Approximating Stackelberg Equilibrium in Anti-UAV Jamming Markov Game with Hierarchical Multi-Agent Deep Reinforcement Learning Algorithm

Zikai Feng
Hainan University
Yuanyuan Wu
Hainan University
Mengxing Huang (hainugzlab@163.com)
Hainan University
Di Wu
Shanghai Jiao Tong University

Research Article

Keywords: Anti-Jamming, Markov Game, Hierarchical Multi-agent Deep Reinforcement Learning, Stackelberg Equilibrium

Posted Date: December 28th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-1156014/v1

Approximating Stackelberg Equilibrium in Anti-UAV Jamming Markov Game with Hierarchical Multi-Agent Deep Reinforcement Learning Algorithm

Zikai Feng, *Student member*, *IEEE*, Yuanyuan Wu, Mengxing Huang, *Member, IEEE* and Di Wu

Abstract In order to avoid the malicious jamming of the intelligent unmanned aerial vehicle (UAV) to ground users in the downlink communications, a new anti-UAV jamming strategy based on multi-agent deep reinforcement learning is studied in this paper. In this method, ground users aim to learn the best mobile strategies to avoid the jamming of UAV. The problem is modeled as a Stackelberg game to describe the competitive interaction between the UAV jammer (leader) and ground users (followers). To reduce the computational cost of equilibrium solution for the complex game with large state space, a hierarchical multi-agent proximal policy optimization (HMAPPO) algorithm is proposed to decouple the hybrid game into several sub-Markov games, which updates the actor and critic network of the UAV jammer and ground users at different time scales. Simulation results suggest that the hierarchical multi-agent proximal policy optimization -based anti-jamming strategy achieves comparable performance with lower time complexity than the benchmark strategies. The well-trained HMAPPO has the ability to obtain the optimal jamming strategy and the optimal anti-jamming strategies, which can approximate the Stackelberg equilibrium (SE).

Keywords Anti-Jamming · Markov Game · Hierarchical Multi-agent Deep Reinforcement Learning · Stackelberg Equilibrium

Corresponding author: Mengxing Huang and Di Wu.

Zikai Feng, Yuanyuan Wu and Mengxing Huang are with State Key Laboratory of Marine Resource Utilization in South China Sea、College of Information and Communication Engineering, Hainan University, Haikou, 570228, China, (e-mail: 13523011824@163.com; wyuanyuan82@163.com; hainugzlab@163.com). Di Wu is with Department of Automation, Shanghai Jiao Tong University, Shanghai, 200240, China, (e-mail: hainuwudi@163.com).

I. INTRODUCTION

WITH the urgent demand of wireless communication for high-performance data transmission, people have done a lot of research to improve network capacity and to resist various security attacks [1].

For the security problems in wireless communication systems, UAVs are exploited as different components [2], [3]. UAV shows great potential in many fields because of its low cost, easy deployment, strong mobility and wide application. Considering the high maneuverability and flexibility of UAVs, they are often used to improve the ground wireless communication, such as the mobile air base stations for ground disaster rescue [4].

Additionally, some works use UAVs for relay or friendly interference to enhance the security of communication. For example, in the double-UAV communication system, one UAV is utilized as a relay to connect the interaction of multiple ground users and the other is used as a friendly jammer to disturb the eavesdropper [5].

At the same time, UAVs are also used to interfere with malicious scenes with security threats [6]. For example, UAVs use continuous interference attacks or discontinuous short pulses to conduct malicious interference, resulting in blocking of communication channels and serious deterioration of signal-to-noise ratio (SNR). Therefore, the problem of anti-UAV jamming is worth studying.

To solve the anti-UAV malicious interference problem, some meaningful research has been carried out. In some works, the problem is described as a Markov decision model, and then solved by the method of game theory (GT) [7]. Game theory is a science that studies the decision-making and its equilibrium when decision-makers interact directly. As a powerful mathematical tool, GT is used in many fields such as economics [8], biology [9] and computer science [10].

With the iterative updating of communication technology, UAV interference has become more realistic, harmful and intelligent. The traditional game theory method cannot meet the decision-making solution of high-dimensional action and state space. There is an urgent need to use artificial intelligence methods to solve problems. A powerful tool is reinforcement learning.

Reinforcement learning (RL) emphasizes that the agent

learns the best strategy to interacts with the environment, so as to obtain the maximum cumulative reward. RL algorithms include the value-based algorithms [11], [12] and the policy-based algorithms [13], [14]. The classic value function algorithm is the Q-Learning algorithm [15]. Mnih et al. [16] combined Q-Learning with deep neural network (DNN) and proposed deep Q network (DQN), where DNN is used to represent action value functions. Although Q learning is not affected by the complexity of modeling, it does require discretization of the state and action space, which brings dimensional disasters, especially multi-dimensional continuous states or actions. In this case, the exponential growth of the lookup table will make the problem tricky. In addition, the discretization of continuous variables also limits the search space and may lead to sub-optimal solutions. Double Q learning decouples the selection and evaluation of actions and reduces the overestimation in action evaluation [17].

In reinforcement learning, gradients are adopted to estimate the value of a strategy or the strategy can be estimated directly. The classic strategy gradient method is the Reinforcement algorithm [18], which uses Monte Carlo to estimate the cumulative expected return. Deep deterministic strategy gradient (DDPG) method is extended on the basis of deep Q-learning, and an actor-critic structure is proposed, which successfully solves the RL problem of continuous action domain [19].

However, the strategy gradient method has some difficulties to achieve good results, because this type of method is very sensitive to the learning rate: if the step length is too short, the convergence process will be too slow; to the opposite, it may even make the model performance drop avalanche [20]. The sampling efficiency of this type of method is often very low, and learning simple tasks requires a total number of iterations ranging from one million to one billion. Based on the above analysis, trust region policy optimization (TRPO) is proposed to get rid of the problem of learning rate [21]. By decomposing the reward of new strategy into the reward of the old strategy and other items, the monotonic convergence is achieved. Schulman et al. further proposed the PPO algorithm, which simplifies the solution process of TRPO, and uses generalized advantage estimation (GAE) to balance the variance and bias for the advantage function calculation [22].

So far, reinforcement learning has achieved excellent performance in the single agent field, where the environment is stable [23]. But, in multi-agent scene, the dimensions of solution space increase greatly, which is difficult for learning. In multi-agent domains, each agent not only needs to learn and improve its own strategy, but also learn strategies of other agents in the environment. When multiple agents are involved, the dimensionality of multi-agent systems will become very large and the calculations will be complicated.

In multi-agent systems, the relationship between agents involve cooperation and competition. Combining game theory with RL can give a solution for multi-agent problem. For example, the multi-agent deep deterministic strategy gradient (MADDPG) method is used to approximate Nash equilibrium (NE) of Markov game in power market bidding of multiple strategic power generation companies [5]. In [1], a deep recursive Q network (DRQN) algorithm is used to get optimal communication trajectory of anti-intelligent UAV jamming attack in discrete space, so as to obtain the equilibrium solution.

In most multi-agent tasks, the reinforcement learning method based on Markov decision-making, each agent needs to consider all the environment and other agent information detected by the system sensor at the same time, and search the optimal strategy in all its own action spaces. The learning efficiency is low, and the on-line performance of the learning system is difficult to grasp. At the same time, with the state space expands, the number of learning parameters also increases, which will lead to the disaster of dimension.

In this paper, a hierarchical multi-agent proximal policy optimization algorithm, HMAPPO, is proposed to deal with the anti-UAV jamming task in the three-dimensional (3-D) continuous action space. The jamming strategy and anti-UAV jamming strategies are described as sub-Markov games, which can be used to model the hierarchical competition between the UAV jammer (leader) and multiple ground users (followers) and to make a sequential decision-making. HMAPPO decouples the hybrid game into several sub-Markov games and updates the actor and critic network of the UAV jammer and ground users at different time scales to obtain the optimal jamming strategy and the optimal anti-jamming strategies, so as to approximate the Stackelberg equilibrium.

The contributions of this paper are summarized as follows:

1) The competitive interaction between the UAV jammer and ground users is modeled as a Stackelberg game in the three-dimensional environment with continuous action space. Compared with the existing research on the two-dimensional scene or discrete action space [4], the solution space is larger, and the complexity is higher, which is more realistic.

2) A hierarchical multi-agent proximal policy optimization algorithm is proposed to solve the above Stackelberg game. HMAPPO decouples the hybrid game into several sub-Markov games, in which the actor and critic network of the leader and followers are updated at different time scales.

3) The well-trained HMAPPO algorithm can achieve or exceed the performance of the benchmark reinforcement learning algorithm with lower time cost in different anti-jamming scenarios. Moreover, the Nash equilibrium of the jamming sub-game and anti-jamming sub-games are approximated by HMAPPO to form the Stackelberg equilibrium of the hybrid game.

This paper is structured as below. The system modelling and the problem description are given in Section 2. Section 3 presents the anti-UAV jamming Stackelberg game based on hierarchical MARL method. Section 4 analyzes the simulation results. Section 5 concludes this paper.

II. System Modeling And Problem Description

In this part, the scene of anti-intelligent UAV jamming is modeled firstly, and then the optimization solution of the problem is given.

A. System modeling

The system model is shown in Figure 1. Suppose there is a UAV jammer (leader) and multiple ground users (followers). The downlink transmission from the base station to ground users under the jamming attack of the UAV is studied. The

UAV jammer is represented by *J*, the base station is represented by *B*, and *i* is the user *i*, $i \in \{1, ..., U\}$.

In the system model, the height of the base station is set to HB and its top is fixed at {0,0,5} of the three-dimensional motion space coordinate system. The motion speed of each agent is a vector. Because the resources of the equipment are limited, they all have only one antenna, and the way of downlink communication is frequency division multiple access (FDMA). The overall bandwidth is *W* Hz. When the interference is strongest, the UAV jammer carries out malicious interference and shield the entire bandwidth of the network [24]. The confined space includes a single base station, multiple ground users and one aerial jammer. Ground users and the jammer and are all regarded as smart agents, who aim to obtain the best moving strategies to optimize the rewards, that is, signal to interference plus noise ratio (SINR) [25].

The coordinates of the base station, any user *i*, and the jammer are expressed as $P_B = (0,0,HB)$, $P_{Ui} = (x_{Ui}, y_{Ui}, 0)$ and $P_J = (x_J, y_J, z_J)$, where $x_J \in (-100,100)$, $y_J \in (-100,100)$, $z_J \in (0,100)$, $y_{Ui} \in (0, 100)$ and $y_{Ui} \in (0, 100)$. The UAV jammer flies in three-dimensional space, and the ground user can only move in two-dimensional space. In this paper, the flight altitude of UAV shall not be lower than the top altitude of base station, ie. 5m. In the 3-D coordinate system of motion space, the unit velocity of the UAV is $v_J = (\vec{i}, \vec{j}, \vec{k})$, and the unit velocity of the ground user *i* is $v_{Ui} = (\vec{i}, \vec{j}, 0)$.

Then at time *t*, the spatial location update formula of the UAV jammer is:

$$P_J(t+1) = P_J(t) + v_J * dt, \quad dt = 0.1s \tag{1}$$

Similarly, the location update formula of user *i* is:

$$P_{Ui}(t+1) = P_{Ui}(t) + v_{Ui} * dt, \quad dt = 0.1s \tag{2}$$

In time slot *t*, the UAV jammer selects a velocity vector v_J to represent the direction of flight, and user *i* selects a velocity vector v_{Ui} to determine its moving direction.

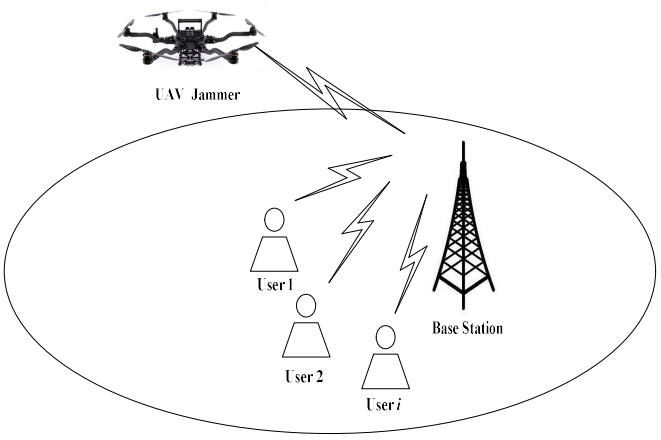


Fig. 1. Schematic diagram of anti-UAV jamming Game

Remark 1.

In this work, the speed of every agent is set as a vector, that is, the motion space of each agent is continuous. In addition, the motion space of UAV is truly three-dimensional, which is different from setting the flight altitude of UAV to a fixed value in most previous work.

The channel coefficient in the downlink communication is expressed as $h_{Bi} = \sqrt{d_{Bi}^{-\eta}}\,\widetilde{h_{Bi}}$, where d_{Bi} is the distance from the base station to user *i*, η is the path loss index, and $\widetilde{h_{Bi}}$ is a small-scale fading. Besides, the air-to-ground channel contains strong line-of-sight (LoS), reflected non-line-of-sight (NLoS) and small-scale fading. Small-scale fading is generally ignored because its influence is less than that of LoS and NLoS [26].

The air-to-ground channel loss is expressed as:

$$PL(J,i) = \begin{cases} \beta_{LoS} \left| d_{Ji} \right|^{-\alpha}, & \text{for } LoS \text{ link} \\ \beta_{NLoS} \left| d_{Ji} \right|^{-\alpha}, & \text{for } NLoS \text{ link} \end{cases} \tag{3}$$

$d_{Ji} = \sqrt{(x_i - x_J)^2 + (y_i - y_J)^2 + z_J^2}$ presents the distance from the jammer *J* to user *i*, and α presents the path loss index. β_{LoS} and β_{NLoS} is the additional attenuation factor of the LoS link and the NLoS link, respectively.

The probability of LoS connection P_{LoS} is affected by the elevation angle θ_i between user *i* and the UAV, which is expressed as:

$$P_{LoS} = \frac{1}{1 + \Phi \exp(-\Psi[\theta_i - \Phi])} \tag{4}$$

Where the S-curve parameters Φ = 150 and Ψ = 15 are commonly used in urban areas, and the angle is:

$$\theta_i = \frac{180}{\pi} \arcsin(\frac{z_J}{d_{Ji}}) \tag{5}$$

The probability of NLoS is $P_{NLoS} = 1 - P_{LoS}$. Therefore, the expected interference power received by user *i* is as follows:

$$I_{Ji} = p_J P_{LoS} \beta_{LoS} \left| d_{Ji} \right|^{-\alpha} + p_J P_{NLoS} \beta_{NLoS} \left| d_{Ji} \right|^{-\alpha} \tag{6}$$

Where P_J is the energy cost of jammer *J*, and the SINR received at user *i* is expressed as:

$$\Gamma_i = \frac{p_B d_{Bi}^{-\eta} \left| \widetilde{h_{Bi}} \right|^2}{I_{Ji} + \sigma^2} \tag{7}$$

Where P_B is the energy cost of base station and σ^2 is the variance of noise.

B. Problem Description

For the UAV jammer, it is a challenge to obtain complete observation information (COI) of users. The known observable information of UAV jammer is the user's location, expressed as the distance from the user to the base station, given by the following formula:

$$d_{Bi} = \sqrt{x_i^2 + y_i^2 + H_B^2}, \quad i \in \{1,...,U\} \tag{8}$$

At the same time, the information that the user constantly observes is the interference power received from the jammer. In order to describe the hierarchical interaction between the UAV and the ground user, we use a Stackelberg game G{{*J*, *i*}, $\{d_J, d_i\}$, $\{r_J, r_i\}$} to model the anti-jamming problem. In the developed game, the visionary UAV jammer is modeled as the leader, and the nearsighted user $i \in \{1, ..., U\}$ is modeled as the follower. The jammer takes the action $a_J \in A_J$ firstly, and

each user makes action $a_i \in A_i$ accordingly. The position of user *i* in the previous time slot is $(x_i, y_i, 0)$, the position in the current time slot is set to be $(x_i^{'}, y_i^{'}, 0)$, and the action is a_i. The position of the UAV jammer J in the previous time slot is (x_J, y_J, z_J), and in the current time slot is $(x_J^{'}, y_J^{'}, z_J^{'})$, action a_J, ie. $(x_J^{'}, y_J^{'}, z_J^{'}) = (x_J, y_J, z_J) + \alpha_J$. The instant reward of user *i* is expressed as:

$$r_i[\varphi(a_J), \ell(a_i)] = \frac{p_B d_{Bi}^{-\eta} \left|\widetilde{h_{Bi}}\right|^2}{I_{Ji} + \sigma^2} - C_U d_i \tag{9}$$

Where $\varphi(a_J) = (x_J^{'}, y_J^{'}, z_J^{'})$ is the current trajectory of the jammer, and the current trajectory of the user *i* is $\ell(a_i) = (x_i^{'}, y_i^{'}, 0)$; C_U is the movement cost per unit distance of users. The distance from UAV jammer *J* to user *i* is:

$$d_{Ji} = \sqrt{(x_J^{'} - x_i^{'})^2 + (y_J^{'} - y_i^{'})^2 + z_J^{'2}} \tag{10}$$

The current distance between the base station and user *i* is:

$$d_{Bi} = \sqrt{x_i^{'2} + y_i^{'2} + H_B^2} \tag{11}$$

The moving distance of user *i* per moment is:

$$d_i = \sqrt{(x_i^{'} - x_i)^2 + (y_i^{'} - y_i)^2} \tag{12}$$

The instant reward of the UAV jammer is determined by:

$$r_J[\varphi(a_J), \ell(a_i)] = \sum_{i=1}^{U} \frac{I_{Ji}}{p_B d_{Bi}^{-\eta} \left|\widetilde{h_{Bi}}\right|^2 + \sigma^2} - C_J d_J \tag{13}$$

Where C_J presents the unit power budget of the UAV, and its moving distance d_J of each step is as follows:

$$d_J = \sqrt{(x_J^{'} - x_J)^2 + (y_J^{'} - y_J)^2 + (z_J^{'} - z_J)^2} \tag{14}$$

The target of the optimization problem is to optimize the moving strategies of the UAV jammer *J* and each ground user *i* respectively, so as to obtain their long-term maximum cumulative rewards. In the optimization model, the moving distance in each step and the range of the moving space of the two entities are limited. Then, the mathematical description of the optimization problem is as follows:

$$\max_{a_J, a_i} R_J[\varphi(a_J), \ell(a_i)], \quad R_i[\varphi^*(a_J), \ell(a_i)] \tag{15}$$

$$s.t. \ |a_J| \le 1, \quad |a_i| \le 1, i \in \{1, ..., U\} \tag{16}$$

Where $R_J = \sum_{k=0}^{\infty} \gamma^k r_J(k)$ and $R_i = \sum_{k=0}^{\infty} \gamma^k r_i(k)$ represent the cumulative reward of *k* steps of the UAV and user *i*; γ represents the discount factor.

The dynamics and complexity of the anti-jamming scenario make the optimization problem face challenges, such as the inability to obtain all the state information, the need to know the state transition probability, and the convexity of the optimization problem. We give the following solutions to solve the optimization problem.

III. Stackelberg Game Based on Multi-agent Deep Reinforcement Learning

In this part, multi-agent deep reinforcement learning is used to optimize the anti-UAV jamming strategy in continuous action space. The problem is modeled as a Stackelberg game to describe the dynamic competition between the UAV jammer and ground users. The UAV first makes an interference strategy, and then ground users make real-time exploration to avoid the UAV's interference, so as to achieve the best communication effect. Considering the stateless transition probability of the model of the game, a model-free method of hierarchical multi-agent proximal policy optimization is used to approximate the Stackelberg equilibrium (SE), and its existence and uniqueness are proved.

A. Anti-UAV Jamming Strategy Based on Hierarchical Game Optimization

Firstly, the hierarchical game is used to model the above anti-UAV jamming optimization problem. The hierarchical game of interference countermeasure can be mathematically expressed as G = (*i*, *J*, A_i, A_J, r_i, r_J), $i \in N$, where A_i and A_J represent the policy space of users and jammers respectively; r_i and r_J represent the effect functions of user *i* and UAV, respectively. In this hierarchical game, UAV jammer is the leader and ground users are the followers. Each game participant independently carries out environmental perception and strategy update to optimize its effect function. The game models of leaders and followers are introduced below.

1) Leader Sub-game

Considering the dynamic strategy environment, the jamming process of UAV is modeled as a Markov sum-game (MSG). MSG is defined as 4 tuples $G_J = (S, A_J, r_J, P)$, where:

S is the observation;

A_J presents the action;

r_J [*s*, a_J] presents the immediate reward;

$P(\cdot|s, a_J)$ presents the state transition probability from the current state to the next state, provided that action a_J is selected in state $s \in S$.

The optimal interference trajectory of the UAV is expressed as

$$\varphi^*(a_J^*) = (x_{J0}, y_{J0}, z_{J0}) + a_J^{0*} + a_J^{1*} + ... + a_J^{t*} \tag{17}$$

Where (x_{J0}, y_{J0}, z_{J0}) is the initial coordinate, and $a_J^* = \arg\max_{a_J} R_J(a_J, a_i)$.

2) Follower Sub-game

The trajectory optimization problem of ground users is also described as a Markov sum-game, and the moving strategy of user *i* is affected by the state and the action of the jammer. The MSG of every user can be described as a 4-tuple $G_i = (S_J, A_i, r_i, P(\cdot \mid s_J, a_i))$, where:

S_J is the observation space;

A_i is the action space;

r_i is the immediate reward;

$P(\cdot \mid s_J, a_i)$ is the probability from the *i*-th state to the *i+1*-th state, suppose that action a_i is selected in state $s_J \in S_J$.

The optimal moving trajectory of ground user i $\ell^*(a_i)$ is expressed as:

$$\ell^*(a_i^*) = (x_{i0}, y_{i0}, 0) + a_i^{0*} + a_i^{1*} + ... + a_i^{t*} \quad (18)$$

in which $(x_{i0}, y_{i0}, 0)$ is the starting point of ground user *i*, and $a_i^* = \arg\max_{a_i} R_i(a_J, a_i)$.

For the hierarchical game framework of anti-interference countermeasures constructed above, the Stackelberg equilibrium can be used to analyze the properties of game *G*. The definition of Stackelberg is as follows.

Lemma 1. Nash equilibrium is the most commonly used concept of equilibrium solution. It constitutes a stable point of non-cooperative game. At this point, no participant can improve the income through unilateral strategy change.

Hierarchical game is a non-cooperative game. Based on this, the Stackelberg equilibrium solution of hierarchical game G can be obtained by solving the Nash equilibrium solution of leader game G_J and follower game G_i.

Given a hierarchical game, where the leader aims to get the maximum reward $R_J(a_J, b)$ and the *N* followers wants to maximize their own reward function $R_i(a_J, b_i, b_{-i})$ by choosing a_J, b_i from action space A_J and B_i, respectively. Then (*a**, *b**) constitutes a Stackelberg equilibrium of hierarchical game *G*. That is, for $i \in N$, the following formula holds

$$\begin{cases} R_J(a_J^*, b^*) \geq R_J(a_J, b^*), \\ R_i(a_J^*, b_i^*, b_{-i}^*) \geq R_i(a_J^*, b_i, b_{-i}^*) \end{cases} \quad (19)$$

Where the vector a_J^* represents the best jamming strategy of UAV jammer, a_J represents the jamming strategy other than the best strategy, and b^* represents the best anti-jamming strategy; b_i^* represents the best anti-jamming strategy of ground user *i*, b_i represents the strategies of user *i* other than the best anti-jamming strategy, and b_{-i}^* represents the best anti-jamming strategy of users other than user *i*.

B. Analysis of Equilibrium Solution For Hierarchical Game

In this subsection, the existence of Stackelberg equilibrium for the hierarchical game *G* is proved.

Theorem 1. In the hierarchical game with one jammer and *U* ground users, the optimal trajectory set $[\varphi^*(a_J), \ell^*(a_i)]$ is a Stackelberg equilibrium in the hierarchical game *G*.

Proof: For ground users, when the attack strategy of UAV jammer is determined, the follower game G_i constitutes a non-cooperative game. Then, in G_i, the participant's policy space Ai is a non-empty subset of Euclidean space, which is convex and compact. In addition, the effect function of each participant is a continuous concave function about action selection. According to reference [27], when the jamming strategy of a given jammer selects form b, there is at least one Nash equilibrium solution in G_i. Similarly, when the anti-jamming strategy of a given ground user selects form *a*, there is at least one Nash equilibrium solution in G_J. The stability strategy of leader game G_J and follower game G_i constitutes a Stackelberg equilibrium, that is

$$\begin{aligned} a^* &= \arg\max_a G_J(a, b) \\ b^* &= \arg\max_b G_i(a, b(a)) \end{aligned} \quad (20)$$

Then $(a^*, b^*(a^*))$ constitutes a Stackelberg equilibrium solution of hierarchical game *G*.

The proof is completed.

Remark 2.

The monotonic improvement performence of TRPO has been proved in [28]. PPO use the same algorithm architecture as TRPO, but put the constraints into the objective function, which can obtain the same data efficiency and reliability. So, PPO is also monotonically convergent [29]. In this paper, PPO is extended to the multi-agent domain, and the proposed HMAPPO has the same basic framework as PPO, so it has the same monotonic convergence performance as TRPO. Therefore, the HMAPPO algorithm is used to figure out the anti-UAV jamming game. With the continuous iteration of the algorithm, the global optimization of each agent can be approached infinitely, so as to achieve the approximate Stackelberg equilibrium solution.

C. Hierarchical Multi-Agent Proximal Policy Optimization Algorithm

In practical problems, the observation space and action space of the agent are generally large, and the state transition probability is usually difficult to evaluate. Therefore, the model-free deep reinforcement learning method is adopted to optimize the strategy by learning from the historical interactive data directly [30].

Reinforcement learning has made many successes in solving complex multi-agent challenges [31], [32]. For example, Alpha star's performance in StarCraft II reached the level of professional human players [33], and OpenAI five beat world champion II in Dota [34]. These successes are achieved using distributed architecture RL algorithms [35], such as TRPO.

The main contribution of TRPO algorithm is to approximate the complex function on a certain confidence region interval, and then solve the maximization of the approximate function. The core problem of the policy gradient (PG) algorithm is data deviation [36]. If the strategy is updated too far at a time, the next sampling will completely deviate, causing the strategy to be updated to a completely deviated position, thus forming a vicious circle. Therefore, the core idea of TRPO is to keep each policy update within a confidence zone to ensure a monotonous improvement of the policy. The problem of the TRPO algorithm is defined as follows:

$$\begin{aligned} &\max E_{s\sim\rho_{\theta_{old}}, a\sim\pi_{\theta_{old}}}[\frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)} A_{\theta_{old}}(s,a)] \\ &s.t. \bar{D}_{KL}^{\rho_{\theta_{old}}}(\pi_{\theta_{old}}, \pi_\theta) \leq \delta \end{aligned} \quad (21)$$

Where $\pi_\theta(a|s)$ is the new strategy that is constantly updated during the strategy training process; $\pi_{\theta_{old}}(a|s)$ is the old strategy; $A_{\theta_{old}}(s|a)$ is the advantage function of the value network output. The second behavior constraint of formula (21)

requires that the probability of the new strategy cannot be too far from the probability distribution of the old strategy to ensure the stability of the optimization.

1) PPO :

The PPO algorithm simplifies the TRPO algorithm. Its basic idea is to transform the constraints into simple ratio constraints of the old and new strategies, and a new objective function is proposed:

$$L(\theta) = E_t(\min\{r_t(\theta)A_t, clip[r_t(\theta),1-\varepsilon,1+\varepsilon]A_t\}) \quad (22)$$

where:

$$r_t(\theta) = \frac{\pi_\theta(a \mid s)}{\pi_{\theta_{old}}(a \mid s)} \quad (23)$$

is the importance ratio. The clip $(\cdot)$ is the lock value function, which limits the ratio $r_t(\theta)$ of the probabilities of the new and old strategies to [1-ε,1+ε] to ensure that each update will not fluctuate too much. The function of min $(\cdot)$ is to select a lower value among the two values as the result.

PPO inherits some advantages of TRPO, but it is easier to implement, and the utilization of samples is higher.

In PPO, the advantage function $A^{\pi_{old}}(s_t, a_t)$ is estimated by the generalized advantage estimation:

$$A_t = \delta_t + (\gamma\lambda)\delta_{t+1} + ... + (\gamma\lambda)^T \delta_T \quad (24)$$

Given the policy $\pi_{\theta_{old}}$ with parameters θ_{old} and corresponding trajectory τ, the optimal solution is obtained by updating θ with the gradient and updating ω by minimizing $J(\omega) = \hat{E}_t[(y_t - V_\omega(s_t))^2]$. Then, θ_{old} and $\bar{\omega}$ continue to iterate until models converge. The UAV and the ground users are all be seen as agents. Algorithm 1 gives the learning procedure for each agent, where L is the total number of iterations, K is the training epoch, and B is the mini-batch size.

2) HMAPPO :

In this paper, PPO is extended to multi-agent field. Different from PPO algorithm, the literature [37] for the first time proposed the use of global state information MAPPO to solve the dynamic game problem in the cooperative environment, and achieved excellent performance in a variety of environments.

The key idea of MAPPO is that the critic network of each agent can get the state observations of all other agents for centralized training and decentralized execution (CTDE). During training, the critic network can obtain the overall observations to guide the actor network, and the test only use local information to make mobile strategies.

Considering the dynamic game between the UAV jammer and ground users, a hierarchical multi-agent proximal policy optimization algorithm is proposed. In the proposed algorithm, the UAV jammer and ground users update their policy choices at different time scales, that is, the leader updates the jamming policy once in cycle k, and the follower updates the communication policy once in time slot t. Each cycle k contains Z time slots t, i.e. $k = Z*t$. For the convenience of expression, the policy selection of game participant i is extended to a dynamic strategy. The architecture of the actor network and critic network in HMAPPO is shown as Figure 2, in which CTDE is also used. The framework of HMAPPO is shown as

Figure 3. The pseudo-code of HMAPPO is presented as bellow.

Algorithm 1: Proximal Policy Optimization for UAV and ground users

Initialize V_ω, π_θ. Initialize $\pi_{\theta_{old}}$ with $\theta_{old} \leftarrow \theta$, and $V_{\bar{\omega}}$ with $\bar{\omega} \leftarrow \omega$

for iteration l =1, 2, ..., L

  for an episode t =1, 2, ..., T

    Agent takes action according to $\pi_{\theta_{old}}(a_t \mid s)$

    Get a reward r_t, and the next observation s_{t+1}

  end for

  Get a trajectory: $\tau = \{s_t, a_t, r_t\}_{t=1}^T$

  Compute advantages $\{A_t\}_{t=1}^T$

  Compute $y_t = V_{\bar{\omega}}(s_t) + A_t$

  Get data $\{s_t, a_t, y_t, A_T\}_{t=1}^T$

  for k = 1, 2, ..., K

    Disorganize and renumber the data

    for j = 0, 1, ..., $\frac{T}{B} - 1$

      Select B group of data $\{s_t, a_t, y_t, A_T\}_{i=1+Bj}^{B(j+1)}$

      Compute gradient:

$$\Delta\theta = \frac{1}{B}\sum_{i=1}^{B}\{\nabla_\theta f(r_i(\theta), A_i)\}$$

$$\Delta\omega = \frac{1}{B}\sum_{i=1}^{B}\{\nabla_\omega (y_i^u - V_\omega(s_i))^2\}$$

      Apply gradient ascent on θ using $\Delta\theta$ by Adam

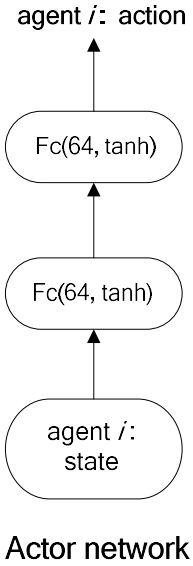
      Apply gradient descent on ω using $\Delta\omega$ by Adam

    end for

  end for

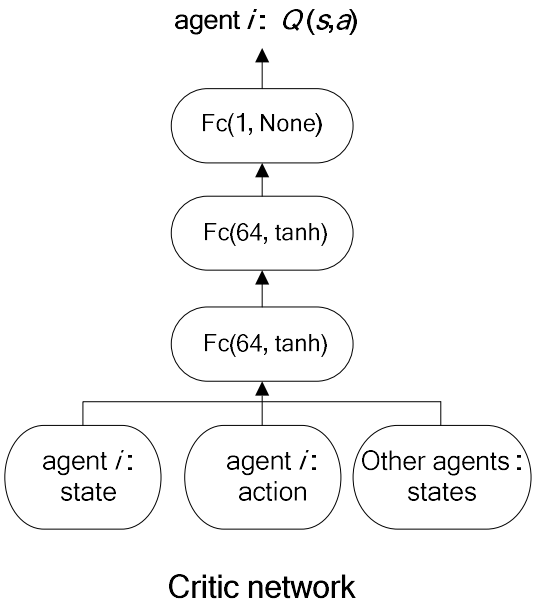
  Update $\theta_{old} \leftarrow \theta$ and $\bar{\omega} \leftarrow \omega$

end for



Actor network

(a)



(b)

Fig. 2. The Architecture of Actor Network and Critic Network in HMAPPO

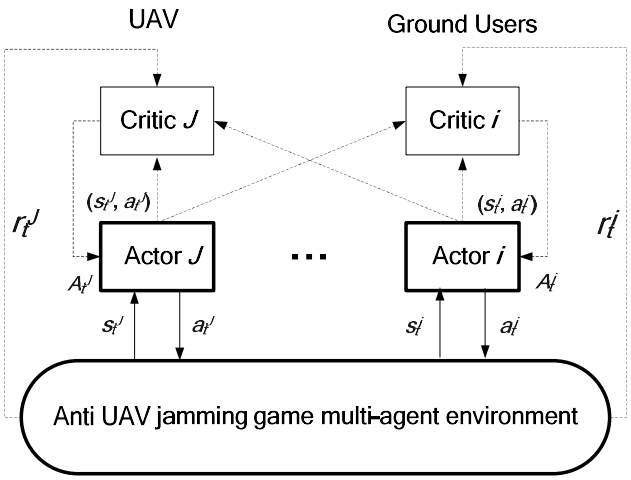


Fig. 3. Overview of The CTDE Framework of HMAPPO

IV. Simulation Results

In this part, we conduct simulation experiments and discuss the performance of the proposed algorithm in three scenarios. PPO and MAPPO are compared as baselines to verify the performance of the proposed hierarchical multi-agent proximal policy optimization algorithm in the anti-UAV jamming dynamic games.

A. Parameter Settings

The parameters of the simulation experiments are set as: the signal power of base station is $p_B = 100$ mW; the noise is $\sigma^2 = 1$ mW; the power budget of the UAV is $p_J = 30$ mW; the unit power budget of the jammer and ground user are $C_J = 1.22$ mW and $C_i = 1.13$ mW respectively. As in [39], the path loss indexes for jammer-to-user channel α and user-to-user channel η are set as 3 and 2 respectively; the attenuation factors β_{LoS} is set to 1 dB, and β_{NLoS} is set to 20 dB.

Algorithm 2: Hierarchical Multi-agent Proximal Policy Optimization

Initiate critic Q_ω and actor π with θ, $\forall$ agent.
Initiate the current policies $\pi_{\theta_{old}}$ with $\theta_{old} \leftarrow \theta$, and the target critic $Q_{\bar{\omega}}$ with $\bar{\omega} \leftarrow \omega$.
Initiate a memory buffer D
for iteration l =1, 2, ..., L
  s_1 = initial state
  for an episode t =0,1, 2, ..., T
    Agent executes action according to $\pi_{\theta_{old}}(a_t \mid s)$
    Get the reward r_t, and the next observation s_{t+1}
  end for
  Get a trajectory for each agent: $\tau = \{s_t, a_t, r_t\}_{t=1}^{T}$
  Compute $\{\hat{Q}(s_t, a_t)\}_{t=1}^{T}$
  Compute advantages $\{A_t(s_t, a_t)\}_{t=1}^{T}$
  Store data $\{[s_t, a_t, \hat{Q}(s_t, a_t), A(s_t, a_t)]_{t=1}^{N}\}_{t=1}^{T}$ into D
  for k = 1, 2, ..., K
    Disorganize and renumber the data
    for j = 0, 1, ..., $\frac{T}{B} - 1$
      Select B group of data D_j:
      $D_j = \{[s_t, a_t, \hat{Q}(s_t, a_t), A(s_t, a_t)]^N\}^{B(j+1)}$
      Compute gradient:
      if l **%** Z ==*0*:

$$\Delta\theta^J = \frac{1}{B}\sum_1^B \{\nabla_{\theta^J} f(r(\theta^J), A^J(s,a))\}$$

$$\Delta\omega^J = \frac{1}{B}\sum_1^B \{\nabla_{\omega^J} (\hat{Q}^J(s,a) - Q_{\omega^J}(s,a))^2\}$$

        Apply gradient ascent on θ^J and $\Delta\theta^J$ by Adam
      else:
        for i =1, 2, ..., N

$$\Delta\theta^i = \frac{1}{B}\sum_1^B \{\nabla_{\theta^i} f(r(\theta^i), A^i(s,a))\}$$

$$\Delta\omega^i = \frac{1}{B}\sum_1^B \{\nabla_{\omega^i} (\hat{Q}^i(s,a) - Q_{\omega^i}(s,a))^2\}$$

        Apply gradient ascent on θ^i and ω^i by Adam
        end for
    end for
  end for
  if l **%** Z ==*0*:
    Update $\theta_{old}^J \leftarrow \theta^J$ and $\bar{\omega}^J \leftarrow \omega^J$
  else:
    for i =1, 2, ..., N
      Update $\theta_{old}^i \leftarrow \theta^i$ and $\bar{\omega}^i \leftarrow \omega^i$
    end for
  Empty D
end for

TABLE I
PARAMETER SETTINGS

Parameter	Conversion from Gaussian and CGS EMU to SI [a]
learning rate	0.0003
γ	0.99
PPO_epoch	4
epsilon	0.2
GAE parameter	0.95
Max step per episode	64
Minibatch size	8
iteration	10000

To eliminate the randomness of the simulation experiment, we randomly generate 10 seeds for each algorithm. In addition to the single-run results specified, each algorithm carries out 10000 episodes, with each episode containing 64 steps. In this paper, the discount factor $\gamma = 0.99$, the GAE parameter is 0.95, epsilon is 0.2, minibatch size is 8, the epoch is 4, and learning rate $\varepsilon = 0.0003$.

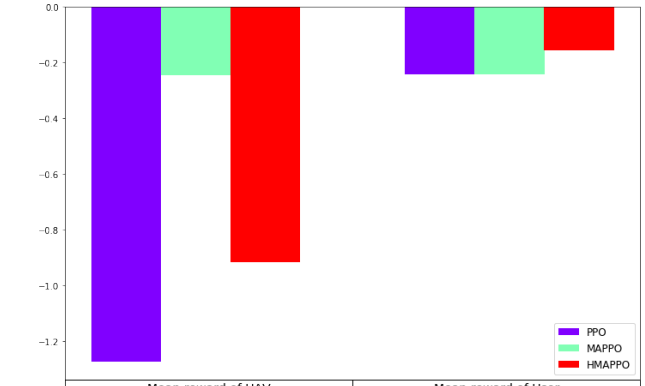
B. Result Analysis

Below, simulations are carried out under different number of agents, and results are mainly analyzed using the convergence performance.

The experiments in this paper are all run on the computer of Intel(R) Xeon(R) Silver 4210 with 40xCPU and NVIDIA GPU P4000 at 2.20 GHz, Ubuntu 18.04.1, Python. The networks are updated using the Adam optimizer [40]. The details of the simulation results are presented as follows.

1) Comparison of 3 algorithms when the number of agents is 2

We first examine the scenario with a UAV jammer and a ground user. The UAV explores the bounded room in the 3-D space and the user moves in the two-dimensional (2-D) space of the ground surface. Both the UAV and the user interact with the environment to optimize their policies, so as to get maximum reward. This scene can be seen as a zero-sum game.

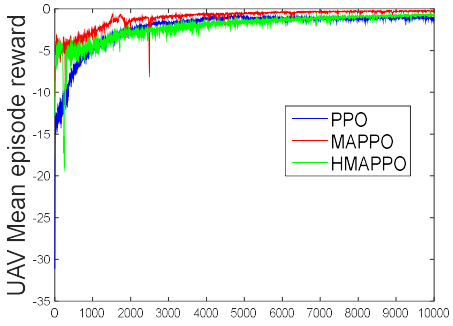


	Mean reward of UAV	Mean reward of User
PPO	-1.275	-0.241
MAPPO	-0.244	-0.242
HMAPPO	-0.916	-0.157

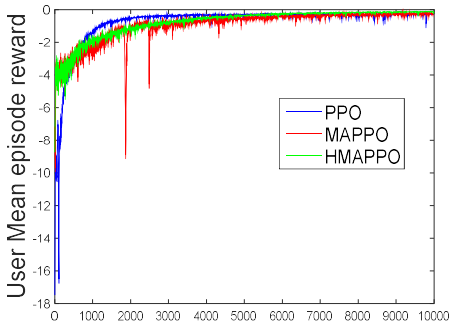
Fig. 4. Comparison of 3 algorithms when the number of agents is 2

The results are shown in Figure 4 regarding the final mean episode reward. The mean episode rewards of UAV based on HMAPPO is -0.916, which is better than PPO and is also comparable to MAPPO. In addition, the mean episode rewards of the ground user based on HMAPPO is -0.157, which is better than both PPO and MAPPO. This means that the jamming strategy and the anti-jamming strategy based on HMAPPO is better or at least comparable to the benchmark reinforcement algorithms.

Figure 5 shows the convergence curve of the mean episode reward for UAV and the ground user. It is clear that the learning curves all converges, which means that the model-free reinforcement learning method has the ability to optimize the strategy by learning from the historical interactive data. As the strategy iterates, both the UAV and the ground user can finally get the optimal strategy, and cannot gain a greater reward by changing their strategies alone. This constitutes a Nash equilibrium. Since UAV and the user adopt the hierarchical strategy, the Nash equilibrium can also be called Stackelberg equilibrium.



(a) UAV



(b) User

Fig. 5. Convergence curve of the mean episode reward for the UAV and the User

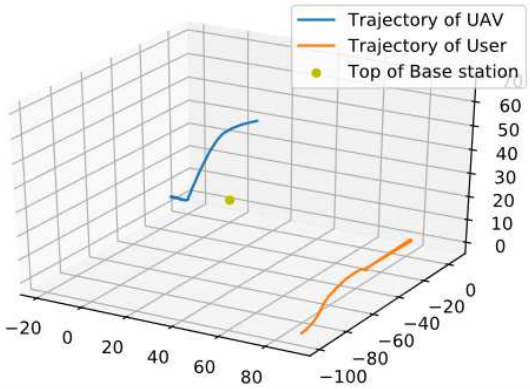


Fig. 6. Optimal trajectory of the UAV and the user based on HMAPPO in scenario 1

The mean time spent of three algorithms is also considered. As shown in Table 2, the mean time per episode of PPO, MAPPO and HMAPPO are 5.4282 s, 2.2543 s and 1.6918 s

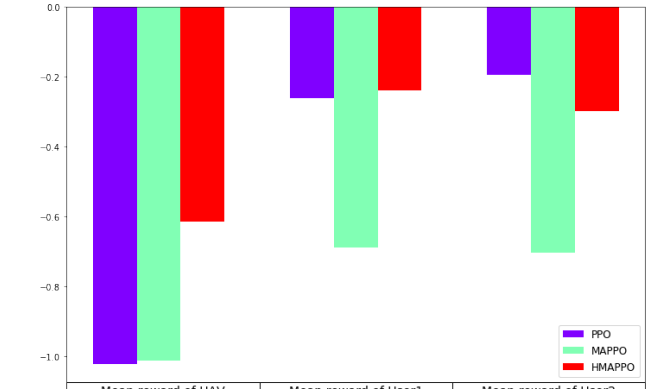
respectively, and HMAPPO's time cost is 68.8% less than PPO and is 24.9% less than MAPPO. This is because that the proposed hierarchical optimization strategy decouples the original hybrid game into two sub-games and updates the policy network of UAV and ground users by time slots. Then, the time cost is reduced.

The optimal trajectories of the UAV and the ground user are shown as Figure 6. It can be seen from the blue track in the figure that the UAV descends from the initial position to the lowest flight altitude (5m) and finally hovers over the base station to ensure the best interference effect. In order to avoid the impact of interference and achieve better communication effect as much as possible, the ground user lingers at a certain position from the base station. This constitutes a Nash equilibrium.

Thus, the well-trained HMAPPO has the ability to achieve the optimal jamming strategy and the optimal anti-jamming strategies, so as to approach the Stackelberg equilibrium. Besides, the performance of HMAPPO is comparable to the benchmark reinforcement algorithms with less time complexity. Therefore, HMAPPO has the highest convergence efficiency and the best comprehensive performance.

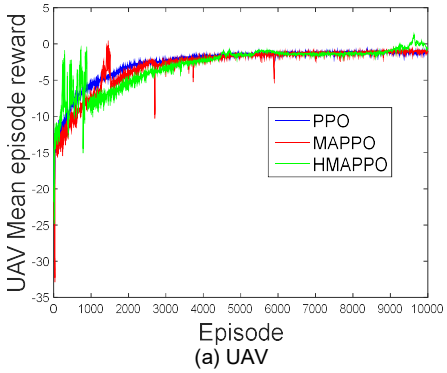
2) Comparison of 3 algorithms when the number of agents is 3

The second scenario has one UAV jammer and two ground users. UAV and two users interact with the environment, so as to get their own maximum reward. This scene can be seen as a hybrid game, in which the UAV needs to interfere with two ground users at the same time to get reward.

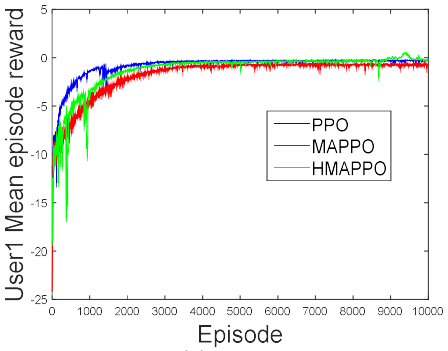


	Mean reward of UAV	Mean reward of User1	Mean reward of User2
PPO	-1.023	-0.262	-0.194
MAPPO	-1.012	-0.688	-0.703
HMAPPO	-0.614	-0.24	-0.298

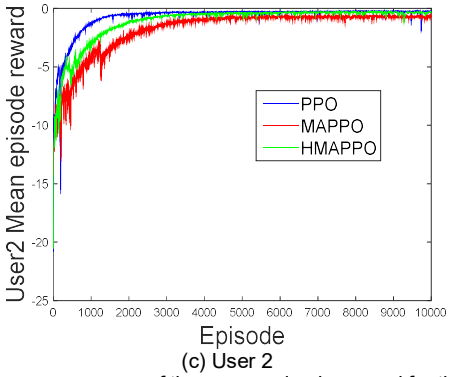
Fig. 7. Comparison of 3 algorithms when the number of agents is 3



(a) UAV



(b) User 1



(c) User 2

Fig. 8. Convergence curve of the mean episode reward for the UAV and 2 Users

As shown in Figure 7, the mean episode rewards of UAV based on HMAPPO is -0.614, which is bigger than both PPO and MAPPO. The mean episode rewards of the ground user 1 based on HMAPPO is -0.241, which is better than both PPO and MAPPO. In addition, the mean episode rewards of the ground user 2 based on HMAPPO is -0.298, which is bigger than MAPPO and is close to PPO. This means that the jamming strategy and the anti-jamming strategy based on HMAPPO is better than the benchmark reinforcement algorithms.

Figure 8 shows the convergence curve of the mean episode reward for the UAV and the two ground users. It is clear that the learning curves all converges. As the training progresses, the convergence curve of HMAPPO gradually catches up with or exceeds the convergence curve of PPO and MAPPO. With the continuous iterative update of the strategy, the UAV and two ground users gradually obtain the optimal strategy, and cannot gain a greater reward by changing their strategies alone. This constitutes a Stackelberg equilibrium.

The average training time of the four algorithms is also considered. As shown in Table 2, the mean time per episode of PPO, MAPPO and HMAPPO are 7.7787 s, 8.9842 s and 6.4820 s respectively, and HMAPPO's time cost is 16.6% less than PPO and is 27.8% less than MAPPO. This is also because that the proposed hierarchical optimization strategy decouples the original hybrid game into three sub-games and updates the policy network of UAV and two ground users by time slots, so that the cost time is reduced.

Figure 9 shows the optimal trajectory of the UAV and the two ground users. It can be seen that the UAV descends from the initial position to the lowest flight altitude and hovers in the upper position between the two ground users, thus forming the same interference effect on the two ground users at the same

time. The two ground users have learned to wander on different sides of the UAV and approach the base station on time, so as to achieve the maximum cumulative reward. This constitutes a Stackelberg equilibrium.

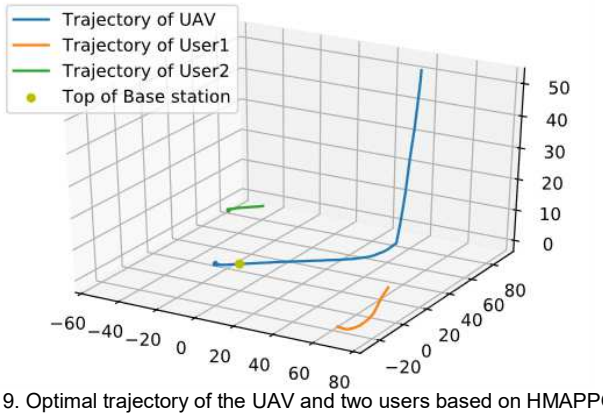
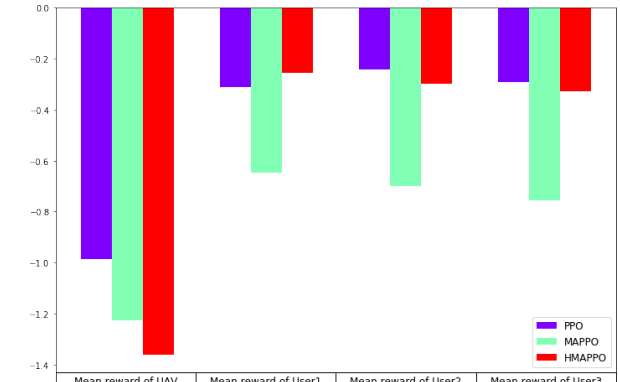


Fig. 9. Optimal trajectory of the UAV and two users based on HMAPPO in scenario 2

Thus, the well trained HMAPPO algorithm has the ability to achieve the optimal jamming strategy and the optimal anti-jamming strategies, so as to approach the Stackelberg equilibrium. The proposed algorithm has better performance than the benchmark reinforcement algorithms in this three-agent scenario. Therefore, HMAPPO has the best comprehensive performance.

3) Comparison of 3 algorithms when the number of agents is 4

The third scenario has one UAV jammer and three ground users. UAV and three users interact with the environment, so as to get their own maximum reward. This scene is also a hybrid game, in which the UAV needs to interfere with three ground users at the same time to get reward.



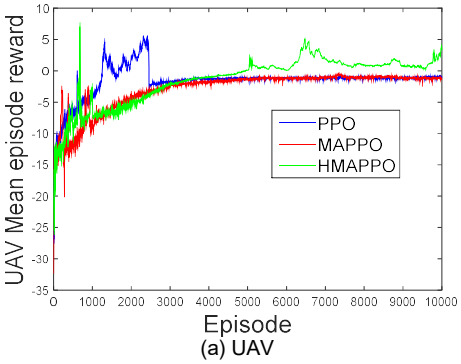
	Mean reward of UAV	Mean reward of User1	Mean reward of User2	Mean reward of User3
PPO	-0.985	-0.31	-0.241	-0.293
MAPPO	-1.228	-0.646	-0.701	-0.755
HMAPPO	-1.363	-0.255	-0.3	-0.329

Fig. 10. Comparison of 3 algorithms when the number of agents is 4

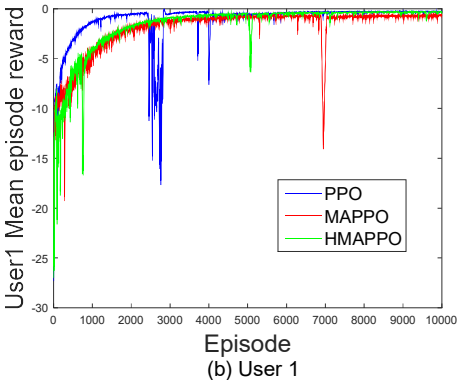
As shown in Figure 10, the mean episode rewards of UAV based on HMAPPO is 3.347, which is bigger than both PPO and MAPPO. The mean episode rewards of the ground user3 based on HMAPPO is -0.281, which is also better than both PPO and MAPPO. In addition, the mean episode rewards of the ground user1 based on HMAPPO is -0.448, which is bigger than MAPPO and is close to PPO. This shows that the jamming strategy based on HMAPPO is better than the benchmark reinforcement algorithms, and the anti-jamming strategy based on HMAPPO is better than or comparable to the benchmark reinforcement algorithms.

Figure 11 shows the convergence curve of the mean episode reward for the UAV and three ground users. It is clear that the learning curves all converges. In (a) and (b) of Figure 11, with the training progresses, the convergence curve of HMAPPO gradually exceeds the convergence curve of PPO and MAPPO. With the continuous iterative update of the strategy, the UAV and three ground users gradually achieve the optimal strategy, and cannot gain a greater reward by changing their strategies alone. This constitutes a Stackelberg equilibrium.

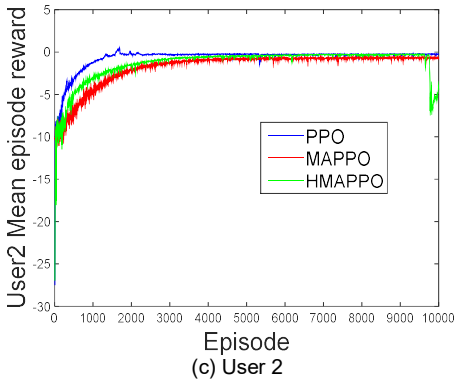
The average training time of the four algorithms is also considered. As shown in Table 2, the mean time per episode of PPO, MAPPO and HMAPPO are 11.9110 s, 12.3067 s and 9.2306 s respectively, and HMAPPO's time cost is 22.5% less than PPO and is 24.9% less than MAPPO. The original hybrid game into four sub-games and updates the policy network of UAV and three ground users by time slots. Then, the cost time is reduced.



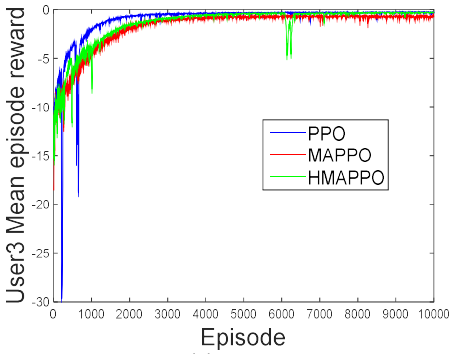
(a) UAV



(b) User 1



(c) User 2



(d) User 3

Fig. 11. Convergence curve of the mean episode reward for the UAV and 3 Users

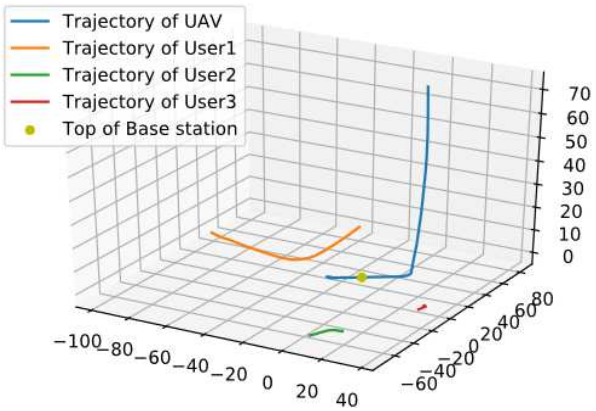


Fig.12. Optimal trajectory of the UAV and three users based on HMAPPO in scenario 3

TABLE II
MEAN TIME PER EPISODE

	2agent	3agent	4agent
PPO	5.4282	7.7787	11.9110
MAPPO	2.2543	8.9842	12.3067
HMAPPO	**1.6918**	**6.4820**	**9.2306**

Figure 12 shows the optimal trajectory of the UAV and the three ground users. It can be seen that the UAV descends from the initial position to the lowest flight altitude, hovering over the base station, thus forming the same interference effect on the three ground users at the same time. The three ground users have learned to wander on three different sides of the UAV and approach the base station on time, so as to achieve their maximum cumulative reward. This also constitutes a Stackelberg equilibrium.

Therefore, the overall performance of HMAPPO is competitive with other benchmark enhancement algorithms in this scenario.

To sum up, the well trained hierarchical multi-agent proximal policy optimization algorithm has the ability to obtain the optimal jamming strategy and the optimal anti-jamming strategies, so as to approach the Stackelberg equilibrium. In addition, the good convergence efficiency and comprehensive performance of HMAPPO is validated in three anti-jamming scenarios with different agents.

Remark 3.

The proposed algorithm decouples the hybrid game into sub-Markov games, and the hierarchical update strategy effectively improves the convergence efficiency of the algorithm. In three different multi-agent scenarios, HMAPPO reduces the training time by 16.6% ~ 68.8% compared with other benchmark algorithms.

V. CONCLUSIONS

To solve the problem of the malicious jamming of UAV to ground users in the downlink communications, this paper proposes a hierarchical multi-agent proximal policy optimization strategy to optimize the mobile trajectories of the UAV jammer and ground users. The advantages of the stable promotion strategy of PPO algorithm in the trust region are extended to the multi-agent field, so that the strategies of ground users and UAV can adapt to each other and approach the Stackelberg equilibrium. The high-dimensional joint action space is decoupled by hierarchical decision-making at different agent levels and the actor-critic networks of the UAV jammer and ground users are updated at different time scales. Compared with centralized decision-making, it greatly reduces the dimension of action space. Finally, simulation experiments show that the proposed hierarchical strategy based on hierarchical multi-agent reinforcement learning can achieve the same good monotonic convergence as the benchmark strategy with lower time complexity. The future work will study how to solve the distributed training and distributed execution of assisted communication game in the larger scale multi-agent environment.

Acknowledgements This work is supported by Hainan Provincial Natural Science Foundation of China (2019CXTD400), the National Key Research and Development Program of China (2018YFB1404400), the Scientific Research Fund Project of Hainan University (KYQD(ZR)-21007) and the Scientific Research Fund Project for Youth Teachers of Hainan University (HDQN202103).

Data Availability Statements The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

REFERENCES

[1] Kiss A K, Avedisov S S, Bachrathy D, et al. On the global dynamics of connected vehicle systems[J]. Nonlinear Dynamics, 2019, 96: 1865–1877.

[2] Zhang J, Zhang T, Yang Z, et al. Altitude and number optimisation for UAV-enabled wireless communications[J]. IET Communications, 2020, 14(8): 1228-1233.

[3] Xu X, Duan H, Guo Y, et al. A Cascade Adaboost and CNN Algorithm for Drogue Detection in UAV Autonomous Aerial Refueling[J]. Neurocomputing, 2020, 408(3): 121-134.

[4] Wang Q, Zhang W, Liu Y, et al. Multi-UAV Dynamic Wireless Networking with Deep Reinforcement Learning[J]. IEEE Communications Letters, 2019, 23(12): 2243-2246.

[5] Du Y, Li F, Zandi H, et al. Approximating Nash Equilibrium in Day-ahead Electricity Market Bidding with Multi-agent

Deep Reinforcement Learning[J]. Journal of Modern Power Systems and Clean Energy, 2021, 9(3): 534-544.
[6] Sheng Z, Tuan H D, Nasir A A, et al. Secure UAV-Enabled Communication Using Han–Kobayashi Signaling[J]. IEEE Transactions on Wireless Communications, 2020, 19(5): 2905-2919.
[7] Sadana U, Reddy P V, Zaccour G. Feedback Nash Equilibria in Differential Games with Impulse Control[J]. European Journal of Operational Research, 2021, 295(2): 792-805.
[8] Nagurney A. Supply chain game theory network modeling under labor constraints: Applications to the Covid-19 pandemic[J]. European Journal of Operational Research, 2021, 293(3): 880-891.
[9] Javidi M M. Feature selection schema based on game theory and biology migration algorithm for regression problems[J]. International Journal of Machine Learning and Cybernetics, 2021, 12(2): 303-342.
[10] F Parise, Gentile B, Lygeros J. A distributed algorithm for average aggregative games with coupling constraints[J]. IEEE Transactions on Control of Network Systems, 2020, 7(2): 770-782.
[11] Gwa B, Gt A, Jd B, et al. Distributed Reinforcement Learning Algorithm of Operator Service Slice Competition Prediction Based on Zero-Sum Markov Game - ScienceDirect[J]. Neurocomputing, 2021, 439(7): 212-222.
[12] Zhao S, Liang H, Ahn C K, et al. Observer-based adaptive neural optimal control for discrete-time systems in nonstrict-feedback form[J]. Neurocomputing, 2019, 350(20): 170-180.
[13] Liu Y C, Huang C Y. DDPG-Based Adaptive Robust Tracking Control for Aerial Manipulators with Decoupling Approach[J]. IEEE Transactions on Cybernetics, 2021, 99: 1-14.
[14] Zhong S, Tan J, Dong H, et al. Modeling-Learning-Based Actor-Critic Algorithm with Gaussian Process Approximator[J]. Journal of Grid Computing, 2020, 18(2): 181-195.
[15] Watkins C, Christopher J, Dayan P. Q-learning[J]. Machine Learning, 1992, 8(3): 279-292.
[16] Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning[J]. Nature, 2015, 518(7540):529-533.
[17] Li K, Ni W, Wei B, et al. Onboard Double Q-Learning for Airborne Data Capture in Wireless Powered IoT Networks[J]. IEEE Networking Letters, 2020, 2(2): 71-75.
[18] Schulman J, Wolski F, Dhariwal P, et al. Proximal policy optimization algorithms[J]. arXiv preprint arXiv:1707.06347,2017.
[19] Hu H, Wang Q L. Proximal policy optimization with an integral compensator for quadrotor control[J]. Frontiers of Information Technology & Electronic Engineering, 2020, 21(5): 777-795.
[20] Wang L, Wang M, Yue T. A fuzzy deterministic policy gradient algorithm for pursuit-evasion differential games[J]. Neurocomputing, 2019, 362(14): 106-117.
[21] Schulman J, Levine S, Moritz P, et al. Trust Region Policy Optimization[J]. Computer Science, 2015, 37: 1889-1897.
[22] Wu Y, Mansimov E, Grosse R B, et al. Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation[C]//Advances in Neural Information Processing Systems, 2017: 5279-5288.
[23] Heess N, Dhruva T B, Sriram S, et al. Emergence of Locomotion Behaviours in Rich Environments[C]. arXiv preprint arXiv: 2103.01955, 2021.
[24] A. Mpitziopoulos, D. Gavalas, C. Konstantopoulos, et al. A survey on jamming attacks and countermeasures in WSNs[J]. IEEE Communications Surveys & Tutorials, 2009, 11(4): 42–56.
[25] WU Guanhan, JIA Weimin, ZHAO Jianwei, et al. MARL-based Design of Multi-Unmanned Aerial Vehicle Assisted Communication System with Hybrid Gaming Mode[J]. Journal of Electronics & Information Technology, 2021, 43: 1-11.
[26] Feng Q, Mcgeehan J, Tameh E K, et al. Path Loss Models for Air-to-Ground Radio Channels in Urban Environments[C]// IEEE Vehicular Technology Conference. IEEE, 2006.
[27] ROSE N J B. Existence and uniqueness of equilibrium points for concave n-person games[J]. Journal of the Econometric Society, 1965, 33(3): 520-534.
[28] Schulman J, Levine S, Moritz P, et al. Trust Region Policy Optimization[J]. Computer Science, 2015: 1889-1897.
[29] Schulman J, Wolski F, Dhariwal P, et al. Proximal Policy Optimization Algorithms[J]. arXir preprint arXiv:1707.06347, 2017.
[30] M. Hausknecht and P. Stone. Deep recurrent Q-learning for partially observable MDPs[J]. In AAAI Fall Symposium - Technical Report, 2015,15(6): 1–9.
[31] Hengyuan Hu and Jakob N Foerster. Simplified action decoder for deep multi-agent reinforcement learning[C]. arXiv:1912.02288, 2020.
[32] Shariq Iqbal, Christian A, Schröder de Witt, et al. Ai-qmix: Attention and imagination for dynamic multi-agent reinforcement learning[C]. CoRR, abs/2006.04222, 2020.
[33] Oriol Vinyals, Igor Babuschkin, M Wojciech Czarnecki, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning[J]. Nature, 2019, 575: 350–354.
[34] Christopher Berner, Greg Brockman, Brooke Chan, et al. Dota 2 with large scale deep reinforcement learning[C]. CoRR, abs/1912.06680, 2019.
[35] Lasse Espeholt, Hubert Soyer, Remi Munos, et al. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures[C]. In International Conference on Machine Learning, pages 1407–1416, 2018.
[36] Sf A, Xz A, Zs A. Reinforced Knowledge Distillation: Multi-class Imbalanced Classifier Based on Policy Gradient Reinforcement Learning[J]. Neurocomputing, 2021, 439(7): 212-222.
[37] Yu C, Velu A, Vinitsky E, et al. The Surprising Effectiveness of MAPPO in Cooperative, Multi-Agent Games[J]. arXiv:2111.01100, 2021.
[38] Ryan Lowe, Yi Wu, Aviv Tamar, et al. Multi-agent actor-critic for mixed cooperative-competitive environments[C]. Neural Information Processing Systems (NIPS), 2017.
[39] A. Hourani, S. Kandeepan, and A. Jamalipour. Modeling air-to-ground path loss for low altitude platforms in urban environments[C]. in Proc. IEEE Globecom, 2014, 2898–2904.
[40] D. Kingma and J. Ba. Adam: A method for stochastic optimization[C]. arXiv preprint arXiv:1412.6980, 2014.