

Received 8 July 2024, accepted 22 July 2024, date of publication 29 July 2024, date of current version 7 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3434621

RESEARCH ARTICLE

Improved Q-Learning Algorithm Based on Flower Pollination Algorithm and Tabulation Method for Unmanned Aerial Vehicle Path Planning

LAN BO¹, TIEZHU ZHANG¹, HONGXIN ZHANG¹, JIAN YANG^{1,2}, ZHEN ZHANG¹, CAIHONG ZHANG³, AND MINGJIE LIU¹

¹College of Mechanical and Electrical Engineering, Qingdao University, Qingdao 266071, China

²School of Mechanical Engineering, University of Science and Technology Beijing, Beijing 100083, China

³School of Automation, Qingdao University, Qingdao 266071, China

Corresponding authors: Hongxin Zhang (zhx@qdu.edu.cn) and Jian Yang (joey8533@163.com)


This work was supported by the National Natural Science Foundation of China under Grant 52075278.

ABSTRACT Planning a path is crucial for safe and efficient Unmanned aerial vehicle flights, especially in complex environments. While the Q-learning algorithm in reinforcement learning performs better in handling such environments, it suffers from slow convergence speed and limited real-time capability. To address these problems, this study proposes an enhanced initialization process using the flower pollination algorithm and employs a tabulation method to improve local obstacle avoidance ability. An improved Q-learning algorithm based on the flower pollination algorithm and tabulation method (IQ-FAT) is proposed, which can perform both global and local path planning, enhance the convergence time of Q-learning, and expedite obstacle avoidance. Evaluation results on various obstacle maps demonstrate that the modified algorithm has a significant improvement convergence speed of approximately 40% compared to the original algorithm while enabling global path planning and local obstacle avoidance. Furthermore, the algorithm demonstrates superior path-planning capabilities in complex environments and enhances the dynamic response time of UAVs by approximately 90% compared to the artificial potential field method.

INDEX TERMS Path planning, unmanned aerial vehicle, obstacle avoidance, reinforcement learning, flower pollination algorithm.

I. INTRODUCTION

Unmanned aerial vehicles (UAVs) are increasingly being utilized in various civilian [1] and defense sectors [2], with path planning theory and methodology playing a pivotal role in ensuring the safe flight of UAVs [3]. Collision avoidance is pivotal in path planning [4], as it guarantees the safe navigation of UAVs [5]. The enhancement of UAV performance and safety heavily relies on accurate and feasible path planning as well as fast and effective obstacle avoidance strategies [6]. Autonomous UAV flights require consideration of real-time performance, cruising range, computing power, and storage capacity constraints, imposing high demands on the speed

The associate editor coordinating the review of this manuscript and approving it for publication was Yang Tang .

and simplicity of path-planning algorithms [7]. Traditional path planning can be divided into global path planning for determining optimal routes when a map is available and local path planning focuses on addressing obstacles not reflected in the map through localized obstacle avoidance techniques [8]. Depending on whether obstacles are stationary or dynamic, local path planning can further be categorized into static or dynamic local obstacle avoidance scenarios [9]. In this paper, we propose specific solutions tailored to each scenario.

In numerous previous studies, UAV path-planning algorithms have been categorized into five main groups. The first category encompasses heuristic algorithms, such as the A* algorithm [10]. However, its applicability is limited in large-scale map environments and high latitudes [11]. The second category includes random sampling-based

algorithms, with RRT being a representative example [12]. The application of RRT has limitations, such as the algorithm's lack of discernment being overly pronounced and its efficacy being suboptimal [13]. The third type involves the artificial potential field method commonly used for local obstacle avoidance; nevertheless, it tends to fall into local optima [14] and performs poorly in dense obstacle scenarios. Additionally, when the UAV reaches non-target points with balanced forces applied, it may result in unreachable targets [15]. The fourth category encompasses path planning algorithms based on mathematical principles like utilizing Dubins curves [16] for UAV path planning purposes. Lastly, intelligent bionic algorithms also find application in path planning by providing more possibilities through simple computation and achieving favorable experimental results. Commonly employed intelligent bionic algorithms include ant colony algorithm(ACA) [17], particle swarm optimization(PSO) [18], flower pollination algorithm(FPA) [19], grey wolf optimizer (GWO) [20], cuckoo search algorithm(CSA) [21], firefly algorithm(FA) [22], genetic algorithm(GA) [23], differential evolution(DE) [24], sparrow search algorithm(SSA) [25], reinforcement learning [26], etc. Combining reinforcement learning with artificial neural networks [27] forms deep reinforcement learning [28], which is promising in path planning.

Based on the analysis above, traditional path planning exhibits low fitness [29] and is limited to a single purpose in complex environments. Conversely, intelligent bionic algorithms demonstrate strong robustness and ability to effectively handle environmental changes, making them well-suited for complex scenarios. Significantly, the integration of diverse, intelligent algorithms can be effectively employed to yield favorable outcomes. For instance, Yu and Luo [21] proposed using reinforcement learning to enhance the cuckoo algorithm for three-dimensional UAV path planning. Pehlivanoglu and Pehlivanoglu [30] improved the initial population of the genetic algorithm and accelerated its convergence speed for UAV path planning.

In this paper, we utilize Q-learning in reinforcement learning for path planning, where the agent is trained and optimized through iterative interactions with the environment [31]. The reason for selecting Q-learning lies in its superior ability to handle complex environments, better real-time performance, and robustness compared with the genetic algorithm (GA) and particle swarm optimization algorithm (PSO). Despite the strong practicality of reinforcement learning algorithms that are widely used in various fields such as game design [32], intelligent control [33], energy management [34], path planning [35], etc., there still exist some issues, including slow convergence speed, re-training requirements when facing local obstacle avoidance, and low timeliness. Improvements in Q-learning can be mainly reflected in three aspects: Firstly, the convergence speed can be improved by enhancing the initialization. Konar et al. [31] proposed a novel deterministic Q-learning

algorithm that effectively updates the Q-values by leveraging the four derived Q-learning properties. Pouyan et al. [36] used the concept of opposite action to make the agent update the Q-value for each action and the corresponding opposite action. Both of these methods enhance the convergence rate. They are secondly adjusting the greedy strategy to balance the random selection action mechanism with the optimal action mechanism to avoid falling into local optima. Wang et al. [37] integrated the Sarsa and Q-learning algorithms to influence the action selection mechanism on the side. Thirdly, it proposes a deep reinforcement learning method combined with an artificial neural network, making it more applicable for continuous problems since a traditional Q-table matrix [38] requires ample memory space and only applies to discrete states. For instance, Duguleana and Mogan [39] combined path planning and neural networks to realize robot path planning in multiple dynamic and static environments. Xia et al. [40] proposed a deep reinforcement learning (DRL) method based on proximal policy optimization (PPO) to solve the active collision avoidance problem of surface vehicles.

This paper introduces a novel and highly promising flower pollination algorithm to address the abovementioned issues. This method was initially introduced by renowned British scholar Yang [41] and has been demonstrated to be more efficient than GA and PSO. Since its inception, it has been widely applied in various fields, such as scheduling processes [42], navigation [43], and data mining [44], among others. In terms of path planning specifically, there are two ways in which the flower pollination method can be utilized: either as an algorithm to optimize the planning problem [45] or as an optimization algorithm to enhance existing path planning methods. Low et al. [19] integrated Q-learning with a flower pollination method to achieve four-directional path planning in a two-dimensional plane. However, the effectiveness diminishes when extending the movement direction to eight directions. This paper adopts the latter approach by proposing a new mechanism that optimizes Q-learning for eight-directional path planning in a two-dimensional environment suitable for UAVs.

We propose an improved Q-learning algorithm using the flower pollination algorithm and the tabulation method (IQ-FAT) to improve efficiency when solving practical problems. Our motivation for developing this approach is due to limitations inherent within original Q-learning about convergence time and local obstacle avoidance.

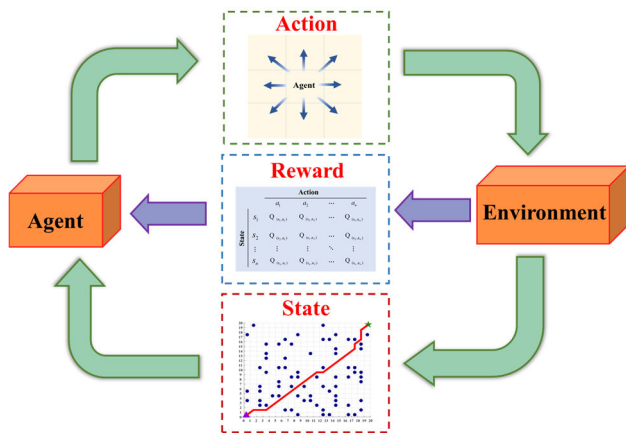
- 1) The flower pollination algorithm is employed to enhance the performance of Q-learning, resulting in a significant reduction in convergence time.
- 2) We utilize the tabulation method to aid Q-learning in local obstacle avoidance by fully using its Q-value storage matrix, saving computational resources during actual runtime.
- 3) The proposed model can realize global path planning and local obstacle avoidance.

This paper is structured as follows: Section II introduces the principle of employing a single algorithm. In Section III, we propose the specific operational mechanism of the IQ-FAT algorithm. Section IV validates the effectiveness of IQ-FAT in diverse environments and conducts a comparative analysis with similar products. Finally, in Section V, we conclude the paper and explore future research directions.

II. PRINCIPLES OF Q-LEARNING AND FLOWER POLLINATION ALGORITHM

A. Q-LEARNING ALGORITHM

Q-learning is a type of reinforcement learning that enables an ideal agent to be trained without prior knowledge. As illustrated in Fig. 1(a), the agent acts as an explorer and continuously interacts with its environment, receiving rewards or punishments for actions taken in specific states. If the agent collides with an obstacle, it gets a negative value punishment for the current state's act. Conversely, if the agent takes action towards the goal point without encountering obstacles, it receives a positive reward. These rewards are calculated and transformed into state-action values recorded in a matrix known as the Q-table.



(a) Q-learning algorithm framework.

| | | Action | | | |
|-------|----------|-----------------|-----------------|----------|-----------------|
| | | a_1 | a_2 | \dots | a_n |
| State | s_1 | $Q_{(s_1,a_1)}$ | $Q_{(s_1,a_2)}$ | \dots | $Q_{(s_1,a_n)}$ |
| | s_2 | $Q_{(s_2,a_1)}$ | $Q_{(s_2,a_2)}$ | \dots | $Q_{(s_2,a_n)}$ |
| | \vdots | \vdots | \vdots | \ddots | \vdots |
| | s_n | $Q_{(s_n,a_1)}$ | $Q_{(s_n,a_2)}$ | \dots | $Q_{(s_n,a_n)}$ |

(b) Q-table structure diagram

FIGURE 1. Schematic of the Q-learning algorithm.

The Q-table in traditional Q-learning is initially populated with obstacle information, wherein the Q-values corresponding to actions leading to obstacle states are assigned a significantly negative value. Assume that at time t , the agent is in the state S_t . The agent will select an action a_t based on the action selection policy and execute this action to promptly acquire a reward r and the subsequent state S_{t+1} from the environment. Simultaneously, the agent can adapt its action selection strategy based on the r value. The Q-table values tend to converge through multiple training iterations, enabling the Agent to rely solely on these values for selecting a sequence of states with maximum reward value, thereby accomplishing optimal path selection. The formula for updating Q-value is:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \quad (1)$$

where $Q(s, a)$ signifies the anticipated cumulative reward stemming from the execution of a specific action a within a given state s . S_t is the current state, a_t is the action chosen in the state S_t , r is the reward obtained for executing the action a_t , S_{t+1} is the next state, γ is the discount factor ($0 \leq \gamma < 1$), and α is the coefficient of learning ($0 < \alpha < 1$). The process is given as Algorithm 1.

Algorithm 1 Q-Learning Algorithm

Input: start point, goal point, and environment information
Output: the optimal path from the start to the end

- 1 initialize s
- 2 **while** $n < \text{episode}$
- 3 initialize the start point
- 4 **while** goal point not reached
- 5 choose an action a from s based on the strategy
- 6 take action a and observe r, s_{t+1}
- 7 update Q-table using (1)
- 8 $s = s_{t+1}$
- 9 **end**
- 10 **end**

B. FLOWER POLLINATION ALGORITHM

Did you know that flowering plants make up about 80% [41] of all plant species on Earth? This is because they have a competitive edge in natural environments. Flowering plants have evolved to reproduce through pollination, which can occur via biotic or abiotic means. Biotic pollinators such as bees and butterflies facilitate long-distance pollen transfer. In contrast, abiotic pollination occurs via wind or rain and typically results in localized pollen dispersal. Furthermore, the constancy of flowers is exemplified by the exclusive pollination behavior of certain insects, thereby minimizing cross-pollination and maximizing intraspecific breeding opportunities within individual flowers. The following four principles emerge from the process of flower pollination simplification:

- 1) Biological pollination and cross-pollination are considered global pollination, with the pollen's travel distance following a Levy flight pattern.
- 2) Abiotic pollination and self-pollination are commonly called local pollination.
- 3) The concept of flower constancy posits that the reproductive success of a flower is directly proportional to the degree of similarity between the two flowers involved.
- 4) The switching probability p is adjusted to regulate the balance between global and local pollination.

Let $X_i^G = (x_{i,1}, x_{i,2} \dots x_{i,D})$ denote the position of individual i in the search space at generation G , where i represents an integer within the range of $[1, n]$. First, we initialize a population of n flowers with random positions, define a switch probability p , and obtain the best solution g -best in the initial population by the objective function. Subsequently, during the global search phase, individuals will engage in levy flight behavior to approach the optimal individual within the population. This phenomenon can be mathematically represented as:

$$X_i^{G+1} = X_i^G + L(\lambda) * (X_i^G - G_{best}) \quad (2)$$

where X_i^{G+1} is the spatial location of individual i at generation $G+1$ within the search domain, G_{best} is the best individual of the population in generation G , and $L(\lambda)$ is the step size of simulated pollen for Levy flight.

The local search stage involves the individual performing local optimization. The fundamental concept is to utilize two randomly generated difference vectors between individuals to perturb the current individual randomly, ultimately obtaining the optimal individual within the local scope. This process can be expressed as:

$$X_i^{G+1} = X_i^G + \varepsilon(X_j^G - X_k^G) \quad (3)$$

where ε is a random number within $[0, 1]$, X_k^G is pollen j randomly collected from the population of generation G , and X_j^G is pollen k randomly collected from the population of generation G .

The probability p is utilized to balance global exploration and local exploration. Following a specific number of iterations, the fittest individual in the population tends to converge, with this convergence value representing the optimal solution for the given problem. The process is given as Algorithm 2.

III. MAIN IDEAS OF THE PROPOSED IQ-FAT

Q-learning algorithm is an effective and easy-implement algorithm applied to complex environments. The Q-learning navigates the optimal trajectory by continuously interacting with the environment. However, this exploration process can take time and effort.

This paper introduces the IQ-FAT algorithm to improve the original Q-learning approach's slow convergence time and limited adaptability. The IQ-FAT algorithm utilizes the

Algorithm 2 Flower Pollination Algorithm

Input: population size: n

Output: final G_{best}

1 objective min or max $f(x)$, $x = (x_1, x_2, \dots, x_n)$

2 initialize a population of n flowers with random positions

3 define a switch probability $p \in [0, 1]$

4 obtain the best solution g -best in the initial population

5 **while** $<$ Max Generation

6 **for** $i = 1: n$

7 **if** $\text{rand} < p$

8 obtain global pollination using (2)

9 **else**

10 obtain local pollination using (3)

11 **end**

12 evaluate new solutions

13 **if** new solutions are better than g -best

14 g -best = new solution

15 **end**

16 $G_{best} = g$ -best

17 **end**

FPA to speed up the initialization process and employs the tabulation method to facilitate obstacle avoidance for unknown obstacles. The IQ-FAT algorithm has demonstrated remarkable efficacy in achieving path planning for UAVs.

A. GLOBAL PATH PLANNING BASED ON FLOWER POLLINATION ALGORITHM

The global path planning of the IQ-FAT algorithm comprises an initialization process and an exploration process. IQ-FAT employs the flower pollination algorithm to enhance the prior information acquired during initialization. Subsequently, IQ-FAT adjusts specific parameters in the exploration process of the original algorithm. These enhancements result in improved convergence speed for IQ-FAT.

The initialization process of IQ-FAT can be divided into two steps: defining an initial population and implementing an iterative optimization procedure. Firstly, a random generation of i pollen particles is performed in a two-dimensional space with p coordinates (x_n, y_n) assigned to each pollen. Currently, the map serves as a boundary condition to ensure (x_n, y_n) remains within the scope of the map. The flower pollination method's initial population is denoted as follows:

$$Flower_i = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \dots & \dots \\ x_n & y_n \end{bmatrix} \quad (4)$$

Subsequently, a line L is drawn to connect the starting and ending points, and the distance between the h coordinates and line L is calculated and summed as a measure of flower fitness. To ensure adequate dispersion among coordinates, pollen with coinciding coordinates will receive reduced fitness scores. Then, the best individual G_{best} in the population is selected based on its fitness score, after which position

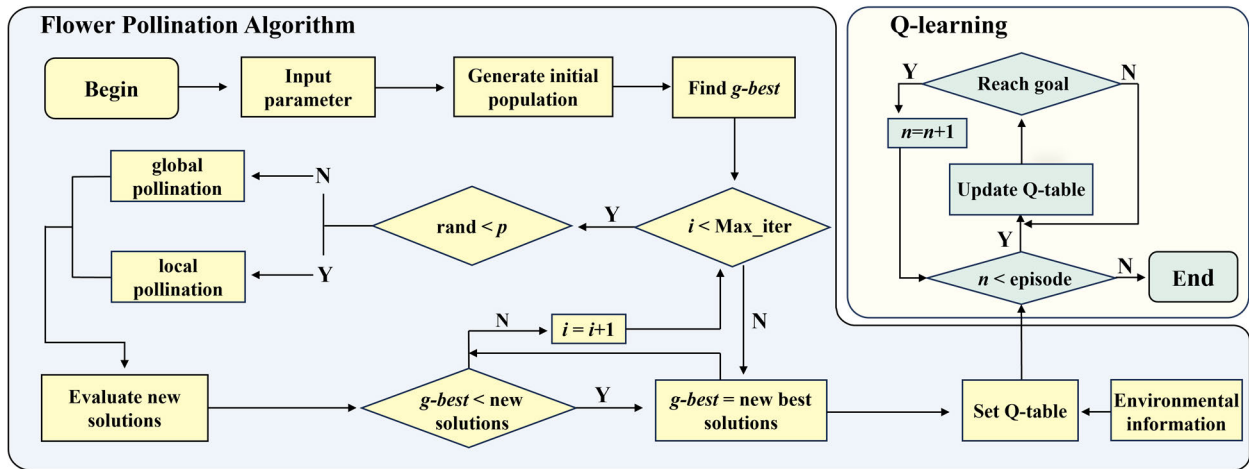


FIGURE 2. The flowchart of Q-learning initialization improved by the flower pollination algorithm.

coordinates for remaining pollen are updated through global or local pollination while ensuring they stay within integer boundaries. Afterward, the entire population’s fitness is re-evaluated to determine G_{best} again. By repeating the steps mentioned above n times, optimal pollen with p coordinates along the line and distributed according to a specific dispersion can be obtained.

Increasing Q values corresponding to these coordinates provides prior information for reinforcement learning while reducing learning process blindness. Finally, obstacle state Q values are set at larger negative values to offer correct guidance even if obstacle and pollen coordinates coincide. The flowchart of Q-learning initialization improved by the flower pollination algorithm is illustrated in Fig. 2:

The flower pollination algorithm incorporates two mechanisms: global search and local search. The former mechanism facilitates the convergence of individuals toward the optimal solution. At the same time, the latter ensures a more systematic approach to finding the optimal solution and mitigates the occurrence of local optima. Utilizing this algorithm leads to a significant reduction in Q-learning’s convergence time.

Then, the Q-learning algorithm embarks on environmental exploration by building upon the prior knowledge encoded in the initialized Q-table. In this study, the aircraft is unrestricted in eight directions, comprising its action set: north (up), south (down), (west) left, (east) right, northwestward (left-up), southwestward (left-down), northeastward (right-up), and southeastward (right-down). The schematic of these sets of actions is depicted in Fig. 3(a). The key to exploring the environment is strategically selecting actions and formulating reward functions.

The commonly used action selection strategy is the ϵ -greedy strategy, which aims to introduce a certain level of randomness in the agent’s decision-making process. Specifically, the agent has a probability of $1-\epsilon$ to select the action with maximum reward value and a probability ϵ to choose an action randomly. In this paper, we adopt the following action

selection strategy:

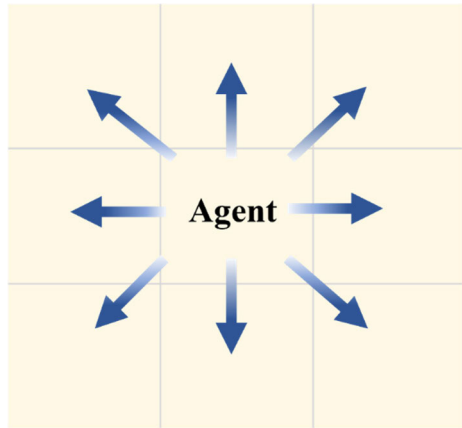
$$\epsilon(n) = (\epsilon_{\max} - \epsilon_{\min}) \frac{n_{\max} - n}{n_{\max}} + \epsilon_{\min} \quad (5)$$

where n is the current iteration count, $\epsilon(n)$ is the ϵ value of the n th iteration. The parameter ϵ decreases with increasing n . Therefore, ϵ_{\max} is the value of ϵ at the beginning of the iteration, ϵ_{\min} is the value of ϵ at n_{\max} iterations, and n_{\max} is the maximum number of iterations.

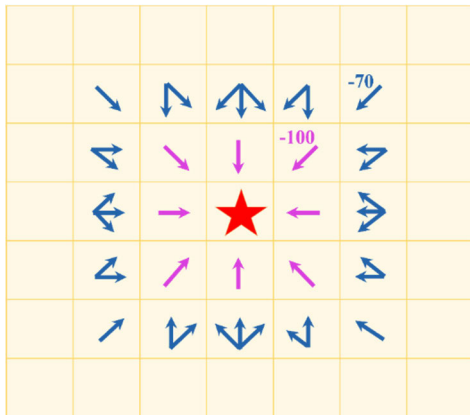
As the optimal policy gradually approaches convergence in the later stages of exploration, there is no longer a need to assign a high probability for random action selection. Therefore, this paper proposes a linear decrease in ϵ with an increase in the number of iterations to enhance confidence in the optimal policy during these later stages. Furthermore, the reward function assesses the quality of action selection, and its configuration directly influences the impact of agent-environment interaction. In this study, we define the reward function as follows:

$$R(t) = \begin{cases} 10 & \text{Reach the goal point} \\ -1 + \frac{d(t-1) - d(t)}{d_{\max}} & \text{Select action: up,} \\ & \text{down, left, right} \\ -1.41 + \frac{d(t-1) - d(t)}{d_{\max}} & \text{Select action: left-up,} \\ & \text{left-down, right-up,} \\ & \text{right-down} \\ -100 & \text{Collision obstacles} \end{cases} \quad (6)$$

The variable $d(t)$ represents the Euclidean distance between the agent and the goal point at time t , whereas



(a) Illustration of UAV action set



(b) Schematic diagram of local obstacle avoidance based on tabulation method

FIGURE 3. Schematic of action impact of the UAV.

d_{max} denotes the distance between the starting and ending points. As the agent approaches the target point, it receives a positive reward value, whereas moving away from the target results in a negative reward. When the agent approaches the target point, it receives a relatively high reward value, whereas when it moves away from the target point, it receives a relatively low reward value. This incentivizes the agent to move towards the goal point. Furthermore, since diagonal movement is more efficient for the agent, the reward function also encourages oblique flying.

B. LOCAL OBSTACLE AVOIDANCE BASED ON THE TABULATION METHOD

Since the Q-table trained by the original algorithm is discarded after finding the optimal path, which wastes computational resources, this paper has adequately utilized the record-ability of the Q-table to its surrounding environment by directly processing it to save computing time and memory. Specifically, when encountering a new obstacle, the aircraft will modify the value of states moving towards it in the Q-table so that their Q values become negative. Furthermore, this modification can continue spreading out for a week to achieve gradient descent in reducing Q values.

This approach ensures agility and timeliness for UAVs while avoiding unnecessary training time.

As depicted in Fig. 3(b), the marked positions with pentagram indicate the locations of obstacles. Once detected, the Q value of the state surrounding the obstacle is immediately reduced by 100, whereas the Q value of the corresponding state one layer further out can decrease by 70. This gradient descent approach effectively mitigates repetitive training and enhances reaction speed.

Algorithm 3 Improved Q-Learning Algorithm Based on Both Flower Pollination Algorithm and Tabulation Method

- Input:** start point, goal point, and environment information
- Output:** the optimal path from the start to the end
- 1 execute Algorithm 2
- 2 obtain lots of position coordinates
- 3 set a large Q-value to the states corresponding to the coordinates in G
- 4 set a larger negative Q-value to the states corresponding to the obstacle
- 5 obtain the Q-table with prior information
- 6 execute Algorithm 1 except for initialization
- 7 obtain the optimal path
- 8 **while** the new obstacle
- 9 obtain the states corresponding to the obstacle coordinates
- 10 set a large negative Q-value for these states using gradient descent
- 11 obtain the optimal path using Q-table
- 12 **end**

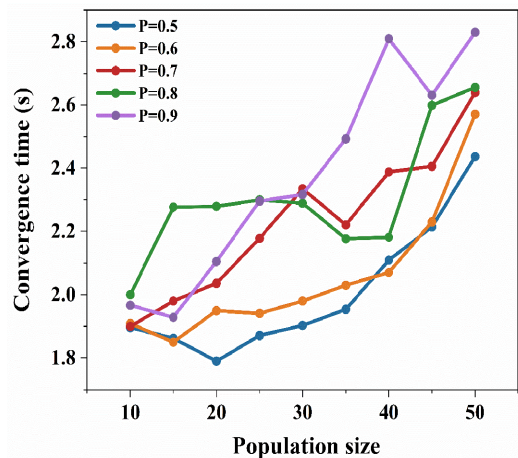


FIGURE 4. Variation in convergence time with different switch probabilities and population sizes used in the IQ-FAT in map.

IV. SIMULATION TEST OF IQ-FAT ALGORITHM

This section evaluates IQ-FAT in various scenarios, including global and local path planning. The three-dimensional (3D) map size used is 20*20*10. The initial and final coordinates of the UAV are designated at (0.5,0.5,0) and (19.5, 19.5,10), denoted by a purple triangle and a green five-pointed star,

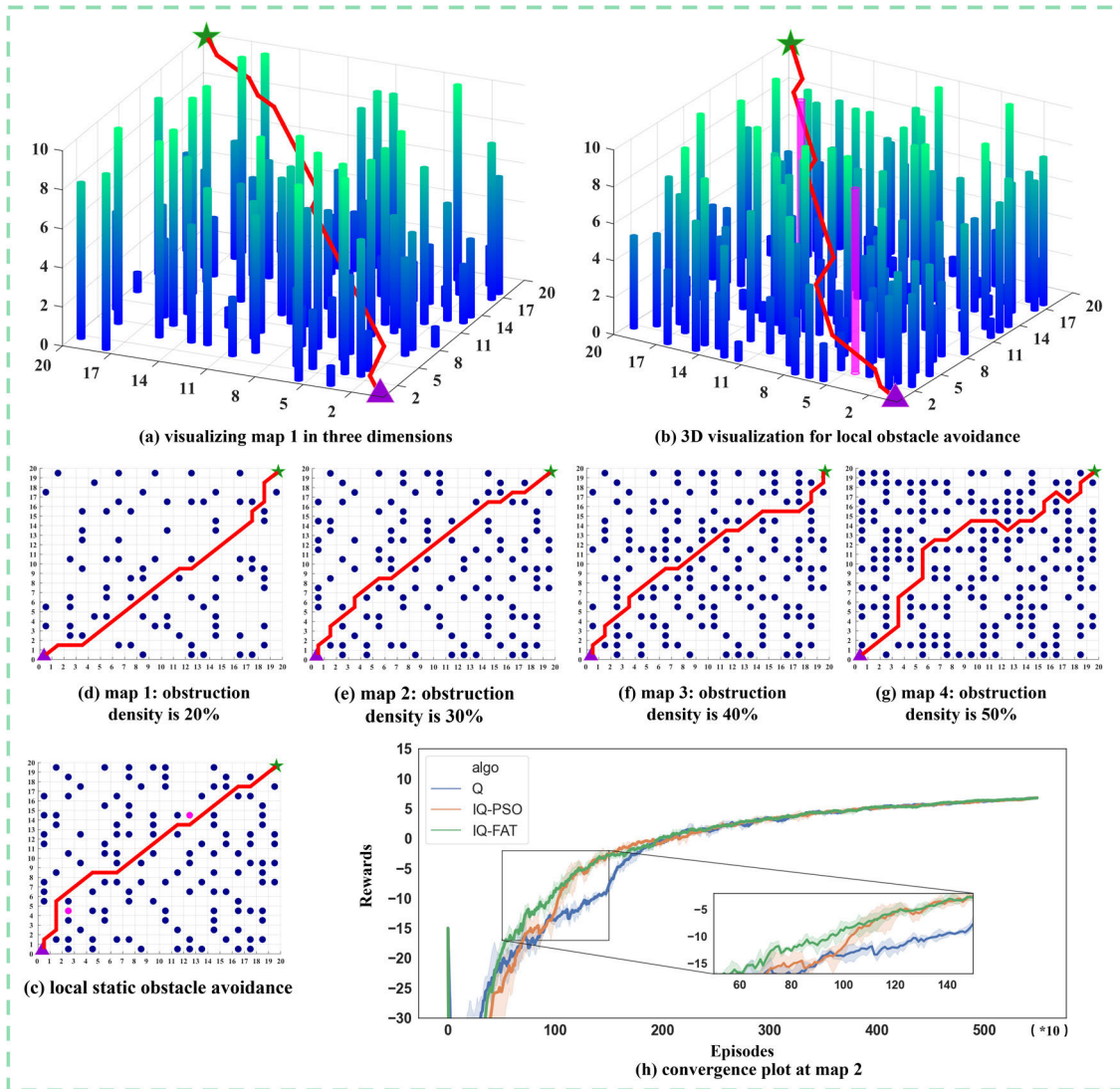


FIGURE 5. Path planning diagram and convergence diagram of maps 1-4.

TABLE 1. Obstacle avoidance time comparison among IQ-FAT, APF, and ACA.

| | Algorithm | Reaction time |
|--------|-----------|---------------|
| map 2 | IQ-FAT | 1ms |
| | APF | 7.4ms |
| | ACA | 8.9ms |
| map 5 | IQ-FAT | 0.4ms |
| | APF | 4.2ms |
| | ACA | 5.8ms |
| map 10 | IQ-FAT | 0.2ms |
| | APF | 3.6ms |
| | ACA | 4.7ms |

respectively. The two-dimensional (2D) map size used is 20*20, and the starting and ending points of the aircraft are

set at (0.5,0.5) and (19.5, 19.5), respectively. Each action's Q value related to a specific position is stored in a Q-table of size 400*8. Meanwhile, this subsection presents a comparative analysis of IQ-FAT with conventional Q-learning(Q) and improved Q-learning using particle swarm optimization (IQ-PSO). It's worth mentioning that IQ-PSO is another algorithm we've developed, which utilizes the PSO algorithm instead of the FPA method to enhance the initialization process of Q-learning.

A. OBSTACLE AVOIDANCE SIMULATION TEST FOR HIGH-DENSITY REGULAR OBSTACLES

In this test, a grid map contains random obstacles, which depict a complex environment characterized by a high density of blocks. To begin with, different environments require distinct search transition switch probabilities p and population sizes for the flower pollination method to achieve optimal results. Therefore, the convergence time is depicted in Fig. 4 for various switch probabilities and population sizes.

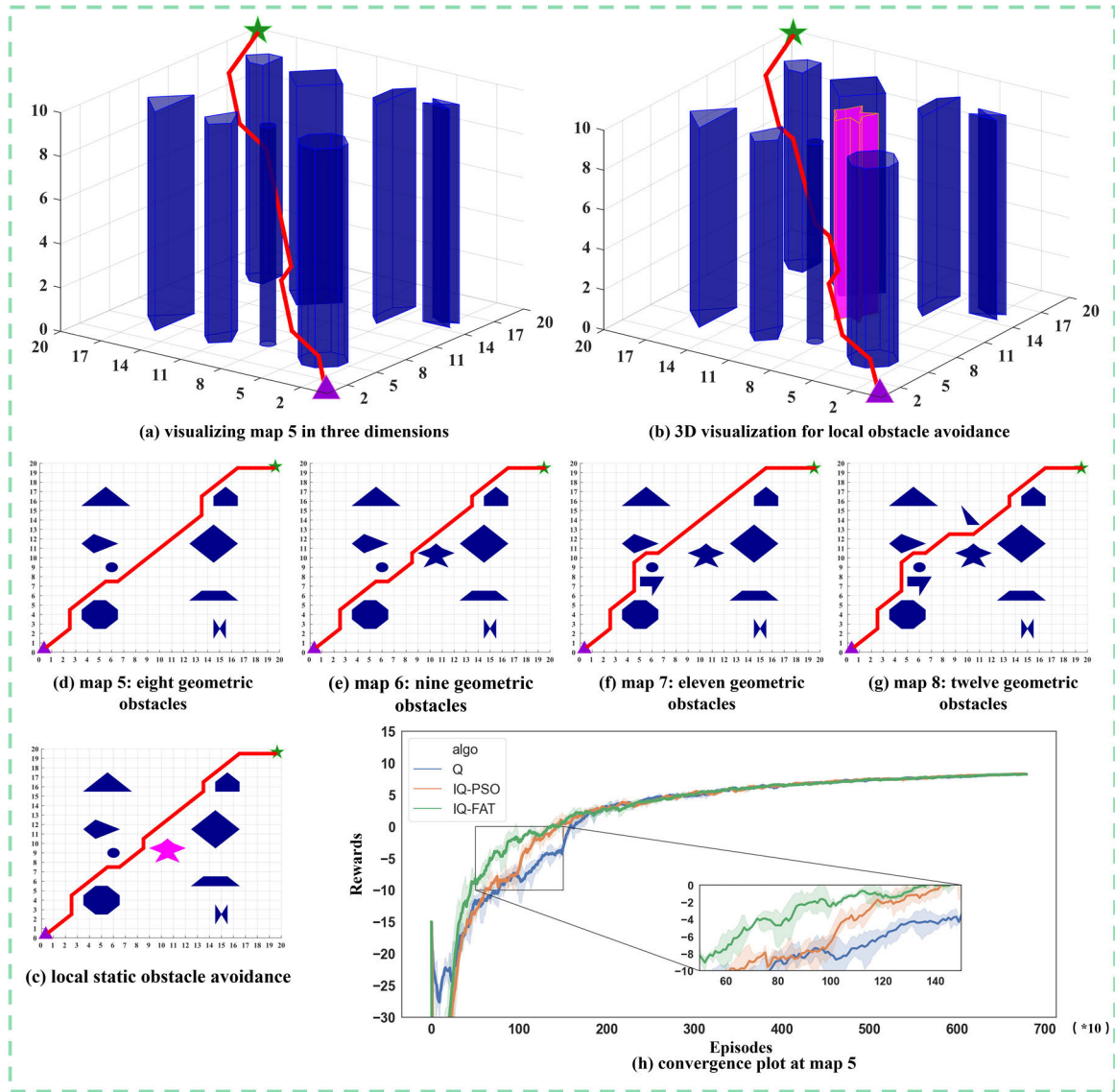


FIGURE 6. Path planning diagram and convergence diagram of maps 5-8.

Fig. 4 demonstrates the time the UAV took to complete path planning under different p values and population sizes in an environment with 30% obstacle density. The overall experimental outcomes are superior when $p=0.5$. However, it is worth noting that the convergence time generally exhibits an upward trend, with occasional decreases observed for larger population sizes. Therefore, considering a comprehensive perspective, this study opts for $p=0.5$ and a population size of 20 to conduct path-planning tasks. Since similar results were obtained on other maps, we refrain from repeating them herein. Instead, we adopt this combination approach for all other maps.

The IQ-FAT can productively plan the optimal path within a designated three-dimensional space altitude. Furthermore, it effectively achieves sub-optimal 3D path planning for aircraft by implementing a uniform ascent strategy

from the initial point to the destination, as exemplified in Figs. 5(a) (b). The blue graphics represent obstacles, and the pink graphic is the obstacle without prior information.

Subsequently, the paths planned by the IQ-FAT on maps with obstacle densities of 20%, 30%, 40%, and 50% are depicted as (d)-(g) in Fig. 5. Experimental results demonstrate that the algorithm completes path planning even at an obstacle density of 50%. Additionally, Fig. 5(c) assesses IQ-FAT's ability in local obstacle avoidance. IQ-FAT promptly responds and re-plans its trajectory when confronted with obstacles beyond the scope of prior information. Moreover, Fig. 5(h) demonstrates the convergence speeds of Q, IQ-PSO, and IQ-FAT under this map configuration. Notably, IQ-FAT achieves faster convergence rates accompanied by smoother convergence curves than other approaches

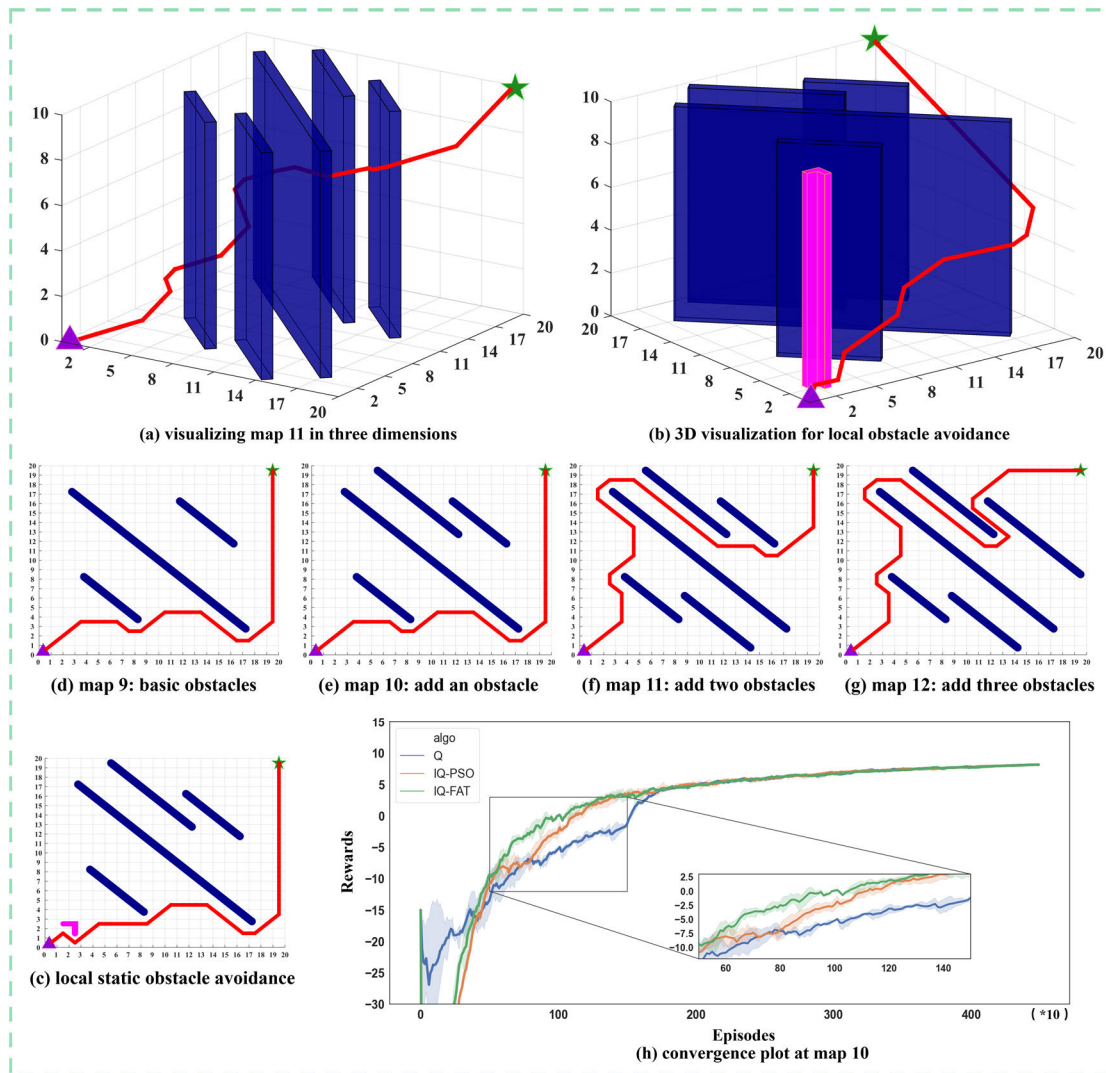


FIGURE 7. Path planning diagram and convergence diagram of maps 9-12.

tested herein, thus confirming its superiority in convergence speed and stability.

Additionally, this paper employs a variety of distinct methodologies for comparison: the traditional Q-learning algorithm and its variant IQ-PSO for global path planning comparison. Comparative analysis of various meta-heuristic algorithms with IQ-FAT in terms of path quality. The classical artificial potential field method (APF) and ant colony algorithm (ACA) for local obstacle avoidance experiments. The comparison results presented in Table 1 demonstrate that the IQ-FAT exhibits significantly enhanced reaction speed compared to the APF and ACA, rendering it highly suitable for low-load UAVs with limited processing time.

To further analyze and compare the performance of IQ-FAT and IQ-PSO, we present the convergence times and the number of convergences for both methods on various maps, along with their corresponding promotion ratios. Fig. 8 compares the convergence times and convergence time among the three algorithms. It is evident that as obstacle

density increases, all three algorithms exhibit improved convergence times and convergence time. Specifically, IQ-FAT demonstrates a 45%-60% enhancement in convergence times compared to the original algorithm, with this improvement ratio diminishing as map complexity increases. This phenomenon may arise from the intricate nature of the map, necessitating additional exploration iterations for enhanced Q-learning convergence. Moreover, when contrasted with the original algorithm, IQ-FAT exhibits an improvement ratio between 40% and 55% in terms of convergence time, which escalates alongside increasing map complexity. These findings indicate that IQ-FAT displays superior adaptability to complex environments and outperforms IQ-PSO in two key aspects.

B. OBSTACLE AVOIDANCE SIMULATION TEST FOR IRREGULAR GEOMETRY OBSTACLES

In practical scenarios, obstacles cannot be treated as mere particles. Hence, we constructed geometrically shaped obstacles

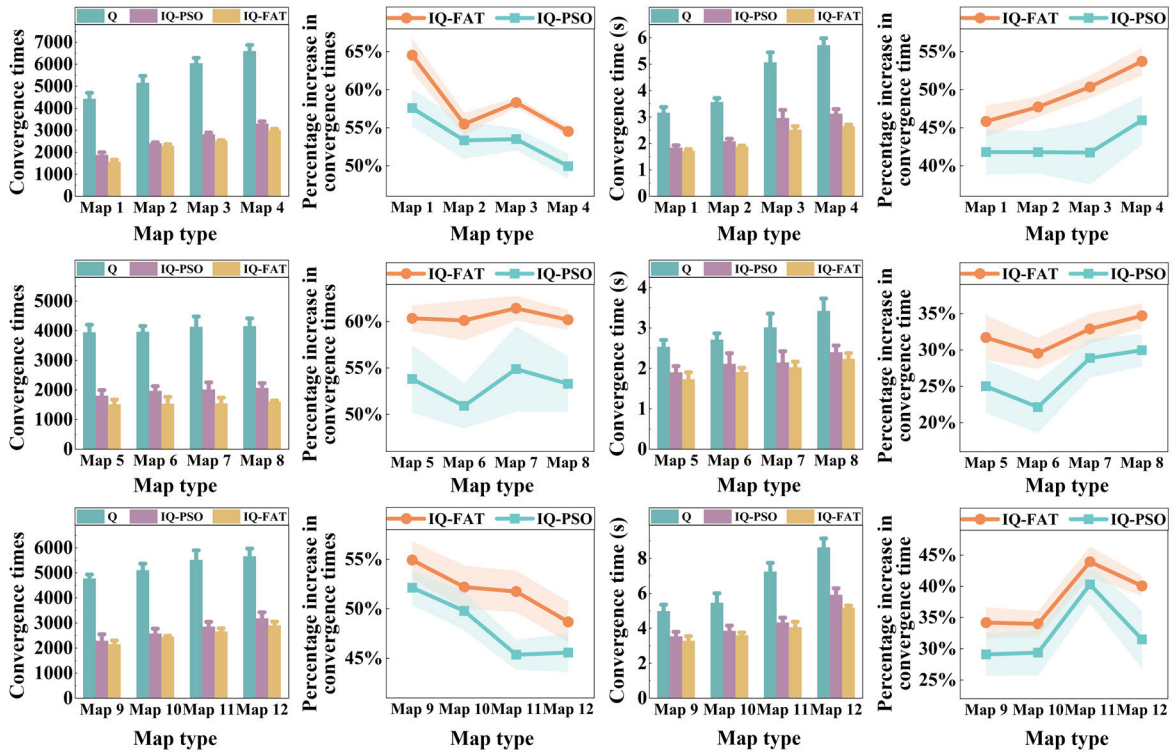


FIGURE 8. Convergence times, convergence time, and growth ratios of algorithms.

TABLE 2. Convergence times, convergence time, and growth ratios of algorithms in maps1-12.

| Map type | Convergence time(s) | | | | Convergence times | | | |
|----------|---------------------|--------|--------|-------------------------------|-------------------|--------|--------|-------------------------------|
| | Q | IQ-PSO | IQ-FAT | Growth ratios (IQ-PSO/IQ-FAT) | Q | IQ-PSO | IQ-FAT | Growth ratios (IQ-PSO/IQ-FAT) |
| Map1 | 3.160 | 1.839 | 1.711 | 41.8%/45.8% | 4420 | 1874 | 1567 | 57.6%/64.5% |
| Map2 | 3.561 | 2.077 | 1.874 | 41.8%/47.7% | 5145 | 2400 | 2290 | 53.4%/55.5% |
| Map3 | 5.065 | 2.956 | 2.514 | 41.7%/50.4% | 6036 | 2806 | 2516 | 53.5%/58.31% |
| Map4 | 5.714 | 3.120 | 2.644 | 45.9%/53.7% | 6592 | 3299 | 2998 | 49.9%/54.6% |
| Map5 | 2.530 | 1.897 | 1.728 | 25.0%/31.7% | 3943 | 1801 | 1513 | 53.8%/60.4% |
| Map6 | 2.707 | 2.108 | 1.907 | 22.1%/29.5% | 3960 | 1962 | 1529 | 50.9%/60.1% |
| Map7 | 3.012 | 2.142 | 2.021 | 28.9%/32.9% | 4131 | 2013 | 1543 | 54.9%/61.44% |
| Map8 | 3.419 | 2.395 | 2.233 | 30.0%/34.7% | 4151 | 2063 | 1602 | 53.3%/60.2% |
| Map9 | 4.976 | 3.528 | 3.275 | 29.1%/34.2% | 4772 | 2284 | 2151 | 52.1%/54.9% |
| Map10 | 5.448 | 3.847 | 3.596 | 29.4%/34.0% | 5101 | 2572 | 2442 | 49.8%/52.2% |
| Map11 | 7.239 | 4.322 | 4.059 | 40.3%/44.1% | 5510 | 2848 | 2658 | 45.4%/51.7% |
| Map12 | 8.626 | 5.909 | 5.170 | 31.5%/40.1% | 5657 | 3176 | 2902 | 45.6%/48.7% |

on the map to test whether our algorithm can effectively plan paths around them. Firstly, we created a primary environment with eight obstacles and then gradually increased the number of obstacles along the route, drawing maps with nine, ten, and eleven obstacles for comparison. As shown in Figs. 6(d)-(g),

our experimental results demonstrate that IQ-FAT can avoid geometrically shaped obstacles while maintaining a safe distance.

Moreover, the local obstacle avoidance experiment depicted in Fig. 6(c) involves placing an unknown obstacle of

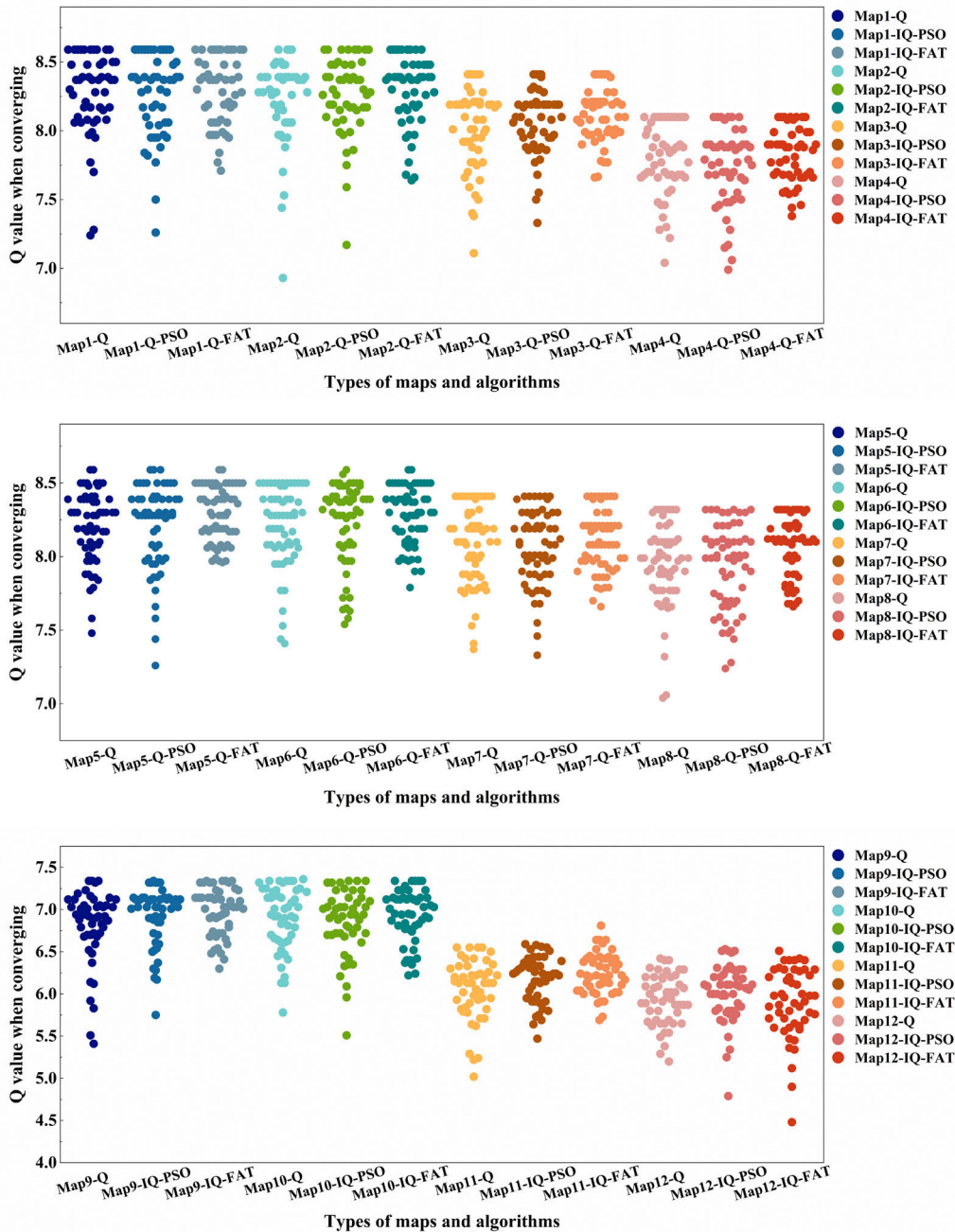


FIGURE 9. Q-values at the convergence of algorithms in maps 1-12.

a specific shape on the planned trajectory of the UAV to assess its ability for local obstacle avoidance. Based on Table 1 and Fig. 6(c), IQ-FAT effectively avoids such unknown obstacles by employing a tabulation method, exhibiting significantly reduced response time compared to the artificial potential field approach. Furthermore, agents showed enhanced response times when encountering obstacles with a specific geometric configuration as opposed to barriers composed of a solitary particle, potentially attributable to the sparse distribution of impediments throughout the map, thereby affording more excellent maneuverability.

As for the comparison of convergence time and convergence times of the three algorithms on such maps, Fig. 8 shows that the improvement ratio of IQ-FAT in terms of convergence times is approximately 60%, exhibiting minimal fluctuations. In contrast, the enhancement effect of IQ-PSO fluctuates between 50% and 55%. Meanwhile, increasing obstacles does not result in a significant increase in the convergence times. This is because adding geometric obstacles does not significantly increase the obstacle density of the map, and therefore, the algorithm only requires a little trial-and-error time to operate.

TABLE 3. The length and smoothness of the paths planned by five algorithms in maps1-12. (The path smoothness is calculated by the second-order difference method.)

| Map type | Path length(km) | | | | | Path smoothness | | | | |
|----------|-----------------|--------|--------|--------|--------|-----------------|--------|--------|--------|--------|
| | IQ-FAT | GA | SSA | GWO | A* | IQ-FAT | GA | SSA | GWO | A* |
| Map1 | 28.627 | 28.662 | 28.761 | 30.201 | 36.609 | 7 | 17.414 | 13.445 | 13.922 | 33.431 |
| Map2 | 28.627 | 30.889 | 30.019 | 30.582 | 36.661 | 11 | 23.121 | 23.294 | 20.861 | 34.438 |
| Map3 | 29.799 | 34.924 | 32.541 | 34.876 | 37.982 | 13 | 24.075 | 25.967 | 25.703 | 36.557 |
| Map4 | 33.213 | / | / | / | 39.238 | 19 | / | / | / | 40.898 |
| Map5 | 29.213 | 29.342 | 29.305 | 29.204 | 29.813 | 7 | 25.225 | 25.211 | 18.728 | 22.367 |
| Map6 | 29.213 | 29.475 | 29.475 | 29.305 | 30.218 | 9 | 25.405 | 25.409 | 25.211 | 26.539 |
| Map7 | 29.799 | 30.896 | 30.654 | 29.919 | 33.624 | 7 | 21.552 | 25.333 | 18.715 | 30.946 |
| Map8 | 30.384 | 34.345 | 34.007 | 32.991 | 34.562 | 11 | 18.414 | 19.527 | 22.663 | 32.451 |
| Map9 | 39.556 | 43.871 | 43.112 | 42.720 | 42.731 | 9 | 24.893 | 18.344 | 19.122 | 30.893 |
| Map10 | 39.556 | 43.871 | 43.112 | 42.720 | 45.491 | 9 | 24.893 | 18.344 | 19.122 | 33.173 |
| Map11 | 49.6985 | 58.986 | 55.123 | 54.629 | 58.775 | 15 | 30.649 | 28.763 | 29.631 | 36.924 |
| Map12 | 51.9411 | / | / | / | 62.912 | 17 | / | / | / | 38.223 |

The correlation between convergence time and obstacle geometry is weak, whereas the correlation between convergence time and obstacle density is strong across the entire map. This also can be seen in the agent's walking path, which remains relatively easy to plan even with added obstacles and does not significantly change its trajectory. Regarding convergence time, IQ-FAT achieves a range of 30% to 35%, with an overall increase in the lifting ratio as obstacles intensify. Based on experimental results, all three algorithms demonstrate remarkable path-planning capabilities when confronted with geometric obstacles. The IQ-FAT algorithm exhibits superior stability and marginally improved convergence time compared to the other two algorithms.

C. OBSTACLE AVOIDANCE SIMULATION TEST IN THE INDOOR ENVIRONMENT

In the indoor environment, obstacles with specific lengths pose challenges for traditional algorithms during optimization, often resulting in local optima and subsequent failure of path planning. Hence, this study evaluates the ability of IQ-FAT to navigate obstacles with specific lengths smoothly.

As shown in Fig. 7(d), the base environment has three long side walls, which makes path planning more challenging. Furthermore, the environment in Fig. 7(d) is symmetrical, which means there are two equiprobable paths for the agent

to arrange. Subsequently, when an upper obstacle is added in Fig. 7(e), the UAV can only schedule one optimal route. Continuing to Fig. 7(f), obstacles are placed along the path, which forces the UAV to fly alongside them until it can avoid them. In Fig. 7(g), additional obstacles have been introduced that block the original optimal path of the UAV. Nonetheless, the UAV still finds an optimal way through two closely spaced long obstacles.

Then, Fig. 7(c) introduces a right-angle obstacle to evaluate the local obstacle avoidance capability. Right-angle obstacles pose more significant challenges than others, demanding the UAV make prompt and precise decisions. Nevertheless, IQ-FAT consistently demonstrates its exceptional ability in obstacle avoidance while maintaining remarkable reaction speed.

Fig. 8 and Table 2 illustrate that as the number of obstacles increases on the indoor environment map, there is an overall upward trend in convergence times. However, this increase is insignificant. The growth proportion of convergence times for IQ-FAT and IQ-PSO shows a downward trend. Specifically, IQ-FAT improves convergence times by 47% to 55%, whereas IQ-PSO improves them by 30% to 40%. In contrast, Q-learning experiences a significant increase in convergence time with increasing obstacles. The improvement ratio for IQ-FAT ranges from 45% to 53%, and

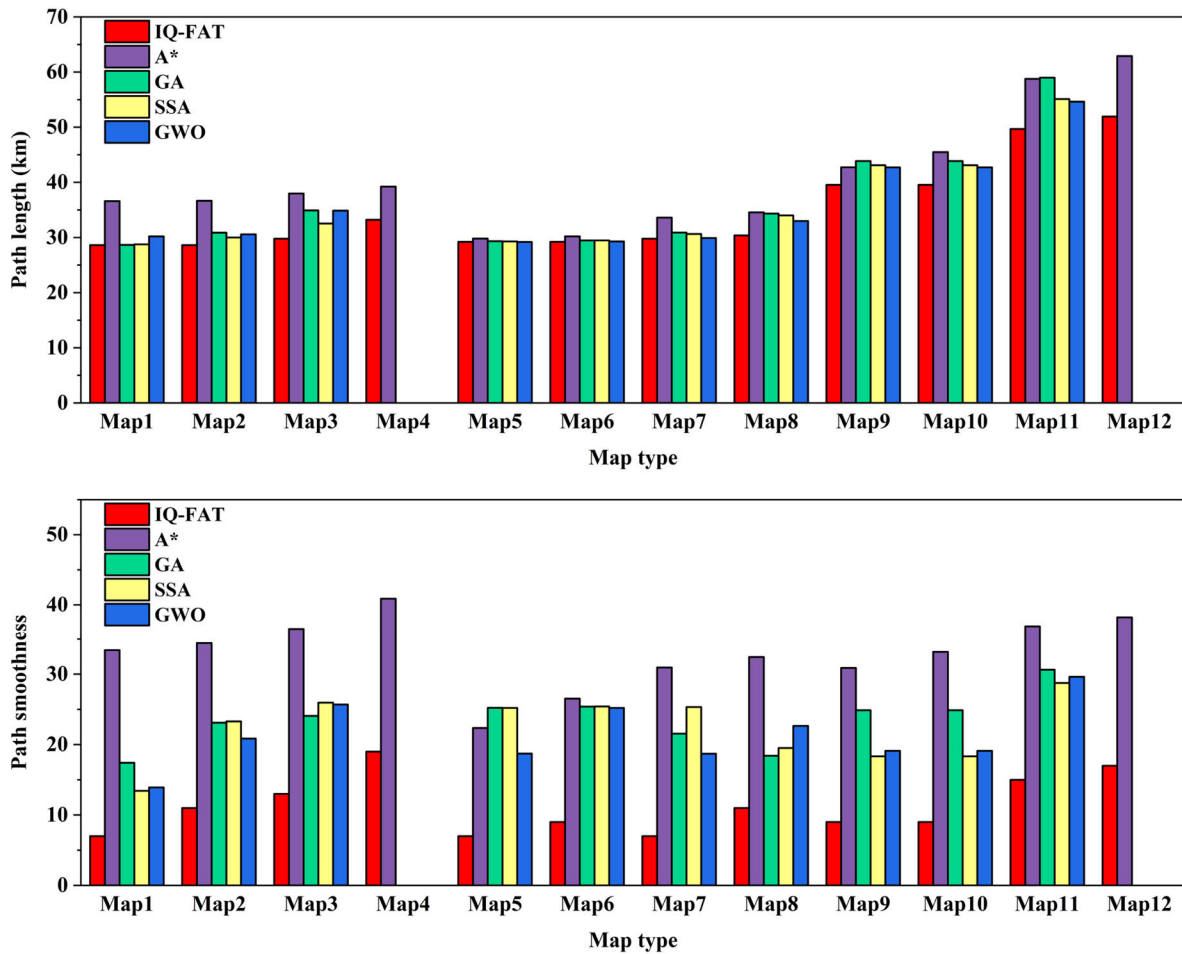


FIGURE 10. The length and smoothness of the paths planned by five algorithms in maps 1-12.

for IQ-PSO, it is between 30% and 40%. The growth ratio of both exhibits an upward trend overall.

Moreover, through a comprehensive comparison of the algorithm's performance across the three maps, we can ascertain the influence of obstacles on the algorithm's convergence. Compared to dense particle obstacles and geometric shapes, indoor obstacles with a specific length significantly enhance the complexity of Q and IQ-FAT for path planning.

Specifically, from the perspective of the fundamental environment comprising indoor obstacles, the obstacle density is approximately 19.5%. However, it takes an additional 1.796s for convergence compared to maps with particle obstacles at a thickness of 20%, resulting in a time increase of 36.24%. Hence, it can be concluded that obstacle length influences path planning difficulty. Meanwhile, longer obstacles also impact the number of steps required for aircraft navigation, necessitating more bypassing maneuvers. In this process, traditional algorithms are prone to local optima. However, Q and IQ-FAT exhibit certain advantages while ensuring feasibility in planned paths.

To further evaluate the convergence of the three algorithms, we recorded the Q-values of Q-learning, IQ-PSO,

TABLE 4. Dynamic obstacle avoidance time comparison among IQ-FAT, APF, and ACA.

| | Algorithm | Reaction time |
|-----------------------------|-----------|---------------|
| The first dynamic obstacle | IQ-FAT | 0.9ms |
| | APF | 6.5ms |
| | ACA | 7.3ms |
| The second dynamic obstacle | IQ-FAT | 1.2ms |
| | APF | 7.0ms |
| | ACA | 7.1ms |

and IQ-FAT as they converged on different maps. We then selected 50 data points from each group to generate Fig. 9 and observe the convergence behavior of each algorithm.

In a comprehensive comparison, the convergent Q-value obtained in the map with high obstacle density is the highest, whereas the convergent Q-value obtained in the indoor environment map when solving long obstacle path planning problems is the lowest. Hence, the convergence effect of final Q-learning is influenced by the environment where obstacles are encountered. Furthermore, across all experiments, there is

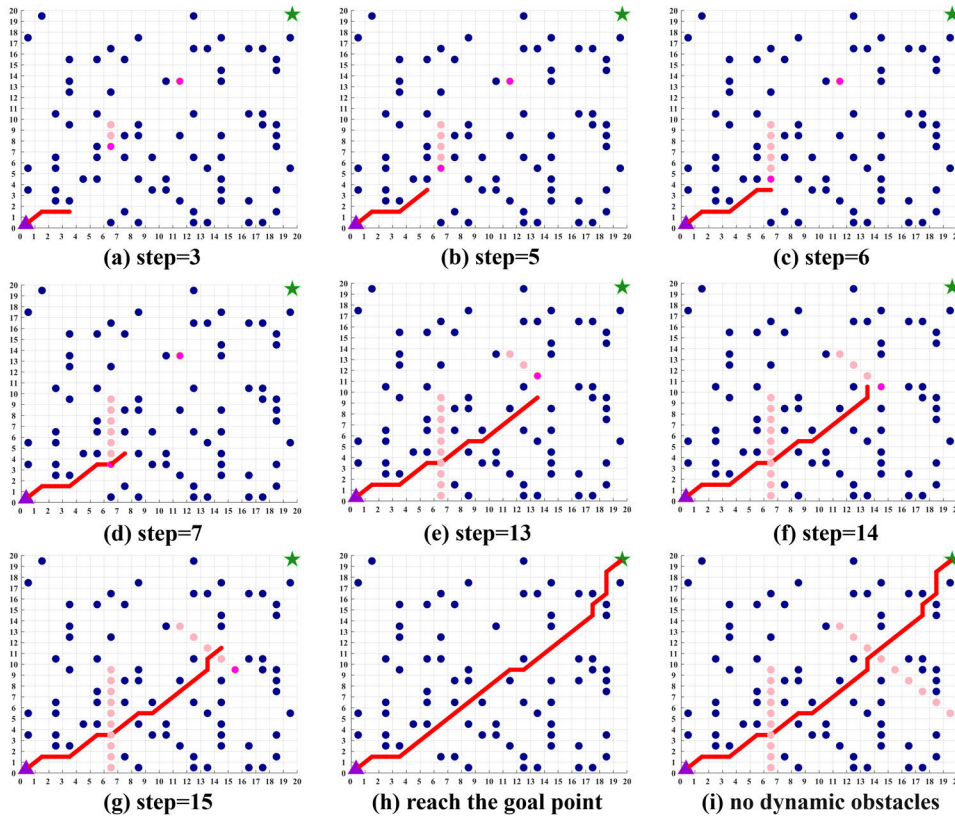


FIGURE 11. The motion trajectory of the UAV under dynamic obstacles.

a significant decrease in convergence Q-value as environmental complexity increases. Therefore, ecological complexity also affects algorithm convergence, and higher complexity leads to lower convergence values. IQ-FAT exhibits superior convergence characteristics on each map and demonstrates more concentrated convergent Q-values than the other two algorithms.

In addition to horizontally comparing Q learning and its variants, we also conduct comparisons between Q learning and the A* algorithm as well as various intelligent algorithms, including genetic algorithm (GA), sparrow search algorithm (SSA), and gray Wolf Optimizer (GWO). We employ five algorithms for path planning in maps 1-12 and to compute the length of its two-dimensional trajectory, as illustrated in Table 3 and Fig. 10. The experimental results show that the path generated by Q-learning exhibits a small deviation in length from those produced by heuristic algorithms in some straightforward environments. As the complexity of the environment increases, the paths navigated by intelligent algorithms become longer. In map 4 and map 12, the intelligent algorithms are even unable to determine the location of the endpoint due to the complexity of the environment. In terms of path smoothness, the trajectory generated by Q-learning outperforms that of other algorithms. In conclusion, Q learning demonstrates superior path-planning capabilities in intricate environments, yielding shorter and smoother paths.

D. SIMULATION TEST OF DYNAMIC OBSTACLE AVOIDANCE

In this scenario, the efficacy of IQ-FAT in efficiently navigating dynamic obstacles is evaluated. Firstly, the detectability of the obstacle was assumed, but its positional information was not included in the training process. Next, two moving obstacles were introduced: one descending and the other diagonally falling to the right, represented by pink circles. The trajectory of these dynamic obstacles is visible as light-colored circles, whereas blue circles represent known static obstacles. The motion results of the UAV's dynamic obstacle avoidance are depicted in Fig. 11.

According to Fig. 11, in case of an impending collision with a barrier at the next node, the UAV will navigate in a relatively safer direction to avoid it along its planned trajectory. The above comparison demonstrates the capability of the IQ-FAT algorithm to address dynamic obstacles.

In addition, we compared the reaction time of the IQ-FAT, artificial potential field method (APF), and ant colony algorithm (ACA) to avoid dynamic obstacles. Table 4 indicates that IQ-FAT has the shortest reaction time when facing dynamic obstacles.

V. CONCLUSION

Committed to addressing the requirements of efficient path planning for UAVs in complex environments, this paper

presents a new algorithm called IQ-FAT. IQ-FAT combines the Q-learning algorithm and flower pollination algorithm to plan two-dimensional paths in eight directions. Additionally, the algorithm includes the tabulation method to enhance its effectiveness in local obstacle avoidance. IQ-FAT utilizes the flower pollination algorithm to enhance the initial Q-value along the line connecting the starting and endpoints, thereby providing Q-learning with more prior information, reducing exploration process blindness, and improving convergence speed. Moreover, IQ-FAT exploits the Q-table by employing gradient-based techniques to handle static or dynamic obstacles outside of prior knowledge to enhance local obstacle avoidance reaction speed.

The IQ-FAT algorithm demonstrates superior convergence speed and accuracy compared to other algorithms, as evidenced by its performance in a simulation test across 12 maps. Furthermore, IQ-FAT effectively accomplishes path planning in a three-dimensional space. The path quality of IQ-FAT planning is also excellent, especially in complex environments. Moreover, IQ-FAT can handle unexpected obstacles that may appear suddenly, demonstrating its versatility and adaptability. For example, IQ-FAT exhibits significantly improved reaction speed when dealing with static blocks outside of prior information, surpassing the capabilities of the artificial potential field method. Meanwhile, the IQ-FAT proves to be effective in handling dynamic situations.

The IQ-FAT is of great significance and reference value for UAVs to effectively navigate complex environments and swiftly accomplish obstacle avoidance. However, important objectives and constraints such as navigation errors have not been considered in this method. Additionally, the algorithm has only been validated through simulation experiments, and real-world experimentation will be considered at a later stage. Furthermore, the algorithm will incorporate neural networks, which are anticipated to play a pivotal role in future advancements. We will further enhance the effectiveness and practicability of the algorithm through conducting more comprehensive research.

REFERENCES

- [1] J. E. Lavín-Delgado, Z. Zamudio Beltrán, J. F. Gómez-Aguilar, and E. Pérez-Careta, "Controlling a quadrotor UAV by means of a fractional nested saturation control," *Adv. Space Res.*, vol. 71, no. 9, pp. 3822–3836, May 2023.
- [2] Y. Chen, Q. Dong, X. Shang, Z. Wu, and J. Wang, "Multi-UAV autonomous path planning in reconnaissance missions considering incomplete information: A reinforcement learning method," *Drones*, vol. 7, no. 1, p. 10, Dec. 2022.
- [3] W. Yi, M. Sutrisna, and H. Wang, "Unmanned aerial vehicle based low carbon monitoring planning," *Adv. Eng. Informat.*, vol. 48, Apr. 2021, Art. no. 101277.
- [4] S. Huang, R. S. H. Teo, and K. K. Tan, "Collision avoidance of multi unmanned aerial vehicles: A review," *Annu. Rev. Control*, vol. 48, pp. 147–164, Jan. 2019.
- [5] J. N. Yasin, S. A. S. Mohamed, M.-H. Haghbayan, J. Heikkinen, H. Tenhunen, and J. Plosila, "Unmanned aerial vehicles (UAVs): Collision avoidance systems and approaches," *IEEE Access*, vol. 8, pp. 105139–105155, 2020.
- [6] A. Ait-Saadi, Y. Meraihi, A. Soukane, S. Yahia, A. Ramdane-Cherif, and A. B. Gabis, "An enhanced African vulture optimization algorithm for solving the unmanned aerial vehicles path planning problem," *Comput. Electr. Eng.*, vol. 110, Sep. 2023, Art. no. 108802.
- [7] S. Aslan and T. Erkin, "A multi-population immune plasma algorithm for path planning of unmanned combat aerial vehicle," *Adv. Eng. Informat.*, vol. 55, Jan. 2023, Art. no. 101829.
- [8] Y. Shin and E. Kim, "Hybrid path planning using positioning risk and artificial potential fields," *Aerosp. Sci. Technol.*, vol. 112, May 2021, Art. no. 106640.
- [9] G. Zhang, Y. Deng, W. Zhang, and C. Huang, "Novel DVS guidance and path-following control for underactuated ships in presence of multiple static and moving obstacles," *Ocean Eng.*, vol. 170, pp. 100–110, Dec. 2018.
- [10] Z. Zhang, J. Wu, J. Dai, and C. He, "A novel real-time penetration path planning algorithm for stealth UAV in 3D complex dynamic environment," *IEEE Access*, vol. 8, pp. 122757–122771, 2020.
- [11] J. Fan, X. Chen, and X. Liang, "UAV trajectory planning based on bi-directional APF-RRT* algorithm with goal-biased," *Expert Syst. Appl.*, vol. 213, Mar. 2023, Art. no. 119137.
- [12] Q.-C. Luo, K.-W. Sun, T. Chen, Y.-F. Zhang, and Z.-W. Zheng, "Trajectory planning of stratospheric airship for station-keeping mission based on improved rapidly exploring random tree," *Adv. Space Res.*, vol. 73, no. 1, pp. 992–1005, Jan. 2024.
- [13] Y. Guo, X. Liu, Q. Jia, X. Liu, and W. Zhang, "HPO-RRT*: A sampling-based algorithm for UAV real-time path planning in a dynamic environment," *Complex Intell. Syst.*, vol. 9, no. 6, pp. 7133–7153, Jun. 2023.
- [14] J. Zhang, J. Yan, and P. Zhang, "Fixed-wing UAV formation control design with collision avoidance based on an improved artificial potential field," *IEEE Access*, vol. 6, pp. 78342–78351, 2018.
- [15] O. Montiel, U. Orozco-Rosas, and R. Sepúlveda, "Path planning for mobile robots using bacterial potential field for avoiding static and dynamic obstacles," *Expert Syst. Appl.*, vol. 42, no. 12, pp. 5177–5191, Jul. 2015.
- [16] P. Váña and J. Faigl, "Optimal solution of the generalized Dubins interval problem: Finding the shortest curvature-constrained path through a set of regions," *Auto. Robots*, vol. 44, no. 7, pp. 1359–1376, Sep. 2020.
- [17] W. Ye, D.-W. Ma, and H.-D. Fan, "Algorithm for low altitude penetration aircraft path planning with improved ant colony algorithm," *Chin. J. Aeronaut.*, vol. 18, no. 4, pp. 304–309, Nov. 2005.
- [18] S. Shao, Y. Peng, C. He, and Y. Du, "Efficient path planning for UAV formation via comprehensively improved particle swarm optimization," *ISA Trans.*, vol. 97, pp. 415–430, Feb. 2020.
- [19] E. S. Low, P. Ong, and K. C. Cheah, "Solving the optimal path planning of a mobile robot using improved Q-learning," *Robot. Auto. Syst.*, vol. 115, pp. 143–161, May 2019.
- [20] C. YongBo, M. YueSong, Y. JianQiao, S. XiaoLong, and X. Nuo, "Three-dimensional unmanned aerial vehicle path planning using modified wolf pack search algorithm," *Neurocomputing*, vol. 266, pp. 445–457, Nov. 2017.
- [21] X. Yu and W. Luo, "Reinforcement learning-based multi-strategy cuckoo search algorithm for 3D UAV path planning," *Expert Syst. Appl.*, vol. 223, Aug. 2023, Art. no. 119910.
- [22] A. Chowdhury and D. De, "RGSO-UAV: Reverse glowworm swarm optimization inspired UAV path-planning in a 3D dynamic environment," *Ad Hoc Netw.*, vol. 140, Mar. 2023, Art. no. 103068.
- [23] X. Wang, Z. Liu, and X. Li, "Optimal delivery route planning for a fleet of heterogeneous drones: A rescheduling-based genetic algorithm approach," *Comput. Ind. Eng.*, vol. 179, May 2023, Art. no. 109179.
- [24] Y. Fu, M. Ding, C. Zhou, and H. Hu, "Route planning for unmanned aerial vehicle (UAV) on the sea using hybrid differential evolution and quantum-behaved particle swarm optimization," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 43, no. 6, pp. 1451–1465, Nov. 2013.
- [25] J. Zhang, D. Chen, G. Han, and Y. Qian, "Formation path planning for collaborative autonomous underwater vehicles based on consensus-sparrow search algorithm," *IEEE Internet Things J.*, vol. 11, no. 8, pp. 13810–13823, Apr. 2024.

- [26] W. Wang, Y. Liu, R. Srikant, and L. Ying, "3M-RL: Multi-resolution, multi-agent, mean-field reinforcement learning for autonomous UAV routing," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 8985–8996, Jul. 2022.
- [27] S. Malakar, M. Ghosh, S. Bhowmik, R. Sarkar, and M. Nasipuri, "A GA based hierarchical feature selection approach for handwritten word recognition," *Neural Comput. Appl.*, vol. 32, no. 7, pp. 2533–2552, Apr. 2020.
- [28] C. Yan, X. Xiang, and C. Wang, "Towards real-time path planning through deep reinforcement learning for a UAV in dynamic environments," *J. Intell. Robot. Syst.*, vol. 98, no. 2, pp. 297–309, May 2020.
- [29] C. Yan and X. Xiang, "A path planning algorithm for UAV based on improved Q-learning," in *Proc. 2nd Int. Conf. Robot. Autom. Sci. (ICRAS)*, Wuhan, China, Jun. 2018, pp. 1–5.
- [30] Y. V. Pehlivanoglu and P. Pehlivanoglu, "An enhanced genetic algorithm for path planning of autonomous UAV in target coverage problems," *Appl. Soft Comput.*, vol. 112, Nov. 2021, Art. no. 107796.
- [31] A. Konar, I. Goswami Chakraborty, S. J. Singh, L. C. Jain, and A. K. Nagar, "A deterministic improved Q-learning for path planning of a mobile robot," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 43, no. 5, pp. 1141–1153, Sep. 2013.
- [32] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, Jan. 2016.
- [33] H. Liu, C. Yu, C. Yu, C. Chen, and H. Wu, "A novel axle temperature forecasting method based on decomposition, reinforcement learning optimization and neural network," *Adv. Eng. Informat.*, vol. 44, Apr. 2020, Art. no. 101089.
- [34] Z. Zhang, T. Zhang, J. Hong, H. Zhang, J. Yang, and Q. Jia, "Double deep Q-network guided energy management strategy of a novel electric-hydraulic hybrid electric vehicle," *Energy*, vol. 269, Apr. 2023, Art. no. 126858.
- [35] D. L. Cruz and W. Yu, "Path planning of multi-agent systems in unknown environment with neural kernel smoothing and reinforcement learning," *Neurocomputing*, vol. 233, pp. 34–42, Apr. 2017.
- [36] M. Pouyan, A. Mousavi, S. Golzari, and A. Hatam, "Improving the performance of Q-learning using simultaneous Q-values updating," in *Proc. Int. Congr. Technol., Commun. Knowl. (ICTCK)*, Mashhad, Iran, Nov. 2014, pp. 1–6.
- [37] Y.-H. Wang, T.-H.-S. Li, and C.-J. Lin, "Backward Q-learning: The combination of Sarsa algorithm and Q-learning," *Eng. Appl. Artif. Intell.*, vol. 26, no. 9, pp. 2184–2193, Oct. 2013.
- [38] J. Qiao, Z. Hou, and X. Ruan, "Q-learning based on neural network in learning action selection of mobile robot," in *Proc. IEEE Int. Conf. Autom. Logistics*, Jinan, China, Aug. 2007, pp. 263–267.
- [39] M. Duguleana and G. Mogan, "Neural networks based reinforcement learning for mobile robots obstacle avoidance," *Expert Syst. Appl.*, vol. 62, pp. 104–115, Nov. 2016.
- [40] J. Xia, X. Zhu, Z. Liu, Y. Luo, Z. Wu, and Q. Wu, "Research on collision avoidance algorithm of unmanned surface vehicle based on deep reinforcement learning," *IEEE Sensors J.*, vol. 23, no. 11, pp. 11262–11273, Jun. 2023.
- [41] X.-S. Yang, "Flower pollination algorithm for global optimization," in *Unconventional Computation and Natural Computation (Lecture Notes in Computer Science)*, vol. 7445, J. Durand-Lose and N. Jonoska, Eds., Berlin, Germany: Springer, 2012, pp. 240–249.
- [42] Z. A. Abdalkareem, M. A. Al-Betar, A. Amir, P. Ehkan, A. I. Hammouri, and O. H. Salman, "Discrete flower pollination algorithm for patient admission scheduling problem," *Comput. Biol. Med.*, vol. 141, Feb. 2022, Art. no. 105007.
- [43] K. J. Singh, A. Nayyar, D. S. Kapoor, N. Mittal, S. Mahajan, A. K. Pandit, and M. Masud, "Adaptive flower pollination algorithm-based energy efficient routing protocol for multi-robot systems," *IEEE Access*, vol. 9, pp. 82417–82434, 2021.
- [44] K. Venkatasalam, P. Rajendran, and M. Thangavel, "Improving the accuracy of feature selection in big data mining using accelerated flower pollination (AFP) algorithm," *J. Med. Syst.*, vol. 43, no. 4, p. 96, Apr. 2019.
- [45] Y. Chen, D. Pi, and Y. Xu, "Neighborhood global learning based flower pollination algorithm and its application to unmanned aerial vehicle path planning," *Expert Syst. Appl.*, vol. 170, May 2021, Art. no. 114505.



LAN BO received the Bachelor of Engineering degree from Qingdao University, in 2022, where she is currently pursuing the master's degree. Her major research interests include reinforcement learning, unmanned aerial vehicles, and automation.



TIEZHU ZHANG received the Ph.D. degree in vehicle engineering from Jilin University, in 1991. In January 2019, he was elected as a Foreign Academician with Georgia National Academy of Sciences and Russian Academy of Natural Sciences, in July 2020. Currently, he is a Professor with Qingdao University.



HONGXIN ZHANG received the Ph.D. degree in vehicle engineering from Jilin University, in 2002. Currently, he is a Professor with Qingdao University.



JIAN YANG received the master's degree from Qingdao University, in 2022. He is currently pursuing the Ph.D. degree with the University of Science and Technology Beijing.



ZHEN ZHANG received the master's degree from Qingdao University, in 2024. He has published more than seven SCI articles. His research interests include reinforcement learning and optimization.



CAIHONG ZHANG received the Ph.D. degree from the Ocean University of China. Currently, she is a Teacher with Qingdao University.



MINGJIE LIU received the Bachelor of Engineering degree from Qingdao University, in 2022, where she is currently pursuing the master's degree. Her research interests include artificial potential field method, unmanned aerial vehicles, and automation.

...