IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Multi-agent Deep Reinforcement Learning for Trajectory Design and Power Allocation in Multi-UAV Networks

NAN ZHAO(Member, IEEE), ZEHUA LIU, YIQIANG CHENG
Hubei Collaborative Innovation Center for High-efficiency Utilization of Solar Energy, Hubei University of Technology, Wuhan 430068, China
Corresponding author: Nan Zhao (e-mail: nzhao@mail.hbut.edu.cn)

ABSTRACT Unmanned aerial vehicle (UAV) is regarded as an effective technology in future wireless networks. However, due to the non-convexity feature of joint trajectory design and power allocation (JTDPA) issue, it is challenging to attain the optimal joint policy in multi-UAV networks. In this paper, a multi-agent deep reinforcement learning-based approach is presented to achieve the maximum long-term network utility while satisfying the user equipments' quality of service requirements. Moreover, considering that the utility of each UAV is determined based on the network environment and other UAVs' actions, the JTDPA problem is modeled as a stochastic game. Due to the high computational complexity caused by the continuous action space and large state space, a multi-agent deep deterministic policy gradient method is proposed to obtain the optimal policy for the JTDPA issue. Numerical results indicate that our method can obtain the higher network utility and system capacity than other optimization methods in multi-UAV networks with lower computational complexity.

INDEX TERMS UAV networks, trajectory design, power allocation, multi-agent deep reinforcement learning.

I. INTRODUCTION

RECENTLY, unmanned aerial vehicles (UAVs) have been regarded as an important technology in the future wireless networks [1]. Since the UAVs can be deployed and configured flexibly, it can be utilized as relays between ground user equipments (UEs) for cooperative communication. Furthermore, considering that UAVs can smartly alter their spots to offer on-demand wireless services for ground UEs, UAVs can be used as aerial base stations (ABSs) for wireless communication [2]. Thus, multi-UAV networks have been applied to varied applications, such as remote sensing, traffic monitoring, public safety, and military [3], [4].

In multi-UAV networks, many technical design problems should be considered, including trajectory design, resource allocation as well as interference management. Through appropriately designing the trajectories of UAVs, UAVs can provide UEs communication services, which may ease co-channel interference and increase system capacity. Furthermore, the transmission powers of UAVs should also be taken into account to meet the trade-off between spectrum efficiency and interference management. Thus, the problem of trajectory design, power allocation, and interference management should be studied jointly in multi-UAV networks.

The problem of joint trajectory design and power allocation (JTDPA) has drawn much attention, which has been investigated in [5]–[7]. However, due to the non-convex feature of the JTDPA issue, it may be challenging to obtain a global optimal solution. Several methods try to solve this issue, i.e., the alternating optimization approach [9], Lagrange dual method [10], and iterative algorithm [11], [12]. Nearly accurate information is always needed to deal with the JTDPA issue. However, it is challenging to attain the optimal policy without complete knowledge of the network environment. Thus, in this work, we propose a reinforcement learning (RL) method to tackle the JTDPA optimization problem in the multi-UAV networks.

RL approach [13] has been widely adopted in the artificial intelligence and wireless communication fields [14]. The authors in [15] utilized an RL method to investigate the resource management scheme in the Internet of Vehicles communication networks. In [16], the RL approach was proposed to obtain the joint power control and channel allocation strategy in dense wireless local area networks. Moreover, by combining the deep neural networks with RL, deep reinforce-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3012756, IEEE Access

ment learning (DRL) [17] method has been recently attracted increasing interests in wireless communication domains. The authors in [18] proposed a DRL-based relay selection method for cooperative communication in wireless sensor networks. In [19], a DRL-based method was studied to solve the joint mode selection and resource management issue in fog radio access networks. Chen et al. proposed a DRL scheme to solve resource allocation problem in the collaborative mobile edge computing network [20]. A DRL method was investigated in [21] to obtain the resource allocation policy for smart cities. Our previous work proposed a DRL approach for trajectory design and power allocation in UAV networks [22]. However, most of these centralized methods may achieve an expensive computational complexity. Thus, multi-agent DRL (MADRL) may be a possible way to obtain the policy with a low computational complexity. The authors in [23] proposed an MADRL approach to deal with the large-scale crowd path planning issue. In our previous work [24], an multi-agent dueling-double deep Q-network method was investigated to tackle the joint user association and resource allocation problem. In [25], an MADRL strategy was studied for the large-scale traffic signal control problem. However, to our best knowledge, little works have been done to solve the MADRL method for the JTDPA optimization problem.

In this paper, an MADRL method is introduced to tackle the JTDPA optimization problem in multi-UAV networks. The main contributions are presented as follows. Considering the demand of UEs' quality of service (QoS), the JTDPA joint optimization issue is formulated to obtain the maximum cumulative discounted reward. Then, due to the non-convex and combinatorial nature of the JTDPA optimization issue, such problem is modeled as a stochastic game, which is solved by the proposed MADRL approach. Specifically, the state, action and reward function are defined for all UAVs. Then, the optimal strategy is achieved by jointly designing the UAVs' trajectory and allocating UAVs' transmission power. Moreover, considering the continuous action space and large state space of the stochastic game, multi-agent deep deterministic policy gradient (MADDPG) approach is proposed to learn the optimal policy. A DDPG algorithm is designed for each UAV to solve the joint optimization issue. Target network and experience replay strategies are leveraged to improve the learning stability. Numerical simulations with different parameters are presented to show the effectiveness of our proposed method. Simulation results indicate that the MADDPG scheme can improve the system capacity and network utility by over 15% with lower computational cost in multi-UAV networks, compared with the other learning optimization approaches.

The rest of this paper is organized as follows. System model and problem formulation are given in Section II. Section III presents an MADRL method to solve the JTDPA problem. Simulation results are provided in Section IV. Section V gives the conclusion of this paper.

II. SYSTEM MODEL AND PROBLEM FORMULATION

A. SYSTEM MODEL

In the typical multi-UAV networks, K UAVs are used as ABSs to offer communication service to M UEs in K non-overlapping hotspots. The UEs' set and UAVs' set are represented as $\mathcal{M}$ and $\mathcal{K}$, respectively. Assume that the number of UEs in hotspot i is $M(i)$. For the simplicity of discussion, we assume that each UAV can only assist to no more than one hotspot. Furthermore, since each UE only belongs to one hotspot, we have $\sum_{i=1}^{K} M(i) = M$. The UEs in the same hotspot can be served by the same UAV through using FDMA [26].

Assume that $v_m = [x_m, y_m]^{\mathrm{T}}, m \in \mathcal{M}$ are the 2D coordinates of UE m, where x_m and y_m are the coordinates of UE m, respectively. Then, the horizontal coordinate of UAV i is represented as $v_i(t) = [x_i(t), y_i(t)]^{\mathrm{T}}, i \in \mathcal{K}$, where $x_i(t)$ and $y_i(t)$ are the X and Y coordinates of UAV i at time t, respectively. The horizontal distance between UE m and UAV i at time t can be defined as

$$l_{i,m}(t) = \sqrt{[x_i(t) - x_m]^2 + [y_i(t) - y_m]^2}. \tag{1}$$

Next, the vertical flight position of UAV i is denoted by $z_i(t) \in [Z_{min}, Z_{max}]$, where Z_{min} and Z_{max} are the minimum height and maximum height of UAVs, respectively.

Then, the distance between UAV i and UE m at time t is obtained as

$$d_{i,m}(t) = \sqrt{{z_i}^2(t) + l_{i,m}(t)^2}. \tag{2}$$

Due to the limited flying speed of UAVs, each UAV may have a maximum flight distance, which is defined as

$$\|v_i(t+1) - v_i(t)\| \le V_H T, \tag{3}$$

$$\|z_i(t+1) - z_i(t)\| \le V_A T, \tag{4}$$

where V_H and V_A are the horizontal-flight and vertical-flight speeds of UAVs in each time slot T, respectively.

Furthermore, to avoid collision between UAVs, collision avoiding constraints of UAVs should be taken into account, which is given by

$$\|v_i(t) - v_j(t)\|^2 + \|z_i(t) - z_j(t)\|^2 \ge D_{min}^2, \forall i, j \in \mathcal{K}, i \ne j, \tag{5}$$

where D_{min} is the minimum distance between arbitrary two UAVs.

Note that the time slot T should be small enough so as to treat the channel as approximate constant. Then, in order to avoid collision between arbitrary two UAVs, the time slot T should satisfy the following constraint, that is,

$$T \le T_{max} = \frac{D_{min}}{2\sqrt{V_L^2 + V_A^2}}, \tag{6}$$

where T_{max} is the maximum value of a time slot.

Then, the maximum horizontal distance L_{max}^h and the maximum vertical distance L_{max}^v can be expressed as,

$$L_{max}^h = V_H T_{max}, \tag{7}$$

$$L^{v}_{max} = V_A T_{max}. \tag{8}$$

Next, considering that the radio signals radiated from the UAVs are comprised of Line-of-Sight (LoS) or non Line-of-Sight (NLoS). The probability of the LoS connection between UE m and UAV i at time t can be defined as [27]

$$P^{LoS}_{i,m}(t) = \frac{1}{1 + a\exp(-b(\frac{180}{\pi}\tan^{-1}(\alpha_{i,m}(t)) - a))}, \tag{9}$$

where a and b are parameters related with the environment, $\alpha_{i,m}(t)$ is the angle of UAV i. Then, the probability of the NLoS can be derived as

$$P^{NLoS}_{i,m}(t) = 1 - P^{LoS}_{i,m}(t). \tag{10}$$

Correspondingly, at time t, the path loss models of the LoS and the NLoS in dB can be represented as [27],

$$L^{LoS}_{i,m}(t) = 20\log(\frac{4\pi f_c d_{i,m}(t)}{c}) + \eta_{LoS}, \tag{11}$$

$$L^{NLoS}_{i,m}(t) = 20\log(\frac{4\pi f_c d_{i,m}(t)}{c}) + \eta_{NLoS}, \tag{12}$$

where f_c represents the carrier frequency, η_{LoS} and η_{NLoS} are the mean extra losses for the LoS and NLoS, respevtively.

Next, the expected mean path loss[1] can be obtained as

$$L_{i,m}(t) = L^{LoS}_{i,m}(t) \times P^{LoS}_{i,m}(t) + L^{NLoS}_{i,m}(t) \times P^{NLoS}_{i,m}(t). \tag{13}$$

Assume that the bandwidth B is allocated to each UE equally. Then, we can derive the bandwidth of UE m in hotspots i, which is given by

$$B_{i,m} = B/M(i). \tag{14}$$

Furthermore, each UAV's transmission power is allocated equally to all UEs in hotspot i, which can be represented as

$$p_{i,m}(t) = p_i(t)/M(i), \tag{15}$$

where $0 \leq p_i(t) \leq P_{max}$ is the transmission power of UAV i, and P_{max} is the maximum transmission power.

Next, based on the transmission power of UAV $p_i(t)$, the received SINR of UE m from UAV i can be given by

$$\varphi_{i,m}(t) = \frac{p_{i,m}(t)g_{i,m}(t)}{B_{i,m}N_0 + \sum\limits_{j\neq i} p_{j,m}(t)g_{j,m}(t)}, \forall i, j \in \mathcal{K}, \tag{16}$$

where $g_{i,m}(t)$ represents the channel gain between UAV i and UE m, N_0 is the noise power spectral density.

Then, the rate of UE m served by UAV i can be obtained as

$$\phi_{i,m}(t) = B_{i,m}\log_2(1 + \varphi_{i,m}(t)). \tag{17}$$

The total rate of UAV i can be derived as

$$\phi_i(t) = \sum_{m=1}^{M(i)} \phi_{i,m}(t) = \sum_{m=1}^{M(i)} B_{i,m}\log_2(1 + \varphi_{i,m}(t)). \tag{18}$$

[1]Other models of UAV communication [28] can also be applied in this paper. Such path loss model will achieve similar performance by using our proposed method.

Then, we define the utility $w_i(t)$ of UAV i as the difference between the profit and the transmission cost, that is,

$$w_i(t) = \rho_i\phi_i(t) - \lambda_p p_i(t) = \sum_{m=1}^{M(i)} [\rho_i\phi_{i,m}(t) - \lambda_p p_{i,m}(t)], \tag{19}$$

where ρ_i represents the profit per rate, λ_p is the cost of UAV's transmit power.

B. PROBLEM FORMULATION

In multi-UAV networks, to ensure that all UEs achieve the QoS requirements from the connected UAVs, the SINR $\varphi_{i,m}(t)$ of UE m should be not less than the minimum QoS requirement Ω_m, which can be defined as

$$\varphi_{i,m}(t) \geq \Omega_m. \tag{20}$$

Therefore, the JTDPA optimization issue is to maximize the overall network utility via the optimization of each UAV's trajectory ($v_i(t)$ and $z_i(t)$) and transmission power ($p_i(t)$), which can be formulated as

$$\begin{aligned} \max_{p_i(t), v_i(t), z_i(t)} \quad & \sum_{i=1}^{K} w_i(t) = \sum_{i=1}^{K}\sum_{m=1}^{M(i)} [\rho_i\phi_{i,m}(t) - \lambda_p p_{i,m}(t)], \\ s.t. \quad & (3), (4), (5), (20), \\ & Z_{min} \leq z_i(t) \leq Z_{max}, \\ & 0 \leq p_i(t) \leq P_{max}. \end{aligned} \tag{21}$$

Considering that the JTDPA problem has the non-convex and combinatorial characteristics, it will be intractable to deal with the optimization issue. Exhaustive search algorithm may find the optimal policy with the high computational complexity. Moreover, since the network information (i.e., UEs' information and channel condition) is hardly to obtain, which makes it challenging to obtain the optimal policy with traditional optimization methods. In the next section, a reinforcement learning method will be proposed to find the optimal JTDPA strategy.

III. MULTI-AGENT DRL FOR JTDPA OPTIMIZATION ISSUE

In order to obtain the maximum network utility, the trajectory and transmission power of UAVs should be determined according to the network environment. In this section, the above issue is modeled as a stochastic game, which is then tackled with an MADRL approach.

A. GAME FORMULATION

In multi-UAV networks, assume that each UAV decides its own trajectory and transmission power to acquire its maximum utility $w_i(t)$. The utility of each UAV can be determined based on the current state of the network environment and other UAVs' actions. Then, the network environment turns into a new stochastic state [29], which depends on the former state and actions taken previously. The JTDPA problem (21) is then modeled as a stochastic game $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R} \rangle$ [30],

- $\mathcal{S}$ represents the state space;

- $\mathcal{A}_i$ is the action space of UAV i;
- $\mathcal{P}$ represents the state transition probability. $\mathcal{P}_{ss'}(\times_i \mathcal{A}_i)$ describes the state transition probability from state s to state s' by jointly taking action $\times_i \mathcal{A}_i$;
- $\mathcal{R}_i$ denotes the reward function of UAV i.

In the stochastic game, the state $\mathcal{S}(t)$ is defined to reflect whether the minimum QoS requirement of each UE is satisfied or not, that is,

$$\mathcal{S}(t) = \{s_1(t), s_2(t), \ldots, s_M(t)\}, \tag{22}$$

where $s_m(t) \in \{0, 1\}$. If the UE m achieves the minimum QoS requirement $\varphi_{i,m}(t) \geq \Omega_m$, $s_m(t) = 1$, else $s_m(t) = 0$. Note that the state space [] is 2^M, which can be very huge with the large M.

Then, considering that each UAV needs to decide its own trajectory and transmission power at time t, we define the action space $\mathcal{A}_i(t)$ of UAV i as

$$\mathcal{A}_i(t) = \{p_i(t), l_i(t), \vartheta_i(t), \Delta h_i(t)\}, \tag{23}$$

where $p_i(t) \in \{0, P_{max}\}$, $l_i(t)$, $\vartheta_i(t) \in \{0, 2\pi\}$, and $\Delta h_i(t)$ are the transmission power, the horizontal distance, the direction angle, and the vertical travel distance of UAV i, respectively. From the horizontal trajectory constraint (3), we have $l_i(t) \in \{0, L^h_{max}\}$. Considering the vertical trajectory constraint (4), $\Delta h_i(t) = [h_i(t) - h_i(t-1)] \in \{-L^v_{max}, L^v_{max}\}$.

Moreover, as for the reward function, in order to ensure that all UEs are served by UAVs, the coverage of UAVs should be taken into account. If a UE is not in the coverage of any UAV, a punishment will be imposed on the reward function. In addition, to ensure that all UEs' minimum QoS requirements are satisfied, the state $s_m(t)$ of each UE should be considered in the reward function. Then, based on (5) and (20), the reward function of UAV i can be defined as

$$\begin{aligned}\mathcal{R}_i(t) = &\sum_{m=1}^{M'(i)} s_m(t)\left[\rho_i \phi_{i,m}(t) - \lambda_p p_{i,m}(t)\right] \\ &-\eta_1 \left[M - \sum_{i=1}^{K} M'(i)\right] - \eta_2^i,\end{aligned} \tag{24}$$

where $M'(i)$ is the number of UEs covering by UAV i, η_1 represents the punishment factor relating to UAVs' coverage, η_2^i represents the punishment of UAVs' collision. The first part of (24) is the overall network utility. If UE m achieves the minimum QoS demand, $s_m(t) = 1$, else $s_m(t) = 0$. The second part of (24) is the punishment of UAVs' coverage. If all UEs covered by all UAVs, this section is equal to zero. The final part of (24) represents the punishment of UAVs' overlapping. When the horizontal distance between arbitrary two UAVs is less than the sum of their coverage radius, each UAV would be obtained a punishment η_2^i. Otherwise, the final part of (24) is equal to zero.

Note that, when UAV i takes an action $\mathcal{A}_i(t)$ and other UAVs take actions $\mathcal{A}_{-i}(t)$, UAV i may obtain the reward $\mathcal{R}_i(t) = \mathcal{R}_i(t, \mathcal{S}(t), \mathcal{A}_i^*(t), \boldsymbol{\mathcal{A}}_{-i}^*(t))$. Here, the action vector $(\mathcal{A}_i(t), \mathcal{A}_{-i}(t))$ is defined as the feasible solution to our game. When the following inequality is satisfied for each UAV in any $\mathcal{S}(t)$, the Nash equilibrium (NE) state can be achieved in this game [31]:

$$\mathcal{R}_i(t, \mathcal{S}(t), \mathcal{A}_i^*(t), \boldsymbol{\mathcal{A}}_{-i}^*(t)) \geq \mathcal{R}_i(t, \mathcal{S}(t), \mathcal{A}_i(t), \boldsymbol{\mathcal{A}}_{-i}^*(t)). \tag{25}$$

In the NE state, the action of each UAV can be regarded as the optimal reaction to the actions of other UAVs. All UAVs achieve no benefit from unilateral deviation [31]. Moreover, considering that this stochastic game is periodic, the state of the network environment will be reset after each episode ends. In each episode, the policies of all UAVs are carried out to obtain the accumulative rewards from the environment. If all UAVs can obtain information about the reward function and the state transition, the NE strategy can be found with integer programming methods. However, in this stochastic game, such information is not available for UAVs. Therefore, in order to deal with this issue, the MADRL approach is proposed to achieve an NE policy at any state through interacting with the network environment.

B. MULTI-AGENT DRL METHOD

Considering the continuous action space of the JTDPA issue in multi-UAV networks, a MADDPG approach is proposed to obtain the optimal joint trajectory design and power allocation policy. The framework of the MADDPG approach for the JTDPA issue is shown in Figure 1. In our stochastic game, each UAV is modeled as an DDPG agent, which consists of actor and critic [32]. The MADDPG approach is utilized to learn the optimal policy for each UAV to obtain the maximum expected discounted reward, which is defined as

$$\Phi(t) = \sum_{t'=t}^{t+T_p-1} \gamma^{t'-t} \sum_{i=1}^{K} \mathcal{R}_i(t'), \tag{26}$$

where γ is the discount factor and $0 \leq \gamma < 1$, T_p is the total number of epochs.

Moreover, in order to increase the learning stability, both actor and critic consist of *online network* and *target network*. Specially, the *online critic network* of each UAV evaluates the performance of the actor $\mathcal{A}_i(t)$ with the state-action value function $Q(\mathcal{S}(t), \mathcal{A}_i(t)|\theta_Q^i)$, which is defined as

$$Q(\mathcal{S}(t), \mathcal{A}_i(t)|\theta_Q^i) = E\left[\Phi(t)|\mathcal{S}(t), \mathcal{A}_i(t)\right], \tag{27}$$

where $E[\cdot]$ represents the expectation operator, θ_Q^i is the weight of the *online critic network*.

In each UAV, the *target networks* of actor and critic are the replica of the corresponding *online networks*. With the weights of the most recent corresponding *online networks*, the weights of *target actor network* and *target critic network* can be updated through

$$\begin{aligned}\theta_{\mu'}^i &= \tau\theta_\mu^i + (1-\tau)\theta_{\mu'}^i, \\ \theta_{Q'}^i &= \tau\theta_Q^i + (1-\tau)\theta_{Q'}^i,\end{aligned} \tag{28}$$

where τ is the soft updating rate of *target networks*, θ_μ^i and θ_Q^i denote the weights of *online actor network* and *online critic network*, respectively. $\theta_{\mu'}^i$ and $\theta_{Q'}^i$ are the weights of *target actor network* and *target critic network*, respectively.

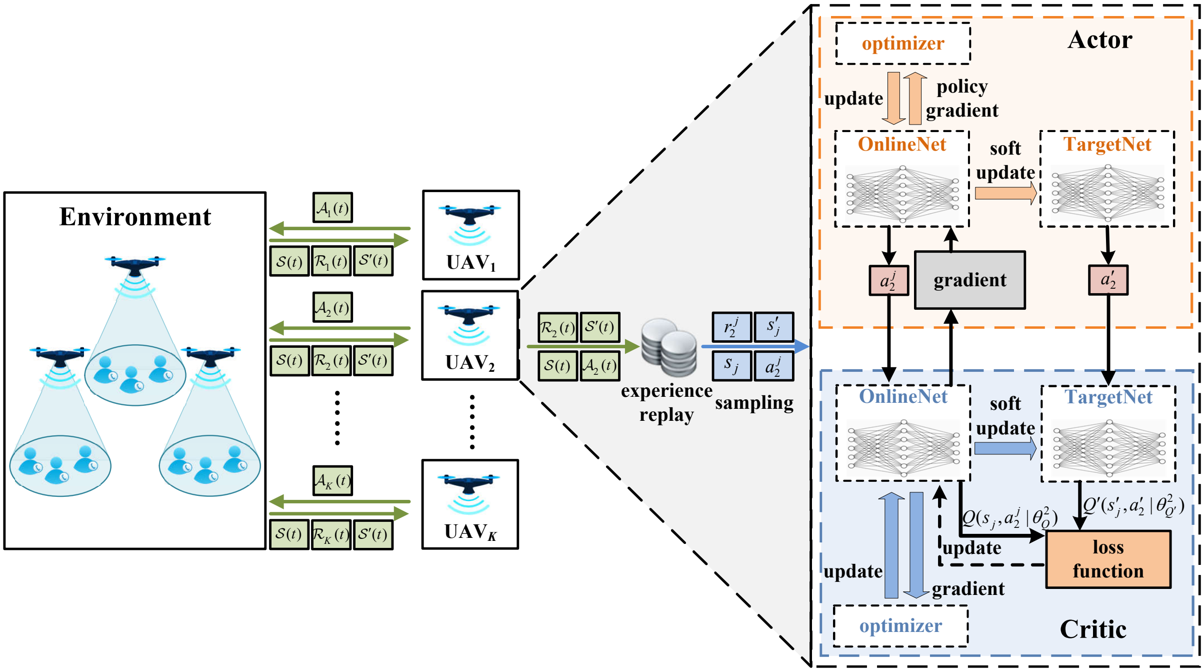


FIGURE 1: Multi-agent DDPG approach for JTDPA issue.

Furthermore, in order to guarantee the non-correlation in the training data, a *experience replay* strategy is applied to store the transition samples (state $\mathcal{S}(t)$, next state $\mathcal{S}'(t)$, action $\mathcal{A}_i(t)$, and reward $\mathcal{R}_i(t)$) in the *experience replay buffer* $\mathcal{B}$. By randomly sampling mini-batches (state s_j, next state s'_j, action a_i^j, and reward r_i^j) from the *experience replay buffer* $\mathcal{B}$, the *online actor network* can be updated with the policy gradient scheme [33], which is given by

$$\nabla_{\theta_\mu^i} J(\theta_\mu^i) = \frac{1}{M_b}\sum_{j=1}^{M_b} \nabla_{\theta_\mu^i}\mu(s_j|\theta_\mu^i)\nabla_{a_i^j}Q(s_j,a_i^j|\theta_Q^i), \quad (29)$$

where j is the index of the mini-batches, M_b is the size of mini-batches, $\mu(s_j|\theta_\mu^i)$ is the policy of *online actor network* θ_μ^i to map the state s_j to action a_i^j.

Moreover, the *online critic network* of each UAV is updated through minimizing the loss function $L(\theta_Q^i)$, which is defined as

$$L(\theta_Q^i) = \frac{1}{M_b}\sum_{j=1}^{M_b}\left[y_j - Q(s_j,a_i^j|\theta_Q^i)\right]^2, \quad (30)$$

where $y_j = r_i^j + \gamma Q'(s'_j, a'_i|\theta_{Q'}^i)|_{a'_i=\mu'(s'_j|\theta_{\mu'}^i)}$ is the target value generated by *target critic network* with weight $\theta_{Q'}^i$.

Then, based on (29) and (30), the weights of *online actor network* and *online critic network* can be updated by

$$\begin{aligned}\theta_\mu^i &\leftarrow \theta_\mu^i - \delta\nabla_{\theta_\mu^i}J(\theta_\mu^i),\\ \theta_Q^i &\leftarrow \theta_Q^i - \delta\nabla_{\theta_Q^i}L(\theta_Q^i),\end{aligned} \quad (31)$$

where δ is the learning rate of the two online networks.

The MADDPG approach for the JTDPA issue is summarized in Algorithm 1. At the beginning of the MADDPG algorithm, the replay buffer $\mathcal{B}$, the weights of actor and critic in each UAV are initialized. Notice that the training procedure comprises of D episodes, each of which consists of T_p epochs. Generally, at the beginning of each episode, we first initialize the state $\mathcal{S}(t)$. Then, in each epoch t, the action of each UAV at state $\mathcal{S}(t)$ is generated by its *online actor network* $\mu(\mathcal{S}(t)|\theta_\mu^i)$ with a random noise $\varepsilon\varsigma$, where $\varsigma \sim \mathcal{N}(0,1)$ is a random noise and ε is a decay factor decreasing over time. Based on the action taken above, each UAV set its three-dimensional trajectory and transmission power. If certain UAV flies beyond the network area, the UAV will choose a random direction angle $\phi_i(t)$. Furthermore, once the height of a UAV $z_i(t)$ is lower than Z_{min} or higher than Z_{max}, it will keep the height at Z_{min} or Z_{max}. After certain UAV covers a hotspot, it will stay without making movement and just adjust the transmission power.

Then, considering the minimum QoS requirement, each UE reports its state to its associated UAV. Through message passing, each UAV can obtain the global next state $\mathcal{S}'(t)$ and reward $\mathcal{R}_i(t)$. Then, the tuple $(\mathcal{S}(t), \mathcal{A}_i(t), \mathcal{R}_i(t), \mathcal{S}'(t))$ is stored in the *replay buffer* $\mathcal{B}$. After randomly sampling from the replay buffer $\mathcal{B}$, the *online networks* of actor and critic can be updated. The *target networks* of actor and critic are updated in (28). When the total number of UEs covering by all UAV is equal to M, all UAVs cover all UEs. Then, if the horizontal distance between arbitrary two UAVs is not less than the sum of their coverage radius, all UAVs cover all hotspots without overlapping. In this case, if the distance between any two UAVs is not less than D_{min}, the algorithm will go to the next episode until $episode > D$.

Note that, according to the theorem of Selten, a subgame

Algorithm 1 MADDPG Approach for JTDPA Issue

- Initialize the replay buffer $\mathcal{B}$.
- Initialize *online critic network* and *online actor network* with weights θ_Q^i and θ_μ^i, respectively.
- Initialize *target critic network* and *target actor network* with weights $\theta_{Q'}^i$ and $\theta_{\mu'}^i$, respectively.
- $episode = 1$.
- **while** $episode \le D$ **do**
- Initialize the environment state $\mathcal{S}(t) = \{0, \ldots, 0\}$.
- **for** $epoch$ $t = 1, \ldots, T_p$
- At the state $\mathcal{S}(t)$, each UAV selects the action $\mathcal{A}_i(t) = \mu(\mathcal{S}(t)|\theta_\mu^i) + \varepsilon\varsigma$.
- Each UAV sets their own trajectories and transmission power based on the given action $\mathcal{A}_i(t)$.
- Each UAV achieves the immediate reward $\mathcal{R}_i(t)$ and obtains the global next state $\mathcal{S}'(t)$ through message passing.
- The transition $(\mathcal{S}(t), \mathcal{A}_i(t), \mathcal{R}_i(t), \mathcal{S}'(t))$ is stored in $\mathcal{B}$.
- Let $\mathcal{S}(t) \leftarrow \mathcal{S}'(t)$.
- **for** UAV $i = 1, \ldots, K$
- Mini-batch of transitions $(s_j, a_i^j, r_i^j, s_j')$ is sampled stochastically from $\mathcal{B}$.
- Update the weight θ_μ^i of *online actor network* with (29).
- Update the weight θ_Q^i of *online critic network* by minimizing loss function $L(\theta_Q^i)$ in (30).
- **end for**
- Update the weights of the *target critic network* and *target actor network* in (28).
- **If** all hotspots covered by all UAVs without overlapping and the state $\mathcal{S}(t) = \{1, \ldots, 1\}$, **then**
- **If** the distance between any two UAVs is greater than D_{min}, **then**
- $episode \leftarrow episode + 1$.
- break.
- **end If**
- **end If**
- **end for**
- **end while**

perfect NE can exist in all the limited game with perfect memory [31]. In this stochastic game, the reward of each UAV is finite. The number of UAVs and the state-action space are also limited. Thus, this game is a finite game. Furthermore, due to the *experience replay* strategy adopted in the MADDPG method, essential historical information can be stored. Thus, in order to obtain the essential historical information, each UAV needs to communicate with UEs to acquire the global state by message passing. Since the state $s_m(t)$ is the only information passing between each UAV and each UE, the communication overhead is only one bit (0 or 1), which is relatively low and acceptable. Then, our proposed MADDPG approach can guarantee to converge to the subgame perfect NE in this stochastic game.

Considering that the hyperparameter plays a significant role in deep learning approaches, it is difficult to achieve the convergence of our MADDPG algorithm with analytical schemes. Furthermore, since it may be intractable to design the optimal hyperparameters of our MADDPG algorithm in advance, a trial-and-error strategy can be adopted. Thus, this issue is commonly in the literature to prove the optimality and convergence qualitatively. Here, this paper limits the convergence analysis with quantitative experiment results in Section IV-A, which is also adopted in the similar literatures [34], [35]. The performances with various learning rates and mini-batch sizes are given to ensure the convergence of our method. With the hyperparameters chosen properly, the convergence of our MADDPG method can be guaranteed.

IV. PERFORMANCE EVALUATION

In this section, the performance of the presented MADRL approach is numerically evaluated. In a $500m \times 500m$ network environment, the UEs and the UAVs are distributed arbitrarily. The main simulation parameters are shown in Table 1. Moreover, In the MADDPG method, both the actor and critic networks are designed with the two hidden layer (64 and 32 neurons). ε is set to decay from 2 with a decay rate of 0.9995. More detailed parameters of the MADDPG approach are presented in Table 2. This simulation is executed on a server with Intel Core i7 CPU and Tesla P100 GPU. The memory size is 128GB. The software platform of the server is Ubuntu 16.04 with Tensorflow 0.12.1.

TABLE 1: Network Environment Parameters

Parameters	Value
Channel bandwidth B	1 MHz
Downlink carrier frequency f_c	1950 MHz
Maximum transmit power of UAVs P_{max}	30 dBm
Maximum height of UAVs H_{max}	300 m
Minimum height of UAVs H_{min}	100 m
Noise power density N_0	-174 dBm/Hz
Minimum QoS requirement Ω_m	2 dB
Unit price per transmit power λ_p	2
Punishment cofficient of UEs' coverage η_1	120
Punishment of UAVs' collision η_2^i	20
Mean excessive pathloss for LoS η_{Los}	1 dB
Mean excessive pathloss for NLoS η_{NLos}	20 dB
Elevation angle $\alpha_{i,m}$	42.44°
Level-flight speed V_L	20 m/s
Vertical-flight speed V_A	5 m/s
Minimum distance of UAVs D_{min}	50 m

A. TRAINING EFFICIENCY OF DDPG OPTIMIZATION METHOD

We first evaluate the training performance with different common learning hyperparameters, such as learning rate and batch size. In every episode, 50 UEs are arbitrarily distributed over the square place of $[50, 150]$, $[350, 450]$, and one UAV starts at an arbitrary position.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3012756, IEEE Access

TABLE 2: Main Hyperparameters of MADDPG

Parameter	Value
Episodes D	1000
Epochs T_p	200
Rate of soft weight updating τ	0.01
Random noise ς	$\varsigma \sim \mathcal{N}(0,1)$
Mini-batch size M_b	32
Discount rate γ	0.9
Learning rate δ	0.0001
Replay buffer $\mathcal{B}$	1000
Optimizer of MADDPG framework	RMSPropOptimizer

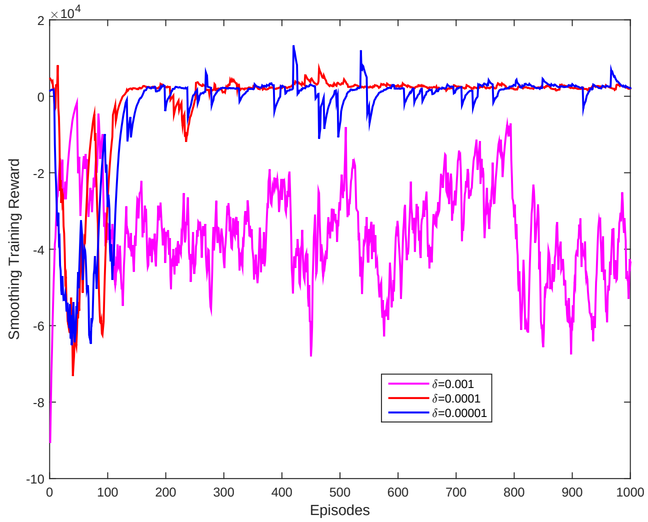


FIGURE 2: Smoothing training reward with different learning rates δ.

Figure 2 demonstrates the training performance with varied learning rates δ. In all three cases, the low smoothing training rewards are obtained at the beginning of training process. With the training episodes increasing, the training rewards have an obviously tendency to increase and converge in the cases of $\delta = 10^{-4}$ and $\delta = 10^{-5}$. Moreover, when the learning rate δ increases, fewer training episodes are needed to achieve the minimum QoS requirement of each UE. The converging speed of $\delta = 10^{-4}$ is faster than that of $\delta = 10^{-5}$. Nevertheless, if the learning rate is too large, the algorithm may converge to a local optimum, which can be seen in the case of $\delta = 10^{-3}$. Thus, considering the training reward and training speed, the learning rate $\delta = 10^{-4}$ is a proper choice in the next several experiments.

Next, the training performance with different batch sizes M_b is presented in Figure 3. The smoothing training rewards are very low at the first 100 training episodes in all cases. With the training episodes increasing, the rewards of all cases tend to converge within about 500 training episodes. However, as the training episodes continue to increase, when the batch size M_b is too small (i.e, $M_b = 16$), the training reward has a tendency to decrease. Furthermore, if the batch

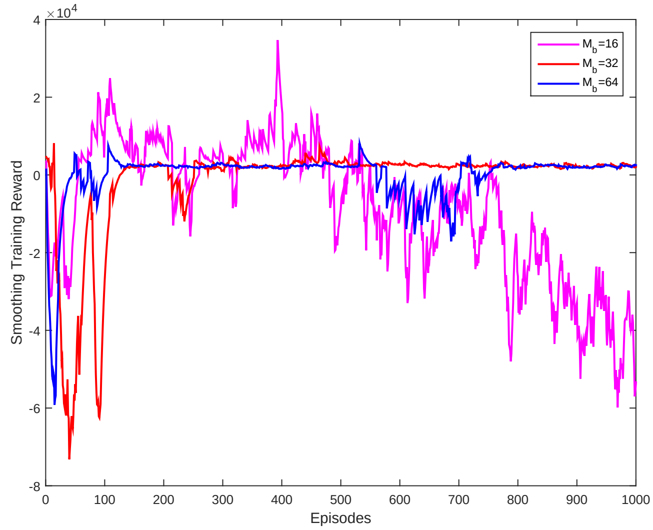


FIGURE 3: Smoothing training reward with different memory size M_b.

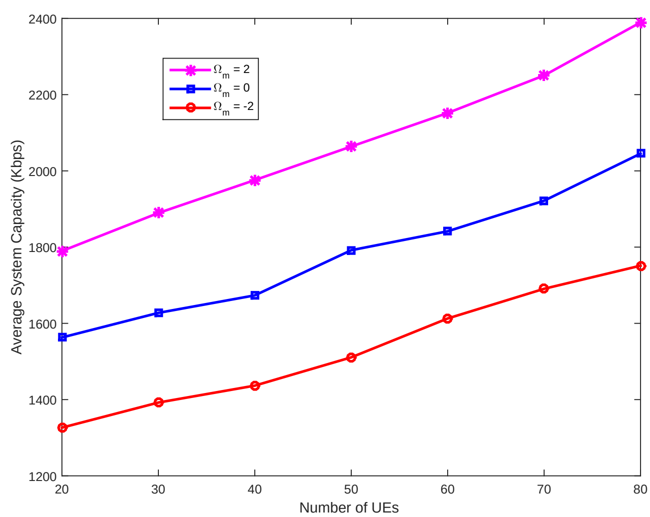


FIGURE 4: Average system capacity with different numbers of UEs M and minimum QoS requirements Ω_m.

size M_b is relatively large (i.e, $M_b = 64$), the curve of the smoothing training reward may be less stable. The training reward of $M_b = 32$ has an obviously tendency to increase and converge. Therefore, the batch size $M_b = 32$ is a good choice by considering the training reward.

Then, the training performance with different numbers of UEs is evaluated in one UAV scenario. Figure 4 shows the average system capacity with various UEs' numbers M and minimum QoS requirements Ω_m. When the minimum QoS requirement of each UE is achieved, the more the number of UEs is served, the higher system capacity can be achieved. When the UEs' number is small, only a few epoches needed to achieve the minimum QoS requirement of each UE, which causes the low capacity. Moreover, the average system

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3012756, IEEE Access

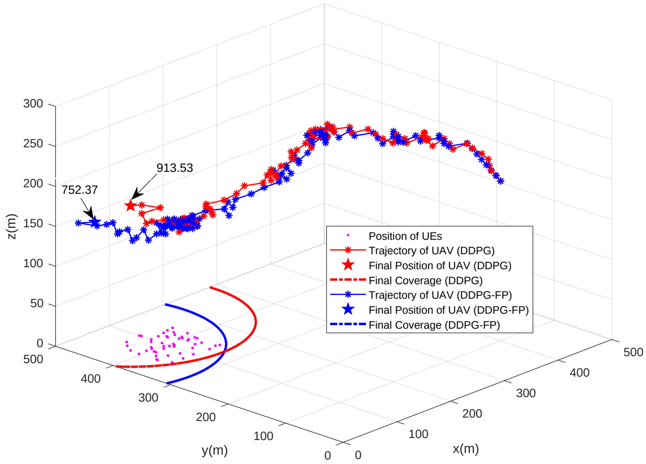
capacity increases with the minimum QoS requirements Ω_m increasing. The capacity in the case of $\Omega_m = 2$ is always higher than that of $\Omega_m = 0$ and $\Omega_m = -2$.

B. OPTIMIZATION PERFORMANCE WITH DIFFERENT METHODS

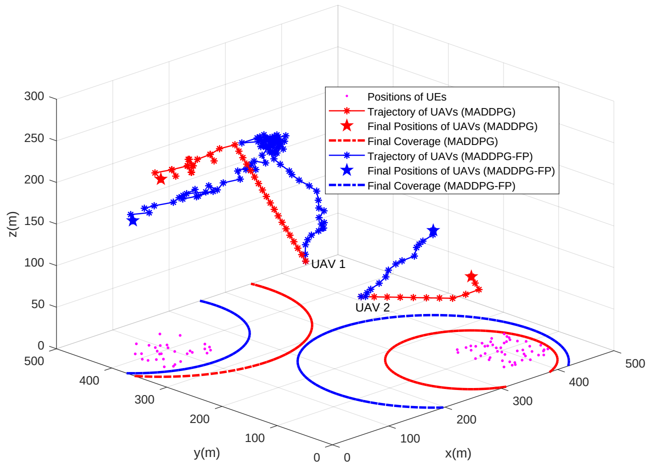
Finally, the performance with different optimization approaches is evaluated. We compare our proposed MADDPG method with the following four other optimization baselines. A degraded version of our MADDPG method with the fixed power allocation strategy ($p_i(t) = P_{max}$) is considered, which is denoted as MADDPG-FP. Multi-agent actor-critic (MAAC) approach is considered without the target network and experience replay strategies. In the random scheme, at every time slot, each UAV randomly select a moving angle, a vertical moving distance, a horizontal moving distance, and a transmission power within the constraints. With the greedy strategy, each UAV takes a discretized action to obtain the maximum immediate reward in a distributed manner at every time slot.

Figure 5 shows the joint strategy of the three-dimensional trajectory and power allocation. The performances of the DDPG (red star) and DDPG-FP (blue star) methods are considered. Figure 5(a) and Figure 5(b) present one possible joint strategy in the single-UAV scenario and the two-UAV scenario, respectively. In each episode, each UAV starts from the same position to provide UEs with the wireless service. In the two scenarios, both the two approaches demonstrate the same flying direction of UAV to cover all UEs. Moreover, in the two-UAV scenario, the two UAVs can cover all UEs in each hotpot without overlapping by using the two optimization algorithms. Furthermore, unlike the DDPG-FP strategy with fixed power allocation, the DDPG approach jointly considers the tradeoff between spectrum efficiency and interference. Thus, the DDPG method always results in the higher network utility (913.53 for the single-UAV scenario and 1933.2 for two-UAV scenario) than that of DDPG-FP (752.37 for the single-UAV scenario and 1432.9 for two-UAV scenario).

Figure 6 plots the average system capacity (ASC) with different minimum QoS requirements Ω_m and optimization methods. In order to meet with the minimum QoS requirements Ω_m of all UEs, the five optimization approaches (DDPG, DDPG-FP, AC, random, and greedy) are considered. In the greedy strategy, since the UAV takes actions to maximize the immediate reward at each time slot, the highest system capacity can be achieved by comparing with the other four approaches at all minimum QoS requirements. With Ω_m increasing, the system capacity achieved by the greedy method keeps almost unchanged. As for the other four approaches (DDPG, DDPG-FP, AC, and random), the UAV takes actions to make sure that all UEs are covered by the UAV with the minimum QoS requirements satisfied. As Ω_m increases, the average system capacity rises in all the four methods (DDPG, DDPG-FP, AC, and random). In the case of certain high minimum QoS requirement Ω_m, these



(a) Single-UAV scenario with 50 UEs



(b) Multi-UAV scenario with 80 UEs

FIGURE 5: Positions of the UEs and UAVs with the trajectory design and power allocation strategies ($\Omega_m = 2$).

four methods may achieve the similar system capacity with the greedy approach. Furthermore, the DDPG method always obtains a slightly higher capacity than that of the other three approaches (DDPG-FP, AC, and random).

Finally, the performance of different optimization approaches with various numbers of UAVs K is evaluated. Here, the average network utility (ANU), ASC, and computational time (CT) are considered in both the uniform scenario (Table 3) and non-uniform scenario (Table 4). In the uniform scenario, 80 UEs are distributed over K hotspots uniformly. As for the non-uniform scenario, the UEs are scattered based on the non-uniform distribution. Notice that when the number of UAV K is equal to one, the single-agent DRL approaches (DDPG, DDPG-FP and AC) are utilized to address the JTDPA issue, instead of the multi-agent DRL methods (MADDPG, MADDPG-FP, and MAAC).

Since all UEs are covered with the minimum QoS requirements satisfied, all methods can obtain the high ASC and ANU in both the uniform scenario and non-uniform

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3012756, IEEE Access

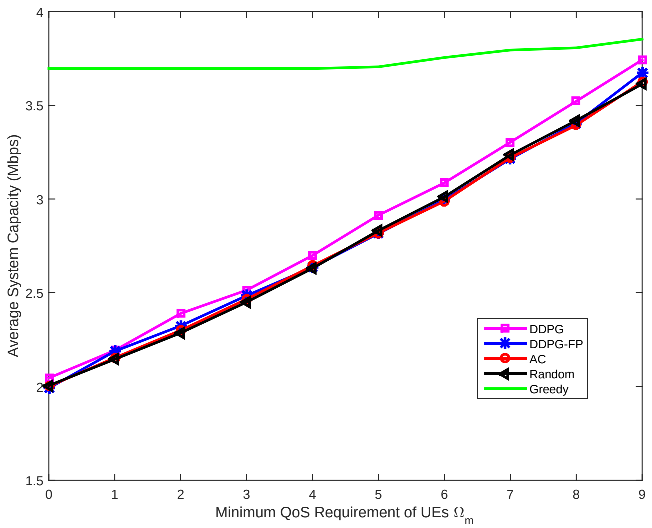


FIGURE 6: Average system capacity with different minimum QoS requirements Ω_m $(M = 80)$.

TABLE 3: Performance with the uniform distribution of UEs ($\Omega_m = 0$, $D = 200$ and $M = 80$)

K	Method	ASC(Mbps)	ANU	CT(sec.)
K=1	DDPG	2.11	$0.88e3$	34.23
	DDPG-FP	2.01	$0.83e3$	**18.76**
	AC	2.01	$0.84e3$	29.79
	Random	1.96	$0.81e3$	62.55
	Greedy	**3.59**	$\mathbf{1.81e3}$	247.45
K=2	MADDPG	3.65	$1.63e3$	63.56
	MADDPG-FP	3.48	$1.55e3$	**33.18**
	MAAC	3.53	$1.59e3$	41.83
	Random	3.35	$1.49e3$	113.54
	Greedy	**5.42**	$\mathbf{2.74e3}$	381.49
K=3	MADDPG	4.95	$2.28e3$	90.48
	MADDPG-FP	4.95	$2.27e3$	**44.63**
	MAAC	4.69	$2.17e3$	50.65
	Random	4.67	$2.15e3$	149.53
	Greedy	**6.89**	$\mathbf{3.48e3}$	403.09

scenario. With the UAVs' number K increasing, the ASC, ANU, and CT of all methods increase. The ASC and ANU in the uniform scenario are always smaller than that in the non-uniform scenario, which is closer to the real multi-UAV networks. Moreover, among the five approaches, since the greedy method obtains the actions to maximize the immediate reward at each time slot, the largest ASC and ANU can always be achieved with huge computational time. As for the random approach, the smallest ASC and ANU are obtained by randomly selecting the actions. In the three learning methods, our MADDPG approach can obtain a higher ASC and ANU than that of the other two learning methods (MADDPG-FP, MAAC) with less computational complexity in most cases, especially in the non-uniform scenario. In the non-uniform scenario, the ASC and ANU of our MADDPG method are about more than 15% of that of the other two learning approaches with $K = 3$, respectively.

Furthermore, notice that only when all UAVs cover all hotspots without overlapping and all UEs' minimum QoS

TABLE 4: Performance with the non-uniform distribution of UEs ($\Omega_m = 0$, $D = 200$ and $M = 80$)

K	Method	ASC(Mbps)	ANU	CT(sec.)
$K = 1$	DDPG	2.66	$1.16e3$	50.18
	DDPG-FP	2.34	$0.99e3$	63.19
	AC	1.95	$0.81e3$	**32.56**
	Random	2.03	$0.85e3$	66.55
	Greedy	**3.79**	$\mathbf{1.93e3}$	240.32
$K = 2$	MADDPG	3.85	$1.73e3$	77.22
	MADDPG-FP	3.42	$1.53e3$	105.18
	MAAC	3.43	$1.53e3$	**50.08**
	Random	3.43	$1.55e3$	117.62
	Greedy	**5.74**	$\mathbf{2.95e3}$	392.09
$K = 3$	MADDPG	5.53	$2.57e3$	139.98
	MADDPG-FP	5.30	$2.45e3$	315.95
	MAAC	4.70	$2.17e3$	**63.24**
	Random	4.58	$2.11e3$	171.01
	Greedy	**7.03**	$\mathbf{3.56e3}$	504.44

requirements are satisfied, Algorithm 1 can go to the next episode. Considering that the maximum epoch is 200 in each episode. That is to say, even if very few epochs are needed in the MADDPG approach, the maximum difference between the epochs in all methods is no more than 200 in each episode, which will be a quite small difference in computational time.

V. CONCLUSION

In this paper, an MADRL approach is proposed to obtain the optimal JTDPA policy in multi-UAVs networks. The JTDPA optimization problem is modeled to achieve the maximum long-term reward while satisfying the minimum QoS requirements of all UEs. Furthermore, considering the non-convex and combinatorial characteristics of the JTDPA optimization issue, an MADRL method is investigated to design the three-dimensional trajectory and transmission power of UAVs. By combining the experience replay with target networks, the MADDPG algorithm can effectively obtain the optimal policy with the fast converging speed. Simulation results indicate that our method can provide better performance compared with other approaches.

REFERENCES

[1] Y. Zeng, R. Zhang, and T. J. Lim, "Wireless communications with unmanned aerial vehicles: Opportunities and challenges," *IEEE Communications Magazine*, vol. 54, no. 5, pp. 36-42, 2016.
[2] N. Zhao, Y. Liang, and Y. Pei, "Dynamic contract incentive mechanism for cooperative wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 10970-10982, 2018.
[3] M. Mozaffari, W. Saad, M. Bennis, Y. Nam, and M. Debbah, "A tutorial on UAVs for wireless networks: Applications, challenges, and open problems," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2334-2360, 2019.
[4] Z. Hu, Z. Zheng, L. Song, T. Wang and X. Li, "UAV offloading: Spectrum trading contract design for UAV-assisted cellular networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 9, pp. 6093-6107, 2018.
[5] Q. Wang, Z. Chen, H. Li, and S. Li, "Joint power and trajectory design for physical-layer secrecy in the UAV-aided mobile relaying system," *IEEE Access*, vol. 6, pp. 62849-62855, 2018.
[6] G. Zhang, Q. Wu, M. Cui, and R. Zhang, "Securing UAV communications via joint trajectory and power allocation," *IEEE Transactions on Wireless Communications*, vol. 18, no. 2, pp. 1376-1389, 2019.

[7] S. Zhang, H. Zhang, Q. He, K. Bian, and L. Song, “Joint trajectory and power optimization for UAV relay networks,” *IEEE Communications Letters*, vol. 22, no. 1, pp. 161-164, 2018.

[8] X. Liu, Y. Liu, Y. Chen and L. Hanzo, “Trajectory design and power control for multi-UAV assisted wireless networks: A machine learning approach,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 7957-7969, 2019.

[9] Y. Gao, H. Tang, B. Li, and X. Yuan, “Joint trajectory and power design for UAV-enabled secure communications with no-fly zone constraints,” *IEEE Access*, vol. 7, pp. 44459-44470, 2019.

[10] Y. Wu, J. Xu, L. Qiu, and R. Zhang, “Capacity of UAV-enabled multicast channel: Joint trajectory design and power allocation,” in *IEEE International Conference on Communications (ICC)*, 2018, pp. 1-7.

[11] G. Yang, R. Dai, and Y.-C. Liang, “Energy-efficient UAV backscatter communication with joint trajectory and resource optimization,” in *IEEE International Conference on Communications (ICC)*, 2019, pp. 1-6.

[12] F. Ding, L. Lv, J. Pan, X.K. Wan, and X.B. Jin, “Two-stage gradient-based iterative estimation methods for controlled autoregressive systems using the measurement data,” *International Journal of Control Automation and Systems*, vol. 18, no. 4, pp. 886-896, 2020.

[13] R. S. Sutton and A. G. Barto, Reinforcement learning: An introduction. MIT Press Cambridge, 1998.

[14] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y.-C. Liang, and D. I. Kim, “Applications of deep reinforcement learning in communications and networking: A survey,” *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3133-3174, 2019.

[15] H. Yang, X. Xie, and M. Kadoch, “Intelligent resource management based on reinforcement learning for ultra-reliable and low-latency IoV communication networks,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 4157-4169, 2019.

[16] G. Zhao, Y. Li, C. Xu, Z. Han, Y. Xing, and S. Yu, “Joint power control and channel allocation for interference mitigation based on reinforcement learning,” *IEEE Access*, vol. 7, pp. 177254-177265, 2019.

[17] V. Mnih, K. Kavukcuoglu, D. Silver, et al., “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, no. 7540, pp. 529-533, 2015.

[18] Y. Su, X. Lu, Y. Zhao, L. Huang, and X. Du, “Cooperative communications with relay selection based on deep reinforcement learning in wireless sensor networks,” *IEEE Sensors Journal*, vol. 19, no. 20, pp. 9561-9569, 2019.

[19] Y. Sun, M. Peng, and S. Mao, “Deep reinforcement learning based mode selection and resource management for green fog radio access networks,” *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1960-1971, 2019.

[20] J. Chen, S. Chen, Q. Wang, B. Cao, G. Feng, and J. Hu, “iRAF: A deep reinforcement learning approach for collaborative mobile edge computing IoT networks,” *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 7011-7024, 2019.

[21] Y. He, F. R. Yu, N. Zhao, V. C. Leung, and H. Yin, “Software-defined networks with mobile edge computing and caching for smart cities: A big data deep reinforcement learning approach,” *IEEE Communications Magazine*, vol. 55, no. 12, pp. 31-37, 2017.

[22] N. Zhao, Y. Cheng, Y. Pei, Y.-C. Liang, and D. Niyato. “Deep reinforcement learning for trajectory design and power allocation in UAV networks,” in *IEEE International Conference on Communications (ICC)*, 2020, pp. 1-6.

[23] S. Zheng and H. Liu, “Improved multi-agent deep deterministic policy gradient for path planning-based crowd simulation,” *IEEE Access*, vol. 7, pp. 147755-147770, 2019.

[24] N. Zhao, Y.-C. Liang, D. Niyato, Y. Pei, M. Wu, and Y. Jiang, “Deep reinforcement learning for user association and resource allocation in heterogeneous cellular networks,” *IEEE Transactions on Wireless Communications*, vol. 18, no. 11, pp. 5141-5152, 2019.

[25] T. Chu, J. Wang, L. Codec, and Z. Li, “Multi-agent deep reinforcement learning for large-scale traffic signal control,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 1086-1095, 2019.

[26] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, “Unmanned aerial vehicle with underlaid device-to-device communications: Performance and tradeoffs,” *IEEE Transactions on Wireless Communications*, vol. 15, no. 6, pp. 3949-3963, 2016.

[27] A. Al-Hourani, S. Kandeepan, and S. Lardner, “Optimal LAP altitude for maximum coverage,” *IEEE Wireless Communications Letters*, vol. 3, no. 6, pp. 569-572, 2014.

[28] C. Yan, L. Fu, J. Zhang and J. Wang, “A Comprehensive Survey on UAV Communication Channel Modeling," *IEEE Access*, vol. 7, pp. 107769-107792, 2019.

[29] F. Ding, X. Zhang, and L. Xu, “The innovation algorithms for multivariable state-space models,” *International Journal of Adaptive Control and Signal Processing*, vol. 33, no. 11, pp. 1601-1608, 2019.

[30] A. Neyman and S. Sorin, Stochastic games and applications. Kluwer Academic Publishers, 2003.

[31] M. J. Osborne, An introduction to game theory. Oxford University Press, 2004.

[32] T. p. Lillicrap, J, J. Hunt, A. Pritzel, N. Heess, T. Erez, and Y. Tassa, “Continuous control with deep reinforcement learning,” *Comput. Sci.*, vol. 8, no. 6, pp. 1-14, 2015.

[33] F. Ding, L. Xu, D.D. Meng, et al., “Gradient estimation algorithms for the parameter identification of bilinear systems using the auxiliary model,” *Journal of Computational and Applied Mathematics*, vol. 369, p. 112575, 2020.

[34] U. Challita, L. Dong, and W. Saad, “Proactive resource management for LTE in unlicensed spectrum: A deep learning perspective,” *IEEE Transactions on Wireless Communications*, vol. 17, no. 7, pp. 4674-4689, 2018.

[35] Y. Wei, F. R. Yu, M. Song, and Z. Han, “User scheduling and resource allocation in hetnets with hybrid energy supply: An actor-critic reinforcement learning approach,” *IEEE Transactions on Wireless Communications*, vol. 17, no. 1, pp. 680-692, 2018.