3D UAV Trajectory Design and Frequency Band Allocation for Energy-Efficient and Fair Communication: A Deep Reinforcement Learning Approach

Ruijin Ding, Feifei Gao, *Fellow, IEEE*, and Xuemin (Sherman) Shen, *Fellow, IEEE*

***Abstract*— Unmanned Aerial Vehicle (UAV)-assisted communication has drawn increasing attention recently. In this paper, we investigate 3D UAV trajectory design and band allocation problem considering both the UAV's energy consumption and the fairness among the ground users (GUs). Specifically, we first formulate the energy consumption model of a quad-rotor UAV as a function of the UAV's 3D movement. Then, based on the fairness and the total throughput, the *fair throughput* is defined and maximized within limited energy. We propose a deep reinforcement learning (DRL)-based algorithm, named as EEFC-TDBA (energy-efficient fair communication through trajectory design and band allocation) that chooses the state-of-the-art DRL algorithm, deep deterministic policy gradient (DDPG), as its basis. EEFC-TDBA allows the UAV to: 1) adjust the flight speed and direction so as to enhance the energy efficiency and reach the destination before the energy is exhausted; and 2) allocate frequency band to achieve fair communication service. Simulation results are provided to demonstrate that EEFC-TDBA outperforms the baseline methods in terms of the fairness, the total throughput, as well as the minimum throughput.**

***Index Terms*— 3D UAV trajectory, band allocation, energy-efficient, fair communication, deep reinforcement learning.**

I. Introduction

UNMANNED Aerial Vehicle (UAV)-assisted communication has attracted increasing attention recently for its providing flexible and cost-effective communication service as well as enhancing coverage under various scenarios [1], [2]. Due to the high mobility, the UAV-assisted communication systems are widely used for emergency communications and broadband connectivity at remote areas [3], [4]. For example, when the terrestrial communication infrastructures are severely damaged by catastrophic natural disasters [5], [6], UAVs are capable of serving as temporary base stations (BSs) and can connect users to the backbone network. Besides, UAVs are likely to provide better communications due to the higher chance of line-of-sight (LoS) links to ground users (GUs), compared to traditional terrestrial communication systems.

Existing researches on UAV-assisted communications can mainly be categorised into two groups: 1) UAVs are used for enlarging the communication coverage [7]–[11] and act as mobile BSs to serve GUs; and 2) UAVs are dispatched as data relay to connect two or more distant users or user groups [12]–[16], and can provide low latency service [17]. In [18], the authors leverage better link qualities from the line of-sight (LoS) links of UAV communication systems to enable low latency communications. In [19], the authors design a real-time data delivery feature for oneM2M standard platform. In [20], the authors design a 3D trajectory planning and scheduling algorithm to minimize the average pathloss. In [21], the authors apply penalty dual-decomposition (PDD) technique to maximize the total throughput of UAV communications with orthogonal multiple access (OMA) and non-orthogonal multiple access (NOMA) modes, which is a nonconvex optimization problem. In [22], Al-Hourani *et al.* present an analytical approach to optimize the altitude of low-altitude aerial platforms (LAPs) in order to provide maximum communication coverage. Lyu *et al.* [23] propose a new cyclical multiple access (CMA) scheme to allocate communication time between UAV and GUs in order to maximize their minimum throughput. Zhang *et al.* [24] find a closed-form low-complexity solution with joint trajectory design and power control to minimize the outage probability of a UAV relay network.

Despite the advantages, such as high mobility, flexible deployment, and low operational costs, UAV-assisted communication systems face the energy constraint challenge. For example, due to the size and weight constraints, the on-board

Manuscript received December 29, 2019; revised March 15, 2020, May 2, 2020, and June 24, 2020; accepted July 27, 2020. Date of publication August 19, 2020; date of current version December 10, 2020. 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, Grant 61771274, and Grant 61531011; and in part by the Beijing Municipal Natural Science Foundation under Grant 4182030 and Grant L182042. This article was presented in part at Globecom, 2020. The associate editor coordinating the review of this article and approving it for publication was C. Huang. *(Corresponding author: Feifei Gao.)*

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

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

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TWC.2020.3016024

energy of UAV is limited, which leads to endurance and performance degradation. As a result, the energy efficiency, defined as the information bits per unit energy consumption, is a key issue in UAV-assisted communications. In [25], the authors derive a mathematical model on the propulsion energy consumption of fixed-wing UAVs and then maximize the energy-efficiency through UAV trajectory design. Zeng *et al.* [26], [27] obtain a closed-form propulsion power consumption model for rotary-wing UAVs, and then optimize UAV trajectory and time allocation among GUs to minimize the total energy consumption during the flight. All these existing problems can be summarized as follows:

- The energy consumption models in [25]–[27] are all restricted to horizontal flight, while none of them consider UAV's 3D trajectory.
- Most works transform the original nonconvex problem into a convex one and then apply the convex optimization toolbox such as CVX [28] to solve the problem. Actually, such approach converts the continuous trajectory design into the fly-hover-communicate design, which simplifies the problem at the cost of accuracy.
- The complexities of these algorithms increase rapidly with the number of users, flight time, and the number of iterations. Besides, the traditional optimization methods cannot deal with users' movement, i.e., they are not capable of selecting the UAV trajectory or allocating communication resources according to the users' current location.

Recently, there have been some researches leveraging deep reinforcement learning (DRL) [29] for UAV-assisted communications. They model the problems as the Markov decision process (MDP) [30], where an agent observes the state of environment, takes an action, and obtains a reward. Then the environment changes into another state. The objective of DRL algorithms is to maximize the expected accumulative reward [31] without the need of transforming nonconvex problem to convex one. DRL can utilize the data process ability of deep neural network (DNN) to deal with the users' movements. In [32], Yin *et al.* optimize UAV trajectory to maximize the throughput without considering energy consumption and resource allocation. In [33], K-means is used to obtain the cell partition of the users, and a Q-learning based algorithm is proposed to select the deployment location. In [34], the authors propose a DRL-based energy-efficient control method for fair communication coverage. However, the algorithm only considers the coverage of each UAV, while the objective is purely to enhance the coverage time of each cell rather than the throughput. Besides, [34] does not apply the resource allocation to further improve the performance.

In this paper, we investigate the problems of energy-efficiency and fair communication service for a quad-rotor UAV-assisted system by jointly designing the 3D trajectory of UAV and the frequency band allocation of moving GUs. We propose a DRL based algorithm, named as EEFC-TDBA, to enable the UAV to provide energy-efficient and fair communication service through trajectory design and frequency band allocation. To cope with the continuous control problem with unlimited state and action space, EEFC-TDBA is designed based on deep deterministic policy gradient (DDPG) [35]. The main contributions of this paper are summarized as follows.

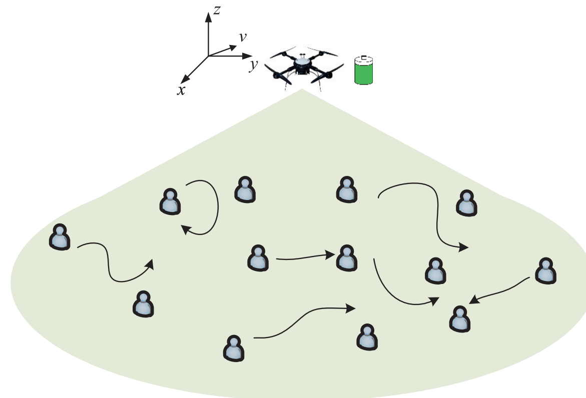


Fig. 1. UAV-assisted wireless communication scenario.

- We derive a mathematical model for the propulsion energy-consumption of a quad-rotor UAV in 3D flight.
- We formulate a UAV energy-efficient and fair communication problem where the UAV chooses the flying direction and the band allocated to each GU, and propose a DDPG based algorithm to solve it. When applying DDPG, there exist dimension imbalance, the gradient vanishing, and the training oscillation issues. We next design dimension spread, pre-activation, softmax reference techniques to address the issues respectively.
- The proposed algorithm takes the UAV's and GUs' location as well as the destination coordinates into consideration, and hence can tackle the GUs' movement issue.
- A fairness utility is applied to trade off the throughput maximization and fairness among the GUs.

The rest of this paper is organized as follows. Section II presents the system model and formulates the fair throughput maximization problem. Section III-A provides a brief review of DRL. In Section III, the EEFC-TDBA algorithm is described in detail, while Section IV presents the performance evaluation. Finally, we conclude the paper in Section V.

Notations: In this paper, scalars are denoted by italic letters, and vectors are denoted by boldface letters. The Euclidean norm of a vector is denoted by $\|\cdot\|$, and $\{\cdot\}$ denotes an array; For a time-dependent function $\boldsymbol{a}(t)$, $\dot{\boldsymbol{a}}(t)$ and $\ddot{\boldsymbol{a}}(t)$ denote the first- and secondorder derivatives with respect to t. $\mathbb{R}^M$ denotes the space of M-dimensional real vectors; $\mathbb{E}_\pi[\cdot]$ denotes expectation of a random variable following policy π.

II. System Model and Problem Formulation

A. System Model

As shown in Fig. 1, we consider a quad-rotor UAV system where a UAV serves as BS for K GUs, $\mathcal{K} = \{1, \cdots, K\}$, and has a total energy E_{max}. Denote $E(t)$ as the remaining energy of UAV at time t and denote T_t as the total flight time of UAV, i.e., $E(0) = E_{max}$ and $E(T_t) = 0$. The GUs move on the ground at constant speeds, and the location of GU k

at time t is denoted by $\boldsymbol{w}_k(t) \in \mathbb{R}^3, 0 \leq t \leq T_t$.[1] Meanwhile, the 3D Cartesian coordinate of the quad-rotor UAV location at time t is denoted by $\boldsymbol{u}(t) = [(x(t), y(t)), z(t)] \in \mathbb{R}^3$, where $(x(t), y(t))$ is the UAV location projected on the horizontal plane, and $z(t)$ is the UAV altitude. The UAV adopts the frequency division multiple access (FDMA) to serve K GUs with a total bandwidth B. Let $B_k(t)$ be the communication bandwidth allocated to GU k at time t. Then there is

$$\sum_{k=1}^{K} B_k(t) = B, \ \forall t \in [0, T_t]. \quad (1)$$

Due to the higher chance of the line-of-sight (LoS) connectivity [9], the air-to-ground (ATG) channel is different from the terrestrial channel and mainly depends on the elevation angle and the type of the propagation environment. To take the occurrence of LoS links into account, we adopt the probabilistic pathloss model in [22], where the probability of LoS connectivity between the UAV and GU k at time t is

$$h_k^{LoS}(t) = \frac{1}{1 + \eta_a \exp\left(-\eta_b \left(\arcsin\left(\frac{z(t)}{d_k(t)}\right) - \eta_a\right)\right)}. \quad (2)$$

Here η_a and η_b are constants related to the type of the propagation environment, and $d_k(t) = \|\boldsymbol{u}(t) - \boldsymbol{w}_k(t)\|$ denotes the time-varying distance from the UAV to GU k. It can be inferred that the corresponding probability of the non-LoS (NLoS) links is $h_k^{NLoS}(t) = 1 - h_k^{LoS}(t)$.

The pathloss expressions of LoS and NLoS links between the UAV to and GU k are

$$L_k^{LoS}(t) = L_k^{FS}(t) + \eta_{LoS}, \quad (3)$$
$$L_k^{NLoS}(t) = L_k^{FS}(t) + \eta_{NLoS}, \quad (4)$$

where $L_k^{FS}(t) = 20 \log d_k(t) + 20 \log f_c + 20 \log\left(\frac{4\pi}{v_c}\right)$ is the free space pathloss, f_c denotes the carrier frequency, and v_c represents the velocity of light. Besides, η_{LoS} and η_{NLoS} are the excessive pathloss [36] for LoS and NLoS links, respectively. Then the pathloss expression between the UAV and GU k becomes

$$\begin{aligned} L_k(t) &= h_k^{LoS}(t) \times L_k^{LoS}(t) + h_k^{NLoS}(t) \times L_k^{NLoS}(t) \\ &= L_k^{FS}(t) + h_k^{LoS}(t)\eta_{LoS} + \left(1 - h_k^{LoS}(t)\right)\eta_{NLoS}. \end{aligned} \quad (5)$$

The transmission rate between the UAV and GU k can be expressed as

$$R_k(t) = B_k(t) \log_2\left(1 + \frac{P_c}{\beta_k(t) n_0 B_k(t)}\right), \quad (6)$$

where P_c is the communication power for each GU k, n_0 is the noise power spectral density, and $\beta_k(t) = 10^{L_k(t)/10}$.[2] The power allocated to each GU is equal. The overall capacity of the UAV is

$$R_c(t) = \sum_{k=1}^{K} R_k(t). \quad (7)$$

Note that given the trajectories $\{\boldsymbol{w}_k(t)\}_{k \in \mathcal{K}}$ of all GUs, $R_c(t)$ is a function of the UAV location $\{\boldsymbol{u}(t)\}$ and the frequency band allocation $\{B_k(t)\}$. Thus, the total UAV throughput before time t is

$$\bar{R}(t) = \bar{R}(\{\boldsymbol{u}(t)\}, \{B_k(t)\}) = \int_0^t R_c(\tau) d\tau, \quad (8)$$

and the throughput for each GU is

$$\bar{R}_k(t) = \bar{R}_k(\{\boldsymbol{u}(t)\}, \{B_k(t)\}) = \int_0^t R_k(\tau) d\tau. \quad (9)$$

B. Energy Consumption Model for Quad-Rotor UAV

The energy consumption of the UAV consists of two parts: one is for the communication, and the other is the propulsion energy for generating thrusts[3] to help the UAV overcome the drag[4] and gravity. In practice, the energy for communication is often much smaller than that for flight by two orders of magnitude [25]. Therefore, the energy consumption of communication is ignored in this paper.

For better exposition, we only consider the acceleration that is in a straight line with velocity while omit the acceleration component that is perpendicular to velocity. Thus the UAV can change direction instantaneously without extra energy consumption, which is reasonable because the quad-rotor UAV can easily steer by adjusting the rotation rate of four rotors. As derived in Appendix A, the thrust of each rotor is a function of UAV velocity $\boldsymbol{v}$ and acceleration $\boldsymbol{a}_c$. The velocity direction vector is denoted by $\boldsymbol{v}_d$, and $v = \|\boldsymbol{v}\|$ is the UAV speed. Then the thrust of each rotor can be expressed as

$$T_h(\boldsymbol{v}, \boldsymbol{a}_c) = \frac{1}{n_r} \left\| (m\|\boldsymbol{a}_c\| + \frac{1}{2}\rho v^2 S_{FP})\boldsymbol{v}_d - m\boldsymbol{g} \right\|, \quad (10)$$

where n_r is the number of rotors; m denotes the UAV mass; ρ and S_{FP} are the air density and fuselage equivalent flat plate area; $\boldsymbol{g}$ is the gravity acceleration vector. Then the propulsion power can be expressed as (11), shown at the bottom of the page, where δ is the local blade section drag coefficient; c_T denotes the thrust coefficient based on disc area; A and c_s are the disc area for each rotor and rotor solidity respectively; c_f is the incremental correction factor of induced power; τ_c

[1] $\boldsymbol{w}_k(t)$ is the 3D Cartesian coordinates of the GUs with altitudes of 0.

[2] The above pathloss expressions are all in dB.

[3] The thrusts are the forces produced by the rotors and move the UAV forward.

[4] The drag is the aerodynamic force component that is in the opposite direction of the UAV's motion.

$$P(v, T_h) = n_r \left[\frac{\delta}{8}\left(\frac{T_h}{c_T \rho A} + 3v^2\right)\sqrt{\frac{T_h \rho c_s^2 A}{c_T}} + (1 + c_f) T_h \left(\sqrt{\frac{T_h^2}{4\rho^2 A^2} + \frac{v^4}{4}} - \frac{v^2}{2}\right)^{\frac{1}{2}} + \frac{mgv}{n_r} \sin \tau_c + \frac{1}{2} d_0 v^3 \rho c_s A\right] \quad (11)$$

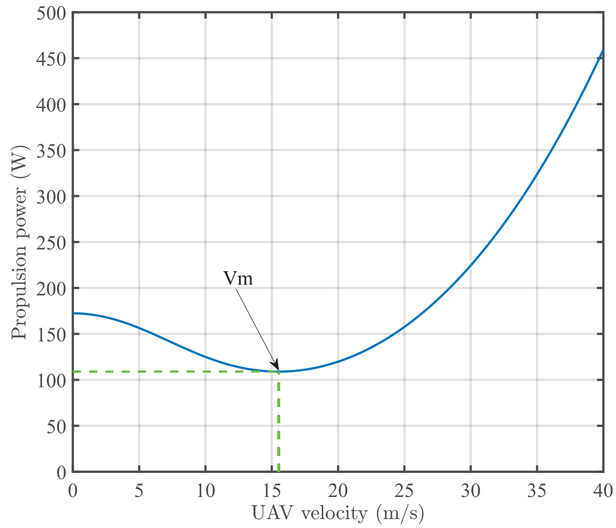


Fig. 2. The propulsion power versus the UAV velocity.

denotes the climb angle; d_0 is the fuselage drag ratio for each rotor. The detailed introduction of the parameters is in listed in Appendix A.

Remark 1: Considering $\|\boldsymbol{a}_c\| = 0$ and the UAV flies in the horizontal plane, i.e., $\tau_c = 0$, we plot the propulsion power versus the UAV velocity in Fig. 2. It is seen that the propulsion power consumption does not increase monotonously with the UAV velocity. When the UAV flies at about 15 m/s, the power consumption achieves the minimum. Therefore, the UAV would be able to fly for the longest time with the fixed energy, which may lead to the maximum total throughput.

Considering $\boldsymbol{v}(t) \triangleq \dot{\boldsymbol{u}}(t)$ and $\boldsymbol{a}_c(t) \triangleq \ddot{\boldsymbol{u}}(t)$, the propulsion power is essentially a function of the UAV trajectory $\{\boldsymbol{u}(t)\}$. Then with a given trajectory $\{\boldsymbol{u}(t)\}$, the remaining energy of the UAV can be expressed as

$$E(t) = E_{max} - \int_0^t P\left(\dot{\boldsymbol{u}}(\tau), T_h(\dot{\boldsymbol{u}}(\tau), \ddot{\boldsymbol{u}}(\tau))\right) d\tau. \quad (12)$$

C. Problem Formulation

The UAV energy efficiency of the UAV-assisted communication system can be defined as $\bar{R}(T_t)/E_{max}$. Since the UAV battery capacity E_{max} is a constant, maximizing the energy efficiency is equivalent to maximizing the total throughput before the UAV battery runs out.

However, maximizing the total throughput may lead to unfairness problem, where the UAV may tend to hover close to some GUs, while the other GUs suffer from small throughput all the time. To tackle this issue, we refer to [34] and define the throughput ratio of GU k as:

$$f_k(t) = \frac{\bar{R}_k(t)}{\bar{R}(t)}. \quad (13)$$

Then we apply Jain's fairness index [37] to measure the fairness among GUs:

$$\hat{f}(t) = \frac{(\sum_{k=1}^K f_k(t))^2}{K(\sum_{k=1}^K f_k(t)^2)}. \quad (14)$$

Obviously, $\hat{f}(t) \in [0,1]$ holds. The smaller the differences among the throughput ratios $\{f_k\}_{k\in k}$ are, the greater fair index $\hat{f}$ is. As a result, larger fair index means fairer communication service. Note that both the fair index and the total throughput of UAV are the functions of the UAV trajectory $\{\boldsymbol{u}(t)\}_{t\in[0,T_t]}$ and the frequency band allocation $\{B_k(t)\}_{t\in[0,T_t]}$. Then we may define the total *fair throughput* during the whole mission as

$$\bar{R}_f(\{\boldsymbol{u}(t)\}, \{B_k(t)\}) = \int_0^{T_t} \hat{f}(\tau) R_c(\tau) d\tau. \quad (15)$$

In order to balance the total throughput and fairness, our objective is to maximize the fair throughput by designing the UAV trajectory $\{\boldsymbol{u}(t)\}_{t\in[0,T_t]}$ and allocating the frequency band $\{B_k(t)\}_{t\in[0,T_t]}$. The problem can be mathematically formulated as

$$\begin{aligned}
(P1): \quad & \max_{\{\boldsymbol{u}(t)\},\{B_k(t)\}} \bar{R}_f(\{\boldsymbol{u}(t)\}, \{B_k(t)\}) \\
s.t. \quad & E(0) = E_{max}, \ E(T_t) = 0, && (16)\\
& \boldsymbol{u}(0) = \boldsymbol{u}_0, \ \boldsymbol{u}(T_t) = \boldsymbol{u}_c, && (17)\\
& v(0) = 0, && (18)\\
& B_k(t) \geq B_{min}, \ \forall t \in [0, T_t], && (19)\\
& \sum_{k=1}^K B_k(t) = B(t), \ \forall t \in [0, T_t], && (20)\\
& 0 \leq v(t) \leq v_{max}, \ \forall t \in [0, T_t], && (21)\\
& 0 \leq \|a_c(t)\| \leq a_{max}, \ \forall t \in [0, T_t], && (22)\\
& z_{min} \leq z(t) \leq z_{max}, \ \forall t \in [0, T_t], && (23)
\end{aligned}$$

where $\boldsymbol{u}_0, \boldsymbol{u}_c \in \mathbb{R}^3$ are the initial and final locations of the UAV, B_{min} is the minimum bandwidth to guarantee the basic communication service for each GU, and z_{min} and z_{max} are the altitude constraints. At $\boldsymbol{u}_0$, the UAV takes off with $v(0) = 0$ and needs to reach $\boldsymbol{u}_c$ before it runs out of energy so as to be recharged for the next mission. Note that, problem $(P1)$ is difficult to solve because the optimization parameters like the UAV trajectory $\{\boldsymbol{u}(t)\}_{t\in[0,T_t]}$ and the frequency band allocation $\{B_k(t)\}_{t\in[0,T_t]}$ are continuous time series.

To make problem $(P1)$ trackable, we divide the time into multiple slots with duration δ_t. Then the UAV trajectory $\boldsymbol{u}(t)$ can be characterized by discrete-time UAV location $\boldsymbol{u}[n] = \boldsymbol{u}(n\delta_t)$, $n = 0, 1, \ldots, N_t$, where $N_t = \lceil T_t/\delta_t \rceil$. Similarly, we have $\hat{f}[n] = \hat{f}(n\delta_t)$, $\boldsymbol{v}[n] = \boldsymbol{v}(n\delta_t)$, $\boldsymbol{a}_c[n] = \boldsymbol{a}_c(n\delta_t)$ and $B_k[n] = B_k(n\delta_t)$. The objective (15) can be rewritten as

$$\bar{R}_f(\{\boldsymbol{u}[n]\}, \{B_k[n]\}) = \sum_{n=0}^{N_t} \hat{f}(n) R_c(n) \delta_t. \quad (24)$$

During each time slot, we assume that the acceleration remains constant and the flight direction is fixed. Thus we have

$$v[n+1] = v[n] + a_c[n]\delta_t, \tag{25}$$

$$\boldsymbol{u}[n+1] = \boldsymbol{u}[n] + (v[n]\delta_t + \frac{1}{2}a_c[n]\delta_t^2)\boldsymbol{v}_d[n], \tag{26}$$

where $a_c = \xi_{a_c}\|\boldsymbol{a}_c\|$, and ξ_{a_c} is an indicator that if $\xi_{a_c} = 1$, then the direction of acceleration is the same as that of velocity; otherwise, if $\xi_{a_c} = -1$, then the acceleration and velocity are in opposite directions. Moreover, the remaining energy (12) in time-slotted fashion is re-written as

$$E(n) = E_{max} - \sum_{i=0}^{n-1} P(v(i), T_h(i))\delta_t. \tag{27}$$

According to (25) and (26), $\{\boldsymbol{u}[n]\}$, $\{\boldsymbol{v}[n]\}$ and $\{\boldsymbol{a}_c[n]\}$ are linear with each other. Thus we could optimize the UAV velocity $\{\boldsymbol{v}[n]\}$ or the UAV acceleration $\{\boldsymbol{a}_c[n]\}$ instead of directly optimizing the UAV trajectory $\{\boldsymbol{u}[n]\}$. As a result, the problem $(P1)$ can be re-expressed as

$$
\begin{aligned}
(P2): \quad & \max_{\{\boldsymbol{v}[n]\},\{B_k[n]\}} \quad \bar{R}_f(\{\boldsymbol{u}[n]\},\{B_k[n]\}) \\
s.t. \quad & E[0] = E_{max},\ E[N_t] = 0, && (28)\\
& \boldsymbol{u}[0] = \boldsymbol{u}_0,\ \boldsymbol{u}[N_t] = \boldsymbol{u}_c && (29)\\
& v[0] = 0 && (30)\\
& B_k(n) \geq B_{min},\ n = 0,1,\ldots,N_t && (31)\\
& \sum_{k=1}^{K} B_k[n] = B,\ n = 0,1,\ldots,N_t && (32)\\
& 0 \leq v[n] \leq v_{max},\ n = 0,1,\ldots,N_t && (33)\\
& -a_{max} \leq a_c[n] \leq a_{max},\ n = 0,1,\ldots,N_t && (34)\\
& z_{min} \leq z[n] \leq z_{max},\ n = 0,1,\ldots,N_t. && (35)
\end{aligned}
$$

Problem $(P2)$ is a non-convex optimization problem with thousands of optimization variables since they are time-varying. In fact, without considering the complex energy consumption model (11) and frequency band allocation, problem $(P2)$ reduces to a travelling salesman problem (TSP) [38], which is known to be NP-hard. Hence, problem $(P2)$ is too difficult to be solved by traditional optimization methods. Fortunately, DRL can search the solution from a large policy space and can capture the feature of the energy consumption model with the powerful data processing capability.

III. DRL Based UAV Trajectory Design and Frequency Band Allocation

In this section, we present the EEFC-TDBA algorithm for 3D UAV trajectory design and communication frequency band allocation. In EEFC-TDBA, the UAV is treated as the agent. At each time slot, the UAV observes the state $s(i)$, inputs it to the network, and outputs the action $a(i)$. Then the UAV receives reward $r(i)$, and the state turns into $s(i+1)$. The corresponding experience $(s(i), a(i), r(i), s(i+1))$ is stored in a replay buffer for the training of the network.

The DRL approach has two phases: training phase and implementation phase. In the training phase, the DNN is trained offline, and the exploration is needed to search the optimal policy. While in the implementation phase, DNN only forwards propagation, which consumes much less resources than training. Besides, there is no need for exploration in implementation phase.

A. Preliminaries

We first provide a brief introduction for DDPG that is a DRL algorithm within actor-critic framework [39]. In DDPG, the critic $Q(s, a; \theta^Q)$ evaluates the action-value function under the actor policy $\pi(s; \theta^\pi)$, where θ^π and θ^Q refer to the parameters of actor and critic networks.

However, a non-linear function approximator, e.g., DNN, is known to be unstable and even cause divergence when applied in DRL. Two techniques are usually used in DRL to resolve this issue: experience replay and target network [40]. DRL samples a mini-batch of experiences from the replay buffer that stores state transition samples collected during learning. The random samples break the correlation between sequential samples and stabilize the training process. Moreover, the target networks of actor and critic $\pi'(s; \theta^{\pi'})$ and $Q'(s, a; \theta^{Q'})$ are used to compute the update target and have the same architectures as the learned networks $Q(s, a; \theta^Q)$ and $\pi(s; \theta^\pi)$. Specifically, the critic network can be trained by minimizing the loss:

$$L(\theta^Q) = \frac{1}{N_b}\sum_i \left[y_t(i) - Q(s(i), a(i); \theta^Q)\right]^2, \tag{36}$$

where

$$y_t(i) = r(i) + \gamma Q'\left(s(i+1), \pi'\left(s(i+1); \theta^{\pi'}\right); \theta^{Q'}\right) \tag{37}$$

is the update target, and N_b is the batch size. In addition, the actor is trained by minimizing the actor loss

$$L(\theta^\pi) = \frac{1}{N_b}\sum_i -Q\left(s(i), \pi(s(i); \theta^\pi); \theta^Q\right). \tag{38}$$

The parameters of the target networks are updated by slowly tracking the learned networks: $\theta' \leftarrow \varepsilon\theta + (1-\varepsilon)\theta'$ with $\varepsilon \ll 1$ [35].

B. State Space

As mentioned in Section II-A, the throughput is related to the pathloss between the UAV and GUs. However, compared with the pathloss, the locations of the UAV and GUs can be obtained more easily, since most cell phones are equipped with GPS sensors. Besides, the locations of GUs change all the time. As a result, the state includes the GUs' locations $\{\boldsymbol{w}_k(n)\}_{k\in\mathcal{K}}$ and the UAV location $\boldsymbol{u}(n)$ such that UAV can deal with movement of GUs. According to Section II-C, the UAV needs to reach the destination $\boldsymbol{u}_c$ before its energy runs out. Thus the target spot $\boldsymbol{u}_c$ and the current energy $E(n)$ should be taken into consideration in the state. Moreover, the state includes the current speed of the UAV $v(n)$, which aims to remind the UAV not to exceed the acceleration constraint (34).

In summary, the state can be formulated as

$$s(n) = \{\{\boldsymbol{w}_k(n)\}_{k\in\mathcal{K}}, \boldsymbol{u}(n), v(n), \boldsymbol{u}_c, E(n)\}, \tag{39}$$

and has $(3K+8)$ dimensions.

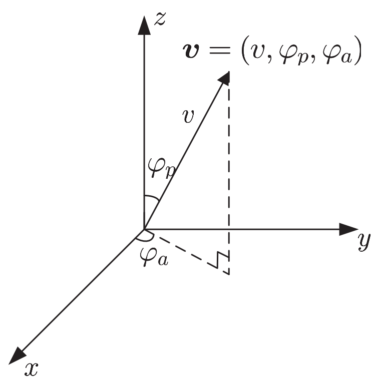


Fig. 3. The spherical coordinate.

C. Action Space

The action $a(n)$ of the UAV trajectory design and frequency band allocation consists of two parts:

- $\boldsymbol{v}(n+1)$: the UAV velocity at the next time slot with $\|\boldsymbol{v}(n+1)\| \in [0, v_{max}]$;
- $\{B_k(n)\}_{k\in\mathcal{K}}$: the frequency band allocation.

1) UAV Velocity: As shown in Fig. 3, we use the spherical coordinate $\{v, \varphi_p, \varphi_a\}$ in order to describe the UAV speed and flight direction more conveniently, where φ_p is polar angle from the positive z-axis with $0 \le \varphi_p \le \pi$, and φ_a is known to be the azimuthal angle in the xy-plane from the x-axis with $-\pi \le \varphi_a \le \pi$.

For convenience, we apply the normalized representation for the UAV velocity:

$$v = \lambda_v \cdot v_{max}, \tag{40}$$

$$\varphi_p = \lambda_{\varphi_p} \cdot \pi, \tag{41}$$

$$\varphi_a = \lambda_{\varphi_a} \cdot \pi, \tag{42}$$

where $\lambda_v, \lambda_{\varphi_p} \in [0, 1]$ and $\lambda_{\varphi_a} \in [-1, 1]$.

2) Frequency Band Allocation: The frequency band allocation strategy should satisfy the constraints (31) and (32). The actions for frequency band allocation $\{B_k(n)\}_{k\in\mathcal{K}}$ can be re-expressed as the ratios of bandwidth allocated to GU k to the total bandwidth $\{\lambda_k^b(n)\}_{k\in\mathcal{K}}$, i.e.,

$$B_k(n) = \lambda_k^b(n) B, \tag{43}$$

where $\sum_{k=0}^{K} \lambda_k^b(n) = 1$ and $\lambda_k^b(n) \ge \frac{B_{min}}{B}$.

In summary, the action $a(n)$ has $(3 + K)$ dimensions, in which "3" refers to the UAV velocity related action, and "K" refers to the frequency band allocation part for K GUs.

D. Reward Design

In DRL, the reward signal is used to evaluate how good an action is under a state, and we can transform hard-to-optimize objectives into maximizing the accumulative reward through reward design. Our objectives are twofold: maximizing the total fair throughput within limited energy and letting the UAV reach the destination before the energy is exhausted.

1) Fair Throughput Maximization: First, the reward at time slot n should include the fair throughput defined in $(P2)$:

$$r_{th}(n) = \kappa_{th} \hat{f}(n) R_c(n) \delta_t, \tag{44}$$

where κ_{th} is a positive constant that is used to adjust the reward of fair throughput maximization part.

2) Reach-Destination Task: For the reach-destination task, the straightforward rewards are very sparse, which means the UAV would be rewarded if and only if it reaches the destination when the battery runs out. Since the UAV's battery could support its flying for thousands of time slots, the agent can receive just one signal indicating whether the UAV reaches the destination every thousands of time slots. In addition, the initial policy is randomly generated, and the UAV would arrive at the destination with probability almost zero. Accordingly, the agent could hardly learn the reach-destination task through straightforward rewards.

In order to deal with the sparse reward issue, we apply *reward shaping* [41] to make the rewards more trackable. The reward of reach-destination part is designed as

$$r_{rd}(n) = \frac{d_{dis}}{\lfloor E(n)/\zeta_{rd} \rfloor \kappa_{rd} + \epsilon_{rd}} \tag{45}$$

where d_{dis} denotes the reduced distance between the UAV's current location and the destination after one-time-slot movement, κ_{rd} is the positive constant coefficient like κ_{th}, ζ_{rd} denotes the energy step size and ϵ_{rd} is the value preventing the denominator from being zero. When the remaining energy $E(n)$ is abundant, the reward of reach-destination part will be quite small and the agent focuses on maximizing the fair throughput. With more energy consumption, the agent focuses more on the reach-destination task.

Moreover, when the energy is exhausted, we should set a reward r_{ar} indicating whether the UAV arrives at the target spot, i.e.,

$$r_{ar}(n) = \begin{cases} 0, & n = 1, \cdots, N_t - 1 \\ \xi_{ar}\kappa_{ar} + (1 - \xi_{ar})\kappa_{nar}, & n = N_t \end{cases}, \tag{46}$$

where $\xi_{ar} = 1$ if the UAV arrives at the destination when the battery runs out, while $\xi_{ar} = 0$, otherwise. Furthermore, κ_{ar} is a positive constant to encourage arrival, while κ_{nar} is a negative constant to punish nonarrival.

3) Constraints: We set penalty rewards to punish the actions that violate the constraints (34) and (35):

$$r_{ac}(n) = \xi_{ac\_v}(n) \cdot \kappa_{ac}, \tag{47}$$

$$r_{al}(n) = \xi_{al\_v}(n) \cdot \kappa_{al}, \tag{48}$$

where $\xi_{ac\_v}(n)$ is the binary acceleration constraint indicator with $\xi_{ac\_v}(n) = 1$ implying the acceleration violates the constraint (34) and $\xi_{ac\_v}(n) = 0$ otherwise. Similarly, $\xi_{al\_v}(n)$ is the binary altitude constraint indicator. The two negative constants κ_{ac} and κ_{al} are specific penalty rewards for the violation of constraints (34) and (35) respectively.

In summary, the shaped reward can be formulated as

$$r(n) = r_{th}(n) + r_{rd}(n) + r_{ar}(n) + r_{ac}(n) + r_{al}(n). \tag{49}$$

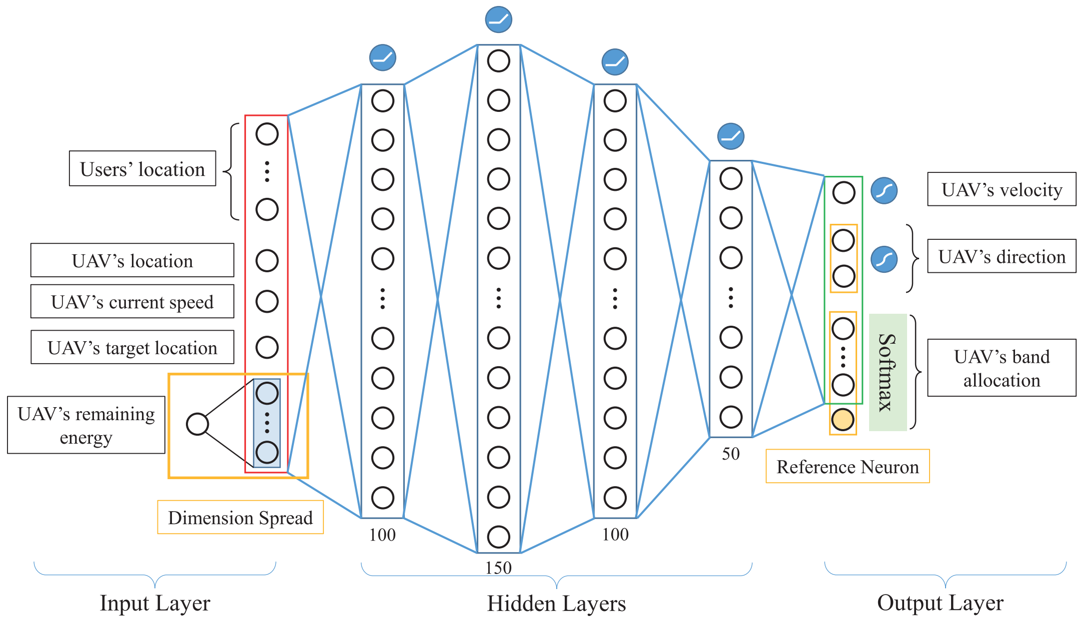


Fig. 4. The actor network architecture.

E. Actor Network

The actor network takes the state $s(n)$ as input, and outputs the action $a(n)$. As shown in Fig. 4, we use the activation function *sigmoid* to output λ_v and λ_{φ_p}, *tanh* to output λ_{φ_a}, and *softmax* to output $\{\lambda_k^b\}$, respectively, i.e.,

$$\lambda_v = \mathrm{sigmoid}(\chi_v), \tag{50}$$

$$\lambda_{\varphi_p} = \mathrm{sigmoid}(\chi_{\varphi_p}), \tag{51}$$

$$\lambda_{\varphi_a} = \tanh(\chi_{\varphi_a}), \tag{52}$$

$$\{\lambda_k^b\} = (1 - \frac{KB_{min}}{B})\mathrm{softmax}(\{\chi_k^b\}) + \frac{B_{min}}{B}, \tag{53}$$

where χ_v, χ_{φ_p}, χ_{φ_a}, and $\{\chi_k^b\}$ are the pre-activation values of the corresponding neurons. There appear two specific problems: dimension imbalance and saturation. We propose *dimension spread* and *pre-activation penalty* to deal with these two problems respectively. Besides, we use a *softmax reference* technique to stabilize the training process.

1) Dimension Spread: According to Section III-B, most of the state dimensions are about location information, while only one dimension is about energy. Nevertheless, the energy dimension is very important since the UAV must reach the destination when the energy is exhausted. Thus there exists the dimension imbalance problem, and we spread the energy dimension to make it comparable to the location dimensions. As shown in Fig. 4, the energy dimension first connects to a spread network with size $1 \times N_e$ to extend the dimension to N_e, and then combines the other dimensions, e.g., uses' locations, UAV's location, UAV's speed and UAV's target location, to formulate the input of the actor network.

2) Pre-Activation Penalty: The activation functions sigmoid and tanh suffer from saturation problem. As shown in Fig. 5, when the absolute pre-activation value is greater than the tanh saturation value ζ_t, i.e., in the saturation area, the output of activation function is approximately 1, and thus the back

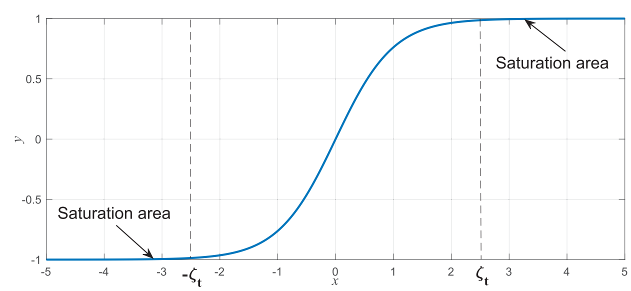


Fig. 5. The tanh function.

propagation process would face gradient vanishing problem. In order to avoid the saturation, we add the pre-activation penalty to the actor loss function, and (38) becomes

$$\begin{aligned} L(\theta^\pi) = &-Q\left(s(i), \pi(s;\theta^\pi);\theta^Q\right) \\ &+\kappa_v\Big(\max(\chi_v - \zeta_s, 0) + \max(-\chi_v - \zeta_s, 0)\Big)^2 \\ &+\kappa_{\varphi_p}\Big(\max(\chi_{\varphi_p} - \zeta_s, 0) + \max(-\chi_{\varphi_p} - \zeta_s, 0)\Big)^2 \\ &+\kappa_{\varphi_a}\Big(\max(\chi_{\varphi_a} - \zeta_t, 0) + \max(-\chi_{\varphi_a} - \zeta_t, 0)\Big)^2, \end{aligned} \tag{54}$$

where κ_v, κ_{φ_a} and κ_{φ_a} are the coefficients of the the pre-activation penalties for χ_v, χ_{φ_p} and χ_{φ_a}, and ζ_s is the sigmoid saturation value. The large pre-activation value of the neurons would cause large actor loss. In other words, minimizing the actor loss would let the pre-activation value stay in the unsaturation area.

3) Softmax Reference: The softmax activation function normalizes an input vector of K real numbers into a probability distribution consisting of K probabilities proportional to the exponentials of the input numbers. Note that it only makes

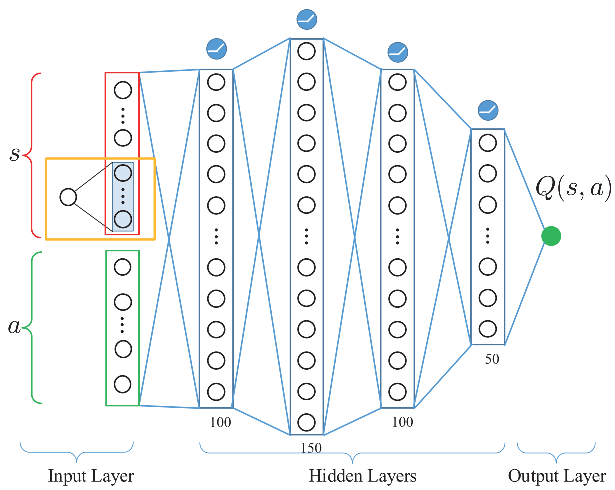


Fig. 6. The critic network architecture.

sense when there are differences among these K real numbers. For example, the output will be the same no matter the input is an all-zeros or all-ones vector, which may lead to oscillation and divergence since the larger pre-activation value may not lead to the larger output. We set a pre-activation neuron to a fixed value as the reference of the input in order to stabilize the training process. Specifically, the actor network outputs $(K-1)$ neurons for the frequency band allocation and concatenates the neurons with fixed value 0 to the previous $(K-1)$ neurons. Next, the softmax function activates the neurons, and we get the frequency band allocation action $\{\lambda_k^b\}_{k\in\mathcal{K}}$.

F. Critic Network

As shown in Fig. 6, the critic network takes the state $s(n)$ and action $a(n)$ as input, and outputs the action-value $Q(s(n), a(n))$. The hidden layer architecture of critic network is the same as that of actor network. In addition, the dimension spread technique is retained in the critic network.

G. Training Algorithm

The EEFC-TDBA algorithm is episodic with each episode starting from the departure spot and ending up when the battery runs out.

In the training phase, the UAV's departure and target locations as well as the GUs' initial locations are initialized at the beginning of each episode. The UAV is stationary at first with fully-charged battery of energy E_{max}. The GUs move around on the ground, with their locations changing over time.

At each time slot, the UAV chooses action $a(n)$ through the actor network $\pi(s(n);\theta^\pi)$ and then adds an exploration noise $\mathcal{N}$ that is used to prevent the agent from falling into local optimal policy. We select the normal distribution noise with zero mean and deviation $\sigma_{\mathcal{N}}$. We also need to deal with violations of the altitude and acceleration constraints. If the action causes the violation of the altitude limit, then the altitude would be readjusted to the corresponding altitude limit boundary, and the agent receives a penalty reward, which has been expounded in Section III-D. Similarly, if the action causes the violation

Algorithm 1 EEFC-TDBA

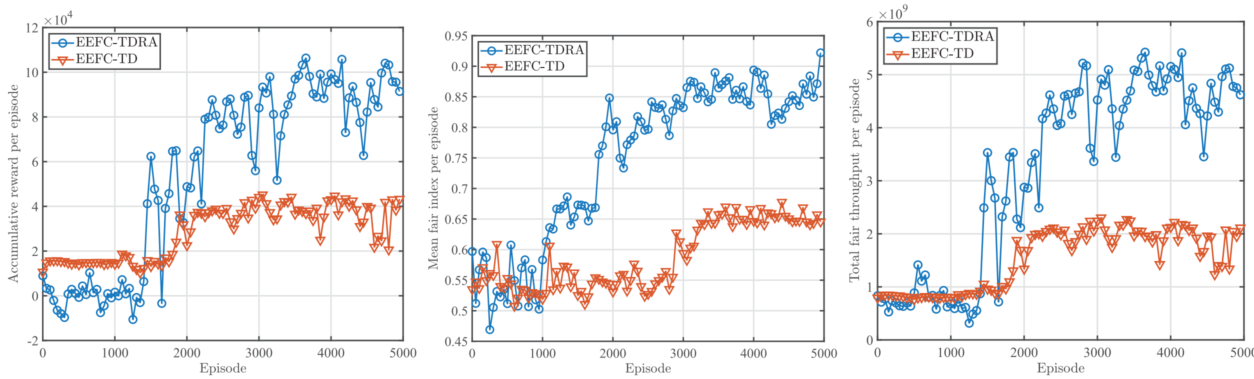
1: Initialize the networks, including actor network $\pi(s;\theta^\pi)$, the target actor network with weights $\theta^{\pi'} = \theta^\pi$, the critic network $Q(s,a;\theta^Q)$ and the target critic network with weights $\theta^{Q'} = \theta^Q$
2: Initialize the experience replay buffer $\mathcal{C}$
3: **for** each episode **do**
4:   Initialize the locations of the UAV and the GUs, as well as the destination.
5:   The UAV's initial speed is zero, and the battery energy is E_{max}
6:   **for** each time slot n **do**
7:     The UAV gets the GUs' locations and formulate the state $s(n)$
8:     $a(n) = \pi(s(n);\theta^\pi) + \mathcal{N}$, where $\mathcal{N}$ is exploration noise
9:     The UAV takes the action $a(n)$
10:     **if** the action violates the altitude constraint **then**
11:       Readjust the altitude to the corresponding altitude limit boundary
12:     **end if**
13:     **if** the action violates the acceleration constraint **then**
14:       $v(n+1) = v(n) + \xi_{ac} a_{max}$
15:     **end if**
16:     Update the state $s(n+1)$ and obtain the reward $r(n)$
17:     Store $\{s(n), a(n), r(n), s(n+1)\}$ in the experience replay buffer $\mathcal{C}$
18:   **end for**
19:   Sample several random minibatches of N_b transitions from $\mathcal{C}$
20:   Calculate the critic target according to (37)
21:   Update the critic network θ^Q by minimizing the critic loss (36)
22:   Update the actor network θ^π by minimizing the actor loss (54)
23:   Soft updates for the target networks:
24:   $\theta^{Q'} = \varepsilon\theta^Q + (1-\varepsilon)\theta^{Q'}$
25:   $\theta^{\pi'} = \varepsilon\theta^\pi + (1-\varepsilon)\theta^{\pi'}$
26: **end for**

of acceleration constraint, then the UAV would accelerate or decelerate with acceleration magnitude a_{max}, which depends on the sign of the original acceleration. Then the agent obtains the next state $s(n+1)$ and the reward $r(n)$, and stores the corresponding transition tuple $(s(n), a(n), s(n+1), r(n))$ in the experience replay buffer $\mathcal{C}$. At each episode, we uniformly sample batches of experience from the replay buffer and update the networks through minimizing the actor and critic loss. Then the target networks are slowly updated according to Line 24 and Line 25 in Algorithm 1.

In the implementation phase, the UAV chooses the flight and frequency band allocation strategies through the well-trained actor network according to the current state.

IV. Performance Evaluation

In this section, simulations are conducted to evaluate the performance of the proposed EEFC-TDBA.



(a) Accumulative reward per episode. (b) Mean fair index per episode. (c) Total fair throughput per episode.

Fig. 7. The training process for EEFC-TDBA and EEFC-TD.

TABLE I
REWARD SETTING

Reward Parameters	Simulation value
κ_{th}	10^{-5}
κ_{rd}	0.5
ζ_{rd}	7.5×10^3
ϵ_{rd}	0.1
κ_{ar}	1.0×10^4
κ_{nar}	-1.0×10^3
κ_{ac}	-5.0
κ_{al}	-5.0

A. Simulation Settings

In our simulation, there is no limit on the horizontal coordinates of the UAV trajectory, while the upper limit of the altitude is set as $z_{max} = 300$ m and the lower limit is $z_{min} = 100$ m. The departure location of UAV is set as $\boldsymbol{u}_0 = (-500, 0, 100)$ and the destination is $\boldsymbol{u}_c = (500, 0, 100)$. The fully charged energy of the on-board battery is $E_{max} = 1 \times 10^5$ Joule. The maximum flying speed is $V_{max} = 20$ m/s, and the maximum acceleration magnitude is $a_{max} = 5$ m/s^2. There are $K = 10$ GUs sharing the total communication bandwidth $B = 1$ MHz with the noise power spectral density $n_0 = 10^{-17}$ W/Hz and the power for communication $P_c = 1$ W. The minimum bandwidth is set as $B_{min} = 5$ kHz. In each episode, the initial locations of the K users are randomly generated in a 700 m $\times$ 700 m square area with uniform distribution, while the speeds of the users have a uniform distribution on the interval [0 m/s, 5 m/s]. Each user follows a fixed mobility pattern that are taken from "straight line", "circle", and "triangle". The initial directions of the users are also randomly distributed on the interval $[0, 2\pi]$. Since the state includes the GUs' locations, the UAV can choose its action according to the GUs' current location. As a result, the mobility pattern of GUs has no substantial impact on the performance. Furthermore, the channel-related parameters are $\eta_a = 12.08$, $\eta_b = 0.11$, $\eta_{LoS} = 1.6$ dB, and $\eta_{NLoS} = 23$ dB. The time slot is fixed as $\delta_t = 0.2$ s. The specific reward settings are shown in Table I.

B. Network Architecture

The specific actor network architecture is shown in Fig. 4, where there are four hidden layers with 100, 150, 100, 50 neurons respectively. The spread dimension is set as $N_e = 10$, and thus the input layer has $3K + 7 + N_e = 47$ neurons. The network outputs the UAV speed, the UAV direction and the frequency band allocation strategy, and thus the output layer has $3 + K = 13$ neurons with one fixed reference neuron. The activation functions of all layers are all *ReLU* functions except for the output layer. Moreover, the critic network has the same hidden layers as actor network with input size $47 + 13 = 60$ and output size 1. The states of the UAV are normalized to $[0, 1]$. We employ ADAM optimizer [42] with a learning rate of 10^{-3} for both the actor and critic networks. The discount factor is $\gamma = 0.99$ and the soft update rate is $\varepsilon = 0.005$. The exploration noise deviation is $\sigma_{\mathcal{N}} = 0.1$.

C. Performance and Analysis

We compare EEFC-TDBA with two baseline methods, *EEFC-TD* and *straight-flight*.[5]

- EEFC-TD: This is a simplified version of EEFC-TDBA without frequency band allocation part, where we re-implement the trajectory design with the same reward settings and hyper-parameters as EEFC-TDBA.
- Straight-flight: The UAV flies straight to the destination. We calculate the speed so that the UAV can reach the destination exactly when it runs out of the energy.

The frequency band allocation strategy for both EEFC-TD and straight-flight are randomly generated.

We compare EEFC-TDBA with the two baselines in terms of fair index, total throughput, fair throughput and minimum throughput. The minimum throughput represents the minimum throughput among GUs. We use the well-trained neural network to execute the energy-efficient fair communication service mission during the implementation phase and calculate these metrics for the entire episode. As illustrated in Table II, EEFC-TDBA outperforms both baselines for all metrics. Compared to straight-flight, EEFC-TD can improve the throughput via trajectory design from 2.151×10^9 to 4.235×10^9, i.e., a 96.9% improvement. Due to the 14.5% improvement of fair index, the increase in fair throughput is greater than that in throughput about 103.9%, and the minimum throughput

[5] To the best of the authors' knowledge, there is existing paper considering the same scenario, and hence we cannot make a comparison with other works.

TABLE II
PERFORMANCE COMPARISON

Metrics \ Methods	**EEFC-TDBA**	EEFC-TD	Straight-flight
Fair index per episode	**0.931**	0.672	0.587
Total throughput per episode	$\mathbf{6.431 \times 10^9}$	4.235×10^9	2.151×10^9
Fair throughput per episode	$\mathbf{5.308 \times 10^9}$	2.654×10^9	1.308×10^9
Minimum throughput per episode	$\mathbf{3.695 \times 10^8}$	1.177×10^8	3.705×10^7

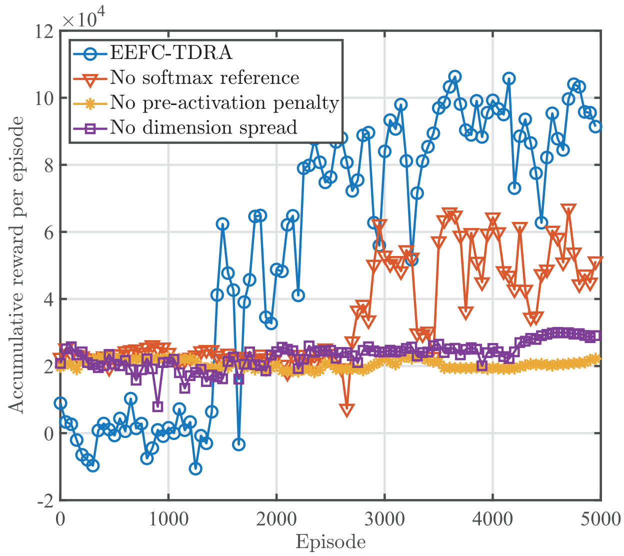


Fig. 8. Effectiveness demonstrations of different techniques. *No softmax reference* means that we apply EEFC-TDBA without softmax reference technique. Similar for the others.

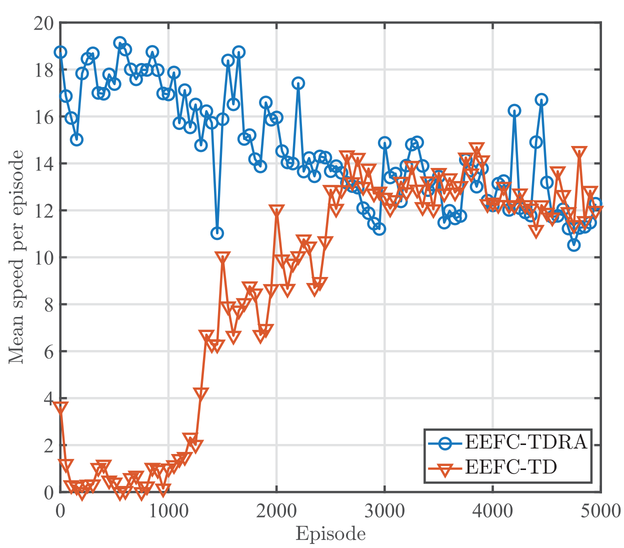


Fig. 9. Mean speed per episode with training.

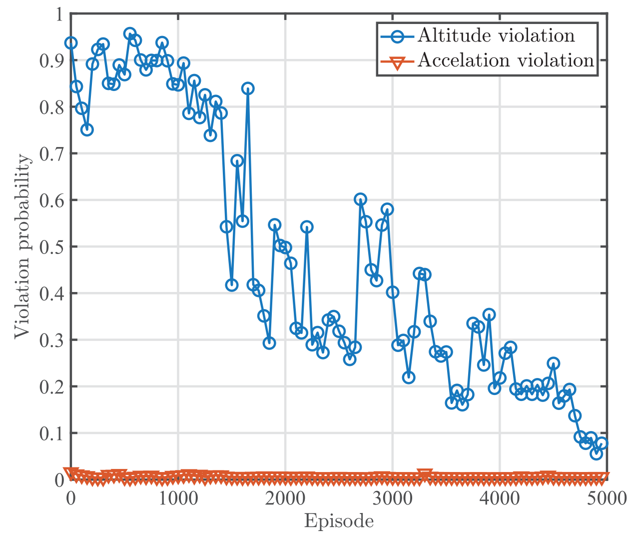


Fig. 10. Probability of constraints violation.

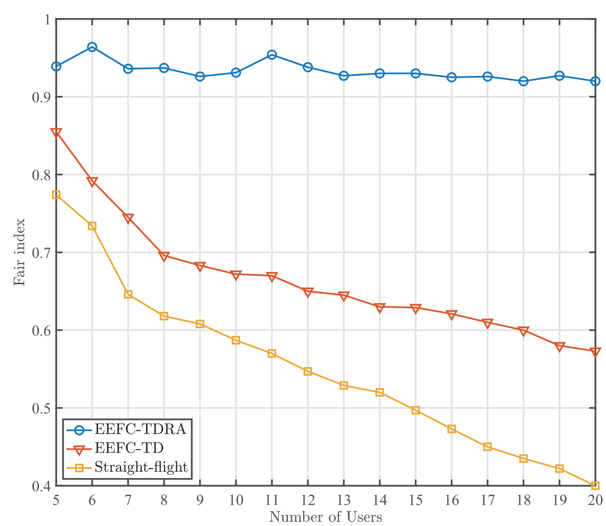


Fig. 11. Fair Index vs number of GUs.

can achieve an improvement of 217.7%. Combining the frequency band allocation to EEFC-TD, EEFC-TDBA can further improve the performance. For example, the fair index can be improved by 38.5%. Note that the closer the fair index is to 1, the harder it is to make further improvement. As a result, compared to trajectory design, frequency band allocation can significantly increase the fairness among the GUs. The total throughput for EEFC-TDBA is about 1.5 times that of EEFC-TD, while EEFC-TDBA has twice the fair throughput and more than three times the minimum throughput of EEFC-TD.

In Fig. 7, we plot the training curves for the accumulative reward, mean fair index, and total fair throughput of EEFC-TDBA and EEFC-TD. With frequency band allocation, EEFC-TDBA outperforms EEFC-TD with sufficient training. We observe an interesting phenomenon that in the first

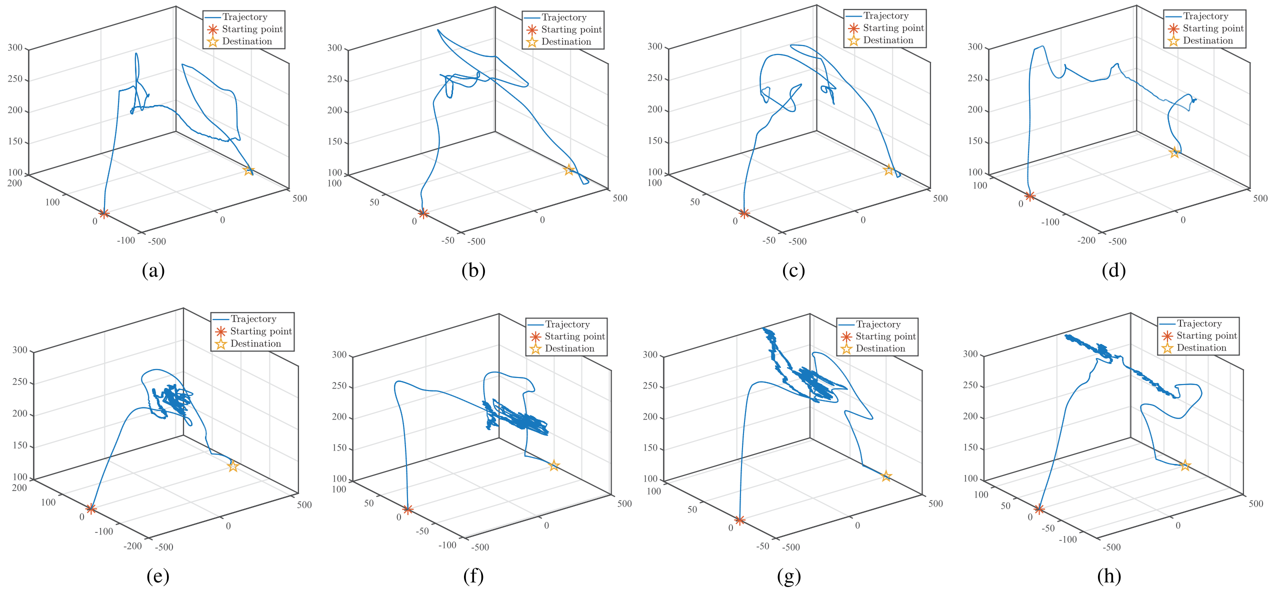


Fig. 12. The UAV 3D trajectories for EEFC-TDBA and EEFC-TD. The red stars and yellow pentagrams denote departure locations and destinations, respectively. Subfigures (a)-(d) plot the trajectories of EEFC-TD, and subfigures (e)-(h) depict the trajectories of EEFC-TDBA. The total flight times of (a)-(d) are 752.8s, 741.4s, 770.4s and 784.4s respectively, while the total flight times of (e)-(h) are 725.8s, 766.0s, 736.6s and 734.0s respectively.

1000 episodes, the fair index and fair throughput of EEFC-TDBA are similar to that of EEFC-TD, but the accumulative reward of EEFC-TDBA is smaller than EEFC-TD and is even smaller than zero. This is because EEFC-TDBA needs to deal with the frequency band allocation problem in addition to trajectory design, which makes it more difficult to control the UAV flying to the destination. The UAV may fly far away from the destination, which leads to the negative reward of reach-destination part. The performance of EEFC-TDBA changes significantly when the number of the episode is around 1500. There are two reasons to explain this observation. First, DRL is different from supervised learning, since it has no clear label information. Thus the training curve of DRL often has obvious oscillation. Second, in an actor-critic based method, the training of actor network rely on the critic network, which means that a bad critic would lead to poor performance. When the training of critic network oscillates, the performance of EEFC-TDBA will change more significantly. After 2000 episodes of training, both EEFC-TDBA and EEFC-TD can control the UAV to reach the destination, and the accumulative reward starts to grow significantly. At 3000 episodes, the fair index of EEFC-TD has a remarkable growth, leading to an increase of the accumulative reward. On the other hand, EEFC-TDBA can further improve the performance, including fair index and fair throughput by frequency band allocation. After 4000 episodes of training, the accumulative reward of EEFC-TDBA converges without significant improvement.

In Fig. 8, we demonstrate the effectiveness of the techniques introduced in Section III-E. We can observe that EEFC-TDBA without pre-activation penalty has no improvement over the course of training. The pre-activation values of the neurons in the output layer are in the saturation area, and there is no effective gradient information for back propagation. Besides, without pre-activation penalty, UAV can not learn to reach the destination when the battery is exhausted. Similarly, the UAV can not accomplish the reach-destination task without dimension spread technique due to the dimension imbalance problem. However, EEFC-TDBA without dimension spread can still improve the accumulative reward by increasing the reward of fair throughput part and achieve a 42.9% improvement compared with no pre-activation penalty. The lack of softmax reference slows the training down and results in a decrease of about 40% compared to EEFC-TDBA, but does not affect the learning of reach-destination task.

As illustrated in Fig. 9, the mean speed of EEFC-TD and EEFC-TDBA converge around 10 m/s to 16 m/s after 3000 episodes of training. As shown in Fig. 2, the propulsion power consumption is smaller in this speed range, which leads to longer flight time and larger accumulative reward.

As mentioned in II-C, the UAV trajectory should satisfy the acceleration and altitude constraints (34) and (35). Fig. 10 plots the violation probability of the acceleration and altitude constraints. We can see that at the beginning of training, the acceleration violation probability drops to almost zero, while the altitude violation probability decreases slowly with training. This is because the acceleration violation is only related to the speed v in the output action, while whether the altitude constraint is violated depends on both the velocity $\boldsymbol{v}$ and the UAV location $\boldsymbol{u}$, which increases the difficulty of training. In addition, once the altitude constraint is violated, the UAV would be located at altitude limit boundary, which further increases the possibility of continuing to violate the altitude constraint. Note that the accumulative reward converges after 4000 episodes training, while the altitude violation probability converges to almost zero at 5000 episodes. The reason is that as the training goes on, the accumulative reward gets larger while the penalty reward of altitude violation gets

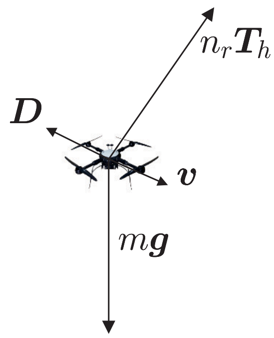


Fig. 13. Schematics of the main forces acting on the UAV.

smaller. In other words, the impact of the altitude violation on accumulative reward decreases significantly. After 4000 episodes of training, the accumulative reward reaches 1×10^5, while the altitude violation penalty reward is only about 1.5×10^3.

Fig. 11 illustrates the impact of the number of GUs on the fair index. We can observe that the fair index of two baselines decreases monotonously as the number of GUs increases, while EEFC-TDBA maintains the fair index at a high level. Increasing the number of GUs would cause more difficulty in fair communication service and it is hard to completely offset this growing difficulty only through trajectory design. As a result, the fair index of EEFC-TD decreases like straight-flight. Nevertheless, EEFC-TDBA can adjust frequency band allocation among GUs to provide fair communication service. Hence, the fair index of EEFC-TDBA will not decline significantly when the number of GUs increases,.

The UAV trajectory is shown in Fig. 12, where both EEFC-TDBA and EEFC-TD can control the UAV to reach the destination when the energy is exhausted. Note that the UAVs will raise their altitudes first. This is because the higher altitude leads to greater probability of obtaining LoS channel and smaller pathloss. It can be seen that the trajectories of both EEFC-TDBA and EEFC-TD are irregular. This is because these two methods choose flight direction and speed according to the locations of GUs. Since GUs are randomly distributed on the ground and move around, the trajectory of a better design should intuitively "match" the distribution of GUs and would look more irregular. We can also see that the trajectories of EEFC-TD method are smoother than that of EEFC-TDBA. This is due to the fact that the output action of EEFC-TDBA also includes the frequency band allocation, which affects the stability of the output of trajectory part.

V. Conclusion

In this paper, we have proposed a DRL based algorithm EEFC-TDBA for 3D UAV trajectory design and frequency band allocation, which enables the UAV to provide energy-efficient and fair communication service. Specifically, the UAV chooses its flight speed, direction and frequency band allocation strategy based on the current location, the current speed, the destination, the remaining energy of UAV, and the GUs' locations. EEFC-TDBA maximizes the fair throughput within limited on-board energy to balance between the throughput maximization and the fairness among the GUs. Simulation results have demonstrated that EEFC-TDBA can achieve much better performance than EEFC-TD and straight-flight in terms of the total throughput and the fairness among the GUs. For the future work, we will investigate the trajectory design and resource allocation of multiple UAVs to provide better communication service for the GUs.

TABLE III
Parameters About Aerodynamics

Notation	Physical meaning	Simulation value
m	UAV mass in kg	2
g	Gravity acceleration in m/s^2	9.8
l_r	Rotor radius in meter (m)	0.1
A	Disc area for each rotor in m^2	0.0314
Ω	Blade angular velocity in radian/second.	—
ρ	Air density in kg/m^3	1.293
c_T	Thrust coefficient based on disc area, $T_h = c_T \rho A \Omega^2 {l_r}^2$	0.302
c_w	Weight coefficient for each rotor, $c_w = \frac{mg}{n_r \rho c_s A \Omega^2 l_r^2}$	—
n_r	Number of rotors	4
n_b	Number of blades	2
l_c	Blade chord length in meter (m)	0.015
c_s	Rotor solidity, $c_s = \frac{n_b l_c}{\pi l_r}$	0.0955
S_{FP}	Fuselage equivalent flat plate area in m^2	0.01
d_0	Fuselage drag ratio for each rotor, $d_0 = \frac{S_{FP}}{n_r c_s A}$	0.834
δ	Local blade section drag coefficient	0.012
c_f	Incremental correction factor to induced power relative to that for linear induced velocity	0.131
v	Forward velocity of the quad-rotor UAV in m/s	—
$\hat{v}$	Forward speed normalized on tip speed, $\hat{v} = \frac{v}{\Omega l_r}$	—
μ	Advance ratio, $\mu \approx \hat{v}$	—
v_0	Thrust velocity, $v_0 = \sqrt{\frac{T_h}{2\rho A}}$	—
v_{i0}	Mean induced velocity	—
λ_i	Mean induced velocity normalized on tip speed, $\lambda_i = \frac{v_{i0}}{\Omega l_r}$	—
D	Drag of fuselage, which is in the opposite direction of the UAV velocity. $D = \frac{1}{2}\rho v^2 S_{FP}$ Equation (4.5) in [27]	—
τ_c	Climb angle	—
T_h	Thrust of each rotor	—
t_{cD}	Thrust coefficient referred to disc axes, $t_{cD} \approx \frac{T_h}{\rho c_s A \Omega^2 L_r^2}$	—
q_c	Torque coefficient, which is related to the power for each rotor. The total power for the UAV is $P = n_r q_c \rho c_s A \Omega^3 l_r^3$	—

Appendix A
Energy Consumption Model for Quad-Rotor UAV in 3D Environment

In the appendix, we formulate the energy consumption model for a quad-rotor UAV in 3D flight.[6] We model the

[6]Inspired by the energy consumption model for single-rotor UAV in 2D flight in [27], we derive a energy consumption model for quad-rotor UAV in 3D environment.

$$P(v,T_h) = n_r q_c \rho c_s A \Omega^3 {l_r}^3$$
$$= n_r \left[\frac{\delta}{8}(\Omega^2 {l_r}^2 + 3v^2)\rho c_s A \Omega l_r + \frac{1}{2} d_0 v^3 \rho c_s A + \frac{mgv}{n_r} \sin\tau_c + (1+c_f) T_h \left(\sqrt{\frac{T_h^2}{4\rho^2 A^2} + \frac{v^4}{4}} - \frac{v^2}{2} \right)^{\frac{1}{2}} \right] \quad (58)$$

energy consumption for the quad-rotor UAV with the assumption that the rotors produce thrusts with the same magnitude and direction. The detailed parameters are provided in Table III.

Based on [43], the propulsion power for a quad-rotor UAV can be calculated by

$$P = n_r q_c \rho c_s A \Omega^3 {l_r}^3, \quad (55)$$

where n_r, ρ c_s and A are constants. The torque coefficient q_c is given by [43, (4.20)]

$$q_c = \frac{\delta}{8}(1+3\mu^2) + (1+c_f)\lambda_i t_{cD} + c_w \hat{v} \sin\tau_c + \frac{1}{2} d_0 \hat{v}^3. \quad (56)$$

By substituting $\mu \approx \hat{v} = \frac{v}{\Omega l_r}$, $t_{cD} \approx \frac{T_h}{\rho c_s A \Omega^2 \ L_r^2}$, $\lambda_i = \frac{v_{i0}}{\Omega l_r}$, $v_{i0} = \left(\sqrt{v_0^4 + \frac{v^4}{4}} - \frac{v^2}{2} \right)^{\frac{1}{2}}$, $v_0 = \sqrt{\frac{T_h}{2\rho A}}$, and $c_w = \frac{mg}{n_r \rho c_s A \Omega^2 \ l_r^2}$ into (56), q_c can be re-written as

$$q_c = \frac{\delta}{8}(1 + \frac{3v^2}{\Omega^2 {l_r}^2}) + (1+c_f) \frac{\left(\sqrt{v_0^4 + \frac{v^4}{4}} - \frac{v^2}{2} \right)^{\frac{1}{2}} T_h}{\rho c_s A \Omega^3 {l_r}^3}$$
$$+ \frac{mgv}{n_r \rho c_s A \Omega^3 {l_r}^3} \sin\tau_c + \frac{1}{2} d_0 \frac{v^3}{\Omega^3 {l_r}^3}. \quad (57)$$

Then the propulsion power can be expressed as a function of UAV speed v and thrust for each rotor T_h (58), shown at the top of the page. According to $T_h = c_T \rho A \Omega^2 {l_r}^2$, the propulsion power can be re-written as (11).

We use *force analysis* on the UAV to establish the relationship between the thrust and the UAV movement. As shown in Fig. 13, the UAV is affected by gravity, air drag and the thrust from the rotors. According to Newton's second law, we have

$$m\boldsymbol{a_c} = n_r \boldsymbol{T_h} + \boldsymbol{D} + m\boldsymbol{g} \quad (59)$$

where $\|\boldsymbol{D}\| = \frac{1}{2}\rho v^2 \ S_{FP}$ is the air drag and is in the opposite direction of the UAV's velocity. Therefore, we obtain (10) as shown in Section II-B.

References

[1] 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.

[2] 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.

[3] 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.

[4] G. Ding, Q. Wu, L. Zhang, Y. Lin, T. A. Tsiftsis, and Y.-D. Yao, "An amateur drone surveillance system based on the cognitive Internet of Things," *IEEE Commun. Mag.*, vol. 56, no. 1, pp. 29–35, Jan. 2018.

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

[6] 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.

[7] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Efficient deployment of multiple unmanned aerial vehicles for optimal wireless coverage," *IEEE Commun. Lett.*, vol. 20, no. 8, pp. 1647–1650, Aug. 2016.

[8] 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.

[9] R. I. Bor-Yaliniz, A. El-Keyi, and H. Yanikomeroglu, "Efficient 3-D placement of an aerial base station in next generation cellular networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2016, pp. 1–5.

[10] J. Lyu, Y. Zeng, R. Zhang, and T. J. Lim, "Placement optimization of UAV-mounted mobile base stations," *IEEE Commun. Lett.*, vol. 21, no. 3, pp. 604–607, Mar. 2017.

[11] J. Zhao, F. Gao, Q. Wu, S. Jin, Y. Wu, and W. Jia, "Beam tracking for UAV mounted SatCom on-the-Move with massive antenna array," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 2, pp. 363–375, Feb. 2018.

[12] L. Xiao, X. Lu, D. Xu, Y. Tang, L. Wang, and W. Zhuang, "UAV relay in VANETs against smart jamming with reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4087–4097, May 2018.

[13] D. Orfanus, E. P. de Freitas, and F. Eliassen, "Self-organization as a supporting paradigm for military UAV relay networks," *IEEE Commun. Lett.*, vol. 20, no. 4, pp. 804–807, Apr. 2016.

[14] X. 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.

[15] 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.

[16] 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.

[17] Y. Yu, X. Bu, K. Yang, H. Yang, and Z. Han, "UAV-aided low latency mobile edge computing with mmWave backhaul," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–7.

[18] C. She, C. Liu, T. Q. S. Quek, C. Yang, and Y. Li, "UAV-assisted uplink transmission for ultra-reliable and low-latency communications," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, May 2018, pp. 1–6.

[19] S.-C. Choi, I.-Y. Ahn, J.-H. Park, and J. Kim, "Towards real-time data delivery in oneM2M platform for UAV management system," in *Proc. Int. Conf. Electron., Inf., Commun. (ICEIC)*, Jan. 2019, pp. 1–3.

[20] 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.

[21] 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.

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

[23] J. Lyu, Y. Zeng, and R. Zhang, "Cyclical multiple access in UAV-aided communications: A throughput-delay tradeoff," *IEEE Wireless Commun. Lett.*, vol. 5, no. 6, pp. 600–603, Dec. 2016.

[24] S. Zhang, H. Zhang, Q. He, K. Bian, and L. Song, "Joint trajectory and power optimization for UAV relay networks," *IEEE Commun. Lett.*, vol. 22, no. 1, pp. 161–164, Jan. 2018.

[25] 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.

[26] Y. Zeng, J. Xu, and R. Zhang, “Rotary-wing UAV enabled wireless network: Trajectory design and resource allocation,” in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–6.
[27] 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.
[28] M. Grant and S. Boyd. (2008). *MATLAB Software for Disciplined Convex Programming (Version 2.1)*. [Online]. Available: http://cvxr.com/cvx
[29] V. Mnih *et al.*, “Playing atari with deep reinforcement learning,” in *Proc. NIPS DEEP Learn. Workshop*, Dec. 2013, pp. 1–9.
[30] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.
[31] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, “Deterministic policy gradient algorithms,” in *Proc. ICML*, 2014, pp. 387–395.
[32] S. Yin, S. Zhao, Y. Zhao, and F. R. Yu, “Intelligent trajectory design in UAV-aided communications with reinforcement learning,” *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 8227–8231, Aug. 2019.
[33] X. Liu, Y. Liu, and Y. Chen, “Reinforcement learning in multiple-UAV networks: Deployment and movement design,” *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 8036–8049, Aug. 2019.
[34] 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.
[35] T. P. Lillicrap *et al.*, “Continuous control with deep reinforcement learning,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2016.
[36] A. Al-Hourani, S. Kandeepan, and A. Jamalipour, “Modeling air-to-ground path loss for low altitude platforms in urban environments,” in *Proc. IEEE Global Commun. Conf.*, Dec. 2014, pp. 2898–2904.
[37] 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, 1984.
[38] G. Laporte, “The traveling salesman problem: An overview of exact and approximate algorithms,” *Eur. J. Oper. Res.*, vol. 59, no. 2, pp. 231–247, Jun. 1992.
[39] V. R. Konda and J. N. Tsitsiklis, “Actor-critic algorithms,” in *Proc. NIPS*, 2000, pp. 1008–1014.
[40] V. Mnih *et al.*, “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, no. 7540, p. 529, 2015.
[41] A. Y. Ng *et al.*, “Policy invariance under reward transformations: Theory and application to reward shaping,” in *Proc. ICML*, vol. 99, Jun. 1999, pp. 278–287.
[42] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2013.
[43] A. R. S. Bramwell, D. Balmford, and G. Done, *Bramwell’s Helicopter Dynamics*. Amsterdam, The Netherlands: Elsevier, 2001.

Ruijin Ding received the B.Eng. degree in electrical and information engineering from the Dalian University of Technology, Dalian, China, in 2017. He is currently pursuing the Ph.D. degree with Tsinghua University, Beijing, China.

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.

Since 2011, he has been with the Department of Automation, Tsinghua University, Beijing, China, where he is currently an Associate Professor. His research interests include signal processing for communications, array signal processing, convex optimizations, and artificial intelligence assisted communications. He has authored/coauthored more than 150 refereed IEEE journal articles and more than 150 IEEE conference proceeding papers that are cited more than 8800 times in Google Scholar. He has served as a technical committee member for more than 50 IEEE conferences. He has also served as the Symposium Co-Chair of the 2019 IEEE Conference on Communications (ICC), the 2018 IEEE Vehicular Technology Conference Spring (VTC), the 2015 IEEE Conference on Communications (ICC), the 2014 IEEE Global Communications Conference (GLOBECOM), and the 2014 IEEE Vehicular Technology Conference Fall (VTC). He has served as an Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, the IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING, the IEEE WIRELESS COMMUNICATIONS LETTERS, and *China Communications*, a Lead Guest Editor for the IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING, and a Senior Editor for the IEEE SIGNAL PROCESSING LETTERS and the IEEE COMMUNICATIONS LETTERS.

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

He is currently a University Professor with the Department of Electrical and Computer Engineering, University of Waterloo, Canada. His research interests include network resource management, wireless network security, the Internet of Things, 5G and beyond, and vehicular ad-hoc and sensor networks. 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 Fellow, and a Distinguished Lecturer of the IEEE Vehicular Technology Society and Communications Society. He is a member of the IEEE Fellow Selection Committee. He received the R. A. Fessenden Award in 2019 from the 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 the Education Award in 2017 from the IEEE Communications Society, the Technical Recognition Award from the Wireless Communications Technical Committee in 2019 and the AHSN Technical Committee in 2013, the Excellent Graduate Supervision Award in 2006 from the University of Waterloo, and the Premier’s Research Excellence Award (PREA) in 2003 from the Province of Ontario. He has served as the Technical Program Committee Chair/Co-Chair of the IEEE GLOBECOM 2016, the IEEE INFOCOM 2014, the IEEE VTC2010 Fall, and the IEEE GLOBECOM2007, the Symposia Chair of the IEEE ICC2010, and the Chair of the IEEE Communications Society Technical Committee on Wireless Communications. He was/is the Editor-in-Chief of the IEEE INTERNET OF THINGS JOURNAL, *IEEE Network*, *IET Communications*, and *Peer-to-Peer Networking and Applications*. He is the elected IEEE Communications Society Vice President for Technical and Educational Activities, the Vice President for Publications, a Member-at-Large on the Board of Governors, and the Chair of the Distinguished Lecturer Selection Committee.