THE PUBLISHING HOUSE
OF THE ROMANIAN ACADEMY

PROCEEDINGS OF THE ROMANIAN ACADEMY, Series A,
Volume 26, Number 3/2025, pp. 299–310
DOI: 10.59277/PRA-SER.A.26.3.10

THREE DIMENSIONAL PATH PLANNING FOR LOW ALTITUDE RESTRICTED MAP OBSTACLE AVOIDANCE OF POLICE UNMANNED AERIAL VEHICLES BASED ON MULTI-OBJECTIVE FUNCTION Q-LEARNING

Xiaolong TIAN[1], Jin PENG[1], Nan WANG[2]

[1] Urban Rail Transit Security Department, Zhengzhou Police University, Nongye Road, Zhengzhou, 450053, Henan，China
[2] College of Information Engineering, Henan University of Science and Technology, NO.263 Kaiyuan Road, Luoyang, 471023, Henan, China
E-mails: tianxiaolong@rpc.edu.cn, pengjin@rpc.edu.cn, wswn2019@163.com
Corresponding author: Jin PENG, E-mail: pengjin@rpc.edu.cn

Abstract. Police unmanned aerial vehicle (UAV) path planning in three-dimensional environments plays a crucial role in inspection and reconnaissance along mountainous regions and railway lines. However, due to altitude restrictions in mountainous areas or along railways, the minimum-cost trajectory obtained by algorithms may not always meet practical requirements. Moreover, algorithms often focus on a single objective, such as the shortest length, whereas in actual flights, considerations must also include safety and smoothness in addition to minimizing distance. Therefore, a multi-objective Q-learning based three-dimensional path planning method for police UAVs to avoid obstacles in low-altitude constrained maps is proposed in this paper. Firstly, path length, smoothness, and safety are comprehensively considered as objective functions. Secondly, to improve the balance between exploration and exploitation, thereby enhancing path planning performance, an exploration strategy based on exponential decay is designed, which is beneficial for improving convergence speed and shortening path length. To enhance training efficiency, an automatic switching strategy between exploration and exploitation based on Q-matrix sparsity is devised. Finally, an environment is constructed and simulations are conducted on the Matlab platform to verify the effectiveness of the proposed algorithm in three-dimensional path planning for police UAVs.

Key words: police unmanned aerial vehicle, path planning, Q-learning, multi-objective function.

1. INTRODUCTION

In recent years, unmanned aerial vehicle (UAV) technology has made tremendous progress, and its application fields have exhibited an increasingly diversified trend. From efficient logistics delivery in civil sectors to precise military reconnaissance missions, from intelligent agricultural monitoring to rapid deployment in disaster emergency responses, UAV technology is profoundly transforming our work patterns and lifestyles at an unprecedented pace [1–2]. With the continuous rapid development of UAV technology, the performance requirements for UAV autonomous navigation systems have become increasingly stringent. Especially in the specific application scenario of police UAVs conducting inspection and reconnaissance tasks along mountainous railway lines, UAVs need to fly at low altitudes through complex and changing mountainous terrains. These areas are not only dense with obstacles but also impose extremely strict limitations on flight altitudes, posing unprecedented technical challenges to UAV path planning [3–4]. When performing low altitude reconnaissance missions in mountainous areas, on the one hand, unmanned aerial vehicles need to fly as close to the terrain as possible to obtain more detailed on-site information. At the same time, flying beyond altitude may have the disadvantages of being obstructed by forests, making it difficult to detect on-site situations, and consuming too much energy. Therefore, unmanned aerial vehicles need to fly autonomously along the terrain within a limited altitude range in a three-dimensional environment. This necessitates continuous exploration and innovation to enhance the autonomous navigation and path planning capabilities of UAVs in complex environments.

Path planning constitutes a fundamental problem in the guidance, navigation, and control of unmanned vehicles. The objective of path planning is to devise a route that minimizes cost while ensuring collision-free travel. Currently, prevalent path planning methods include the Dijkstra algorithm [5–6], A* algorithm [7–8], artificial potential field method [9–10], rapidly exploring random tree approach(RRT) [11–12], and artificial bee colony(ABC) algorithm [13–15].Among the various popular path planning techniques, the Dijkstra algorithm can generate an ideal path; however, its application is restricted to static environments with known global information, and its planning efficiency is relatively low. The A* algorithm, by incorporating a heuristic function, tends to expand nodes in the direction of the goal, significantly enhancing search efficiency. Nevertheless, this algorithm struggles in uncertain environments. To address the issues of excessive nodes and large turning angles in the traditional A* algorithm, Karadeniz et al. [7] proposed a geometric A* algorithm, effectively reducing the number of nodes. The artificial potential field method conceptualizes the robot's motion in an environment as movement within an artificial potential field, where the goal exerts an "attraction" on the robot, while obstacles generate a "repulsion". The robot's motion is controlled by calculating the resultant force of these two influences, enabling the planning of relatively smooth paths. However, when the attractive and repulsive forces are comparable, the robot may easily fall into local optima.

The RRT approach initiates from a root node and incrementally constructs a randomly expanding tree by adding leaf nodes through random sampling. A feasible path is deemed found when a leaf node in the tree contains the goal node. However, in environments with narrow regions, the growth of the RRT becomes challenging, and exploration efficiency diminishes. To enhance adaptability in cluttered environments, Li et al. [11] proposed an adaptive random tree algorithm that effectively tackles the narrow passage problem. The ABC algorithm is an optimization technique that mimics bee behavior, where nectar sources are randomly generated during the initialization phase and have opportunities for effective updates during the employed bee and onlooker bee phases. Consequently, this algorithm exhibits good exploration capabilities and can generate a diverse set of feasible paths. Nevertheless, it is important to note that the path optimization process inherently involves a degree of randomness and is not fully controllable [15]. To address this issue, Xu et al. introduced a cooperative framework into the algorithm, designing a globally optimized approach that has yielded promising results in robot path planning [16]. Hentout et al. designed and optimized a control layer planning structure, incorporating deterministic structural algorithms in the optimization layer to quickly generate the shortest path in the form of node sequences; The control layer uses a fuzzy logic controller to guide the robot around dynamic obstacles [17].

In the practical flight requirements of unmanned aerial vehicles, the complex three-dimensional environmental spatial structure, the abundance of obstacles, and the presence of numerous uncertainties impose higher demands on both the efficiency and quality of path planning [17–18]. While various path planning methods possess their unique advantages, real-world scenarios are characterized by changing and uncertain environments, and these methods lack the capability to learn from unknown environments. Furthermore, UAVs often cannot obtain precise environmental information, leading to the inadequacy of traditional methods in planning an effective flight path. Reinforcement learning, a branch of machine learning, possesses adaptive learning capabilities. It optimizes its action policies through the interaction between an agent and the environment in the absence of prior knowledge, thus attracting considerable attention from researchers [19]. Compared with traditional methods, numerous improved or hybrid algorithms proposed by researchers currently exhibit varying degrees of improvement in path smoothness, ability to escape local optima, planning speed, path length, and adaptability to complex environments [20].

Q-learning, a model-free learning algorithm in reinforcement learning, has garnered significant attention from scholars in the field of UAV path planning due to its alignment with the application of UAVs in certain real-world scenarios, where an environmental model is not required. Sonny *et al.* [21] proposed a Q-learning algorithm that incorporates a shortest-distance-first strategy to address path planning problems for UAVs under both static and dynamic obstacle avoidance conditions, resulting in a slight reduction in the distance required for UAVs to reach their targets. Zhou *et al.* [22] introduced a UAV path planning algorithm based on Q-learning, which features a novel Q-table initialization method, combines new action selection strategies and reward functions, and employs the root mean square propagation method for learning rate adjustment to accelerate learning and enhance path planning efficiency. Mao *et al.* [23] improved the learning speed to some extent by reducing the search space. Q-learning, as a form of reinforcement learning, has recently gained popularity in path planning for autonomous mobile robots owing to its self-learning capabilities without the need for prior environmental models. Despite these advantages, to address the slow convergence rate of Q-learning to the optimal solution, Low *et al*. [24] utilized the flower pollination algorithm (FPA) to improve the initialization of Q-learning. Experimental evaluations of the proposed enhanced Q-learning in challenging environments with different obstacle layouts demonstrated that appropriate initialization of Q-values using FPA can accelerate the convergence of Q-learning.

The current research status of path planning and reinforcement learning reveals that, due to the superior performance of reinforcement learning algorithms, utilizing reinforcement learning as a solution for path planning problems is emerging as a prominent trend. Numerous scholars have attempted to combine reinforcement learning with other methods to address path planning challenges in various domains, yielding substantial achievements.

However, the underlying concept of existing Q-learning-based UAV trajectory planning methods remains aligned with traditional trajectory planning algorithms, relying on predefined cost functions to generate a trajectory with minimal cost. Despite the substantial theoretical and applied achievements of these methods, they exhibit two significant limitations due to the omission of trajectory constraints such as the UAV's maximum climb/descent rate and minimum turning radius during the planning process: ① The minimum-cost trajectory generated by the algorithm may not satisfy practical requirements, or even be unflyable for the UAV, particularly in mountainous terrain or along railways with height restrictions; ② These algorithms typically optimize for a single objective, such as minimizing trajectory length, whereas in real-world flight, multiple factors including safety, smoothness, and shortest distance must be considered. Consequently, their applicability is limited to two-dimensional trajectory planning problems. When applied to UAV trajectory planning, these methods fail to fully leverage the UAV's three-dimensional flight capabilities, resulting in inherently suboptimal trajectories.

To address the aforementioned issues, this paper proposes a multi-objective Q-learning-based three-dimensional path planning method for obstacle avoidance in low-altitude restricted maps for police drones, building upon existing Q-learning-based trajectory planning algorithms. This method employs a multi-objective function that considers distance, safety, and smoothness. Furthermore, an exploration strategy based on exponential decay and an automatic switching strategy based on the sparsity of the Q-matrix are designed to achieve a balance between exploration and exploitation, thereby enhancing the efficiency of path planning. The main contributions of this paper are summarized as follows:

1) The multi-objective function is designed to incorporate path length, smoothness, and safety, aiming to optimize the trajectory planning process by passively considering these critical factors simultaneously. This approach ensures that the resulting paths are not only short but also smooth and safe, thereby enhancing the overall path planning performance. This holistic consideration of multiple objectives allows for the creation of more practical and feasible paths, balancing the trade-offs between efficiency, comfort, and risk mitigation in complex environments.

2) To enhance the balance between exploration and exploitation, thereby improving path planning performance, a novel exploration strategy based on exponential decay is designed. This strategy is conducive to accelerating convergence speed and reducing path length, ultimately leading to more efficient and optimal path planning outcomes.

3) To improve training efficiency, an automatic switching strategy between exploration and exploitation, grounded in the sparsity of the Q-matrix, has been designed. This approach effectively avoids premature convergence to local optima and reduces inefficient exploration.

2. PROBLEM STATEMENTS

2.1. Low altitude restriction 3D map model

As shown in Fig. 1a 100×100×50 three-dimensional grid map is utilized to represent the low-altitude environment in a detailed and structured manner. The map is composed of a series of cells, each one representing a distinct spatial location within this virtual space. The cells located below the terrain surface are clearly marked as non-flyable zones, indicating areas where the drone is not allowed to fly due to the risk of collision with the ground. Conversely, the spaces situated above the terrain surface are designated as flyable areas, providing the drone with safe passageways to navigate through. To add complexity and realism to the simulation, dynamic obstacles with a radius of 7.5 meters are introduced. These obstacles are designed to move and change positions over time, posing a significant challenge for the drone's real-time path planning capabilities. The drone must constantly adapt its flight path to avoid these moving obstacles and reach its intended destination.

We assume that the drone is simplified as a point with three-dimensional coordinates, needs to fly at low altitude, and the flight altitude is limited by terrain and obstacles; Using a three-dimensional grid map to represent the flight environment of drones, where each grid cell represents a specific spatial location; Terrain height information is crucial in restricting the drone's flight altitude, ensuring that it does not fly too low and risk colliding with the ground or other obstacles. The height constraint condition is autonomously set, and in this article, it is a curve generated by the system that is slightly higher than the terrain. At the same time, obstacle information is employed for avoidance planning, allowing the drone to navigate around both static and dynamic obstacles with precision. The starting point and the destination are strategically set within the grid, positioned on opposite sides of the obstacles. This setup requires the drone to plan and execute a path that effectively navigates around the obstacles to reach its destination.

Although the map is fully capable of accommodating both static and dynamic obstacles, this paper primarily focuses on scenarios involving static obstacles. By examining the drone's path planning and avoidance strategies in the presence of fixed obstacles, valuable insights can be gained into the development of more advanced and robust drone navigation systems. The constructed 3D environment example fully reproduces the core challenges of police drones in low altitude restricted environments through precise scale matching, constraint embedding, and multi-objective conflict design, while balancing the rigor of algorithm validation with the representativeness of real-world scenarios. Its simplified design focuses on the core contribution of this article (multi-objective path planning framework), providing a scalable benchmark testing platform for subsequent research. The current example does not introduce dynamic obstacles, mainly based on the following considerations: static obstacle avoidance is the highest priority core requirement in police drone reconnaissance tasks, and the handling of dynamic obstacles highly relies on real-time perception modules, which will increase costs. In future work, it is necessary to expand the scene and introduce dynamic obstacles. At present, after verifying the basic performance of the algorithm through static scenarios, it can be easily extended to dynamic environments.

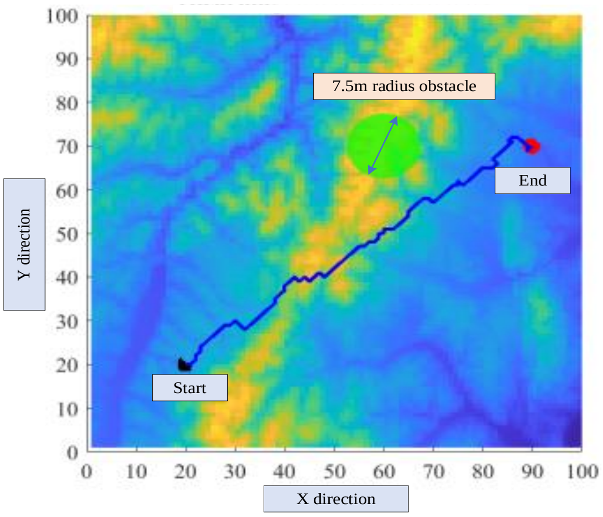


Fig. 1 – The 3D map model.

2.2. UAV model

The unmanned aerial vehicle is simplified as a point, disregarding its size and orientation. The state of the UAV is represented by its three-dimensional coordinates (x, y, z). The actions available to the UAV include movements in three directions: forward, backward, left, right, up, and down, as well as combinations of these directional movements. This simplification allows us to focus on the fundamental aspects of path planning and obstacle avoidance without the added complexity of considering the UAV's physical dimensions and attitude.

By treating the UAV as a point, we can more easily model its movement through the three-dimensional grid map and develop algorithms for navigating around obstacles to reach the intended destination.

2.3. Problem statement

Given a three-dimensional low-altitude restricted map, along with the starting point and target point of an unmanned aerial vehicle, the objective of this paper is to design a multi-objective Q-learning path planning algorithm that takes into account distance, safety, and smoothness. The algorithm aims to enable the UAV to reach the target point from the starting point safely and efficiently while avoiding obstacles, minimizing the flight path length as much as possible, and ensuring that the training time and convergence speed are not excessively slow.

3. MULTI-OBJECTIVE Q-LEARNING FOR 3D PATH PLANNING OF POLICE DRONES

3.1. Q-learning

Q-learning is a reinforcement learning algorithm rooted in value iteration. Its central idea revolves around continuously interacting with the environment to learn a Q-value function, denoted as $Q(s,a)$, which represents the cumulative reward obtainable by executing action a in state s in Fig. 2. By iteratively updating this Q-value function, the algorithm ultimately converges to an optimal policy that maximizes the cumulative reward for the unmanned aerial vehicle.

$$Q_\pi(s,a) \leftarrow (1-\alpha)Q_\pi(s,a)+\alpha\left[R_s^a+\gamma\max Q_\pi(s',a')\right], \tag{1}$$

where α represents the learning rate, γ denotes the discount factor, and R_s^a is the reward value obtained when action a is taken in state s.

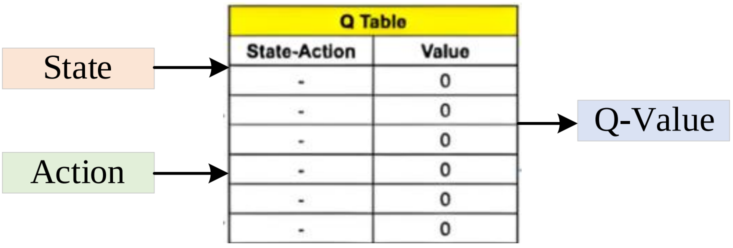


Fig. 2 – Q-learning process diagram.

3.2. Improved Q-learning algorithm based on multi-objective function

(1) An exponential decay function is employed to adaptively adjust the exploration rate throughout the training process:

$$\varepsilon = \varepsilon_{initial} e^{(-times/\lambda)} \tag{2}$$

where ε represents the exploration rate, $\varepsilon_{initial}$ is the initial exploration rate, $times$ denotes the current time step or training episode, and λ is the decay factor that controls the rate of decrease (a larger value results in slower decay, while a smaller value leads to faster decay). The exponential decay method is utilized to achieve the following objectives: High exploration initially: At the beginning of training, ε is relatively large, allowing the agent to explore the environment by trying more random actions. High exploitation later: As training progresses, ε gradually decreases, enabling the agent to rely more on its acquired knowledge (Q-values) to select optimal actions.

(2) Automatically switching between exploration and exploitation strategies based on the sparsity of the Q-matrix to enhance training efficiency.

When the Q-matrix is sparse, exploratory learning is employed. In such cases, a function named noyip (or an appropriately named function indicating exploration) is designed to explore the environment by randomly selecting actions to populate the Q-matrix, thereby preventing premature convergence to a local optimum. Specifically, when the agent's knowledge of the environment is insufficient, a purely random exploration strategy is used to update the Q-value matrix. This forces the agent to try all possible movement directions within a 3×3×3 neighborhood, laying the foundation for subsequent greedy strategies.

When the Q-matrix is dense, exploitative path planning is implemented. The $\varepsilon - greedy$ strategy is employed to balance exploration and exploitation, returning the coordinates of path points and the path length. Specifically, when the agent has accumulated a certain amount of knowledge about the environment, it generates paths and optimizes Q-values using the $\varepsilon - greedy$ strategy. This ensures that the next state selected greedily is valid (i.e., not an obstacle and within bounds), accelerating convergence. A dense Q-matrix contains a wealth of environmental knowledge, allowing the greedy strategy to quickly generate near-optimal paths and reduce ineffective exploration. By automatically switching between exploration and exploitation strategies based on the sparsity of the Q-matrix, training efficiency is enhanced. The Q matrix is a two-dimensional matrix with dimensions consisting of the total number of states multiplied by the total number of actions.

(3) A multi-objective function is adopted as the training objective to enhance both smoothness and safety. All three indicators are quantified through scalars:

a. Shortest path target. This function measures the total length of the path and guides the algorithm to find a shorter path. In actual flight, shorter paths can reduce flight time and energy consumption, and improve mission efficiency. The formula for path length is as follows:

$$F_l = \sum_{i=1}^{n} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2} \tag{3}$$

where n represents the number of waypoints, and x_i, y_i, z_i denotes the position of the i-th waypoint.

b. Path smoothness target. The planned path should minimize angular deviation and sudden height changes, and maintain a smooth path. Due to the steep terrain and large drop in altitude in mountainous areas, drones need to meet their maximum climb angle and climb rate requirements when flying in mountainous areas. This function is mainly used to reduce the angle deviation and height mutation of the path, ensuring a smooth flight path. Smooth paths help improve flight stability, reduce mechanical stress and energy consumption of drones, and enhance flight safety and comfort. The formula for path smoothness is as follows:

$$\begin{cases} \varphi_i = \arctan\left(\dfrac{\| l_i l_{i+1} \|}{l_i l_{i+1}} \right) \\ \phi_i = \arctan\left(\dfrac{z_{i+1} - z_i}{\sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}} \right) \\ F_e = \sum_{i=1}^{n-2} \varphi_i + \sum_{i=1}^{n-1} (\varphi_i - \varphi_{i-1}) \end{cases} \tag{4}$$

where l_i represents the distance between two trajectory points, φ_i denotes the deflection angle, ϕ_i denotes the pitch angle.

c. Safety objectives. This function is used to ensure the safety of drone flight, prompting the algorithm to plan a path that avoids dangerous areas and ensures the safe flight of the drone. If the action brings the drone closer to the target point, a positive reward will be given; Otherwise, a negative reward will be given. Obstacle avoidance reward: If the action causes the drone to collide with an obstacle, a large negative reward will be given; Otherwise, a positive reward will be given. Height limit reward: If the drone's flight altitude exceeds the limit, a larger negative reward will be given. In path planning, it is also necessary to ensure the flight safety of the drone. Therefore, there is an obstacle K with a center coordinate of O and a radius of R_k in the airspace. The perpendicular distance d_k between the drone's flight node and the obstacle should be greater than the safety distance threshold S:

$$F_s = \begin{cases} 0, & d_k \geq S \\ S - d_k, & R_k < d_k < S \\ \infty, & d_k < R_k \end{cases} . \tag{5}$$

The UAV path planning problem in this study is formulated as a multi-objective optimization problem, aiming to minimize the comprehensive cost function F (Eq. (6)) while satisfying spatial constraints and UAV motion capabilities. The above various cost functions are weighted and synthesized to form the objective function F of the multi-objective path planning problem:

$$F = \omega_1 F_l + \omega_2 F_e + \omega_3 F_s . \tag{6}$$

Reward Function. The multi-objective function F in Equation (6) serves as the core of the optimization problem, transforming the path planning task into a process of minimizing cumulative cost while satisfying constraints. In multi-objective path planning, using negative values of the objective function as the reward function is a common design in reinforcement learning, and its core logic is to transform the optimization problem into the process of maximizing cumulative rewards.

State space. State s is composed of the three-dimensional coordinates x_i, y_i, z_i of the drone and surrounding environmental information. The surrounding environment information can be obtained by scanning the grid cells within a certain range around the drone, for example, to determine whether there are obstacles within that range. The state space is three-dimensional, with dimensions of three-dimensional coordinates (x, y, z).

Action space. Action a includes a series of discrete movement instructions, such as small displacements in the x, y, and z directions. The selection of actions needs to consider the motion capability of the drone and the limitations of the map.The action space is one-dimensional, including three directions of movement and stillness, so there are 27 actions in 27. To reduce computational difficulty, 27 different actions are represented by 27 indices of a one-dimensional matrix.

3.3. Algorithm pseudocode

```
Algorithm: Improved Q-Learning for Path Planning
Input: discount factor $\{\gamma_i\}_{i=0,1,2,...}$
       learning rate $\{\alpha_i(a)\}_{i=0,1,2,...}$
       exploration rate $\{\varepsilon_i\}_{i=0,1,2,...}$
Output: Q-matrix , Route length , Return point , Times
Initialization: Initialize Q function,
Initial exploration rate 0.9,
Decay Rate 200,
Minimum exploration rate 0.1
while   convergence flag=false
        If  The number of non-zero elements in the Q matrix is less than 500000:
                Output information: Q matrix is sparse, no greedy qlearning is in progress, iteration times
                Call noyip function for non greedy Q-learning, update Q and routing length;
        Else    Output information: Greedy Q-learning in progress, iteration times;
                    Setup yip = 0.9;
                Call the haveyip function for greedy Q-learning to obtain path points qwaypoints and path length routeLength;
        If  The path length is not 0, the number of path points is less than 100, and the change in path length is less than 5;
        convergence flag=true
        End if
    Record the current iteration count and path length to route length
End while
return (Q-matrix , Route length , Return point , Times )
```

This pseudocode combines the basic framework of Q-learning algorithm and considers the attenuation of exploration rate and (optional) target network updates to improve learning efficiency and stability. The improved Q-learning solves the problem of traditional Q-learning being prone to local optima and slow convergence in complex 3D environments through dynamic policy switching and multi condition termination mechanisms. Its core is to adaptively adjust the exploration and utilization proportion through Q matrix sparsity, combined with multi-dimensional reward function to guide path optimization, and ultimately generate stable and concise feasible paths.

4. SIMULATION ENVIRONMENT AND ANALYSIS

4.1. Simulation environment setup

MATLAB software is utilized to conduct simulation experiments on the aforementioned algorithm. Initially, a three-dimensional map encompassing terrain and obstacles was constructed, with the lower left corner designated as the origin of a coordinate system where the horizontal direction represents the x-axis and the vertical direction the y-axis. The starting position (20, 20, 7) and the target position (90, 70, 5) were marked; the starting position was indicated by a black square, and the target position by a red circle. The interior of the green circles denoted obstacle regions that were impassable, with (60, 70, 20) serving as an example of an obstacle center. The remaining areas within the boundaries constituted the free-roaming zone for the unmanned aerial vehicle (UAV) (as illustrated in Fig. 2). The UAV's action space at any given state s comprised four movements: up, down, left, and right; however, entry into obstacle regions was prohibited. Subsequently, MATLAB code was developed to implement the Q-learning algorithm, with corresponding parameters such as the learning rate α , discount factor γ , and the number of iterations being appropriately set. Finally, through simulation experiments, the flight trajectory of the UAV was observed, and the performance of the algorithm was evaluated.

Mainly comparing the ordinary Q-learning algorithm with the improved multi-objective function Q-learning algorithm. Among them, the experimental parameters and key parameter values of all learning processes of the Q-learning algorithm are shown in Table 1.

Table 1.

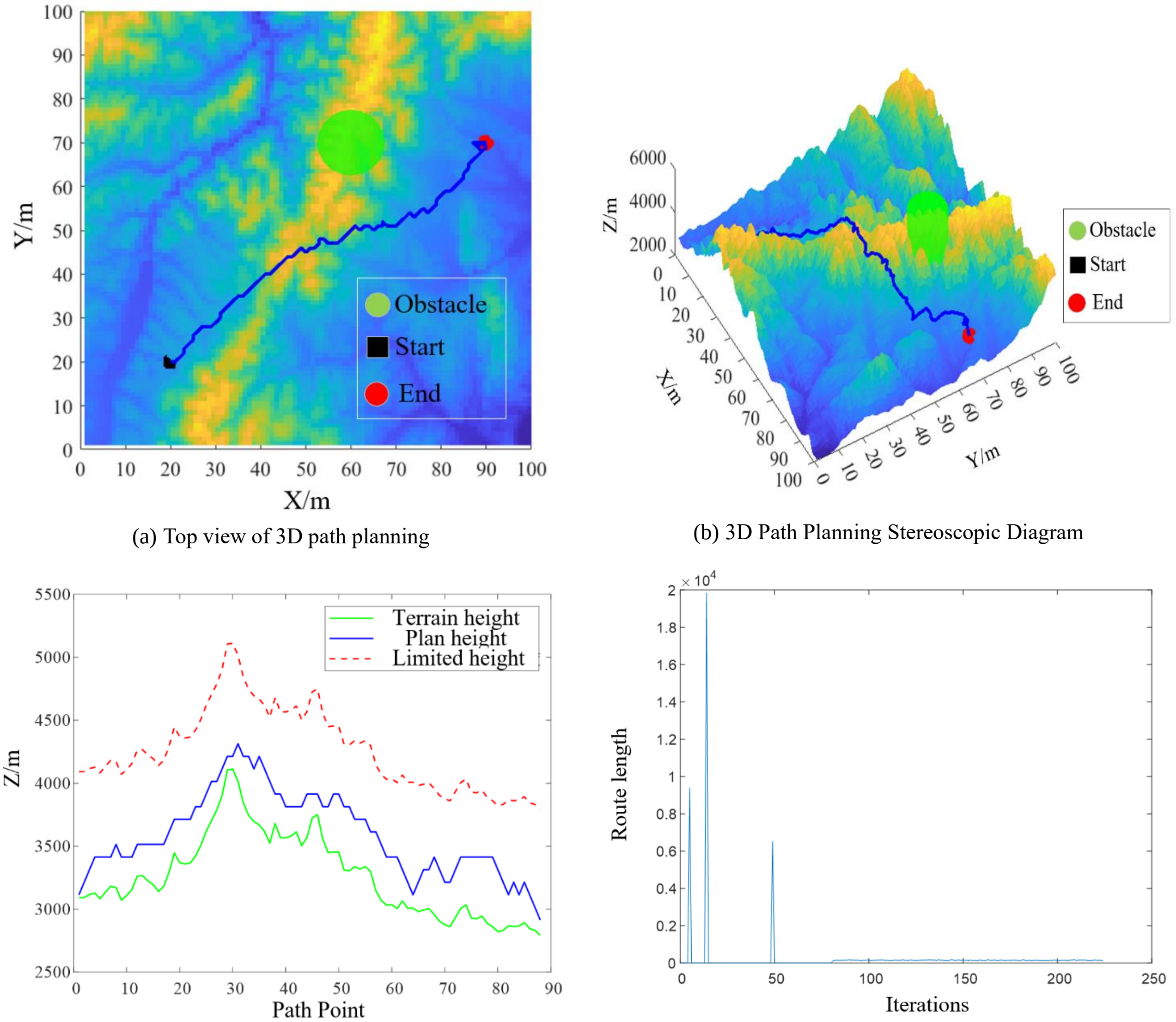
Parameter settings

Name	Value
Minimum exploration rate ε_{min}	0.1
Initial exploration rate $\varepsilon_{initial}$	0.9
Learning rate α	0.9
Discount factor γ	0.7
High penalty coefficient	5
Distance penalty coefficient	15
Smooth penalty coefficient	5
3D perspective setting	60–70
Path point setting	100

4.2. Simulation analysis

Figure 3 shows the 3D path planning effect of unmanned aerial vehicles under the traditional Q-learning algorithm. Figure 3a is a top view of the Q-learning algorithm's 3D path planning, from which it can be visually seen that the path bypasses obstacles and connects the starting and ending points. Figure 3b is a three-dimensional path planning diagram of the Q-learning algorithm, which shows the feasibility of the path in three-dimensional space and can achieve functions such as height obstacle avoidance and terrain following.

Figure 3c is a height comparison diagram of the trajectory profile, mainly used to verify whether the planned trajectory meets the requirements of low altitude flight (i.e. the blue curve is always higher than the red dashed line), ensuring that the trajectory height maintains a safe distance from the terrain and avoiding collision risks. It can be seen that this algorithm effectively avoids collisions. Figure 3d shows the trend of path length variation with the number of iterations, verifying the convergence of the Q-learning algorithm. In the process of path planning, the path length gradually optimizes with learning iterations, and the curve eventually tends to be stable, indicating that the algorithm has converged. However, this algorithm also has some shortcomings, such as slow convergence speed and lack of smoothness.



(a) Top view of 3D path planning

(b) 3D Path Planning Stereoscopic Diagram

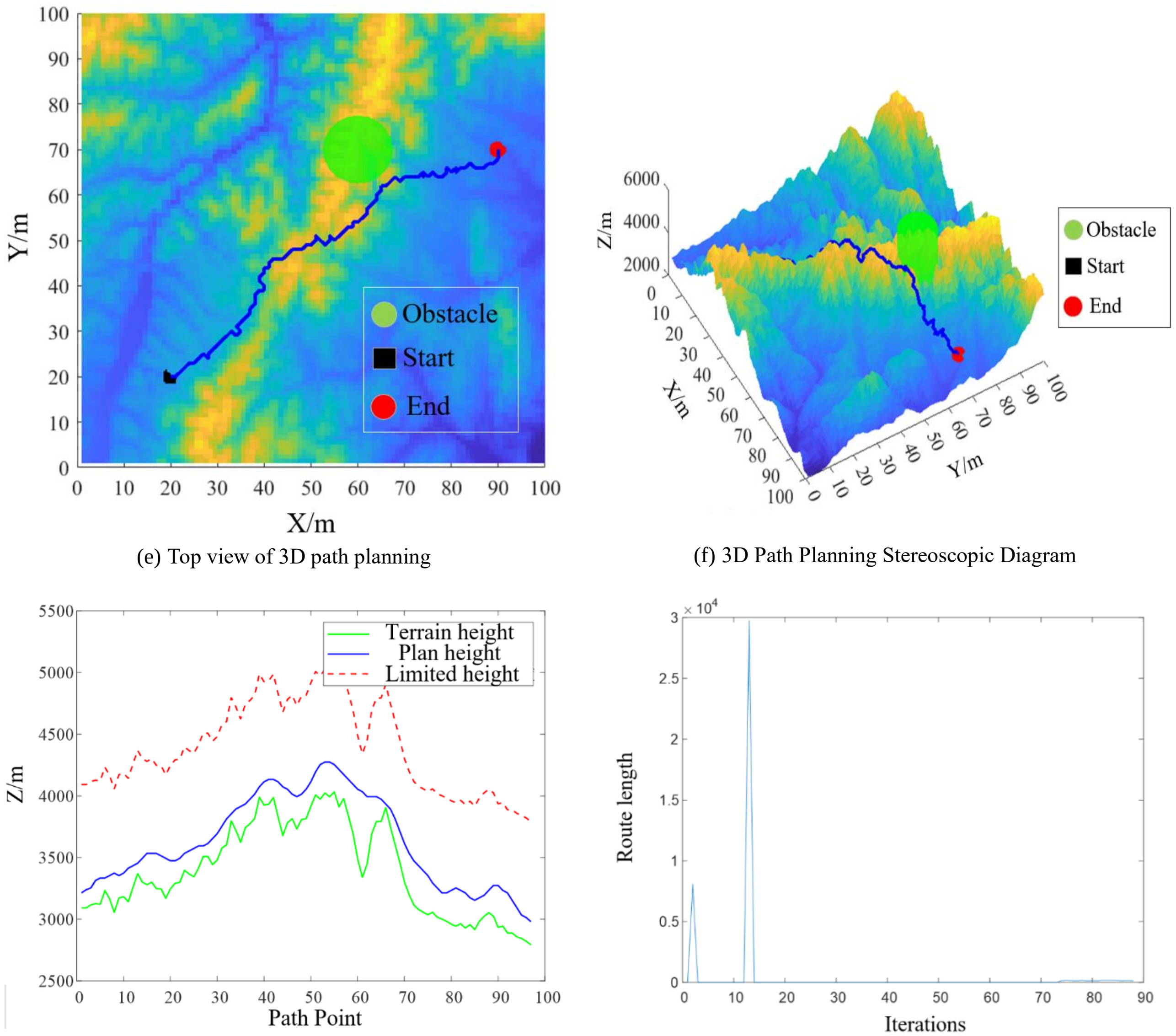
(c) Comparison chart of trajectory profile altitude

(d) The variation of path length with the iterations

Fig. 3 – The effectiveness of traditional Q-learning in 3D path planning.

Compared with the traditional Q-learning based UAV path planning algorithm, the advantages of the improved algorithm can be seen from Fig. 4. The top view of the 3D path planning in Fig. 4e shows that the path bypasses obstacles and connects the starting and ending points, and does not bypass obstacles very far. The three-dimensional path planning diagram of Q-learning algorithm in Fig. 4f shows that the drone trajectory under this algorithm is more in line with the terrain, while avoiding altitude restrictions, and the flight trajectory is smoother. The height comparison of the trajectory profile in Fig. 4g also shows that the algorithm not only effectively avoids collisions, but also benefits from the reasonable setting of the multi-objective function. Figure 4h shows the trend of path length variation with the number of iterations.

It can be seen that during the path planning process, the path length gradually optimizes with fewer iterations and shorter convergence time. Therefore, the proposed algorithm has significant advantages over traditional Q-learning algorithms in 3D path planning for police drones. Table 2 also shows that under the proposed algorithm, the number of iterations decreased by 60%, the running time decreased by 20.9%, and the total path was shortened by 9.9%.



(e) Top view of 3D path planning

(f) 3D Path Planning Stereoscopic Diagram

(g) Comparison chart of trajectory profile altitude

(h) The variation of path length with the iterations

Fig. 4 – The effectiveness of the improved Q-learning algorithm proposed in 3D path planning.

Table 2

Comparison of the effectiveness between the proposed algorithm and traditional Q-learning algorithm

Method	Iterations		Run time		Route length	
	Value	Δ (%)	Value	Δ (%)	Value	Δ (%)
Q-Learning	225	--	24.16	--	14232.91	--
Proposed method	89	↓60.4%	19.11	↓ 20.9%	12818.69	↓ 9.9%

5. CONCLUSIONS

This paper proposes a three-dimensional path planning method for police drones based on multi-objective Q-learning. Compared with general learning methods, Compared with general learning methods, this article has made improvements in the following three aspects. Firstly, in order to meet the actual flight situation, the path length, smoothness, and safety are comprehensively considered as the objective functions. Secondly, in order to improve the balance between exploration and development, an exploration strategy based on exponential decay was designed to enhance convergence speed and shorten path length, thereby improving path planning performance. Meanwhile, in order to improve training efficiency, an automatic switching strategy between exploration and development based on Q matrix sparsity was designed. Finally, an environment was built on the Matlab platform and compared with traditional Q-learning algorithms. Compared with non-learning algorithms, the system can learn the distribution patterns of dynamic obstacles, environmental uncertainties (such as sudden obstacles), and complex three-dimensional terrain features from historical data or real-time interactions, and adjust path planning strategies in real time. The results showed that the police unmanned aerial vehicle 3D path planning method based on multi-objective Q-learning shortened training time and path length while improving smoothness during flight.

Although the proposed method effectively shortens convergence time and improves smoothness in path planning, our research still has limitations. When there are dynamic obstacle targets, more comprehensive considerations may need to be made, and algorithm design may be more complex to meet higher difficulty requirements.

ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China under Grant (Grant Nos. 62301212,62371182), the Fundamental Research Funds for Central Universities (Grant Nos. 2025TJJBKY009, 2025TJJBKY023, 2025TJJBKY006, 2025TJJBKY021).

REFERENCES

[1] Laghari AA, Jumani AK, Laghari RA, Nawaz H. Unmanned aerial vehicles: A review. Cognitive Robotics. 2023; 3(1): 8–22.

[2] Meng K, Wu Q, Xu J, Chen W, Feng Z, Schober R, Swindlehurst AL. UAV-enabled integrated sensing and communication: Opportunities and challenges. IEEE Wireless Communications, 2023; 31(2): 97–104.

[3] Mu J, Zhang R, Cui Y, Gao N, Jing X. UAV meets integrated sensing and communication: Challenges and future directions. IEEE Communications Magazine. 2023; 61(5): 62–67.

[4] Saulnier A, Thompson SN. Police UAV use: Institutional realities and public perceptions. Policing: An International Journal of Police Strategies & Management. 2016; 39(4): 680–693.

[5] Zhou Y, Huang N. Airport AGV path optimization model based on ant colony algorithm to optimize Dijkstra algorithm in urban systems. Sustainable Computing: Informatics and Systems. 2022; 35(2): 123–131.

[6] Maristany de las Casas P, Kraus L, Sedeño-Noda A, Borndörfer R. Targeted multiobjective Dijkstra algorithm. Networks. 2023; 82(3): 277–298.

[7] Karadeniz AM, Hajdu C, Koczy LT. Mobile robot environment representation through fuzzy signatures-integrated quadtrees. Romanian Journal of Information Science and Technology. 2025; 28(1): 103–116.

[8] He Z, Liu C, Chu X, Negenborn RR, Wu Q. Dynamic anti-collision A-star algorithm for multi-ship encounter situations. Applied Ocean Research. 2022; 118: 102995.

[9] Fan X, Guo Y, Liu H, Wei B, Lyu W. Improved artificial potential field method applied for AUV path planning. Mathematical Problems in Engineering. 2020; 20(1): 652–663.

[10] Rao J, Xiang C, Xi J, Chen J, Lei J, Giernacki W, Liu M. Path planning for dual UAVs cooperative suspension transport based on artificial potential field-A* algorithm. Knowledge-Based Systems. 2023; 277(1): 110797–110817.

[11] Li B, Chen B. An adaptive rapidly-exploring random tree. IEEE/CAA Journal of Automatica Sinica. 2021; 9(2): 283–294.

[12] Kelner J, Burzynski W, Stecz W. Modeling UAV swarm flight trajectories using rapidly exploring random tree algorithm. Journal of King Saud University-Computer and Information Sciences. 2024; 36(1): 1019–1025.

[13] Karaboga D, Akay B. A comparative study of artificial bee colony algorithm. Applied mathematics and computation. 2019; 21(1): 108–132.

[14] Kumar NR, Nagabhooshanam E. EKF with artificial bee colony for precise positioning of UAV using global positioning system. IETE Journal of Research. 2021; 67(1): 60–73.

[15] Chen P, Li H, Ma L. Distributed massive UAV jamming optimization algorithm with artificial bee colony. IET Communications. 2023; 17(2): 197–206.

[16] Xu F, Li H, Pun CM, Hu H, Li Y, Song Y, Gao H. A new global best guided artificial bee colony algorithm with application in robot path planning. Applied Soft Computing. 2020; 88: 106037.

[17] Hentout A, Maoudj A, Kouider A. Shortest path planning and efficient fuzzy logic control of mobile robots in indoor static and dynamic environments. Romanian Journal of Information Science and Technology. 2024; 27(1): 21–36.

[18] Qu C, Gai W, Zhong M, Zhang J. A novel reinforcement learning based grey wolf optimizer algorithm for unmanned aerial vehicles (UAVs) path planning. Applied soft computing. 2020; 89: 106099.

[19] Yan C, Xiang X, Wang C. Towards real-time path planning through deep reinforcement learning for a UAV in dynamic environments. Journal of Intelligent & Robotic Systems. 2020; 9(8): 297–309.

[20] Liu Q, Shi L, Sun L, Li J, Ding M, Shu F. Path planning for UAV-mounted mobile edge computing with deep reinforcement learning. IEEE Transactions on Vehicular Technology. 2020; 69(5): 5723–5728.

[21] Sonny A, Yeduri SR, Cenkeramaddi LR. Q-learning-based unmanned aerial vehicle path planning with dynamic obstacle avoidance. Applied Soft Computing. 2023; 147: 110773.

[22] Zhou Q, Lian Y, Wu J, Zhu M, Wang H, Cao J. An optimized Q-Learning algorithm for mobile robot local path planning. Knowledge-Based Systems. 2024; 286: 111400.

[23] Maoudj A, Hentout A. Optimal path planning approach based on Q-learning algorithm for mobile robots. Applied Soft Computing. 2020; 97: 106796.

[24] Low E, Ong P, Cheah K. Solving the optimal path planning of a mobile robot using improved Q-learning. Robotics and Autonomous Systems. 2019; 11(5): 143–161.

Received March 17, 2025