Trajectory Design and Access Control for Air–Ground Coordinated Communications System With Multiagent Deep Reinforcement Learning

Ruijin Ding, *Graduate Student Member, IEEE*, Yadong Xu, Feifei Gao, *Fellow, IEEE*, and Xuemin Shen, *Fellow, IEEE*

***Abstract*—Unmanned-aerial-vehicle (UAV)-assisted communications has attracted increasing attention recently. This article investigates air–ground coordinated communications system, in which trajectories of air UAV base stations (UAV-BSs) and access control of ground users (GUs) are jointly optimized. We formulated this optimization problem as a mixed cooperative–competitive game, where each GU competes for the limited resources of UAV-BSs to maximize its own throughput by accessing a suitable UAV-BS, and UAV-BSs cooperate with each other and design their trajectories to maximize the defined *fair throughput* to improve the total throughput and keep the GU fairness. Moreover, the action space of GUs is discrete, while that of UAV-BS is continuous. To tackle this hybrid action space issue, we transform the discrete actions into continuous action probabilities and propose a multiagent deep reinforcement learning (MADRL) approach, named air–ground probabilistic multiagent deep deterministic policy gradient (AG-PMADDPG). With well-designed rewards, AG-PMADDPG can coordinate two types of agents, UAV-BSs and GUs, to achieve their own objectives based on local observations. Simulation results demonstrate that AG-PMADDPG can outperform the benchmark algorithms in terms of throughput and fairness.**

***Index Terms*—Air–ground coordinated communications, fair communication, multiagent deep reinforcement learning (MADRL), unmanned aerial vehicle (UAV) trajectory design, user access control.**

I. Introduction

UNMANNED-AERIAL-VEHICLE (UAV)-assisted communications as the complement of terrestrial communications has drawn increasing attentions from both academia and industry [1]–[3]. UAVs equipped with wireless transceivers can serve as base stations (UAV-BSs) to provide communication service for ground users (GUs). With the 3-D flying capability, UAV-BSs can offload burst data traffic for GUs in hotspot areas, such as stadiums, cinemas, theaters, etc. [4], [5] and provide flexible capacity for QoS guaranty during rush hours and special events [6].

Compared with the terrestrial communications system, UAV-assisted communications system has three advantages: 1) the UAV-to-ground links have higher Line-of-Sight (LoS) probabilities than terrestrial BS-to-user links [7], which can improve the robustness and performance of the system; 2) the fully controlled mobility [8], [9] of UAVs allows the UAV-assisted network to improve the Quality of Service (QoS) through trajectory design [10]–[13]; and 3) the onboard calculation and caching modules enable UAV-BSs to execute computing tasks [14]–[16].

However, the trajectory design of UAV-BSs is a sequential optimization problem, which has tremendous decision variables and is nonconvex and is very difficult to be solved directly [17]. Most existing works [1]–[22] simplify the original nonconvex problem into multiple convex subproblems and solve the subproblems iteratively until reaching the convergency. Such simplification makes the origin problem trackable with the cost of accuracy. Besides, the computational complexity increases exponentially as the number of UAVs and GUs increases [13]. Recalculation of the optimization process is also mandated when the environment changes, which constrains the implementation and deployment of those algorithms in dynamic scenarios. Moreover, optimization-based methods often need global information to find the global optima, which is hard to be ensured in real scenarios.

Recently, deep reinforcement learning (DRL) [23] has been applied in UAV-assisted communications [24]–[27], since it can solve the sequential problem modeled as a Markov decision process (MDP) [28]. In DRL, the hard-to-optimize problems can be transformed into maximizing accumulative reward through reward design. Although the DRL approaches consume much time in the training phase, only a little time is needed to make decisions with well-trained deep neural networks (DNNs) after sufficient training. Besides,

Manuscript received September 28, 2020; revised December 26, 2020; accepted February 17, 2021. Date of publication February 25, 2021; date of current version April 7, 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2018AAA0102401; in part by the National Natural Science Foundation of China under Grant 61831013 and Grant 61771274; and in part by the Beijing Municipal Natural Science Foundation under Grant L182042 and Grant 4212002. *(Corresponding author: Feifei Gao.)*

Ruijin Ding and Feifei Gao are with the Institute for Artificial Intelligence, State Key Laboratory of Intelligent Technologies and Systems, Beijing National Research Center for Information Science and Technology, and Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: drj17@mails.tsinghua.edu.cn; feifeigao@ieee.org).

Yadong Xu is with the Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing 100084, China (e-mail: xuyd17@mails.tsinghua.edu.cn).

Xuemin Shen is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: sshen@uwaterloo.ca).

Digital Object Identifier 10.1109/JIOT.2021.3062091

multiagent DRL (MADRL) [29] can solve problems with multiple agents, and each agent can make decisions based on its local information.

In this article, we investigate air–ground coordinated communications system in which both UAV-BS trajectories and GU access control are optimized to further improve the system performance. This is different from most previous works [1]–[15], [17]–[19], [24]–[27], [30]–[32] which focus on either UAVs or GUs. Specifically, each GU requests to access one UAV-BS to maximize its own throughput and the UAV-BSs decide their flying directions to maximize the *fair throughput* which is jointly defined by the total throughput of all GUs and the fairness among GUs. Since the resources of UAV-BSs are limited and the UAV-BSs share the same objective, the optimization problem is a mixed cooperative–competitive game. Besides, the action space of GUs is discrete while that of UAV-BSs is continuous, and there arises a hybrid action space issue. The main contributions of this article are summarized as follows:

1) We study the air–ground coordinated communications system where the UAV-BS trajectories and GU access control are optimized together instead of just optimizing UAV-BS trajectories or GU access control.
2) Based on MADDPG [33], we propose a probabilistic multiagent deep deterministic policy gradient (PMADDPG) approach to address the problems with hybrid action space. Specifically, PMADDPG transforms the discrete actions into continuous probabilities and samples an action according to the distribution. We then prove the existence of policy gradient, and thus DNN can be optimized in the way of training MADDPG.
3) We apply PMADDPG to the air–ground coordinated communications system and propose air–ground PMADDPG (AG-PMADDPG) to enable GUs to maximize their own throughput and UAV-BSs to provide fair and high-throughput communication service.
4) We analyze the air–ground coordinated communications system from the perspective of game theory. To be specific, we calculate the price of anarchy (POA) [34] of the induced game among competitive GUs and show that the objective would suffer from severe performance loss if GUs learn independently.

The remainder of this article is organized as follows. The literature review is conducted in Section II. Section III introduces the system model and formulates the optimization problem for UAV-BSs and GUs, respectively. In Section IV, we give a brief introduction to the MADDPG algorithm. Section V presents details of the proposed AG-PMADDPG. Simulation results are provided in Section VI. Finally, we conclude this article in Section VII.

Notations: In this article, scalars are denoted by lowercase letters, and vectors are denoted by boldface letters. The Euclidean norm of vector $\boldsymbol{a}$ is denoted by $\|\boldsymbol{a}\|$ and $\boldsymbol{a}[i]$ denotes the ith element of vector $\boldsymbol{a}$. $\mathbb{R}^M$ denotes the space of M-dimensional real vectors; and $\mathbb{E}_\pi[\cdot]$ denotes the expectation of a random variable following policy π.

II. Realted Work

A. UAV Trajectory Design and GU Access Control

In order to fully utilize the high mobility of UAV-BSs, there exist many works targeting at improving communication performance through UAV trajectory design. For instance, Zeng and Zhang [10] derived an energy consumption model of the fixed-wing UAV as a function of UAV's trajectory and proposed a sequential convex optimization-based trajectory design algorithm to maximize the energy efficiency. In [11], a closed-form energy consumption model for the rotary-wing UAV was derived, and a successive convex approximation (SCA)-based algorithm was proposed to optimize the hovering locations of the UAV. Zhan *et al.* [12] leveraged the classic traveling salesman problem (TSP) and applied convex optimization techniques to enable a UAV to collect data from distributed wireless sensor nodes through trajectory design. Cui *et al.* [13] proposed a penalty dual-decomposition (PDD)-based algorithm to maximize the minimum rate among GUs through joint trajectory design and resource allocation. Shi *et al.* [17] adopted the block coordinate descent (BCD) mechanism to devise a multidrone base station (DBS) 3-D trajectory planning and scheduling algorithm to improve user fairness and network performance.

On the other hand, there are several works focusing on GU optimization. Cao *et al.* [32] proposed a distributed DRL framework for GU access control in UAV-assisted communications. The GUs share a common neural network to make their own access decision independently according to the local network state. Based on the optimal transport theory, Mozaffari *et al.* [35] proposed a cell-association algorithm to minimize the average network delay. Given the locations of BSs and UAVs as well as the distributions of GUs, the optimal cell partitions of the UAVs and terrestrial BSs are determined.

However, these works only focus on either UAV trajectory design with fixed GU access or GU access control given predefined UAV trajectory. In this article, we optimize both UAV trajectory design and GU access control to further improve system performance.

B. UAV-Assisted Communications Enabled by DRL

There is another line of works applying DRL techniques in UAV-assisted communications. Liu *et al.* [24] proposed a DRL-based method to enable UAVs to provide energy efficient and fair communication coverage for GUs while preserving their connectivity. The proposed method maximizes an energy efficiency function with comprehensive consideration of connectivity, fairness, energy consumption, and communication coverage. In [25], a UAV positioning scheme was presented to find the optimal links between UAV nodes and fine-tune UAV positions to improve network performance. A deep Q-learning [deep Q network (DQN)]-based method is designed to handle dynamic swarm topology and time-varying link conditions. In order to allocate the resource of multiple UAVs dynamically, Cui *et al.* [26] proposed an agent-independent method, for which all agents share a common structure based on Q-learning, but conduct a

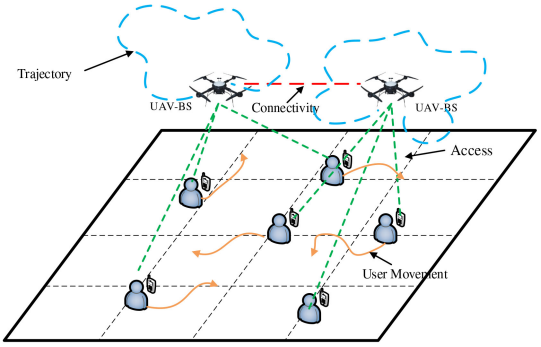


Fig. 1. UAV-BS network providing communication service for GUs.

decision algorithm independently. Each UAV selects its communicating GU, power level, and subchannel without any information exchange among UAVs. Ding *et al.* [27] proposed a DRL-based method to allow the UAV to adjust the flight speed and direction to provide energy-efficient service for GUs and allocate the frequency band resources to enhance fairness among GUs.

However, these works only deal with problems with only discrete action space or only continuous action space. In this article, we investigate the air–ground coordinated communications system with hybrid action space and propose PMADDPG to tackle this issue.

III. System Model and Problem Formulation

A. System Model

As shown in Fig. 1, we consider a communications system in which multiple open-access UAV-BSs fly horizontally to provide communication service for the GUs. We assume that there are M UAV-BSs, denoted by the set $\mathcal{M} \triangleq \{1, \ldots, M\}$, flying at the same fixed height H and speed V. The horizontal location of UAV-BS m at time t is denoted by $\boldsymbol{u}_m(t) \in \mathbb{R}^2$, $0 \le t \le T$, where T is the service time for the UAV-BSs. There are K GUs moving randomly on the ground, and the location of GU k is denoted by $\boldsymbol{w}_k(t) \in \mathbb{R}^2$, $k \in \mathcal{K} \triangleq \{1, \ldots, K\}$, $0 \le t \le T$. The M UAV-BSs share the total frequency bandwidth B, which is assigned equally to each UAV-BS.

The air–ground coordinated communications system is assumed to work in a time-slotted manner, i.e., the service time T is divided into multiple slots with duration δ_t, and then we have $t \in \mathcal{T} \triangleq \{0, 1, \ldots, \hat{T}\}$, where $\hat{T} = \lfloor T/\delta_t \rfloor$. The time slot length is short enough so that the flying directions of the UAV-BSs $\{\boldsymbol{e}_m(t)\}_{m\in\mathcal{M}}$ can be considered as constant with $\|\boldsymbol{e}_m(t)\| = 1$ during each time slot. Then, we have

$$\boldsymbol{u}_m(t+1) - \boldsymbol{u}_m(t) = V\boldsymbol{e}_m(t)\delta_t. \tag{1}$$

The communication links between UAVs and GUs are dominated by the LoS channel that can improve the system performance. We assume that the Doppler effects caused by the mobilities of the UAV-BSs are compensated. As a result, the time-varying channel gain between UAV-BS m and GU k follows the free-space path-loss model:

$$h_{k,m}(t) = \sqrt{\frac{\rho_0}{\|\boldsymbol{u}_m(t) - \boldsymbol{w}_k(t)\|^2 + H^2}} \tag{2}$$

where ρ_0 denotes the channel power gain at the reference distance 1 m.

We assume that all UAV-BSs have a common transmitting power P_t, and then the received power of GU k from UAV-BS m is

$$P^r_{k,m}(t) = P_t \cdot |h_{k,m}(t)|^2. \tag{3}$$

The signal-to-noise ratio (SNR) at GU k from UAV-BS m can be expressed as

$$\alpha_{k,m}(t) = \frac{P^r_{k,m}(t)}{n_0 B/M} \tag{4}$$

where n_0 is the noise power spectral density.

In the considered communication model, each GU can request one UAV-BS to access at every time slot. We introduce a binary variable $\eta_{k,m}(t) \in \{0, 1\}$ as the access indicator, with $\eta_{k,m}(t) = 1$ indicating that GU k requests UAV-BS m at time slot t and $\eta_{k,m}(t) = 0$ otherwise. Thus, we have

$$\sum_{m\in\mathcal{M}} \eta_{k,m}(t) = 1 \ \forall t \in \mathcal{T}. \tag{5}$$

If there are multiple GUs choosing the same UAV-BS to access, the UAV-BS will adopt time-division multiple access (TDMA) to serve the connected GUs. The time slot would be further divided equally into multiple subtimeslots for the connected GUs. Then, the achievable downlink throughput from the UAV-BS m to GU k at time slot t is

$$c_{k,m}(t) = \begin{cases} \frac{\eta_{k,m}(t)B}{MN_m(t)} \log_2(1 + \alpha_{k,m}(t))\delta_t, & N_m(t) > 0 \\ 0, & N_m(t) = 0 \end{cases} \tag{6}$$

where $N_m(t) = \sum_{k\in\mathcal{K}} \eta_{k,m}(t)$ is the number of GUs that request to access UAV-BS m at time slot t.

B. Problem Formulation

In this article, we aim to find GU access policy that determines how each GU chooses UAV-BS and find UAV-BS flying policy that determines how each UAV-BS moves.

1) GU: In the considered system, the objective of each GU is to maximize its long-term throughput by choosing its access UAV-BS. The throughput of GU k at time slot t is

$$d_k(t) = \sum_{m\in\mathcal{M}} c_{k,m}(t). \tag{7}$$

Then, the objective of GU k can be mathematically formulated as

$$\max_{\{\eta_{k,m}(t)\}_{t\in\mathcal{T}}} \sum_{t\in\mathcal{T}} d_k(t) \tag{8a}$$

$$\text{s.t.} \ \ \eta_{k,m}(t) \in \{0, 1\} \ \forall t \in \mathcal{T} \tag{8b}$$

$$\sum_{m\in\mathcal{M}} \eta_{k,m}(t) = 1 \ \forall t \in \mathcal{T}. \tag{8c}$$

Note that $d_k(t)$ also depends on the decisions of other GUs. Nevertheless, to keep user privacy, there is no information exchange among GUs. That is to say, each GU can only make decisions based on its local information.

Since the time and frequency resource are limited, problem (8) induces a noncooperative game among the

GUs [36]–[38], where *Nash equilibrium* is the canonical solution [39]. The self-interested agents in this game select actions to maximize their utility functions, i.e., the throughput, which makes the equilibria inefficient. POA [34] is the most popular measure of the inefficiency of equilibria, which is defined as the ratio between the worst objective function value of the game equilibrium and that of an optimal outcome. We will calculate the POA of this allocation game among GUs in Appendix A, demonstrating that the GUs will suffer from a severe loss in terms of the objectives if each GU learns independently.

2) UAV-BS: The UAV-BSs in this air–ground communications system share the same objective, that is, to provide high-quality communication service for GUs through designing trajectories of UAV-BSs $\{\boldsymbol{u}_m(t)\}_{m\in\mathcal{M},t\in\mathcal{T}}$.

However, directly maximizing the total throughput of all GUs may cause unfairness problem, since the UAV-BSs may fly around some GUs, leading to poor communication service for the other GUs. We introduce Jain's fairness index [40] to evaluate the fairness among GUs

$$f(t)=\frac{(\sum_{k=1}^{K} d_k(t))^2}{K(\sum_{k=1}^{K} d_k(t)^2)}. \tag{9}$$

According to Cauchy–Buniakowsky–Schwarz inequality, we can easily get $0 < f(t) \leq 1$. It can be seen that the larger the throughput differences among GUs are, the smaller the fairness index will be. In other words, a larger fairness index implies a fairer communication service.

In order to balance the total throughput and the fairness among the GUs, we define the *fair throughput* as

$$\bar{d}_f(t)=f(t)\cdot\sum_{k\in\mathcal{K}} d_k(t). \tag{10}$$

Then, the objective of the UAV-BS m is to maximize the long-term fair throughput through designing its trajectory. Mathematically, the problem can be expressed as

$$\max_{\{\boldsymbol{u}_m(t)\}_{t\in\mathcal{T}}} \sum_{t\in\mathcal{T}} \bar{d}_f(t) \tag{11a}$$

$$\text{s.t. } \|\boldsymbol{u}_m(t+1)-\boldsymbol{u}_m(t)\|=V\delta_t \ \forall t\in\mathcal{T}, \tag{11b}$$

$$\boldsymbol{u}_m(0)=\boldsymbol{u}_m^0 \tag{11c}$$

$$\boldsymbol{b}_l \leq \boldsymbol{u}_m(t) \leq \boldsymbol{b}_u{}^1 \ \forall t\in\mathcal{T} \tag{11d}$$

$$\|\boldsymbol{u}_i(t)-\boldsymbol{u}_m(t)\|\geq\delta_d \ \forall t\in\mathcal{T} \ \forall i\neq m \tag{11e}$$

where $\boldsymbol{u}_m^0$ denotes the initial location of UAV-BS m; $\boldsymbol{b}_l$ and $\boldsymbol{b}_u$ are the lower and upper boundary coordinates of the UAV-BS service region, respectively; and δ_d is the safe distance between two UAV-BSs.

According to (1), $\{\boldsymbol{u}_m(t)\}$ and $\{\boldsymbol{e}_m(t)\}$ are linear with each other, thus the optimization variables $\{\boldsymbol{u}_m(t)\}_{t\in\mathcal{T}}$ in (11a) can be replaced by $\{\boldsymbol{e}_m(t)\}_{t\in\mathcal{T}}$. The advantage of this approach is that constraint (11b) can be ignored. Hence, problem (11) can be reexpressed as

$$\max_{\{\boldsymbol{e}_m(t)\}_{t\in\mathcal{T}}} \sum_{t\in\mathcal{T}} \bar{d}_f(t) \tag{12a}$$

$$\text{s.t. } \boldsymbol{u}_m(0)=\boldsymbol{u}_m^0 \tag{12b}$$

$$\boldsymbol{b}_l \leq \boldsymbol{u}_m(t) \leq \boldsymbol{b}_u \ \forall t\in\mathcal{T} \tag{12c}$$

$$\|\boldsymbol{u}_i(t)-\boldsymbol{u}_m(t)\|\geq\delta_d \ \forall t\in\mathcal{T} \ \forall i\neq m. \tag{12d}$$

We assume that UAV-BSs can communicate with each other on an extra narrow channel for exchanging some necessary information. In other words, the available information for the UAV-BSs is the same. Besides, the UAV-BSs are fully cooperative, since they optimize a common objective function in a distributed manner.

As a result, the optimization problem is a mixed cooperative–competitive game, where UAV-BSs are cooperative and GUs are competitive. In summary, we aim to: 1) find the access policies for GUs that can maximize the long-term throughput of each GU and 2) find the control policies for UAV-BSs that can maximize the long-term fair throughput.

However, it is difficult for the conventional optimization algorithms to deal with the problems (8) and (12). This is because:

1) the GUs and UAV-BSs have different objectives. Specifically, the objectives of all UAV-BSs are the same, which means the UAVs need cooperation. While the objectives of GUs are exclusive, and all GUs compete for limited communication resources of UAV-BSs;
2) the GUs and UAV-BSs interact with each other. More concretely, the trajectories of UAV-BSs affect the objectives of GUs, while the access policies of GUs influence the objectives of UAV-BSs. Moreover, the common objective of UAV-BSs is related to the sum of the GUs' objectives. Traditional optimization algorithms cannot consider the relationships among the objectives and decisions;
3) the throughput maximization problem (8) for each GU is an integer programming problem, which is nonconvex;
4) there is no information exchange among GUs, while traditional optimization algorithms require global information;
5) GUs move randomly while both (8) and (12) are to maximize the long-term throughput that is influenced by the trajectories of GUs. However, conventional optimization algorithms require predefined GU trajectories, which is not available due to the randomness.

Fortunately, MADRL can solve problems with multiple objectives. Each agent focuses on maximizing its own long-term accumulative reward. MADRL can address the interaction among GUs and UAV-BSs and the relationships among the objectives of GUs and UAV-BSs can be considered through proper reward design. The derived integer programming problem corresponds to the discrete action space problem in DRL. Moreover, MADRL enables agents to make decisions based on current local information and does not need to obtain the trajectories of GUs in advance.

IV. Preliminary of MADRL

MADRL algorithms can be applied to solve problems that are modeled as *Markov games* [41]. A Markov game for N agents can be defined by a tuple $(\mathcal{S},\mathcal{A},\mathcal{R},\mathcal{P},\gamma)$, where $\mathcal{S}$ is the set of states describing the environment; $\mathcal{A}\triangleq\{\mathcal{A}_1,\ldots,\mathcal{A}_N\}$ is the set of joint actions for all agents; $\mathcal{R}\triangleq\{\mathcal{R}_1,\ldots,\mathcal{R}_N\}$ is the set of reward functions that map the state and agents' actions to rewards, i.e., $\mathcal{R}:\mathcal{S}\times\mathcal{A}\mapsto\mathbb{R}^N$; $\mathcal{P}$ is the state transition function that describes the probability

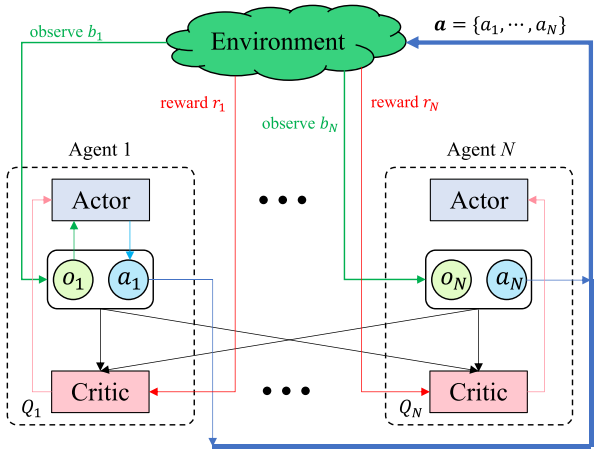


Fig. 2. Framework of MADDPG.

of the next state after all agents take joint actions under the current state, i.e., $\mathcal{P} : \mathcal{S} \times \mathcal{A}_1 \times \cdots \times \mathcal{A}_N \times \mathcal{S} \mapsto [0, 1]$; and γ is the reward discount factor. Many Markov games are partially observable, where the agents can only observe part of the environment state [42]. The observation set of all agents is denoted by $\mathcal{O} = \{\mathcal{O}_1, \ldots, \mathcal{O}_N\}$. At each time slot, the environment state is $s(t)$. Each agent receives a local observation based on the state $o_i(t) = b_i(s(t))$ and chooses the action based on the observation $a_i(t) = \pi_i(o_i(t))$, where b_i and π_i denote the observation function and the policy of agent i. Let $\boldsymbol{o}(t) = \{o_1(t), \ldots, o_N(t)\}$ and $\boldsymbol{a}(t) = \{a_1(t), \ldots, a_N(t)\}$ denote the joint observations and actions, respectively. After choosing $\boldsymbol{a}(t)$, the agents receive the corresponding rewards $\boldsymbol{r}(t) = \{r_1(t), \ldots, r_N(t)\}$ based on reward functions $\mathcal{R}$. Then, the environment turns into the next state $s(t+1)$ according to state transition function $\mathcal{P}$.

We choose MADDPG, the multiagent extension of DDPG [43], as the starting point of our design. MADDPG belongs to the paradigm of *centralized training and decentralized execution*. In the centralized training phase, the agents can use extra information that is not needed in the execution part, e.g., the observations and actions of other agents, while in the decentralized execution phase, each agent can make the decision based on local observation. The objective of each agent is to find a policy that maximizes its own expected long-term accumulative reward. Moreover, MADDPG is an actor–critic algorithm [44]. As shown in Fig. 2, each agent has an actor network $\pi_i(o_i; \theta_i^\pi)$ with weights θ_i^π for decentralized execution. The actor network takes the local observation as input and then outputs the action. The critic network $Q_i(\boldsymbol{o}, \boldsymbol{a}; \theta_i^Q)$ of each agent can evaluate the output of the actor network through accessing all agents' observations and actions during the centralized training phase. MADDPG adopts two techniques to avoid training oscillations or divergence: 1) experience replay and 2) target network. At each time slot, the agents store corresponding experience tuples $(\boldsymbol{o}(t), \boldsymbol{a}(t), \boldsymbol{r}(t), \boldsymbol{o}(t+1))$ into a replay buffer with size $\mathcal{B}$. If the replay buffer is full, the newly generated experience tuple will replace the old one. The training data of the actor and critic networks are sampled in batches from the replay buffer. The random samples break the correlation among sequential samples and reduce the training oscillation. Besides, both actor and critic networks have corresponding target actor $\pi_i'(o_i; \theta_i^{\pi'})$ and critic $Q_i'(\boldsymbol{o}, \boldsymbol{a}; \theta_i^{Q'})$ that share the same architectures as θ_i^π and θ_i^Q. The critic network of agent i is updated by minimizing the mean square error (MSE) loss

$$L\left(\theta_i^Q\right) = \frac{1}{N_b} \sum_j \left[y_i(j) - Q_i\left(\boldsymbol{o}(j), \boldsymbol{a}(j); \theta_i^Q\right) \right]^2 \tag{13}$$

where

$$y_i(j) = r_i(j) + \gamma Q_i' \left(\boldsymbol{o}(j+1), \left\{ \pi_1'\left(\boldsymbol{o}(j+1); \theta_1^{\pi'}\right), \ldots, \pi_N'\left(\boldsymbol{o}(j+1); \theta_N^{\pi'}\right) \right\}; \theta_i^{Q'}\right) \tag{14}$$

is the update target, and N_b is the batch size. The actor network of agent i can be updated by minimizing the loss

$$L\left(\theta_i^\pi\right) = \frac{1}{N_b} \sum_j -Q_i\left(\boldsymbol{o}(j), \{a_1(j), \ldots, a_i, \ldots, a_N(j)\}; \theta_i^Q\right) \tag{15}$$

where $a_i = \pi_i(o_i(j); \theta_i^\pi)$. The parameters of the target networks are updated by tracking the learned networks

$$\theta' \leftarrow \varepsilon\theta + (1 - \varepsilon)\theta' \tag{16}$$

with $\varepsilon \ll 1$.

V. Multiagent Deep Reinforcement Learning for Air–Ground Coordinated Communications System

In this section, we propose AG-PMADDPG for the air–ground coordinated communications system, where UAV-BSs and GUs are regarded as agents. Specifically, UAV-BSs are indexed by $\{1, \ldots, M\}$, and GUs are indexed by $\{M + 1, \ldots, M + K\}$. At every time slot, the GUs choose one UAV-BS to request access based on their local observations, while the UAV-BSs determine their flying directions to provide fair and high-quality service for all GUs.

A. Observation Space

1) UAV-BS: As mentioned in Section III-A, the downlink throughput is directly related to the channels between UAV-BSs and GUs. However, due to the high mobility of UAV-BSs, the accurate channel state information is hard to obtain. In comparison, it is easier to obtain the locations of UAV-BSs and GUs through the global positioning system (GPS) sensors equipped on them. GUs report their locations before data transmission to their selected access UAV-BSs. Then, UAV-BSs exchange the information with each other in order to know the locations of all UAV-BSs and GUs. Therefore, the observation of each UAV-BS includes the locations of all UAV-BSs and GUs and can be formulated as a vector with $2(M + K)$ elements, that is

$$o_m(t) \triangleq \{\{\boldsymbol{u}_m(t)\}_{m \in \mathcal{M}}, \{\boldsymbol{w}_k(t)\}_{k \in \mathcal{K}}\}. \tag{17}$$

2) GU: Due to the demand of user privacy, only the following local information is included in the observation of each GU.

1) The current received signal power $\{P^r_{k,m}(t)\}_{m\in\mathcal{M}}$ from all UAV-BSs is available, which can be directly measured by GU k.
2) The outdated received signal power $\{P^r_{k,m}(t-1)\}_{m\in\mathcal{M}}$ can also be included in the observation.
3) The connected UAV-BS of GU k at time slot $t-1$ $\{\eta_{k,m}(t-1)\}_{m\in\mathcal{M}}$ can be stored as the current observation.
4) The number of connected GUs of each UAV-BS $\{N_m(t-1)\}_{m\in\mathcal{M}}$ can be fed back from each UAV-BS.
5) The throughput $d_k(t-1)$ is included such that the neural network can learn the relationship among the received signal power, the connected UAV-BS, and the throughput.

In summary, there are $4M+1$ elements in the observation of GU k, that is

$$o_{M+k}(t) \triangleq \big\{\{P^r_{k,m}(t)\}_{m\in\mathcal{M}}, \{P^r_{k,m}(t-1)\}_{m\in\mathcal{M}}, \{\eta_{k,m}(t-1)\}_{m\in\mathcal{M}}, \{N_m(t-1)\}_{m\in\mathcal{M}} d_k(t-1)\big\}. \tag{18}$$

B. Action Space

1) UAV-BS: The UAV-BSs choose their flying directions to provide fair communication service. We use the polar angle $\varphi_m(t) \in (-\pi, \pi)$ to denote the flying direction, and then the action of UAV-BS m is

$$a_m(t) \triangleq \varphi_m(t). \tag{19}$$

Thus, we have

$$\boldsymbol{e}_m(t)[1] = \cos\varphi_m(t) \tag{20a}$$

$$\boldsymbol{e}_m(t)[2] = \sin\varphi_m(t). \tag{20b}$$

For convenience, we apply normalized representation for the flying direction of UAV-BS m as

$$\lambda_{\varphi_m}(t) = \varphi_m(t)/\pi \tag{21}$$

where $\lambda_\varphi(t) \in (-1, 1)$.

2) GU: At each time slot, each GU needs to request one UAV-BS to access. As a result, there are M elements in the action of GU k

$$a_{M+k}(t) \triangleq \big\{\eta_{k,m}(t)\big\}_{m\in\mathcal{M}}. \tag{22}$$

Therefore, the actions of UAV-BSs are continuous, while each GU has a discrete action space. However, traditional DRL algorithms can only solve the problems where the actions are all continuous or discrete, but cannot deal with the hybrid action space.

To tackle this issue, we propose PMADDPG on the basis of MADDPG, which transforms the discrete actions to continuous action probabilities, and then samples the action according to the probability distribution. As a result, the outputs of actor networks are all continuous, and we can adopt the training framework of MADDPG.

We then apply PMADDPG to the air–ground coordinated communications system and formulate AG-PMADDPG. In AG-PMADDPG, the actor network of GU k is denoted by $\bar{\pi}_{M+k}(o_{M+k}; \bar{\theta}_{M+k})$, and outputs the probability distribution $\bar{a}_{M+k}(t) = \bar{\pi}_{M+k}(o_{M+k}(t); \bar{\theta}_{M+k})$, where the ith element of $\bar{a}_{M+k}$ is the probability of GU k accessing UAV-BS i. The action of GU k is $a_{M+k} = \psi(\bar{a}_{M+k})$, where $\psi(\cdot)$ means sampling according to the probability distribution. The outputs of the actor networks are $\bar{\boldsymbol{a}} = \{a_1, \ldots, a_M, \bar{a}_{M+1}, \ldots, \bar{a}_{M+K}\}$.[2] The actions of GUs in (13)–(15) are all replaced by corresponding probability distribution $\{\bar{a}_{M+k}\}_{k\in\mathcal{K}}$.

Theorem 1: Consider learning a discrete policy $a = \pi(s)$ for a single-agent MDP. The actor network outputs the probability of each action $\bar{\boldsymbol{a}} = \bar{\boldsymbol{\pi}}(s; \bar{\theta})$. The policy gradient of agent objective function $J(\bar{\theta})$ is

$$\nabla_{\bar{\theta}} J(\bar{\theta}) = \int_{\mathcal{S}} \rho^{\bar{\boldsymbol{\pi}}}(s) \nabla_{\bar{\theta}} \bar{\boldsymbol{\pi}}(s; \bar{\theta}) \nabla_{\bar{\boldsymbol{a}}} Q^{\bar{\boldsymbol{\pi}}}(s, \bar{\boldsymbol{a}})|_{\bar{\boldsymbol{a}}=\bar{\boldsymbol{\pi}}(s;\bar{\theta})} \mathrm{d}s \tag{23}$$

where $\rho^{\bar{\pi}}(s)$ is the discounted state distribution under policy $\bar{\boldsymbol{\pi}}(s; \bar{\theta})$ (see Appendix B for detailed proof).

Remark 1: The existence of policy gradient in (23) means that we can use DDPG to solve single-agent MDP with discrete action space through converting discrete actions to continuous action probabilities. Since MADDPG is a heuristic multiagent extension of DDPG, Theorem 1 makes it possible for PMADDPG to solve hybrid action space issue to solve hybrid action space issue with PMADDPG.

C. Reward Design

1) UAV-BS: Based on (12), the common part of rewards for all UAV-BSs can be defined as

$$r^{ft}(t) = \kappa_r \bar{d}_f(t) \tag{24}$$

where κ_r is the ratio coefficient between reward and fair throughput. If UAV-BS m violates boundary constraint (12c), then a penalty r^b is deducted from (24). Similarly, if safe distance constraint (12d) is violated, then the penalty is r^d. The boundary violation indicator is denoted by $\xi^b_m(t)$ with $\xi^b_m(t) = 1$ implying UAV-BS m flies out of the boundary, and $\xi^b_m(t) = 0$ otherwise. The safe distance violation indicator is denoted by $\xi^d_m(t)$ with $\xi^d_m(t) = 1$ implying UAV-BS m flies into the safety zones of other UAV-BSs, and $\xi^d_m(t) = 0$ otherwise.

Therefore, the reward of UAV-BS m is

$$r_m(t) = r^{ft}(t) - \Big(\xi^b_m(t) \cdot r^b + \xi^d_m(t) \cdot r^d\Big). \tag{25}$$

2) GU: According to (8), the reward of GU k can be directly defined as the achievable throughput

$$r_{M+k}(t) = \kappa_r d_k(t). \tag{26}$$

D. Neural Networks

1) Actor Network: The actor network takes the local observation of the agent as input and then outputs the action.

[2]Note that the executed actions of UAV-BSs are the same as the outputs of corresponding actor networks.

a) UAV-BS: To output the actions defined in (21), we use the activation function *tanh*, that is

$$\lambda_{\varphi_m}(t) = \tanh(\chi_{\varphi_m}(t)) \tag{27}$$

where $\chi_{\varphi_m}(t)$ is the preactivation value of $\lambda_{\varphi_m}(t)$. However, the activation function tanh is susceptible to a saturation problem that leads to gradient vanishing [27]. We then add a preactivation penalty to the actor loss, and (15) can be rewritten as

$$L\left(\theta_i^{\pi}\right) = \frac{1}{N_b}\sum_{j} -Q_i\Big(\boldsymbol{o}(j), \{a_1(j), \ldots, a_i, \ldots, a_N(j)\}; \theta_i^{Q}\Big) + \kappa_l\Big(\max(\chi_{\varphi}(j) - \zeta, 0) + \max(-\chi_{\varphi}(j) - \zeta, 0)\Big)^2 \tag{28}$$

where $a_i = \pi_i(o_i(j); \theta_i^{\pi})$, κ_l is a weight factor, and ζ is the saturation value of tanh. Minimizing the actor loss prevents the preactivation value from staying in the saturation area and thus eliminates gradient vanishing.

b) GU: The output of a GU's actor network is the probability of selecting each UAV-BS to access. Thus, the activation function in the output layer is *softmax*.

2) Critic Network: The critic network of agent i takes the observations $\boldsymbol{o}$ and the actions $\bar{\boldsymbol{a}}$ of all agents as input, and then outputs the centralized action-value function $Q_i(\boldsymbol{o}, \bar{\boldsymbol{a}}; \theta_i^{Q})$ to provide guidance for the training of the actor network.

E. Training Algorithm

The proposed training algorithm is episodic with episode length $\hat{T}$. At the beginning of each episode, every UAV-BS is located at its initial point, and GUs are randomly distributed on the ground. At each time slot, GUs move around at a fixed speed and random direction.

In the training phase, each UAV-BS inputs its observation $o_m(t)$ into the actor network $\pi_m(o_m(t); \theta_m^{\pi})$ and outputs the flying direction, while the actor network of each GU takes the local observation $o_{M+k}(t)$ as input and outputs the action probability distribution $\bar{a}_{M+k}$. Exploration noise $\mathcal{N}$ is required to encourage exploration. The noise is normally distributed with zero mean and deviation $\sigma_{\mathcal{N}}$. Exploration is necessary in the training phase to prevent the learned policy from falling into local optimum. Then, GU k samples an action a_{M+k} based on $\bar{a}_{M+k}$. Next, the environment proceeds with agents taking the joint action $\boldsymbol{a}(t)$, and transitions to the next state $s(t+1)$. The agents obtain their rewards $\boldsymbol{r}(t)$ and next observations $\boldsymbol{o}(t+1)$. If UAV-BS m flies out of the boundary or into the safety zones of other UAV-BSs, its reward receives a penalty, and the corresponding flight is canceled. The experience tuple $(\boldsymbol{o}(t), \bar{\boldsymbol{a}}(t), \boldsymbol{r}(t), \boldsymbol{o}(t+1))$ is stored in the replay buffer. If the buffer is full, then the batches of experience tuples are sampled uniformly, and actor and critic networks of all agents are updated by minimizing the corresponding losses. Finally, the target networks of all agents are updated by slowly tracking the learned networks. The training algorithm is summarized in Algorithm 1.

In the execution phase, the decisions of UAV-BS flying direction and GU access are based on the well-trained networks.

Algorithm 1 Centralized Training of PMADDPG for the Air–Ground Coordinated Communications System

Input: Architectures of UAV-BS actor and critic networks, architectures of GU actor and critic networks, buffer size $\mathcal{B}$, episode length $\hat{T}$, batch size N_b, UAV-BS speed V, the number of UAV-BSs M and the number of GUs K.

Output: Well-trained actor network parameters of all UAV-BSs and GUs.

Randomly initialize the actor networks of all agents, $\{\theta_i^{\pi}\}_{i=\{1,\ldots,M+K\}}$, target actor networks $\{\theta_i^{\pi'}\}_{i=\{1,\ldots,M+K\}} = \{\theta_i^{\pi}\}_{i=\{1,\ldots,M+K\}}$, critic networks $\{\theta_i^{Q}\}_{i=\{1,\ldots,M+K\}}$ and target critic networks $\{\theta_i^{Q'}\}_{i=\{1,\ldots,M+K\}} = \{\theta_i^{Q}\}_{i=\{1,\ldots,M+K\}}$.
Initialize the experience replay buffer.
for each episode **do**
  Initialize locations of UAV-BSs and GUs.
  for each time slot t **do**
    for each UAV-BS m **do**
      The UAV-BSs get the GUs' locations and formulate observation $o_m(t)$.
      Choose action $a_m(t) = \pi_m(o_m(t); \theta_m^{\pi}) + \mathcal{N}$, where $\mathcal{N}$ is exploration noise.
    end for
    for each GU k **do**
      Get observation $o_{M+k}(t)$.
      Calculate probability distribution of discrete actions $\bar{a}_{M+k}(t) = \bar{\pi}_{M+k}(o_{M+k}(t); \bar{\theta}_{M+k}^{\pi})$.
      Sample action $a_{M+k}(t)$ based on probability $\bar{a}_{M+k}(t)$.
    end for
    All agents take actions.
    if a UAV-BS flies out of the boundary of service region or into the safety zone of other UAV-BSs **then**
      Recalibrate the UAV-BS position.
    end if
    The agents obtain their corresponding rewards $\boldsymbol{r}(t)$ and get the next observations $\boldsymbol{o}(t+1)$.
    Store $(\boldsymbol{o}(t), \bar{\boldsymbol{a}}(t), \boldsymbol{r}(t), \boldsymbol{o}(t+1))$ in experience replay buffer.
  end for
  if Replay buffer is full **then**
    Sample several random minibatches of N_b experience tuples from replay buffer.
    for each agent i **do**
      Calculate the critic target y_i according to (14).
      Update the critic network θ_i^{Q} by minimizing the critic loss (13).
      Update the actor network θ_i^{π} by minimizing the actor loss (15) and (28).
      Soft updates for the target networks:
      $\theta_i^{Q'} = \varepsilon\theta_i^{Q} + (1-\varepsilon)\theta_i^{Q'}$.
      $\theta_i^{\pi'} = \varepsilon\theta_i^{\pi} + (1-\varepsilon)\theta_i^{\pi'}$.
    end for
  end if
end for

VI. Simulation Results

In this section, we conduct simulations to validate the proposed AG-PMADDPG.

A. Simulation Settings

The service region of UAV-BSs is limited to be a square area with the size of 2 km × 2 km, while the flying altitudes of all UAV-BSs are 100 m. The origin of the coordinate system is in the center of the region, and thus $\boldsymbol{b}_l$ and $\boldsymbol{b}_u$ are

TABLE I
DRL-RELATED PARAMETERS

Notation	Meaning	Simulation value
N_b	Batch size	32
$\mathcal{B}$	Experience replay buffer size	50000
γ	Discount factor	0.999
ε	Soft update rate	0.005
$\sigma_{\mathcal{N}}$	Exploration noise deviation	0.1
κ_r	Ratio coefficient between reward and fair throughput	10^{-6}
r^b	Penalty of boundary constraint violation	50
r^d	Penalty of safe distance constraint violation	100

$[-1000, -1000]$ and $[1000, 1000]$, respectively. At the beginning of each episode, the UAV-BSs are evenly and equally distributed on a circle with radius 500 m. For example, if there are two UAV-BSs, the initial locations of them are $[500, 0]$ and $[-500, 0]$, respectively. All UAV-BSs fly at a common speed of $V = 10$ m/s, and a safe distance of $\delta_d = 5$ m between UAV-BSs needs to be maintained. The total frequency bandwidth $B = 1$ MHz, and each UAV-BS transmits signals with power $P_t = 10$ dBm. The noise power spectral density is -170 dBm/Hz, and the reference channel power gain is $\rho_0 = -50$ dB. GUs are randomly distributed in the service region and move around with random direction. Each GU has a constant speed that follows uniform distribution on the interval [1 m/s, 5 m/s]. The service time is assumed to be $T = 200$ s, which is discretized with time slot size $\delta_t = 0.2$ s.

The observation of each agent is normalized to $[0, 1]$. All hidden layers of all networks are activated by *ReLU* functions. The ADAM optimizers [45] with learning rate of 0.001 are applied to update all actor and critic networks of the agents. The DRL-related parameters are shown in Table I.

B. Performance Analysis

We compare the proposed AG-PMADDPG, with three other benchmark algorithms.

1) *Access Control PMADDPG (AC-PMADDPG):* In order to demonstrate the superiority of air–ground coordinated communications system and PMADDPG, we only optimize the GU access control based on PMADDPG.
2) *Common DQN [32]:* The DQN algorithms in [32] are designed for GU access control. All GUs share a common network. The experiences of all GUs are shared as the training data.
3) *Distributed DQN:* Inspired by [32], each GU has its own network to train in distributed DQN, and the training data of each GU are private.

The UAV-BS trajectories of the three benchmarks are all circles with the same radius.

In order to compare AG-PMADDPG with the other three benchmarks, we plot the training curves of the four algorithms in the simplest scenario, in which there are two UAV-BSs and two GUs. As shown in Fig. 3, AG-PMADDPG outperforms the other three benchmarks after sufficient training for

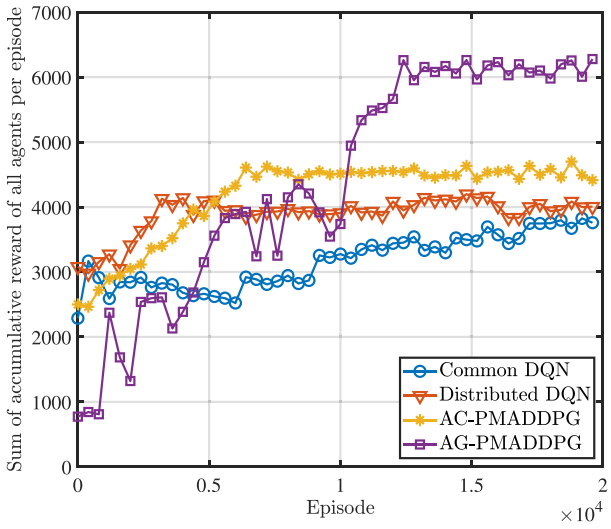


Fig. 3. Sum of the accumulative reward of all agents per episode. There are two UAV-BSs and two GUs. Each data point is the maximum value in every 400 episodes.

the reason that AG-PMADDPG can optimize the UAV trajectories and GU access control at the same time. However, we can observe that in the first 4000 episodes, the sum accumulative reward of AG-PMADDPG is smaller than that of the other three benchmarks. This is because at the very beginning, the UAV-BSs fly around for exploration and may fly far away from the GUs, which leads to small throughput. The UAV-BSs may even violate the boundary constraint (12c) and receive corresponding penalty. During the training process, the performance of AG-PMADDPG oscillates significantly, because there is no clear label information in DRL-based algorithms. Moreover, even though only access control is considered, AC-PMADDPG achieves better performance than the other DQN-based algorithms, which validates the analysis in Appendix A. The centralized critic in PMADDPG can utilize the observations of all agents, while in the DQN-based algorithms, each agent can only focus on its own reward and thus the system incurs the loss in the total rewards of all agents, i.e., POA. Compared with common DQN, distributed DQN converges much faster because common DQN needs to learn a common policy for all agents, which is much more complex than the distributed way. Besides, distributed DQN converges faster than AG-PMADDPG, because searching policy in the discrete action space is simpler than in the continuous probability action space. In the simplest scenario, we can easily obtain the optimal policy, that is, each UAV-BS serves one GU and flies just above it. The maximum SNR can be obtained as 33 dB when the UAV-BSs are just above the corresponding GUs. As a result, the possible maximum throughput for each GU is $B/2 \times \log_2(1 + 2000)T = 1.0967 \times 10^9$. Since the throughput of two GUs is the same, the fair index is 1, and then the fair throughput of each UAV is 2.1934×10^9. Thus, the possible maximum sum accumulative reward is 6.58×10^3. However, the sum of the accumulative reward of AG-PMADDPG is about 6.2×10^3, which does not achieve the maximum. This is because the UAV-BSs need to fly to

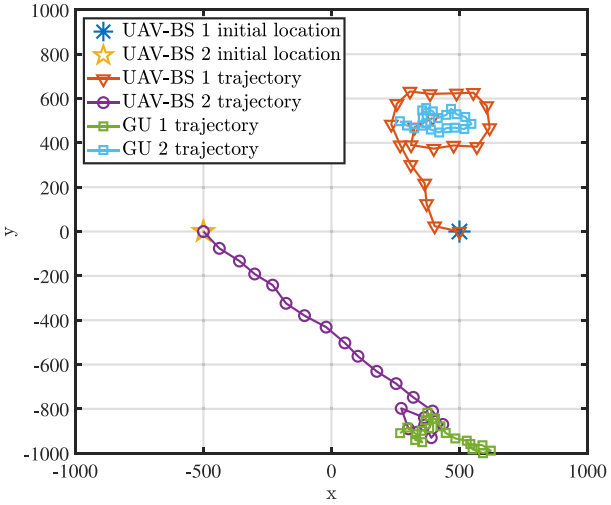


Fig. 4. Trajectories of UAVs and GUs.

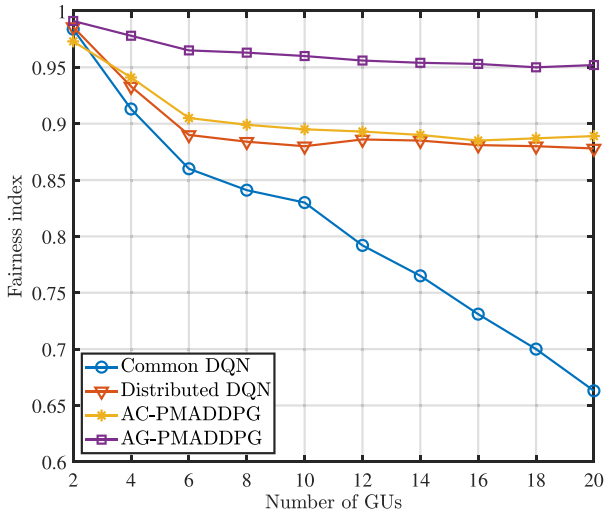


Fig. 5. Impact of the number of GUs on the fairness index. There are two UAV-BSs.

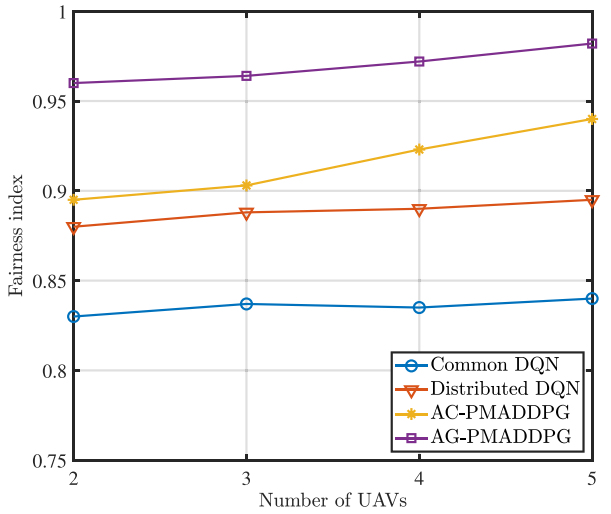


Fig. 6. Impact of the number of UAVs on the fairness index. There are ten GUs.

the GUs first, and the SNR cannot achieve its maximum value during the flight.

We plot the trajectories of UAVs and GUs in Fig. 4. It can be observed that each UAV-BS flies directly to the corresponding GU, and then flies around it. The trajectories are very close to the optimal trajectories.

Fig. 5 illustrates the relationships between the fairness index and the number of GUs. The number of UAVs is fixed to 2. We can observe that as the number of GUs increases, the fairness index of the four algorithms has some decline, in which AG-PMADDPG has the smallest decline. The fairness index of AG-PMADDPG decreases from 0.995 to around 0.95, and tends to be flat. Both AC-PMADDPG and distributed DQN yield the similar performance, and the fairness indexes of them drop from 0.985 to around 0.88. With trajectory design, AG-PMADDPG can achieve a higher fairness index compared with AC-PMADDPG and distributed DQN. The reason that the fairness index tends to be flat may be that when the number of GUs reaches a certain level, the GUs are almost distributed throughout the service region, and thus further increasing the number of GUs has little impact on the fairness index. The fairness index of common DQN keeps decreasing, because the increase of the number of GUs brings more difficulty in finding a common access control policy for all GUs.

In Fig. 6, we plot the fairness index for different numbers of UAV-BSs. The number of GUs is fixed to 10. We can see that the fairness indexes of two PMADDPG-based algorithms increase significantly as the number of UAV-BSs increases, while the growth of fairness indexes of two DQN-based algorithms is not obvious. This is because more UAV-BSs means larger action space for GUs, which leads to more difficulty in the training task. Since DQN-based algorithms cannot leverage the global information, the benefits of increasing the number of UAV-BSs cannot be fully utilized. Besides, the increase of fairness index in common DQN is less than that in distributed DQN because of the difficulty in finding a common policy for all GUs. On the other hand, the fairness index of AC-PMADDPG increases slightly more than that of AG-PMADDPG, because the closer the fairness index is to 1, the harder it could be improved.

The sum of all GUs' long-term throughput in one episode is shown in Fig. 7. We observe an interesting phenomenon that as the number of GUs increases, the total throughput of AG-PMADDPG decreases from 2026 to 1880, while the total throughput of the other three benchmarks has a certain degree of growth. When the number of GUs is small, the UAV-BSs can fly next to the GUs to achieve high throughput. However, the effect of trajectory design is decreasing as the number of GUs increases, and UAV-BSs cannot fly close to GUs all the time. On the other hand, the UAV-BSs of the three benchmarks have fixed trajectories. As the number of GUs increases, there is a higher probability that GUs appear near the trajectories, which makes the frequency band utilization more efficient. Note that due to the superiority of AG-PMADDPG, even though the total throughput of AG-PMADDPG decreases, it is still higher than that of the three benchmarks.

Fig. 8 plots the minimum throughput of all GUs in one episode. We can observe that as the number of GUs increases,

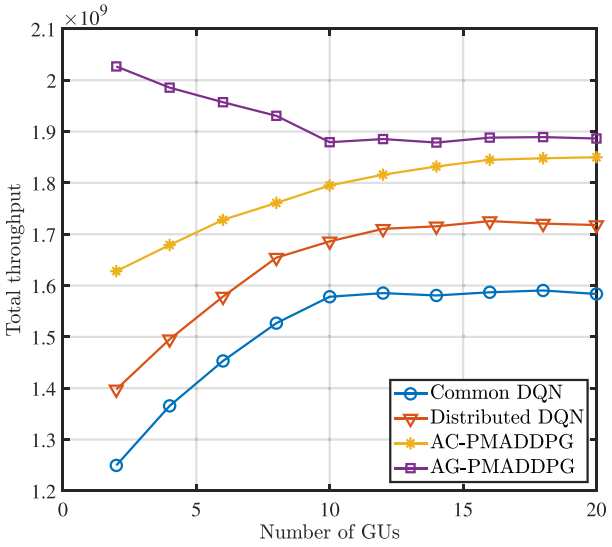


Fig. 7. Impact of the number of GUs on total throughput. There are two UAV-BSs.

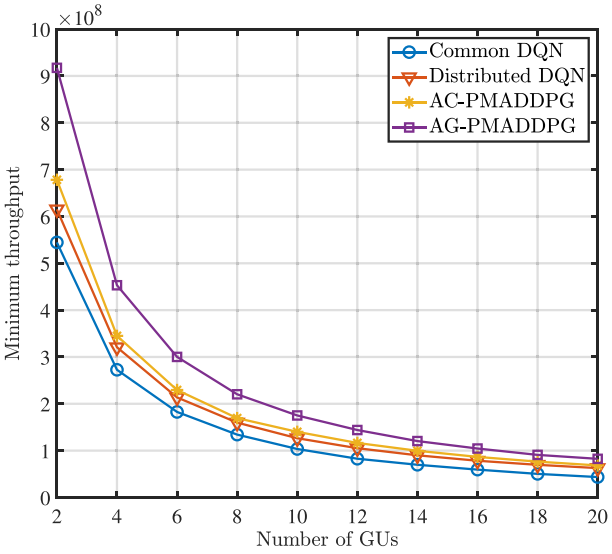


Fig. 8. Impact of the number of GUs on minimum throughput. There are two UAV-BSs.

the minimum throughput of all algorithms decreases. This is because more GUs sharing limited communication resources means that each GU will be allocated with fewer resources. If the air–ground coordinated communications system requires the throughput of each GU to be higher than a threshold, then AG-PMADDPG allows the UAV-BSs to serve 4–8 GUs more than the benchmarks.

VII. Conclusion

In this article, we have proposed AG-PMADDPG for the air–ground coordinated communications system. The proposed AG-PMADDPG enables UAV-BSs to provide fair communication service for GUs on the ground via coordinating both UAV-BSs and GUs. Specifically, each GU maximizes its own throughput by choosing proper UAV-BS to access, and each GU maximizes the fair throughput through trajectory design.

TABLE II
Payoff Matrix of the Induced Game

	$a_2=1$	$a_2=2$
$a_1=1$	$\frac{l_{1,1}}{2}, \frac{l_{2,1}}{2}$	$l_{1,1}, l_{2,2}$
$a_1=2$	$l_{1,2}, l_{2,1}$	$\frac{l_{1,2}}{2}, \frac{l_{2,2}}{2}$

Simulation results have demonstrated that AG-PMADDPG can achieve much better performance than the existing benchmarks in terms of fairness index, total throughput, and minimum throughput. For future work, we will investigate the frequency band allocation of the UAV-BSs to further improve the system performance.

Appendix A
Game-Theoretic View of GU Access Control

We focus on the communications system at time t. Given the locations of UAV-BSs, let us consider the induced normal-form game $\mathcal{G} = (\mathcal{K}, \mathcal{A}^g, u)$ for GUs, where:

1) $\mathcal{K}$ is the set of players, i.e., GUs, indexed by k;
2) $\mathcal{A}^g = \mathcal{A}_{M+1} \times \cdots \times \mathcal{A}_{M+K}$, where $\mathcal{A}_{M+k} = \{1, \ldots, M\}$ is the action set of GU k. And $a_{M+k} \in A_{M+k}$ is the action selected by GU k. Besides, $\boldsymbol{a}^g = (a_{M+1}, \ldots, a_{M+K}) \in \mathcal{A}^g$ denotes the action profile;
3) $u = (u_1, \ldots, u_K)$, and $u_k : \mathcal{A}^g \mapsto \mathbb{R}$ is a real-valued utility function for GU k, i.e., the achievable downlink throughput.

Recall the aforementioned notations, the utility function u_k is

$$u_k = \sum_{m=1}^{M} c_{k,m} = \sum_{m=1}^{M} \eta_{k,m} \frac{B}{N_m} \log_2(1 + \alpha_{k,m})\delta. \quad (29)$$

Let $l_{k,m}$ denote $B \cdot \log_2(1 + \alpha_{k,m}) \cdot \delta$. Then, there is $u_k = \sum_{m=1}^{M} (l_{k,m}/N_m \eta_{k,m})$.

A natural solution concept of this induced game is the Nash equilibrium, where each GU has no incentive to deviate its current strategy. In the multiagent setting, when each agent learns to maximize its own utility, the outcome usually converges to some equilibrium. Consider the simplest setting where $M = K = 2$, the payoff matrix is shown in Table II, where GU 1's actions are represented by the two rows while GU 2's actions are represented by the two columns, and the first and second number of each entry represent the utility of GU 1 and GU 2, respectively. Therefore, we can compute the equilibrium case by case.

1) If $l_{1,1} \leq (l_{1,2}/2)$ and $l_{2,1} < (l_{2,2}/2)$, then both GUs prefer to request UAV-BS 2.
2) If $l_{1,1} \leq (l_{1,2}/2)$ and $l_{2,1} \geq (l_{2,2}/2)$, then GU 1 requests UAV-BS 2 while GU 2 requests UAV-BS 1.
3) If $l_{1,1} \geq 2l_{1,2}$ and $l_{2,1} > 2l_{2,2}$, then both GUs will request UAV-BS 1.
4) If $l_{1,1} \geq 2l_{1,2}$ and $l_{2,1} \leq 2l_{2,2}$, then $\boldsymbol{a}^g = (1, 2)$ forms the equilibrium.
5) If $(l_{1,2}/2) < l_{1,1} < 2l_{1,2}$ and $l_{2,1} \leq (l_{2,2}/2)$, then $\boldsymbol{a}^g = (1, 2)$ forms the equilibrium.
6) If $(l_{1,2}/2) < l_{1,1} < 2l_{1,2}$ and $l_{2,1} \geq 2l_{2,2}$, then $\boldsymbol{a}^g = (2, 1)$ forms the equilibrium.

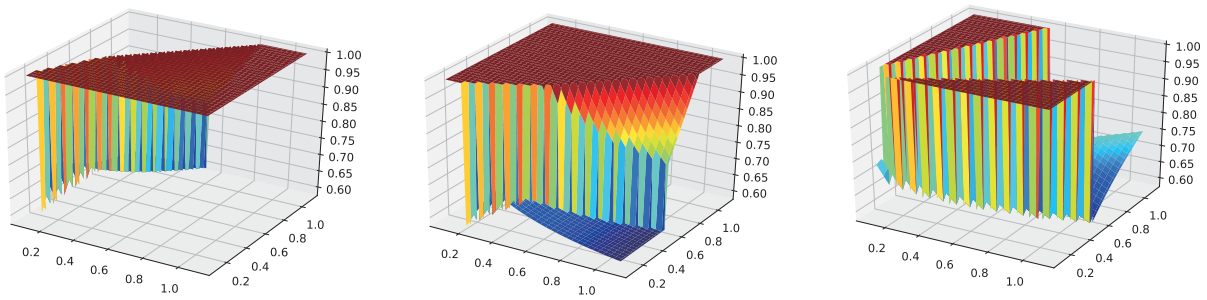


Fig. 9. POA as the function of $l_{1,1}$ and $l_{1,2}$, where $l_{1,1}, l_{1,2} \in [0.1, 1.1]$, (a) case where $l_{2,1} = 0.25, l_{2,2} = 0.51$, the minimum of POA is 0.5820; (b) case where $l_{2,1} = 0.51, l_{2,2} = 0.25$, the minimum is 0.5820; and (c) case where $l_{2,1} = l_{2,2} = 0.5$, the minimum is 0.5807.

7) If $(l_{1,2}/2) < l_{1,1} < 2l_{1,2}$ and $(l_{2,2}/2) < l_{2,1} < 2l_{2,2}$, there does not exist a pure Nash equilibrium. Instead, there is a mixed-strategy equilibrium, where GU 1 requests UAV-BS 1 and UAV-BS 2 with probability p and $1-p$, respectively, and GU 2 requests UAV-BS 1 and UAV-BS 2 with probability q and $1-q$. Specifically, p and q can be derived as

$$p = \frac{2l_{2,1} - l_{2,2}}{l_{2,1} + l_{2,2}}, \quad q = \frac{2l_{1,1} - l_{1,2}}{l_{1,1} + l_{1,2}}$$

and it is easy to verify that $p, q \in (0, 1)$.

The maximum total throughput can be written as

$$\max\left\{\frac{l_{1,1} + l_{2,1}}{2}, l_{1,1} + l_{2,2}, l_{1,2} + l_{2,1}, \frac{l_{1,2} + l_{2,2}}{2}\right\}.$$

We use the definition POA to denote the ratio between the total throughput under the equilibrium and the maximum total throughput. To illustrate the value of POA, we fix the value of $l_{2,1}$ and $l_{2,2}$, and calculate the POA as the function of $l_{1,1}$ and $l_{1,2}$. As demonstrated in Fig. 9, the total throughput suffers from a loss under the equilibrium, and such loss is nearly half of the maximum in the worst case. This phenomenon implies that if GUs request UAV-BSs independently to maximize their own utilities, then the achievable total throughput will suffer severe loss even in the simplest setting.

Appendix B
Proof of Deterministic Policy Gradient for Discrete Action Problem

The proof follows along similar lines of the standard deterministic policy gradient (DPG) for MDPs in [46]. In a single-agent MDP with discrete action space $\mathcal{A} = \{1, 2, \ldots, M\}$, the agent receives the state of environment $s \in \mathcal{S}$ and then selects an action $a \in \mathcal{A}$. Next, the agent receives a numerical reward according to the reward function $r(s, a)$, and the state turns into $s' \in \mathcal{S}$ according to the state transition probability $p(s'|s, a)$.[3]

We denote the probability action by a vector $\bar{\boldsymbol{a}}$ with each element representing the probability of the corresponding action, i.e., the output of the actor network $\bar{\boldsymbol{a}} = \bar{\boldsymbol{\pi}}(s; \bar{\theta})$. For convenience, we define a new reward function

$$\bar{r}^{\bar{\boldsymbol{\pi}}}(s, \bar{\boldsymbol{a}}) = \sum_{a \in \mathcal{A}} \bar{\boldsymbol{a}}[a] r(s, a). \tag{30}$$

Thus, we have $\mathbb{E}_{\bar{\boldsymbol{\pi}}}[r|S(t) = s] = \bar{r}^{\bar{\boldsymbol{\pi}}}(s, \bar{\boldsymbol{a}}) = \mathbb{E}_{\bar{\boldsymbol{\pi}}}[\bar{r}(s, \bar{\boldsymbol{a}})]$. We also define a new state transition probability function

$$\bar{p}\big(s'|s, \bar{\boldsymbol{a}}\big) = \sum_{a \in \mathcal{A}} \bar{\boldsymbol{a}}[a] p\big(s'|s, a\big) \tag{31}$$

with which we can denote the probability at state s' after transitioning for t time steps from state s under policy $\bar{\boldsymbol{\pi}}$ by $\bar{p}^{\bar{\boldsymbol{\pi}}}(s, s', t)$. Obviously, there is $\bar{p}^{\bar{\boldsymbol{\pi}}}(s, s', 1) = \bar{p}(s'|s, \bar{\boldsymbol{\pi}}(s; \bar{\theta}))$, and $\bar{p}^{\bar{\boldsymbol{\pi}}}(s, s', t)$ has the following recurrence relation:

$$\bar{p}^{\bar{\boldsymbol{\pi}}}\big(s, s', t\big) = \int_{\mathcal{S}} \bar{p}^{\bar{\boldsymbol{\pi}}}\big(s, s,'' t-1\big) \bar{p}^{\bar{\boldsymbol{\pi}}}\big(s,'' s', 1\big) \mathrm{d}s.'' \tag{32}$$

In RL, the accumulative discounted reward is defined as *return*

$$G(t) = \sum_{i=0}^{\infty} \gamma^i r(t + i). \tag{33}$$

The state-value function is the expected return starting from s and following $\bar{\boldsymbol{\pi}}$:

$$V^{\bar{\boldsymbol{\pi}}}(s) = \mathbb{E}_{\bar{\boldsymbol{\pi}}}[G(t)|S(t) = s] \tag{34}$$

and the action-value function is

$$Q^{\bar{\boldsymbol{\pi}}}(s, \bar{\boldsymbol{a}}) = \mathbb{E}_{\bar{\boldsymbol{\pi}}}\big[\bar{r}(s, \bar{\boldsymbol{a}}) + \gamma G(t+1)|S(t) = s\big]. \tag{35}$$

Since the objective of the agent is to maximize the expected discounted accumulative reward, the objective function can be formulated as

$$J\big(\bar{\theta}\big) = \int_{\mathcal{S}} p_1(s) V^{\bar{\boldsymbol{\pi}}}(s) \mathrm{d}s \tag{36}$$

where $p_1(s)$ is the initial state distribution.

The gradient of the state-value function with respect to $\bar{\theta}$ can be derived as

$$\nabla_{\bar{\theta}} V^{\bar{\boldsymbol{\pi}}}(s) = \nabla_{\bar{\theta}} \mathbb{E}_{\bar{\boldsymbol{\pi}}}[G(t)|S(t) = s] \tag{37a}$$

$$= \nabla_{\bar{\theta}} \mathbb{E}_{\bar{\boldsymbol{\pi}}}\big[r(t) + \gamma G(t+1)|S(t) = s\big] \tag{37b}$$

$$= \nabla_{\bar{\theta}} \mathbb{E}_{\bar{\boldsymbol{\pi}}}\big[\bar{r}\big(s, \bar{\boldsymbol{\pi}}\big(s; \bar{\theta}\big)\big)\big] + \nabla_{\boldsymbol{\theta}} \mathbb{E}_{\bar{\boldsymbol{\pi}}}\big[\gamma G(t+1)|S(t) = s\big] \tag{37c}$$

[3]The notations in this appendix are a little different from that in text, since the appendix is about single-agent RL.

$$= \nabla_{\bar{\theta}} \mathbb{E}_{\bar{\pi}} \left[\bar{r}\left(s, \bar{\boldsymbol{\pi}}\left(s;\bar{\theta}\right)\right)\right] + \nabla_{\bar{\theta}} \int_{\mathcal{S}} \gamma \bar{p}\left(s'|s, \bar{\boldsymbol{\pi}}\left(s;\bar{\theta}\right)\right) V^{\bar{\boldsymbol{\pi}}}\left(s'\right) \mathrm{d}s' \tag{37d}$$

$$= \nabla_{\bar{\theta}} \bar{\boldsymbol{\pi}}\left(s;\bar{\theta}\right) \nabla_{\bar{\boldsymbol{a}}} \mathbb{E}_{\bar{\pi}} \left[\bar{r}(s, \bar{\boldsymbol{a}})|_{\bar{\boldsymbol{a}}=\bar{\boldsymbol{\pi}}\left(s;\bar{\theta}\right)} \right] + \int_{\mathcal{S}} \gamma \left(\bar{p}\left(s'|s, \bar{\boldsymbol{\pi}}\left(s;\bar{\theta}\right)\right) \nabla_{\bar{\theta}} V^{\bar{\boldsymbol{\pi}}}\left(s'\right) + \nabla_{\bar{\theta}} \bar{\boldsymbol{\pi}}\left(s;\bar{\theta}\right) \nabla_{\bar{\boldsymbol{a}}} \bar{p}\left(s'|s, \bar{\boldsymbol{a}}\right)|_{\bar{\boldsymbol{a}}=\bar{\boldsymbol{\pi}}\left(s;\bar{\theta}\right)} V^{\bar{\boldsymbol{\pi}}}\left(s'\right) \right) \mathrm{d}s' \tag{37e}$$

$$= \nabla_{\bar{\theta}} \bar{\boldsymbol{\pi}}\left(s;\bar{\theta}\right) \nabla_{\bar{\boldsymbol{a}}} (\mathbb{E}_{\bar{\pi}}[\bar{r}(s, \bar{\boldsymbol{a}})] + \int_{\mathcal{S}} \gamma \bar{p}\left(s'|s, \bar{\boldsymbol{a}}\right) V^{\bar{\boldsymbol{\pi}}}\left(s'\right) \Big) \Big|_{\bar{\boldsymbol{a}}=\bar{\boldsymbol{\pi}}\left(s;\bar{\theta}\right)} + \int_{\mathcal{S}} \gamma \bar{p}\left(s'|s, \bar{\boldsymbol{\pi}}\left(s;\bar{\theta}\right)\right) \nabla_{\bar{\theta}} V^{\bar{\boldsymbol{\pi}}}\left(s'\right) \mathrm{d}s' \tag{37f}$$

$$= \nabla_{\bar{\theta}} \bar{\boldsymbol{\pi}}\left(s;\bar{\theta}\right) \nabla_{\bar{\boldsymbol{a}}} (\mathbb{E}_{\bar{\pi}}[\bar{r}(s, \bar{\boldsymbol{a}})] + \mathbb{E}_{\bar{\pi}}\left[\gamma G(t+1)|S(t)=s\right]) \big|_{\bar{\boldsymbol{a}}=\bar{\boldsymbol{\pi}}\left(s;\bar{\theta}\right)} + \int_{\mathcal{S}} \gamma \bar{p}\left(s'|s, \bar{\boldsymbol{\pi}}\left(s;\bar{\theta}\right)\right) \nabla_{\bar{\theta}} V^{\bar{\boldsymbol{\pi}}}\left(s'\right) \mathrm{d}s' \tag{37g}$$

$$= \nabla_{\bar{\theta}} \bar{\boldsymbol{\pi}}\left(s;\bar{\theta}\right) \nabla_{\bar{\boldsymbol{a}}} Q^{\bar{\boldsymbol{\pi}}}(s, \bar{\boldsymbol{a}})\big|_{\bar{\boldsymbol{a}}=\bar{\boldsymbol{\pi}}\left(s;\bar{\theta}\right)} + \int_{\mathcal{S}} \gamma \bar{p}^{\bar{\boldsymbol{\pi}}}\left(s, s', 1\right) \nabla_{\bar{\theta}} V^{\bar{\boldsymbol{\pi}}}\left(s'\right) \mathrm{d}s' \tag{37h}$$

$$= \nabla_{\bar{\theta}} \bar{\boldsymbol{\pi}}\left(s;\bar{\theta}\right) \nabla_{\bar{\boldsymbol{a}}} Q^{\bar{\boldsymbol{\pi}}}(s, \bar{\boldsymbol{a}})\big|_{\bar{\boldsymbol{a}}=\bar{\boldsymbol{\pi}}\left(s;\bar{\theta}\right)} + \int_{\mathcal{S}} \Big[\gamma \bar{p}^{\bar{\boldsymbol{\pi}}}\left(s, s', 1\right) \nabla_{\bar{\theta}} \bar{\boldsymbol{\pi}}\left(s';\bar{\theta}\right) \nabla_{\bar{\boldsymbol{a}}} Q^{\bar{\boldsymbol{\pi}}}\left(s', \bar{\boldsymbol{a}}\right)|_{\bar{\boldsymbol{a}}=\bar{\boldsymbol{\pi}}\left(s';\bar{\theta}\right)} \Big] \mathrm{d}s' + \int_{\mathcal{S}} \gamma \bar{p}^{\bar{\boldsymbol{\pi}}}\left(s, s', 1\right) \int_{\mathcal{S}} \gamma \bar{p}^{\bar{\boldsymbol{\pi}}}\left(s', s'', 1\right) V^{\bar{\boldsymbol{\pi}}}\left(s''\right) \mathrm{d}s'' \mathrm{d}s' \tag{37i}$$

$$= \nabla_{\bar{\theta}} \bar{\boldsymbol{\pi}}\left(s;\bar{\theta}\right) \nabla_{\bar{\boldsymbol{a}}} Q^{\bar{\boldsymbol{\pi}}}(s, \bar{a})\big|_{\bar{\boldsymbol{a}}=\bar{\boldsymbol{\pi}}\left(s;\bar{\theta}\right)} + \int_{\mathcal{S}} \Big[\gamma \bar{p}^{\bar{\boldsymbol{\pi}}}\left(s, s', 1\right) \nabla_{\bar{\theta}} \bar{\boldsymbol{\pi}}\left(s';\bar{\theta}\right) \nabla_{\bar{\boldsymbol{a}}} Q^{\bar{\boldsymbol{\pi}}}\left(s', \bar{\boldsymbol{a}}\right)|_{\bar{\boldsymbol{a}}=\bar{\boldsymbol{\pi}}\left(s';\bar{\theta}\right)} \Big] \mathrm{d}s' + \int_{\mathcal{S}} \gamma^2 \bar{p}^{\bar{\boldsymbol{\pi}}}\left(s, s', 2\right) \nabla_{\theta} V^{\bar{\boldsymbol{\pi}}}\left(s'\right) \mathrm{d}s' \tag{37j}$$

$$\vdots$$

$$= \int_{\mathcal{S}} \sum_{t=0}^{\infty} \Big[\gamma^t \bar{p}^{\bar{\boldsymbol{\pi}}}\left(s, s', t\right) \nabla_{\bar{\theta}} \bar{\boldsymbol{\pi}}\left(s';\bar{\theta}\right) \nabla_{\bar{\boldsymbol{a}}} Q^{\bar{\boldsymbol{\pi}}}\left(s', \bar{\boldsymbol{a}}\right)|_{\bar{\boldsymbol{a}}=\bar{\boldsymbol{\pi}}\left(s';\bar{\theta}\right)} \Big] \mathrm{d}s'. \tag{37k}$$

Here, (37j) is derived via exchanging the order of integrals and using the recurrence relation (32). Then, the gradient of the objective function with respect to $\bar{\theta}$ is

$$\nabla_{\bar{\theta}} J\left(\bar{\theta}\right) = \nabla_{\bar{\theta}} \int_{\mathcal{S}} p_1(s) V^{\bar{\boldsymbol{\pi}}}(s) \mathrm{d}s = \int_{\mathcal{S}} p_1(s) \nabla_{\bar{\theta}} V^{\bar{\boldsymbol{\pi}}}(s) \mathrm{d}s$$

$$= \int_{\mathcal{S}} \int_{\mathcal{S}} \sum_{t=0}^{\infty} \Big[\gamma^t p_1(s) \bar{p}^{\bar{\boldsymbol{\pi}}}\left(s, s', t\right) \nabla_{\bar{\theta}} \bar{\boldsymbol{\pi}}\left(s';\bar{\theta}\right) \nabla_{\bar{\boldsymbol{a}}} Q^{\bar{\boldsymbol{\pi}}}\left(s', \bar{\boldsymbol{a}}\right)|_{\bar{\boldsymbol{a}}=\bar{\boldsymbol{\pi}}\left(s';\bar{\theta}\right)} \Big] \mathrm{d}s' \mathrm{d}s = \int_{\mathcal{S}} \rho^{\bar{\boldsymbol{\pi}}}(s) \nabla_{\bar{\theta}} \bar{\boldsymbol{\pi}}\left(s;\bar{\theta}\right) \nabla_{\bar{\boldsymbol{a}}} Q^{\bar{\boldsymbol{\pi}}}(s, \bar{\boldsymbol{a}})|_{\bar{\boldsymbol{a}}=\bar{\boldsymbol{\pi}}\left(s;\bar{\theta}\right)} \mathrm{d}s. \tag{38}$$

Recall from the definition of $\rho^{\bar{\boldsymbol{\pi}}}$, the last equality of (38) is derived by exchanging the order of integrals.

REFERENCES

[1] S. Chandrasekharan *et al.*, "Designing and implementing future aerial communication networks," *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 26–34, May 2016.

[2] 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.

[3] W. Wang *et al.*, "Joint precoding optimization for secure SWIPT in UAV-aided NOMA networks," *IEEE Trans. Commun.*, vol. 68, no. 8, pp. 5028–5040, Aug. 2020.

[4] F. Cheng *et al.*, "UAV trajectory optimization for data offloading at the edge of multiple cells," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6732–6736, Jul. 2018.

[5] N. Cheng *et al.*, "Space/aerial-assisted computing offloading for IoT applications: A learning-based approach," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 5, pp. 1117–1129, May 2019.

[6] N. Zhao *et al.*, "UAV-Assisted emergency networks in disasters," *IEEE Wireless Commun.*, vol. 26, no. 1, pp. 45–51, Feb. 2019.

[7] M. Alzenad, A. El-Keyi, F. Lagum, and H. Yanikomeroglu, "3-D placement of an unmanned aerial vehicle base station (UAV-BS) for energy-efficient maximal coverage," *IEEE Wireless Commun. Lett.*, vol. 6, no. 4, pp. 434–437, Aug. 2017.

[8] J. Zhao, F. Gao, G. Ding, T. Zhang, W. Jia, and A. Nallanathan, "Integrating communications and control for UAV systems: Opportunities and challenges," *IEEE Access*, vol. 6, pp. 67519–67527, 2018.

[9] J. Zhao, F. Gao, L. Kuang, Q. Wu, and W. Jia, "Channel tracking with flight control system for UAV mmWave MIMO communications," *IEEE Commun. Lett.*, vol. 22, no. 6, pp. 1224–1227, Jun. 2018.

[10] 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.

[11] Y. Zeng, J. Xu, and R. Zhang, "Energy minimization for wireless communication with rotary-wing UAV," *IEEE Trans. Wireless Commun.*, vol. 18, no. 4, pp. 2329–2345, Apr. 2019.

[12] C. Zhan, Y. Zeng, and R. Zhang, "Trajectory design for distributed estimation in UAV-enabled wireless sensor network," *IEEE Trans. Veh. Technol.*, vol. 67, no. 10, pp. 10155–10159, Oct. 2018.

[13] F. Cui, Y. Cai, Z. Qin, M. Zhao, and G. Y. Li, "Multiple access for mobile-UAV enabled networks: Joint trajectory design and resource allocation," *IEEE Trans. Commun.*, vol. 67, no. 7, pp. 4980–4994, Jul. 2019.

[14] S. Jeong, O. Simeone, and J. Kang, "Mobile edge computing via a UAV-mounted cloudlet: Optimization of bit allocation and path planning," *IEEE Trans. Veh. Technol.*, vol. 67, no. 3, pp. 2049–2063, Mar. 2018.

[15] H. Wu, F. Lyu, C. Zhou, J. Chen, L. Wang, and X. Shen, "Optimal UAV caching and trajectory in aerial-assisted vehicular networks: A learning-based approach," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 12, pp. 2783–2797, Dec. 2020.

[16] J. Zhao, Q. Li, Y. Gong, and K. Zhang, "Computation offloading and resource allocation for cloud assisted mobile edge computing in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 7944–7956, Aug. 2019.

[17] W. Shi *et al.*, "Multi-drone 3-D trajectory planning and scheduling in drone-assisted radio access networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 8145–8158, Aug. 2019.

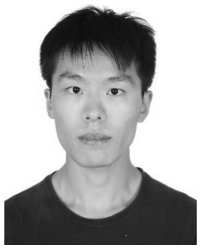
[18] Y. Gu, Y. Jiao, X. Xu, and Q. Yu, "Request-response and censoring-based energy-efficient decentralized change-point detection with IoT applications," *IEEE Internet Things J.*, early access, Oct. 2, 2020, doi: 10.1109/JIOT.2020.3028387.

[19] X. Mao, Y. Gu, and W. Yin, "Walk proximal gradient: An energy-efficient algorithm for consensus optimization," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2048–2060, Apr. 2019.

[20] J. Zhao, X. Guan, and X. Li, "Power allocation based on genetic simulated annealing algorithm in cognitive radio networks," *Chin. J. Electron.*, vol. 22, no. 1, pp. 177–180, Jan. 2013.
[21] X. Shen and Y. Gu, "Nonconvex sparse logistic regression with weakly convex regularization," *IEEE Trans. Signal Process.*, vol. 66, no. 12, pp. 3199–3211, Jun. 2018.
[22] C. Yang, X. Shen, H. Ma, Y. Gu, and H. C. So, "Sparse recovery conditions and performance bounds for ℓ_p-minimization," *IEEE Trans. Signal Process.*, vol. 66, no. 19, pp. 5014–5028, Oct. 2018.
[23] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.
[24] C. H. Liu, Z. Chen, J. Tang, J. Xu, and C. Piao, "Energy-efficient UAV control for effective and fair communication coverage: A deep reinforcement learning approach," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 2059–2070, Sep. 2018.
[25] A. M. Koushik, F. Hu, and S. Kumar, "Deep *Q*-learning-based node positioning for throughput-optimal communications in dynamic UAV swarm network," *IEEE Trans. Cogn. Commun. Netw.*, vol. 5, no. 3, pp. 554–566, Sep. 2019.
[26] J. Cui, Y. Liu, and A. Nallanathan, "Multi-agent reinforcement learning-based resource allocation for UAV networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 2, pp. 729–743, Feb. 2020.
[27] R. Ding, F. Gao, and X. S. Shen, "3D UAV trajectory design and frequency band allocation for energy-efficient and fair communication: A deep reinforcement learning approach," *IEEE Trans. Wireless Commun.*, vol. 19, no. 12, pp. 7796–7809, Dec. 2020.
[28] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT press, 2018.
[29] K. Zhang, Z. Yang, H. Liu, T. Zhang, and T. Başar, "Fully decentralized multi-agent reinforcement learning with networked agents," in *Proc. 35th Int. Conf. Mach. Learn. (ICML)*, Jul. 2018, pp. 5867–5876.
[30] F. Cheng, G. Gui, N. Zhao, Y. Chen, J. Tang, and H. Sari, "UAV-relaying-assisted secure transmission with caching," *IEEE Trans. Commun.*, vol. 67, no. 5, pp. 3140–3153, May 2019.
[31] H. Wang, J. Wang, G. Ding, J. Chen, Y. Li, and Z. Han, "Spectrum sharing planning for full-duplex UAV relaying systems with underlaid D2D communications," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 1986–1999, Sep. 2018.
[32] Y. Cao, L. Zhang, and Y. Liang, "Deep reinforcement learning for multi-user access control in UAV networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Shanghai, China, May 2019, pp. 1–6.
[33] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Proc. 31th Adv. Neural Inf. Process. Syst.*, Dec. 2017, pp. 6382–6393.
[34] N. Andelman, M. Feldman, and Y. Mansour, "Strong price of anarchy," *Games Econ. Behav.*, vol. 65, no. 2, pp. 289–317, 2009.
[35] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Optimal transport theory for cell association in UAV-enabled cellular networks," *IEEE Commun. Lett.*, vol. 21, no. 9, pp. 2053–2056, Sep. 2017.
[36] Y. Shoham and K. Leyton-Brown, *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge, U.K.: Cambridge Univ. Press, 2008.
[37] X. Zhang, K. Zhang, E. Miehling, and T. Başar, "Non-cooperative inverse reinforcement learning," in *Proc. NIPS*, Vancouver, BC, Canada, Dec. 2019, pp. 9482–9493.
[38] Z. Junhui, Y. Tao, G. Yi, W. Jiao, and F. Lei, "Power control algorithm of cognitive radio based on non-cooperative game theory," *Chin. Commun.*, vol. 10, no. 11, pp. 143–154, Nov. 2013.
[39] K. Zhang, Z. Yang, and T. Başar, "Policy optimization provably converges to Nash equilibria in zero-sum linear quadratic games," in *Proc. NIPS*, Vancouver, BC, Canada, Dec. 2019, pp. 11598–11610.
[40] R. K. Jain, D.-M. W. Chiu, and W. R. Hawe, "A quantitative measure of fairness and discrimination," Eastern Res. Lab., Digit. Equip. Corp., Hudson, MA, USA, Rep. DEC-TR-301, 1984.
[41] M. L. Littman, "Markov games as a framework for multi-agent reinforcement learning," in *Proc. 11th Int. Conf. Mach. Learn. (ICML)*, New Brunswick, NJ, USA, Jul. 1994, pp. 157–163.
[42] K. Zhang, Y. Liu, J. Liu, M. Liu, and T. Başar, "Distributed learning of average belief over networks using sequential observations," *Automatica*, vol. 115, May 2020, Art. no. 108857.
[43] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," in *Proc. 4th Int. Conf. Learn. Represent. (ICLR)*, May 2016, p. 10.
[44] V. R. Konda and J. N. Tsitsiklis, "Actor-critic algorithms," in *Proc. 12th Adv. Neural Inf. Process. Syst.*, Denver, CO, USA, Dec. 1999, pp. 1008–1014.
[45] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, San Diego, CA, USA, May 2015, pp. 1–15.
[46] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Proc. 31st Int. Conf. Mach. Learn. (ICML)*, Beijing, China, Jun. 2014, pp. 387–395.

Ruijin Ding (Graduate Student Member, IEEE) received the B.Eng. degree in electrical and information engineering from Dalian University of Technology, Dalian, China, in 2017. He is currently pursuing the Ph.D. degree with the Department of Automation, Tsinghua University, Beijing, China.

His research interests include optimization, UAV communications, and deep reinforcement learning.



Yadong Xu received the B.S. degree in information and computing science from Nanjing University, Nanjing, China, in 2017. He is currently pursuing the Ph.D. degree with the Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China.

His research interests include game theory, mechanism design, and machine learning.

Feifei Gao (Fellow, IEEE) received the B.Eng. degree from Xi'an Jiaotong University, Xi'an, China, in 2002, the M.Sc. degree from McMaster University, Hamilton, ON, Canada, in 2004, and the Ph.D. degree from the National University of Singapore, Singapore, in 2007.

In 2011, he joined the Department of Automation, Tsinghua University, Beijing, China, where he is currently an Associate Professor. He has authored/coauthored more than 150 refereed IEEE journal papers and more than 150 IEEE conference proceeding papers that are cited more than 9900 times in Google Scholar. His research interests include signal processing for communications, array signal processing, convex optimizations, and artificial intelligence-assisted communications.

Prof. Gao has served as an Editor for IEEE TRANSACTIONS ON COMMUNICATIONS, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING (Lead Guest Editor), IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING, IEEE SIGNAL PROCESSING LETTERS (Senior Editor), IEEE COMMUNICATIONS LETTERS (Senior Editor), IEEE WIRELESS COMMUNICATIONS LETTERS, and *China Communications*. He has also served as the Symposium Co-Chair for 2019 IEEE Conference on Communications, 2018 IEEE Vehicular Technology Conference Spring, 2015 IEEE Conference on Communications, 2014 IEEE Global Communications Conference, and 2014 IEEE Vehicular Technology Conference Fall, as well as technical committee members for more than 50 IEEE conferences.

Xuemin (Sherman) Shen (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Rutgers University, New Brunswick, NJ, USA, in 1990.

He is currently a University Professor with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. His research focuses on network resource management, wireless network security, Internet of Things, 5G and beyond, and vehicular *ad hoc* and sensor networks.

Dr. Shen received the R. A. Fessenden Award in 2019 from IEEE, Canada, the Award of Merit from the Federation of Chinese Canadian Professionals (Ontario) in 2019, the James Evans Avant Garde Award in 2018 from the IEEE Vehicular Technology Society, the Joseph LoCicero Award in 2015 and Education Award in 2017 from the IEEE Communications Society, and the Technical Recognition Award from Wireless Communications Technical Committee in 2019 and AHSN Technical Committee in 2013. He has also received the Excellent Graduate Supervision Award in 2006 from the University of Waterloo and the Premier's Research Excellence Award in 2003 from the Province of Ontario, Canada. He served as the Technical Program Committee Chair/Co-Chair for IEEE Globecom'16, IEEE Infocom'14, IEEE VTC'10 Fall, and IEEE Globecom'07, and the Chair for the IEEE Communications Society Technical Committee on Wireless Communications. He was the elected IEEE Communications Society Vice President for Technical and Educational Activities, the Vice President for Publications, the Member-at-Large on the Board of Governors, the Chair of the Distinguished Lecturer Selection Committee, and a member of the IEEE ComSoc Fellow Selection Committee. He was the Editor-in-Chief of the IEEE INTERNET OF THINGS JOURNAL, IEEE NETWORK, and *IET Communications*. He is a registered Professional Engineer of Ontario, Canada, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, a Chinese Academy of Engineering Foreign Member, and a Distinguished Lecturer of the IEEE Vehicular Technology Society and Communications Society.