electronics

MDPI

Article

Dynamic Path Planning for Multiple UAVs with Incomplete Information

Junjie Xue [1], Jie Zhu [1,*], Jiangtao Du [2], Weijie Kang [3] and Jiyang Xiao [1]

[1] Air Traffic Control and Navigation College, Air Force Engineering University, Xi'an 710038, China
[2] School of Aerospace and Architectural Engineering, Harbin Engineering University, Harbin 150006, China
[3] Department of Electronic Engineering, Rocket Force University of Engineering, Xi'an 710025, China
* Correspondence: 18392888960@163.com

Abstract: To address the dynamic path planning for multiple UAVs using incomplete information, this paper studies real-time conflict detection and intelligent resolution methods. When the UAVs execute the task under the condition of incomplete information, the mission strategy of different UAVs may conflict with each other due to the difference in target, departure place, time and other factors. Based on the multi-agent deep deterministic policy gradient algorithm (MADDPG), we designed new global reward and partial local reward functions for the UAVs' path planning and named the improved algorithm as a complex memory driver-MADDPG (CMD-MADDPG). Thus, the trained UAVs can effectively and efficiently perform path planning tasks in conditions of incomplete information (each UAV does not know its reward function and so on). Finally, the simulation verifies that the proposed method can realize fast and accurate dynamic path planning for multiple UAVs.

Keywords: UAVs; path planning; incomplete information; MADDPG; reinforcement learning

Citation: Xue, J.; Zhu, J.; Du, J.; Kang, W.; Xiao, J. Dynamic Path Planning for Multiple UAVs with Incomplete Information. *Electronics* **2023**, *12*, 980. https://doi.org/10.3390/electronics12040980

Academic Editors: Srikanta Patnaik, Shengqing Li and Jianqi Li

Received: 29 December 2022
Revised: 1 February 2023
Accepted: 9 February 2023
Published: 16 February 2023

1. Introduction

As early as the Stone Age, human beings learned that group warfare could exert a combat power far beyond the cumulative effect of individuals. In recent years, the formation flight of Unmanned Aerial Vehicles (UAVs) has developed rapidly and attracted the attention of all parties. It is a new operational concept model with quantitative advantages, cost advantages and intelligent synergy advantages rapidly gaining popularity among military powers [1–3].

Due to their advantages of high flexibility, strong maneuverability, low safety risk, low cost and good robustness, the use of UAVs will be a significant aspect of future confrontations. UAVs can make full use even of contested airspace. Increasing numbers of UAVs in that airspace will inevitably lead to multi-directional UAV flight vectors, which will significantly increase the possibility of flight conflicts and seriously affect the safe flight of UAVs. Therefore, a question that must be answered is how the UAVs in the swarm can approach enemy UAVs and strike collectively through cooperative decision-making to minimize losses and quickly complete their strike missions. This kind of multi-agent game research also has important practical significance [4]. Among them, the problem of multi-agent reinforcement learning has been proposed as early as the last century, and stochastic games are generally used to generate mathematical definitions. Unlike Markov models, stochastic games have multiple action spaces and reward functions and are used extensively, from open AI training in games to robotics. In industrial applications, reinforcement learning is becoming a practical component of large systems. However, most of reinforcement learning's successes are in the domain of a single agent, so multi-agent reinforcement learning needs to study more complex problems, such as teamwork, conflict detection and resolution [5,6].

Conflict detection and intelligent resolution are the main problems to be solved in UAV path planning. Flight conflict refers to a state in which the distance between two

aircraft is less than a specific minimum interval, which threatens the safety of the UAV. Conflict detection determines whether a UAV has entered the protected area of another UAV based on information such as UAV performance, current flight status, flight plan, etc. Conflict resolution refers to measures that could avoid conflict, such as planning a good trajectory and getting rid of possible conflicts when a flight conflict is detected.

There have been many studies on detecting UAV conflict, and the methods used by various researchers often differ. For example, detection methods for UAVs conflict can be divided into two main categories: deterministic conflict detection, based on real-time flight dynamics, meteorological information, UAV flight plans and careful consideration of navigation performance errors; and probabilistic conflict detection, based on assessing the influence of uncertain factors such as meteorology on future tracks to determine the probability that UAVs will collide in the future. Among them, incomplete information is a typical feature of probabilistic conflict detection, which is shown in Figure 1. In the figure, the black and white circle is the UAVs of different sides, the gray circle is the obstacle and the dotted circle indicates the uncertainty of the obstacle. In the research process, the uncertainties of obstacles, opponents and environment are reflected in the reward function, so it is of great significance to study the path planning of UAVs with unknown reward functions.

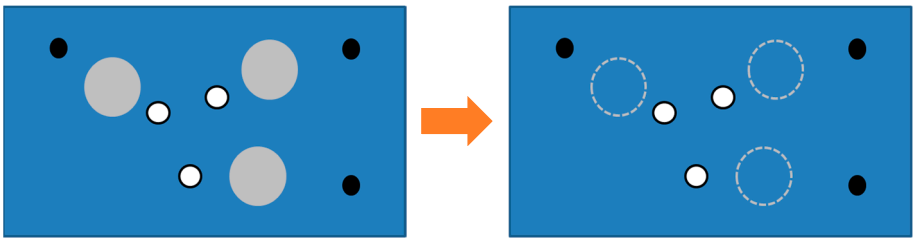


Figure 1. Probabilistic conflict detection with incomplete information.

For deterministic conflict detection, Florence Ho [7] introduced an ORCA adaptive algorithm to resolve conflict detection and resolution (CDR) for possible conflicts between UAVs of different service providers. The adaptive ORCA algorithm solves the practical problems inherent in deploying UAVs in shared airspace, such as navigation inaccuracy, communication overhead and flight phase. Flying multiple unmanned aircraft or operating these aircraft in commercial airspace increases the likelihood of a collision. B. M. Albaker [8] developed a new functional architecture for the UAV collision avoidance system and an algorithm to determine the collision avoidance criteria based on the nominal state projection. Roberto Conde [9] proposed a conflict detection and resolution method for cooperative UAVs in shared airspace. It is based on the axis-aligned minimum bounding box algorithm to detect conflicts. The detected disputes are cooperatively resolved using a genetic algorithm that modifies the UAV trajectory at a minimal cost. Florence Ho [10] studied first-come, first-serve (FCFS) and "batch" processing of Unmanned Aircraft Systems (UASs) operation requests. The throughput of them was compared. The air traffic topology was analyzed for UAV delivery. Then, they developed a new MAPF model for the pre-flight CDR method. This CDR method supports decentralized conflict resolution, with different "agents" (here UAS Service Providers) managing their UAV operations, providing all UAVs with collision-free flight paths before takeoff [11]. The above researchers conducted analysis and intelligent resolution of conflict detection based on deterministic information. Although they achieved good experimental results, they did not fully adapt to the emergency and uncertain information in the actual situation, so the processing ability was unstable.

For probabilistic conflict detection, Yu Wan [12] proposed a multi-UAV coordination technique based on consensus algorithm and policy coordination. This model used a distributed conflict detection and resolution method for human-machine formation, an improved space-time integrated conflict detection model, an improved distributed coordination token allocation strategy, and proposed coordination damping to solve the

problem of data loss and transmission delay in the same airspace at the same time. Mingrui Lao [13] proposed an algorithm for conflict detection and another for conflict resolution to generate all possible solutions for potential conflicts, thereby selecting the best strategy for multi-threat scenarios. Chin E. Lin [14] used Automatic Dependent Surveillance-Broadcast (ADS-B) to collect aircraft data to establish collision avoidance. Based on flight maneuvers, they proposed that a detection algorithm create sector ranges to cover UAVs and helicopters' possible flight direction changes. Zhaoxuan Liu [15] first developed a conflict network to analyze pairwise conflict relationships between aircraft, where the detection of a particular aircraft is called an edge, and the conflict severity is measured as the weight of this edge. In addition, they designed an improved PageRank algorithm to identify critical aircraft that are system safety bottlenecks and implemented centralized Conflict resolution Sequence Assignment (SA) to ensure that these critical aircraft take primary responsibility for discrimination and are deconflicted first. Austin L. Smith [16] developed and implemented a collision avoidance algorithm based on an aggregate collision cone approach, ranging from a single platform capable of independently performing all collision avoidance functions to a diversity of collision avoidance commands that execute ground station calculations. Jian Yang [17] used a geometric method to describe the relationship between UAV conflicts, considering actual and potential conflicts, and formalized the CDR problem as a nonlinear optimization problem to minimize maneuvering costs. Furthermore, they designed a two-layer strategy consisting of Stochastic Parallel Gradient Descent (SPGD) and an interior-point algorithm to efficiently solve non-convex optimization problems. The researchers consider the problem of UAV cluster conflict detection in the case of incomplete or uncertain information and effectively solve the problem of UAV cluster conflict detection and intelligent resolution in the case of uncertain data. However, the stability and efficiency of their results still need to be revised to be satisfactory.

The problem of multi-UAV CDR is one of the main topics of UAV cooperative control system research. However, conflict avoidance requires studying the multi-agent path planning (MAPF) problem to calculate the optimal result, thereby improving escape efficiency. In a MAPF setting, agents in the environment must follow paths to reach their target location without colliding with each other, usually in a distributed setting, and generally considering the multi-agent independence case, first calculating individual payoffs and then considering global payoffs. Researchers from various countries have proposed many methods to minimize the "global indication" and maximize the "benefit" or the optimization method. This paper will use the improved multi-agent deep deterministic policy gradient (MADDPG) algorithm to construct a multi-agent system and regards the time-space multi-domain UAV conflict detection and intelligent resolution as a complete system optimization problem. In this paper, to make this system work, the two sub-problems of conflict detection and intelligent resolution will be solved simultaneously [18–23].

Global scholars have conducted much research on UAV games and obtained many valuable results, but many problems still need to be solved. First, while there are many research results on multi-UAV conflict detection in circumstances of complete information, there are few on multi-UAV conflict detection and resolution in uncertain environments with incomplete information. Second, current intelligent algorithms for UAV conflict detection and resolution are mostly traditional path-planning algorithms. For an algorithm in a multi-agent environment setting, the strategy of each agent changes with the progress of training. In the human-machine environment, the defects of slow convergence speed and low precision are magnified due to the large number of agents involved and the resulting complexity. In addition, the algorithm's applicability decreases with increasing numbers of UAVs. Finally, UAV path planning is primarily concentrated in a single centralized space. At present, there are few studies on the combination of the time domain and the space domain, which is difficult to adapt to the modern large-scale battlefield environment.

In view of the above problems, a UAV path planning strategy based on the MADDPG algorithm is proposed to realize the time-space multi-domain conflict detection and intelligent resolution of UAVs without the player knowing their reward function. At a

time step, each agent chooses an action and receives a numerical value as its payoff or perceived payoff in the game. Unlike virtual games and optimal response dynamics that require knowledge of other players' behavioral histories, our learning algorithm relaxes this assumption. It is often unreasonable and unrealistic in applications to assume the ability to observe the actions of other parties, i.e., to expect to have complete information. Furthermore, we believe that the state space of the game and its transition laws between states is unknown. In addition, the agent does not know the action space of other agent units, the migration strategy and specific speed information of enemy UAV and the location information of the threat area. Therefore, we want to address how much the agent can expect to learn in this situation [24–26].

2. Problem Modelling and Description

2.1. The Problem of UAVs Conflict Detection

The problem of UAV conflict detection arises because UAVs perform their tasks, and each UAV needs task coordination. Our UAV needs to "find" the enemy UAV and avoid it geographically. In a conflict, our UAV must be highly coordinated in time and space to avoid collisions and repeat appearances in the same "area" during detection. Enemy UAVs need to perform patrol tasks in the "enemy base camp," and when they find our UAVs, they can escape or confront. To focus on the problem of UAV conflict detection and intelligent resolution, our UAVs and the enemy UAVs correspond one-to-one and do not affect each other (physical factors such as collision will not occur).

2.2. Kinematics Model of UAVs

The UAV motion collision problem will be considered in the two-dimensional plane to reduce the complexity of the problem, enabling the following two-dimensional UAV kinematics simplified model to be obtained:

$$\dot{x} = v\cos\varphi \tag{1}$$

$$\dot{y} = v\sin\varphi \tag{2}$$

where (x, y) is the real-time position coordinates of the UAV, v is the cruising speed of the UAV, and φ is the flight heading angle. The continuous trajectory of the UAV can be modeled as a sequence of discrete points (waypoints), which is convenient for computer processing. Because of the velocity vector, there is a direction between every two adjacent waypoints. The two-dimensional position and corresponding time constitute the trajectory of each UAV.

For UAV$_i$ ($i = 1, 2, \ldots, n$), its dynamic characteristics can be described in the Cartesian coordinate system: the initial center position of the UAV$_i$ is $\left(p_0^i, q_0^i\right)$ and the velocity vector of the UAV$_i$ in the time t step is $\left(v_t^i\cos\varphi, v_t^i\sin\varphi\right)$. The center coordinate of the nth threat area is (a_n, a_n), the radius is l, and the velocity vector is $\left(v_t^k\cos\varphi, v_t^k\sin\varphi\right)$. Consequently, the position update of the UAV after time Δt is:

$$p_t^i = \begin{bmatrix} x_t^i \\ y_t^i \end{bmatrix},\ v_t^i = \frac{dp_t^i}{dt} = \begin{bmatrix} v_{x,t}^i \\ v_{y,t}^i \end{bmatrix} = \begin{bmatrix} v_t^i\cos\varphi \\ v_t^i\sin\varphi \end{bmatrix} \tag{3}$$

$$\frac{d\varphi_t^i}{dt} = \omega_t^i \tag{4}$$

In this research, to definitely find and reach the target position, the speed of our UAV should be greater than that of the enemy UAV, so the speed constraint formula is set as follows:

$$v_{pmax} > v_{emax} \tag{5}$$

At the same time, in the process of confrontation, the UAV cannot leave the established "battlefield," and the basic parameters of the arena constrain it. There are various threat

zones in the environment, and the coordinates of the threat zone are unknown. When flying, the UAV cannot pass over the threat zone or collide, so the distance l between the UAV and the threat area and other UAVs should satisfy as follows:

$$l \geq l_{meance} + Ro_U \tag{6}$$

In the formula, l_{meance} is the radius of the threat area and Ro_U is the radius of the UAV. In order to use an algorithm to avoid such collisions, this paper sets a pseudo-collision reward function. When a collision occurs, the UAV receives a negative reward. The critical area around the threat area and the UAV based on the original location is extended, which is equivalent to increasing the radius of the threat area and the UAV, to improve the efficiency of collision avoidance. The initial collision detection calculation standard is that the distance between the objects is less than the sum of their radii, and its occurrence is considered a collision. After adding the critical area, two UAVs collide on the edge of the critical area, and the agent is given a negative reward, equivalent to an early warning mechanism for collision. In this way, a specific collision avoidance reaction time can be assigned to the UAV.

We constructed extensive simulations with different tuning parameters and sensing range (SR) values to observe performance efficiency. However, UAVs' maneuvering efficiency should be optimal so that UAVs can reach their respective destinations in the shortest possible time.

$$efficiency = \frac{1}{n}\sum_{i=1}^{n}\frac{t_i^e}{t_i^r} \tag{7}$$

In the formula, t_i^e is the expected flight time, and t_i^r is the actual flight time of the actual UAV$_i$ ($i = 1, 2, \ldots, n$).

3. Improved MADDPG Algorithm

3.1. Multi-Agent Reinforcement Learning Algorithm

The continuous improvement and development of multi-agent reinforcement learning provide a new solution for multi-UAV target assignment and path planning. MADDPG performs well in multi-agent games, wherein target allocation and path planning problems are the games' ultimate underlying basis. Both sides of the game essentially require UAVs to select "appropriate" targets to strike (or defend) and to minimize the total distance of the UAV formation (or prevent this trend). Moreover, despite the dynamism of environmental information and target attributes, the MADDPG model enables the UAVs to deal with the changes in the environment increasingly expertly as the training progresses.

In a multi-agent extension of Markov decision processes (MDPs), an MDP consists of a five-tuple $\langle S, A, P, R, \gamma \rangle$ in N agents, where S and A represent the state space and action space: they have their state space $S = \{S_1, S_2, \ldots, S_N\}$ and action space $A = \{A_1, A_2, \ldots, A_N\}$. $P : S \times A \to S$ represents the state transition probability matrix, $R : S \times A \times S \to R$ represents the reward function and γ is the decay coefficient of the cumulative discount reward.

In a multi-agent system, the reward obtained by state transition depends on the joint strategy $\mu : \theta(a|s) = \prod_{i \in N} \pi_i(a_i|s)$, which is the joint decision-making strategy of all agents. The value function of each agent is as follows:

$$V^{\pi}(s) = \mathbb{E}^{\pi}[\sum_{t=0}^{T}\gamma^t R(t+1)|s_0 = s] \tag{8}$$

In this formula, T is the total time, t is the current simulation time and s is the previous state of the environment. The ultimate goal of the multi-agent Markov game is to find the optimal joint strategy π^*, which maximizes the cumulative expected return of the entire system.

The Bellman equations of state-value function under multi-agent are as follows:

$$V_\pi(s) = \sum_a \pi(a|s) \sum_{s'} p(s'|s,a) \left[R(s,s') + \gamma V_\pi(s') \right] \tag{9}$$

where the expected value of the reward R with states is $V_\pi(s)$.

The MADDPG algorithm used in this paper integrates the UAVs of the player and the enemy into the same agent system for training. The essence of this method is a Markov decision process, and the problem of multi-UAV target assignment and path planning is discrete across multiple time steps. After each step, the UAVs and the environment are treated as a state, and each UAV can observe the current environment and then take the following action according to its policy network. However, each UAV cannot fully monitor the location of or receive comprehensive intelligence on the enemy target. Furthermore, because multiple enemy UAVs' values are unclear, our UAVs operate and attempt to fight in an incomplete information state.

3.2. MADDPG Algorithm

In order to solve the problem of reinforcement learning with incomplete information, we introduced the observation space of a partially observable Markov decision process (POMDP), which is on the basis of MDPs. A POMDP is defined as A tuple of $\langle S, A, P, R, \gamma, O, P_o \rangle$, where S, A, P, R and γ are similar to the definition of the MDP. $O = \{\overline{o}_1, \overline{o}_2, \ldots, \overline{o}_K\}$ is the observation space, note that $\overline{o}_1$ is different from o_1 that is the observation perceived by agent 1. Agents may observe differently at the same state because of the observation probability P_o. The MADDPG algorithm [15] is an extension of the DDPG algorithm in multi-agent reinforcement learning. It uses a "centralized training, decentralized execution" architecture, which requires additional state and action information about other agents only in the training phase. The state of the agent itself is necessary to output the policy action. The architecture frame diagram of MADDPG is shown in Figure 2. Each agent has two networks: an Actor-network π and a critic network Q. The actor network calculates the action to be performed based on the agent's state, and the critic network is responsible for evaluating the movement to improve the performance of the Actor-network. Using the Q-value network to break the correlation by randomly reading the experience pool data makes the training results more stable. At the same time, during the training process, the Actor-network only copies and observes its information, while the critic network is responsible for monitoring other agents.

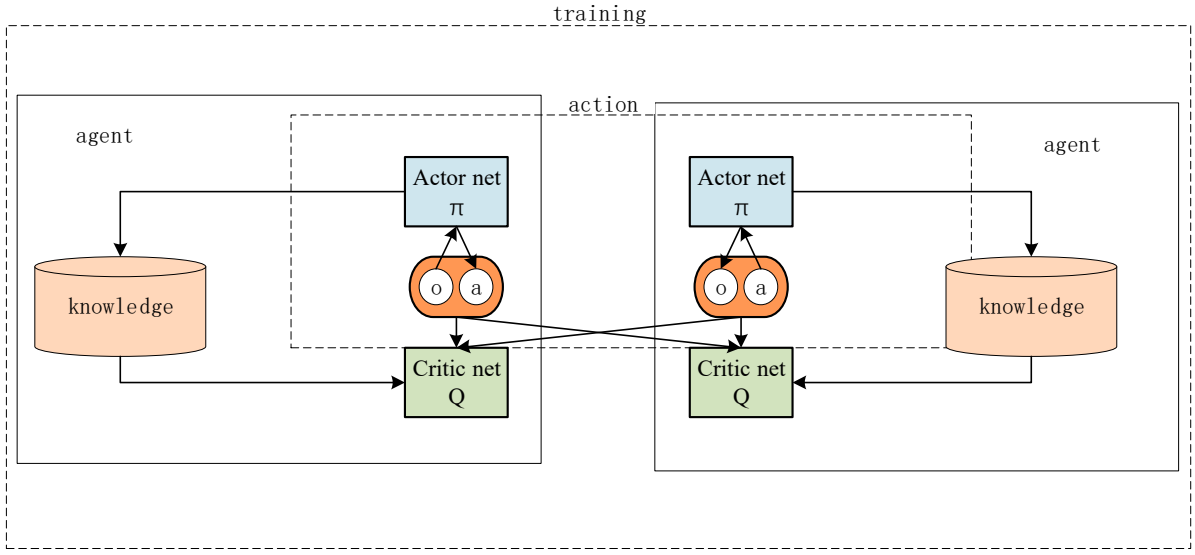


Figure 2. MADDPG algorithm framework.

A random policy θ_i used by agent i in the MADDPG algorithm, among them the policy should depend on the history of observation. $\theta_i : O_i \times A_i = \{\theta_1, \theta_2, \ldots, \theta_N\}$, informs the strategies of all agents $\mu = \{\mu_{\theta_1}, \mu_{\theta_2}, \ldots, \mu_{\theta_N}\}$. The expected policy gradient for agent i can thus be obtained as:

$$\nabla\theta_i J(\theta_i) = \mathbb{E}_{o\sim\chi,D}\left[\nabla\theta_i \mu_{\theta_i}(a_i|o_i)\nabla a_i Q_i^{\mu}(\chi, a_1, \ldots, a_N)\Big|_{a_j=\mu_{\theta_j}(o_j)}\right] \tag{10}$$

where o_i is the observed value of agent i; $\chi = \{O_1, O_2, \ldots, O_N\}$ and represents the state of the agent, which could simplify the value of Q function expression; and $Q_i^{\mu}(\chi, a_1, \ldots, a_N)\Big|_{a_i=\mu_{\theta_i}(o_i)}$ is the Q value function, which uses State χ and all agent actions to estimate the state-action value Q of agent i. Since each Q_i^{μ} is learned individually, the agent can have arbitrary reward structures, including conflicting rewards in competitive environments. D represents the experience pool contains a series of tuples $(X, X', a_1, \ldots, a_N, r_1, \ldots, r_N)$ to record all agent training samples. X' is the new state of the agent after acting, and r_i is the reward value of agent i.

Updating the critic network loss function can be shown as:

$$L(\theta_i) = \mathbb{E}_{x,a,r,x'}\left[\left(Q_i^{\mu}(x, a_1, \ldots, a_k) - y\right)^2\right] \tag{11}$$

where $y = r_i + \gamma Q_i^{\mu'}\left(X', a_1', \ldots, a_k'\right)\Big|_{a_j'=\mu_{\theta_j}'(o_j)}$.

3.3. CMD-MADDPG Algorithm with Incomplete Information

Based on the MADDPG algorithm, the complex memory driver (CMD) communication mechanism is introduced to enable agents to use the shared memory as the communication channel. Before performing an operation, the agent reads the memory first and then writes the response. In this case, the agent's strategy is related to its observation and interpretation of the memory set. Based on the above analysis and applying relevant game theory, it is possible to obtain the following improvements to the incomplete information scenario in the MADDPG algorithm for UAVs conflict detection:

(1) N represents the participant function, where N= {1, . . . , M, . . . , N}, M is the number of our UAVs, and N-M is the number of enemy UAVs;
(2) Agent state $\chi = \{O_1, \ldots, O_M, \ldots, O_N\}$;
(3) The probability of the enemy selecting strategy S is δ under the state χ;
(4) The Q value function can be obtained from the previous discoveries and written as $Q_i^{\mu}(\chi, a_1, \ldots, a_M, \delta_{M+1}a_{M+1}, \ldots, \delta_N a_N)\Big|_{a_i=\mu_{\theta_i}(o_i)}$;
(5) The expected policy gradient for agent i can thus be changed to:

$$\nabla\theta_i J(\theta_i) = \mathbb{E}_{o\sim\chi,D}\left[\nabla\theta_i \mu_i(a_i|o_i)\nabla a_i Q_i^{\mu}(\chi, a_1, \ldots, a_M, \delta_{M+1}a_{M+1}, \ldots, \delta_N a_N)\Big|_{a_i=\mu_i(o_i)}\right] \tag{12}$$

The critic network loss function can therefore be updated to:

$$L(\theta_i) = \mathbb{E}_{x,a,r,x'}\left[\left(Q_i^{\mu}(x, a_1, \ldots, a_m, \delta_{m+1}a_{m+1} \ldots, \delta_k a_k) - y\right)^2\right] \tag{13}$$

where $y = r_i + \gamma Q_i^{\mu'}\left(X', a_1', \ldots, a_k'\right)\Big|_{a_j'=\mu_{\theta_j}'(o_j)}$.

3.4. Analysis of Reward Function

The reward function has been set as global and location local rewards. Its primary purpose is to guide the UAV to reach the dynamic target in the shortest distance and avoid conflicts.

When multiple UAVs perform tasks, two possible conflicts should avoid the conflict between the UAV and the threat area and the conflict between various UAVs. Therefore, this paper needs to design an appropriate collision reward function to avoid collision. When a collision occurs, the UAV gets a negative reward.

In order to get the UAV to the dynamic target quickly, we need to simulate and calculate the distance to the enemy's dynamic target. We will approximate the action of the dynamic target in each time step according to the binomial distribution to make the corresponding action space and state space. According to the constraint design in this paper, it can push step t each time and the distance d_t of the UAV$_i$ from the dynamic enemy target i is:

$$\cos\varphi = \frac{\left(L_t^i\right)^2 + \left(d_0^i\right)^2 - \left(M_t^i\right)^2}{2L_t^i d_{0,t}^i} \tag{14}$$

$$d_t^i = L_t^i - \sqrt{\left(p_{t+\Delta t}^i\right)^2 + \left(q_{t+\Delta t}^i\right)^2} \tag{15}$$

The relationship between the initial position of our UAV, its position at time step t and the dynamic target position of the enemy at time step t is approximately linear. L_t^i is the initial distance, $d_{0,t}^i$ is the distance between the enemy's initial and current position. M_t^i is the moving distance of the dynamic target, and φ is the angle between the initial position of the UAV and the current position at step t. The distance between the UAVs can be described in Figure 3.

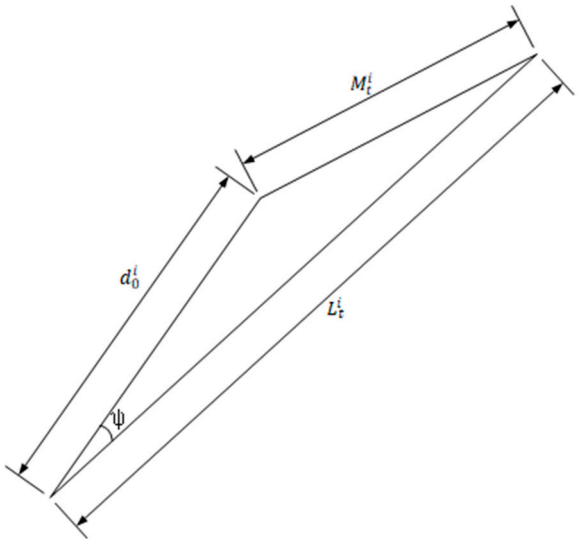


Figure 3. The distance between the UAVs.

To make the UAV sufficiently flexible in completing the task, the UAV will directly defeat the target if it "accidentally" finds other enemy UAVs during the training process. At this time, the UAV corresponding to the captured target will continue to perform the different tasks. The target will be recalculated and allocated according to the enemy's fuzzy position reward.

The pseudocode for the flow of training in the UAV training algorithm is given below in Algorithm 1:

Algorithm 1: CMD-MADDPG algorithm

1: Initialize the number of UAVs k, the number of targets m, the number of threat areas L and the critical area σ
2: Initialize the policy network π_i and evaluation network Q_i of UAV$_i$ and the parameters of the network θ^{π_i} and θ^{Q_i}
3: For episode = 1 to MAX Episode do
4: Randomly initialize UAVs, obstacles and target positions in a set UAV environment
5: For t = 1 MaxStep do
6: get environment status S
7: Get UAV action a_i
8: Interact the joint actions $a = [a_1, \ldots, a_N]$. of all UAVs with the environment, and return the UAV's return r_i, the number of collisions and the next state χ'
9: Store samples $(S,\ a,\ r_i,\ \chi')$ into the experience pool
10: Update environment state
11: for i = 1 to k do
12: Randomly sample S samples $(S,\ a,\ r_i,\ \chi')$ from the experience pool to form batch samples
13: Compute the objective of the joint behavior function from the sampled data
14: Update the policy network of UAVs by formula
15: Update the evaluation network of UAVs by formula
16: end for
17: Update each UAV's target policy network and target evaluation network in a soft-update manner
18: end for
19: end for

4. Simulation Results and Analysis

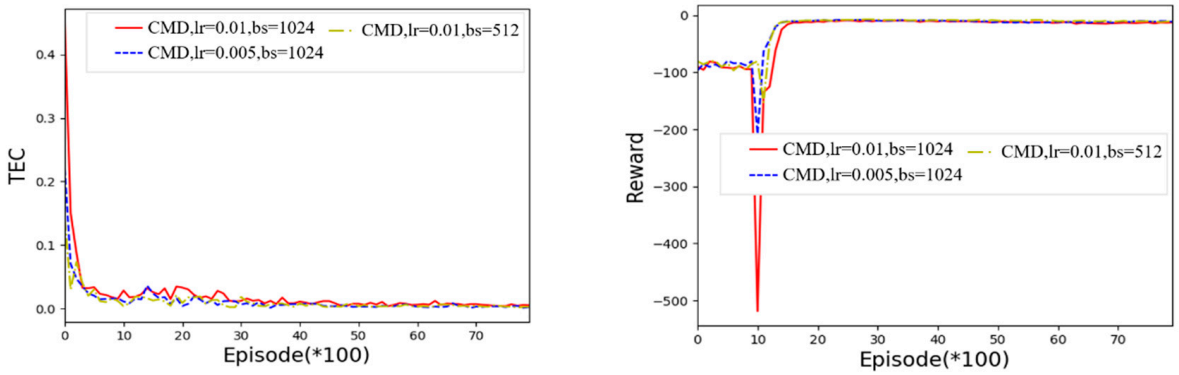
Based on the OPENAI platform, we used the CMD-MADDPG to create an incomplete information training environment for multi-UAV conflict detection. The experiment will use several offensive UAVs and dynamic targets while the environment contains multiple fixed threat zones that randomly appear. Therefore, for offensive UAVs, real-time conflict detection and resolution are needed. The offensive UAVs communicate with each other, but they do not know the moving direction of the dynamic target and the location information of the fixed threat zone. In this condition, it is difficult to realize the final path planning with traditional methods. The simulated environment will take the geometric center of the environment as the origin of the Cartesian coordinate system. The agent size is 0.05, the target size is 0.07 and the threat area size is 0.09. Positions will randomly generate in each training scenario, and while the speed of the UAV is set as 0.02, the target movement speed changes with time steps. The experiment will use two indicators to measure the algorithm performance in the CMD-MADDPG path planning of dynamic targets in two modes. The indicators are the number of collisions (between agent and threat areas, total training episode (TEC) and global reward. The dynamic target has a random direction (every 45° is a direction) movement pattern. The following Table 1 lists the hyperparameters.

Table 1. Hyper Parameters.

Hyper Parameters	Size
Episodes	20,000
Learning-rate	0.01
Discount factor	0.95
Batch-size	1024

By establishing the CMD-MADDPG algorithm model, three UAVs are trained to plan a good path in the case of two fixed threat areas and reach the dynamic target location moving in random directions in the fastest time and the shortest distance. From Figure 4a,b, it can be seen that as the number of training increases, the effect of batch size (bs) and learning rate (LR) on the algorithm reward is gradually different. In Figure 4a, the TEC gradually

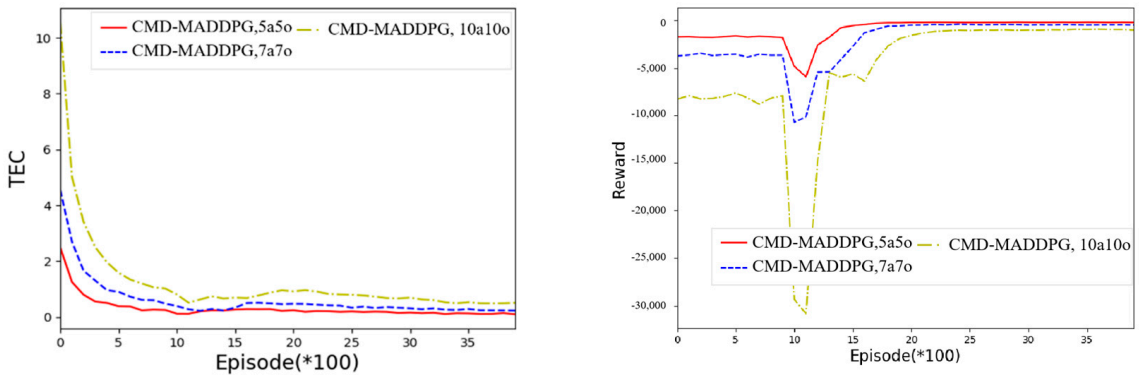
converges after the training round reaches 1000 episodes. In Figure 4b, the reward function is lower before the training round reaches 1000 episodes due to the unsatisfactory training effect. Between steps 1000 and 2000 of the training round, the reward function decreases abruptly and then gradually converges and stabilizes. It can be seen that the learning rate is 0.005, and the training effect is at its best when the batch size is 1024 by comparing the two figures.



(**a**) (**b**)

Figure 4. Collision rate curves and reward values for three models: (**a**) Collision rates between UAVs and threat zones (TEC); (**b**) global reward.

To solve the conflict reasonably and intelligently despite the threat area and to reach the target location in the fastest and shortest distance, 5, 7 and 10 UAVs are trained in the mode. However, it can be seen that the training is entirely effective in the multi-UAV environment. Moreover, the algorithm successfully reduces the collision rate of the threat zone to 0 under incomplete information (i.e., it avoids the actual collision), realizes the capture of the target, and the reward successfully converges. The experimental results are shown in Figure 5.

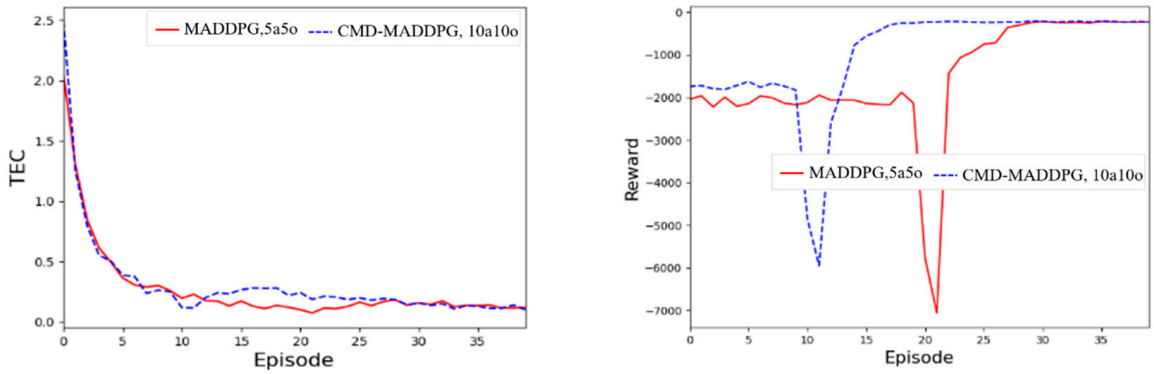


(**a**) (**b**)

Figure 5. Collision rate and reward value of three types of UAVs: (**a**) The collision rate between the UAV and the threat zone; (**b**) global reward. (xayo refers to x targets and y obstacles).

There were two model structures established for training MADDPG and CMD-MADDPG. The resulting reward function is shown in Figure 6. Figure 6b shows the reward changes of two UAVs in each training round during the training process. The x-coordinate represents the number of training rounds, and the y-coordinate sub-represents the cumulative rewards of two UAVs in each training round. It can be seen from the figure that with the increase in training times, the reward gradually increases. When the number of training rounds reaches 2500, the reward curve area of the two algorithms is gentle and

tends to converge. However, it can be seen from Figure 6a that the UAV collision rate of the MADDPG model is much higher than that of the multi-UAV collision rate trained by the CMD-MADDPG model. Comparing the two algorithms shows that the CMD-MADDPG algorithm has more robust and faster convergence than the MADDPG algorithm. As the path planning method proposed in this paper is a real-time planning method, timeliness is of great significance in the practical application of UAVs, especially in combat and reconnaissance scenarios. Therefore, the actual running time of the algorithm is critical. We used MADDPG and CMD-MADDPG algorithms to conduct five experiments and recorded the time consumption of conflict detection and intelligent resolution, as shown in Table 2. Comparing the two algorithms shows that the CMD-MADDPG algorithm has more robust and faster convergence than the MADDPG algorithm.



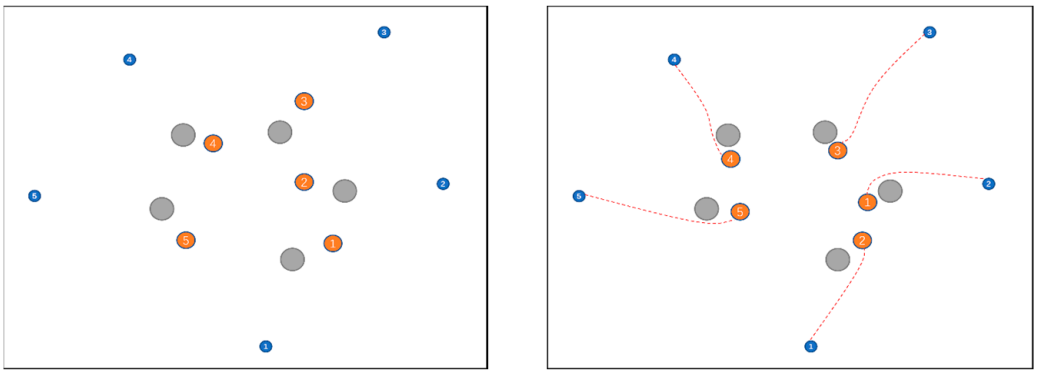
(**a**) (**b**)

Figure 6. Comparison diagram of return collision rate of two algorithms: (**a**) The collision rate between the UAV and the threat zone; (**b**) global reward. (xayo refers to x targets and y obstacles).

Table 2. Comparison diagram of experimental consumption time.

	MADDPG	CMD-MADDPG
1	10.93	3.804
2	10.377	3.652
3	8.338	3.669
4	7.872	3.613
5	7.641	3.614

Figure 7 shows the schematic diagram of cross-domain conflict detection of UAVs in the scenario of five UAVs and five threat areas.



(**a**) (**b**)

Figure 7. 5-UAV trajectory diagram: (**a**) UAV random initial positions; (**b**) UAV training result diagram.

Figure 7a shows the grey threat area and cyan dynamic target randomly set before the test. The orange circle represents our UAVs. Different numbers in the circle correspond to different UAV numbers. Figure 7b illustrates the UAV training result following the initialization condition shown in Figure 7a, with the red curves representing the trajectory of the UAV tracking target. It can be seen from the figure that the UAV can successfully avoid the threat zone and successfully reach the target position. During this training process, UAVs 1 and 2 intelligently exchanged cross-domain dynamic targets according to the algorithm to avoid conflicts, perform intelligent resolution and complete target capture.

It is noteworthy that the MADDPG algorithm is improved in this paper to make it applicable to incomplete information conditions. Specifically, multi-UAV path planning is realized under unknown reward function conditions. And the experimental results show that the proposed CMD-MADDPG algorithm has improved the convergence speed and accuracy.

5. Conclusions

We proposed a multi-UAV deep strategy reinforcement learning algorithm to solve the problem of a multi-UAV scenario involving dynamic threat targets. Since the traditional UAV reinforcement learning algorithm is based on having global information, we suggest adopting a design of multi-UAV reinforcement learning task reward for incomplete information and conducting a multi-UAV conflict based on the CMD-MADDPG algorithm. Our experiments showed that the multi-UAV reinforcement learning algorithm CMD-MADDPG has a good application effect, showing a certain practical value.

Author Contributions: Conceptualization, J.X. (Junjie Xue) and J.Z.; methodology, J.X. (Junjie Xue); software, J.D.; validation, W.K., J.D. and J.X. (Jiyang Xiao); investigation, J.X. (Junjie Xue); writing—original draft preparation, J.D.; writing—review and editing, W.K.; visualization, J.Z.; supervision, J.X. (Junjie Xue); funding acquisition, J.X. (Junjie Xue). All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Natural Science Foundation of Shaanxi Province, grant number 2021JQ-368 and National Social Science Foundation of China, grant number 2020-SKJJ-C-033.

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Xuan, S.; Zhou, H.; Ke, L. Summary of UAV cluster confrontation game. *Command. Inf. Syst. Technol.* **2021**, *2*, 27–31.
2. Rios, J. *NASA UTM Strategic Deconfliction Final Report*; Technical Report; NASA Ames Research Center: Washington, DC, USA, 2018.
3. Zhang, H.; Xin, B.; Ding, Y. Online Path Planning of Messenger UAV in Air-Ground Collaborative System. In Proceedings of the 2019 Chinese Control Conference (CCC), Guangzhou, China, 27–30 July 2019; pp. 5875–5880.
4. Du, W.; Ding, S. A survey on multi-agent deep reinforcement learning: From the perspective of challenges and applications. *Artif. Intell. Rev.* **2021**, *54*, 3215–3238. [CrossRef]
5. Pallottino, L.; Feron, E.M.; Bicchi, A. Conflict resolution problems for air traffic management systems solved with mixed integer programming. *IEEE Trans. Intell. Transp. Syst.* **2002**, *3*, 3–11. [CrossRef]
6. Wollkind, S.; Valasek, J.; Ioerger, T. Automated Conflict Resolution for Air Traffic Management Using Cooperative Multi-Agent Negotiation. In Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, Providence, RI, USA, 16–19 August 2004; p. 4992.
7. Ho, F.; Geraldes, R.; Goncalves, A.; Cavazza, M.; Prendinger, H. Improved conflict detection and resolution for service UAVs in shared airspace. *IEEE Trans. Veh. Technol.* **2018**, *68*, 1231–1242. [CrossRef]
8. Albaker, B.M.; Rahim, N.A. Straight Projection Conflict Detection and Cooperative Avoidance for Autonomous Unmanned Aircraft Systems. In Proceedings of the 2009 4th IEEE Conference on Industrial Electronics and Applications, Xi'an, China, 25–27 May 2009; pp. 1965–1969.
9. Conde, R.; Alejo, D.; Cobano, J.A.; Viguria, A.; Ollero, A. Conflict detection and resolution method for cooperating unmanned aerial vehicles. *J. Intell. Robot. Syst.* **2012**, *65*, 495–505. [CrossRef]
10. Ho, F.; Geraldes, R.; Goncalves, A.; Rigault, B.; Oosedo, A.; Cavazza, M.; Prendinger, H. Pre-flight conflict detection and resolution for UAV integration in shared airspace: Sendai 2030 model case. *IEEE Access* **2019**, *7*, 170226–170237. [CrossRef]

11. Ho, F.; Geraldes, R.; Goncalves, A.; Rigault, B.; Sportich, B.; Kubo, D.; Cavazza, M.; Prendinger, H. Decentralized Multi-Agent Path Finding for UAV Traffic Management. *IEEE Trans. Intell. Transp. Syst.* **2020**, *23*, 997–1008. [CrossRef]
12. Wan, Y.; Tang, J.; Lao, S. Distributed Conflict-Detection and Resolution Algorithm for UAV Swarms Based on Consensus Algorithm and Strategy Coordination. *IEEE Access* **2019**, *7*, 100552–100566. [CrossRef]
13. Lao, M.; Tang, J. Cooperative Multi-UAV Collision Avoidance Based on Distributed Dynamic Optimization and Causal Analysis. *Appl. Sci.* **2017**, *7*, 83. [CrossRef]
14. Lin, C.E.; Lai, Y.H.; Lee, F.J. UAV Collision Avoidance Using Sector Recognition in Cooperative Mission to Helicopters. In Proceedings of the 2014 Integrated Communications, Navigation and Surveillance Conference (ICNS) Conference Proceedings, Herndon, VA, USA, 8–10 April 2014; pp. F1-1–F1-9.
15. Liu, Z.; Cai, K.; Xie, J.; Zhu, Y. A Network-Based Conflict Resolution Approach for Unmanned Aerial Vehicle Operations in Dense Nonsegregated Airspace. *IEEE Intell. Transp. Syst. Mag.* **2021**, *14*, 212–232. [CrossRef]
16. Smith, A.L.; Harmon, F.G. UAS collision avoidance algorithm based on an aggregate collision cone approach. *J. Aerosp. Eng.* **2011**, *24*, 463–477. [CrossRef]
17. Yang, J.; Yin, D.; Cheng, Q.; Xie, X. Two-Layer Optimization to Cooperative Conflict Detection and Resolution for UAVs. In Proceedings of the 2015 IEEE 18th International Conference on Intelligent Transportation Systems, Gran Canaria, Spain, 15–18 September 2015; pp. 2072–2077.
18. Fu, X.; Wang, H.; Xu, Z. Multi UAV cooperative pursuit strategy based on de-maddpg. *Acta Aeronaut. Sin.* **2022**, *5*, 530–543.
19. Lowe, R.; Wu, Y.I.; Tamar, A.; Harb, J.; Pieter Abbeel, O.; Mordatch, I. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Volume 30.
20. Xiao, C.; Zou, Y.; Li, S. UAV Multiple Dynamic Objects Path Planning in Air-Ground Coordination Using Receding Horizon Strategy. In Proceedings of the 2019 3rd International Symposium on Autonomous Systems (ISAS), Shanghai, China, 29–31 May 2019; pp. 335–340.
21. Wang, X. Research on Path Planning and Scheduling of UAV in Air to Ground Collaborative Scenario. Master's Thesis, Beijing University of Posts and Telecommunications, Beijing, China, 2021. Available online: https://kns.cnki.net/KCMS/detail/detail.aspx?dbname=CMFD202201&filename=1021130984.nh (accessed on 15 January 2022).
22. Hou, Y.; Hong, H.; Sun, Z.; Xu, D.; Zeng, Z. The control method of twin delayed deep deterministic policy gradient with rebirth mechanism to multi-dof manipulator. *Electronics* **2021**, *10*, 870. [CrossRef]
23. Li, H.; Liu, D.; Wang, D. Integral Reinforcement Learning for Linear Continuous-Time Zero-Sum Games With Completely Unknown Dynamics. *IEEE Trans. Autom. Sci. Eng.* **2014**, *11*, 706–714. [CrossRef]
24. Yan, T. Research and Application of Incomplete Information Game Decision Based on Deep Learning. Master's Thesis, Nanchang University, Nanchang, China, 2019. Available online: https://kns.cnki.net/kcms2/article/abstract?v=4AeVXcGBmm1GHx5c05TR7_gFE8d0ZbemZKxCBTj2KWE9AI9zqjUVFsHNhfny8Uvg2CsEedOmVNglWwpblGlCUOuFk7lXy5CVnyLG5rxgB8P7X-4-7a3KkA==&uniplatform=NZKPT&language=CHS (accessed on 15 February 2022).
25. Zhu, Q.; Tembine, H.; Başar, T. Heterogeneous Learning in Zero-Sum Stochastic Games with Incomplete Information. In Proceedings of the 49th IEEE Conference on Decision and Control (CDC), Atlanta, GA, USA, 15–17 December 2010; pp. 219–224.
26. Dai, W.; Lu, H.; Xiao, J.; Zheng, Z. Task Allocation Without Communication Based on Incomplete Information Game Theory for Multi-robot Systems. *J. Intell. Robot. Syst.* **2018**, *94*, 841–856. [CrossRef]