Heuristic Q-learning based on experience replay for three-dimensional path planning of the unmanned aerial vehicle

Ronglei Xie, Zhijun Meng, Yaoming Zhou, Yunpeng Ma and Zhe Wu
School of Aeronautic Science and Engineering, Beihang University, Beijing, China

Abstract
In order to solve the problem that the existing reinforcement learning algorithm is difficult to converge due to the excessive state space of the three-dimensional path planning of the unmanned aerial vehicle, this article proposes a reinforcement learning algorithm based on the heuristic function and the maximum average reward value of the experience replay mechanism. The knowledge of track performance is introduced to construct heuristic function to guide the unmanned aerial vehicles’ action selection and reduce the useless exploration. Experience replay mechanism based on maximum average reward increases the utilization rate of excellent samples and the convergence speed of the algorithm. The simulation results show that the proposed three-dimensional path planning algorithm has good learning efficiency, and the convergence speed and training performance are significantly improved.

Keywords
Path planning, unmanned aerial vehicle, Q-learning, heuristic information, experience replay

Introduction
Unmanned aerial vehicle (UAV) has attracted wide attention from scholars all over the world in past decade. UAV has the advantages of small size, low cost, convenient use, low requirements for the operational environment, flexible, no casualty risk, and so on. It is widely used in aerial photography, plant protection, express
transportation, disaster rescue, surveying and mapping, power inspection, reconnaissance, and other fields. Path planning is one of the key technologies for UAVs to accomplish the above tasks. Path planning refers to finding an optimal or feasible trajectory from the starting point to the target point under given space and constraints. UAV path planning is an NP complex problem with multiple constraints, which include fuel consumption and maneuvering constraints, terrain obstacles, threat information, and so on. Many scholars have done a lot of work in path planning. A* algorithms are widely used in path planning due to its ease of implementation and high efficiency, but it is difficult to ensure that an optimal path is found.

Artificial potential field method realizes path planning by establishing the gravitational field and repulsion field function, but it is easy to fall into the local optimal solution. The Voronoi diagram has a good performance in obstacle avoidance, but it needs to describe the obstacle information with geometric structure, which is difficult to adapt to large dynamic environment. Swarm intelligence algorithm (such as ant colony algorithm, particle swarm algorithm) has the advantages of simple structure, high precision, fast convergence, and so on. However, there are still complex calculation problems in high-dimensional and complex planning space.

Reinforcement learning is an important method in the field of machine learning. Unlike supervised learning and unsupervised learning, reinforcement learning relies on the agent to continuously interact with the environment, and each interaction receives an evaluative feedback signal in return for learning the optimal behavior by maximizing its cumulative reward. Theoretically, reinforcement learning does not depend on the exact model. Whatever exploration and utilization strategy is adopted, it will converge to the optimal value after a long enough time. But for a practical problem, when the state space is large and each state can perform more actions, there will be a very large state-action space. In order to accelerate the convergence speed of reinforcement learning algorithm, researchers have proposed several effective methods. Hengst uses hierarchical reinforcement learning to decompose large-scale reinforcement learning problems into several sub-problems, which reduces the state space of the problem. However, reasonable hierarchical processing is a challenging task and is difficult to achieve. Andrew et al. introduced shaping function into reinforcement learning and added heuristic value to the returns of agents, which effectively improved the convergence speed. Asmuth et al. used potential field function as a priori knowledge to enlighten the reinforcement learning process and proved the effectiveness of the algorithm.

Based on the existing reinforcement learning algorithm for path planning, this article proposes a three-dimensional path planning method for UAV based on heuristic return function and maximum average reward experience replay mechanism. First, the state space of UAV is discretized to reduce the scale of path planning problem. Then, a heuristic reward function is constructed based on the maneuverability of UAV, fuel consumption, terrain obstacles, flight altitude, and other factors to improve the convergence speed of the algorithm. The experience replay mechanism is improved, and the importance of samples is evaluated by the maximum average return value at a small computational cost. Finally, the validity of
the method is verified by the three-dimensional path planning simulation experiment of UAV, which achieves the effective approximation of the value function, and has good learning efficiency and generalization performance. The convergence speed and training performance are obviously improved.

**Problem formulation**

The state space of UAV in real environment is continuous, which greatly increases the difficulty of problem-solving. Therefore, three-dimensional discretization of planning space should be carried out first, so the search space of path planning problem is reduced to a discrete set of spatial nodes; each node $v_k$ represents the three-dimensional coordinates $(x_k, y_k, z_k)$ of UAV. The UAV path planning problem can be described as finding several nodes in the space node set, which minimizes the total cost function of UAV flying along the path composed of these nodes. Let the set of all discrete state space nodes set $V$, as follows

$$V = \{v_1, v_2, v_3, \ldots, v_n | v_k = (x_k, y_k, z_k)\}$$

The set of all flight paths, including the starting point and the target point, is represented by $L$

$$L = \{L_1, L_2, L_3, \ldots, L_m\}$$

The cost from state space node $v_i$ to state space node $v_j$ can be expressed in $c(v_i, v_j)$, and the UAV path planning problem can be described in mathematical language as follows

$$C(L_k) = \sum_{(v_i, v_j) \in L_k} c(v_i, v_j)$$

s.t. $L_k \in L, v_i, v_j \in V$

where $C(L_k)$ represents the total cost of the available path $L_k$.

**Q-learning**

**Algorithm principle**

The research of reinforcement learning is based on the theoretical framework of Markov decision process (MDP). The state and reward of the next moment only depend on the current state. The MDP can be represented by a quaternion $(S, A, p_{sa}, R)$:

1. $S$ is the set of states, and any state $s \in S$, $s_t$ represents the state of step $t$ of UAV, the state $s_t$ only includes three-dimensional position coordinates $(x_t, y_t, z_t)$. 

---

Xie et al. 3
2. $A$ is the set of actions, and any action $a \in A$, and $a_t$ represents the actions that UAV executes in step $t$. The action of UAV can be composed of horizontal and vertical actions. The horizontal actions include left turning 45°, direct flying, and right turning 45°. The vertical actions include climbing 45°, horizontal flying, and descending 45°. Therefore, through different combinations of horizontal and vertical actions, UAV’s action space includes nine kinds of actions.

3. $p_{sa}$ represents the probability distribution of state transition of UAV after executing an action $a_t$ in state $s_t$, and the probability of transition to state $s_{t+1}$ is $p_{sd}(s_{t+1}|s_t, a_t)$. For UAV three-dimensional path planning, the probability of UAV moving to other states after executing its action $a_t$ in the state $s_t$ is determined $p_{sa}(s_{t+1}|s_t, a_t) = 1$.

4. $R$ is the reward function, which represents the immediate reward that UAV receives when it executes an action $a_t$ in a state $s_t$ and transfers to a state $s_{t+1}$. It can also be expressed as $R(s_t, a_t, s_{t+1}) = r_t$.

The model of reinforcement learning can be described as UAV–environment interaction in Figure 1. The UAV chooses an action $a_t$ in the state $s_t$. It then transfers to the next state $s_{t+1}$ according to the probability $p_{sa}$ of state transition and gets a reward value. Then, the UAV executes another action $a_{t+1}$ according to state $s_{t+1}$ and repeats the above process.

The goal of reinforcement learning is to learn a good strategy to maximize future cumulative rewards in sequential decision-making. The sum of discount cumulative rewards is called expected reward $G(t)$

$$G(t) = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \gamma^3 r_{t+4} + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \quad (2)$$

where $\gamma$ represents the discount factor, the range of values is between 0 and 1. The closer to 0, the more important the present reward is, and the closer to 1, the more important the future reward is.
The state-action value function $Q^\pi(s, a)$ represents the mathematic expectation of reward that UAV can obtain after choosing the action $a$ in the state $s$ and executing the strategy $\pi$ all the time. It can be written as follows

$$Q^\pi(s, a) = E_\pi \left[ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s, a_t = a \right]$$

(3)

Q-learning algorithm is a model-free offline strategy reinforcement learning control algorithm, which has good convergence in discrete MDP. The state-action value function $Q(s_t, a_t)$ is updated according to Bellman equation

$$Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \alpha \left( r_t + \max_{a' \in A} Q(s_{t+1}, a') - Q_t(s_t, a_t) \right)$$

(4)

where $\alpha$ is the learning rate, $0 < \alpha < 1$.

**Training strategy**

Because of the huge state space of UAV’s three-dimensional path planning, it is difficult to record all state-action value functions in tabular form. At present, the common method is to approximate the optimal action-value function with neural network and adjust the weight parameters of neural network by reducing the mean square error of Bellman equation. However, the convergence of this method is poor. Experiential playback and target separation can accelerate the convergence speed of the algorithm.

Experience replay mechanism is proposed by Lin. A fixed-length cached sample data set $D_t = \{d_1, d_2, d_3, \ldots, d_t\}$, $d_i = \{s_t, a_t, r_t, s_{t+1}\}$ is set up. With the increase in data samples, the old data in the data set are updated with new data. Episode $e_{ps_k}$ refers to the process of UAV from the starting point to the collision or arrival at the target point. Every $n$ episode, $m$ samples are randomly selected from the cached sample data set and put into Q-learning algorithm. Experience replay mechanism has two main advantages:

1. Each sample $d_t$ may be used many times, which improves the efficiency of data utilization.
2. It breaks the correlation of continuous samples in traditional Q-learning algorithm and helps to prevent the algorithm from falling into local optimum.

Minh et al. proposed to use two neural networks for learning, one is target neural network predicting $Q(s', a; \theta_{\text{target}}^j)$, and the other is estimation neural network predicting $Q(s, a; \theta_{\text{est}}^i)$. Bellman equation can be expressed as follows

$$l_{ai} = Q(s, a; \theta_{\text{est}}^i) + \alpha \left( r_t + \max_{a' \in A} Q(s', a; \theta_{\text{target}}^j) - Q(s, a; \theta_{\text{est}}^i) \right)$$

(5)
When the estimated neural network is trained, the loss function of the neural network can be written as follows

$$J(\theta_i^{\text{val}}) = \frac{1}{2} (\text{lab}_i - Q(s, a; \theta_i^{\text{est}}))^2$$ \hspace{1cm} (6)

The estimated neural network is continuously updated. When it is updated to a certain number of times $o$, the parameters of the estimated neural network are copied to the target neural network. Target neural network delays the influence of parameter updating, eliminates the correlation between the estimated value and the target value, and reduces the possibility of non-convergence of the algorithm.

**Action selection strategy**

In order to ensure good convergence, action selection strategies need to consider both exploration and application. The commonly used action selection strategy is $\varepsilon$-greedy, which uses the probability of $1 - \varepsilon$ to select the maximum action in the state and the probability of $\varepsilon$ to randomly select any action in the available action space

$$\pi(s) = \begin{cases} \text{random choice} & \varepsilon \\ \arg \max_{a \in A} Q(s, a) & 1 - \varepsilon \end{cases} \hspace{1cm} (7)$$

---

**Algorithm 1: Q-learning**

**Input:** discount factor $\gamma$, learning rate $\alpha$, exploring factor $\varepsilon$

Neural network training parameters $n, o, m, u$, number of training times $N$, starting point $s_0$, target point $s_{\text{target}}$

Initialize: $Q(s, a; \theta_0^{\text{est}}), Q(s, a; \theta_0^{\text{target}}), (\forall s \in S, a \in A)$

For $k = 1:N$ do

For $t = 1, 2, 3, \ldots$ do

Choose $a_t$ from $A$ using equation (7)

Take action $a_t$ that leads to new state $s_{t+1}$

Calculate return value $r_t$ by

$$r_t = \begin{cases} 1 & \text{beyond or collision} \\ -1 & \text{arrive the goal} \\ 0 & \text{otherwise} \end{cases}$$

$\varepsilon_{sk} \leftarrow [s_t, a_t, r_t, s_{t+1}]$

$s_{t+1} \leftarrow s_{t+1}$

Until $r_t \neq 0$

If $\text{rem}(k, n) = 0$

Choose $m$ samples to update $Q(s, a; \theta_i^{\text{est}})$ using equation (6)

If $\text{rem}(k, o) = 0$

$\theta_i^{\text{target}} \leftarrow \theta_i^{\text{est}}$
Heuristic Q-learning

In order to improve the convergence speed of Q-learning, this article proposes a heuristic Q-learning method by improving the reward function and action selection strategy.

Reward function

Traditional Q-learning mostly uses sparse reward function. For path planning problem, if UAV reaches the target point, it will give a positive reward value, and if it collides with obstacles or exceeds the state space range, it will generate a negative penalty value. In other cases, the return value is generally 0. Sparse reward function has the advantages of simple calculation and easy design. But for UAV three-dimensional path planning problem, the state space is very large, the reward value of most actions is 0, and the probability of finding meaningful reward value is relatively small which leads to the slow convergence speed of learning algorithm. In order to overcome the shortcomings of sparse return function, a heuristic reward function $R(s_t, a_t, s_{t+1})$ is constructed by introducing more factors affecting path performance, such as UAV maneuverability, fuel consumption, terrain threat, distance to target, and flight altitude

$$ r_t = R(s_t, a_t, s_{t+1}) = \omega_1 p_1 + \omega_2 p_2 + \omega_3 p_3 + \omega_4 p_4 + \omega_5 p_5 $$

where $\omega_1 \sim \omega_5$ is the influence weight of each factor, and $p_1 \sim p_5$ is the evaluation factor of path performance. The concrete connotation and calculation method are as follows:

1. $p_1$ represents the fuel consumption cost of the UAV from state $s_t$ to state $s_{t+1}$. The fuel cost is to ensure that the UAV reaches its destination in as short a time as possible

$$ p_1 = -\sqrt{(x_t-x_{t+1})^2 + (y_t-y_{t+1})^2 + (z_t-z_{t+1})^2} $$

2. In the artificial potential field method, the target point will attract the UAV. According to this idea, this article constructs $p_2$. The closer to the target point, the more returns will be obtained, which will provide direction guidance for the choice of UAV operation

$$ em = \sqrt{x_{target}^2 + y_{target}^2 + z_{target}^2} - \sqrt{(x_t-x_{target})^2 + (y_t-y_{target})^2 + (z_t-z_{target})^2} $$

$$ p_2 = \log_{10}(em + \delta) $$

where $(x_{target}, y_{target}, z_{target})$ represents the three-dimensional coordinates of the target point, $\delta$ is a constant greater than 0, which prevents calculation errors when $em$ is equal to 0.
3. $p_3$ is the cost of UAV deviating from cruise altitude $z_0$, which is designed to ensure UAV flying at cruise altitude as far as possible

$$p_3 = -|z_t - z_0|$$  \hspace{1cm} (12)

4. $p_4$ is the cost of collision between UAV and obstacle or beyond the scope of state space

$$p_4 = \begin{cases} R_0 & \text{beyond or collision} \\ 0 & \text{otherwise} \end{cases}$$  \hspace{1cm} (13)

where $R_0$ is a large negative constant.

5. $p_5$ is the reward value of the UAV when it reaches the target point. A larger reward value can make the algorithm converge quickly

$$p_5 = \begin{cases} R_1 & \text{arrive the goal} \\ 0 & \text{otherwise} \end{cases}$$  \hspace{1cm} (14)

where $R_1$ is a large positive constant.

**Action selection strategy**

The key point of Q-learning algorithm convergence is to explore and utilize the balance problem and choose the appropriate action selection strategy which can not only make UAV explore the environment fully to avoid falling into local optimal solution but also make the learning algorithm converge faster. The optimal strategy of Q-learning is usually based on the estimation of state-action value function $Q(s,a)$. However, in the initial stage of algorithm learning, UAV cannot get enough information about the environment, and the optimal estimation of state-action value function is not accurate, which often makes UAV collide with obstacles, seriously affecting the learning efficiency of the algorithm.

Therefore, this article designs an evaluation function $E(s,a)$ to assist the state-action value function $Q(s,a)$ to select the action. The meaning of $E(s,a)$ is that the UAV performs nine actions in the action space separately in the state $s$. When the UAV collides with the obstacle after the action, it returns to $-100$, and if there is no collision, it returns to 0.

$$E(s,a) = \begin{cases} -100 & \text{collision} \\ 0 & \text{otherwise} \end{cases}$$  \hspace{1cm} (15)

If $E(s,a_i)$ is $-100$, then $a_i$ is removed from the action space in the state $s$. The evaluation function $E(s,a)$ can reduce the available action space of UAV in the state $s$ and avoid a large number of collisions in the initial stage. The action selection strategy of heuristic Q-learning is $\epsilon$-greedy, which chooses the action $a$ when $(Q(s,a) - E(s,a))$ takes the maximum in the state $s$ with probability $1 - \epsilon$ and randomly chooses any action in the available action space with probability $\epsilon$. 


\[
\pi(s) = \begin{cases} \text{random choice } a \\
\arg \max_{a \in A} (Q(s, a) - E(s, a)) \\
1 - \varepsilon
\end{cases}
\]

(16)

In the early stage of learning, UAV is ignorant of environmental information, so action selection strategies should pay more attention to exploration. With the deepening of learning, the reliability of state-action value function is getting higher, and action selection strategies should be paid more attention to and utilized. So \(\varepsilon\) is not a fixed value, but a relatively big value at the beginning, and then gradually reduces to nearly 0 with the increase of learning times

\[
\varepsilon = \varepsilon - \varepsilon_{\text{increment}} (\varepsilon_{\text{min}} < \varepsilon < \varepsilon_0)
\]

(17)

where \(\varepsilon_0\) is the initial value of exploratory factor, \(\varepsilon_{\text{min}}\) is the minimum value of exploratory factor, \(\varepsilon_{\text{increment}}\) is the increment of exploratory factor.

---

**Algorithm 2: Heuristic Q-learning**

Input: discount factor \(\gamma\), learning rate \(\alpha\), exploring factor \(\varepsilon_0\), \(\varepsilon_{\text{min}}\), \(\varepsilon_{\text{increment}}\)

Neural network training parameters \(m, n, o, u\), number of training times \(N\), reward parameters \(\omega_1 - \omega_5, R_0, R_1, \delta, z_0\), starting point \(s_0\), target point \(s_\text{target}\)

Initialize: \(Q(s, \alpha; \theta^\text{init}_0), Q(s, \alpha; \theta^\text{init}_\text{target})\), \((\forall s \in S, \alpha \in A)\)

For \(k = 1 : N\) do
- Initial state \(s_0\)
- For \(\tau = 1, 2, 3, \ldots\) do
  - Choose \(\alpha_t\) using equation (16)
  - Take action \(\alpha_t\) that leads to new state \(s_{t+1}\)
  - Calculate return value \(r_t\) using equation (8)
  - \(\varepsilon_{s_t} \leftarrow [s_t, \alpha_t, r_t, s_{t+1}]\)
  - \(s_t \leftarrow s_{t+1}\)
  - until \(p_4 + p_5 \neq 0\)
  - If \(\text{rem}(k, n) = 0\)
    - Choose \(m\) samples to update \(Q_k(s, \alpha; \theta^\text{init}_0)\) using equation (6)
  - If \(\text{rem}(k, o) = 0\)
    - \(\theta^\text{target} \leftarrow \theta^\text{init}\)

---

**MARER Q-learning**

The experience replay mechanism has the drawback that the sampling is randomly performed in the buffered sample data set, and the quality of the sample is neglected. The sample that the UAV has reached the target point after thousands of explorations may not have the Q-learning algorithm selected. In order to improve the learning efficiency, Schaul et al.\(^{14}\) proposed the sampling strategy of prioritized replay and sorted the samples according to their importance. However, the algorithm needs to continuously sort the samples in the data set, which greatly increases the computational complexity.
In order to further improve the efficiency of the algorithm and reduce the complexity of the algorithm, this article proposes a preferred cached experience replay mechanism using the maximum average reward value (MARER Q-learning) based on heuristic Q-learning:

Calculate the average of reward values \( r_{k}^{\text{ave}} \) for the episode \( \text{epsk} \), as follows

\[
    r_{k}^{\text{ave}} = E \left( \sum_{r_{i} \in \text{epsk}} r_{i} \right)
\]  

(18)

Then update the maximum average reward value \( r_{\text{max}}^{\text{ave}} \)

\[
    r_{\text{max}}^{\text{ave}} = \begin{cases} 
    r_{k}^{\text{ave}} & \text{if} \ r_{k}^{\text{ave}} > r_{\text{max}}^{\text{ave}} \\
    r_{\text{max}}^{\text{ave}} & \text{otherwise}
\end{cases}
\]  

(19)

If the maximum average reward value is updated, indicating that there is a good sample in this episode, all the samples included in this episode are directly put into the algorithm for learning, and the samples in this episode are copied three times and then put into the cached sample data set, which improves the proportion of excellent samples. Then, algorithm randomly selects \( m \) samples from the cached sample data set every \( n \) episode into the Q-learning algorithm for learning.

Algorithm 3: MARER Q-learning

Input: discount factor \( \gamma \), learning rate \( \alpha \), exploring factor \( e_{0}, e_{\text{min}}, e_{\text{increment}} \)

Neural network training parameters \( m, n, o, u \), number of training times \( N \), reward parameters \( \omega_{l} \sim \omega_{5}, R_{0}, R_{1}, \delta, z_{0} \), starting point \( s_{0} \), target point \( s_{\text{target}} \)

Initialize: \( Q(s, a; \theta_{0}^{\text{est}}) \), \( Q(s, a; \theta_{0}^{\text{target}}) \), \( \forall s \in S, a \in A \), \( r_{\text{ave}}^{\text{max}} \)

For \( k = 1:N \) do

Initial state \( s_{0} \), \( \text{epsk} = [ ] \)

For \( t = 1, 2, 3, \ldots \) do

Choose \( a_{t} \) from \( A \) using equation (16)

Take action \( a_{t} \) that leads to new state \( s_{t+1} \)

Calculate return value \( r_{t} \) using equation (8)

\( \text{epsk} \leftarrow [s_{t}, a_{t}, r_{t}, s_{t+1}] \)

\( s_{t} \leftarrow s_{t+1} \)

Until \( p_{4} + p_{5} \neq 0 \)

Calculate the average reward \( r_{k}^{\text{ave}} \) using equation (18)

If \( r_{k}^{\text{ave}} > r_{\text{ave}}^{\text{max}} \)

Update \( Q(s, a; \theta_{0}^{\text{est}}) \) using equation (19) with sample set \( \text{epsk} \)

Copy \( \text{epsk} \) three times and put it in \( D_{t} : D_{t} \leftarrow \text{epsk} \times 3 \)

Update maximum average reward \( r_{\text{max}}^{\text{ave}} \) using equation (19)

If \( \text{rem}(k, n) = 0 \)

Choose \( m \) samples to update \( Q_{k}(s, a; \theta_{0}^{\text{est}}) \) using equation (6)
Simulation results

Assume that the mission area of the drone is 200 km × 100 km × 250 m, and a representative topographic map is generated according to typical terrain features. The discrete step in the horizontal direction is 5 km, and the discrete step in the height direction is 5 m. The discrete space of the three-dimensional path planning of the machine is 40 × 20 × 50, and there are 40 × 20 × 50 × 9 state-action values.

Figure 2. Three-dimensional topographic map.

Figure 3. The number of episode of successful reach to the target point.
The expression for the terrain model used for the simulation is as follows

\[ Z(X, Y) = \sum_{i=1}^{8} h_i e^{-\left(\frac{X-a_i}{c_i}\right)^2 - \left(\frac{Y-b_i}{c_i}\right)^2} \]  

\[ \text{(20)} \]

\((X, Y)\) is the coordinate of the horizontal direction, \(Z(X, Y)\) is the height of the mountain in this horizontal coordinate. \(h_i\) determines the maximum height of the mountain, \(c_i\) determines the horizontal range of the mountain, and \((a_i, b_i)\) determines the horizontal coordinates of the mountain (Figure 2)

Figure 4. Average reward per episode: (a) heuristic Q-learning and (b) MARER Q-learning.
\[ h = [125, 150, 250, 200, 150, 150, 150] \]
\[ c = [200, 300, 200, 100, 300, 300, 200, 100] \]
\[ a_i = [25, 40, 50, 80, 75, 140, 140, 175] \]
\[ b_i = [25, 80, 45, 70, 11, 60, 15, 30] \]

Simulation experiment parameters are as follows:

1. Reward parameters \( \omega_1 = 0.05, \omega_2 = 0.2, \omega_3 = 0.003, \omega_4 = 0.01, \omega_5 = 0.01, \)
   \( R_0 = -100, R_1 = 100, \) Cruising altitude, \( z_0 = 30, \) bias \( \delta = 0.01. \)
2. Discount factor \( \gamma = 0.9, \) learning rate \( \alpha = 0.01, \) exploring factor \( \varepsilon_0 = 0.7, \)
   \( \varepsilon_{\text{max}} = 0.95, b_i = [25, 80, 45, 70, 11, 60, 15, 30], \varepsilon_{\text{increment}} = 0.00001. \)
3. Neural network training interval \( n = 5, \) neural network replacement interval \( \alpha = 100, \)
   number of training samples per batch \( m = 200, \) size of sample data set \( u = 1000, \) number of training times \( N = 30,000. \)
4. Position coordinates of starting point \( s_0 = (10 \text{ km}, 1 \text{ km}, 30 \text{ m}), \) position
   coordinates of target point \( s_{\text{target}} = (200 \text{ km}, 30 \text{ km}, 30 \text{ m}). \)
5. The maximum number of steps per learning episode is 200.

By statistically reaching the episode of the target point successfully, as shown in
Figures 3 and 4, we can clearly see the difference in convergence speed of the three
reinforcement learning methods. Heuristic Q-learning begins to converge after
about 1700 episodes. MARER Q-learning converges after about 1300 episodes.
However, Q-learning has not reached the target point after 2000 episodes. The fast-
est convergence among the three algorithms is the MARER Q-learning algorithm,
while the convergence of Q-learning is the slowest. Heuristic Q-learning constructs
a reward function by introducing path information such as height and distance, so that the UAV can have a deeper understanding of the environment and actively eliminates adverse actions to reduce the motion search space, thereby accelerating the convergence speed of the algorithm. Based on the heuristic Q-learning, MARER Q-learning evaluates the pros and cons of the sample by the maximum average return value and makes the algorithm select the excellent samples in a targeted manner, thus achieving more excellent results.

The optimal three-dimensional path given by heuristic Q-learning is shown in Figure 5, and the projection of the optimal trajectory (heuristic Q-learning) on the height profile and the horizontal profile is shown in Figure 6. The optimal path given by MARER Q-learning is shown in Figure 7, and the projection of the

Figure 6. Three-dimensional path profile of UAV (heuristic Q-learning): (a) vertical plane and (b) horizontal plane.
optimal path (MARER Q-learning) on the height profile and the horizontal profile is shown in Figure 8. By comparing the vertical plane of Figures 6 and 8, it can be found that the flight height of the latter is lower, and the latter has better obstacle avoidance ability; by comparing the horizontal plane of the two, it can be found that the track of the latter is more straight and has better economic performance. In summary, the MARER Q-learning algorithm is better than the other algorithms in terms of convergence speed and planning results.

It can be seen from the figure that the two reinforcement learning methods can successfully find the lower position of the two mountain joints and pass through them, climb up after encountering obstacles, and the altitude will drop to cruise altitude after overturning obstacles. The path planning of MARER Q-learning is better than that of heuristic Q-learning. The three-dimensional path planning of MARER Q-learning is mostly flat, basically the best path from the starting point to the target point.

**Conclusion**

Traditional Q-learning methods are less efficient. When the scale of state space increases linearly, the complexity of the problem increases exponentially, so the traditional reinforcement learning method is difficult to solve the problem of UAV three-dimensional path planning. This article proposes a Q-learning algorithm based on the heuristic function and the maximum average reward value of the experience replay mechanism. By synthetically considering the constraints of UAV path planning, a heuristic function is constructed to guide the learning behavior of UAV effectively, which can break away from blind exploration to a certain extent and improve the learning efficiency. The improved empirical playback mechanism greatly improves the convergence speed of the algorithm with a small computational cost. The
simulation results show that the three-dimensional trajectory obtained by the proposed method of UAV’s three-dimensional path planning achieves the expectation.

**Declaration of conflicting interests**

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

**Funding**

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work received support from National Natural Science Foundation of China (No 61702023, 61976014) and Fundamental Research Funds for the Central Universities.
ORCID iD
Ronglei Xie https://orcid.org/0000-0001-5762-5093

References

1. Meng B-B and Gao X. UAV path planning based on bidirectional sparse A* search algorithm. In: Proceedings of 2010 international conference on intelligent computation technology and automation, Changsha, China, 11–12 May 2010, pp. 1106–1109. New York: IEEE.
2. Jaradat MAK, Garibeh MH and Feilat EA. Autonomous mobile robot dynamic motion planning using hybrid fuzzy potential field. Soft Comput 2011; 16(1): 153–164.
3. Pehlivanoglu YV. A new vibrational genetic algorithm enhanced with a voronoi diagram for path planning of autonomous UAV. Aerosp Sci Technol 2012; 16(1): 47–55.
4. Vincent R, Mohammed T and Gilles L. Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning. IEEE T Ind Inform 2013; 9(1): 132–141.
5. Sutton RS and Barto AG. Introduction to reinforcement learning. Cambridge, MA: MIT Press, 1998.
6. Hengst B. Discovering hierarchical reinforcement learning. Sydney, NSW, Australia: University of New South Wales, 2003.
7. Andrew YN, Harada D and Russell S. Policy invariance under reward transformations: theory and application to reward shaping. In: Proceedings of the sixteenth international conference on machine learning, Bled, 27–30 June 1999, pp. 278–287. San Francisco, CA: Morgan Kaufmann Publishers.
8. Asmuth J, Littman M and Zinkov R. Potential-based shaping in-model-based reinforcement learning. In: Proceedings of AAAI conference on artificial intelligence, Chicago, IL, 13–17 July 2008.
9. Watkins PD. Q-learning. Mach Learn 1992; 8(3): 279–292.
10. Sutton RS and Barto AG. Reinforcement learning: an introduction, vol. 1. Cambridge, MA: MIT press, 1998.
11. Mnih V, Kavukcuoglu K, Silver D, et al. Playing atari with deep reinforcement learning [EB/OL], https://arxiv.org/abs/1312.5602 (accessed 16 March 2017).
12. Lin LJ. Self-improvement based on reinforcement learning, planning and teaching. In: Proceedings of the eight international workshop in machine learning, Evanston, IL, 1 June 1991, pp. 323–327. San Francisco, CA: Morgan Kaufmann.
13. Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning. Nature 2015; 518(7540): 529–533.
14. Schaul T, Quan J, Antonoglou I, et al. Prioritized experience replay [EB/OL], https://arxiv.org/abs/1511.05952 (3 March 2017).

Author biographies

Ronglei Xie received the BEng in 2015 from Northwestern Polytechnical University. Currently, he is working toward the PhD degree at the School of Aeronautic Science and Engineering, Beihang University of China. His current research interests include UAV intelligent mission planning and machine learning.
Zhijun Meng received the BEng and the PhD degree at the School of Aeronautic Science and Engineering, Beihang University of China. He is presently Associate Professor in the department. His current research interests include artificial intelligence, UAV design, and control.

Yaoming Zhou received the BEng and the PhD degree, both from the School of Aeronautic Science and Engineering at Beihang University, China. He is currently an Assistant Professor in the School of Aeronautic Science and Engineering at Beihang University, where his research interests include artificial intelligence, applications of unmanned aerial vehicles (UAVs), aircraft design, intelligent control of UAVs, and unmanned rotorcraft.

Yunpeng Ma is currently a Lecturer in the School of Aeronautic Science and Engineering at Beihang University. His current research interest is design of aircraft.

Zhe Wu received the PhD degree in 1988 at Harbin Institute of Technology, China. He is currently Professor in the School of Aeronautic Science and Engineering at Beihang University. His current research interests include aircraft design and aircraft stealth design.