolution urban environment can impose substantial computational demands. To address this issue, a hierarchical scenario model is established for UAV path planning, which reduces computational complexity while retaining high-resolution risk constraints in local urban areas. This provides an efficient and realistic planning environment for UAV operations.

The proposed hierarchical path-planning scenario model incorporates two distinct layers: graph search-based global path-planning and intelligent optimization-based local path-planning.

For the modelling of the first-level path-planning scenario as illustrated in Figure 4(a), there are 26 neighboring locations available that can be used as the next waypoint from current location if no obstacles or boundary conditions are present. These available locations can be further grouped into three main categories according to the movement directions of UAVs.

The first group is defined as a face-adjacent node. There are six in total, which are in the up, down, left, right, forward, and backward directions. These nodes share a common face with the current node and would incur the minimum Euclidean distance cost. The second group is a kind of edge-adjacent node, whose number is twelve. Each of them shares a common edge with the current node and constitutes the largest subset of the candidate neighboring nodes. The third group is the vertex-adjacent node. There are eight vertex-adjacent nodes in total. They are located along the space diagonals, which are the farthest from the current location.

Based on the preceding classification, the first-level path-planning for UAV operations is performed, as shown in Figure 4(b). The colored voxels denote different risk levels. In this situation, the planned UAV flight path must avoid the high-risk voxels to minimize the overall risk cost. For the second-level local path-planning, an intelligent optimization algorithm is adopted.



Figure 4. First-level path-planning scenario.

Combining the first-level and the second-level path-planning scenarios, a hierarchical mechanism for the generation of UAV path-planning model is implemented. The adjacent waypoints obtained from the first-level path-planning scenario are used to define a spatial extent which is then located in the urban risk map. Within this defined spatial extent, a local operation scenario is established that incorporates relevant urban ground features, such as buildings and terrain.

Similarly, the second-level path-planning scenarios can be classified into two types, as shown in Figure 5. The first is a kind of coarse grids that denote the voxels in the first-level global path-planning scenario with a size of $100\text{ m} \times 100\text{ m} \times 20\text{ m}$. The red frame highlights two adjacent waypoints generated by a global planner. The second is defined as fine grids, which depict the risk distribution in the local planning scenario with a voxel size of $1\text{ m} \times 1\text{ m} \times 1\text{ m}$. The grid color encodes their corresponding risk value in each voxel.

Figure 5(a) depicts the second-level local planning environment, generated from the first-level global planning framework. It is composed of many left-right face-adjacent nodes. There are two coarse-grid voxels with an overall dimension of $100\text{ m} \times 200\text{ m} \times 20\text{ m}$. Similar to Figure 5(a), Figure 5(b) shows the available up-down face-adjacent nodes with an overall dimension of $100\text{ m} \times 100\text{ m} \times 40\text{ m}$. Figure 5(c) presents local path-planning scenario in the second-level for the edge-adjacent nodes consisting of four coarse-grid voxels with a total size of $200\text{ m} \times 200\text{ m} \times 20\text{ m}$. Figure 5(d) shows a local planning scenario generated by the vertex-adjacent nodes, which comprises eight coarse-grid voxels with a size of $200\text{ m} \times 200\text{ m} \times 40\text{ m}$.

Obstacle avoidance is enforced within the adopted discrete environmental representation, which commonly employs grid maps and probabilistic maps. In this study, an obstacle height-layer model is used, which shares the same resolution with the second-level path-planning scenario. In this way, it can be applied directly as an obstacle avoidance constraint for the local planner.

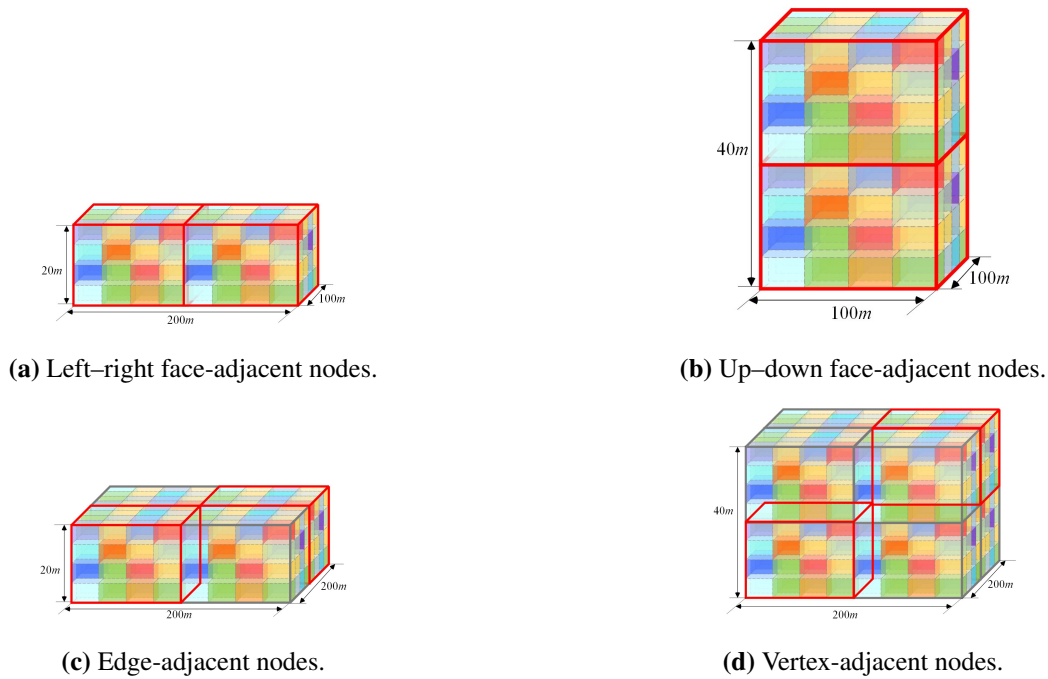


Figure 5. Second-level path-planning scenario.

However, it cannot be directly applied to the first-level global planning scenario because of the different voxel sizes. To address this mismatch, a large voxel-based obstacle volume-ratio calculation is introduced as follows:

$$\begin{cases} O_{\text{rate}}(i, j, k) = \frac{V_o(i, j, k)}{G_x G_y G_z} \\ V_o(i, j, k) = \sum_{(x,y) \in C(i,j)} \max \{0, \min (O_h(x, y), kG_z) - (k-1)G_z\}. \end{cases} \quad (18)$$

In Eq (18), $O_{\text{rate}}(i, j, k)$ represents the ratio of the obstacle volume to the total volume of a large voxel located at (i, j, k) . $V_o(i, j, k)$ denotes the obstacle volume within that voxel, which can be obtained by calculating the volume of each obstacle according to its height and summing these volumes. G_x , G_y , and G_z are the dimensions of a large voxel with $G_x = 100$ m (X -axis), $G_y = 100$ m (Y -axis), and $G_z = 20$ m (Z -axis). $O_h(x, y)$ is the height of an obstacle in the position (x, y) . Once the obstacle volume ratio has been determined, a binary function can be applied to define the obstacles in the first-level global path-planning scenario as shown in Eq (19).

$$f_o(i, j, k) = \begin{cases} 1, & O_{\text{rate}}(i, j, k) > \theta_o, \\ 0, & O_{\text{rate}}(i, j, k) \leq \theta_o. \end{cases} \quad (19)$$

In Eq (19), $f_o(i, j, k)$ denotes the obstacle matrix. θ_o is the threshold used to determine whether a voxel is classified as an obstacle or not. $O_{\text{rate}}(i, j, k)$ represents the ratio of the obstacle volume to the total volume of a large voxel located at (i, j, k) . If $f_o(i, j, k) = 1$, the voxel would be marked as an obstacle region. Oppositely if $f_o(i, j, k) = 0$, the voxel would not be an obstacle region that would be feasible for UAV operations.

4. Hierarchical RDA*- and IDB-based UAV path-planning

4.1. First-level global path guidance with RDA*

The conventional A* algorithm is a classic heuristic search approach. Its core strategy is to construct an open node list and a closed node list. A cost function is then used to evaluate the accessibility and search priority of each candidate node, thereby enabling an efficient search for a minimum-cost path between the start and goal nodes [29]. The path quality is evaluated by combining the actual cost from the start node to the current node with an estimate of the remaining cost from the current node to the goal node. Equation (20) gives the conventional A* cost function:

$$f(n) = g(n) + h(n). \quad (20)$$

In Eq (20), $f(n)$ denotes the total estimated cost associated with node n . $g(n)$ represents the actual accumulated cost from the start node to the current node n , and $h(n)$ denotes the estimated remaining cost from node n to the goal node. The heuristic function influences both the search direction and the quality of the generated path. Manhattan distance and Euclidean distance are commonly used distance-based heuristic functions.

In this part, to implement the first-level global path guidance under complex urban conditions, the RDA* algorithm is proposed by jointly considering path risk and path length in the cost function of the conventional A* algorithm. A weighted heuristic information is introduced to integrate the risk and distance costs so that the search can be guided toward airspace regions with relatively lower estimated risk while maintaining path efficiency.

Figure 6 shows how the heuristic information is used in the RDA* mechanism based on the risk distributions in the environment. Because path distance alone cannot adequately reflect the heuristic value in a spatially heterogeneous risk field, and the risk from the current node (x_i, y_i, z_i) to the goal node (x_N, y_N, z_N) is generally unevenly distributed, a 3D Bresenham line is introduced to sample the risk values along the direct line between the two nodes.

According to the 3D Bresenham line, a set of discrete voxels between the current node and the goal node can be generated, as illustrated by the black dots in Figure 6. These discrete voxels are used only to sample the spatial risk distribution and estimate the general risk tendency toward the goal. Therefore, the generated line should not be interpreted as an actual collision-free or dynamically reachable UAV path.

The flowchart of the 3D Bresenham line algorithm is shown in Figure 7. First, the algorithm takes the positions of the current node (x_i, y_i, z_i) and the goal node (x_N, y_N, z_N) as the inputs and calculates the coordinate increments $(\Delta x, \Delta y, \Delta z)$ together with the step directions (x_s, y_s, z_s) . Second, the primary stepping axis is determined by the largest increment, namely $\max(\Delta x, \Delta y, \Delta z)$. Two error terms (p_1, p_2) are initialized to control the secondary-axis stepping. Third, during the iteration, the algorithm advances along the primary axis, and the error terms determine whether a step on a secondary axis is required. In each step, the current coordinate is updated, and all the discrete points are stored until the goal node is reached. Finally, the algorithm outputs a complete set of discrete sampling points.

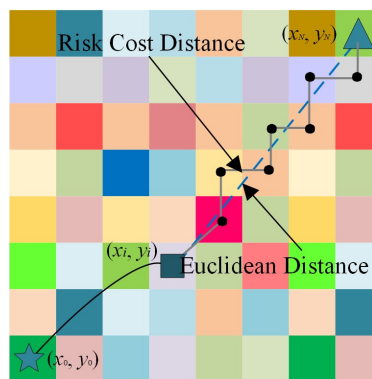


Figure 6. Risk-based heuristic sampling using the three-dimensional Bresenham line.

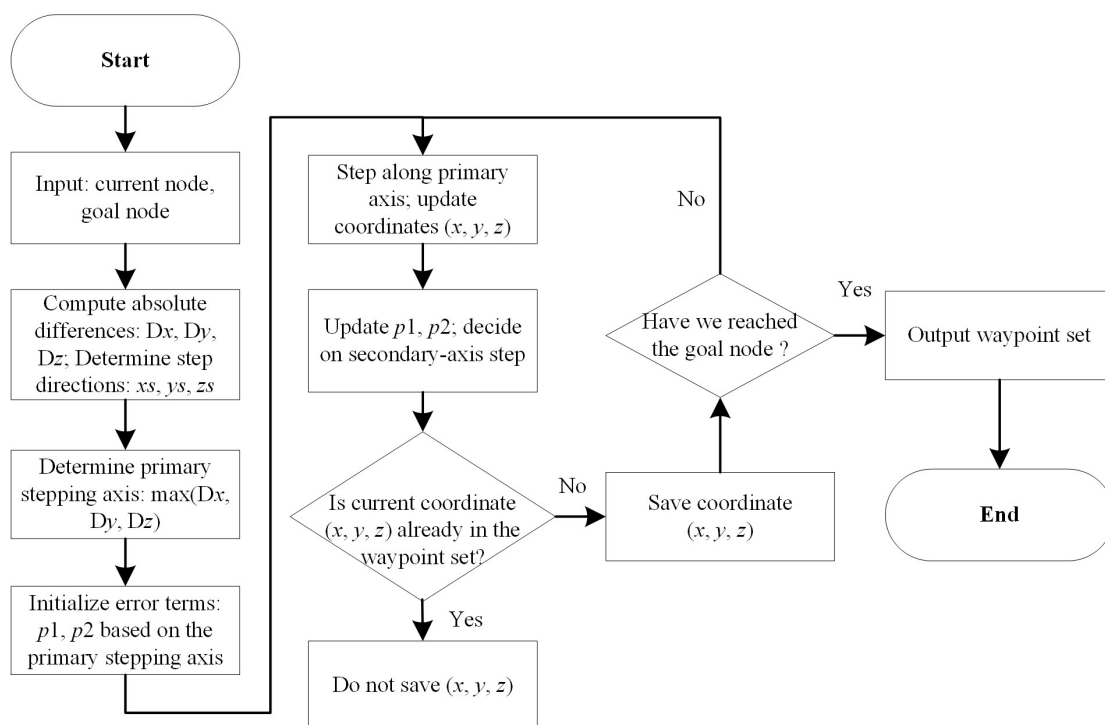


Figure 7. Flowchart of the 3D Bresenham line algorithm.

To explicitly incorporate risk cost into first-level global path-planning, the RDA* algorithm introduces risk cost into the conventional A* framework as an additional metric. The corresponding risk cost function is given in Eq (21):

$$f_R(n) = g_R(n) + h_R(n). \quad (21)$$

In Eq (21), $g_R(n)$ denotes the cumulative risk cost from the start node to the current node n along the feasible path that has already been traversed. In contrast, $h_R(n)$ denotes the approximate remaining risk cost from the current node n to the goal node, which is estimated from the discrete voxels generated by the 3D Bresenham line.

According to Eq (5), the calculations for $g_R(n)$ and $h_R(n)$ are given by Eqs (22) and (23), respectively. Specifically, $g_R(n)$ is computed as the sum of the risk cost values of all nodes along the already traversed

path, whereas $h_R(n)$ is obtained by applying the Bresenham line to generate the discrete voxels between the current node n and the goal node and then summing the corresponding risk costs to estimate the remaining risk of the path:

$$g_R(n) = \sum_{k=1}^{i-1} \frac{L_k [C_u(x_k, y_k, z_k) + C_u(x_{k+1}, y_{k+1}, z_{k+1})]}{2} \quad (22)$$

$$h_R(n) = \sum_{k=i}^{N-1} \frac{L_k [C_u(x_k, y_k, z_k) + C_u(x_{k+1}, y_{k+1}, z_{k+1})]}{2}. \quad (23)$$

It should be noted that in Eq (23), the discrete voxels between the current node and the goal node are the sampling points generated by the 3D Bresenham line rather than the nodes of an already verified feasible path. Because the sampled straight line does not explicitly account for obstacles or feasible detours, the value of $h_R(n)$ may either overestimate or underestimate the actual minimum remaining risk cost, depending on the spatial distributions of obstacles and environmental risks. Therefore, $h_R(n)$ is used as a risk-informed approximate heuristic to guide the search toward regions with relatively lower estimated risk rather than as an exact lower bound or an exact representation of the actual remaining path risk. Accordingly, it is not guaranteed to satisfy the admissibility and consistency conditions of the classical A* heuristic, and a strict global optimality guarantee is not claimed for the proposed RDA* algorithm.

In order to reconcile the competing objectives of path efficiency and flight safety, weighting coefficients are introduced into the cost function. These coefficients determine the proportional influence of risk versus distance cost, thereby enabling the generation of a safe and efficient UAV path. The objective function of the proposed RDA* algorithm is given by Eq (24).

$$f_{RD}(n) = \alpha_R \omega_R [g_R(n) + h_R(n)] + \alpha_D \omega_D [g(n) + h(n)]. \quad (24)$$

In Eq (24), $g_R(n)$ and $h_R(n)$ denote the cumulative risk cost and the heuristic risk cost at the current node n , whereas $g(n)$ and $h(n)$ represent the cumulative distance cost and the heuristic distance cost, respectively. The weighting coefficients α_R and α_D correspond to the path risk cost and the path length cost, with $\alpha_R + \alpha_D = 1$. It should be emphasized that α_R and α_D are mission-dependent preference parameters rather than intrinsic constants of the RDA* algorithm. They represent the relative priorities assigned to flight safety and path efficiency, respectively, and their appropriate values may vary with the spatial risk distribution, obstacle configuration, start–goal pair, and operational requirements. Therefore, the selected weight combination should be interpreted as a scenario-specific operating setting and should be recalibrated when the planning environment or mission requirements change. To place the risk cost and distance cost terms on the same order of magnitude, two normalization factors, ω_R and ω_D , are introduced. ω_R is defined as the reciprocal of the accumulated risk cost along the space-diagonal path from the left-front-bottom voxel to the right-rear-top voxel in the first-level global planning scenario. ω_D is the reciprocal of the total Euclidean distance of that same diagonal path.

It should be distinguished that Eq (24) is the node priority evaluation function used during the RDA* search, whereas Eq (14) defines the target optimization problem for the complete global path. The accumulated terms $g_R(n)$ and $g(n)$ represent the risk and distance costs already incurred from the

start node to the current node, whereas $h_R(n)$ and $h(n)$ estimate the remaining risk and distance costs from the current node to the goal node. Therefore, Eq (24) provides the algorithmic search mechanism used to approximate the solution of the global path-planning problem rather than defining a separate final-path objective.

Figure 8 shows the workflow of first-level global path guidance with the proposed RDA* algorithm. Prior to the search, the global path-planning parameters are initialized, the start node is inserted into the open list (OPEN), and the closed list (CLOSE) is created by the planner. Next, the RDA* algorithm checks whether OPEN is empty. If it is empty, no further search is possible, and the RDA* algorithm terminates. Otherwise, all the nodes in OPEN are sorted according to the integrated cost function, and the node $N_{\min}$ with the smallest $f_{RD}(n)$ is identified.

The selected node $N_{\min}$ is then removed from OPEN and placed into CLOSE. If $N_{\min}$ is the goal node, the planned global path is obtained by backtracking through the parent-node chain, and the RDA* algorithm terminates. If $N_{\min}$ is not the goal node, its neighboring nodes are expanded. During this process, nodes located outside the permitted flight area or inside obstacle voxels are excluded before their integrated costs are evaluated. Therefore, although the Bresenham-based heuristic itself does not represent an obstacle-free path, the feasibility and collision avoidance of the final planned path are maintained through explicit obstacle and boundary checks during node expansion. After the integrated costs are updated, each feasible node is inserted into OPEN, or its existing entry in OPEN is updated. This process iterates until the goal node is found or OPEN becomes empty. When the goal node is reached, the algorithm outputs the planned first-level global path.

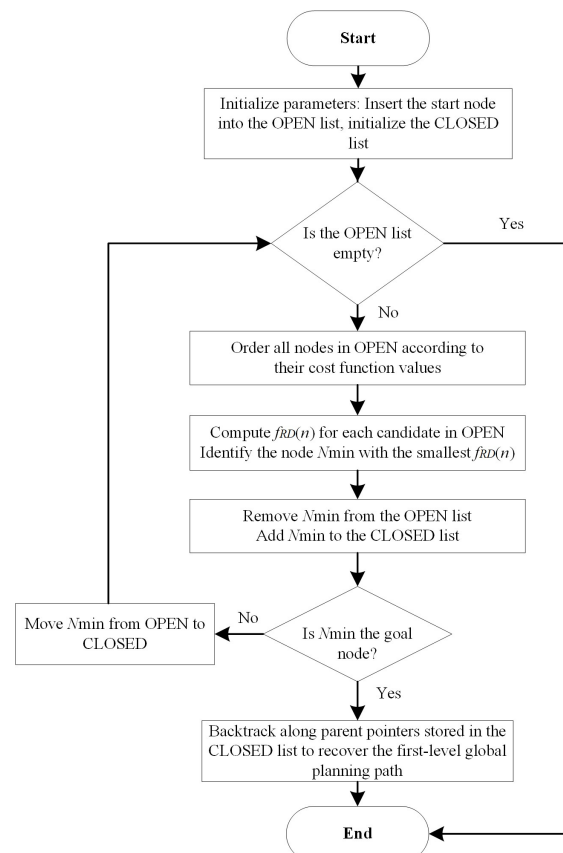


Figure 8. Flowchart of global path-planning with the RDA* algorithm.

4.2. Second-level local path optimization with IDB

After the first-level RDA* algorithm generates adjacent global waypoints, each second-level planning task can be regarded as a bounded continuous optimization problem in which the intermediate waypoint coordinates are adjusted to minimize the local path length and turning angle costs while satisfying the local boundary and obstacle avoidance constraints. The dung beetle optimization (DBO) algorithm [33] is selected for this local trajectory refinement task because its population is divided into several behavioral groups with different position update rules. The cooperative search of rolling, breeding, juvenile, and stealing beetles provides complementary exploration and exploitation mechanisms, which are suitable for searching candidate waypoint sequences within a constrained local domain. In addition, the population-based structure of DBO allows multiple candidate trajectories to be evaluated and progressively refined without requiring gradient information. Nevertheless, the conventional DBO algorithm generally relies on random population initialization, which may result in uneven coverage of the local search domain. During subsequent iterations, the population may also lose diversity and stagnate around a locally optimal trajectory, particularly in local scenarios containing densely distributed obstacles. Therefore, tent chaotic initialization is introduced to improve the spatial coverage and diversity of the initial population, and Lévy flight perturbation is incorporated into the stealing-beetle update to retain occasional long-range exploration and improve the ability to escape from local optima. Thus, DBO is adopted because its population structure is compatible with the continuous local waypoint optimization problem, and the two improvement strategies are introduced to address its limitations in initialization and search diversity. The selection of DBO is based on this structural compatibility rather than on an assumption that it is universally superior to all other metaheuristic algorithms.

4.2.1. Dung beetle (DB) algorithm

Dung beetle optimization (DBO) emulates four bio-inspired behaviors including rolling, foraging, breeding, and stealing. The DBO algorithm divides the population into four behavioral groups: Rolling beetles, breeding beetles, juvenile beetles, and stealing beetles. Each of them adopts a distinctive update strategy that reflects its biological role. With cooperative and competitive interactions, these groups balance global exploration and local exploitation dynamically, thereby enabling the DBO algorithm to solve complex optimization problems efficiently. This DB algorithm therefore comprises the following four beetle types.

The first group consists of rolling beetles. Rolling a dung ball is a typical survival strategy of dung beetles, which serves for egg-laying, reproduction, and food storage. Throughout this process, dung beetles utilize celestial navigation cues, including solar position and wind patterns, to maintain straight-line dung ball trajectories. The mathematical representation of this behavior is given by Eq (25).

$$\begin{cases} x_i(t+1) = x_i(t) + a \times k \times x_i(t-1) + b \times \Delta x, \\ \Delta x = |x_i(t) - X^w|. \end{cases} \quad (25)$$

The position vector $x_i(t+1)$ defines the i th beetle's location at iteration $t+1$. The vector $x_i(t)$ specifies its position at iteration t . The turning amplitude during the update is controlled by a deviation coefficient denoted k , which is restricted to the range $(0, 0.2]$. b is a constant in $(0, 1)$ used to adjust the step size. a takes the value -1 or 1 according to a probability rule, which is used to determine whether

and how the beetle deviates from the original direction. X^w denotes the current worst position found during a global exploration. Δx simulates the influence of light intensity changes during the search process.

When a rolling beetle encounters an obstacle and cannot proceed, it performs a dance behavior to select a new movement direction and obtain an alternative route. The position update rule for this dance behavior is given in Eq (26). In this equation, $\theta \in [0, \pi]$. When θ equals 0, $\pi/2$, or π , the rolling beetle maintains its current heading, and its position is not updated:

$$x_i(t+1) = x_i(t) + \tan(\theta) |x_i(t) - x_i(t-1)|. \quad (26)$$

The second category comprises breeding beetles, whose behavior inspires a boundary selection strategy simulating the oviposition process, where female dung beetles secure dung balls in protected locations to safeguard offspring. This mechanism, formalized in Eq (27), establishes L_λ and U_λ as the lower and upper bounds of the egg-laying region centered around the current local optimum X_{opt} , with the iteration-dependent parameter $R = 1 - t/T_{\text{max}}$ (T_{max} being the maximum iteration count) governing boundary contraction and L and U maintaining the global optimization domain constraints.

$$\begin{cases} L_\lambda = \max \{(1-R) \cdot X_{\text{opt}}, L\}, \\ U_\lambda = \min \{(1+R) \cdot X_{\text{opt}}, U\}. \end{cases} \quad (27)$$

Following egg-laying region determination, the female beetle deposits a single egg within that safe area during each iteration. Because the region is dynamically adjusted as the iterations proceed, the breeding beetle's position is continuously updated. The update rule is given in Eq (28). In this equation, $O_i(t+1)$ specifies the offspring's position at iteration $t+1$. The independent random vectors β_1 and β_2 both possess $1 \times D$ dimensionality, with D defining the optimization search space dimension.

$$O_i(t+1) = X_{\text{opt}} + \beta_1 \times [O_i(t) - L_\lambda] + \beta_2 \times [O_i(t) - U_\lambda]. \quad (28)$$

The third type is the juvenile beetle. The juvenile beetle category comprises mature dung beetles departing secure zones to forage. Simulating this foraging behavior requires establishing an optimal foraging region. The definition of this optimal foraging region is given below. In Eq (29), L_{bound} and U_{bound} delineate the region's lower and upper boundaries. X_{best} represents the global best position. Accordingly, the position update of a juvenile beetle is given by Eq (30):

$$\begin{cases} L_{\text{bound}} = \max \{X_{\text{best}} \cdot (1-R), L\}, \\ U_{\text{bound}} = \min \{X_{\text{best}} \cdot (1+R), U\}. \end{cases} \quad (29)$$

In Eq (30), $x_i(t+1)$ specifies the position of the i th juvenile beetle at iteration $t+1$, where μ_1 is a normally distributed random vector, and μ_2 is uniformly distributed on $(0, 1)$.

$$x_i(t+1) = x_i(t) + \mu_1 \times [x_i(t) - L_{\text{bound}}] + \mu_2 \times [x_i(t) - U_{\text{bound}}]. \quad (30)$$

The fourth type is the stealing beetle. Stealing beetles obtain their food by pilfering resources from other beetle groups, which are modeled as a movement toward the current best food source. Similarly to Eq (29), X_{best} denotes the global best position. The region surrounding X_{best} is regarded as an ideal

venue for resource competition. Consequently, the position update rule for a stealing beetle is given in Eq (31). In this equation, $x_i(t + 1)$ specifies the i th stealing beetle's position at iteration $t + 1$. g is a $1 \times D$ random vector drawn from a normal distribution. S is a constant.

$$x_i(t + 1) = X_{\text{best}} + S \times g \times (|x_i(t) - X_{\text{opt}}| + |x_i(t) - X_{\text{best}}|). \quad (31)$$

Figure 9 shows a random initialization of the four agent groups in the DB algorithm. Each small rectangle represents one beetle. Assuming a total population of 30 individuals, the four groups contain 7, 6, 8, and 9 beetles, respectively.

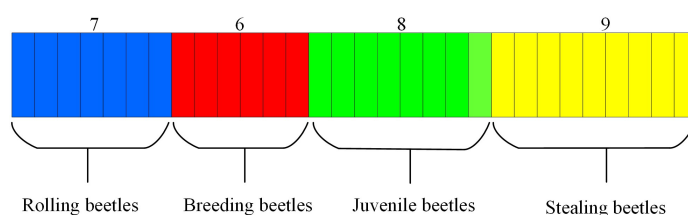


Figure 9. Initial distribution of the four dung beetle groups.

4.2.2. IDB optimization algorithm

The conventional DBO algorithm generally initializes individual positions randomly, which may result in uneven coverage of the search space and insufficient population diversity. This strategy cannot cover the search space completely and often produces an uneven initial distribution which degrades the whole optimization performance greatly.

By contrast, initializing the population with a chaotic map yields a more uniform and diverse distribution, which improves a global exploration capability and overall optimization quality. Also, throughout the search process, local optima tend to trap the stealing beetles, an issue that impairs the algorithm's overall capability.

To solve these problems, an improved dung beetle (IDB) algorithm is proposed by incorporating a tent chaotic map and a Lévy flight strategy in the second-level local path optimization for UAV operations.

The tent map [34] is suitable to produce chaos through a piecewise linear function. The resulting sequence exhibits a strong ergodicity and spatial coverage, enabling a more uniform and diverse distribution of the initial population. As a result, the global search capability of the IDB algorithm could be enhanced.

Accordingly, in this paper, a tent chaotic map for population initialization is adopted, whose mathematical model is given in Eq (32). In this equation, the interval $(0, 1]$ is divided into two sub-intervals and a piecewise linear formulation is applied to produce a chaotic map. When the current iterate x_k is less than or equal to the parameter α ($0 < \alpha < 1$), the mapping is evaluated as $x_{k+1} = x_k/\alpha$. Otherwise, $x_{k+1} = (1 - x_k)/(1 - \alpha)$. Because the function increases or decreases linearly on each segment, it can traverse the entire interval $(0, 1]$ within only a few iterations, which further yields a superior coverage and randomness. As a result, the IDB algorithm obtains a more uniform and diverse population in a random initialization improving a global search performance:

$$x_{i+1} = \begin{cases} \frac{x_k}{\alpha}, & 0 < x_k \leq \alpha, \\ \frac{1 - x_k}{1 - \alpha}, & \alpha < x_k \leq 1. \end{cases} \quad (32)$$

Lévy flight [35] implements a heavy-tailed random walk, alternating short displacements with rare long-distance jumps in its step size distribution. By introducing a Lévy flight strategy into our proposed IDB algorithm, the effective search space is enlarged greatly, and the population diversity would therefore be further enhanced. The local optimum could also be avoided in the end.

In the position update of a stealing beetle, a Lévy flight can be simulated by allowing long-distance jumps with a certain probability. The specific update rule is given in Eq (33). In this equation, *levy* does not represent a fixed numerical value or a predetermined constant; it instead denotes a stochastic perturbation coefficient sampled from a Lévy distribution during each position update, and its value therefore varies among different individuals and iterations. In Eq (33), this coefficient provides a random scaling effect on the current global best position X_{best} . With this update strategy, the stealing beetle alternates short-range local exploration with occasional long-range jumps during the search process. Thereby, the global search capability of the proposed IDB algorithm is enhanced, and the probability of becoming trapped in a local optimum is reduced.

$$x_i(t+1) = \text{levy} \times X_{\text{best}} + S \times g \times (|x_i(t) - X_{\text{opt}}| + |x_i(t) - X_{\text{best}}|). \quad (33)$$

In the proposed IDB algorithm, the fitness function of the j th local path-planning problem is constructed by combining the path length cost and the turning angle cost, as defined in Eq (34). In Eq (34), $J_{D,j}(X_j)$ denotes the fitness value associated with the decision vector X_j in the j th local planning domain. $J_{\text{path},j}(X_j)$ denotes the path length cost of the corresponding interpolated local path and is calculated according to Eq (2), whereas $J_{\text{turn},j}(X_j)$ denotes the turning angle cost constructed from the turning angles defined in Eq (4). The weight coefficients ω_{path} and ω_{turn} determine the relative contributions of the two objective components and satisfy $\omega_{\text{path}} + \omega_{\text{turn}} = 1$:

$$J_{D,j}(X_j) = \omega_{\text{path}} J_{\text{path},j}(X_j) + \omega_{\text{turn}} J_{\text{turn},j}(X_j). \quad (34)$$

$$J_{\text{turn},j}(X_j) = \sum_{s=2}^{N_s-2} |\theta_{j,s+1} - \theta_{j,s}|. \quad (35)$$

In Eq (35), $\theta_{j,s}$ is calculated according to Eq (4), and $J_{\text{turn},j}(X_j)$ represents the accumulated variation between consecutive turning angles.

Figure 10 illustrates the flowchart for second-level local path optimization based on the IDB algorithm. First, all the parameters needed in the IDB algorithm are initialized, and the initial beetle population is generated using a tent chaotic map. Second, all the beetle groups, including rolling, breeding, juvenile, and stealing beetles, update their positions according to their respective strategies. The fitness values are recalculated, and the iteration counter is incremented with $t = t + 1$. As a third step, a check is performed by the IDB algorithm to see whether the termination condition is true. Should it be true, the best solution is emitted, after which the algorithm comes to a stop. If it is not, the algorithm continues to iterate. Finally, the IDB algorithm returns the fitness value and the optimal path with every waypoint.

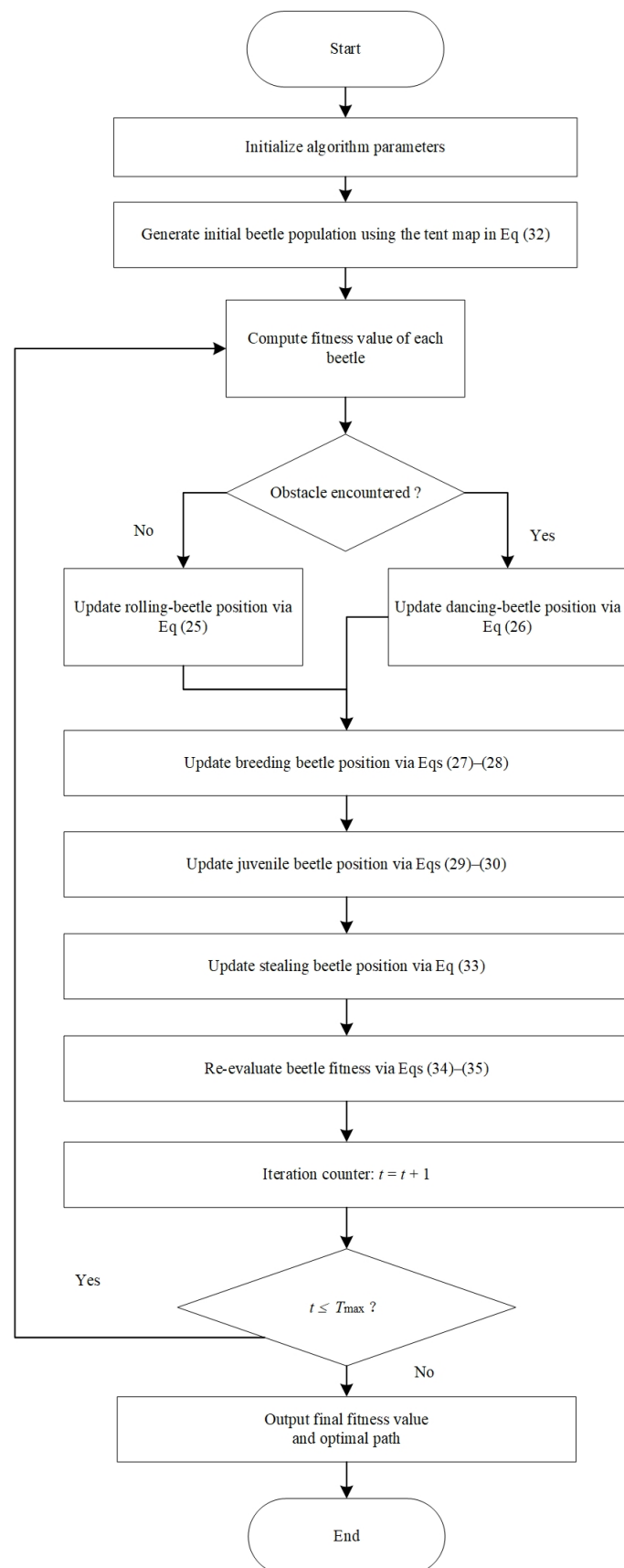


Figure 10. Flowchart of local path-planning with the IDB algorithm.

In the IDB-based local optimization, each dung beetle individual represents an ordered sequence of three-dimensional intermediate waypoints between two fixed adjacent global waypoints. In the present implementation, each local path contains five intermediate waypoints, and the position of the i th individual is encoded as

$$\mathbf{X}_i = [x_{i,1}, y_{i,1}, z_{i,1}, \dots, x_{i,5}, y_{i,5}, z_{i,5}],$$

giving a decision space dimension of $D = 3n = 15$, where $n = 5$. The local start and goal positions are fixed and are not treated as decision variables. After an individual is decoded, the local start point, the five intermediate waypoints, and the local goal point are sequentially connected to form a complete candidate path. The fitness value of each candidate is evaluated using the path length and turning angle costs defined in (34).

The tent map parameter is set to $\alpha = 0.5$. The population size is set to 30, and the perturbation coefficient is restricted to $k \in (0, 0.2]$. After each position update, decision variables outside the normalized interval $[0, 1]$ are processed using periodic boundary wrapping rather than clipping, reflection, or random regeneration. The updated variables are then mapped to the spatial limits of the corresponding local planning domain. When a mapped local start or goal position does not satisfy the obstacle height requirement, a limited adjustment is performed by searching for a feasible neighboring position.

For each decoded candidate solution, cubic spline interpolation is applied to the local start point, the five intermediate waypoints, and the local goal point to generate 500 discrete path samples. This interpolation is incorporated into each fitness evaluation rather than being applied as an independent post-optimization smoothing procedure. The path length and turning angle costs, together with the obstacle height and local boundary feasibility conditions, are evaluated using these interpolated samples. Therefore, the feasibility evaluation is performed on the 500 sampled points along the interpolated path rather than only on the five intermediate waypoints. Because the implementation uses high-density discrete sampling, it does not constitute an exact continuous geometric intersection test for every path segment.

Infeasible candidate paths are handled through bounded coordinate encoding, limited endpoint adjustment, and sample-based feasibility evaluation. Candidate paths containing sampled points that violate the obstacle height or local boundary conditions are treated as infeasible and are not accepted as the final local path. No complete repair procedure is applied to reconstruct an entire infeasible candidate path.

The spline-interpolated path evaluated during optimization is also used as the final local path. After the optimal individual is decoded, no additional smoothing or waypoint adjustment operation is applied. Consequently, the geometry of the final output path is identical to that subjected to the sample-based feasibility evaluation during optimization. To maintain consistency with the environmental representation, the local path coordinates are transformed back to the global coordinate system according to the predefined spatial origin, scale, and resolution conversion relationship of the corresponding local planning domain. The transformed path therefore remains spatially aligned with the fine-resolution obstacle and risk maps.

After the different dung beetle groups update their positions according to their respective search operators, the updated candidate solutions are evaluated and compared with the best solution obtained so far. The current best individual is retained throughout the iterative process. The optimization terminates when the iteration counter reaches the prescribed maximum number of 200, after which the

best individual is decoded as the optimized local path. No additional fitness tolerance or consecutive-no-improvement stopping criterion is employed in the present simulations.

To provide an explicit mathematical characterization of the convergence property of IDB, the analysis is formulated in terms of the best-so-far fitness sequence retained during the iterative process. Let $X_i(t)$ denote the position of the i th individual at iteration t , let N_p denote the population size, and let $F_{\text{best}}(t)$ denote the minimum fitness value retained up to iteration t . The definition of the best-so-far fitness and its recursive update relationship are given as follows:

$$F_{\text{best}}(t) = \min_{0 \leq \tau \leq t} \min_{1 \leq i \leq N_p} J_D(X_i(\tau)),$$

$$F_{\text{best}}(t+1) = \min \left\{ F_{\text{best}}(t), \min_{1 \leq i \leq N_p} J_D(X_i(t+1)) \right\} \leq F_{\text{best}}(t). \quad (36)$$

Because the path length and turning angle costs are non-negative, and their corresponding weights are non-negative, the local fitness function is bounded below by zero. Therefore, the retained best-fitness sequence satisfies the following relationship:

$$0 \leq F_{\text{best}}(t+1) \leq F_{\text{best}}(t), \quad \lim_{t \rightarrow \infty} F_{\text{best}}(t) = \inf_{t \geq 0} F_{\text{best}}(t) = F_{\infty} \geq 0. \quad (37)$$

Equations (36) and (37) show that the best-so-far fitness sequence generated by IDB is monotonically nonincreasing and bounded below and therefore converges to a finite limit. This result establishes the convergence of the retained best fitness value, but it does not guarantee that the limiting value F_{∞} is the global optimum. The practical convergence speed and solution reliability of IDB are further evaluated through the repeated paired simulations presented in Section 5.3.2.

5. Numerical simulations and result analysis

To evaluate the proposed hierarchical risk-aware UAV path-planning framework, numerical simulations were conducted using MATLAB R2021b on a workstation equipped with an Intel Core i9-13900HX processor operating at up to 5.4 GHz and running Windows 11. The theoretical formulations and hierarchical planning framework described in Sections 2–4 were implemented in the considered urban simulation scenario. The flight region and obstacle avoidance conditions determined the feasible search space, and the normalized integrated risk values assigned to the three-dimensional airspace voxels provided the environmental risk input. The risk map resolution conversion procedure is used to generate the coarse-resolution representation required by the first-level global planner.

At the first planning level, the RDA* algorithm operates on the coarse-resolution risk map and generates a global waypoint sequence by jointly considering accumulated path risk and path distance costs. Each pair of adjacent global waypoints subsequently defines a fine-resolution local planning domain. At the second planning level, the IDB algorithm optimizes the intermediate waypoints within each local domain by considering local path length and turning angle costs, and the fine-resolution obstacle representation is used to perform sample-based collision feasibility evaluation. The locally optimized path segments are then sequentially connected to form the complete UAV path. Finally, path length, accumulated path risk cost, unit path average risk value (UARV) and UARV reduction rate are used to evaluate the resulting paths.

5.1. Scenario and parameter settings

An area with 25 km² in Changqing District, Jinan, China was selected as a real-world scene for UAV operations, whose longitude and latitude range from 116.75°E to 116.81°E and from 36.55°N to 36.60°N, respectively. Figure 11 shows the satellite image of the selected area at a 1-meter resolution. Several representative urban ground features are distributed throughout this area. The lower-left quadrant is dominated by residential neighborhoods with a dense and highly regular housing layout. The central and upper-right quadrants consist of industrial and commercial mixed zones containing numerous factory buildings, commercial structures, and associated road infrastructure. The lower-right quadrant is characterized by green spaces and undeveloped land with rivers and other water bodies. A well-connected and dense road network links these functional zones, reflecting the diversity and complexity of a typical urban environment. Although the risk map is constructed using real-world urban spatial data, it is treated as a static preflight risk snapshot during each planning process. Time-varying pedestrian density, traffic flow, and event-related risks are not considered in the current study.

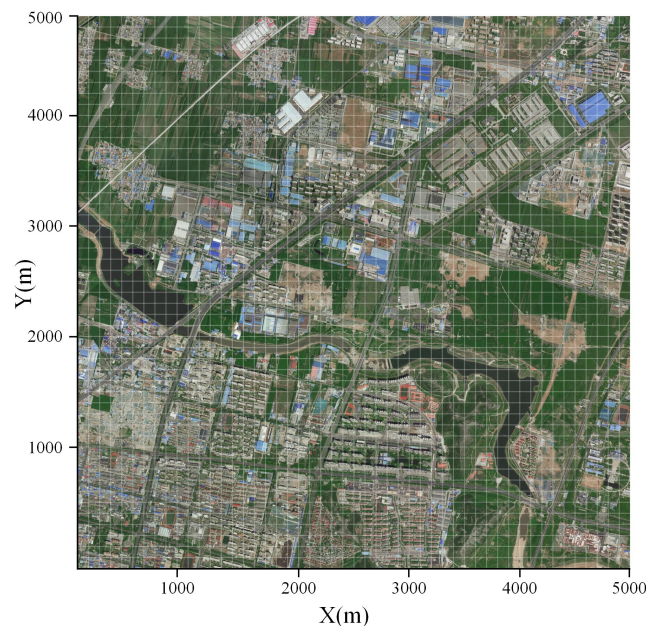


Figure 11. Satellite image of the study area.

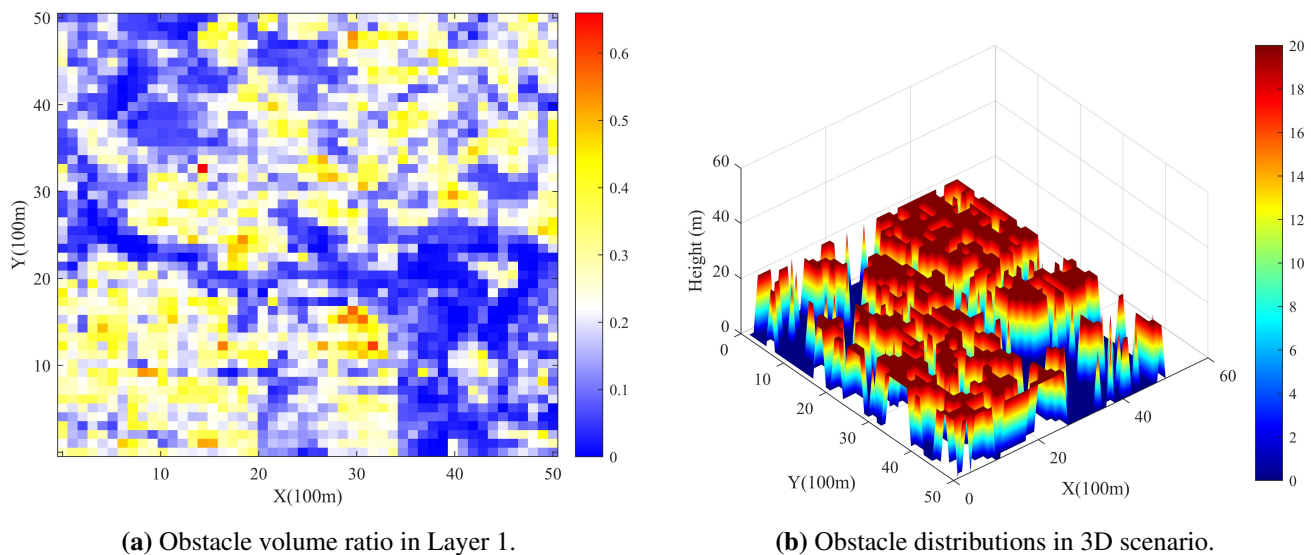
A DJI M210 RTK UAV was employed for the experimental verifications. This UAV's mass, length, width, and frontal area are 4.27 kg, 0.887 m, 0.880 m, and 0.234 m². The crash probability P_{crash} used in Eqs (9) and (12) was set as 3.42×10^{-4} events per hour.

Table 2 presents the key parameters used for urban risk assessment caused by UAV operations. The air density ρ_A , drag coefficient C_D , and ground-vehicle speed v_{car} are set to 1.225 kg/m³, 0.3, and 16 m/s, respectively. Another two constants, α and β , were set as 10⁶ J and 100 J.

Table 2. Parameters for UAV's risk assessment.

Parameter	Value
g	9.8 m/s ²
ρ_A	1.225 kg/m ³
C_D	0.3
v_{car}	16 m/s
α	10 ⁶ J
β	100 J

The scenario of first-level global path guidance with RDA* is discretized at a resolution of $50 \times 50 \times 5$. In this way, the size of each voxel is 100 m $\times$ 100 m $\times$ 20 m. The average ratio of obstacle and volume is 0.0384. Figure 12(a) shows the spatial ratio distributions of obstacle and volume in Layer 1, whose average value is 0.1835. It reveals that the high ratio regions are mainly concentrated in the lower-left quadrant, which are some dense low-rise residential zones. Several elevated values also appear in the upper-right quadrant, which corresponds to an industrial area. The maximum ratio occurs at a coordinate of (32, 15), located within a high-rise district. The mean ratio of obstacle and volume in Layer 2 is 0.0084, whereas Layers 3 and above contain no obstacles yielding to ratio values of zero. On the basis of this analysis, the threshold θ_o is set to 0.2, which produces a first-level global path guidance scenario for UAVs as given in Figure 12(b).

**Figure 12.** First-level global path guidance scenario with RDA*.

Additional parameters for the first-level global path guidance scenario are listed in Table 3. These include the search space, start node, goal node, risk map, obstacle matrix, and the weight coefficients for risk and distance cost. These settings together define the baseline of first-level global path guidance with RDA* for UAV operations.

Table 3. Parameters for the global path-planning scenario.

Parameter	Symbol	Value
Search space	map	$50 \times 50 \times 5$
Start node	G_{start}	(1, 1, 1)
Goal node	G_{end}	(50, 50, 5)
Risk map	C_u	[0, 1]
Obstacles	f_o	0 or 1
Risk cost weight	α_R	0.6
Distance cost weight	α_D	0.4

The scenario of second-level local path optimization with IDB is generated from the edge-adjacent nodes obtained in the first-level global path guidance with RDA*. As shown in Figure 13, the size of second-level local path optimization scenario is $200 \text{ m} \times 200 \text{ m} \times 20 \text{ m}$, which consists of four coarse voxels. The obstacle distributions in this scenario are relatively complex, which stands for a real urban environment.

In the figure below, pink and green markers denote the given start node and goal node, respectively. Initially, both nodes are set at the voxel center with their own coordinates and then mapped to the actual scenario. If this mapped position lies inside an obstacle, a horizontal search is performed to locate the nearest feasible coordinate. For example, the start node is ultimately placed at (42, 48, 10) because the original center coordinate (50, 50, 10) was detected as an obstacle and had to be adjusted.

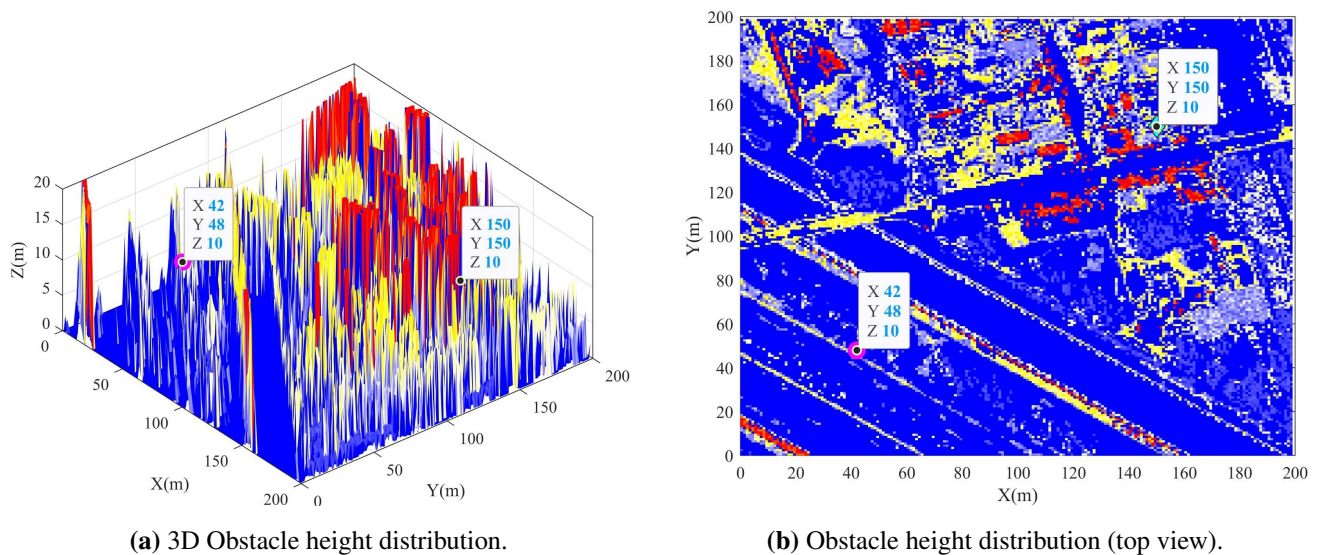


Figure 13. IDB-based local path-planning scenario: (a) three-dimensional view; (b) top view.

Other additional parameters for the second-level local path optimization scenario with IDB are listed in Table 4. Take the largest second-level local path optimization scenario as an example with a size of $200 \text{ m} \times 200 \text{ m} \times 40 \text{ m}$. The start node G_{start} and goal node G_{end} are set to (50, 50, 10) and (150, 150, 30), respectively. The maximum number of iterations T_{max} is fixed at 200 to ensure a

convergence within a reasonable runtime. The deviation coefficient k lies in $(0, 0.2]$, which introduces some controlled stochastic perturbations during the path updates. The obstacle parameter Barrier has a height range of $(0, 37.47]$. The population size Pop is set to 30, providing sufficient diversities during the search.

Table 4. Parameters for local path-planning scenario.

Parameter	Symbol	Value
Search space	map	$200 \times 200 \times 40$
Start node	G_{start}	$(50, 50, 10)$
Goal node	G_{end}	$(150, 150, 30)$
Iterations	T_{max}	200
Deviation coefficient	k	$(0, 0.2]$
Obstacle height	$Barrier$	$(0, 37.47]$
Population size	Pop	30

5.2. Risk analysis of UAV operations

According to the risk model defined above, the fatality risk and property damage risk weights are set to $\alpha_1 = 0.65$ and $\alpha_2 = 0.35$, respectively. These values follow the safety-oriented weighting strategy adopted in our previous urban UAV risk-assessment study [31] and assign greater importance to consequences involving human life. The same coefficients are used to construct the fixed three-dimensional risk map supplied to all compared planning algorithms, thereby ensuring a consistent environmental input. These coefficients should be interpreted as model calibration parameters for the considered urban risk map rather than universally optimal constants. Because the present study focuses on hierarchical path planning and treats the risk map as a fixed environmental input, a separate sensitivity analysis of α_1 and α_2 is not conducted here. Consequently, their transferability to urban environments with different population, traffic, and building distributions has not yet been established and requires further investigation. The normalization factors ω_1 and ω_2 are set to 2.2943×10^4 and 10.3263, respectively.

The modelling roles, principal dependencies, and expected transferability of the key risk and planning parameters are summarized in Table 5.

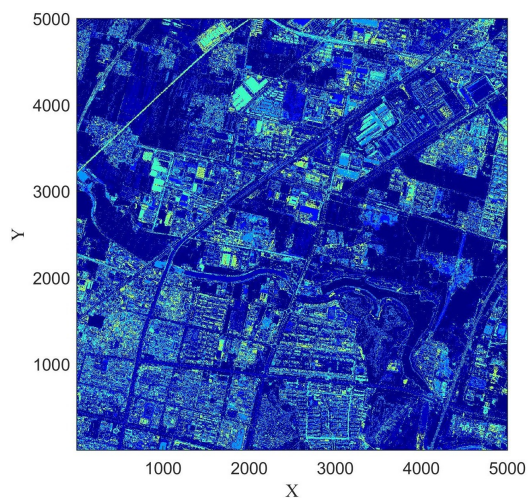
As summarized in Table 5, the numerical parameter settings adopted in this study are associated with the considered urban scenario, UAV platform, and mission preferences. By contrast, the hierarchical optimization formulation, risk normalization procedure, and coarse-to-fine waypoint-to-domain mapping mechanism are not restricted to these particular numerical values and can be transferred to other urban environments after the relevant parameters are recalibrated.

Table 5. Sensitivity and transferability of key risk and planning parameters.

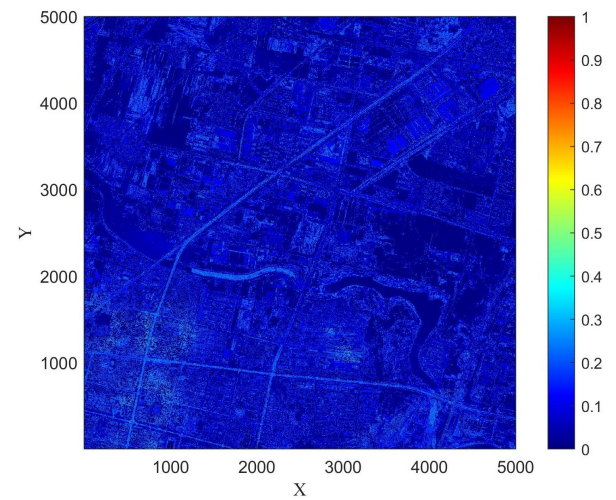
Parameter	Current setting	Principal dependence	Transferability and recalibration
α_R, α_D	0.6, 0.4	Determine the balance between global path-risk cost and distance cost; their effects depend on the urban risk distribution, obstacle configuration, start-goal pair, and mission preference.	Scenario-specific operating parameters. They should be recalibrated through risk distance sensitivity analysis when the planning task or environment changes.
α_1, α_2	0.65, 0.35	Determine the relative contributions of fatality risk and property damage risk and are affected by population, traffic, building distributions, and risk management priorities.	Not universal constants. They should be recalibrated according to local urban data and the relative priority assigned to human safety and property protection.
θ_o	0.2	Determines whether a coarse voxel is classified as an obstacle and depends on voxel resolution, obstacle height distribution, urban morphology, and planning conservatism.	Should be adjusted when the coarse grid resolution, obstacle distribution, or required safety margin changes.
P_{crash}	$3.42 \times 10^{-4} \text{ h}^{-1}$	Depends mainly on UAV platform reliability, maintenance condition, operating environment, and the adopted failure assumptions.	Transferable only to comparable UAV platforms and operating conditions; platform-specific reliability data should be used when available.
$\omega_{\text{path}}, \omega_{\text{turn}}$	0.8, 0.2	Determine the balance between local path efficiency and turning angle variation and are influenced by UAV maneuverability and mission requirements.	Mission- and platform-dependent parameters that should be adjusted according to vehicle dynamics and local trajectory quality priorities.

The generated risk map of UAV operations is shown in Figure 14. The risk distributions of different layers are given, which are Layer 5 ($h = 5 \text{ m}$), Layer 25 ($h = 25 \text{ m}$), Layer 45 ($h = 45 \text{ m}$), Layer 50 ($h = 50 \text{ m}$), Layer 75 ($h = 75 \text{ m}$), and Layer 100 ($h = 100 \text{ m}$).

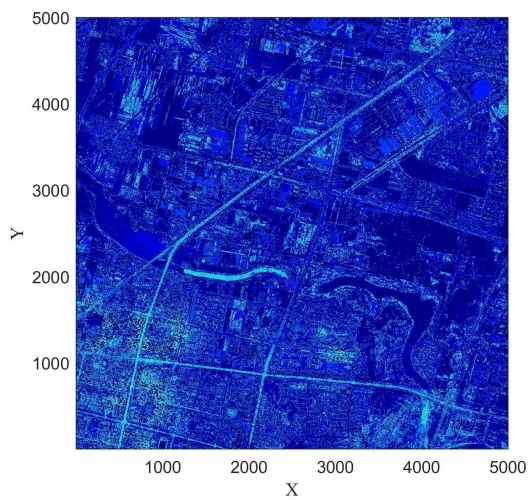
Figure 14(a) displays the risk distributions when the UAV is operating at a height of 5 m above the ground, whose average risk value is 0.131. As the flight altitude increases, the average risk value of Layer 25 decreases to 0.065, corresponding to a reduction of approximately 50.38% compared with Layer 5, which is given in Figure 14(b). The average risk value of Layer 45 shown in Figure 14(c) is 0.095. That is because no buildings can reach 45 m in the selected urban area. In this way, only fatality risk exists. The average risk value of Layer 50 is 0.102, which is 7.37% higher than that of Layer 45. Layer 75 has an average risk of 0.125 and is 22.55% higher than the Layer 50. The average risk of Layer 100 is 12.80% higher than Layer 75, with a value of 0.141.



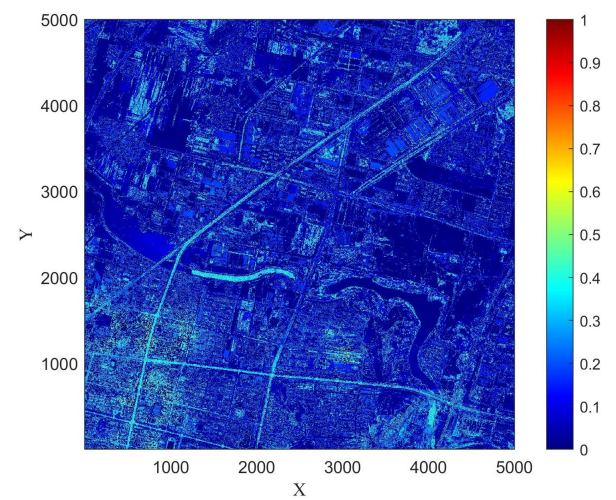
(a) Risk distribution in Layer 5.



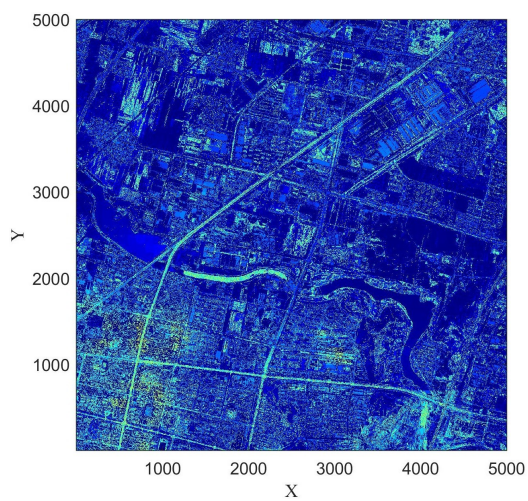
(b) Risk distribution in Layer 25.



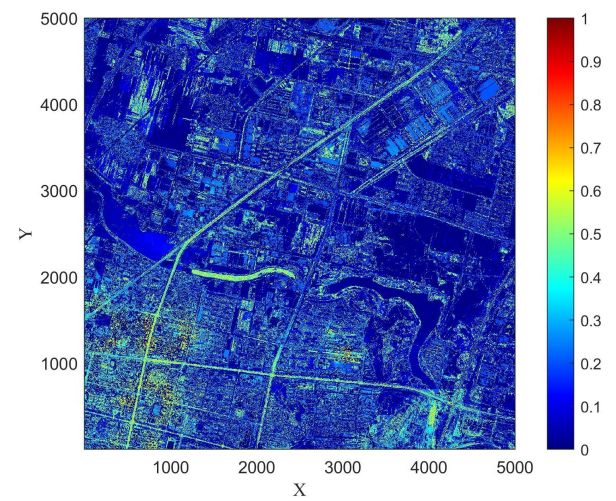
(c) Risk distribution in Layer 45.



(d) Risk distribution in Layer 50.



(e) Risk distribution in Layer 75.



(f) Risk distribution in Layer 100.

Figure 14. Risk distributions of UAV operations at different flight heights.

From the risk distributions at different flight heights, it can be concluded that there is more risk in densely populated residential areas and road networks on the ground. The lower-left quadrant shows the highest risk values, which are around 0.7. The risk level of road corridors is relatively high, with a value approximately 0.5. By contrast, the risk values of lakes, water bodies, parks, and green spaces are with values near zero and show a distinctly blue appearance.

Figure 15 shows the numerical risk analysis of UAV operations. Figure 15(a) presents the average risk values across different layers, and Figure 15(b) demonstrates the risk differentials between adjacent layers, calculated as the difference between upper-layer and lower-layer mean risk values. Across the entire study area, the mean urban risk is 0.10612. As the flight height increases, there is a downtrend in the overall risk, and then it becomes a rising trend. Specifically, the average risk from Layer 1 to Layer 14 changes from 0.09439 to 0.13113, where the high building density is the main factor that increases the property damage risk. From Layer 15 to Layer 55, the average risk values are between 0.05871 and 0.08982, and the minimum value is 0.05871, occurring in Layer 21. From Layer 56 to Layer 100, the average risk value of each layer is between 0.10846 and 0.14021. It proves that a fallen UAV from higher heights would produce a higher fatality risk, which leads to more integrated risk. According to the data given in Figure 15(a), a UAV will introduce less risk when operating between Layer 15 and Layer 55, that is, between the height 15 m and 55 m.

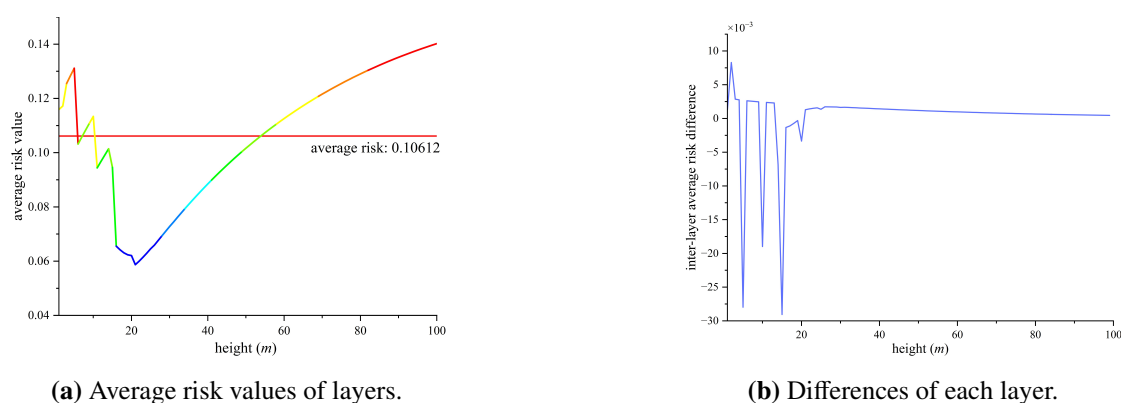


Figure 15. Numerical risk analysis of UAV operations.

5.3. Risk-constrained hierarchical UAV path planning

5.3.1. First-level global path guidance with RDA*

To verify the effectiveness of the proposed RDA* algorithm in the first-level global path guidance, the comparisons are made first between a traditional distance cost A* algorithm and a risk cost A* algorithm, which is given in Figure 16. In the first-level global path guidance scenario, the red voxels denote some high-risk cells. There are 50 waypoints on the generated UAV path by the traditional distance cost A* algorithm with a distance cost of 7271.77 and a risk cost of 2946.16. The risk cost A* generated UAV path is composed of 79 waypoints with a distance cost of 8051.08 and a risk cost of 2192.75. The differences between these two strategies are obvious: The path generated by the distance cost A* algorithm is nearly a straight line that traverses multiple high-risk regions, whereas the path given by the risk cost A* algorithm bends to avoid many of these high-risk voxels. Although the risk cost A* path introduces a longer distance, it reduces the path risk markedly. The path length is

increased by 10.72%, but the path risk is reduced by 25.57%.

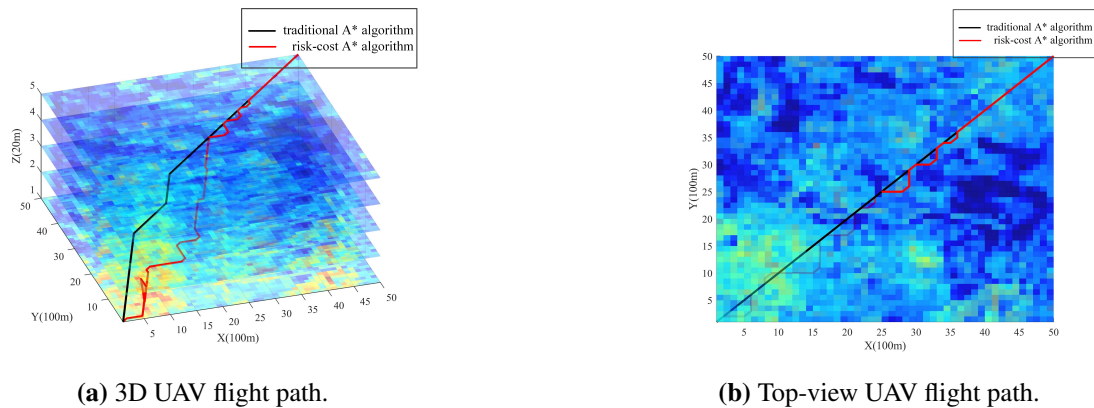


Figure 16. Comparisons of distance-cost A* and risk-cost A*.

Next, a scenario-specific parameter sensitivity analysis is conducted to investigate the influence of the risk weight coefficient α_R on the trade-off between path risk and path distance. Seven candidate values of α_R are examined under the current urban risk map and start-goal configuration. The purpose of this analysis is to select an appropriate operating weight for the subsequent simulations in the considered scenario rather than to identify a universally optimal or directly transferable coefficient. Figure 17 presents the paths generated under the five candidate settings. The black path with $\alpha_R = 0.2$ is the shortest and contains 52 waypoints. The green path with $\alpha_R = 0.4$ provides a moderate balance between risk and distance and contains 57 waypoints. The red path with $\alpha_R = 0.5$, in which risk and distance are assigned equal weights, contains 64 waypoints. The yellow path with $\alpha_R = 0.6$ places greater emphasis on risk avoidance and contains 67 waypoints. Finally, the blue path with $\alpha_R = 0.8$ avoids more high-risk voxels but produces a more tortuous route, containing 75 waypoints.

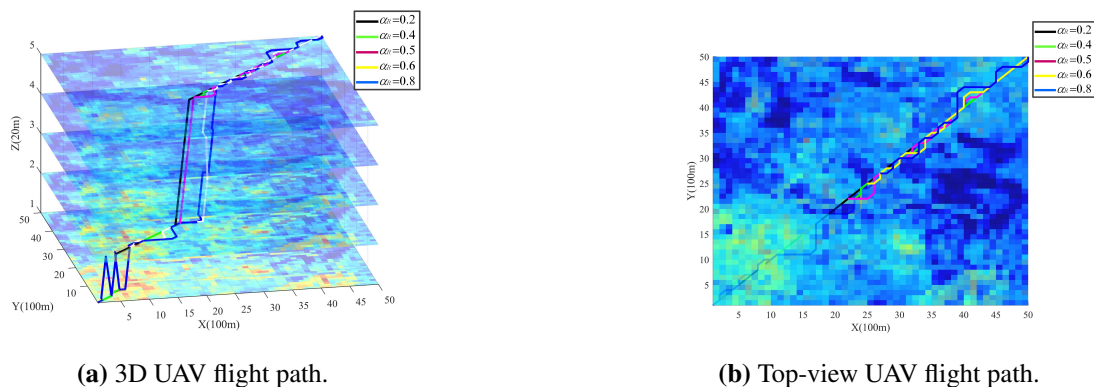


Figure 17. RDA* generated UAV paths with different α_R settings.

Table 6 summarizes the characteristics of the UAV paths generated by the proposed RDA* algorithm under different values of α_R , and the conventional distance-based A* algorithm corresponds to $\alpha_R = 0$. As α_R increases, the generated path generally exhibits a higher distance cost but a lower risk cost, demonstrating a clear trade-off between flight efficiency and risk avoidance. For example, when $\alpha_R = 0.6$, the distance cost increases from 7271.77 to 7937.88, corresponding to an increase of approximately 9.1%, whereas the risk cost decreases from 2946.16 to 2271.71, corresponding to a

reduction of approximately 22.9%. When α_R is further increased to 0.8, the path avoids more high-risk voxels but becomes considerably more tortuous, indicating that excessive emphasis on risk may lead to a disproportionate increase in path distance.

Based on the safety-oriented objective of the present study and the quantitative trade-off observed in the considered scenario, $\alpha_R = 0.6$ and $\alpha_D = 0.4$ are selected as the operating weight setting for the subsequent simulations. This selection is not an arbitrary assignment, as it is supported by the sensitivity analysis of seven candidate weight settings. Compared with the traditional distance-based A* algorithm, this weight combination reduces the risk cost by 22.9% while introducing a comparatively moderate increase of 9.1% in the distance cost. Although the operating weights are selected based on the current urban risk map, obstacle distribution, and original start–goal configuration, their robustness under different planning tasks is further examined through the multi-scenario simulation presented below. Therefore, $\alpha_R = 0.6$ and $\alpha_D = 0.4$ are regarded as an appropriate operating weight combination for the present study rather than a universally optimal coefficient pair.

Table 6. Path analysis of RDA* with different α_R settings.

α_R	0	0.2	0.4	0.5	0.6	0.8	1.0
Total waypoints	50	52	57	64	67	75	79
Total distance cost	7271.77	7397.27	7545.23	7761.58	7937.88	7984.36	8051.08
Distance ratio (vs. A*)	100%	101.7%	103.7%	106.7%	109.1%	109.8%	110.7%
Total risk cost	2946.16	2729.57	2609.09	2386.81	2271.71	2224.36	2192.75
Risk ratio (vs. A*)	100%	92.6%	88.6%	81.1%	77.1%	75.5%	74.4%

To examine whether the risk reduction performance of RDA* depends on the original deterministic start–goal pair, an additional Monte Carlo evaluation is conducted using 30 randomly generated start–goal pairs within the same urban area. The start and goal nodes are sampled from collision-free flight cells, and a minimum horizontal distance constraint is imposed to exclude trivial short-distance planning tasks. For all 30 scenarios, A* and RDA* use the same urban risk map, grid representation, start–goal pairs, flight constraints, and parameter settings. The operating weights of RDA* are fixed at $\alpha_R = 0.6$ and $\alpha_D = 0.4$. The statistical results are summarized in Table 7.

Table 7. Performance comparison of A* and RDA* over 30 randomly generated start–goal scenarios.

Metric	A*	RDA*	Statistical result
Planning success rate	100.00%	100.00%	–
Path length (m)	3422.79 ± 1041.35	3441.25 ± 1044.43	$p = 0.000001732216233$
Unit path average risk value	0.1801 ± 0.0523	0.1193 ± 0.0309	$p = 0.000001734397628$
UARV reduction rate relative to A*	–	$32.51\% \pm 11.44\%$	–
Path length change rate relative to A*	–	$0.56\% \pm 0.71\%$	–

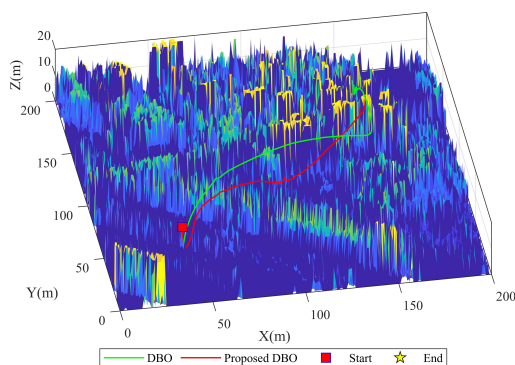
Both A* and RDA* successfully generate feasible paths in all 30 randomly generated scenarios, resulting in a planning success rate of 100%. Compared with A*, RDA* reduces the unit path average risk value from 0.1801 ± 0.0523 to 0.1193 ± 0.0309 , corresponding to an average risk reduction

of 32.51%. The paired Wilcoxon signed-rank test confirms that the reduction in the unit path average risk value is statistically significant, with $p = 0.00000173$. Meanwhile, the mean path length increases by only 0.56%, from 3422.79 ± 1041.35 m to 3441.25 ± 1044.43 m. These results indicate that the UARV reduction performance of RDA* is not restricted to the original deterministic start–goal pair and remains stable across different planning tasks within the tested urban area. Because the random scenarios are generated within one urban area, the present results demonstrate within-area robustness; validation across urban environments with different spatial structures will be considered in future work.

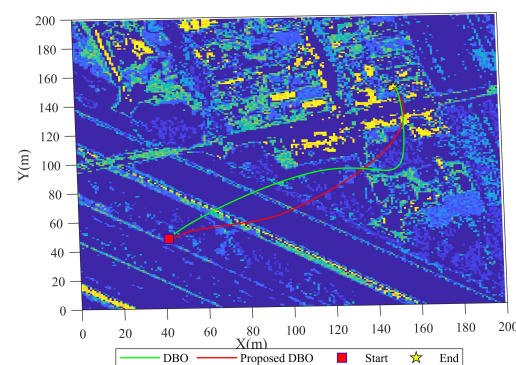
In the first-level global path guidance with RDA*, $\alpha_R = 0.6$ is identified as an appropriate balance between the path risk cost and path distance cost, producing a total of 67 waypoints.

5.3.2. Second-level local path optimization with IDB

Based on the proposed hierarchical UAV path planning scenario model, 67 waypoints would generate 66 second-level local planning scenarios. One of the second-level local path optimization scenarios is given in Figure 18, which is formed by the coarse-voxel coordinates (1, 1, 1) and (2, 2, 1). The size of this scenario is 200 m $\times$ 200 m $\times$ 20 m, with the coordinates of start node and goal node (42, 48, 10) and (150, 150, 10), respectively. The green path represents the route computed by the DB algorithm, and the red path is produced by the proposed IDB algorithm in this paper. The IDB algorithm generates a smoother trajectory that is easier for the UAV to follow, indicating improved trajectory smoothness under the considered local planning scenario.



(a) 3D UAV flight path.



(b) Top-view UAV flight path.

Figure 18. Comparison of local paths generated by DB and IDB: (a) three-dimensional view; (b) top view.

To complement the best-so-far convergence relationships established in Eqs (36) and (37), the practical convergence performance and statistical reliability of the proposed IDB algorithm were further evaluated over 30 paired independent runs. DB and IDB were evaluated under the same population size, maximum number of iterations, objective function, and search bounds. Within each paired run, the two algorithms used the same random seed to reduce the influence of differences in initial random conditions. The simulation results are reported as the mean and standard deviation, and the statistical differences were examined using the two-sided Wilcoxon signed-rank test at a

significance level of 0.05.

The final best fitness values obtained by DB and IDB were 229.4579 ± 34.0121 and 204.9330 ± 11.9448 , respectively. The Wilcoxon signed-rank test indicated that IDB obtained significantly lower final fitness than DB ($p = 4.20 \times 10^{-4}$), demonstrating better final solution quality and lower run-to-run variability.

Because convergence defined relative to each algorithm's own terminal fitness may incorrectly regard premature stagnation at a poorer solution as rapid convergence, an additional common target analysis was conducted. The common fitness target was set to $f \leq 220$, and runs that did not reach the target within 200 iterations were assigned a penalized hitting time of 201. As shown in Figure 19, IDB reached the common target in 28 of the 30 runs, corresponding to a target attainment rate of 93.33%, whereas DB reached the target in only 13 runs, corresponding to a target attainment rate of 43.33%. The difference was statistically significant according to the exact McNemar test ($p = 6.10 \times 10^{-5}$).

The penalized hitting iterations of DB and IDB were 187.23 ± 27.47 and 64.73 ± 67.80 , respectively. The Wilcoxon signed-rank test confirmed that IDB reached the common fitness target in significantly fewer iterations than DB ($p = 7.25 \times 10^{-6}$), with a paired rank biserial effect size of 0.9704. Furthermore, among the 13 paired runs in which both algorithms reached the target, DB and IDB required 169.23 ± 34.70 and 55.00 ± 58.85 iterations, respectively, and the difference remained statistically significant ($p = 0.00122$). These results demonstrate that IDB reaches the same high-quality fitness level more rapidly and reliably than the traditional DB algorithm.

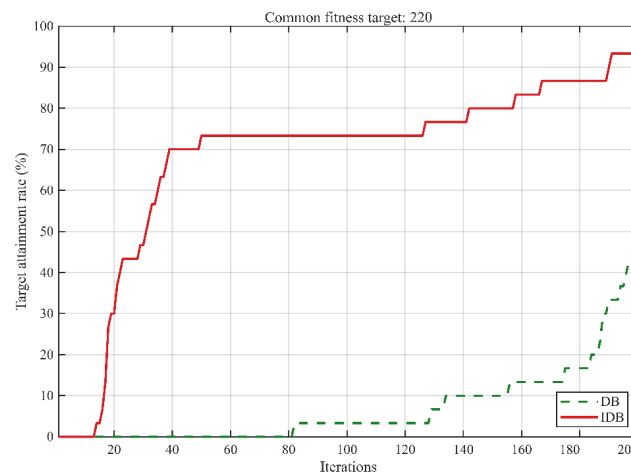


Figure 19. Cumulative target attainment rates of DB and IDB under the common fitness target $f \leq 220$ over 30 paired independent runs.

Figure 20 shows a complete path generated by the proposed hierarchical UAV path planning mechanism, including a first-level global path guidance with RDA* and a second-level local path optimization with IDB. In this figure, the green circles that are connected by black lines represent the first-level global waypoints produced by RDA* algorithm, and the red line denotes a second-level local path with a higher resolution that is generated by the IDB algorithm. The second-level IDB path refines the coarse global path segment at a higher spatial resolution by optimizing the local path length and turning angle costs under the fine-resolution obstacle constraint. Therefore, the second level primarily performs geometric trajectory refinement, whereas the risk-aware guidance of the complete

path is provided by the first-level RDA* planner.

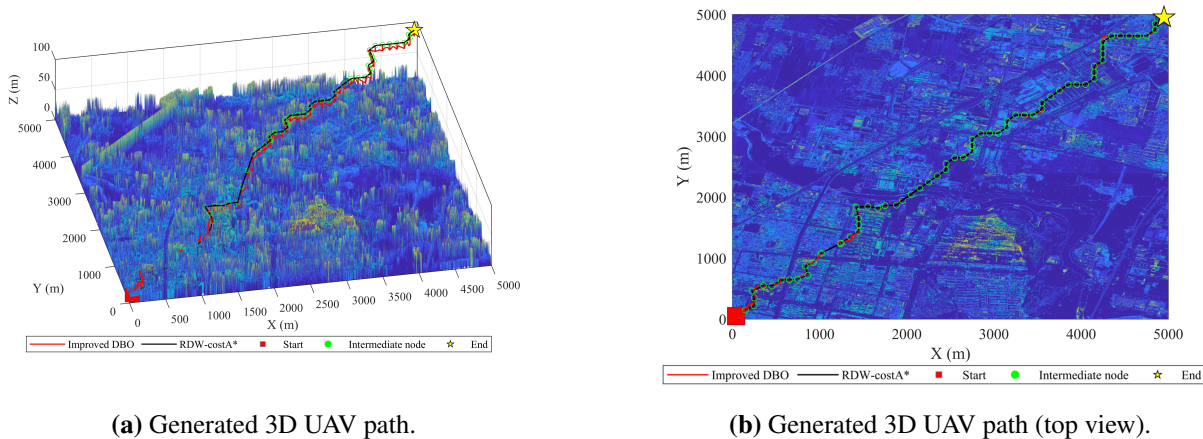


Figure 20. UAV path generated by hierarchical planning mechanism.

5.4. Performance evaluation metrics and result analysis

To compare the single-scenario performance of the proposed RDA*-IDB framework with representative path-planning methods, A*, RRT*, and PSO were selected as comparative algorithms from graph-based search, sampling-based planning, and population-based optimization, respectively. This controlled comparison assesses the proposed hierarchical global-to-local planning framework under common simulation conditions and is intended to provide a comparison with selected representative baselines rather than an exhaustive evaluation against all existing risk-aware path-planning approaches.

For the overall comparison, an additional single simulation scenario was designed with the start node (1, 1, 1) and the goal node (50, 50, 5). All compared algorithms were evaluated using the same start and goal positions, three-dimensional risk map, obstacle environment, flight boundaries, and performance metrics.

Because RRT*, PSO, and IDB employ different search mechanisms, mechanically assigning identical numerical parameter values or iteration numbers would not necessarily constitute a meaningful comparison. Therefore, algorithm-specific parameters were selected according to their respective search procedures and were kept fixed throughout the final comparison. For RRT*, the maximum number of iterations, expansion step size, goal-sampling probability, and neighborhood radius were set to 4000, 4, 0.20, and 7, respectively. For PSO, the population size and maximum number of iterations were set to 40 and 180, respectively, with an inertia weight of $w = 0.72$ and acceleration coefficients of $c_1 = c_2 = 1.49$. For IDB, the population size and maximum number of iterations were set to 30 and 200, respectively.

The stochastic performance of the local optimizer was evaluated separately through the 30 paired independent runs reported in Section 5.3.2. By contrast, the full-framework comparison with A*, RRT*, and PSO was conducted using one fixed random seed and fixed algorithm-specific operating parameters. Because the numerical performance of stochastic path-planning algorithms is parameter-dependent, these parameter settings are not claimed to be globally optimal, and exhaustive hyperparameter optimization is not performed separately for every baseline algorithm. Accordingly,

the results in Table 8 should be interpreted as an illustrative and reproducible comparison under one controlled scenario and the stated parameter settings rather than as repeated-run statistical evidence or a definitive ranking of the compared algorithms under all possible parameter configurations.

Table 8. Illustrative comparison of path-planning performance under the controlled single-scenario setting.

Algorithm	Path length (m)	Accumulated path risk cost	UARV	UARV reduction rate
A*	6930.0	737.54	0.10643	0
RRT*	7329.7	802.50	0.10949	−2.88%
PSO	7138.9	941.56	0.13189	−23.92%
RDA*-IDB	8275.1	880.12	0.10636	0.066%

Considering that the main objective of this study is to reduce UAV flight risk in urban low-altitude environments rather than to pursue geometric smoothness alone, path smoothness is not adopted as a primary evaluation metric in the overall comparison. Instead, path length, accumulated path risk cost, UARV, and UARV reduction rate are selected as the main evaluation metrics. The accumulated path risk cost is defined in Eq (5), and path length, UARV, and UARV reduction rate are defined in Eqs (38)–(42). The comparison results are presented in Table 8.

The path length is used to measure the total flight distance from the start point to the target point, which is defined as

$$L = \sum_{i=1}^{N-1} l_i, \quad (38)$$

where L denotes the total path length, N represents the total number of discrete path points, and l_i denotes the length of the path segment between the i th path point and the $(i + 1)$ -th path point. The segment length l_i is calculated as

$$l_i = \|P_{i+1} - P_i\|_2, \quad (39)$$

where P_i denotes the i th path point, P_{i+1} denotes the adjacent next path point, and $\|\cdot\|_2$ represents the Euclidean norm. A smaller value of L indicates a shorter flight path and a lower geometric travel cost.

The unit path average risk value is used to measure the average risk level per unit path length during UAV flight, which is defined as

$$\text{UARV} = \frac{\sum_{i=1}^{N-1} \bar{C}_{u,i} l_i}{\sum_{i=1}^{N-1} l_i} \quad (40)$$

where UARV denotes the unit-path average risk value, $\bar{C}_{u,i}$ represents the average integrated risk value of the i th path segment, and l_i denotes the length of the i th path segment. Specifically, $\bar{C}_{u,i}$ is calculated from the normalized integrated risk values assigned to the airspace voxels containing the two endpoints of the corresponding path segment. UARV is calculated as the length-weighted mean risk encountered along the complete UAV path and therefore represents the average relative risk intensity per unit flight distance. This normalization is necessary because the compared algorithms may generate paths with different total lengths. The accumulated path risk is jointly affected by the risk level of the traversed airspace and the total travel distance; consequently, a longer path may produce a larger accumulated risk even when it mainly passes through relatively low-risk regions. By normalizing the accumulated

risk with respect to the total path length, UARV reduces the direct influence of path length differences and provides a more comparable measure of the risk avoidance tendency of different paths.

The segment length weighting also reduces the influence of the number and spacing of discrete waypoints. A simple arithmetic mean of waypoint risk values may change when the same geometric path is represented by different numbers of waypoints, whereas the length-weighted formulation more consistently reflects the risk exposure accumulated along the continuous path. Nevertheless, UARV is not intended to replace the path length metric or to independently represent the total mission risk.

The average risk value of the i th path segment is defined as

$$\bar{C}_{u,i} = \frac{C_u(P_i) + C_u(P_{i+1})}{2}, \quad (41)$$

where $C_u(P_i)$ and $C_u(P_{i+1})$ denote the normalized integrated risk values assigned to the airspace voxels containing path points P_i and P_{i+1} , respectively, and $\bar{C}_{u,i}$ represents the average integrated risk value of the two endpoints of the i th path segment. This endpoint averaging provides a trapezoidal approximation of the risk variation along the corresponding path segment and is consistent with the accumulated path risk calculation in Eq (5).

According to Eqs (5) and (40), UARV is obtained by dividing the accumulated path risk cost by the total path length. Therefore, the accumulated path risk cost and UARV are mathematically related rather than independent risk measures. The accumulated path risk cost reflects the length-weighted relative risk accumulated along the complete path and is jointly determined by the risk values of the traversed airspace and the total travel distance. By contrast, UARV represents the average relative risk intensity per unit path length and facilitates the comparison of paths with different lengths. Because C_u is a normalized relative risk index, the accumulated path risk cost should be interpreted as a relative path risk measure rather than as an absolute accident probability or monetary loss.

To further quantify the relative reduction in UARV achieved by different algorithms, A^* is selected as the reference algorithm, and the UARV reduction rate is defined as

$$\eta_{\text{UARV}} = \frac{\text{UARV}_{A^*} - \text{UARV}_i}{\text{UARV}_{A^*}} \times 100\%. \quad (42)$$

In Eq (42), η_{UARV} denotes the UARV reduction rate, UARV_{A^*} represents the unit path average risk value obtained by A^* , and UARV_i represents the unit path average risk value obtained by the i th compared algorithm. When $\eta_{\text{UARV}} > 0$, the corresponding algorithm obtains a lower UARV than A^* . When $\eta_{\text{UARV}} < 0$, its UARV is higher than that of A^* .

As shown in Table 8, A^* obtains the shortest path length of 6930.0 m, indicating its advantage in geometric shortest-path search. The path lengths generated by RRT* and PSO are 7329.7 m and 7138.9 m, respectively. The proposed RDA*-IDB framework generates the longest path among the evaluated methods, with a path length of 8275.1 m.

For accumulated path-risk cost, A^* obtains the lowest value of 737.54, followed by RRT* with 802.50, RDA*-IDB with 880.12, and PSO with 941.56. As defined in Eqs (5) and (40), the accumulated path risk cost is jointly determined by UARV and total path length. Because RDA*-IDB and A^* have nearly identical UARV values, the higher accumulated path risk cost of RDA*-IDB is primarily associated with its longer generated path rather than with a higher average risk intensity in the traversed airspace.

From the perspective of UARV, RDA*-IDB achieves the lowest value of 0.10636, compared with 0.10643 for A*, 0.10949 for RRT*, and 0.13189 for PSO. The corresponding UARV reduction relative to A* is 0.066%, indicating a modest improvement in average relative risk intensity per unit path length in this single scenario. Therefore, the results reveal a scenario-specific trade-off between path length and average risk intensity: RDA*-IDB obtains the lowest UARV among the evaluated methods, whereas its longer path results in a higher accumulated path risk cost than those of A* and RRT*.

The fixed single-scenario comparison in Table 8 should also be considered together with the 30 randomly generated start-goal scenarios reported in Table 7. In those repeated simulations, the first-level RDA* global planner reduces UARV by an average of 32.51% relative to A* while increasing the mean path length by only 0.56%. Thus, the single-scenario comparison illustrates the risk-distance trade-off obtained by the complete hierarchical path under a fixed planning task, whereas the repeated random scenario evaluation provides complementary statistical evidence for the risk-aware global guidance capability of RDA* across different planning tasks within the study area.

From a broader networked systems perspective, risk-aware UAV path planning can be interpreted as a robustness-oriented route design problem, because spatially heterogeneous risk exposure and infrastructure vulnerability affect the consequences of localized failures and the availability of safer routes. This interpretation is consistent with the robustness and resilience framework discussed by Artime et al. [36]. However, the present study focuses on static preflight risk avoidance; dynamic disruption response and path recovery remain topics for future work.

6. Conclusions

This paper presents a hierarchical risk-aware path-planning framework for UAV operations in complex urban airspace. First, a voxel-based 3D urban risk map is constructed by integrating fatality risk and property damage risk, which provides spatial risk information for subsequent path planning. The risk distribution analysis indicates that the flight bands from Layer 15 (15 m) to Layer 55 (55 m) are relatively safer for UAV operations in the study area. Based on the constructed risk map, a hierarchical path-planning framework is developed. In the first level, the RDA* algorithm is used for global path guidance by balancing distance cost and risk cost. In the second level, the IDB algorithm is employed for local path optimization to achieve fine-scale trajectory refinement. Each pair of adjacent global waypoints defines a corresponding high-resolution local planning domain, within which the trajectory is refined under path length, turning angle, and obstacle feasibility considerations. The optimized local path segments are then sequentially connected to form the complete UAV path.

A real-world urban spatial dataset from Jinan, China, covering an area of 5 km×5 km, together with the flight parameters of a DJI M210 RTK UAV, is used for simulation validation. Under the considered simulation conditions, the proposed RDA*-IDB framework achieves the lowest unit path average risk value (UARV) among the selected baseline algorithms, including A*, RRT*, and PSO. However, its UARV of 0.10636 is only 0.066% lower than the value of 0.10643 obtained by A*. Meanwhile, the longer path generated by RDA*-IDB results in an accumulated path risk cost of 880.12, which is higher than the values obtained by A* and RRT* but lower than that obtained by PSO. Because UARV is calculated by normalizing the accumulated path risk cost with respect to the total path length, these results indicate that RDA*-IDB traverses airspace with a slightly lower average relative risk intensity

per unit distance, although the increase in path length leads to a higher accumulated path risk cost than those of A^* and RRT^* . Therefore, the single-scenario comparison reflects a trade-off between path length and average risk intensity under the current simulation setting.

Future work will extend the proposed framework to dynamic urban risk scenarios, multi-UAV cooperative path planning, and real-time airspace information updates. It will also investigate the sensitivity and cross-environment transferability of the internal fatality risk and property damage risk weights and develop recalibration strategies for urban areas with different population, traffic, and building distributions.

Author contributions

Xiao Ming Wu: Conceptualization, methodology, supervision, and writing–review and editing; Hao Li: Methodology, software, data curation, visualization, and writing–original draft; Shan Xing Wang: Data analysis, validation, and investigation; Qi Zhang: Formal analysis, literature review, and manuscript revision; Zhi Han Liang: Visualization, experimental assistance, and data organization; Su Yu Zhou: Resources, data support, validation, and manuscript review. All authors have read and approved the final version of the manuscript.

Use of Generative-AI tools declaration

The authors declare that no generative AI tools were used in the preparation of this manuscript.

Ethics

This study did not involve human participants, human data, or animal experiments. Therefore, ethical approval was not required. The geospatial and environmental data used in this study were processed only for scientific simulation and UAV path planning analysis, and no personally identifiable information was involved.

Data accessibility

The data, code, and other supporting materials related to this study are available from the corresponding author upon reasonable request.

Funding

This research was funded by the Jinan “Solving Problems by Tendering for Champions” Program (Grant Number 202527022).

Conflict of interest

The authors declare that they have no competing interests.

References

1. X. Wang, X. Gao, L. Wang, X. Su, J. Jin, X. Liu, et al., Resilient multi-objective mission planning for UAV formation: A unified framework integrating task pre- and re-assignment, *Def. Technol.*, **45** (2025), 203–226. <http://dx.doi.org/10.1016/j.dt.2024.08.002>
2. H. Zhang, L. Deng, S. Lin, H. Zhang, J. Dong, D. Wan, et al., LES-YOLO: Efficient object detection algorithm used on UAV for traffic monitoring, *Meas. Sci. Technol.*, **36** (2024), 016008. <http://dx.doi.org/10.1088/1361-6501/ad86e2>
3. X. Su, L. Tao, H. Liu, L. Wang, M. Suo, Real-time hierarchical risk assessment for UAVs based on recurrent fusion autoencoder and dynamic FCE: A hybrid framework, *Appl. Soft Comput.*, **106** (2021), 107286. <http://dx.doi.org/10.1016/j.asoc.2021.107286>
4. Joint Authorities for Rulemaking on Unmanned Systems (JARUS), *JARUS guidelines on specific operations risk assessment (SORA)*, 2019.
5. R. A. Paielli, H. Erzberger, Conflict probability estimation for free flight, *J. Guid. Control Dyn.*, **20** (1997), 588–596. <http://dx.doi.org/10.2514/2.4081>
6. R. A. Paielli, H. Erzberger, Conflict probability estimation generalized to non-level flight, *Air Traffic Control Q.*, **7** (1999), 195–222. <http://dx.doi.org/10.2514/atcq.7.3.195>
7. E. R. Mueller, M. J. Kochenderfer, Multi-rotor aircraft collision avoidance using partially observable Markov decision processes, In: *AIAA modeling and simulation technologies conference*, 2016, AIAA 2016–3673. <http://dx.doi.org/10.2514/6.2016-3673>
8. M. Mitici, H. A. P. Blom, Mathematical models for air traffic conflict and collision probability estimation, *IEEE Trans. Intell. Transp. Syst.*, **20** (2019), 1052–1068. <http://dx.doi.org/10.1109/TITS.2018.2839344>
9. M. M. Bivolaru, A comparative analysis of path planning algorithms for unmanned aerial vehicles, *Proc. Rom. Acad. Ser. A*, **25** (2024), 45–53. <http://dx.doi.org/10.59277/PRA-SER.A.25.1.07>
10. J. Fan, X. Chen, X. Liang, UAV trajectory planning based on bi-directional APF-RRT* algorithm with goal-biased, *Expert Syst. Appl.*, **213** (2023), 119137. <http://dx.doi.org/10.1016/j.eswa.2022.119137>
11. M. D. Phung, Q. P. Ha, Safety-enhanced UAV path planning with spherical vector-based particle swarm optimization, *Appl. Soft Comput.*, **107** (2021), 107376. <http://dx.doi.org/10.1016/j.asoc.2021.107376>
12. V. Roberge, M. Tarbouchi, G. Labonté, Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning, *IEEE Trans. Ind. Inform.*, **9** (2013), 132–141. <http://dx.doi.org/10.1109/TII.2012.2198665>
13. X. Luo, Y. Yuan, Y. Fu, H. Huang, J. Wei, A multi-strategy improved Coati optimization algorithm for solving global optimization problems, *Cluster Comput.*, **28** (2025), 264. <http://dx.doi.org/10.1007/s10586-024-04713-y>
14. C. Yang, H. Yang, D. Zhu, Y. Hu, Y. Zhang, H. Ma, et al., A multi-mechanism balanced advanced learning sparrow search algorithm for UAV path planning, *Cluster Comput.*, **27** (2024), 6623–6666. <http://dx.doi.org/10.1007/s10586-024-04290-0>

15. S. Gao, J. Wu, J. Ai, Multi-UAV reconnaissance task allocation for heterogeneous targets using grouping ant colony optimization algorithm, *Soft Comput.*, **25** (2021), 7155–7167. <http://dx.doi.org/10.1007/s00500-021-05675-8>
16. R. Liu, H. Kang, R. Bao, S. Gong, Y. Yao, Y. Wu, An improved aphid optimization algorithm with soft thresholding, *AIMS Mathematics*, **11** (2026), 3534–3559. <http://dx.doi.org/10.3934/math.2026144>
17. L. Liu, X. Dai, Data-driven iterative learning consensus control for linear parabolic distributed parameter multi-agent systems, *AIMS Mathematics*, **11** (2026), 12687–12704. <http://dx.doi.org/10.3934/math.2026522>
18. X. Yang, P. Wei, Autonomous free flight operations in urban air mobility with computational guidance and collision avoidance, *IEEE Trans. Intell. Transp. Syst.*, **22** (2021), 5962–5975. <http://dx.doi.org/10.1109/TITS.2020.3048360>
19. S. Zeng, X. J. Xiang, Y. P. Dou, J. C. Du, G. He, UAV data link anti-interference via SLHS-SVM-AdaBoost algorithm: Classification prediction and route planning, *J. Electron. Sci. Technol.*, **22** (2024), 100279. <http://dx.doi.org/10.1016/j.jnlest.2024.100279>
20. I. Sung, B. Choi, P. Nielsen, On the training of a neural network for online path planning with offline path planning algorithms, *Int. J. Inf. Manage.*, **57** (2021), 102142. <http://dx.doi.org/10.1016/j.ijinfomgt.2020.102142>
21. Y. Hu, J. Chen, C. Qin, Event-triggered prescribed performance control of the multiplayer game nonlinear system via integral reinforcement learning, *Appl. Math. Comput.*, **511** (2026), 129716. <http://dx.doi.org/10.1016/j.amc.2025.129716>
22. S. Meng, K. Chhea, S. Muy, J. R. Lee, Deep reinforcement learning and metaheuristic approaches to maximize downlink sum-rate for Internet of Things systems in non-orthogonal multiple access-based space-air-ground integrated networks, *Rom. J. Inf. Sci. Technol.*, **28** (2025), 327–340. <http://dx.doi.org/10.59277/ROMJIST.2025.4.02>
23. X. H. Chang, X. M. Liu, L. W. Hou, J. H. Qi, Quantized fuzzy feedback control for electric vehicle lateral dynamics, *IEEE Trans. Syst. Man Cybern. Syst.*, **54** (2024), 2331–2341. <http://dx.doi.org/10.1109/TSMC.2023.3343493>
24. S. Primatesta, L. S. Cuomo, G. Guglieri, A. Rizzo, An innovative algorithm to estimate risk optimum path for unmanned aerial vehicles in urban environments, *Transp. Res. Procedia*, **35** (2018), 44–53. <http://dx.doi.org/10.1016/j.trpro.2018.12.006>
25. W. Dai, B. Pang, K. H. Low, Conflict-free four-dimensional path planning for urban air mobility considering airspace occupancy, *Aerosp. Sci. Technol.*, **119** (2021), 107154. <http://dx.doi.org/10.1016/j.ast.2021.107154>
26. E. Rudnick-Cohen, S. Azarm, J. W. Herrmann, Planning unmanned aerial system (UAS) takeoff trajectories to minimize third-party risk, In: *2019 International conference on unmanned aircraft systems (ICUAS)*, 2019, 1306–1315. <http://dx.doi.org/10.1109/ICUAS.2019.8798149>
27. B. Pang, X. Hu, W. Dai, K. H. Low, UAV path optimization with an integrated cost assessment model considering third-party risks in metropolitan environments, *Reliab. Eng. Syst. Saf.*, **222** (2022), 108399. <http://dx.doi.org/10.1016/j.ress.2022.108399>

28. P. Ji, J. Ye, Y. Mu, W. Lin, Y. Tian, C. Hens, et al., Signal propagation in complex networks, *Phys. Rep.*, **1017** (2023), 1–96. <http://dx.doi.org/10.1016/j.physrep.2023.03.005>
29. I. Pohl, Heuristic search viewed as path finding in a graph, *Artif. Intell.*, **1** (1970), 193–204. [http://dx.doi.org/10.1016/0004-3702\(70\)90007-X](http://dx.doi.org/10.1016/0004-3702(70)90007-X)
30. A. Stentz, Optimal and efficient path planning for partially-known environments, In: *Proceedings of the 1994 IEEE international conference on robotics and automation*, **4** (1994), 3310–3317. <http://dx.doi.org/10.1109/ROBOT.1994.351061>
31. S. Zhou, Y. Liu, X. Zhang, H. Dong, W. Zhang, H. Wu, et al., Risk assessment and distribution estimation for UAV operations with accurate ground feature extraction based on a multi-layer method in urban areas, *Drones*, **8** (2024), 399. <http://dx.doi.org/10.3390/drones8080399>
32. A. la Cour-Harbo, Quantifying risk of ground impact fatalities for small unmanned aircraft, *J. Intell. Robot. Syst.*, **93** (2019), 367–384. <http://dx.doi.org/10.1007/s10846-018-0853-1>
33. J. Xue, B. Shen, Dung beetle optimizer: A new meta-heuristic algorithm for global optimization, *J. Supercomput.*, **79** (2023), 7305–7336. <http://dx.doi.org/10.1007/s11227-022-04959-6>
34. K. Zhang, Y. Liu, X. Wang, F. Mei, H. Kang, G. Sun, IBMRFO: Improved binary manta ray foraging optimization with chaotic Tent map and adaptive somersault factor for feature selection, *Expert Syst. Appl.*, **251** (2024), 123977. <http://dx.doi.org/10.1016/j.eswa.2024.123977>
35. G. M. Viswanathan, V. Afanasyev, S. V. Buldyrev, E. J. Murphy, P. A. Prince, H. E. Stanley, Lévy flight search patterns of wandering albatrosses, *Nature*, **381** (1996), 413–415. <http://dx.doi.org/10.1038/381413a0>
36. O. Artime, M. Grassia, M. De Domenico, J. P. Gleeson, H. A. Makse, G. Mangioni, et al., Robustness and resilience of complex networks, *Nat. Rev. Phys.*, **6** (2024), 114–131. <http://dx.doi.org/10.1038/s42254-023-00676-y>



AIMS Press

© 2026 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<https://creativecommons.org/licenses/by/4.0>)