

Multi-UAV Path Planning in 3D based on APF-MADDPG

Ming Bai

yadxbm@yau.edu.cn

Yan'an University

Yan Chen

Yan'an University

Yi Liu

Yan'an University

Research Article

Keywords: deep reinforcement learning, 3D path planning, dense reward, markov game, avoiding obstacle

Posted Date: August 27th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-6267765/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Multi-UAV Path Planning in 3D based on APF-MADDPG

Bai Ming ^{1*}, Chen Yan ^{2†} and Liu Yi ^{3†}

¹School of Mathematics and Computer Science, Yan'an University, Holy Land Road 580, Yan'an, 716000, ShaanXi Province, China.

^{2*}School of Mathematics and Computer Science, Yan'an University, Holy Land Road 580, Yan'an, 716000, ShaanXi Province, China.

³School of Mathematics and Computer Science, Yan'an University, Holy Land Road 580, Yan'an, 716000, ShaanXi Province, China.

*Corresponding author(s). E-mail(s): yadxbm@yau.edu.cn;

Contributing authors: chenyanyyds@163.com; ly@yau.edu.cn;

[†]These authors contributed equally to this work.

Abstract

In the field of UAV research, multi-unmanned aerial vehicle (multi-UAV) path planning is a particularly challenging issue because of its high-dimensional action space, collision avoidance constraints, and NP-hard surroundings. Planning a multi-UAV becomes extremely challenging, especially when there are complicated obstacles in an unknown environment. Therefore, an improved dense reward function based on the artificial potential field (APF) method is developed in this article to accelerate the convergence speed, based on the deep reinforcement learning algorithm for unknown obstacle environments, the multi-agent deep deterministic policy gradient (MADDPG). In addition, to explore an optimal path planning strategy, a hierarchical deep reinforcement learning training mechanism including a path planning layer and an action execution layer has been established. The simulation results show that the proposed method outperforms the independent learner deep deterministic policy gradient (ILDDPG) and MADDPG in terms of both the optimality of the path planning strategy and the speed at which the strategies converge.

Keywords: deep reinforcement learning, 3D path planning, dense reward, markov game, avoiding obstacle

1 Introduction

multi-unmanned aerial vehicle (multi-UAV) are increasingly being employed in military, logistics, environmental monitoring, and other fields as a result of the rapid development of UAV technology [1, 2]. Multi-UAV provide greater flexibility and mission execution efficiency than single UAVs, yet they also have to deal with challenging path planning issues. Multi-UAV path planning has to consider collaboration and collision avoidance between UAVs, as well as the effective completion of the overall mission, while single-UAV path planning focuses on the optimal path for itself [3–5].

Among the traditional path planning methods, early geometric methods, such as the voronoi diagram method and the visibility diagram method, are effective in simple environments, but the computational effort increases dramatically in complex scenarios, making it difficult to adapt to dynamic changes [6]. The artificial potential field (APF) method is another commonly used method. It guides the UAV to avoid obstacles and reach the target by simulating the electric field force to produce attractive and repulsive forces on the UAV [7]. However, the APF method suffers from the problem of local minima, which may result in the UAV getting stuck at the target location and failing to reach it [8]. Heuristic algorithms such as genetic algorithm and particle swarm optimization (PSO) algorithms have also been developed. Although these algorithms can solve the path planning problem to some extent, genetic algorithms are prone to local optimization and slow convergence, and particle swarm algorithms have limited effectiveness in complex high-dimensional problems [9, 10].

In recent years, reinforcement learning algorithms have been widely concerned in the field of UAV path planning. These algorithms enable the agent to perform adaptive learning through interaction with the environment and adjust strategies to achieve optimal path planning [11]. As an effective reinforcement learning method, the deep deterministic policy gradient (DDPG) algorithm has been successfully applied to the path planning of a single UAV, such as solving the path planning problem of a glider in time-varying ocean currents. However, due to the lack of consideration for strategic differences between individual UAV, collisions between UAVs may occur. The multi-agent deep deterministic policy gradient (MADDPG) algorithm provides a new solution for multi-UAV collaborative path planning. Through the coordinated learning among multiple agents, MADDPG can effectively deal with the collision problem between UAVs [12]. Existing research has made significant contributions to the field of multi-UAV path planning. However, the existing research on multi-UAV path planning still faces challenges such as sparse reward and difficult reward function setting [13].

To address these issues, this article proposes an improved algorithm, multi-agent deep deterministic policy gradient algorithm based on artificial potential field (APF-MADDPG), which combines the advantages of MADDPG and APF. The main contributions of this article are:

- 1) Dense reward function: we propose a dense reward function based on APF and integrate it into the MADDPG framework. This solves the problems of sparse rewards and slow convergence in traditional MADDPG.

2) Hierarchical training mechanism: we introduce a hierarchical deep reinforcement learning training mechanism, including a path planning layer and an action execution layer. This mechanism helps to train the optimal path planning strategy more efficiently.

The rest of this paper is organized as follows: Problem formulation and modeling are introduced in Section 2. The multi-UAV three dimensional (3D) path planning problem is formulated as the partially observable Markov game in Section 3. In Section 4, the APF-MADDPG algorithm for multi-UAV 3D path planning is proposed. The simulation experiments are conducted in Section 5. Finally, this work is summarized in Section 6.

2 Problem Formulation and Modeling

This section formulates the 3D path planning problem for multi-UAV in unknown environments. It is expected that multi-UAVs are dispersed throughout space and can safely reach to the target area to assemble while avoiding the entirety of the threat zone, no-fly zone, and collision obstacles as possible. Subsequently, a 3D UAV motion model and an obstacle model are developed.

2.1 Multi-UAV Aggregation Mission Model

Finding the best or most practical path for each UAV from its initial position to a target area inside the planning space is the aim of UAV path planning in the setting of aggregation missions. The UAVs must typically reach the target area as fast as feasible and arrive at the target area simultaneously, as shown in Fig. 1.

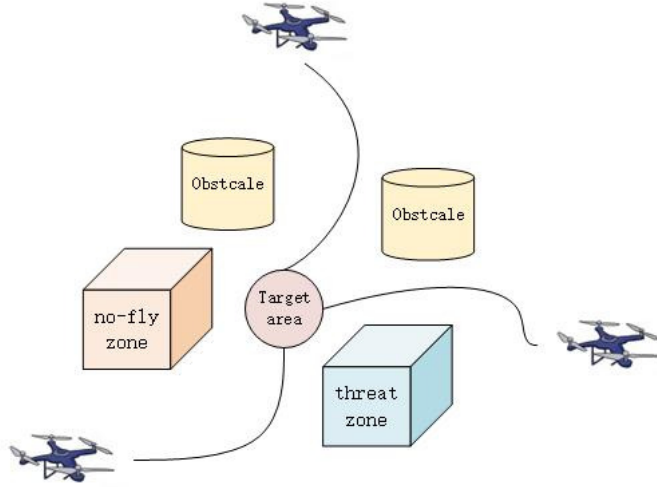


Fig. 1 Multi-UAV path planning under aggregation mission.

For simplification, there are a couple of assumptions as follows:

1) The size of the UAV and the aerodynamic interaction among UAVs are ignored. The UAV is equipped with directional detection radar, which has a certain range of reconnaissance capabilities.

2) Considering the simplified representation of the problem, the obstacle, no-fly zone and threat zone are uniformly represented as obstacle zones.

Moreover, the UAVs are equipped with measurement devices such as the global positioning system (GPS) and inertial measurement unit (IMU) to obtain their real-time position and velocity. For each UAV, the situational information of other UAVs and obstacle zone are known.

2.2 UAV Motion Model

In this work, it is assumed that path planning for multi-UAV occur in 3D space, and the motion model of UAV i can be

$$\begin{cases} x_i(t+1) = x_i(t) + v_i(t) \cdot \cos(\alpha_i(t)) \cdot \cos(\beta_i(t)) \cdot \Delta(t) \\ y_i(t+1) = y_i(t) + v_i(t) \cdot \cos(\alpha_i(t)) \cdot \sin(\beta_i(t)) \cdot \Delta(t) \\ z_i(t+1) = z_i(t) + v_i(t) \cdot \sin(\alpha_i(t)) \cdot \Delta(t) \\ v_i(t+1) = v_i(t) + a_i(t) \cdot \Delta(t) \\ \alpha_i(t+1) = \alpha_i(t) + \omega_{\alpha_i}(t) \cdot \Delta(t) \\ \beta_i(t+1) = \beta_i(t) + \omega_{\beta_i}(t) \cdot \Delta(t) \end{cases} \quad (1)$$

where $(x_i(t), y_i(t), z_i(t))$ refers to three-dimensional coordinates for UAV i at time t , and $\alpha_i(t), \beta_i(t)$ are the pitch angle and yaw angle for UAV i ; $\Delta(t)$ is the time interval; $a_i(t), \omega_{\alpha_i}(t)$ and $\omega_{\beta_i}(t)$ are the linear acceleration, pitch angular acceleration, and yaw angular acceleration, respectively, for UAV i , and $v_i(t)$ is the flying velocity for UAV i .

The motion for UAV i is constrained by

$$\begin{cases} 0 \leq v_i \leq v_i^{max} \\ -a_i^{max} \leq a_i \leq a_i^{max} \\ -\omega_{\alpha_i}^{max} \leq \omega_{\alpha_i} \leq \omega_{\alpha_i}^{max} \\ -\omega_{\beta_i}^{max} \leq \omega_{\beta_i} \leq \omega_{\beta_i}^{max} \\ 0 \leq \alpha_i(t) \leq 2\pi \\ 0 \leq \beta_i(t) \leq 2\pi \end{cases} \quad (2)$$

where $v_i^{max}, a_i^{max}, \omega_{\alpha_i}^{max}, \omega_{\beta_i}^{max}$ are the upper limits of the velocity, linear acceleration, pitch angular acceleration, and yaw angular acceleration for UAV i .

2.3 Obstacles Model

In this work, the obstacles are assumed to be a convex body, which is given by

$$V(x, y, z) = \left(\frac{x - x_0}{l}\right)^{2e} + \left(\frac{y - y_0}{b}\right)^{2f} + \left(\frac{z - z_0}{c}\right)^{2g} \quad (3)$$

where l, b, c determine the coverage ranges of obstacles, and $l, b, c > 0$; e, f, g determine the shapes of obstacles, and $e, f, g > 0$; $[x_0, y_0, z_0]^T$ represents the center of

the obstacle. If $l = b = c$ and $e = f = g$, then obstacles are spheres. If $l = b$ and $e = f = 1, g > 1$, then obstacles are cylinders.

3 Partially Observable Markov Game Modeling

The multi-UAV path planning problem can be regarded as a partially observable Markov game, where each UAV is an agent. The Markov game of N UAVs is defined as a tuple $(N, S_1, \dots, s_N, A_1, \dots, A_N, o_1, \dots, o_N, R_1, \dots, R_N, T, \gamma)$. N is the number of UAVs in the environment. S_i , A_i and R_i represents the set of states, action and joint rewards of UAVs i . o_i represent the local observation space of UAV i . T is the state transition probability function, γ is the discount coefficient.

3.1 State Space and Action Space

The state space of UAV i is defined as

$$S_i = [p_i, v_i, \alpha_i, \beta_i] \quad (4)$$

where $p_i = (x_i, y_i, z_i)$ are the 3D coordinates of UAV i ; v_i is the velocity of UAV i ; α_i , β_i are the pitch angle and yaw angle of UAV i respectively.

The action space for UAVs is as follows:

$$A_i = [a_i, \omega_{\alpha_i}, \omega_{\beta_i}] \quad (5)$$

where a_i , ω_{α_i} and ω_{β_i} are the linear acceleration, pitch angular acceleration, and yaw angular acceleration for UAV i respectively.

3.2 Observation Space

The local observation space o_i of UAV i displays in Table 1, including the information of its own, other UAVs $\{UAV_i^{other}\}_{i=1}^{N-1}$ and obstacle $\{obstacle_j\}_{j=1}^m$, where $N - 1$ and m represent the number of other UAVs and obstacles, respectively. The information of the UAV itself includes its position, velocity, pitch angle and yaw angle. The observation information of other UAVs includes relative position, relative velocity, relative pitch angle, and relative yaw angle. The observation information of obstacles includes relative position.

3.3 Dense Reward Based on Artificial Potential Field

In the process of reinforcement learning, the design of the reward function is crucial. Reward guides the agent toward the target by serving as a reinforcement signal. Reinforcement learning is an algorithm that is very dependent on the interaction environment. The reinforcement signal is typically quite sparse in traditional reinforcement learning, which results in slow convergence during training and a lack of continuing rewards to guide the UAV [14]. A continuous reward model is proposed based on the attraction and repulsion of the APF.

According to the attraction mechanism, the dense reward for each time step between the UAV and the target is expressed as

Table 1 Observation space of UAV

Information	Description
Own	position (x_i, y_i, z_i) velocity v_i pitch angle α_i yaw angle β_i
Other UAVs	UAV 1: relative position, relative velocity, relative pitch angle, relative yaw angle UAV 2: relative position, relative velocity, relative pitch angle, and relative yaw angle ...
Obstacle	UAV N: relative position, relative velocity, relative pitch angle, and relative yaw angle obstacle 1: relative position obstacle 2: relative position ... obstacle m: relative position

$$R_{ut} = \begin{cases} C_1, d(p_{uav}, p_{target}) = 0 \\ C_1 \xi e^{-d(p_{uav}, p_{target})}, otherwise \end{cases} \quad (6)$$

where $p_{target} = (x_{target}, y_{target}, z_{target})$; $d(p_{uav}, p_{target})$ is the Euclidean distance between UAV and target; $\xi \in (0, 1)$ is the attractive strength adjustment coefficient; C_1 is a constant.

According to the repulsion mechanism, the dense reward for each time step between the UAVs is specified as

$$R_{uu} = \begin{cases} 0, d(p_i, p_j) \geq d_0 \\ \eta(d(p_i, p_j) - d_0), 0 < d(p_i, p_j) < d_0 \\ -C_2, d(p_i, p_j) = 0 \end{cases} \quad (7)$$

where $i, j \in (1, N)$; $\eta \in (0, 1)$ is the repulsion strength adjustment coefficient; $d(p_i, p_j)$ is the Euclidean distance between UAVs; d_0 is the safe distance for UAV motion; C_2 is a constant.

According to the repulsion mechanism, the dense reward for each time step between the UAV and the obstacle is defined as

$$R_{uo} = \begin{cases} 0, d(p_{uav}, p_{obstacle}) \geq d_0 \\ -\phi(e^{d_0} - e^{d(p_{uav}, p_{obstacle})}), 0 < d(p_{uav}, p_{obstacle}) < d_0 \\ -C_3, d(p_{uav}, p_{obstacle}) = 0 \end{cases} \quad (8)$$

where $d(p_{uav}, p_{obstacle})$ is the Euclidean distance between UAV and obstacle; $p_{obstacle} = (x_{obstacle}, y_{obstacle}, z_{obstacle})$; $\phi \in (0, 1)$ is the repulsion strength adjustment coefficient, and C_3 is a constant.

It is proposed to develop a reward function far away from the target to solve the problem of unreachable target in APF and also to avoid the agent to keep wandering at the target for getting the maximum cumulative reward of the episode.

$$R_{depart} = \delta(d(p_{last}, p_{target}) - d(p_{now}, p_{target})) \quad (9)$$

where $\delta \in (0, 1)$ is the reward strength adjustment coefficient; p_{last}, p_{now} represent the position of UAV at last time and now, respectively; $d(p_{last}, p_{target})$ is the Euclidean

distance between UAV and target at last time, (p_{now}, p_{target}) the Euclidean distance between UAV and target at now.

Combining Eqs. 6, 7, 8 and 9, the sum of dense rewards at each time step is

$$R_{sum} = w_1 R_{ut} + w_2 R_{uu} + w_3 R_{uo} + w_4 R_{depart} \quad (10)$$

where $w_i (i = 1, 2, 3, 4)$ is the weight coefficient of the corresponding dense reward, and $\sum_{i=1}^4 w_i = 1$.

4 APF-MADDPG Algorithm

In this work, APF-MADDPG is used as a solution to the multi-UAV 3D path planning problem for the following reasons:

1) MADDPG adopts a distributed execution and centralized training framework. The full environmental state is obtainable during centralized training, which helps in addressing non-stationary problems in multi-UAV settings.

2) Each UAV has a more flexible independent actor-critic structure because of the exploitation of MADDPG. The critic network as a value function for evaluating decisions for policy optimization, while the actor network as a decision-making method based on local observations.

3) MADDPG is based on deterministic strategies, which are more efficient than stochastic strategies and can handle continuous action spaces [12].

4) APF can provide dense reward signals for MADDPG. In traditional reinforcement learning path planning, the reward signals are often sparse, resulting in a slow convergence speed [13]. However, APF generates continuous rewards in real time based on the distances between the UAV and the target and obstacles. This helps to accelerate the training convergence process of MADDPG and enables the UAV to learn effective path planning strategies more quickly.

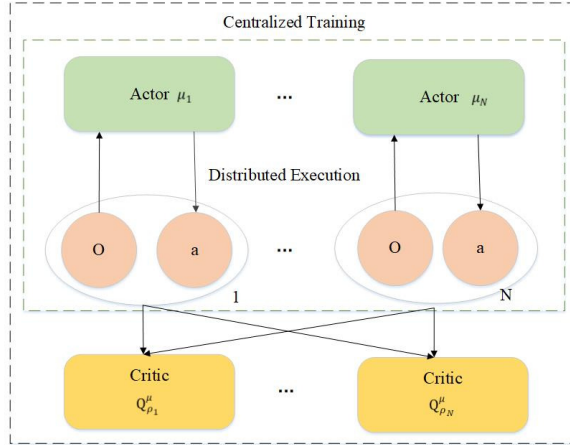


Fig. 2 Framework of fundamental MADDPG.

As the environment is partially observable, it is challenging for the UAVs to obtain a complete and accurate environmental state space[15]. Therefore, based on the framework in Fig. 2, a hierarchical APF-MADDPG training mechanism for multi-UAV 3D path planning is presented in Fig. 3.

In the path planning layer, each UAV has its own actor network and critic network. Based on the observation $\mathbf{o}$ collected from the environment, the actor network takes action $\mathbf{a}$ and adds exploration noise to UAVs in the path planning layer. UAVs perform actions $\mathbf{a}$, and a new observation $\mathbf{o}'$ is obtained by the distributed execution layer. Furthermore, the reward $\mathbf{r}$ is calculated by the dense reward based on the APF in Section 3. In the experience replay buffer, the experience data $(\mathbf{o}, \mathbf{a}, \mathbf{r}, \mathbf{o}')$ is synchronously saved. At each time step, mini batches of data are randomly sampled from the experience replay buffer for training actor and critic networks [16].

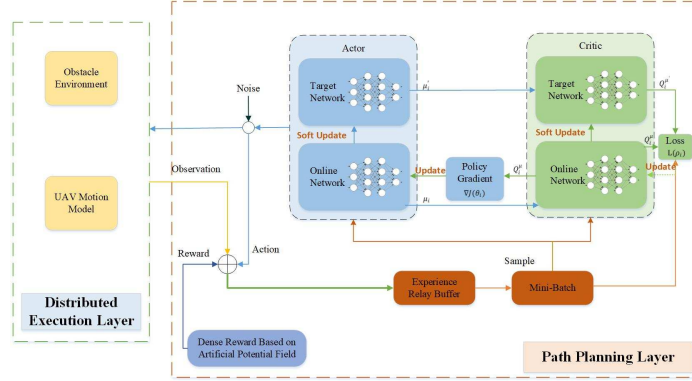


Fig. 3 Hierarchical framework of MADDPG training mechanism for multi-UAV 3D path planning.

In the multi-UAV path planning scenario, it is assumed that N UAVs with local actor networks are $\mu = \{\mu_1, \dots, \mu_N\}$ parameterized by $\theta = \{\theta_1, \dots, \theta_N\}$. The policy gradient for training the actor network is

$$\nabla_{\theta_i} J(\mu_i) = E_{\mathbf{o}, \mathbf{a} \sim D} [\nabla_{\theta_i} \mu_i(\mathbf{a}_i | \mathbf{o}_i) \nabla_{\mathbf{a}_i} Q_i^\mu(\mathbf{o}, \mathbf{a}_1, \dots, \mathbf{a}_N) |_{\mathbf{a}_i = \mu_i(\mathbf{o}_i)}] \quad (11)$$

where $D = (\mathbf{o}, \mathbf{a}, \mathbf{r}, \mathbf{o}')$ is the experience replay buffer, and $\mathbf{o}$ is the set of local observations. $\mathbf{o}_i$ is the local observation of UAV i , $\mathbf{a}$ is the set of action, $\mathbf{r}$ is the set of local rewards, and $\mathbf{o}'$ is the set of local observations at the next time step. $\mathbf{a}_i$ is the local action of UAV i determined by the local observation $\mathbf{o}_i$ according to the policy μ_i , $\mathbf{a}_i = \mu_i(\mathbf{o}_i)$. Q_i^μ is the critic network of UAVs i with parameters ψ_i . Based on the critic network, the Q_i^μ is updated as

$$\begin{cases} L(\psi_i) = E_{\mathbf{o}, \mathbf{a}, \mathbf{r}, \mathbf{o}' \sim D} [(Q_i^\mu(\mathbf{o}, \mathbf{a}_1, \dots, \mathbf{a}_N) - y_i)^2] \\ y_i = r_i + \gamma Q_i^{\mu'}(\mathbf{o}', \mathbf{a}'_1, \dots, \mathbf{a}'_N) |_{\mathbf{a}'_i = \mu'_i(\mathbf{o}'_i)} \end{cases} \quad (12)$$

where r_i is the local reward of UAV i . $Q_i^{\mu'}$ is the target critic network with parameters ψ'_i . γ the discount factor. μ'_i is the target actor network with parameters θ'_i . The target

networks are introduced by the double-network structure for stability, as shown in Fig. 2. The parameters of the target networks are updated through soft update by

$$\begin{cases} \theta'_i = \tau\theta_i + (1 - \tau)\theta'_i \\ \psi'_i = \tau\psi_i + (1 - \tau)\psi'_i \end{cases} \quad (13)$$

where τ is the soft update rate and $\tau \ll 1$.

The pseudocode of the APF-MADDPG algorithm is listed in Algorithm 1.

Algorithm 1 APF-MADDPG Algorithm

```

1: Input: Max_Episode, Max_Step, N
2: Output: Model Parameters of Actor Network  $\theta_i$ 
3: Initialize main and target local actor network  $\mu_{\theta_i}, \mu'_{\theta'_i}$ 
4: Initialize main and target local critic network  $Q_{\psi_i}^\mu, Q_{\psi'_i}^{\mu'}$ 
5: Initialize discount factor  $\gamma$  and soft update rate  $\tau$ 
6: for episode = 1, ..., Max_Episode do
7:   Initialize the environment.
8:   Initialize a random process  $\mathcal{N}$  for action exploration.
9:   Receive initial state  $\mathbf{o}$ .
10:  for t = 1, ..., Max_Step do
11:    UAV  $i$  select action  $a_i = \mu_{\theta}(\mathbf{o}_i) + \mathcal{N}, i = 1, \dots, N$ .
12:    Execute actions  $\mathbf{a} = (\mathbf{a}_1, \dots, \mathbf{a}_N)$  and receive reward  $\mathbf{r}$  and new observation
     $\mathbf{o}'$ .
13:    Store  $(\mathbf{o}, \mathbf{a}, \mathbf{r}, \mathbf{o}')$  into experience replay buffer D.
14:    /* Local actor and network training */
15:    for UAV  $i = 1, \dots, N$  do
16:      K samples  $(\mathbf{o}^j, \mathbf{a}^j, \mathbf{r}^j, \mathbf{o}'^j)$  from D as a mini-batch.
17:      Calculate  $\mathbf{y}_i^j$  by  $\mathbf{r}^j + \gamma Q_{\psi_i}^{\mu'}(\mathbf{o}'^j, \mathbf{a}_1', \dots, \mathbf{a}_N' |_{\mathbf{a}_i' = \mu'_{\theta'_i}(\mathbf{o}'^j)})$ .
18:      Update local critic network by minimizing the loss  $L(\psi) =$ 
       $\frac{1}{K} \sum_{j=1}^K (Q_i^\mu(\mathbf{o}, \mathbf{a}_1^j, \dots, \mathbf{a}_N^j) - \mathbf{y}_i^j)^2$ 
19:      Update actor network:  $\theta_i \leftarrow \theta_i + \nabla_{\theta_i} J(\theta_i)$ .
20:      Update target actor and critic network:  $\theta'_i \leftarrow \tau\theta_i + (1 - \tau)\theta'_i$ , and  $\psi'_i \leftarrow$ 
       $\tau\psi_i + (1 - \tau)\psi'_i$ .
21:    end for
22:    Update global observation  $\mathbf{o} \leftarrow \mathbf{o}'$ .
23:  end for
24: end for

```

5 Simulation Experiments

In this section, the superiority of the APF-MADDPG algorithm in terms of policy learning and path planning is verified through comparison experiments with the MADDPG and ILDDPG algorithms.

5.1 Experimental Settings

The simulation computer configuration is as follows: CPU Intel Core i7-14600F, GPU NVIDIA GeForce RTX 4090, RAM 24GB. The hyperparameters of the MADDPG, ILDDPG, and APF-ADDPG algorithms are listed in Table 2.

Table 2 Hyperparameters of MADDPG, ILDDPG and APF - MADDPG

Parameter	Value
Max_Episode	1200
Max_Step	250
Experience replay buffer size	1000000
Batch Size	128
Discount factor	0.96
Soft update rate	0.01
Learning rate of actor network	0.001
Learning rate of critic network	0.001
Optimizer	Adam

5.2 Experiment on Algorithm Performance Comparison

It is assumed that the three UAVs are distributed in a 10 km * 10 km * 10 km region to evaluate the policy learning and path planning of MADDPG, ILDDPG, and APF-MADDPG. Table 3 displays the initial and target locations of the UAVs within the region. The three UAVs are spread out in different locations in the region to eventually reach the same target location.

Table 3 Settings of the three-UAV

UAV	Initial position (km)	Target position (km)
UAV_1	(1,8,2)	(7,4,1)
UAV_2	(10,10,2)	(7,4,1)
UAV_3	(1,1,1)	(7,4,1)

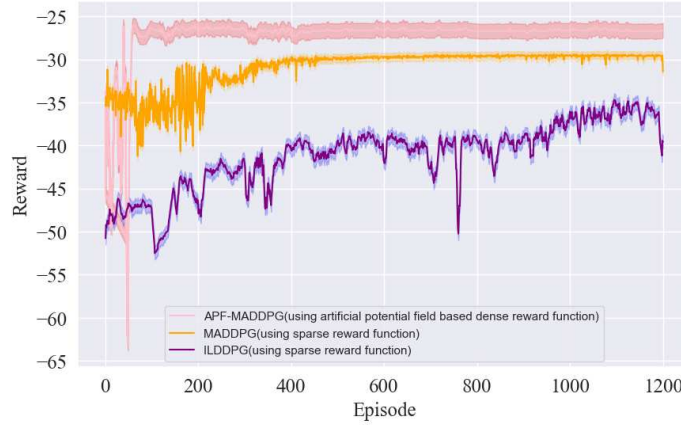
Table 4 displays the obstacle settings constructed in accordance with Eq. 3. The obstacles selected for this experiment were two spherical obstacles and one cylindrical obstacle, and the table lists the center of the sphere of the spherical obstacle, the center of the base of the cylindrical obstacle, and their radiuses, where the height of the cylindrical obstacle was set to 5 km.

The state space and action space of the MADDPG, ILDDPG, and APF-MADDPG algorithms are identical. The stability, cumulative rewards after convergence, and learning speed of the training process can be evaluated by plotting the episode reward curves in Fig. 4.

Table 4 Settings of the obstacles

Obstacle	Center position (km)	Radius(km)
sphere ₁	(2,6,1)	2
cylinder ₁	(4,3,0)	1
sphere ₂	(8,7,1)	2

The same traditional sparse reward function is used to train both MADDPG and ILDDPG. Importantly, in the same scenario, MADDPG has a higher reward after convergence, faster convergence, and better policy learning efficiency. Additionally, APF-MADDPG outperforms the original MADDPG algorithm, which utilizes a traditional sparse reward function, as an improved MADDPG algorithm that uses a proposed APF-based dense reward function provided in Eq. 10. Thus, APF-MADDPG with the proposed artificial potential field-based dense reward function outperforms the other two algorithms, with MADDPG coming in second.

**Fig. 4** Episode reward curves of APF-MADDPG, MADDPG and ILDDPG.

Simulation of multi-UAV path planning based on APF-MADDPG, MADDPG and ILDDPG are shown in Figs. 5, 6, and 7 respectively.

Table 5 Path length comparison

Algorithm	UAV	Length(km)	Total Length(km)
MADDPG	UAV_1	7.9806	25.9426
	UAV_2	9.3982	
	UAV_3	8.5638	
APF - MADDPG	UAV_1	8.4838	24.7056
	UAV_2	8.4642	
	UAV_3	7.7576	

Apparently, the path planned by MADDPG and APF-MADDPG is overall acceptable, avoiding obstacles effectively, and reaching the target successfully. However, according to Fig. 7(a), UAV 1 and UAV 2 on the ILDDPG trained path fail to avoid obstacles. Therefore, ILDDPG cannot be applied to the multi-UAV path planning problem in this scenario.

When determining which of MADDPG and APF-MADDPG is superior, the total path length of all UAVs in the region is taken into consideration. As shown in Table 5, the total path length of the MADDPG algorithm is 25.9426 km, while that of the APF-MADDPG algorithm is 24.7056 km. The planning performance of APF-MADDPG is clearly superior to that of MADDPG.

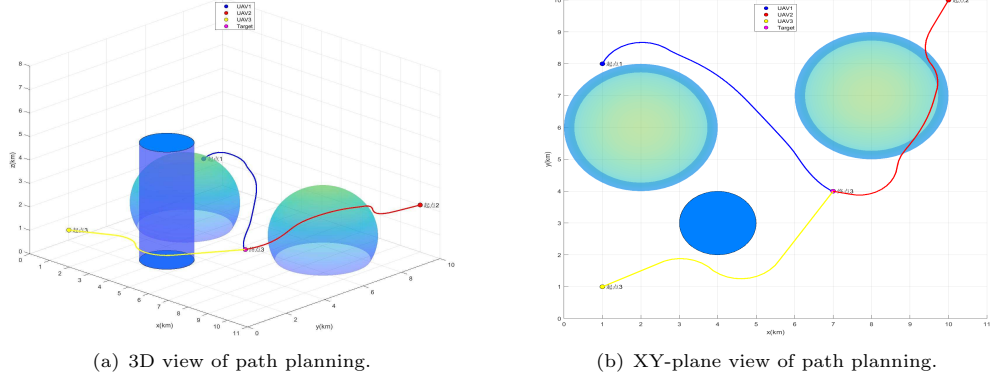


Fig. 5 Simulation of multi-UAV path planning based on APF-MADDPG.

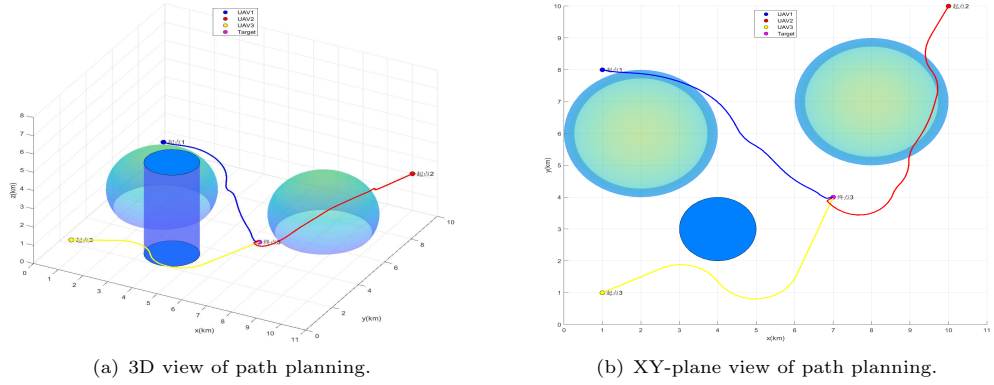


Fig. 6 Simulation of multi-UAV path planning based on MADDPG.

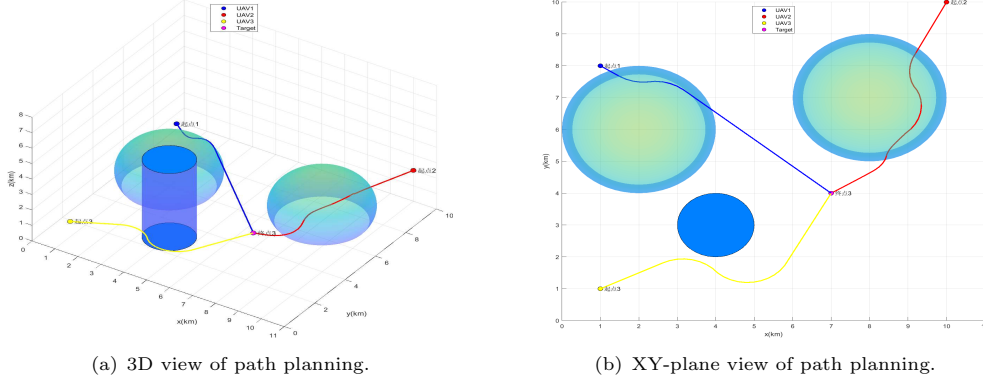


Fig. 7 Simulation of multi-UAV path planning based on ILDDPG.

In conclusion, APF-MADDPG outperforms MADDPG and ILDDPG in this article with respect to policy learning and path planning.

6 Conclusion

This article researches the multi-UAV 3D path planning problem, including path planning and collision avoidance. First, an improved deep reinforcement learning algorithm, APF-MADDPG, is proposed. The new algorithm adopts a dense reward function based on APF, which accelerates the convergence speed. To train the optimal path planning strategy, a hierarchical deep reinforcement learning training mechanism is proposed, which consists of a path planning layer and an action execution layer. Compared with MADDPG and ILDDPG, the proposed APF-MADDPG algorithm has better performance in terms of policy learning, path planning, and convergence speed.

References

- [1] Xiaoning, Z.: Analysis of military application of uav swarm technology. In: 2020 3rd International Conference on Unmanned Systems (ICUS), pp. 1200–1204 (2020). <https://doi.org/10.1109/ICUS50048.2020.9274974>
- [2] Liu, T., Duan, Y., Hao, M.: Research on multi-uav logistics distribution model based on improved particle swarm algorithm. In: 2024 7th International Conference on Advanced Algorithms and Control Engineering (ICAACE), pp. 1281–1285 (2024). <https://doi.org/10.1109/ICAACE61206.2024.10548528>
- [3] Aggarwal, S., Kumar, N.: Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges. *Computer Communications* **149**, 270–299 (2020)

- [4] Sun, G., Gu, H., Lü, J.: Distributed swarm control for multi-robot systems inspired by herding behaviors. *CScience China Technological Sciences* **67**, 2191–2202 (2024)
- [5] Du, X., Guo, Q., Li, H., Zhang, Y.: Multi-uavs cooperative task assignment and path planning scheme. *Journal of Physics: Conference Series* **1856**(1), 012016 (2021) <https://doi.org/10.1088/1742-6596/1856/1/012016>
- [6] Chi, W., Ding, Z., Wang, J., Chen, G., Sun, L.: A generalized voronoi diagram based efficient heuristic path planning method for rrt* in mobile robots. *IEEE Transactions on Industrial Electronics* **PP**(99), 1–1 (2021)
- [7] Garrido, S., Moreno, L., Abderrahim, M., Monar, F.M.: Path planning for mobile robot navigation using voronoi diagram and fast marching. In: *IEEE/RSJ International Conference on Intelligent Robots & Systems* (2006)
- [8] Hao, G., Lv, Q., Huang, Z., Zhao, H., Chen, W.: Uav path planning based on improved artificial potential field method. *Aerospace* (MDPI Publishing) **10**(6) (2023)
- [9] Li, M., Sun, P.-F., Jeon, S.-W., Zhao, X., Jin, H.: Multi-uav path planning with genetic algorithm. In: *2023 14th International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 918–920 (2023). <https://doi.org/10.1109/ICTC58733.2023.10393352>
- [10] He, W., Qi, X., Liu, L.: A novel hybrid particle swarm optimization for multi-uav cooperative path planning. *Applied Intelligence* **51**(10), 7350–7364 (2021) <https://doi.org/10.1007/s10489-020-02082-8>
- [11] Xie, R., Meng, Z., Wang, L., Li, H., Wang, K., Wu, Z.: Unmanned aerial vehicle path planning algorithm based on deep reinforcement learning in large-scale and dynamic environments. *IEEE Access* **9**, 24884–24900 (2021) <https://doi.org/10.1109/ACCESS.2021.3057485>
- [12] Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., Mordatch, I.: Multi-agent actor-critic for mixed cooperative-competitive environments. *ArXiv abs/1706.02275* (2017)
- [13] Xu, L., Chen, W., Liu, X., Chen, Y.-Y.: Maddpg: Multi-agent deep deterministic policy gradient algorithm for formation elliptical encirclement and collision avoidance. In: Ren, Z., Wang, M., Hua, Y. (eds.) *Proceedings of 2021 5th Chinese Conference on Swarm Intelligence and Cooperative Control*, pp. 239–250. Springer, Singapore (2023)
- [14] Eschmann, J.: In: Belousov, B., Abdulsamad, H., Klink, P., Parisi, S., Peters, J. (eds.) *Reward Function Design in Reinforcement Learning*, pp. 25–33. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-41188-6_3

- [15] Li, S., Wu, F., Luo, S., Fan, Z., Chen, J., Fu, S.: Dynamic online trajectory planning for a uav-enabled data collection system. *IEEE Transactions on Vehicular Technology* **71**(12), 13332–13343 (2022) <https://doi.org/10.1109/TVT.2022.3200458>
- [16] Goudarzi, M., Palaniswami, M., Buyya, R.: A distributed deep reinforcement learning technique for application placement in edge and fog computing environments. *IEEE Transactions on Mobile Computing* **22**(5), 2491–2505 (2023) <https://doi.org/10.1109/TMC.2021.3123165>

7 Declarations

- Ethics approval and consent to participate
Not applicable.
- Consent for publication
All authors reviewed and agreed to publish the manuscript.
- Funding
Not applicable
- Authors' contribution
B.M. Wrote the main manuscript and prepared all the manuscript figures. C. Y. contributes to the design of the experiment. B.M., C.Y., and L.Y. reviewed the manuscript. All authors reviewed and carefully proofread the manuscript.
- Acknowledgements
The authors thank Dr. Chen Xinzhuang for technical support and Prof. Liu Yi for critical review of the manuscript.