Multi-Agent Reinforcement Learning based Joint Cooperative Spectrum Sensing and Channel Access for Cognitive UAV Networks

Weiheng Jiang, Member, IEEE, Wanxin Yu

Abstract—Designing clustered unmanned aerial vehicle (UAV) communication networks based on cognitive radio (CR) and reinforcement learning can significantly improve the intelligence level of clustered UAV communication networks and the robustness of the system in a time-varying environment. Among them, designing smarter systems for spectrum sensing and access is a key research issue in CR. Therefore, we focus on the dynamic cooperative spectrum sensing and channel access in clustered cognitive UAV (CUAV) communication networks. Due to the lack of prior statistical information on the primary user (PU) channel occupancy state, we propose to use multi-agent reinforcement learning (MARL) to model CUAV spectrum competition and cooperative decision-making problem in this dynamic scenario, and a return function based on the weighted compound of sensing-transmission cost and utility is introduced to characterize the real-time rewards of multi-agent game. On this basis, a time slot multi-round revisit exhaustive search algorithm based on virtual controller (VC-EXH), a Q-learning algorithm based on independent learner (IL-Q) and a deep Q-learning algorithm based on independent learner (IL-DQN) are respectively proposed. Further, the information exchange overhead, execution complexity and convergence of the three algorithms are briefly analyzed. Through the numerical simulation analysis, all three algorithms can converge quickly, significantly improve system performance and increase the utilization of idle spectrum resources.

Index Terms—clustered CUAV communication networks, MARL, CR, VC-EXH algorithm, IL-Q algorithm, IL-DQN algorithm.

I. INTRODUCTION

CLUSTERED unmanned aerial vehicle (UAV) communication networks technology, which is one of the key supporting technologies of clustered UAV system, has received extensive attention and in-depth research recently [1][2]. Due to the particularities of clustered UAV applications, they generally form a self-organizing network which is in a complex electromagnetic interference environment [3]. The solution of allocating fixed spectrum resources for clustered UAV communication networks will lead to poor system robustness consequently. In addition, the traditional fixed spectrum licensing model has caused an abnormal shortage of spectrum resources and the allocated resources have not been fully utilized. Besides, the cognitive radio (CR) technology has spectrum sensing capabilities and spectrum dynamic adaptive capabilities [4][5] which can solve this problem. Hence, it has become one of the most potential solutions for clustered UAV communication networks to design an intelligent clustered UAV communication system with the help of CR. Nevertheless, designing an efficient and intelligent spectrum sensing and channel access mechanism for clustered CUAV communication networks is still an open issue at present, which is the focus of this paper.

In researches of clustered CUAV communication networks, many scholars have conducted various discussions and researches on different scenarios and specific problems. To overcome the challenges of the mobility of nodes and the dynamicity of network topology by using the traditional centralized cooperative spectrum sensing (CSS) methods, Nie et al. [6] propose a clustering-based distributed CSS scheme to enhance the spectrum detection performance of clustered CUAV networks. In [7], a dynamic cluster head selecting algorithm based on energy, mobility, distance, and node correlation degree for the post-earthquake rescue of CUAV ad hoc network is proposed, which declines collisions and the cluster head changing ratio in a clustered UAV network. In [8], aiming at the problem of lack of spectrum resources and huge energy expenditure in future mobile communication systems, a two-hop cognitive network with the transmitter as a special radio frequency (RF) source is proposed to enhance the secrecy performance. Feng et al. [9] focus on solving the rendezvous problem, and present a multi-channel cognitive MAC protocol (CogMOR-MAC), which can adapt to primary user (PU) activities environment more effectively and offer a reliable data communication services. It can be found that the above work is based on traditional optimization theories and probabilistic analysis tools to model the system. In order to further enhance the adaptabilities of the system in the complex environment, the modeling methods represented by reinforcement learning (RL) have recently been proposed. In [10], the distributed task decision-making problem when clustered UAV network performs remote sensing is discussed, where the UAVs are categorized to two clusters. And authors propose a optimal task allocation strategy based on distributed RL and discuss the convergence of the algorithm. In order to realize the collaboration of multi-UAV tasks in the cluster, a task scheduling algorithm based on RL is proposed in [11], which solves the channel allocation problem of the UAV cluster.

For the spectrum sensing and channel access problems based on RL, there are currently many studies. In [12], a multiple secondary users (SUs) local cooperative sensing
mechanism based on Q-learning is proposed. The SUs use Q-learning to sense spectrum and cooperate with neighbors, and make decisions based on their own and neighbors’ statistical data to maximize the amount of available spectrum for secondary use. With the purpose of reducing the cooperative overhead and improving detection performance under multi-path and related shadow fading channels, a new cooperative sensing method based on RL and a centralized structure are proposed [13]. In [14], a RL enabled CSS scheme consisting of the Q-learning and the discounted upper confidence bound (D-UCB) is used to achieve less number of attempts, higher detection probability, and lower call block rate. Aiming at the problem of data fusion between users with different credibilities in CR networks, Zhang et al. [15] propose a distributed CSS method based on RL, which can effectively identify malicious users and improve the intelligence and stability of the system. The above papers all use tabular RL methods in problem modeling and solving, i.e., storing the Q-value based on the Q-table whose size is determined by the state and action. When there are complex environments in actual applications, such as huge state space and action space, it is difficult to deal with the problems and lead to dimensional disasters. The applicability of tabular RL will be poor and system performance degradation will be serious. As a result, deep RL (DRL) or deep Q-learning network (DQN) [16]–[18] based on strategy space approximation with strong state action representation ability is proposed in further researches. In [19], a multi-agent deep reinforcement learning (MARL) method is adopted to realize CSS in CR networks by implementing DQN with UCB-H to improve the exploration efficiency and achieve faster convergence speed and better reward performance. Also based on DRL, in order to maximize the number of successful transmissions without interrupting the PUs, Li et al. [20] formulate the dynamic spectrum sensing and aggregation problem as a POMDP and propose a DQN framework. In [21], the CSS problems in CR networks under correlated fading is analyzed and the distributed DRL method is adopted to learn the optimal CSS strategy. Further, the authors decompose the problem into a max-plus problem to speed up convergence and improve the reward performance in large networks.

According to the above-mentioned researches, the design of clustered UAV communication networks based on CR can significantly enhance the intelligence level and anti-interference ability of the networks. Furthermore, it can improve the ability of CR networks to cope with time-varying environments by modeling spectrum sensing and channel access with the help of RL or DRL (DQN). However, the current researches on clustered UAV mainly focus on the optimal design of transmission strategies, including access protocol mechanisms, channel allocation and cluster management [22]–[25], etc. How to apply RL to model and design the joint cooperative spectrum sensing and channel access for the clustered CUAV system, and to further enhance the robustness of clustered CUAV communication networks in time-varying channel environments has not been discussed yet.

Hence, we discuss a dynamic cooperative spectrum sensing and channel access problem in clustered CUAV communication networks. Different from most studies based on RL which are only for a single SU scenario and user reward only include the utilization of idle spectrum resources, the proposed problem is modeled as a MARL problem. In the design of the decision-making reward function of the agent, a return function based on the weighted compound of sensing-transmission cost and utility to characterize the real-time reward of multi-agent game is introduced. Moreover, the Q-learning and DQN algorithm based on MARL is also adopted. Specifically, the main contributions of this paper can be summarized as follows:

- To solve the dynamic cooperative spectrum sensing and channel access optimization problem in clustered CUAV communication networks, we introduce MARL (i.e., Markov game) and propose a clustered CUAV channel exploration and utilization protocol based on sensing-fusion-transmission.
- We design three algorithm to solve the proposed problem respectively which are the time slot multi-round revisit exhaustive search algorithm based on virtual controller (VC-EXH), the Q-learning algorithm based on independent learner (IL-Q) and the deep Q-learning algorithm based on independent learner (IL-DQN).
- Our simulation results show that the VC-EXH algorithm can achieve the best performance, but the complexity is extremely high. The IL-DQN algorithm can achieve good performance in different scenarios and has good scalability. The IL-Q algorithm has the worst performance which is only suitable for small-scale application scenarios.

The rest of this paper is organized as follows. In Section II, we formulate the network model, protocol mechanism and optimization design problem. The theory of MARL is introducing to model the proposed problem in Section III. In Section IV, some MARL-based dynamic spectrum sensing and channel access algorithms are proposed for the clustered CUAV networks. Simulation results and analyzes are presented in Section V, and conclusions are drawn in Section VI.

II. SYSTEM MODEL

A. Network Model

Here is a coexistence network of cognitive radio system and PU system as shown in Fig.1. The cognitive users are randomly distributed CUAVs in the airspace and the PUs are transmitters or hostile jammer on the target frequency band of CUAVs. In this network, the clustered UAV communication network uses CR technology to enhance the anti-jamming capabilities and survivability in a hostile environment, i.e., CUAVs sense spectrum to find idle spectrum resources that are not using by PUs. And CUAVs obtain idle spectrum resources to transmit task data through dynamic access technology. Specifically, there are N CUAVs and M PUs.

Further, we assume that the PU network working on the target frequency band is deployed with different services, such as mobile communications, radar or other dynamic spectrum occupancy services. Different PUs are assigned orthogonal channels and have different channel bandwidths. The channel bandwidth of the PU channel $m$ is $B_m$. In addition, the PU
services are bursty which means that the channel occupancy shows dynamic time-varying characteristics, so that a Markov process with the state transition probability \( (\alpha_m, \beta_m) \) of the channel occupancy for any PU \( m \) is defined, as shown in Fig. 2.

**B. Channel Sensing and Access Protocol**

As previously mentioned, CUAVs explore idle spectrum resources based on spectrum sensing and utilize the sensed idle channels through dynamic channel access. Owing to the system cost and load constraints, a CUAV can only sense and access a PU channel at any time, so the exploration and utilization of the channels by multi-CUAV may conflict. As a result, we introduce three basic assumptions here. First, the clustered CUAV network has a dedicated common control channel (CCC), i.e., \( \text{CH}_0 \) in Fig. 3, where the sensing information and selection information can be sharing. Second, the clustered CUAV system has the ability to synchronize the entire network, so the CUAVs sense and access CCC and PU channel spectrum resources are synchronized in time. The third is that CUAVs access CCC based on time division multiple access (TDMA). Based on this, the channel sensing and access process of the clustered CUAV is shown in Fig. 3. The exploration and utilization of PU channels by CUAVs is divided into frames by period \( \tau \). A frame includes three minislots which are sensing minislots, sensing information exchange and cooperation minislot (i.e., cooperation minislot) and channel access minislot. CUAVs (i.e., cognitive users) first select the channel which will be sensed and accessed. Then, all users switch to the channel to sense according to selections, i.e., entering the sensing minislot. Next, each user broadcasts sensing information on CCC in turn which means to enter the cooperation minislot, and users that do not broadcast information receive other users’ sensing information to fuse and determine whether to access channel based on the fusion sensing decision.

It is important to point out that, we have made the assumption that the messages exchanging over the CCC is reliable enough thus without any packet loss. And this is the precondition for the cognitive based cluster UAV networks to reliability and effectively work over the time-varying environments, i.e., the anti-jamming spread spectrum communication is adopted by the CUAVs to work over the CCC. In addition, for the cooperative spectrum sensing, we assume that the same fusion rule is used by all CUAVs, such as “K-out-of-N” rule or “AND” rule [46], [47], which promise that all CUAVs formulate the consistent view about the status of PU channels, i.e., busy or idle. While for the situation that all CUAVs cannot formulate the same view, it is an interesting problem and desired further research but not included herein. In order to simplify the problem, we further assume that the TDMA scheme is adopted by the CUAVs shared the same PU channels, i.e., for each sensed idle PU channel, it is equally divided and shared by multiple cooperative CUAVs. Obviously, the more CUAVs participate the cooperative sensing for each PU channel, the higher reliability of the spectrum sensing result [48], however, less idle channel resources are allocated to a single CUAV. Therefore, the CUAVs should consider the conflict and cooperation trade-off between them in making decision. Also, we can note that, the proposed methods in this paper can be easily extended to the scenario that OFDMA or CDMA scheme is adopted by the CUAVs to share the sensed idle PU channels, and this discussion is omitted herein.

For the considered network model and its protocol mechanism, it can also be applied to multi-CUAV cluster scenarios. Here the exploration and utilization of PU channel resources

---

1 The mode can also use other multiple access methods, such as frequency division multiple access (FDMA), code division multiple access (CDMA), etc.
is based on a cluster rather than a single CUAV. A CUAV cluster completes the cluster information interaction based on the resources received by competition. Moreover, the discussed models and methods can also be extended to scenarios where CUAVs have broadband sensing access capabilities. Here CUAVs can combine multiple PU channel frequency bands to achieve broadband spectrum sensing and access, however, detailed models and methods will not be repeated here.

C. Mathematical Model

From the perspective of the PU network, different PU channels have different channel bandwidths and noise power spectra, and different primary users’ occupation of the home channel also presents significantly different statistical characteristics, specifically in channel occupation probability and channel state transition probability. Therefore, cognitive users who choose different PU channels to access will achieve significantly different sensing idle channel access performance. From the perspective of the cognitive network, different cognitive users will achieve different sensing performances when sensing spectrum because different cognitive-primary user pairs have different spatial position relative relationships and channel environments. In addition, owing to the cost and load constraints of CUAVs, it is difficult to perform broadband spectrum sensing and achieve highly reliable sensing performance by a single CUAV.

As the previous analysis pointed out, although multi-user cooperative spectrum sensing can improve sensing performance while reducing sensing energy consumption, multi-user spectrum competition will result in limited spectrum resources available to users. What’s more, the communication network is the key support module of clustered UAV system whose performance will directly affect the execution efficiency of the UAV missions. Hence, it is an important issue how to optimize the multi-user spectrum sensing and channel access in the time-varying environment, so as to ensure and improve the network communication capability of clustered UAV system during the mission period. Based on this, the following system performance optimization mathematical model is established.

\[
\max \mathbb{E} \left[ \sum_{t=0}^{+\infty} \sum_{n=1}^{N} \sum_{m=1}^{M} a_{n,m}^t \gamma^t U_{n,m}^t \right] \\
\text{s.t.} \sum_{m=1}^{M} a_{n,m}^t \leq 1, \forall n = 1, \ldots, N, \\
a_{n,m}^t \in \{0, 1\},
\]

where \(a_{n,m}^t\) is the selection variable of user \(n\) on channel \(m\) at time \(t\), \(a_{n,m}^t = 1\) denotes that user \(n\) selects channel \(m\) to sense and access, otherwise, the channel is not selected. \(U_{n,m}^t\) characterizes the utility of user \(n\) decision on channel \(m\) at time \(t\). The utility is defined by the weight of user sensing access costs and rewards, which are given in Section III. Obviously, \(U_{n,m}^t = 0\) when \(a_{n,m}^t = 0\).

\(\gamma \in [0, 1]\) is the utility discount factor, which is introduced to ensure that the system objective function to be optimized is meaningful. If \(U_{n,m}^t \geq 0\) is always guaranteed for \(\forall m, n, t\) which leads to \(\sum_{t=0}^{+\infty} \sum_{n=1}^{N} \sum_{m=1}^{M} a_{n,m}^t \gamma^t U_{n,m}^t \rightarrow \infty\) with \(\gamma = 1\), it will be meaningless. It can be seen from Section III, there is still \(U_{n,m}^t \geq 0\) for \(\forall m, n, t\), i.e., \(\Pr[U_{n,m}^t \geq 0] > 0\) in an optimization cycle. Therefore, it is necessary to introduce the discount factor \(\gamma \in [0, 1]\) to ensure that problem (1) is always meaningful. In fact, \(\gamma\) is also consistent with the discount factor of the reward based on RL.

In addition, \(\mathbb{E}[\cdot]\) is the multi-user cumulative weighted utility expectation which is calculated by PU channel occupancy statistics. Without considering the expectation operation \(\mathbb{E}[\cdot]\), the problem (1) will be a typical 0/1 combinatorial optimization problem, which is NP-hard. Furthermore, since PU channel occupancy statistics cannot be received in advance, it will make the problem further complicated caused by the expectation operation \(\mathbb{E}[\cdot]\). Therefore, we propose a method based on MARL to solve the above problem.

III. PROBLEM MODELING BASED ON MARL

As mentioned above, due to the lack of information about PU channel occupancy state and time-varying characteristics of the occupancy state, the proposed problem will be transformed into a MARL problem which is solved based on Markov game (MG) in this section.

A. MARL Problem Model

MARL is mainly used to model and address sequential decision problems in multi-agent scenarios. Here the evolution of the system state and the reward of each agent are influenced by the joint actions of all agents. More precisely, the goal of each agent is to optimize its long-term cumulative return which is a function of all agents’ action strategies in a time-varying environment. At present, MARL has been studied for UAV driving, Go games, robotic soccer and multi-player video games [26]–[31], etc. In essence, MARL can be modeled as a MG for multi-agent, which is defined as follows.

Definition 1 ([32]): A MG is defined by a six tuple \((\mathcal{N}, \mathcal{S}, \{A_i\}_{i \in \mathcal{N}}, \mathcal{P}, \{\mathcal{R}_n\}_{n \in \mathcal{N}}, \gamma)\), where \(\mathcal{N} = \{1, \ldots, N\}\) denotes the set of \(N > 1\) agents and \(\mathcal{S}\) is the state space observed by all agents. \(A_i\) denotes the action space of agent \(n\), and action space of all agent is \(\mathcal{A} := A_1 \times \cdots \times A_N\). \(\mathcal{P} : \mathcal{S} \times \mathcal{A} \rightarrow \nabla(S)\) is the transition probability from any state \(s \in \mathcal{S}\) to any state \(s' \in \mathcal{S}\) for any given action \(a \in \mathcal{A}\). The reward function \(\mathcal{R}_n : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}\) determines the instant reward received by agent \(n\) in the process from \((s, a)\) to \(s'\), and \(\gamma \in [0, 1]\) is the discount factor.

Agent set \(\mathcal{N}\): It is composed of CUAV\(\{\}\) in a clustered CUAV communication network. When the model is extended to multi-CUAV cluster scenario, \(\mathcal{N}\) is a set of multi-CUAV clusters.

State space \(\mathcal{S}\): We define the MG state space as \(\mathcal{S} = \{s = (s_1, \ldots, s_M, o_1, \ldots, o_M)|s_m \in \{0, 1, \ldots, N\}, o_m \in \{0, 1\}, m = 1, \ldots, M\}\), where \(s_m\) denotes the number of agents that have selected channel \(m\) to sense and access in the previous time slot. \(s_m = 0\) and \(s_m = N\) means no
agent selects channel \( m \) for sensing access and all agents select channel \( m \) in the previous time slot. \( o_m \) denotes the occupancy state of channel \( m \) in the previous time slot. \( o_m = 1 \) indicates that PU occupies channel \( m \), otherwise the channel is not occupied. Each agent is able to select at most one channel for sensing and access, there is a constraint \( \sum_{m=1}^{M} s_m = N \) consequently. Thus, the size of the state space is \( |S| = 2^M \cdot M^N \) rather than \( |S| = 2^M \cdot (N+1)^M \). In addition, an initial time (time slot) \( t = 0 \) and the initial state of the system \( s = (s_0^{0}, \ldots, s_0^{M}, o_1^{0}, \ldots, o_{M}^{0}) \) are defined, in which each agent randomly selects a channel as the initial action.

**Action space** \( \mathcal{A}_n \): Action space is defined as \( \mathcal{A}_n = \{0, 1, \ldots, M-1\} \) for agent \( n \). The action \( a_n \in \mathcal{A}_n \) denotes the steps that agent \( n \) moves clockwise in current time slot relative to the channel \( m_n \in \{CH_1, CH_2, \ldots, CH_M\} \) selected in the previous time slot on the PU channel ring which is shown in Fig. 4. It is clear that \( a_n = 0 \) means that agent \( n \) keeps the channel selection unchanged, \( a_n = 1 \) means that agent \( n \) chooses the corresponding channel when it moves one step clockwise on the PU channel ring, and so on. Therefore, the channel selected by agent is \( m_n \) which can be calculated by \( m_n = (m_n' + a_n) \mod M \in \{CH_1, CH_2, \ldots, CH_M\} \), where \( \mod M \) is the modular operation of \( M \). Thus, we can define the joint action space \( \mathcal{A} = \prod_{n=1}^{N} \mathcal{A}_n \) of the system and the joint action \( \mathbf{a}^t = (a_1^t, \ldots, a_N^t) \) in \( t \) time slot.

![Fig. 4 PU channel ring and action selection mode.](image)

**State transition probability** \( P \): Assuming that the PU channel occupancy the decision-making of CUAVs are based on time slots. The PU channel occupancy can be modeled as a two-state Markov random process shown in Fig. 2. Therefore, the system state takes a time slot as the unit of state transition, i.e., \( P(s_{t+1} | s^t, \mathbf{a}^t) \), where \( s^t \) and \( \mathbf{a}^t \) are the system state and the joint actions of all agents in the \( t \) time slot, and \( s_{t+1} \) is the system state in the \( t + 1 \) time slot.

**Reward function** \( R_n \): The environment will generate a reward as a basis for evaluating the quality of the decision. Considering the particularity of clustered CUAV application environment, the spectrum resource exploration (i.e., spectrum sensing) and spectrum utilization (i.e., channel access) utility and cost are used as the reward function of each agent. Here

\[ r_{n}^{t+1}(s^{t+1}, s^t, \mathbf{a}^t) = \begin{cases} 
-E_{s,n}^{t+1} & \text{if } o_{n}^{t+1} = d_{n}^{t+1} = 1 \\
-E_{s,n}^{t+1} - E_{t,n}^{t+1} & \text{if } o_{n}^{t+1} = 1, d_{n}^{t+1} = 0 \\
-\alpha E_{s,n}^{t+1} - \beta E_{t,n}^{t+1} + (1 - \alpha - \beta) R_{n}^{t+1} & \text{if } o_{n}^{t+1} = d_{n}^{t+1} = 0 \\
-\alpha E_{s,n}^{t+1} - (1 - \alpha) R_{n}^{t+1} & \text{otherwise}
\end{cases} \]

where \( o_{n}^{t+1} \) is the same as the definition of state space. \( d_{n}^{t+1} \) is the fusion sensing decision of selected channel (i.e., channel \( m_n^{t+1} \)) received by agent \( n \) in the \( t + 1 \) time slot. \( d_{n}^{t+1} = 1 \) and \( d_{n}^{t+1} = 0 \) denote selected channel is occupied and idle respectively. \( E_{s,n}^{t+1} \) and \( E_{t,n}^{t+1} \) denote the sensing and transmission overhead of agent \( n \) respectively, including the energy consumption of transceiver, sensing and transmission delay, and so on. \( R_{n}^{t+1} \) denotes the utility which is defined as the amount of transmission data received by the user \( n \) when accessing channel successfully. And \( \alpha \) and \( \beta \) are the weighting factors of sensing and transmission overhead respectively.

From equation (2), it is clear that the reward is depended on the four cases: 1) Channel occupancy and sensed as occupied, 2) Channel occupied but sensed as idle, 3) Channel idle and sensed as idle and 4) Channel idle but sensed as occupied. In these four cases, the agent can receive positive reward only in case 3), and the rest are negative reward. The CUAV can receive a positive reward since the actual channel state is idle, but the sensing decision is occupied, which leads to a waste of resources in case 4). Therefore, the reward includes a negative reward of resource waste in case 4).

We will explain the specific meaning of \( E_{s,n}^{t+1} \), \( E_{t,n}^{t+1} \) and \( R_{n}^{t+1} \) in detail below.

\( E_s \) of CUAV comes from receiver’s sensing energy consumption when performing spectrum sensing. The sensing cost is related to receiver working voltage \( V_{DD} \), channel bandwidth \( B \), and sensing time \( \tau \), i.e., \( E_s = \tau V_{DD}^2 B \). Based on this, the sensing overhead for CUAV \( n \) when channel \( m_n^{t+1} \) is selected can be expressed by the following formula.

\[ E_{s,n}^{t+1} = \tau_s V_{DD}^2 B_{m_n^{t+1}}, \]  

where \( \tau_s \) is sensing time which is the same of every CUAV in assumption. \( B_{m_n^{t+1}} \) is the channel bandwidth of channel \( m_n^{t+1} \) selected by CUAV \( n \).
For the transmission cost \( E_{t,n}^{t+1} \), it denotes the cost incurred by CUAV \( n \) for data transmission, which is specifically expressed as follows.

\[
E_{t,n}^{t+1} = \tau_t p_t,
\]

where \( \tau_t \) is transmit time and \( p_t \) is transmit power. To simplify the problem, the \( \tau_t \) and \( p_t \) is assumed to be the same of each CUAV.

The reward also depends on the transmission utility in case 3) and case 4). As the previous analysis pointed out, the transmission utility received by CUAV depends on the number of cognitive users sharing the channel, transmission time and transmission rate. Therefore, the transmission utility \( R_{t,n}^{t+1} \) in \( t + 1 \) time slot for CUAV \( n \) is defined as

\[
R_{t,n}^{t+1} = \frac{\tau_t}{m_{t+1}^{n}} + B_{m_{t+1}^{n}}^{n} \log_2(1 + SNR_{t,n,m_{t+1}^{n}}).
\]

\( SNR_{t,n,m_{t+1}^{n}} \) is signal-to-noise ratio (SNR) received by CUAV \( n \) for this channel. It is considered here that each CUAV receives a different SNR for the reason that there are not the same of the relative spatial position, noise environment, and other influencing factors of different CUAVs. \( N_{t+1}^{t+1} \) denotes the number of CUAVs that selecting channel \( m_{t+1}^{n} \) for sensing and accessing.

The interaction information of CUAVs includes three parts: agent index, sensing channel selection and sensing decision, i.e., \( D_{t+1}^{t+1} = \{n, m_{t+1}^{n}, d_{t+1}^{n}\} \). Obviously, after information interaction, each CUAV can perform fusion based on the cooperative users’ sensing decisions to get a consistent fusion sensing decision on selected channel. The fusion rule here is the “K-out-of-N” rule which can be expressed by the following equation [34]:

\[
d_{t+1}^{n} = \begin{cases} 1, & \text{if } \sum_{n=1}^{N_{t+1}^{t+1}} d_{t+1}^{n} \geq K \\ 0, & \text{otherwise} \end{cases}
\]

Specially, when \( K = 1 \), the “K-out-of-N” rule becomes the “OR” rule, and when \( K = N \), the “K-out-of-N” rule becomes the “AND” rule [34].

### B. MARL Algorithm Framework

Aiming at the MARL problem model of the proposed problem, we further assume that the residence time of each CUAV in the network is unknown. This feature prompts us design online algorithms to optimize the long-term statistical performance of CUAV network, i.e., each CUAV programs dynamic spectrum sensing and access channels to maximize long-term reward. Here the average cumulative discounted reward is used as CUAV decision-making performance. Therefore, the proposed problem [1] is transformed into the following independent optimization problem for each agent.

\[
\max_{\pi_n} \nu_n(s^0, \pi_n) = \sum_{t=0}^{+\infty} \gamma^t \mathbb{E}(r_{t,n}^{t+1} | \pi_n, s^0),
\]

where \( s^0 \) is arbitrary initial state, \( r_{t,n}^{t+1} \) is reward in \( t \) time slot and \( \gamma \in [0, 1) \) is discount factor. In particular, the value of \( \gamma \) reflects the effect of future rewards on optimal decision-making. If \( \gamma \) tends to 0, the decision-making will pay more attention to short-term rewards, otherwise the decision-making will pay more attention to long-term rewards. \( \nu_n(s^0, \pi_n) \) denotes the value function or average cumulative discounted reward under a given state \( s^0 \) and a strategy \( \pi_n \). In this MARL, CUAV aims to find a strategy \( \pi_n \) to maximize its average cumulative discounted reward. The strategy \( \pi_n : S_n \rightarrow \pi_n \) denotes the mapping from state space to action space, which is essentially a set of probability distributions in action space.

In particular, the element \( \pi_n(s^0, a^t_n) \) in its hybrid strategy \( \pi_n(s^0_n, a^t_n) = \{\pi_n(s^0_n, a^t_n) | a^t_n \in \pi_n \} \) denotes the probability of CUAV \( n \) choosing action \( a^t_n \) in state \( s^t_n \), and there is \( \pi_n(s^0_n, a^t_n) \in [0, 1] \).

Without considering the influence of other CUAVs’ strategies, the solution to the problem [7] for CUAV \( n \) is to use an iterative search method to get a fixed point of the following Bellman equation [35].

\[
\nu_n(s^0, \pi_n^*) = \max_{a^t_n \in \pi_n} \{r_{t,n}^{t+1}(s^t, a^t_n) + \gamma \sum_{s^{t+1}} P(s^{t+1} | s^t, a^t_n) \nu_n(s^{t+1}, \pi_n^*)\},
\]

\( r_{t,n}^{t+1}(s^t, a^t_n) \) is the instant reward of CUAV \( n \) taking action \( a^t_n \) when state is \( s^t \) in \( t \) time slot. \( P(s^{t+1} | s^t, a^t_n) \) is the state transition probability of the system.

For the considered scenario, the agent lacks real-time return and system state transition probability information. A classic solution idea is Q-learning method [36] which will be used to solve the proposed problem. The problem [8] is solved by Q-learning mainly by introducing Q-function and performing Q-function update, which is defined as

\[
q^*(s^t, a^t_n) = r_{t,n}^{t+1}(s^t, a^t_n) + \gamma \sum_{s^{t+1}} P(s^{t+1} | s^t, a^t_n) \nu_n(s^{t+1}, \pi_n^*),
\]

\( q^*(s^t, a^t_n) \) is the cumulative discounted reward of CUAV \( n \) taking action \( a^t_n \) in \( s^t \) and adopting the best strategy \( \pi_n^* \) in the subsequent system dynamic changes. Therefore, there is

\[
\nu_n(s^t, \pi_n^*) = \max_{a^t_n \in \pi_n} q^*(s^t, a^t_n).
\]

The optimal strategy \( \pi_n^* \) can be found by identifying the action that maximizes \( q^*(s^t, a^t_n) \) in state \( s^t \) when \( q^*(s^t, a^t_n) \) is known. Based on this, the CUAV optimal strategy design is transformed into finding the best \( q^*(s^t, a^t_n) \) instead of \( \nu_n(s^0, \pi_n^*) \) in the process. Q-learning actually provides a simple search method for the optimal Q-function, i.e., the Q-function can be updated according to the following rule in any initial system state.

\[
q^{t+1}(s^t, a^t_n) \leftarrow (1 - \alpha^t) q^t(s^t, a^t_n) + \alpha^t \left( r_{t,n}^{t+1}(s^t, a^t_n) + \gamma \max_{a} q^t(s^t, a) \right),
\]

where \( \alpha^t \in [0, 1] \) is the time-varying learning rate. It is proved by Watkins and Dayan [37] that when the learning rate satisfies
certain conditions, the iterative sequence based on equation \([11]\) will converge to \(q^*_n(s^t, a^t_n)\) if each state is visited enough times.

Furthermore, for the proposed MARL problem, it can be noted that the reward of each CUAV not only depends on its own action \(a^t_n\) (or strategy \(\pi^*_n\)), is also influenced by other CUAVs’ actions (or strategies). Hence, we introduce a combination of joint strategy and joint action here. Joint strategy \(\pi = \{\pi_1(s^t), \ldots, \pi_N(s^t)\}\) is defined as the strategy vector of \(N\) CUAVs, and joint action vector is \(a^t = (a^t_1, \ldots, a^t_N)\) in \(t\) time slot, where each strategy or each action correspond to an independent CUAV. In addition, \(\pi_{-n} = \{\pi_1(s^t), \ldots, \pi_{n-1}(s^t), \pi_{n+1}(s^t), \ldots, \pi_N(s^t)\}\) and \(a_{-n}^t\) denote the combination of other CUAVs’ strategies and actions except the CUAV \(n\) respectively. Based on this, the optimization problem \((7)\) can be rewritten as follows giving a joint strategy \(\pi = \{\pi_1(s^t), \ldots, \pi_N(s^t)\}\).

\[
\max_{\pi_n} v_n(s^0, (\pi_n, \pi_{-n})) = \sum_{t=0}^{\infty} \gamma^t E_{n}^{t+1}(s^t, (\pi_n, \pi_{-n}))(s^0, (\pi_n, \pi_{-n})). \tag{12}
\]

For this multi-agent scenario, it is necessary to introduce an appropriate evaluation index or solution concept before designing a specific algorithm to find the optimal strategy. Currently, the most widely used solution concept is Nash equilibrium (NE) which is defined as follows.

**Definition 2** \([32]\): The Nash equilibrium of MG \((N, S, \{A_n\}_{n \in N}, \{P_n\}_{n \in N}, \gamma)\) is a joint strategy \(\pi^* = (\pi^*_n, \pi_{-n}^*)\), such that for any \(s^0 \in S\) and \(n \in N\),

\[
v_n(s^0, (\pi^*_n, \pi_{-n}^*)) \geq v_n(s^0, (\pi_n, \pi_{-n}^*)), \forall \pi_n.
\]

NE characterizes an equilibrium point \(\pi^*\), so that no agent has any motivation to deviate from the equilibrium strategy. In other words, for any agent \(n \in N\), the strategy \(\pi^*_n\) is the optimal response of the strategy combination \(\pi_{-n}^*\). It is hard to receive the NE solution from the theoretical analysis \([37]\), so a Q-learning algorithm and a DQN algorithm based on independent learner is proposed which are based on the basic idea of the Q-learning algorithm, see Section IV for details. Because it is difficult to analyze the asymptotic optimality and convergence of the algorithm in theory, we will verify the performance of the proposed algorithms through numerical simulation analysis.

**IV. MARL ALGORITHM**

In this section, we design three optimization algorithms based on MARL. And a brief analysis of the information interaction overhead, execution complexities and convergence of the proposed algorithms are given.

**A. VC-EXH Algorithm**

Firstly, a time slot multi-round revisit exhaustive search algorithm based on virtual controller (VC-EXH) is proposed. There is a virtual controller (VC) in the system which has the ability to revisit the next time slot in multi-round. As a result, by listing every actions combination, the VC can evaluate the performance of the system under that combination, and will select a optimal joint action combination based on it subsequently. Fig.5 visually shows the VC-EXH algorithm, and the specific process is summarized as Algorithm 1.

**Algorithm 1 VC-EXH algorithm**

1. **Initialize:** Set \(t = 0\) and time period \(T\);
2. **while** \(t < T\) **do**
   3. VC arranges all \(M^N\) kinds of action combinations to form action combination set \(\mathcal{A}\);
   4. **for all** \(a \in \mathcal{A}\) **do**
      5. VC broadcasts the action combination \(a\) to each agent;
      6. **for all agent** \(n \in N\) **do**
         7. Based on the action \(a^t_n \in \mathcal{A}_n\) of VC broadcast, select channel for spectrum sensing, and produce sensing decision \(d^t_n\);
         8. Feedback sensing information \(D^t_n = \{n, m^t_n, d^t_n\}\) on CCC;
         9. Receive fusion sensing decision \(d^t_{m_n}\) according to \((6)\);
         10. Access channel based on fusion sensing decision, and receive reward \(r^t\) according to \((2)\) and feedback;
      **end for**
   11. VC calculates the system reward \(v_a = \sum_{n=1}^{N} r^t\) under the action combination;
   **end for**
   12. VC selects the joint action \(a^* = \arg \max_{a \in \mathcal{A}} v_a\) as the best joint actions combination in current time slot;
13. **end for**
14. **Update** \(t = t + 1\);
15. **end while**

The VC and agents repeat \(|\mathcal{A}| = M^N\) times to evaluate the reachability of the system under each possible action combination for each time slot. This algorithm is a non-causal ideal algorithm, so it is used as the performance boundary comparison algorithm of the subsequent proposed algorithm. Further, assuming that the VC has non-causal system state information such as agent state, location information and channel information, the entire algorithm can be transformed...
into a completely centralized algorithm. Therefore, the VC can simulate all processes and select the best actions combination that does not need agents to participate. And agents can perform follow-up work through the best actions combination.

B. IL-Q Algorithm

In view of the modeled the proposed problem, a Q-learning algorithm based on independent learner [36], [38] (i.e., the IL-Q algorithm) is proposed. In essence, the IL-Q algorithm extends the Q-learning algorithm in a single-agent RL problem to a MARL problem. The core idea of the IL-Q algorithm is that each agent regards other agents as part of the environment and uses Q-Learning algorithm to solve problem, thereby transforming the MARL problem into a single-agent RL problem, as shown in Fig. 6.

![Fig. 6 CUAV n uses the IL-Q algorithm for sensing and access.](image)

Specifically, each CUAV implements a standard Q-learning algorithm to learn the best Q-value and determines the best action strategy based on this. When the agent takes action $a_n^t$ in $s^t$, it will receive a corresponding reward $r_n^t$. Based on $r_n^t$, the agent updates the Q-value $Q_n^t(s^t, a^t)$ and transforms $s^t$ to $s^{t+1}$ [36]. $Q_n^t(s^t, a^t)$ is a function of state-action pairs whose value is stored in a Q-table with size $|S| \times |A|^n$. The agent needs to update Q-value according to formula (11) and the updated Q-value is replaced with the original Q-value and stored in the Q-table. $Q_n^t(s^t, a^t)$ is updated by

$$Q_n^{t+1}(s^t, a^t) \leftarrow (1-\alpha^t)Q_n^t(s^t, a^t) + \alpha^t \left( r_n^{t+1} + \gamma \max_{a^{'t}} Q_n^t(s^t, a^{'t}) \right).$$

(14)

In order to ensure the convergence of the algorithm based on independent learner, the value of the time-varying learning rate $\alpha^t$ is set as follows [39]

$$\alpha^t = \frac{1}{(t+c_\alpha)^{\varphi_\alpha}},$$

(15)

where $c_\alpha > 0$, $\varphi_\alpha \in (0, 1]$. From (14), we note that the Q-function is updated in a recursive way. When the step factor of the Q-function iteration and the immediately reward meet certain conditions (i.e., Proposition 1), it will eventually converge to the best Q-value [39]. The details of the convergence analysis will be presented in Section IV-D.

Further, Q-learning needs to continuously optimize the agent action strategy in the iterative process until the optimal action strategy is received. Specifically, the iterative optimization of agent strategy is generally based on the estimated Q-function to guide the actual action selection. It requires a exploration-exploitation tradeoff to avoid getting stuck in local optimal Q-function estimation. A typical strategy for balanced exploration and exploitation is the $\epsilon$-greedy strategy [36], which selects the action with the largest Q-value with probability $1-\epsilon$, and randomly selects any action in the action space with probability $\epsilon$.

$$a_n^{t+1} = \begin{cases} \arg\max_{a^{'t}} Q_n^t(s^t, a^{'t}), & \text{w.p.}(1-\epsilon) \\ \text{random}(A^n), & \text{w.p.} \epsilon. \end{cases}$$

(16)

The Q-value is used to estimate the value of using action $a_n^{t+1}$. While achieving a tradeoff, this method has very little possibility of choosing inferior actions, reducing most of the uneconomical attempts which can also achieve convergence. In summary, the IL-Q algorithm (Algorithm 2) is formed.

Algorithm 2 IL-Q algorithm

1: Initialize: Set $t = 0$, initialize discount factor $\gamma \in [0, 1)$, parameters $c_\alpha > 0$, $\varphi_\alpha \in (0, 1]$, and time period $T$;
2: for all agent $n \in N$ do
3: initialize $Q_n^t(s^t, a^t) = 0$ and $s^0$;
4: end for
5: while $t < T$ do
6: for all agent $n \in N$ do
7: Update the learning rate $\alpha^t$ according to (15);
8: Select an action $a_n^t$ at $s^t$ according to (16);
9: Take action $a_n^t$ to select channel for spectrum sensing and produce sensing decision $d_n^t$;
10: Feedback sensing information $D_n^t = \{n, m_n^t, d_n^t\}$ on CCC;
11: Receive fusion sensing decision $d_n^{m_n^t}$ according to (3);
12: Access channel based on fusion sensing decision, and receive reward $r_n^{t+1}$ according to (3) and observe $s^{t+1}$;
13: Update $Q_n^{t+1}(s^t, a^t)$ according to (14);
14: end for
15: Update $t = t + 1$ and state $s^t \leftarrow s^{t+1}$;
16: end while

C. IL-DQN Algorithm

For the proposed IL-Q algorithm, each agent has an independent Q-table with size $|S| \times |A|^n$ and the size of the entire system Q-table is $N \times |S| \times |A|^n$ consequently. However, when the system action or state space is large even continuous, this method will lead to the curse of dimensionality. In addition, agent needs to visit enough times for each state-action pair to ensure that the Q-value converges to the optimal in Q-learning [39]. Obviously, this method will significantly reduce its applicability or even become unusable for continuous state or action spaces, as well as situations where the state or action spaces are huge. In order to solve the curse of
dimensionality and efficiency problems faced by traditional Q-learning, a generally applicable method is to introduce deep reinforcement learning algorithm (DRL). At present, various DRL algorithm have been proposed, including DQN (deep Q-network), DDQN (double DQN) [16] [21] [31] [45], etc. We will use DQN algorithm to construct a DQN algorithm based on independent learner (IL-DQN), i.e., each agent runs the DQN algorithm independently to achieve strategy evaluation and value function iteration. The agent uses a deep neural network to fit the Q-table to approximate each Q-value of a state-action pair in this process.

The core of the IL-DQN algorithm is the five functional components in DQN which are in Fig. 7 specifically as well as the relationships among them. And each component is described in detail as follows.

**Input Layer**: The input of DQN is a vector with size $2 \times M$, corresponding to the state system $s^t = (s^t_1, \ldots, s^t_M, a^t_1, \ldots, a^t_M)$ in time slot, where the first $M$ values is the number of CUAVs that selecting each PU channel to sense, and the last $M$ values is the occupancy state of each PU channel respectively.

**Output Layer**: The output of DQN is a vector with size $M$, corresponding to the Q-value estimation of all optional actions in the current system state, i.e., $Q^t_n = [Q^t_{n,0}, Q^t_{n,1}, \ldots, Q^t_{n,M-1}]$. 

**Experience Replay**: In DQN, the experience replay component is to store historical samples which are experience tuples $(s^t, a^t_n, r^t_{n+1}, s^{t+1})$ composed of current state $s^t$, action $a^t_n$, reward $r^t_{n+1}$, and next state $s^{t+1}$. During the learning process, the agent randomly samples a batch $B$ experience tuples from the experience replay to fit the deep network to the Q-table which will eliminate the temporal correlation of historical samples generated by RL.

**Action selection strategy**: In order to prevent the action selection falling into the local optimum during the period of unconverged deep neural network training stage, the $\epsilon$-greedy strategy is generally introduced during action selection, i.e.,

$$
a^t_n + 1 = \begin{cases} 
\arg \max_{a^{t+1}} Q^t_n (s^t, a^{t+1}; \theta^t_n), & w.p.(1 - \epsilon) \\
\text{random}(A^n), & w.p.\epsilon
\end{cases}
$$

(17)

**Current Q-Network**: The current Q-network (ie, Q-table fitting deep neural network) realizes the mapping of the input state $s^t$ to the corresponding Q-value $Q^t_n (s^t, a^{t+1}; \theta^t_n)$ of each action $a^{t+1}_n$, where $\theta^t_n$ denotes the parameters of the current Q-network. The experience tuples are mainly used to train the current Q-network to update parameters $\theta^t_n$ until convergence. After training, a action will be selected based on the output Q-value.

**Target Q-Network**: The target Q-network has the same structure as the current Q-network whose initial parameters are also the same. The output target Q-value $Q^{t+1}_n (s^t, a^{t+1}; \theta^t_n)$ is mainly used to construct the supervision of the current Q-network iterative training, where $\theta^t_n$ is the target Q-network parameters. In DQN, $\theta^t_n$ is not updated every time when learning, but is updated in a fixed period $F$. It will directly assigns the value of $\theta^t_n$ to $\theta^t_n$, which is called fixed Q-goals in DQN.

**Loss Function**: The loss function used in training the current Q-network is defined as the following equation (18).

$$
L^t_n (\theta^t_n) = \frac{1}{B} \sum_{i=1}^{B} (y_{n,i} - Q^t_n (s^t, a^t_n; \theta^t_n))^2,
$$

(18)

where

$$
y_{n,i} = r^t_{n+1} + \gamma \max_{a^t_n} Q^t_{n,i} (s^t, a^t_n; \theta^t_n),
$$

(19)

$B$ denotes a batch size. It can be found that the loss function is a mean square error between the output Q-value of the target Q-network and the current Q-network. After receiving the value of the loss function, the gradient descent method is used to update $\theta^t_n$ iteratively, i.e.,

$$
\theta^t_n + 1 \leftarrow \theta^t_n + \zeta \nabla_{\theta^t_n} L^t_n (\theta^t_n),
$$

(20)

where $\zeta$ is the learning rate of DQN. Here the gradient $\nabla_{\theta^t_n} L^t_n (\theta^t_n)$ is calculated by

$$
\nabla_{\theta^t_n} L^t_n (\theta^t_n) = \nabla_{\theta^t_n} \left[ \frac{1}{B} \sum_{i=1}^{B} (y_{n,i} - Q^t_{n,i} (s^t, a^t_n; \theta^t_n))^2 \right].
$$

(21)

For the considered scenario, based on description of the DQN components, the detailed steps of the IL-DQN algorithm are given below, i.e., Algorithm 3.

**D. Algorithm Complexity and Convergence Analysis**

1) Information interaction cost and algorithm execution complexity analysis: VC-EXH algorithm: The information that the VC needs to interact with each agent includes broadcasting the current actions combination to each agent, and receiving the feedback (i.e., immediate reward) from each agent. Since the algorithm needs to exhaust all possible actions combinations, its execution complexity is $M^N$ which increases exponentially as agents increase.

IL-Q algorithm: Since each agent executes the Q-learning algorithm independently, its information interaction is mainly to broadcast its own sensing decision information. The amount of information interaction increases linearly with the increase of agents. At the level of algorithm execution, the main overhead is the storage overhead of the Q-table of size $N \cdot 2^M M^N$, related to the number of state and action.
increases exponentially as agents and PU channels increase. In terms of computational cost, it comes from the update of the Q-table and the search for the optimal action which is small.

IL-DQN algorithm: The information exchange cost is the same as the IL-Q algorithm. At the level of algorithm execution, since the Q-table is no longer used to directly store state-actions, but a deep neural network is used to fit the Q-table, the storage cost mainly depends on the structure of the deep network and its parameters. In contrast, the storage overhead of the IL-DQN algorithm is less than that of the IL-Q algorithm. However, since the IL-DQN algorithm involves updating current Q-network and target Q-network, the computational complexity will be greater than that of the IL-Q algorithm which is related to the IL-DQN deep network structure and the amount of parameters. It increases linearly as the number of IL-DQN deep network parameters increases.

2) Algorithm convergence analysis: Before analyzing the convergence of the proposed algorithms, here is the conclusion of the convergence of the single agent Q-learning algorithm, i.e., the following Theorem 1. The proofs are given in [37] and [39].

Theorem 1 (Q-learning algorithm convergence [37]): For the MDP process and its state $s \in S$, action $a \in A$ and reward function $R(s, a)$. The Q-learning algorithm is

$$Q(s, a) = Q(s, a) + \alpha \left( R(s, a) + \max_{a'} Q(s', a') - Q(s, a) \right).$$

Under satisfying $\sum_{t=0}^{\infty} \alpha_t = \infty$, $\sum_{t=0}^{\infty} (\alpha_t)^2 < \infty$ and $|R(s, a)|$ under bounded conditions, it converges to the best Q-function with the largest state-action value with probability 1 when $t \to \infty$.

Regarding the proposed IL-Q and IL-DQN algorithm, each agent essentially executes the Q-learning or DQN algorithm independently, while other agents are treated as part of the environment. As pointed out in [40], it converges only in some special cases when applying the single-agent Q-learning algorithm to the multi-agent environment, such as iterative dominant solvable games and team game situations [41]. For the proposed IL-Q algorithm or IL-DQN algorithm, it is generally difficult to directly prove its convergence. Nevertheless, similar to [39], it is not difficult to draw the following conclusion about the convergence of independent learners due to the non-cooperative nature of the multi-agent in the considered scenario.

Proposition 1: For the proposed IL-Q algorithm and IL-DQN algorithm, each agent can converge to its optimal Q-value when the iterative process meets the conditions listed in Theorem 1.

It is worth noting that the above Proposition 1 can only guarantee the convergence of the independent iterative process of each agent, which is not the same as the overall convergence of the algorithm. Therefore, we will verify the convergence of the entire algorithm through numerical simulation analysis. In addition, a mechanism to ensure the convergence of the algorithm is proposed in [42], i.e., introducing a strict optimal response mechanism based on a weakly acyclic stochastic team game. All agents are based on a common objective function, and each agent adopts a strict optimal response in the iterative process. More specifically, other agents’ strategy is unchanged before the iteration process of each agent converges, which essentially constructs a staged and stable external environment. Obviously, it can converge satisfying Theorem 1 when the agent use Q-learning. The analysis can also be extended to this paper.

V. SIMULATION AND ANALYSIS

In this section, the system performance of the proposed algorithms is evaluated via simulations. The computer parameter configuration used is Intel Core i5-4460 CPU 3.2GHz and simulation software platform is Anaconda 3. Owing to the algorithm complexity of the VC-EXH algorithm and the IL-Q algorithm, it will take too long simulation time when $N$ and $M$ are large. The size of state space is $2^{10} \cdot 10^{11} \approx 10^{13}$ and the number of system state-action pairs is about $10^{14}$ when
For this reason, we will focus on small-scale network scenarios in simulation.

Specifically, without special instructions, the main simulation parameters are shown in TABLE I below. The PU channel occupancy state obeys the Markov random process with parameter \((1-\alpha_m, 1-\beta_m), \forall m = 1, \ldots, M\), and random initialization in simulation.

| Parameters                  | Value          |
|-----------------------------|----------------|
| PU channels \(M\)           | 6              |
| CUAVs \(N\)                 |                |
| Channel bandwidth \(B_m\)   | 5~8            |
| SNR \(SNR_{\alpha,m}\)     | 20~40dB        |
| False alarm probability \(P_f\) | 0.1 [19]   |
| Detection probability \(P_d\) | 0.9           |
| Transmission power \(P_t\)  | 23dBm [40]     |
| Sensing time \(\tau_s\)    | 0.16 [39]      |
| Transmission time \(\tau_t\) | 0.52          |
| Weighting parameters \(\alpha, \beta\) | 0.01, 0.05   |

Regarding the hyper-parameters of the IL-Q algorithm and the IL-DQN algorithm, the settings are shown in TABLE II below without special circumstances.

| Hyper-parameters          | Value          |
|---------------------------|----------------|
| Greedy rate \(\epsilon\)  | 0.1            |
| Discount factor \(\gamma\) | 0.99           |
| Learning rate of DQN \(\alpha, \varphi\) | 0.5, 0.8 [59] |
| Learning rate of CNN \(\zeta\) | 0.01          |
| Parameters of CNN \(C_{\text{CNN}}\) | (2,2,10) [43] |
| Activation function       | ReLu           |
| Optimizer                 | Adam           |
| Target Q-Network update period \(F\) | 100          |
| Experience replay size \(C\) | 2000          |

We first evaluate the system average cumulative discounted reward of the proposed algorithms, and the results are shown in Fig. [8]. It illustrates that with the increase of time slots, the average cumulative discount rewards of three algorithms increase and converge significantly. They can reach convergence and stability after iterating 500 time slots, which verifies the convergence of the IL-Q algorithm and the IL-DQN algorithm. In addition, it shows that the performance of the VC-EXH algorithm is always optimal in the entire time slot period, followed by the IL-DQN algorithm, and the IL-Q algorithm has the worst performance. The average cumulative discounted reward of the VC-EXH algorithm is about 1.2 times that of the IL-DQN algorithm and 1.4 times that of the IL-Q algorithm, while the IL-DQN algorithm is 1.2 times that of the IL-Q algorithm for the reason that the VC-EXH algorithm ensures that the optimal joint actions combination can be found in each time slot, and the maximum reward can always be received. Relatively speaking, the other two algorithms fluctuate in the early stage of the iteration, and may even experience performance degradation in certain time slots because the Q-table or DQN-network does not reach stability at this stage. After reaching the stability in the later stage, agent will look for the best action to get the maximum reward, and the system average cumulative discounted reward gradually rises.

Subsequently, against the running time of the proposed algorithms in different scenarios which will evaluate the algorithm execution complexity and verify the algorithm complexity analysis in Section IV-D, we have a comparative analysis. \(N\) takes the value of set \(4, 5, 6, 7, 8\), and the running time is calculated through 1000 time slot. Considering that the running time of the VC-EXH algorithm increases exponentially as CUAVs and PU channels increase, the simulations will mainly focus on the IL-Q algorithm and the IL-DQN algorithm. Fig. [9] shows that with the increase of \(N\), the running time of the IL-Q algorithm gradually increases. The running time is as high as 18.5s when \(N = 8\). Correspondingly, the IL-DQN algorithm takes less time consumption which is only \(1/6\) of the IL-Q algorithm. And with the increase of \(N\), the time consumption does not increase obviously, so it is very suitable for larger-scale network scenarios. It also proves the IL-DQN algorithm is more suitable for high-dimensional problems compared with the IL-Q algorithm. It can be noted from Fig. [8] and Fig. [9] that the IL-DQN algorithm can not only achieve better system performance, but also take less time consumption under the same scale problem.
Moreover, we simulate and analyze the impact on system performance under different CUAV-PU ratios (i.e., $M/N$). The ratio $M/N$ increases from 0.6 to 1.4, and the average cumulative discount reward is the reward value after iterating 1000 time slots. The simulation results are shown in Fig.11. It can be observed that the larger the $M/N$ is, the worse the system performance is. Especially, the performance degradation of the IL-Q algorithm is more obvious. Spectrum competition becomes more obvious, and the spectrum resources received by each CUAV generally decline with the increase of CUAVs.

However, the number of CUAVs cannot be too small in order to improve the sensing reliability. The results show that the ratio setting of 0.6 is more suitable for the considered scenarios, and can achieve a compromise between collaboration and conflict.

Lastly, the state transition probability of each channel, i.e., $(\alpha_m, \beta_m)$, is randomly initialized and the percentage of each channel selected under different occupancy-idle ratios $[\alpha_m + (1 - \beta_m)]/[\beta_m + (1 - \alpha_m)]$ is further analyzed. In order to ensure that the simulation is not affected by other parameters, the scenario settings and simulation parameters are almost the same as in Fig.8. Only all bandwidths are set to 100KHz. The performance is analyzed after iterating over 30,000 time slots and the results are shown in Fig.12. TABLE III shows the initial channel state transition probabilities and occupancy-idle ratios. Fig.12 illustrates that $CH_6$ is selected the most times and $CH_2$ is selected the least, i.e., the smaller $[\alpha_m + (1 - \beta_m)]/[\beta_m + (1 - \alpha_m)]$ is, the number of times the channel is selected less. It proves that the agent by using the proposed algorithms can find the idle PU channel with a greater probability for spectrum sensing and access, and improve the utilization of spectrum resources. Further, the IL-DQN algorithm has more obvious differences in the selection of different channels comparing with the IL-Q algorithm. The probability of choosing channel with a low occupancy-idle ratio is high so that the idle channels are selected intelligently.

VI. CONCLUSION

In this paper, considering a dynamic cooperative spectrum sensing and channel access problem of clustered CUAV communication networks in a time-varying environment, we introduce a return function based on the weighted compound of sensing-transmission cost and utility, and use MARL (i.e.,
Markov game) to model the problem. Based on this, three algorithms are proposed: the VC-EXH algorithm, the IL-Q algorithm, and the IL-DQN algorithm to maximize the average cumulative discount reward of the system and improve the network communication capabilities of clustered CUAV system during the mission duration. Numerical simulation results show the superiority and convergence of the proposed algorithms which utilize idle spectrum resources efficiently.

In fact, it is possible to further form the belief state of CUAVs with the help of the historical channel sensing information of CUAVs rather than forming a consistent view which we will study in the future. In addition, it is also worth to be studied designing a more effective distributed mechanism to promote the rapid convergence of the MARL algorithm.

**ACKNOWLEDGMENT**

The authors would like to thank...

**REFERENCES**

[1] J. Li, P. Li, and K. Liu, “Research on UAV communication network topology based on small world network model,” in Proc. IEEE Int. Conf. Unmanned Syst. (ICUS), Beijing, China, Oct. 2017, pp. 444-447.

[2] Z. Ma, et al., “Impact of UAV Rotation on MIMO Channel Characterization for Air-to-Ground Communication Systems,” IEEE Trans. Veh. Technol., vol. 69, no. 11, pp. 12418-12431, Nov. 2020.

[3] X. Tan, S. Su and X. Sun, “Research on Narrowband Interference Suppression Technology of UAV Network Based on Spread Spectrum Communication,” in Proc. IEEE Int. Conf. Artif. Intell. Inf. Syst. (ICAIIS), Dalian, China, Mar. 2020, pp. 335-338.

[4] J. Mitola and G. Q. Maguire, “Cognitive radio: making software radios more personal,” IEEE Pers. Commun., vol. 6, no. 4, pp. 13-18, Aug. 1999.

[5] H. Chen, et al., “Joint Spectrum Sensing and Resource Allocation Scheme in Cognitive Radio Networks with Spectrum Sensing Data Falsification Attack,” IEEE Trans. Veh. Technol., vol. 65, no. 11, pp. 9181-9191, Nov. 2016.

[6] R. Nie et al., “Max-Min Distance Clustering Based Distributed Cooperative Spectrum Sensing in Cognitive UAV Networks,” in IEEE Int. Conf. Commun., Shanghai, China, May. 2019, pp. 1-6.

[7] L. Ye, Y. Zhang, Y. Li, and S. Han, “A Dynamic Cluster Head Selecting Algorithm for UAV Ad Hoc Networks,” in Int. Wirel. Commun. Mob. Comput. (IWCMI), Limassol, Cyprus, Jun. 2020, pp. 225-228.

[8] B. Ji et al., “Secrecy Performance Analysis of UAV Assisted Relay Transmission for Cognitive Network With Energy Harvesting,” IEEE Trans. Veh. Technol., vol. 69, no. 7, pp. 7404-7415, Jul. 2020.

[9] P. Feng et al., “CorgMOR-MAC: A cognitive multi-channel opportunistic reservation MAC for multi-UAVs ad hoc networks,” Comput. Commun., vol. 136, pp. 30-42, Feb. 2019.

[10] A. Shamsoshoara et al., “Distributed Cooperative Spectrum Sharing in UAV Networks Using Multi-Agent Reinforcement Learning,” in IEEE Annu. Consum. Commun. Netw. Conf. (CCNC), Las Vegas, NV, USA, Jan. 2019, pp. 1-6.

[11] J. Yang et al., “Application of reinforcement learning in UAV cluster task scheduling,” Future Gener. Comp. Syst., vol. 95, pp. 140-148, Jun. 2019.

[12] J. Landén, S. R. Kulkarni, V. Koivunen, and H. V. Poor, “Multiagent Reinforcement Learning Based Spectrum Sensing Policies for Cognitive Radio Networks,” IEEE J. Sel. Top. Signal Process., vol. 7, no. 5, pp. 858-868, Oct. 2013.

[13] B. F. Lo and I. F. Akylidiz, “Reinforcement learning for cooperative sensing gain in cognitive radio ad hoc networks,” Wirel. Netw., vol. 19, no. 6, pp. 1237-1250, Aug. 2013.

[14] W. Ning et al., “Reinforcement learning enabled cooperative spectrum sensing in cognitive radio networks,” J. Commun. Netw., vol. 22, no. 1, pp. 12-22, Feb. 2020.

[15] M. Zhang, L. Wang, and Y. Feng, “Distributed cooperative spectrum sensing based on reinforcement learning in cognitive radio networks,” AEU-Int. J. Electron. Commun., vol. 94, pp. 359-366, Sep. 2018.

[16] V. Mnih et al., “Human-level control through deep reinforcement learning,” Nature, vol. 518, no. 7540, pp. 529-533, Feb. 2015.

[17] H. Q. Nguyen et al., “Deep Q-Learning with Multiband Sensing for Dynamic Spectrum Access,” in IEEE Int. Symp. Dyn. Spectr. Access Networks (DySPAN), Seoul, South Korea, Oct. 2018, pp. 1-5.

[18] S. Wang, H. Liu, P. H. Gomes, and B. Krishnamachari, “Deep reinforcement learning for dynamic multichannel access in wireless networks,” IEEE Trans. Cogn. Commun. Netw., vol. 4, no. 2, pp. 257-265, Jun. 2018.

[19] Y. Zhang, P. Cai, C. Pan, and S. Zhang, “Multi-Agent Deep Reinforcement Learning-Based Cooperative Spectrum Sensing With Upper Confidence Bound Exploration,” IEEE Access, vol. 7, pp. 118898-118906, Aug. 2019.

[20] Y. Li et al., “Deep Reinforcement Learning for Dynamic Spectrum Sensing and Aggregation in Multi-Channel Wireless Networks,” IEEE Trans. Cogn. Commun. Netw., vol. 6, no. 2, pp. 464-475, Jun. 2020.

[21] P. Cai, Y. Zhang, and C. Pan, “Cooperation Graph-Based Deep Reinforcement Learning for Cooperative Spectrum Sensing Under Correlated Fading,” IEEE Wirel. Commun. Lett., vol. 9, no. 10, pp. 1778-1781, Oct. 2020.

[22] Z. Shafig, R. Abbas, M. H. Zafar, and M. Bashiri, “Analysis and Evaluation of Random Access Transmission for UAV-Assisted Vehicle-to-Infrastructure Communications,” IEEE Access, vol. 7, pp. 12427-12440, Jan. 2019.

[23] P. Li and J. Xu, “Fundamental Rate Limits of UAV-Enabled Multiple Access Channel With Trajectory Optimization,” IEEE Trans. Wirel. Commun., vol. 19, no. 1, pp. 458-474, Jan. 2020.

[24] M. Y. Arafat and S. Moh, “Localization and Clustering Based on Swarm Intelligence in UAV Networks for Emergency Communications,” IEEE Internet Things J., vol. 6, no. 5, pp. 8958-8976, Oct. 2019.

[25] H. Zhang, H. Zhang, K. Long, and G. K. Karagiannidis, “Deep Learning Based Radio Resource Management in NOMA Networks: User Association, Subchannel and Power Allocation,” IEEE Trans. Netw. Sci. Eng., vol. 7, no. 4, pp. 2406-2415, Oct.-Dec. 2020.

[26] F. Wu, H. Zhang, J. Wu, and L. Song, “Cellular UAV-to-Device Communications: Trajectory Design and Mode Selection by Multi-Agent Deep Reinforcement Learning,” IEEE Trans. Commun., vol. 68, no. 7, pp. 4175-4189, Jul. 2020.

[27] N. Zhao, Z. Liu, and Y. Cheng, “Multi-Agent Deep Reinforcement Learning for Trajectory Design and Power Allocation in Multi-UAV Networks,” IEEE Access, vol. 8, pp. 139670-139679, Jul. 2020.

[28] D. Silver et al., “Mastering the game of go with deep neural networks and tree search,” Nature, vol. 529, no. 7587, pp. 484-489, 2016.

[29] H. Shi et al., “An Adaptive Strategy Selection Method With Reinforcement Learning for Trajectory Design and Power Allocation in Multi-UA V Networks,” IEEE Access, vol. 7, pp. 118898-118906, Aug. 2019.

[30] Y. Ban et al., “MA360: Multi-Agent Deep Reinforcement Learning Based Live 360-Degree Video Streaming on Edge,” in Proc. IEEE Int. Conf. Multimedia Expo., London, United Kingdom, Jul. 2020, pp. 1-6.
[31] L. Lei, M. Han, Y. Sun, and H. Yang, “Video Game Design and Performance Analysis Based on Reinforcement Learning,” in Proc. - Int. Jt. Conf. Inf., Media, Eng.(IJCIME), Osaka, Japan, Dec. 2019, pp. 230-234.

[32] K. Zhang, Z. Yang, and T. Başar, “Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms,” arXiv preprint arXiv: 1911.10635, Nov. 2019.

[33] X. Zhang and K. G. Shin, “E-MIL: Energy-Minimizing Idle Listening in Wireless Networks,” IEEE Trans. Mob. Comput., vol. 11, no. 9, pp. 1441-1454, Sept. 2012.

[34] Z. Chen and R. Qiu, “Cooperative Spectrum Sensing Using Q-Learning with Experimental Validation,” in Conf. Proc. IEEE SOUTHEAST-CON, Nashville, TN, USA, Mar. 2011, pp. 405-408.

[35] J. Hu and M. P. Wellman, “Nash Q-Learning for General-Sum Stochastic Games,” J. Mach. Learn. Res., vol. 4, no. 6, pp. 1039–1069, Aug. 2004.

[36] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction Second Edition. Cambridge, MA, USA: MIT Press, 2019.

[37] C. J. C. H. Watkins and P. Dayan, “Q-learning,” Mach. Learn., vol. 8, no. 3-4, pp. 279-292, May 1992.

[38] M. A. Hossen and S.-J. Yoo, “Q-Learning Based Multi-Objective Clustering Algorithm for Cognitive Radio Ad Hoc Networks,” IEEE Access, vol. 7, pp. 181959-181971, Dec. 2019.

[39] J. Cui, Y. Liu and A. Nallanathan, “Multi-Agent Reinforcement Learning-Based Resource Allocation for UAV Networks,” IEEE Trans. Wirel. Commun., vol. 19, no. 2, pp. 729-743, Feb. 2020.

[40] J. Hu et al., “Cooperative Internet of UAVs: Distributed Trajectory Design by Multi-Agent Deep Reinforcement Learning,” IEEE Trans. Commun., vol. 68, no. 11, pp. 6807-6821, Nov. 2020.

[41] M. Bowling, “Multiagent learning in the presence of agents with limitations,” Ph.D. dissertation, CMU, Pittsburgh, PA, USA, 2003.

[42] A. Kaur and K. Kumar, “Energy-Efficient Resource Allocation in Cognitive Radio Networks Under Cooperative Multi-Agent Model-Free Reinforcement Learning Schemes,” IEEE Trans. Netw. Serv. Manag., vol. 17, no. 3, pp. 1337-1348, Sept. 2020.

[43] P. Cai, Y. Zhang, and C. Pan, “Coordination Graph-Based Deep Reinforcement Learning for Cooperative Spectrum Sensing Under Correlated Fading,” IEEE Wirel. Commun. Lett., vol. 9, no. 10, pp. 1778-1781, Oct. 2020.

[44] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv preprint arXiv:1412.6980 Jan. 2017.

[45] B. Wang, Y. Li, W. Ming, and S. Wang, “Deep Reinforcement Learning Method for Demand Response Management of Interruptible Load,” IEEE Trans. Smart Grid, vol. 11, no. 4, pp. 3146-3155, Jul. 2020.

[46] Z. Chen and R. Qiu, “Cooperative Spectrum Sensing Using Q-Learning with Experimental Validation,” in Conf. Proc. IEEE SOUTHEAST-CON, Nashville, TN, USA, Mar. 2011, pp. 405-408.

[47] W. Han, J. Li, Z. Tian, and Y. Zhang, “Efficient Cooperative Spectrum Sensing with Minimum Overhead in Cognitive Radio,” IEEE Trans. Wirel. Commun., vol. 9, no. 10, pp. 3006-3011, Oct. 2010.

[48] N. Abdi, E. Yazdian, and A. M. Doost Hoseini, “Optimum number of secondary users in cooperative spectrum sensing methods based on random matrix theory,” in Int. Conf. Comput. Knowl. Eng.(ICCKE), Mashhad, Iran, Oct. 2015, pp. 290-294.