Multi-Agent Deep Reinforcement Learning Based Transmission Latency Minimization for Delay-Sensitive Cognitive Satellite-UAV Networks

Shaoai Guo, *Student Member, IEEE*, and Xiaohui Zhao

***Abstract*— With the ubiquitous deployment of a massive number of Internet-of-Things (IoT) devices, the satellite-aerial networks are becoming a promising candidate to provide flexible and seamless service for IoT applications. Concerning about the spectrum scarcity issue, we present a cognitive satellite-aerial network where the multiple unmanned aerial vehicles (UAVs) can share spectrum with the satellite without interfering with satellite communications. To further improve spectral efficiency, non-orthogonal multiple access (NOMA) technique is adopted in this network. Considering the delay-sensitive quality-of-service (QoS) requirement, a joint trajectory and power optimization problem is formulated to minimize the total transmission latency over a long-term task period. In order to reduce the computational complexity and ease the burden of information exchange by using the centralized DRL methods, we propose a multi-agent deep deterministic policy gradient (MADDPG) based algorithm which adopts the framework of centralized training with decentralized execution to solve the sophisticated problem. The simulation results show the proposed algorithm can achieve satisfactory performance through joint trajectory control and power allocation for the UAVs compared with other methods.**

***Index Terms*— Transmission latency, cognitive radio, satellite-UAV network, multi-agent deep reinforcement learning, trajectory control, power allocation.**

I. Introduction

With the increasing demands of seamless and ubiquitous network access in the fifth generation (5G) and upcoming sixth generation (6G), it becomes particularly important to provide services for massive number of devices for internet of things (IoT) anytime and anywhere [1]. A variety of IoT devices, especially in urban areas, can be effectively supported by the terrestrial networks through advanced system architectures and densely deployed infrastructures. However, most IoT devices are deployed in remote areas, such as deserts, oceans and mountains where they are outside the coverage of terrestrial cellular networks [2].

Satellite communication which can provide ubiquitous connections all over the world is a candidate solution for providing services for the IoT devices in areas without infrastructure coverage [3]. However, it is intractable for current satellite communication systems to serve the delay-sensitive devices, due to their limited communication rate and inherent large latency [4]. In recent years, unmanned aerial vehicles (UAVs) have attracted much attention in the field of wireless communication [5], [6]. Compare with terrestrial base stations or high altitude satellites, due to the advantage of flexible mobility, maneuverability and on-demand deployment, UAV can be used as low-altitude aerial base stations in diverse scenarios [7], [8]. In addition, UAVs are likely to have better communication channels of line-of-sight (LoS) links. Thanks to the above advantages, UAVs can provide high quality and low latency communication services for ground devices. Thus, it is a promising way to integrate UAVs with satellite communication networks to construct satellite-UAV networks for better communication services.

Though satellite-UAV networks can create global seamless access platforms for the ubiquitous IoT devices, spectrum scarcity is a bottleneck that prevents further improvements in service capacity of these networks. Traditional spectrum management methods are inefficient for such flexible and reconfigurable satellite-UAV networks because they allocate spectrum statically and lack adaptability. In fact, under traditional spectrum management methods, a large amount of licensed spectrum is underutilized. However, there is also a barrier when the traditional aerial communications work on unlicensed spectrum bands, where the bands are becoming more and more crowded [9]. Thus, it is necessary to find other available spectrum for these mobile base station like UAVs. Spectrum sharing and dynamic spectrum access are the growing trends for optimized spectrum management and the cognitive radio (CR) approach has already demonstrated its potential to effectively improve spectrum efficiency. In the satellite communication bands, Ka, Ku, S, and C bands are considered to be exploitable for many spectrum sharing scenarios [10]. Specially, according to the International Telecommunication Union Radiocommunication (ITU-R), in Ka band, only 19.7-20.2 GHz for the downlink transmission and 29.5-30 GHz for the uplink transmission are exclusively occupied by satellite communications, the rest Ka band allocated to the satellite communications is shared with terrestrial Fixed Services (FSs) [11]. In this context, introducing CR technique to satellite-UAV networks is an

Manuscript received 7 March 2022; revised 28 August 2022; accepted 9 November 2022. Date of publication 16 November 2022; date of current version 16 January 2023. This work was supported by the National Natural Science Foundation of China under Grant 61571209. The associate editor coordinating the review of this article and approving it for publication was M. C. Gursoy. *(Corresponding author: Xiaohui Zhao.)*

The authors are with the College of Communication Engineering, Jilin University, Changchun, Jilin 130012, China (e-mail: xhzhao@jlu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCOMM.2022.3222460.

Digital Object Identifier 10.1109/TCOMM.2022.3222460

attractive approach for the optimized exploitation of spectrum resources [12].

A. Related Works

Recently, the application of CR to satellite-terrestrial and aerial-terrestrial networks has received a lot of attention and some literatures have made important contributions on these fields. The authors in [13] propose energy-efficient power allocation schemes for cognitive satellite-terrestrial networks and solve the optimization problems by employing statistical delay quality-of-service (QoS) metric. The authors in [14] study the power allocation for the cognitive satellite vehicular network and analyze the characteristics of energy efficiency and spectral efficiency performance in different vehicular environments. Literature [15] investigates the mathematical model for the cognitive low earth orbit satellite constellation with terrestrial networks and proposes two power control schemes.

Meanwhile, there are some literatures focusing on the cognitive aerial-terrestrial networks. Literature [16] presents the performance of a CR enabled UAV network configuration in the underlay mode and develops a heuristic approach to maximize the total throughput. The work of [17] investigates a scenario of spectrum sharing between UAV and terrestrial wireless communication and optimizes the UAV's trajectory and power allocation to improve its communication performance while controlling the co-channel interference at primary receivers (PRs). The authors in [18] provide a framework of a UAV assisted CR network where the UAV harvests RF energy from terrestrial sources and employs non-orthogonal multiple access (NOMA) scheme for transmission. In this literature, the transmit power of the UAV is optimized to maximize throughput. The authors in [19] propose specific learning-aided methods for relay-assisted UAVs in cognitive radio networks (CRNs) to cope with the network destruction in the event of a natural disaster.

Compared with these studies on cognitive satellite-terrestrial and cognitive aerial-terrestrial networks, there are few researches on the cognitive satellite-UAV networks. Literature [20] aims to maximize the achievable rate of the terrestrial user by jointly optimizing BS/UAV transmit power allocation and UAV trajectory in a UAV assisted cognitive satellite-terrestrial network. Different from [20] which only considers a single secondary user (SU) scenario, literatures [21], [22], [23] investigate certain multi-user scenarios for IoT applications. The authors in [21] focus on the cognitive satellite-aerial-terrestrial integrated (SATIN) networks and propose a cooperative beamforming scheme to facilitate secure and energy-efficient IoT communications under some constraints. The authors in [22] are interested in a cognitive SATIN network and solve a problem of joint user association, BS/UAV transmission power and UAV trajectory optimization to maximize the average throughput. The research of [23] proposes a multi-domain resource allocation scheme for cell-free cognitive satellite-UAV networks and develops a process-oriented optimization framework for jointly allocating subchannels, power and hovering times.

The aforementioned works mainly focus on offline policies by applying conventional convex optimization which requires certain prior knowledge about varying environment. However, in the highly dynamic satellite-UAV networks, complete non-casual or statistical information is difficult to obtain or be accurately estimated. It is worth noting that the reinforcement learning (RL) and deep reinforcement learning (DRL) are the effective approaches to solve complex, dynamic and non-convex problems where any prior knowledge is unnecessary [24].

The RL and DRL methods have been successfully applied in the UAV-assisted wireless networks to solve the optimization problems. The authors in [25] focus on a downlink power control problem in an ultra-dense UAV network and develop a DRL-based discrete mean field game algorithm to suppress the interference and maximize the energy efficiency. Literatures [26], [27], [28], [29] use multi-agent DRL based algorithms to solve complicated multi-UAV trajectory optimization problems. The authors in [26] work on a cellular Internet of UAVs and study the age of information (AoI) minimization problem for these UAVs by designing their trajectories. A NOMA aided cellular offloading framework is presented in [27] and a joint three-dimensional (3D) trajectory design and power allocation optimization problem for UAVs is proposed to maximize the throughput. The authors in [28] concentrate on the fairness at user level and formulate a weighted throughput maximization problem by designing UAVs' trajectory. The work of [29] presents a cooperative jamming framework under a multi-agent deep reinforcement learning (MADRL) approach to maximize the secure capacity.

B. Motivation and Contribution

According to our analysis about the existing researches on the cognitive satellite-UAV networks, there is no existing work for the maximization on long-term cumulative reward without the non-causal knowledge, especially to minimize transmission latency, which is an import performance index in these networks. In fact, to realize this minimization, we often require long-term trajectory optimization and resource allocation in a highly dynamic environment. In our considered problem, the trajectory design is coupled with the power allocation, and the decision-making of each UAV is also coupled with those of other UAVs. In general, the movement of UAVs, the dynamics of transmission tasks and the coupling between optimization variables cause a challenging NP-hard problem to be solved by traditional optimization methods.

Moreover, the orthogonal multiple access (OMA) approach is commonly used for these cognitive satellite-UAV networks in the existing researches. Whereas, NOMA technique is adopted in this paper to further improve spectral efficiency. Due to the mobility of the UAVs, the decoding orders of multiple users are dynamic, which also makes it intractable to apply traditional methods to solve this problem.

Motivated by the successful and significant researches on the applications of DRL in the UAV-assisted wireless networks, we try to develop a DRL-based algorithm to solve our complicated and dynamic optimization problem

in the proposed cognitive satellite-UAV network. Importantly, as the number of UAVs and ground users (GUs) increases in this multi-UAV network, the computational complexity will increase if we take centralized RL methods. Additionally, using a centralized algorithm to make optimized decisions will cause excessive information exchange among the UAVs, which takes time and wastes valuable spectrum resources. Thus, the multi-agent RL (MARL) is a possible approach to find optimal policies with low computational complexities and less environment information in this system. The main contributions of this paper are summarized as.

- Different from the OMA-based cognitive satellite-UAV networks in existing works, a NOMA-aided cognitive satellite-UAV framework for IoT applications is proposed. In this framework, the satellite network is the primary user network (PUN) with the priority to use spectrum, while the UAVs are the secondary users (SUs) who share the same spectrum with the PUN and serve the GUs through NOMA. To protect the performance of the PUN from being affected, the total interference from the UAVs to the satellite receiver should be controlled through optimized power allocation and trajectory control.
- Taking the delay-sensitive QoS requirement of the secondary user network (SUN) into consideration, we formulate a joint optimization problem of trajectory control and power allocation for the multi-UAVs to minimize the transmission latency for all the GUs over a long-term task period. Considering the model-free dynamic environment and the coupling between optimization variables, we transform the optimization problem into a partially observable Markov decision process (POMDP) based multi-agent RL problem.
- To reduce the computational complexity and signaling exchange in the execution phase, we propose a multi-agent deep deterministic policy gradient (MADDPG) based joint trajectory control and power allocation (JTCPA) algorithm, which is a multi-agent RL algorithm with centralized training and decentralized execution framework. We design a novel reward function with penalty terms for this algorithm to guide the agents to satisfy the optimization purpose and the constraints. The simulations demonstrate the convergence, validity and superior performance of our proposed solution when compared with other solutions from different perspectives.

The rest of this paper is organized as follows. In Sec. II, the system model of the cognitive satellite-UAV network is presented. Then the proposed MADDPG-based JTCPA algorithm is developed in Sec. III. In Sec. IV, the simulation results are provided to show the performance of our proposed algorithm. Finally, the conclusion is given in Sec. V.

II. System Model

As shown in Fig. 1, we consider a downlink cognitive satellite-UAV system which includes a satellite network acting as a PUN and a multi-UAV network working as a SUN.

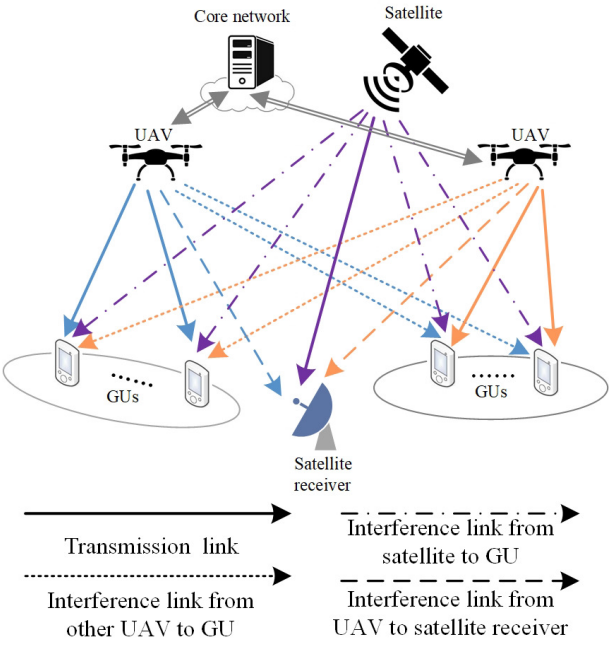


Fig. 1. System model.

The multi-UAV network can share same spectrum with the satellite network to transmit data to N GUs, i.e., the IoT devices. For clear presentation, we denote $n \in \{1, 2, \cdots, N\}$ as the index of the GUs. To protect the communication performance of the satellite network from being affected by the interference of the UAVs, all the UAVs must control their interference to the satellite receiver. The N GUs are distributed in U different areas and the area u is only served by UAV u, where $u \in \{1, 2, \cdots, U\}$. The number of GUs in area u is N_u and $\sum_{u=1}^{U} N_u = N$. The set of GUs in area u is represented as $\mathbf{M}_u$ and the index of a GU in area u is denoted as n_u, $n_u \in \mathbf{M}_u$. Each UAV serves its users through NOMA and the intra-area interference can be subtracted by using successive interference cancellation (SIC). Moreover, since all the UAVs share a spectrum, a UAV will cause interference to the users served by other UAVs. Thus, the inter-area interference also needs to be taken into consideration. To simplify our optimization problem, the task period T of the multi-UAV network is divided into K equal-length time slots and $\tau = T/K$, where τ represents the element time slot. The location change of the UAVs within a time slot can be negligible. Since the positions of the UAVs vary in different time slots, the dynamic decoding orders for each area should be rearranged at each time slot to guarantee the successful SIC. At time slot t, the transmission task data size of user n_u is $D_{n_u}(t)$, which is time-varying, and this delay-sensitive task needs to be accomplished as soon as possible.

A. Mobility and Position Models

A three-dimensional (3D) Cartesian coordinate system is considered in our model. We assume that all the UAVs can adjust their moving direction and flying speed at a certain altitude H_u. At time slot t, the flight direction of UAV u is

determined by the angle of $\theta_u(t) \in (-\pi, \pi]$ and the velocity of $v_u(t) \in [0, v_{\max}]$. The initial coordinates of UAV u are set as $\mathbf{q}_u(0) = [x_u(0), y_u(0), H_u]$. And the coordinate of UAV u at time slot t is denoted as $\mathbf{q}_u(t) = [x_u(t), y_u(t), H_u]$, where

$$x_u(t) = x_u(t-1) + \tau v_u(t) \cos(\theta_u(t)), \quad t = 1, 2, \cdots, K, \tag{1}$$

$$y_u(t) = y_u(t-1) + \tau v_u(t) \sin(\theta_u(t)), \quad t = 1, 2, \cdots, K. \tag{2}$$

In addition, the coordinates of GU n and the satellite receiver are expressed as $\mathbf{w}_n = [x_n, y_n, 0]$ and $\mathbf{w}_r = [x_r, y_r, 0]$ respectively.

B. Channel Models

In this model, we consider a widely used probabilistic path loss model described in [30] that incorporates line-of-sight (LoS) link and non-line-of-sight (NLoS) link. More specifically, the path loss between a UAV and a GU is calculated according to the probability of the LoS link, which relies the elevation angle and environment parameters. The probability of having a line-of-sight (LoS) link between UAV u and GU n at the time slot t is

$$P_{u,n}^{\mathrm{LoS}}(t) = \frac{1}{1 + a \exp(-b(\theta_{u,n} - a))}, \tag{3}$$

where a and b are the constant values depending on working environment, $\theta_{u,n} = \arcsin(H_u / d_{u,n}(t))$ is the elevation angle between UAV u and GU n at time slot t, where $d_{u,n}(t) = \sqrt{\|\mathbf{q}_u(t) - \mathbf{w}_n\|^2}$ is the distance between UAV u and GU n. The probability of NLoS link between UAV u and GU n at the time slot t is

$$P_{u,n}^{\mathrm{NLoS}}(t) = 1 - P_{u,n}^{\mathrm{LoS}}(t), \tag{4}$$

The path loss model for LoS and NLoS links between UAV u and GU n at the time slot t is [16]

$$l_{u,n}(t) = \begin{cases} \eta_{\mathrm{LoS}} d_{u,n}^{-2}(t), & \text{LoS link}, \\ \eta_{\mathrm{NLoS}} d_{u,n}^{-2}(t), & \text{NLoS link}, \end{cases} \tag{5}$$

where η_{LoS} and η_{NLoS} represent the excessive path loss coefficients in LoS and NLoS links with $\eta_{\mathrm{LoS}} > \eta_{\mathrm{NLoS}}$ respectively.

According to Eqn.(3) and Eqn.(5), the average path loss between UAV u and GU n at the time slot t can be expressed by

$$h_{u,n}(t) = \left(P_{u,n}^{\mathrm{LoS}}(t)\eta_{\mathrm{LoS}} + \left(1 - P_{u,n}^{\mathrm{LoS}}(t)\right)\eta_{\mathrm{NLoS}}\right) d_{u,n}^{-2}(t). \tag{6}$$

Similarly, the average path loss between UAV u and the satellite receiver at time slot t can be described by

$$h_{u,r}(t) = \left(P_{u,r}^{\mathrm{LoS}}(t)\eta_{\mathrm{LoS}} + \left(1 - P_{u,r}^{\mathrm{LoS}}(t)\right)\eta_{\mathrm{NLoS}}\right) d_{u,r}^{-2}(t), \tag{7}$$

where $d_{u,r}(t)$ and $P_{u,r}^{\mathrm{LoS}}(t)$ are the distance and the probability of LoS link between UAV u and the satellite receiver at time slot t respectively.

For the communication links between the satellite and the GUs, the LoS model is adopted. Moreover, considering the long distance of the satellite from the earth, the distance between the satellite and GUs, denoted as d_s, is assumed as a constant during the task execution of the UAVs. Thus, the channel power gain from the satellite to the GU n is $h_{s,n} = \eta_0 d_s^{-2}$, where η_0 represents the channel power gain at a reference distance d_0=1m.

C. Transmission and Interference Models

In this NOMA-aided multi-UAV network, UAV u works as a transmitter and non-orthogonally transmits N_u different signals over the same spectrum. The superposition transmit signal $s_u(t)$ of UAV u can be described as

$$s_u(t) = \sum_{n_u=1}^{N_u} \sqrt{P_u^{n_u}(t)} s_u^{n_u}(t), \quad n_u \in \mathbf{M}_u, \tag{8}$$

where $s_u^{n_u}(t)$ represents the normalized transmit signal from UAV u to user n_u and $\mathbb{E}\left[|s_u^{n_u}(t)|^2\right] = 1$, where $\mathbb{E}[\cdot]$ denotes the statistic expectation of the given value. $P_u^{n_u}$ is the allocated power of user n_u.

If we denote the total transmit power of UAV u as P_u, then we have

$$P_u(t) = \sum_{n_u=1}^{N_u} P_u^{n_u}(t). \tag{9}$$

At the receiving ends, each GU (i.e., receiver) in the area u receives the desired signal along with the undesired messages of other GUs in the same area, which causes the intra-area interference. Besides, the GUs also suffer from the interference from the other UAVs and the satellite. Thus, the received signal $z_u^{n_u}$ at GU n_u can be expressed as

$$z_u^{n_u}(t) = \underbrace{\sqrt{h_{u,n_u}(t)} s_u(t)}_{\text{Superimposed signals}} + \underbrace{\sum_{u'=1, u' \neq u}^{U} \sqrt{h_{u',n_u}(t)} s_{u'}(t) + \sqrt{h_{s,n_u}(t) P_s} s_s(t)}_{\text{Interfering signals from outside}} + N_{n_u}(t), \tag{10}$$

where $s_s(t)$ represents the normalized satellite signal and P_s is the transmit power of the satellite which is assumed to be invariant over time slots. $s_{u'}(t)$ is the transmit signal of other UAV u'. $N_{n_u}(t)$ denotes the additive white Gaussian noise (AWGN) at the receiving end with zero mean and variance σ^2. For the sake of clarity, we denote the total interference power from the satellite and the other UAVs to GU n_u as $I_{\mathrm{out}}^{n_u}(t)$ calculated by

$$I_{\mathrm{out}}^{n_u}(t) = \sum_{u'=1, u' \neq u}^{U} h_{u',n_u}(t) P_{u'}(t) + h_{s,n_u} P_s, \tag{11}$$

where $P_{u'}(t)$ is the total transmit power of UAV u' at time slot t. It is worth mentioning that the power loss of the satellite

signal is large due to its long-distance transmission. Therefore, the interference from the satellite, denoted as $h_{s,n_u}P_s$, is usually relatively small.

To perform successful SIC for the intra-area interference, the dynamic decoding orders should be determined by the channel qualities at each time slot. Moreover, since each GU receives the desired signal and intra-area interference over the same spectrum, multiplexing of different signals with different power levels is important to diversify each signal and perform SIC at the GUs. In downlink NOMA, the users with high channel gains are allocated low power whereas the users with low channel gains transmit data with high power levels [31]. Thus, the power allocation should also refer to the channel quality. We define the normalized channel gain from UAV u to user n_u as

$$g_{u,n_u}(t) = \frac{h_{u,n_u}(t)}{I_{\text{out}}^{n_u}(t) + \sigma^2}. \tag{12}$$

We sort $g_{u,n_u}(t)$ in the descending order, i.e., $g_{u,\kappa_u(1)}(t) > g_{u,\kappa_u(2)}(t) \cdots > g_{u,\kappa_u(i)}(t) \cdots > g_{u,\kappa_u(N_u)}(t)$, where $\kappa_u(i)$ represents the sequence number which ranks in the i^{th} and $\kappa_u(i) \in \mathbf{M}_u$, $i \in \{1, 2, \cdots, N_u\}$. Then, the power allocation for all the GUs in the area u needs to meet the following condition,

$$P_u^{\kappa_u(1)}(t) < P_u^{\kappa_u(2)}(t) \cdots < P_u^{\kappa_u(i)}(t) \cdots < P_u^{\kappa_u(N_u)}(t). \tag{13}$$

On the other hand, for the decoding orders, the user $\kappa_u(i)$, $i < N_u$ can decode and successively subtract the signals for all the users from $\kappa_u(i+1)$ to $\kappa_u(N_u)$, whose channel gains are lower than the channel gain of user $\kappa_u(i)$. And the last user $\kappa_u(N_u)$ cannot remove any signals. Thus, the signal to interference plus noise ratio (SINR) at user $\kappa_u(i)$ at time slot t can be expressed as

$$\gamma_u^{\kappa_u(i)}(t) = \begin{cases} \dfrac{P_u^{\kappa_u(i)}(t)\, h_{u,\kappa_u(i)}(t)}{I_{\text{out}}^{\kappa_u(i)} + \sigma^2_{\kappa_u(i)} + \sum_{j=1}^{i-1} P_u^{\kappa_u(j)}(t)\, h_{u,\kappa_u(i)}(t)}, & 1 < i \le N_u, \\ \dfrac{P_u^{\kappa_u(i)}(t)\, h_{u,\kappa_u(i)}(t)}{I_{\text{out}}^{\kappa_u(i)} + \sigma^2_{\kappa_u(i)}}, & i = 1, \end{cases} \tag{14}$$

Accordingly, the transmission rate for user $\kappa_u(i)$ can be calculated by

$$r_u^{\kappa_u(i)}(t) = B\tau \log_2\left(1 + \gamma_u^{\kappa_u(i)}(t)\right), \tag{15}$$

where B represents the spectrum bandwidth and τ is the period of a time slot. Recall that, at time slot t, the transmission task data size of user n_u is $D_{n_u}(t)$. Then, the transmission delay $T_{\kappa_u(i)}$ of user $\kappa_u(i)$ at time slot t is

$$T_{\kappa_u(i)}(t) = \frac{D_{\kappa_u(i)}(t)}{r_u^{\kappa_u(i)}(t)}, \tag{16}$$

where $D_{\kappa_u(i)}(t)$ represents the data size of the transmission task from UAV u to user $\kappa_u(i)$. Finally, the total latency of all the users served by UAV u at time slot t is

$$T_u(t) = \sum_{i=1}^{N_u} T_{\kappa_u(i)}(t). \tag{17}$$

D. Problem Formulation

As discussed above, the objective of this paper is to minimize the total transmission latency for N GUs by jointly optimizing the trajectories and powers of the UAVs over a task period. The trajectory of each UAV is determined by its flight speed and angle.

Let $\mathbf{V}(t) = \{v_u(t),\ u = 1, 2, \cdots, U\}$ and $\boldsymbol{\theta}(t) = \{\theta_u(t),\ u = 1, 2, \cdots, U\}$ denote the set of speeds and the set of flight angles of all UAVs respectively. Meanwhile, the power allocation policy for UAV u is denoted as $\mathbf{P}_u(t) = \{P_u^{n_u}(t),\ n_u = 1, 2, \cdots, N_u\}$. Accordingly, the set of the power allocation policies for all UAVs is $\mathbf{P}(t) = \{\mathbf{P}_u(t),\ u = 1, 2, \cdots, U\}$. Thus, the total latency minimization problem is formulated as

$$\mathbf{OP1}: \min_{\mathbf{V}(t), \boldsymbol{\theta}(t), \mathbf{P}(t)} \sum_{t=1}^{K} \sum_{u=1}^{U} T_u(t), \tag{18-A}$$

$$\text{s.t. } v_u(t) \in [0, v_{\max}],\ \forall u, t, \tag{18-B}$$

$$\boldsymbol{\theta}_u(t) \in (-\pi, \pi],\ \forall u, t, \tag{18-C}$$

$$\mathbf{q}_u(t) \in \mathbf{Q}_u,\ \forall u, t, \tag{18-D}$$

$$\gamma_u^{n_u}(t) \ge \gamma_{\min},\ \forall n_u \in \mathbf{M}_u,\ \forall u, t, \tag{18-E}$$

$$\sum_{u=1}^{U} P_u(t)\, h_{u,r}(t) \le I_{th},\ \forall t, \tag{18-F}$$

$$\sum_{n_u=1}^{N_u} P_u^{n_u}(t) = P_u(t) \le P_{\max}, \quad \forall n_u \in \mathbf{M}_u, \forall u, t, \tag{18-G}$$

$$P_u^{\kappa_u(1)}(t) < P_u^{\kappa_u(2)}(t) \cdots < P_u^{\kappa_u(i)}(t) \cdots < P_u^{\kappa_u(N_u)}(t),\ \forall u, t. \tag{18-H}$$

where (18-B) and (18-C) are the constraints of the flight speed and angle of each UAV. $\mathbf{Q}_u$ denotes the flight area of UAV u and the constraint (18-D) restricts the UAVs to travel in a designated area. Constraint (18-E) requires that the SINR at the receiving end is greater than or equal to the minimum standard. (18-F) requires that the total interference from all UAVs to the satellite receiver is lower than the interference temperature threshold I_{th}. Constraint (18-G) guarantees that the transmit power of each UAV does not exceed the maximum power level. Constraint (18-H) ensures that the power allocation policies meet the SIC condition when $g_{u\kappa_u(1)}(t) > g_{u\kappa_u(2)}(t) \cdots > g_{u\kappa_u(i)}(t) \cdots > g_{u\kappa_u(N_u)}(t)$, which is analyzed in Sec. II-C.

III. MARL-Based Solution

It is difficult to solve the above optimization problem **OP1** using the traditional methods due to the following reasons,

1) The long-term trajectory and power optimization in an unknown and highly dynamic environment is an intricate problem. In this paper, we assume that the transmission tasks

of the UAVs change dynamically and the prior knowledge of the environment cannot be obtained in advance. In such a model-free environment, the traditional model based optimization methods cannot well deal with this kind of optimization problem.

2) Generally, certain trajectory and transmit power of UAVs can be jointly optimized by using the alternating iterative algorithm, which decomposes the problem into several sub-problems and solves the sub-problems alternately until the algorithm converges [32], [33]. However, in our formulated problem, the decision-making of each UAV is coupled with other UAVs due to the interference among them and the interference constraint of the satellite network. When the numbers of UAVs and GUs increase, this non-convex problem would become difficult to solve and the traditional iterative algorithm may not converge.

3) Because of the mobility of UAVs, the decoding orders need to be re-determined in each time slot for efficient SIC.

For the above reasons, we should try to use a new method to find our solution for this optimization problem. RL, as a model-free method, is an effective tool to learn hidden patterns in an unknown environment via trial-and-error [34] and does not need the convexity requirement for complicated problem optimization. In the scenario of multiple UAVs and GUs considered in this paper, using the centralized methods to solve the problem may result in an expensive computational complexity and information exchange. To reduce this computational complexity and ease the information exchange burden among UAVs, a MADDPG-based algorithm with a centralized training and decentralized execution framework is proposed in this section. By this way, each UAV can quickly make decisions based on its own observation and does not need to communicate with other UAVs, which saves time and spectrum and improves resource efficiency. In order to well present the detail development of the proposed method, some necessary principles of DRL used in this paper is provided. Then, a POMDP-based MARL model is investigated. Finally, we propose a MADDPG-based JTCPA algorithm to find the optimal solution for our joint trajectory and power optimization problem.

A. Necessary Principles of DRL

In a RL process, an agent learns to find an optimal strategy to maximize a long-term reward by interacting with the working environment without complete knowledge *a-priori*. Typically, RL is developed on the Markov decision process (MDP) which is a stochastic behavior under discrete time. A MDP consists of a four-tuple, $MDP = (\mathcal{S}, \mathcal{A}, \mathcal{P}_{sa}, \mathcal{R})$ where $\mathcal{S}$ denotes a state space, $\mathcal{A}$ represents an action space, $\mathcal{R}$ is a immediate reward function, and $\mathcal{P}_{sa}$ is the transition probability. According to this MDP model, an agent learns an optimal policy $\pi(s) \in \mathcal{A}$ corresponding to an arbitrary state in $\mathcal{S}$ to maximize an expected discounted cumulative reward defined as

$$G_t = r_t + \beta r_{t+1} + \beta^2 r_{t+2} + \cdots = \sum_{k=t}^{\infty} \beta^{k-t} r_k. \quad (19)$$

Thus, the goal of a RL is to find an optimal π^* to maximize $\mathbb{E}_\pi(G_t)$ where $\mathbb{E}_\pi(\cdot)$ denotes the expected given value by which the agent follows policy π.

To find π^*, a state action value function $Q(s,a)$ called Q-value is introduced to estimate the expected discounted cumulative reward by executing an action a at an environmental state s under policy π, which is dexcribed by

$$Q(s,a) = \mathbb{E}_\pi[G_t]. \quad (20)$$

Take the advantage of the properties of recursion relationships of $Q(s,a)$, we can obtain the recursive expression called the Bellman equation given by

$$\begin{aligned} Q(s,a) &= \mathbb{E}_\pi\left[r_t + \beta G_{t+1} | s_t = s, a_t = a\right], \\ &= \mathbb{E}_\pi\left[r_t + \beta Q^\pi(s_{t+1}, a_{t+1}) | s_t = s, a_t = a\right]. \end{aligned} \quad (21)$$

The deep Q-network (DQN) [35] is a classical DRL algorithm to find optimal policies in a discrete action space. However, it is challenging to efficiently solve the optimization problems with continuous values. Fortunately, an actor-critic framework based deep deterministic policy gradient (DDPG algorithm) which combines DQN and the deterministic policy gradient (DPG) method can better handle the continuous action and state space for a complex system [36], since it well adopts some designed neural networks to estimate both policy and action value functions. This DDPG algorithm is a centralized DRL method which requires global information to make its decisions. In this paper, motivated by the idea of a decentralized MADDPG [37] algorithm derived from the DDPG algorithm, we propose our problem oriented algorithm to solve the multi-agent optimization problem in continuous action and state space. By applying this algorithm, each agent can make decisions according to their partially information observation instead of global one.

B. POMDP-Based MARL Model

In this subsection, we transform the formulated problems **OP1** into a multi-agent extension of MDPs called partially observable Markov games for U agents, i.e., the U UAVs. A Markov game for U agents is defined by a set of states $\mathcal{S}$, a set of observations $\mathcal{O} = \left\{\mathbf{O}^1, \cdots \mathbf{O}^u, \cdots \mathbf{O}^U\right\}$, a set of actions $\mathcal{A} = \left\{\mathbf{A}^1, \cdots \mathbf{A}^u, \cdots \mathbf{A}^U\right\}$ and a set of immediate rewards $\mathcal{R} = \left\{r^1, \cdots r^u, \cdots, r^U\right\}$. The state space $\mathcal{S}$ describes the possible configurations of the cognitive satellite-UAV network. $\mathbf{O}^u$ is the observation space for UAV u and the observation $\mathbf{O}_t^u$ at time slot t is a part of the current state $\mathbf{S}_t$. $\mathbf{A}^u$ and r^u are the action space and immediate reward for UAV u respectively. Considering the unknown statistical knowledge on the model free environment in our problem, the state transition probability $\mathcal{P}_{sa}$ in the typical four tuple of a MDP is unknown. Thus, $\mathcal{P}_{sa}$ is not included in this POMDP. The detailed physical meanings of $\mathcal{S}$, $\mathcal{O}$, $\mathcal{A}$, $\mathcal{R}$ are as follows.

State Space $\mathcal{S}$: The state space includes the positions of all UAVs and the time-varying tasks for all the GUs. We denote the data size of transmission task from UAV u to user n_u at time slot t as $D_{n_u}(t)$. Then the set of transmission task of UAV u at time slot t can be

described as $\mathbf{D}_u(t) = \{D_{n_u}(t) | n_u \in \mathbf{M}_u\}$. Recalling that $\mathbf{q}_u(t)$ represents the position coordinates of UAV u, we can define our current state of the whole system as $\mathbf{S}_t = \{\mathbf{q}_u(t), \mathbf{D}_u(t) | u = 1, 2, \cdots, U\}$.

Observation space $\mathcal{O}$: Since there is no information exchange between the UAVs, each UAV can only observe its own position and task volume. The observation of UAV u at time slot is denoted as $\mathbf{O}_t^u = \{\mathbf{q}_u(t), \mathbf{D}_u(t)\}$. In our POMDP, it is obvious that the observations of all the UAVs constitute the global state of the system, i.e., $\mathbf{S}_t = \{\mathbf{O}_t^1, \cdots \mathbf{O}_t^u, \cdots \mathbf{O}_t^U\}$.

Action space $\mathcal{A}$: Each UAV makes its own local decision on trajectory control and power allocation according to its own observation. As mentioned above, the trajectory of UAV u is determined by the angle $\theta_u(t)$ and velocity $v_u(t)$. Meanwhile, the power allocation policy for the GUs served by UAV u is $\mathbf{P}_u(t) = \{P_u^{n_u}(t) | n_u \in \mathbf{M}_u\}$. Thus, the action for UAV u at time slot t is defined as $\mathbf{A}_t^u = \{\theta_u(t), v_u(t), \mathbf{P}_u(t)\}$ and the action space for all the UAVs is $\mathbf{A}_t = \{\mathbf{A}_t^u | u = 1, 2, \cdots, U\}$.

Immediate reward $\mathcal{R}$: After all the UAVs taking actions in state $\mathbf{S}_t$, the system will transfer to the next state $\mathbf{S}_{t+1}$, meanwhile agent u will get an immediate reward r_t^u. In our system, we consider the situation of a cooperative game where all the UAVs serve the common purpose to minimize the global transmission latency, i.e., $r_t^1 \cdots = r_t{}^u \cdots = r_t{}^U = r_t$. Thus, we define r_t as our immediate reward function for all the UAVs.

The immediate reward function should evaluate the effect of the actions taken for the system performance. In this system, we minimize the total transmission latency over a task period on the premise that the interference of all the UAVs to the satellite receiver is not greater than an allowable interference threshold I_{th}. If the total interference exceeds I_{th}, these actions are considered to compromise the system performance and the agent should be punished. Therefore, the immediate reward should include both the transmission delay of all the GUs and a penalty term $r_p(t)$ defined by

$$r_p(t) = \begin{cases} 0, & \text{if } \sum_{u=1}^{U} P_u(t)\, h_{ur}(t) \le I_{\text{th}}, \\ \chi_1 (I_{\text{th}} - \sum_{u=1}^{U} P_u(t)\, h_{ur}(t)) & \\ \quad + \chi_2 r_{\text{pen}}, & \text{otherwise}, \end{cases} \tag{22}$$

where $(I_{\text{th}} - \sum_{u=1}^{U} P_u(t)\, h_{u,r}(t))$ is the dynamic penalty term to guide the agent to learn how to reduce the interference at the PUN and r_{pen} is the static penalty term to make the actions satisfy the constraint (18-F). χ_1 and χ_2 are the penalty weights which are positive numbers that adapt to the target optimization equation. Then, a comprehensive immediate reward function is defined as

$$r_t = -\sum_{u=1}^{U} T_u(t) + r_p(t). \tag{23}$$

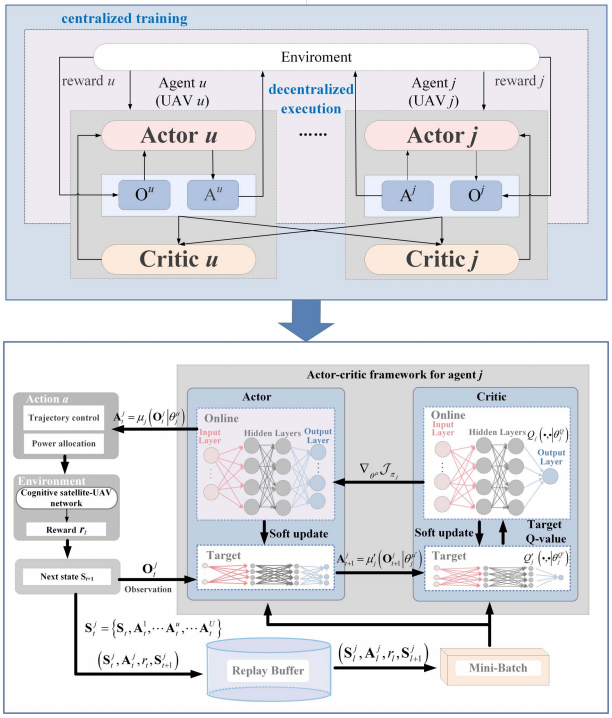


Fig. 2. Structure of MADDPG-based JTCPA algorithm.

Thus, the discount cumulative reward of our problem can be calculated by

$$\begin{aligned} G_t &= \sum_{k=t}^{K} \beta^{k-t} r_k, \\ &= \sum_{k=t}^{K} \beta^{k-t} \left(-\sum_{u=1}^{U} T_u(k) + r_p(k) \right). \end{aligned} \tag{24}$$

Based on the above POMDP model, we transform **OP1** into a MARL framework-based optimization problem,

$$\begin{aligned} \mathbf{OP2}: \max_{\mathbf{V}(t), \theta(t), \mathbf{P}(t)} & \mathbb{E}_\pi [G_t], \quad t \in \{1, 2, \cdots, K\}, \\ \text{s.t. } & \text{(18-B)} - \text{(18-E) and (18-G)} - \text{(18-H)}. \end{aligned} \tag{25}$$

C. MADDPG-Based Solution

In the following subsection, we will propose a MADDPG-based JTCPA algorithm to find our optimal policy. In our proposed algorithm, each UAV acts as an agent to determine its own trajectory planning and power allocation. If we directly apply the single-agent DRL algorithm for each UAV in this multi-agent scenario, each agent will learn policy independently and ignore the actions and states of other agents. In this case, the environment becomes non-stationary from the perspective of any individual agent. This will lead to an unstable learning process. To solve this problem, we adopt the MADDPG-based algorithm with centralized training and distributed execution framework which is allowed to use extra information to ease training presure. The structure and the workflow of our proposed algorithm is shown in Fig. 2.

In our MADDPG-based JTCPA algorithm, each agent has neural networks with actor-critic framework presented in the bottom of Fig. 2. The actor-critic framework contains an actor network and a critic network. The actor network is adopted to approximate the action policy and output certain actions according to its own observation at current time slot. The critic network with global information input is used to approximate the state action value and evaluate performance for the action. Normally, each critic network or actor network consists of an online network and a target network. These two subnetworks have same structure and different updated rate parameters. The purpose of establishing these target networks is to make the learning process of the online networks stable and convergent.

For agent j, its neural network parameters of the online subnet and the target subnet in the actor network are defined as θ_j^{μ} and $\theta_j^{\mu'}$ respectively. Similarly, the parameters of agent j in the critic network are denoted as θ_j^{Q} and $\theta_j^{Q'}$ correspondingly. At time step t, the actor network outputs action value according to the observation of the agent. We define $\mu_j(\mathbf{O}_t^j|{\theta_j}^{\mu})$ as the action policy function with parameters ${\theta_j}^{\mu}$ for agent j. Then, the action output by this actor network of agent j is

$$\mathbf{A}_t^j = \mu_j\left(\mathbf{O}_t^j|\theta_j^{\mu}\right). \tag{26}$$

In order to explore the optimal action under this observation, we take an exploration policy by adding a noise N_t to our actor policy, where N_t is chosen to adapt to the environment. Thus, the actions actually performed by UAV j is described as

$$\mathbf{A}_t^j = \mu_j\left(\mathbf{O}_t^j|\theta_j^{\mu}\right) + N_t. \tag{27}$$

Since each agent aims to maximize the expected long term cumulative reward by finding an optimal action policy, the policy objective function can be denoted as $\mathcal{J}_{\pi_j}\left(\theta_j^{\mu}\right) = \mathbb{E}_{\theta_j^{\mu}}\left[G_t\right]$, where π_j represents the action policy for agent j. Hence, the optimal action policy π_j^* can be obtained by finding θ_j^{μ} to maximize $\mathcal{J}_{\pi_j}\left(\theta_j^{\mu}\right)$, i.e.,

$$\pi_j^* = \arg\max_{\theta_j^{\mu}} \mathcal{J}_{\pi_j}\left(\theta_j^{\mu}\right). \tag{28}$$

As mentioned in Sec.III-A, a state action value function Q-value is used to evaluate the expected discounted cumulative reward $\mathbb{E}_{\pi}\left[G_t\right]$. Considering the relevance between the agents, in the MADDPG algorithm, the Q-value of agent j is not only related to its own action and observation, but also related to the observations and actions of other agents. Therefore, in the centralized training, in addition to the local observation, some extra information, i.e., the observations and actions of all other agents, are also available for an arbitrary agent. In this way, the Q-value of agent j at time slot t can be denoted as $Q_j\left(\mathbf{O}_t^1, \cdots \mathbf{O}_t^j, \cdots \mathbf{O}_t^U, \mathbf{A}_t^1, \cdots \mathbf{A}_t^j, \cdots \mathbf{A}_t^U\right)$. Since in our POMDP, $\mathbf{S}_t = \left\{\mathbf{O}_t^1, \cdots \mathbf{O}_t^u, \cdots \mathbf{O}_t^U\right\}$, the corresponding Q-value can be denoted as

$$\begin{aligned} &Q_j\left(\mathbf{O}_t^1, \cdots \mathbf{O}_t^j, \cdots \mathbf{O}_t^U, \mathbf{A}_t^1, \cdots \mathbf{A}_t^j, \cdots \mathbf{A}_t^U\right) \\ &\quad = Q_j\left(\mathbf{S}_t, \mathbf{A}_t^1, \cdots \mathbf{A}_t^j, \cdots \mathbf{A}_t^U\right). \end{aligned} \tag{29}$$

For clarity, we define a set $\mathbf{S}_t^j$ including the global states and actions of all other agents for agent j as

$$\mathbf{S}_t^j = \left\{\mathbf{S}_t, \mathbf{A}_t^1, \cdots \mathbf{A}_t^u, \cdots \mathbf{A}_t^U\right\}, \tag{30}$$

where u denotes the index of an arbitrary agent except for agent j, i.e., $\forall u \in \{1, 2, \cdots, U\},\ u \neq j$. Then, according to [38], the expression form of the Q-value for agent j can be simplified as

$$Q_j\left(\mathbf{S}_t, \mathbf{A}_t^1, \cdots \mathbf{A}_t^j, \cdots \mathbf{A}_t^U\right) = Q_j\left(\mathbf{S}_t^j, \mathbf{A}_t^j\right). \tag{31}$$

Similarly, we also define a set $\mathbf{S}_{t+1}^j$ for agent j as

$$\mathbf{S}_{t+1}^j = \left\{\mathbf{S}_{t+1}, \mathbf{A}_{t+1}^1, \cdots \mathbf{A}_{t+1}^u, \cdots \mathbf{A}_{t+1}^U\right\}. \tag{32}$$

where $\mathbf{A}_{t+1}^u = \mu_u'\left(\mathbf{O}_{t+1}^u \middle| \theta_u^{\mu'}\right)$ and $\mu_u'(\cdot)$ stand for the two outputs of the target subnet of the actor network of an arbitrary agent u.

Generally, for the problem of finding the optimal action policy, the gradient ascent method is often adopted to find the solution. According to the defined Q-value, the gradient of $\mathcal{J}_{\pi_j}\left(\theta_j^{\mu}\right)$ with respect to θ_j^{μ} is calculated by

$$\begin{aligned} &\nabla_{\theta^{\mu}} \mathcal{J}_{\pi_j} \\ &= \mathbb{E}_{\theta_j^{\mu}}\left[\nabla G_t\right], \\ &= \mathbb{E}_{\theta_j^{\mu}}\left[\nabla_{a_j} Q_j\left(\mathbf{S}_t^j, \mathbf{A}_t^j \middle| \theta_j^Q\right)\Big|_{\mathbf{A}_t^j = \mu_j\left(\mathbf{O}_t^j|\theta_j^{\mu}\right)} \nabla_{\theta_j^{\mu}} \mu\left(\mathbf{O}_t^j\left|\theta_j^{\mu}\right.\right)\right], \end{aligned} \tag{33}$$

where $\nabla_{\theta_j^{\mu}} \bullet$ represents the gradient vector of a function $\bullet$ with respect to θ_j^{μ}. $Q_j\left(\mathbf{S}_t^j, \mathbf{A}_t^j \middle| \theta_j^Q\right)$ is the state action value of agent j which is calculated by its online subnet of the critic network with θ_j^Q. Then, these gradients are back propagated to the online subnet of actor network to update θ_j^{μ} by

$$\theta_j^{\mu} \leftarrow \theta_j^{\mu} + \varsigma \nabla_{\theta_j^{\mu}} \mathcal{J}_{\pi_j}, \tag{34}$$

where $\varsigma \in (0, 1]$ represents the learning rate of the online subnet of the actor network.

To update the online subnet of the critic network, a temporal difference error is used, i.e.,

$$\mathcal{L}(\theta_j^Q) = \mathbb{E}_{\theta_j^Q}\left[\left(Y_t^j - Q\left(\mathbf{S}_t^j, \mathbf{A}_t^j \middle| \theta_j^Q\right)\right)^2\right], \tag{35}$$

where $Y_t^j = r_t + \beta Q'\left(\mathbf{S}_{t+1}^j, {\mu'}_j\left(\mathbf{O}_{t+1}^j \middle| \theta_j^{\mu'}\right) \middle| \theta_j^{Q'}\right)$ stands for the target value. $Q_j'(\cdot)$ and $\mu_j'(\cdot)$ are the outputs of the target subnet of the critic network and the actor network of agent j respectively.

Then, with the differential operation on $\mathcal{L}\left(\theta_j^Q\right)$, we can get

$$\begin{aligned} \nabla_{\theta_j^Q} \mathcal{L}(\theta_j^Q) = -2 \times \mathbb{E}_{\theta_j^Q}\Big[&r_t + \beta Q'\left(\mathbf{S}_{t+1}, \mathbf{A}_{t+1} \middle| \theta_j^{Q'}\right) \\ &- Q\left(\mathbf{S}_t, \mathbf{A}_t \middle| \theta_j^Q\right)\Big] \cdot \nabla_{\theta_j^Q} Q\left(\mathbf{S}_t, \mathbf{A}_t \middle| \theta_j^Q\right), \end{aligned} \tag{36}$$

where $\nabla_{\theta_j^Q} \bullet$ represents the gradient vector of a function $\bullet$ with respect to θ_j^Q.

Defining ξ as the learning rate of the critic network, we can update θ_j^Q by

$$\theta_j^Q \leftarrow \theta_j^Q - \xi \nabla_{\theta_j^Q} \mathcal{L}. \tag{37}$$

The weights of these two target networks of agent j are then updated by making them slowly track the learned online networks, i.e.

$$\theta_j^{\mu'} \leftarrow \lambda \theta_j^{\mu'} + (1-\lambda)\,\theta_j^{\mu'}, \tag{38}$$

$$\theta_j^{Q'} \leftarrow \lambda \theta_j^{Q'} + (1-\lambda)\,\theta_j^{Q'}, \tag{39}$$

where λ is the update rate of the target networks.

In the actual training process, an experience replay memory $\mathcal{D}_j$ with a capacity of $\mathcal{C}$ is introduced to update the networks [39]. We keep tracking the experience in the replay memory with the state transition sample data $\left(\mathbf{S}_t^j, \mathbf{A}_t^j, r_t, \mathbf{S}_{t+1}^j\right)$ at each time slot for agent j. In each training epoch, we randomly draw X sets of experience data from $\mathcal{D}_j$ as a minibatch of the samples, where an arbitrary set of experience data in the minibatch is denoted as $\left(\mathbf{S}_l^j, \mathbf{A}_l^j, r_l, \mathbf{S}_{l+1}^j\right)$. In this way, (33) and (36) can be approximately obtained by

$$\nabla_{\theta_j^Q} \mathcal{J}_{\pi_j} \approx \frac{1}{X} \sum_l \left[\nabla_{a_j} Q_j \left(\mathbf{S}_l^j, \mu_j \left(\mathbf{O}_t^j \left| \theta_j^\mu \right. \right) \middle| \theta_j^Q \right) \nabla_{\theta_j^\mu} \mu \left(\mathbf{O}_t^j \left| \theta_j^\mu \right. \right) \right], \tag{40}$$

$$\nabla_{\theta_j^Q} \mathcal{L}(\theta_j^Q) \approx -\frac{2}{X} \sum_l \left[r_l + \beta Q' \left(\mathbf{S}_{l+1}, \mathbf{A}_{l+1} \middle| \theta_j^{Q'} \right) - Q \left(\mathbf{S}_l, \mathbf{A}_l \middle| \theta_j^{Q'} \right) \right] \cdot \nabla_{\theta_j^Q} Q \left(\mathbf{S}_l, \mathbf{A}_l \middle| \theta_j^Q \right) \tag{41}$$

It is worth mentioning that since UAVs have limited energy capacity and computation ablitiy, the centralized training process is operated at the core network by utilizing the information uploaded by all the UAVs. In the distributed execution process, each UAV, as the agent, downloads the already trained neural network weights from the core network and loads the weights to its own actor network. Each UAV makes its decisions according to its own observations and does not need to communicate with other UAVs which saves spectrum and time and reduces complexity.

We summarize our proposed MADDPG-based JTCPA algorithm in Table I and Table II for the training phase and execution phase respectively.

TABLE I

MADDPG-BASED JOINT TRAJECTORY CONTROL AND POWER ALLOCATION ALGORITHM (TRAINING PHASE)

JTCPA algorithm (training phase)

Initialize experience replay memory $\mathcal{D}_j$ with size C for each agent.
Initialize minibatch size with X.
Initialize θ_j^Q and θ_j^μ with random weights and initialize $\theta_j{}^{Q'}$ and $\theta_j^{\mu'}$ with $\theta_j^{Q'} \leftarrow \theta_j^Q$ and $\theta_j^{\mu'} \leftarrow \theta_j^\mu$ for each agent respectively.
for $episode$ = 1, 2, ..., F **do**
  Initialize environment and obtain environment initial state $\mathbf{S}_0$.
  for t = 1, 2, ..., K **do**
    for agent j = 1 to U, **do**
      Select action $\mathbf{A}_t^j = \mu_j \left(\mathbf{O}_t^j | \theta_j^\mu \right) + N_t$ according to local observation $\mathbf{O}_t^j$.
    end for
    Execute action $\mathbf{A}_t = \{\mathbf{A}_t^1, \cdots \mathbf{A}_t^\mu, \cdots \mathbf{A}_t^U\}$ and receive immediate reward r_t and obtain $\mathbf{S}_{t+1}$.
    for agent j = 1 to U, **do**
      Store transition data $\left(\mathbf{S}_t^j, \mathbf{A}_t^j, r_t, \mathbf{S}_{t+1}^j\right)$.
      if $\mathcal{D}_j$ is full, **do**
        Sample a random minibatch of X transition data $\left(\mathbf{S}_l^j, \mathbf{A}_l^j, r_l, \mathbf{S}_{l+1}^j\right)$ from $\mathcal{D}_j$.
        Update online network of critic by (34) with (38).
        Update online network of actor by (31) with (37).
        Update two target networks by (35) and (36).
      end for
    end for
  end for
end for

TABLE II

MADDPG-BASED JOINT TRAJECTORY CONTROL AND POWER ALLOCATION ALGORITHM (EXECUTION PHASE)

JTCPA algorithm (execution phase)

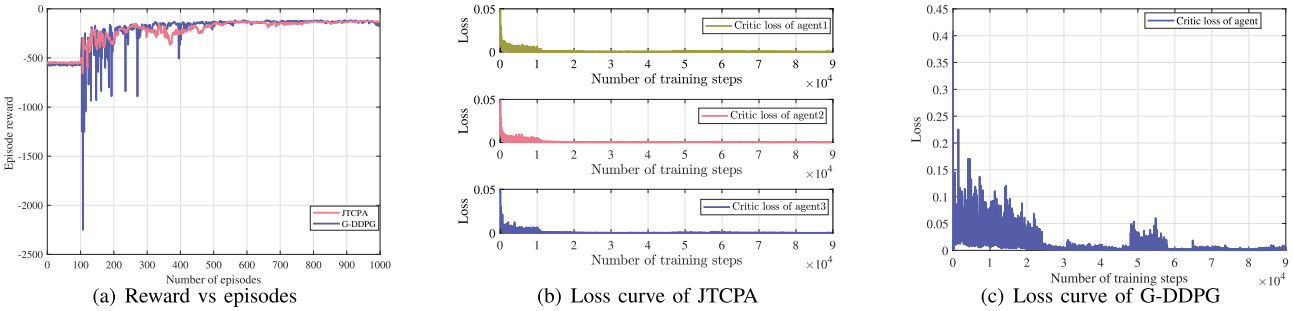
Initialize environment.
for t = 1, 2, ..., K **do**
  for agent j = 1 to U, **do**
    Select action $\mathbf{A}_t^j = \mu_j \left(\mathbf{O}_t^j | \theta_j^\mu \right)$ according to local observation $\mathbf{O}_t^j$.
    Execute action $\mathbf{A}_t^j$.
    Store $\mathbf{O}_t^j$ and $\mathbf{A}_t^j$ for centralized training.
  end for
end for

D. Algorithm Complexity Analysis

Usually, the total algorithm complexity of a deep neural network with fully connected layers is determined by the number of operations of the network model, which includes the dimensions of the input, the number of neurons in each layer and the number of neural network layers. We first analyze the complexity of the algorithm during the training phase. In this phase, both the actor and critic networks are considered. The layer numbers of the actor and critic neural networks are denoted by F^a and F^c respectively. In the $f-$th layer, n_f^a and n_f^c denotes the neuron numbers in the actor and critic networks respectively. Then the total algorithm complexity of the training phase can be calculated by $\mathcal{O}_{com}\left(\sum_{f=0}^{F^a-1} n_f^a n_{f+1}^a + \sum_{f=0}^{F^c-1} n_f^c n_{f+1}^c\right)$, which increases with the number of UAVs and GUs. During the distributed execution phase, each agent employs its actor network to output an action. Thus, the complexity of the algorithm during this phase is only related to the number of operations of the actor network, which is $\mathcal{O}_{com}\left(\sum_{f=0}^{F^a-1} n_f^a n_{f+1}^a\right)$.

IV. SIMULATION RESULTS

In this section, the performance of the proposed MADDPG-based JTCPA algorithm will be presented through the simulation results. We consider the case there are three UAVs serving three adjacent square areas in the simulations,



(a) Reward vs episodes (b) Loss curve of JTCPA (c) Loss curve of G-DDPG

Fig. 3. Convergence performance, $I_{\rm th} = 3 \times 10^{-9} W$.

TABLE III
SIMULATION PARAMETERS

Parameters	Value
Maximum flying speed of UAV $v_{\max}$	30m/s
Maximum transmit power of UAV $P_{\max}$	5W
Flying altitude of UAV $B_{\rm init}$	100m
Bandwidth	4MHz
Duration of a time slot τ	1s
Task duration T	40s~180s
Interference threshold $I_{\rm th}$ of PUN	0.05×10^{-9}W~3×10^{-9}W
Transmission task volume D_{n_u}	[0.5, 2] Mbit
Noise Power σ^2	-110dBW
Channel coefficients $(a, b, \eta_{\rm LoS}, \eta_{\rm NLoS})$	(9.61, 0.16, −41.4dB, −60.4dB)

TABLE IV
NEURON COMPARISON BETWEEN JTCPA AND G-DDPG ALGORITHM

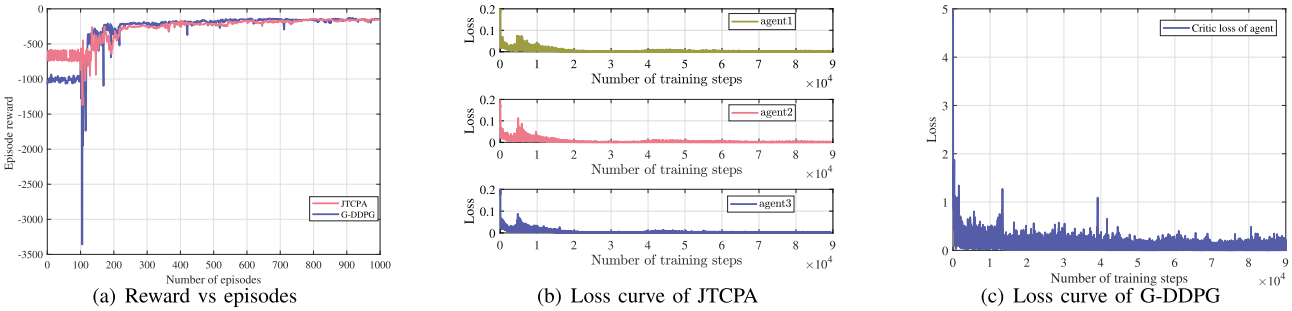
Item		JTCPA	G-DDPG
Number of agents		3	1
Number of neurons in the critic network	Input layer	42	42
	First hidden layer	200	600
	Second hidden layer	50	100
	Output layer	1	1
Number of neurons in the actor network	Input layer	7	21
	First hidden layer	200	600
	Secondary hidden layer	100	100
	Output layer	7	21

and each area is $1000m \times 1000m$. There are 4 randomly distributed GUs in each area. The satellite receiver is deployed in the center of these three areas. The simulation parameters are given in Table III.

For the neural networks of this proposed algorithm, there are two hidden layers in both the actor and critic networks. In the actor network, the first hidden layer contains 200 neurons and the second hidden layer contains 100 neurons. And the actor network uses a fully connected layer with the Tanh activation function for the final outputs. The critic network has 200 neurons and 50 neurons for the first and second hidden layer respectively. We train the deep neural networks with a minibatch size of 32 and the replay memory capacity is 10000. We set 0.00005 and 0.0001 as the learning rate for the actor and critic network respectively.

To evaluate the performance of the proposed JTCPA algorithm, we use a global DDPG algorithm as a benchmark which is referred as G-DDPG algorithm in the following. The G-DDPG algorithm, which is a centralized DRL method, determines the trajectory planning and power allocation according to the global state, instead of the local observation in the MADDPG-based algorithm. To apply the G-DDPG algorithm in the considered system, a lead UAV is required in the execution phase. At the beginning of each time slot, ordinary UAVs transmit their local observation information to the lead UAV. Then the lead UAV decides the actions of all the UAVs according to the collected global information and broadcasts the decision results to the ordinary UAVs. This information exchange occupies time resources of the transmission task. However, in the simulation of this G-DDPG algorithm, we do not consider this signaling transmission delay. In the G-DDPG algorithm, the actor and critic networks both consist of two hidden layers with 600 neurons in the first layer and 100 in the second layer. We summarize the used neurons comparison between the JTCPA and G-DDPG algorithm in Table IV. From Table IV, we can see that since the operation of the JTCPA algorithm is distributed on multiple agents, its computational complexity is lower than the centralized G-DDPG algorithm.

First, we present the convergence properties of our JCARA algorithm compared with G-DDPG algorithm in Fig. 3 and Fig. 4, where Fig. 3 and Fig. 4 are the convergence performances under $I_{\rm th} = 3 \times 10^{-9} W$ and $I_{\rm th} = 0.5 \times 10^{-9} W$ respectively. We set 1000 episodes with 100 time steps in each episode in the training phase. In the first 100 episodes, we do not train the neutral networks, the purpose is to get enough experience in the replay buffers for learning. Fig. 3(a) and 4(a) are the achieved episode reward comparisons between the JTCPA algorithm and G-DDPG algorithm. It is observed that under different $I_{\rm th}$, the achieved episode rewards of the JTCPA algorithm are more stable than those of G-DDPG. Fig. 3(b) and 3(c) are the loss curves of the critic networks of the JTCPA algorithm and G-DDPG algorithm under $I_{\rm th} = 3 \times 10^{-9} W$ respectively. Similarly, fig. 4(b) and 4(c) are the loss curves of the two algorithms under $I_{\rm th} = 0.5 \times 10^{-9} W$ respectively. From these loss curves, it can be seen that each agent of the JTCPA algorithm has a satisfactory rate of convergence and the loss curves of the JTCPA algorithm are more stable than those of the G-DDPG algorithm. The results in Fig. 3 and Fig. 4 illustrate that the convergence performance of the JTCPA algorithm is more stable than that of the G-algorithm. This is because in the G-DDPG algorithm, there are more state and action dimensions than those in the MADDPG-based JTCPA algorithm,



(a) Reward vs episodes (b) Loss curve of JTCPA (c) Loss curve of G-DDPG

Fig. 4. Convergence performance, $I_{\rm th} = 0.5 \times 10^{-9} W$.

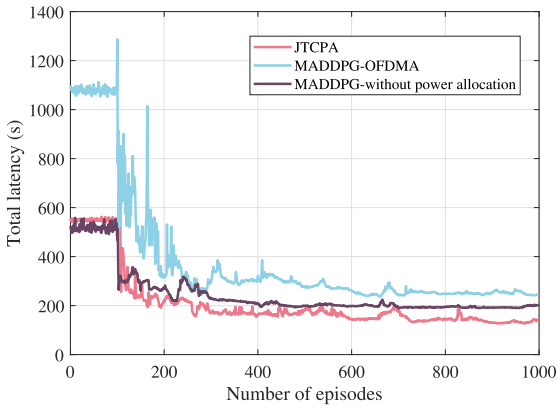


Fig. 5. Total transmission latency among schemes with MADDPG algorithm.

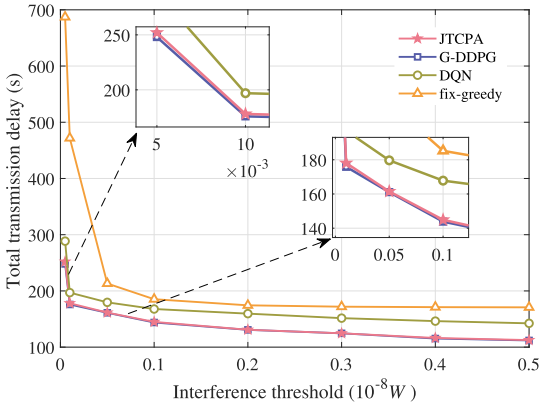


Fig. 6. Total transmission latency among algorithms vs $I_{\rm th}$.

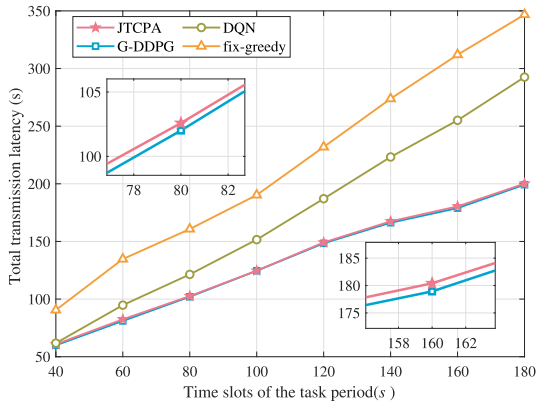


Fig. 7. Total transmission latency among algorithms vs K.

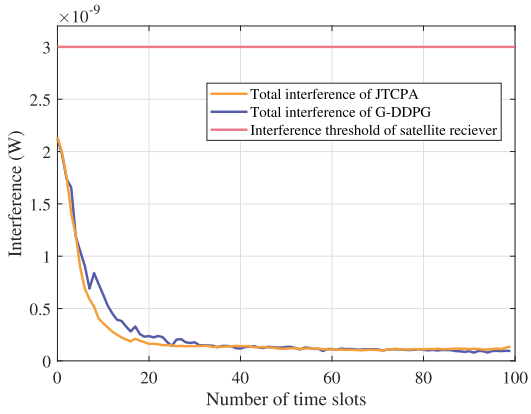
which leads to an increase in the complexity of the G-DDPG algorithm.

Fig. 5 presents the total latency comparison of different transmission schemes based on the MAPPPG algorithm. We compare the proposed JTCPA algorithm-based NOMA transmissiion scheme with the MAPPPG-based trajectory control NOMA transmission scheme (MADDPG-without power allocation scheme), and the orthogonal frequency division multiple access (OFDMA) transmission scheme with joint trajectory and power optimization (MADDPG-OFDMA scheme). It is shown that our proposed transmission scheme has the least transmission latency, which demonstrates the contributions of introducing NOMA and intelligent power allocation to reduce the transmission latency.

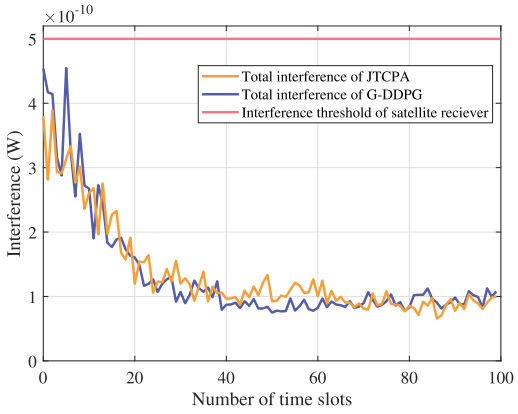
To investigate the performance of our JCARA algorithm, besides the G-DDPG algorithm, we also compare it with the DQN algorithm that uses global information and fix-greedy algorithm on the total transmission latency. Since the DQN algorithm can only deal with discrete actions, using the DQN algorithm to solve the proposed optimization problem requires discretizing the action space and selecting actions from the preprocessed discrete space. In the fix-greedy algorithm, the flight trajectories of the UAVs are fixed and the transmit power is used greedily under the limit of $I_{\rm th}$.

Fig. 6 and Fig. 7 illustrate the total transmission latency among different algorithms with NOMA. We observe that, in the case of ignoring the signaling exchange delay of the G-DDPG algorithm, the JTCPA and G-DDPG algorithms have comparable performance in the transmission latency. The performance of the DQN algorithm in the transmission latency is worse than that of the previous two algorithms, because the discrete action space causes the DQN algorithm to miss some better decisions. The fix-greedy algorithm has the largest transmission latency, because there is no intelligent optimization for it. Fig. 6 also shows that for the four given algorithms, the total latency reduces with the increasing $I_{\rm th}$ and then remain almost unchanged. This demonstrates that when $I_{\rm th}$ is up to a certain upper limit, the effects of varying $I_{\rm th}$ on the total latency can be ignored.

Fig. 8(a) and Fig. 8(b) are the total interference in each time slot produced by the UAVs to the satellite receiver during a task period under $I_{\rm th} = 3 \times 10^{-9} W$ and $I_{\rm th} = 0.5 \times 10^{-9} W$

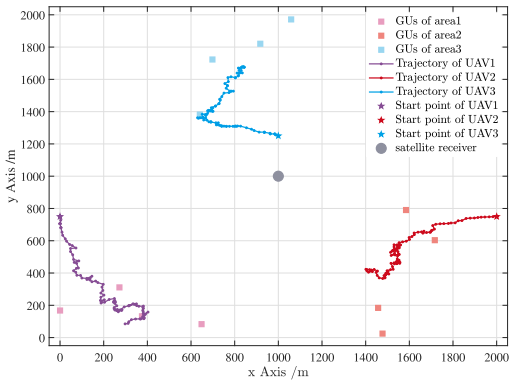


(a) Total interference at interference threshold of $I_{\text{th}} = 3 \times 10^{-9} W$

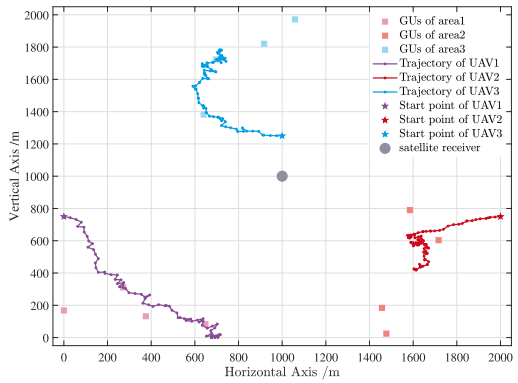


(b) Total interference at interference threshold of $I_{\text{th}} = 0.5 \times 10^{-9} W$

Fig. 8. Total interference at the satellite receiver.

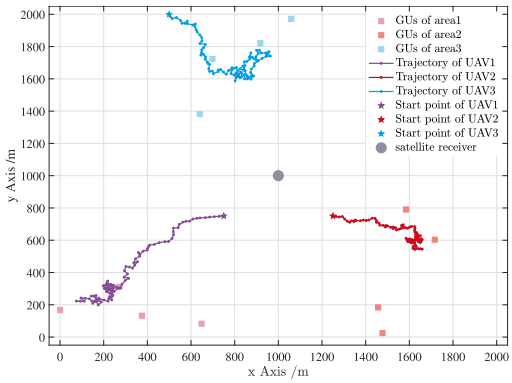


(a) Trajectory of JTCPA

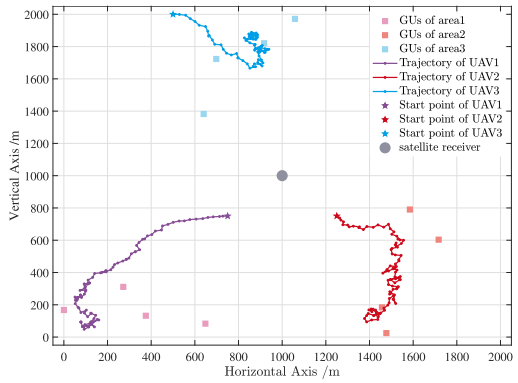


(b) Trajectory of G-DDPG

Fig. 9. Trajectory comparison 1.



(a) Trajectory of JTCPA



(b) Trajectory of G-DDPG

Fig. 10. Trajectory comparison 2.

respectively. The results in Fig. 8 illustrate that the total interference is always below the interference threshold which verifies that our designed reward function in eq.(22) for JTCPA algorithm is effective.

Fig. 9 and Fig. 10 are the trajectory comparisons of the JTCPA and G-DDPG algorithms from different start points. Fig. 9(a) and Fig. 10(a) are the trajectories of the JTCPA algorithm. Fig. 9(b) and Fig. 10(b) are the trajectories of G-DDPG algorithm. In Fig. 9, the start points for the three UAVs are $[0, 750, 100]$, $[2000, 750, 100]$ and $[1000, 1250, 100]$ respectively. And in Fig. 10, the start points are $[750, 750, 100]$, $[1250, 750, 100]$ and $[500, 2000, 100]$ accordingly. The position of the satellite receiver is fixed with coordinates of $[1000, 1000, 0]$. Fig. 9 and Fig. 10 show that the UAVs can control and optimize their flight trajectories to provide service for the GUs. And at different starting points, UAVs tend to

fly to the GUs they serve. We find that although the UAVs only make decisions based on their local observations in our proposed JTCPA algorithm, they can well control their flight. This demonstrates that our proposed algorithm can well dynamically program the trajectories with less information compared to the G-DDPG algorithm.

V. Conclusion

In this paper, we study a joint trajectory and power optimization problem to minimize the total transmission latency of the long-term task period for NOMA-aided cognitive satellite-UAV networks. The mobility of UAVs, the unpredictable transmission tasks and the coupling between optimization variables make the problem non-convex and complicated. To solve this optimization problem, we transform it into a POMDP-based MARL problem and propose a MADDPG-based JTCPA algorithm. In order to prevent the satellite communication from being affected, we introduce penalties to regulate the reward. The simulation results show that our proposed JTCPA algorithm can achieve comparable performance on the total transmission latency to the G-DDPG algorithm with lower complexity, more stable convergence and less information required for decision-making. Additionally, compared with other transmission algorithms, our proposed JTCPA algorithm-based NOMA transmission scheme has a significant improvement to reduce the transmission latency through the optimized trajectory control and transmit power allocation.

References

[1] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," *IEEE Netw.*, vol. 34, no. 3, pp. 134–142, May/Jun. 2020.

[2] A. Ikpehai et al., "Low-power wide area network technologies for Internet-of-Things: A comparative review," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2225–2240, Apr. 2019.

[3] M. de Sanctis, E. Cianca, G. Araniti, I. Bisio, and R. Prasad, "Satellite communications supporting internet of remote things," *IEEE Internet Things J.*, vol. 3, no. 1, pp. 113–123, Feb. 2016.

[4] L. Zhen et al., "Optimal preamble design in spatial group-based random access for satellite-M2M communications," *IEEE Wireless Commun. Lett.*, vol. 8, no. 3, pp. 953–956, Jun. 2019.

[5] Y. Zeng, R. Zhang, and T. J. Lim, "Wireless communications with unmanned aerial vehicles: Opportunities and challenges," *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 36–42, May 2016.

[6] Y. Zeng and R. Zhang, "Energy-efficient UAV communication with trajectory optimization," *IEEE Trans. Wireless Commun.*, vol. 16, no. 6, pp. 3747–3760, Jun. 2017.

[7] S. Zhang, Y. Zeng, and R. Zhang, "Cellular-enabled UAV communication: A connectivity-constrained trajectory optimization perspective," *IEEE Trans. Commun.*, vol. 67, no. 3, pp. 2580–2604, Mar. 2019.

[8] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Mobile unmanned aerial vehicles (UAVs) for energy-efficient Internet of Things communications," *IEEE Trans. Wireless Commun.*, vol. 16, no. 11, pp. 7574–7589, Nov. 2017.

[9] L. Wang, H. Yang, J. Long, K. Wu, and J. Chen, "Enabling ultra-dense UAV-aided network with overlapped spectrum sharing: Potential and approaches," *IEEE Netw.*, vol. 32, no. 5, pp. 85–91, Sep./Oct. 2018.

[10] A. Vanelli-Coralli et al., "Cognitive radio scenarios for satellite communications: The CoRaSat project," *Cooperative and Cognitive Satellite Systems*. Amsterdam, The Netherlands: Elsevier, 2015, pp. 303–336.

[11] P. Naseri, S. A. Matos, J. R. Costa, C. A. Fernandes, and N. J. G. Fonseca, "Dual-band dual-linear-to-circular polarization converter in transmission mode application to K/Ka-band satellite communications," *IEEE Trans. Antennas Propag.*, vol. 66, no. 12, pp. 7128–7137, Dec. 2018.

[12] Y. Liang, J. Tan, H. Jia, J. Zhang, and L. Zhao, "Realizing intelligent spectrum management for integrated satellite and terrestrial networks," *J. Commun. Inf. Netw.*, vol. 6, no. 1, pp. 32–43, 2021.

[13] Y. Ruan, Y. Li, C.-X. Wang, R. Zhang, and H. Zhang, "Energy efficient power allocation for delay constrained cognitive satellite terrestrial networks under interference constraints," *IEEE Trans. Wireless Commun.*, vol. 18, no. 10, pp. 4957–4969, Oct. 2019.

[14] Y. Ruan, Y. Li, C.-X. Wang, R. Zhang, and H. Zhang, "Power allocation in cognitive satellite-vehicular networks from energy-spectral efficiency tradeoff perspective," *IEEE Trans. Cognit. Commun. Netw.*, vol. 5, no. 2, pp. 318–329, Jun. 2019.

[15] J. Hu, G. Li, D. Bian, L. Gou, and C. Wang, "Optimal power control for cognitive LEO constellation with terrestrial networks," *IEEE Commun. Lett.*, vol. 24, no. 3, pp. 622–625, Mar. 2020.

[16] S. K. Nobar, M. H. Ahmed, Y. Morgan, and S. A. Mahmoud, "Resource allocation in cognitive radio-enabled UAV communication," *IEEE Trans. Cognit. Commun. Netw.*, vol. 8, no. 1, pp. 296–310, Mar. 2022.

[17] Y. Huang, W. Mei, J. Xu, L. Qiu, and R. Zhang, "Cognitive UAV communication via joint maneuver and power control," *IEEE Trans. Commun.*, vol. 67, no. 11, pp. 7872–7888, Nov. 2019.

[18] A. Bhowmick, S. Dhar Roy, and S. Kundu, "Throughput maximization of a UAV assisted CR network with NOMA-based communication and energy-harvesting," *IEEE Trans. Veh. Technol.*, vol. 71, no. 1, pp. 362–374, Jan. 2022.

[19] T. Q. Duong, L. D. Nguyen, H. D. Tuan, and L. Hanzo, "Learning-aided realtime performance optimisation of cognitive UAV-assisted disaster communication," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–6.

[20] M. Hua, Y. Wang, M. Lin, C. Li, Y. Huang, and L. Yang, "Joint CoMP transmission for UAV-aided cognitive satellite terrestrial networks," *IEEE Access*, vol. 7, pp. 14959–14968, 2019.

[21] Y. Ruan, Y. Li, R. Zhang, W. Cheng, and C. Liu, "Cooperative resource management for cognitive satellite-aerial-terrestrial integrated networks towards IoT," *IEEE Access*, vol. 8, pp. 35759–35769, 2020.

[22] F. Pervez, L. Zhao, and C. Yang, "Joint user association, power optimization and trajectory control in an integrated satellite-aerial-terrestrial network," *IEEE Trans. Wireless Commun.*, vol. 21, no. 5, pp. 3279–3290, May 2022.

[23] C. Liu, W. Feng, Y. Chen, C.-X. Wang, and N. Ge, "Cell-free satellite-UAV networks for 6G wide-area Internet of Things," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 4, pp. 1116–1131, Apr. 2021.

[24] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[25] L. Li, Q. Cheng, K. Xue, C. Yang, and Z. Han, "Downlink transmit power control in ultra-dense UAV network based on mean field game and deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 15594–15605, Dec. 2020.

[26] F. Wu, H. Zhang, J. Wu, Z. Han, H. V. Poor, and L. Song, "UAV-to-device underlay communications: Age of information minimization by multi-agent deep reinforcement learning," *IEEE Trans. Commun.*, vol. 69, no. 7, pp. 4461–4475, Jul. 2021.

[27] R. Zhong, X. Liu, Y. Liu, and Y. Chen, "Multi-agent reinforcement learning in NOMA-aided UAV networks for cellular offloading," *IEEE Trans. Wireless Commun.*, vol. 21, no. 3, pp. 1498–1512, Mar. 2022, doi: 10.1109/TWC.2021.3104633.

[28] Z. Qin, Z. Liu, G. Han, C. Lin, L. Guo, and L. Xie, "Distributed UAV-BSs trajectory optimization for user-level fair communication service with multi-agent deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 70, no. 12, pp. 12290–12301, Dec. 2021.

[29] Y. Zhang, Z. Mou, F. Gao, J. Jiang, R. Ding, and Z. Han, "UAV-enabled secure communications by multi-agent deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 11599–11611, Oct. 2020.

[30] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal LAP altitude for maximum coverage," *IEEE Wireless Commun. Lett.*, vol. 3, no. 6, pp. 569–572, Dec. 2014.

[31] M. S. Ali, H. Tabassum, and E. Hossain, "Dynamic user clustering and power allocation for uplink and downlink non-orthogonal multiple access (NOMA) systems," *IEEE Access*, vol. 4, pp. 6325–6343, 2016.

[32] X. Zhou, Q. Wu, S. Yan, F. Shu, and J. Li, "UAV-enabled secure communications: Joint trajectory and transmit power optimization," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 4069–4073, Apr. 2019.

[33] Q. Wu, Y. Zeng, and R. Zhang, "Joint trajectory and communication design for multi-UAV enabled wireless networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 2109–2121, Mar. 2018.

[34] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.

[35] V. Mnih et al., "Playing atari with deep reinforcement learning," in *Proc. NIPS Deep Learn. Workshop*, Dec. 2013, pp. 1–9.
[36] T. P. Lillicrap et al., "Continuous control with deep reinforcement learning," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, May 2015, pp. 1–14.
[37] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Proc. NIPS*, 2017, pp. 6382–6393.
[38] H. Peng and X. Shen, "Multi-agent reinforcement learning based resource management in MEC- and UAV-assisted vehicular networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 131–141, Jan. 2021.
[39] P. Wawrzynski and A. K. Tanwani, "Autonomous reinforcement learning with experience replay," *Neural Netw.*, vol. 41, no. 5, pp. 156–167, 2013.

Shaoai Guo (Student Member, IEEE) received the bachelor's degree from the College of Electronic Science and Engineering, Jilin University, Changchun, China, in 2013, and the master's degree in circuits and systems from the University of Chinese Academy of Sciences, Beijing, China, in 2016. She is currently pursuing the Ph.D. degree with the College of Communication Engineering, Jilin University. Her research interests include cognitive radio, UAV communications and artificial intelligence.

Xiaohui Zhao received the Ph.D. degree in applied mathematics and control theory from the Université de Technologie de Compiègne, Compiègne, France, in 1993. He did his post-doctoral research with the Institute of Automatic Control, Southeast University, Nanjing, China, from 1994 to 1996, and he was a Senior Visiting Scholar for half a year with the Laboratoire d'Informatique, Université de Pierre et Marie Curie, Paris, France, in 2006. He is currently a Professor of communication engineering with Jilin University, Changchun, China. His research interests include wireless communication, cognitive radio, and adaptive signal processing.