Deep Q-Network Based Dynamic Trajectory Design for UAV-Aided Emergency Communications

Liang Wang[1], Kezhi Wang[1], Cunhua Pan[2], Xiaomin Chen[1], Nauman Aslam[1]

1. Department of Computer and Information Science, Northumbria University, Newcastle upon Tyne, NE1 8ST, U.K.
2. School of Electronic Engineering and Computer Science, Queen Mary University of London, London, E1 4NS, U.K.

Abstract: In this paper, an unmanned aerial vehicle (UAV)-aided wireless emergence communication system is studied, where an UAV is deployed to support ground user equipments (UEs) for emergence communications. We aim to maximize the number of the UEs served, the fairness, and the overall uplink data rate via optimizing the trajectory of UAV and the transmission power of UEs. We propose a Deep Q-Network (DQN) based algorithm, which involves the well-known Deep Neural Network (DNN) and Q-Learning, to solve the UAV trajectory problem. Then, based on the optimized UAV trajectory, we further propose a successive convex approximation (SCA) based algorithm to tackle the power control problem for each UE. Numerical simulations demonstrate that the proposed DQN based algorithm can achieve considerable performance gain over the existing benchmark algorithms in terms of fairness, the number of UEs served and overall uplink data rate via optimizing UAV's trajectory and power optimization.

Keywords: Deep Reinforcement Learning, Deep Q-Network (DQN), Successive Convex Approximation (SCA), UAV, Power Control.

1 Introduction

Unmanned aerial vehicles (UAVs), also known as drones, have been playing an increasingly important role in emergency situations such as earthquake and large fires, where UAVs could be deployed to provide emergency communications for user equipments (UEs) and support life saving activities. It also has the potential to provide other wireless communication related services, such as ubiquitous coverage, relaying, information dissemination, mobile edge computing (MEC) and data collection [1,2,3]. Considering their low cost, high mobility, fast deployment and the direct Line-of-Sight (LoS) connectivity, UAV-enabled wireless communications are expected to achieve higher throughput compared to traditional terrestrial wireless communications.

In order to fully exploit the potential of UAVs, much research has been conducted in the trajectory design of UAV-enabled communications [4,5,6]. In [7], Zeng *et al.* maximized the throughput of UAV-enabled mobile relaying system, whereas in [8], the authors maximized the energy efficiency in a point-to-point UAV-ground communication system. In [9], the authors optimized the altitude of UAV to maximize the radio coverage on the ground. In [5], the UAV was utilized as a mobile base station to serve the ground UEs, and the authors proposed a successive convex approximation (SCA) based algorithm to maximize the minimum average throughput of UEs. In [10], Lyu *et al.* proposed a new cyclical multiple access scheme, where UAV flies cyclically to serve the ground users. In [11], an UAV-enabled secure transmission scheme was proposed in hyper dense networks. For UAV-enabled wireless power transfer networks, Xu *et al.* optimized the trajectory of UAV for the purpose of maximizing the sum of energy received by users. For multi UAV-enabled multiuser system, Yang *et al.* [12] minimized the sum power of user equipment via jointly optimizing the user association, power control, computation capacity allocation, and location planning in a mobile edge computing (MEC) network.

Recently, UAV has been playing an increasingly important role in emergency communications. For instance, during the earthquake, if the local ground station is destroyed, UAV could be deployed to serve as the flying base station to serve the users. They can dynamically move towards the UEs that are out of the communication range, and transmit/receive the data to/from them. In[13], Mozaffari *et al.* addressed some key challenges of deploying UAVs to serve the ground users, such as the optimal deployment and energy efficiency of UAVs. In[14], multiple UAVs were deployed to receive the information from ground UEs, and in order to achieve the reliable uplink communications, the authors proposed to optimize the UAV trajectory and the transmit power of UEs. In[15], Huang *et al.* proposed a differential evolution algorithm to minimize the energy consumption via optimizing the UAV's deployment, such as the number and location of stop points.

Among the recent development in the field of artificial intelligence (AI) and machine learning (ML), reinforcement learning (RL)[16] has become a hot topic both in academia and industry. In[17], Watkins *et al.* introduced a model-free reinforcement learning: Q-learning, which can be viewed as a method of asynchronous dynamic programming (DP). Also, some fundamental elements like agent, state, action, penalty, reward and Q-value were discussed. However, Q-learning is not practical for complicated applications since the number of states and actions will increase exponentially. Thus, combining deep neural networks (DNNs) with RL creates a feasible approach, which could provide more accurate convergence and approximation. In[18], Mnih *et al.* developed a novel solution, i.e., a deep Q-network (DQN), which has achieved an outstanding performance in the challenging domain of Atari 2600 games.

Against the above background, in this paper, we propose a joint UAV trajectory and power control optimization problem to maximize the number of served UEs, the fairness and the overall uplink data rate of UEs in the emergency communication scenario. To this end, we address the UAV trajectory problem by applying DQN framework. Then, based on the given UAV trajectory, we solve the power control problem via using the convex optimization based algorithm.

The rest of this paper is organized as follows. Section II introduces the system model. In Section III, we introduce the proposed algorithm. In Section IV, numerical results are presented to verify the proposed algorithm. Finally, we conclude the paper in Section V.

The main notations used in this paper are summarized in Table 1.

Table 1 Main Notations.

Notation	Definition
$n, N, \mathcal{N}$	the index, the number, and the set of UEs,
$t, T, \mathcal{T}$	the index, the number, and the set of TSs
l^{max}	the side length of the square area
Z^{min}, Z^{max}	minimal and maximal height of the UAV
e^{max}	the maximum energy level of UAV
e_t	the remaining energy level of UAV in TS t
$\alpha_t, \beta_t, \omega_t$	the flying action of UAV in TS t
X_t, Y_t, Z_t	the coordinate of UAV in TS t
x_n, y_n	the coordinate of UE n
$d_{n,t}$	distance between UE n with UAV in TS t
$c_{n,t}$	coverage status of UE n in TS t
$L(\theta_{n,t}, d_{n,t})$	path loss between UE n and UAV is TS t
$\gamma_{n,t}$	SINR at UAV from UE n in TS t
$r_{n,t}$	uplink data rate from UE n to UAV in TS t

2 System Model

As shown in Fig. 1, we consider the emergence situation, where the ground base station is destroyed and the UAV is deployed to provide communication to all the UEs. Assume the UAV flies over a square area with the side length l^{max}. We assume there are N UEs randomly distributed in the target area, and the set of UEs is denoted as $\mathcal{N} \triangleq \{n = 1, 2, ..., N\}$. Also assume the uplink data transmission lasts for T time slots (TSs), and the set of TSs is denoted as $\mathcal{T} \triangleq \{t = 1, 2, ...T\}$. In each TS, the UAV has a flying action $[\alpha_t, \beta_t, \omega_t]$ to conduct, where α_t is the horizontal angle of the flying direction, β_t is the vertical angle of the flying direction, and ω_t is the flying distance. For simplicity, in this paper, we assume that the possible action A_t is chosen from the following set:

$$\begin{aligned} A_t = &\{[\alpha_t, \beta_t, \omega_t] = [\frac{2\pi}{N_\alpha} i, \frac{\pi}{N_\beta} j, \frac{\omega^{max}}{N_\omega} k], \\ &\forall i \in 0, ...N_\alpha,\ j \in 0, ...N_\beta,\ k \in 0, ...N_\omega\}, t \in \mathcal{T}, \end{aligned} \quad (1)$$

where N_α, N_β, and N_ω are the numbers of flying angles and distance that UAV can move in each TS. This means that the UAV can only fly with some specific angles and distance values. ω^{max} is the maximal flying distance in each TS. Note that if the UAV stays at the current location, the action $[\alpha_t, \beta_t, \omega_t] = [0, 0, 0]$, where one can see $i = 0$, $j = 0$, $k = 0$. Otherwise, it moves with the corresponding angles $\frac{2\pi}{N_\alpha} i$, $\frac{\pi}{N_\beta} j$ and the distance $\frac{\omega^{max}}{N_\omega} k$. Hence, the coordinate of UAV in TS t can be denoted as $[X_t, Y_t, Z_t]$, where $X_t = X_0 + \sum_{t'=1}^{t} \omega_{t'} \sin(\beta_{t'}) \cos(\alpha_{t'})$, $Y_t = Y_0 + \sum_{t'=1}^{t} \omega_{t'} \sin(\beta_{t'}) \sin(\alpha_{t'})$,

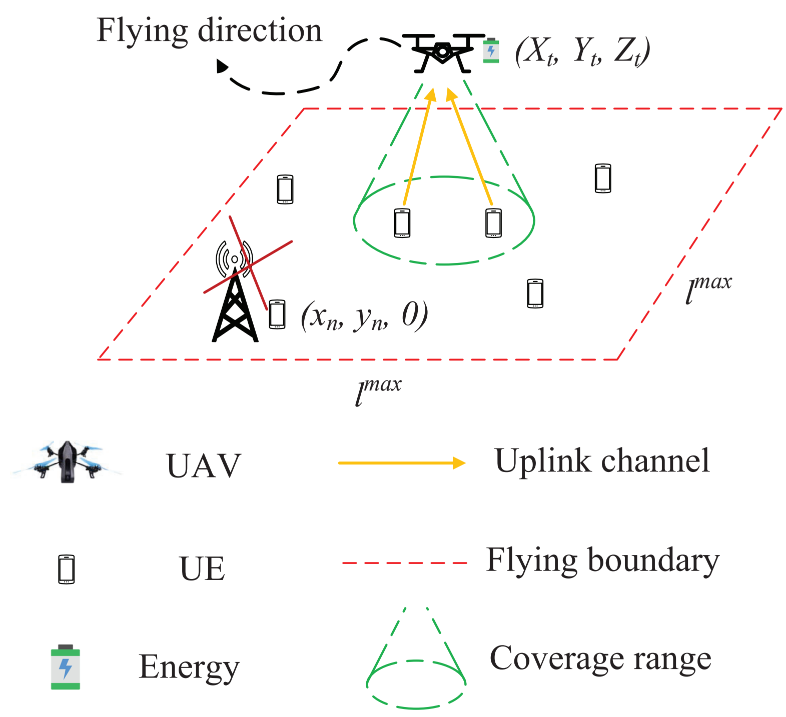


Figure 1 UAV-Aid IoT Data Collection System

and $Z_t = Z_0 + \sum_{t'=1}^{t} \omega_{t'} \cos(\beta_{t'})$, with $[X_0, Y_0, Z_0]$ being the initial coordinate of the UAV. Since the UAV can not fly out of the target area, we have

$$0 \le X_t \le l^{max},\ \forall t \in \mathscr{T}, \tag{2}$$

and

$$0 \le Y_t \le l^{max},\ \forall t \in \mathscr{T}. \tag{3}$$

Additionally, in this paper, we set

$$Z^{min} \le Z_t \le Z^{max},\ \forall t \in \mathscr{T}, \tag{4}$$

where Z^{min}, Z^{max}, are the minimal and maximal flying height of the UAV, for collision avoidance.

Thus, the distance between the UAV and UE n in TS t can be given by

$$d_{n,t} = \sqrt{(X_t - x_n)^2 + (Y_t - y_n)^2 + Z_t^2},\ \forall n \in \mathscr{N}, t \in \mathscr{T}, \tag{5}$$

where $[x_n, y_n]$ is the coordinate of UE n.

Furthermore, in this paper, the UAV has a azimuth angle value of antenna θ', which is based on 3-D Cartesian coordinate, such as x axis, y axis, z axis. Hence, in TS t, the UAV has a maximal horizontal coverage circle with the radius of $R_t^{max} = Z_t \tan(\theta')$ [12] and it varies with the height of the UAV. We also assume that the UAV has the energy constraint e^{max}.

We define the remaining energy level e_t of the UAV in TS t as:

$$e_t = e^{max} - \sum_{t'=0}^{t} \nabla e_{t'}.\ \forall t \in \mathscr{T}, \tag{6}$$

where $\nabla e_{t'}$ is the energy consumed by UAV in TS t', which is defined as [19]

$$\begin{aligned}\nabla e_{t'} = &\left(P_0 \left(1 + 3\frac{v_t^2}{V_r^2} \right) + P_1 \left(\sqrt{1 + \frac{v_t^4}{4V_0^4}} - \frac{v_t^2}{2V_0^2} \right)^{\frac{1}{2}} \right. \\ &\left. + \frac{1}{2} d_0 \rho s \pi R_b^2 v_t^3 \right) T^{max},\end{aligned} \tag{7}$$

where v_t is the flying velocity of UAV in TS t, T^{max} is the maximal time duration of each TS, V_r is the tip speed of the rotor blade, V_0 is the mean rotor velocity when hovering, d_0 is the drag ratio, ρ means the air density, s denotes the rotor solidity, R_b is the radius value of rotor disc. And P_0, P_1 are constant values that can be found in the reference [19]. For simplicity, in this paper, we set $v_t = \frac{\omega_t}{T^{max}}$. Note that we do not consider the energy consumption of data receiving/transmission since it is negligible compared with the moving and hovering energy consumption. Also, to simply the model, we adopt the simplified energy consumption model above, which could be readily extended to the more general model considering different types of UAVs and 3-D flying. In practice, we also assume there is some preserved battery for UAV flying back to the ground, which is ignored here to make the model compact.

In this paper, the 3-D channel model proposed in[9] is adopted. Thus, the mean path loss between the UAV and the UE n in TS t is given by

$$L(\theta_{n,t},d_{n,t})=\frac{\eta_{\text{LoS}}-\eta_{\text{NLoS}}}{1+a\exp\big(-b(\theta_{n,t}-a)\big)}+20\log_{10}(d_{n,t})+ \\ 20\log_{10}\big(\frac{4\pi f_c}{c}\big)+\eta_{\text{NLoS}}, \tag{8}$$

where η_{LoS} and η_{NLoS} (in dB) are the path loss corresponding to the LoS and non-LoS links respectively. a and b are positive constants which can be obtained in[9]. f_c is the carrier frequency (Hz), c is the light speed (m/s), and $\theta_{n,t}=\arctan\big(\frac{Z_t}{\sqrt{(X_t-x_n)^2+(Y_t-y_n)^2}}\big)$.

We denote $c_{n,t}$ as the coverage status of UE n in TS t, and it can be defined as

$$c_{n,t}=\begin{cases}1, & \text{if } \sqrt{(X_t-x_n)^2+(Y_t-y_n)^2}\leq R_t^{max},\\ 0, & \text{Otherwise.}\end{cases} \tag{9}$$

Additionally, we assume that if the UE n is under the coverage of UAV in TS t, i.e., $c_{n,t}=1$, the UE n is served by UAV and the data collection from UE n to UAV is started. Thus, the corresponding signal-to-interference-plus-noise ratio (SINR) at the UAV can be expressed as

$$\gamma_{n,t}=\frac{c_{n,t}P_{n,t}10^{-\frac{L(\theta_{n,t},d_{n,t})}{10}}}{\sum_{n'=1,n'\neq n}^{N}c_{n',t}P_{n',t}10^{-\frac{L(\theta_{n',t},d_{n',t})}{10}}+\sigma^2}, \tag{10}$$

where $P_{n,t}$ means the transmit power of UE n in TS t; σ^2 is the additive white Gaussian noise (AWGN) at the receiver. Therefore, the uplink data rate from UE n to the UAV in TS t is expressed as

$$r_{n,t}=\log_2\big(1+\gamma_{n,t}\big),\ \forall n\in\mathcal{N},t\in\mathcal{T}. \tag{11}$$

One can also apply the power constraint as follows, then we have

$$0\leq P_{n,t}\leq P^{max},\ \forall n\in\mathcal{N},t\in\mathcal{T}, \tag{12}$$

where P^{max} is the maximum transmit power of UEs.

In this paper, we also aim to maximize the number of UEs served by UAV via optimizing the UAV trajectory. Then we define C_t as follows

$$C_t=\frac{1}{N}\sum_{n=1}^{N}c_{n,t},\forall t\in\mathcal{T}, \tag{13}$$

which can represent the proportion of the number of UEs served by UAV in TS t. However, this may lead to unfair serving process since some UEs are covered for many TSs and the rest UEs may be never covered at all. Therefore, similar to the references [20,21], we apply the fairness index among all UEs, which is defined as

$$f_t=\frac{\big(\sum_{n=1}^{N}\sum_{t'=1}^{t}c_{n,t'}\big)^2}{N\sum_{n=1}^{N}\big(\sum_{t'=1}^{t}c_{n,t'}\big)^2}, \tag{14}$$

where f_t reflects the quality of service (QoS) level that the UEs served by UAV from the initial TS to the TS t. More precisely, if all the UEs are served for the similar number of TSs, the fairness value f_t is closer to 1.

Additionally, we define the overall data rate of UEs served by UAV in TS t as

$$R_t=\sum_{n=1}^{N}c_{n,t}r_{n,t},\forall t\in\mathcal{T}. \tag{15}$$

Thus, we formulate the optimization problem as follows

$$\mathscr{P}1:\max_{\boldsymbol{U},\boldsymbol{P}}\sum_{t=1}^{T}\big(f_t\cdot C_t\cdot R_t\big), \tag{16a}$$

subject to:

$$A_t=\{[\alpha_t,\beta_t,\omega_t]=\big[\frac{2\pi}{N_\alpha}i,\frac{\pi}{N_\beta}j,\frac{\omega^{max}}{N_\omega}k\big], \\ \forall i\in 0,\dots N_\alpha,\ j\in 0,\dots N_\beta,\ k\in 0,\dots N_\omega\},t\in\mathcal{T}, \tag{16b}$$

$$0\leq X_t\leq l^{max},\ \forall t\in\mathcal{T}, \tag{16c}$$

$$0\leq Y_t\leq l^{max},\ \forall t\in\mathcal{T}, \tag{16d}$$

$$Z^{min}\leq Z_t\leq Z^{max},\ \forall t\in\mathcal{T}, \tag{16e}$$

$$0\leq P_{n,t}\leq P^{max},\ \forall n\in\mathcal{N},t\in\mathcal{T}, \tag{16f}$$

where $\boldsymbol{U}=\{X_t,Y_t,Z_t,\forall t\in\mathcal{T}\}$ and $\boldsymbol{P}=\{P_{n,t},\forall n\in\mathcal{N},t\in\mathcal{T}\}$. It is readily to see that the above problem cannot be solved by traditional optimization approach as it involves discrete variables $\boldsymbol{U}$ and continuous variables $\boldsymbol{P}$. Additionally, all three factors cannot be achieved optimally at the same time since each factor will have a negative effect on others. Thus, we aim to achieve the optimal balance between them. Then, in this paper, we first propose a DQN-based algorithm to solve the UAV trajectory problem. Next, based on the optimized UAV trajectory, we further propose a successive convex approximation (SCA) based algorithm to solve the power control problem.

3 Proposed Algorithm

Before presenting the proposed algorithm, we first introduce some important knowledge of deep reinforcement learning.

3.1 Background Knowledge

In the traditional reinforcement learning structure, there is an agent interacting with the environment through a series of states, actions and rewards. In each time step, the agent selects the policy that maps the state and action with the aim of maximizing the accumulated reward. Specifically, the process of interacting with the environment can be expressed with an action-value function named Q-function, which is defined as

$$Q(s,a) = \max_{\pi}\mathbb{E}\big[Z|s_t = s, a_t = a\big], \tag{17}$$

where Q is known as Q-value, π denotes the policy by taking the action a at the state s and Z is the reward.

Although DRL combines DNN with Q-learning, it may still have instability or divergence. Since DNN may be seen as the non-linear function approximator, small updates to Q-value may significantly vary the policy, or even change the data distribution as well as the correlations between action-value and target value. Therefore, to address this issue, in[18], Mnih *et al.* introduced the DQN framework, which contains a pair of mechanisms: Firstly, they applied the experience replay, where the mini-batch randomly samples several transitions $\{s_t, a_t, z_t, s_{t+1}\}$ to train the DQN. This mechanism removes the correlation of state sequences and smooths over changes in the data distribution. Secondly, an iterative updating mechanism was deployed. Specifically, there is a target network periodically updating for the purpose of adjusting the action-value towards the target value.

3.2 The Proposed DQN Algorithm

In this section, the proposed DQN algorithm is presented, where we assume there is an agent interacting with the environment. The agent controls the UAV and aims to select the optimal policy that can maximize the accumulated reward $Z_t = \sum_{t'=t}^{T}\gamma^{t'-t}z_{t'}$ by giving a set of states $\mathscr{S} \triangleq \{s_t = s_1, s_2, ...s_T\}$ and actions $\mathscr{A} \triangleq \{a_t = a_1, a_2, ..., a_T\}$, where $\gamma \in [0,1]$ is the discount factor. More specifically, we describe the state, action and reward in TS t as follows:

1. State s_t: the state of agent in TS t has two components.
   (a) UAV 's current coordinate: $\{X_t, Y_t, Z_t\}$.
   (b) UAV's current energy level: $\{e_t\}$.
2. Action a_t: we define action $a_t = \{\alpha_t, \beta_t, \omega_t\}$ as the UAV's horizontal angle α_t, vertical angle β_t and distance ω_t in TS t, where $a_t \in A_t$.
3. Reward Function z_t: we define the reward function as:

$$z_t = f_t \cdot C_t \cdot R_t - p, \tag{18}$$

where p is the penalty if UAV flies out of the target area and R_t can be obtained by the proposed convex optimization based solutions in Algorithm 2.

In the proposed DRL shown in Fig. 2, there are two DQN networks, namely evaluation and target networks, respectively[18]. Note that the evaluation and target networks have the same structure but the latter updates periodically. The agent selects the action according to the evaluation network and the agent follows an ε-greedy policy.

According to the state s_t and action a_t, the agent obtains the reward r_t and then the environment transfers to the next state s_{t+1}. The transition $\{s_t, a_t, z_t, s_{t+1}\}$ can be stored in the experience replay memory with size M^{max}. Once the learning process starts, the mini-batch randomly samples K transitions from the memory. The evaluation network is trained by the sequence of the loss function, which can be expressed as

$$L_i(\theta_i) = \mathbb{E}_{s,a}\Big[\big(y_i - Q(s,a|\theta_i)\big)^2\Big], \tag{19}$$

where i is the index of iteration, $y_i = \mathbb{E}\big[z + \gamma\max_{a'} Q(s', a'|\theta_{i-1})\big]$ and it can be obtained by the target network.

During the interaction with the environment, the agent selects the optimized action of UAV associated with the evaluation network, which follows a ε-greedy policy. Specifically, the agent can select the action that has the largest Q-value with probability ε, or randomly select the action from the action set A_t with probability $1-\varepsilon$. Also, the agent obtains state s_t, next state s_{t+1} and reward z_t from the environment. Note that the data rate R_t of reward z_t is calculated by the proposed convex optimization based algorithm provided in Algorithm 2. Then, the transition, which consists of $\{s_t, a_t, z_t, s_{t+1}\}$, is stored in the experience replay memory. Once the learning procedure starts, the mini-batch randomly samples K transitions from the experience replay memory. Given the Q-value $Q(s,a)$ and the target value y_i obtained by the evaluation and target network, the loss function provided by (19) is used to update the evaluation network and the target network is updated periodically.

Furthermore, we provide the pseudo code of proposed DQN algorithm in Algorithm 1. Specifically, from Line 1 to 2, we initialize the evaluation network, target network and experience replay memory. Then, in each training episode, we initialize the state s_t, and the evaluation network generates the action by the given state s_t. Note that a ε-greedy policy is employed to select the optimized action. Specifically, a variable $\varepsilon_t \in [0,1]$ is generated. If $\varepsilon_t \leq \varepsilon$, we select the action a_t that has the largest Q-value. Otherwise, we select a random action a_t. Next, the agent executes the action a_t, obtains the reward z_t provided by (18) and the environment transfers to the next state s_{t+1}. Note that the UAV stays at the current location and the agent receives a penalty if the UAV flies out of the target area. The transition is stored in the experience replay

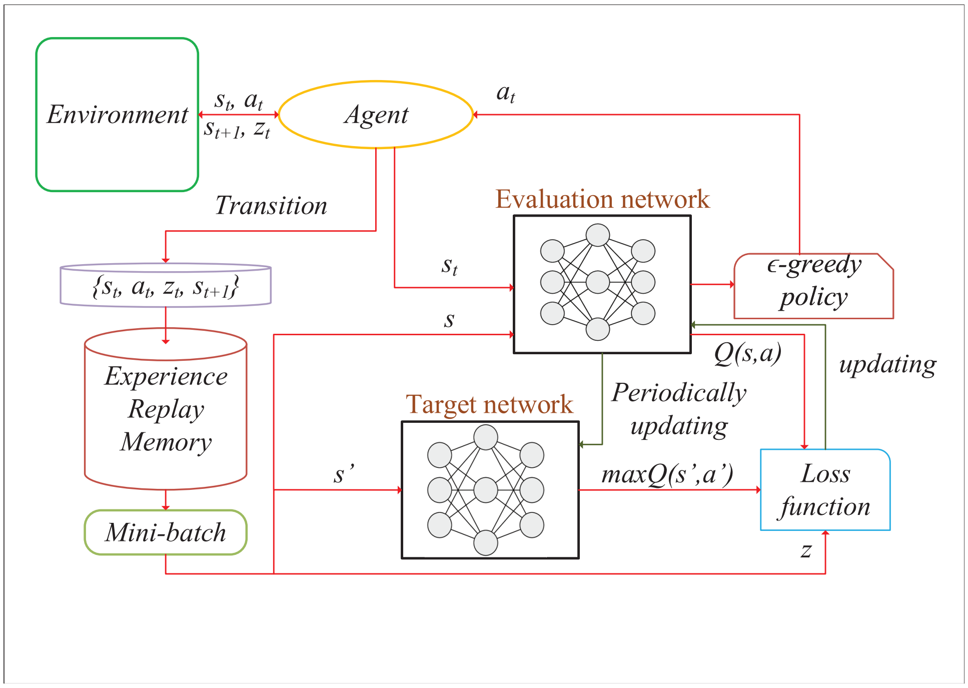


Figure 2 Structure of proposed DQN

memory. From line 17, once the learning process starts, the learning procedure starts. The mini-batch randomly samples K transitions from the memory for calculating the loss value. Then, we perform a gradient descent step on loss value calculated by loss function with respect to the network parameters θ. Finally, we update evaluation network and target network periodically.

3.3 Power Control Algorithm

In order to maximize the reward z_t with the given trajectory, we further propose a convex optimization-based algorithm for handling the power control of all UEs. Then, in TS t, given the UAV trajectory, the maximization problem of reward function (18) is transformed into the following problem:

$$\max_{\boldsymbol{P}} f_t \cdot C_t \cdot R_t - p, \tag{20a}$$

subject to:

$$0 \leq P_{n,t} \leq P^{max}, \forall n \in \mathcal{N}, \tag{20b}$$

from which, both f_t, C_t and p are fixed. Motivated by[22], via introducing the auxiliary variable η, the problem is transformed into

$$\max_{\eta,\boldsymbol{P}} \eta, \tag{21a}$$

subject to:

$$f_t \cdot C_t \cdot \sum_{n=1}^{N} c_{n,t} r_{n,t} - p \geq \eta, \forall n \in \mathcal{N}, \tag{21b}$$

$$0 \leq P_{n,t} \leq P^{max}, \forall n \in \mathcal{N}. \tag{21c}$$

Problem (21) is a non-convex optimization since (21b) is a non-convex constraint. It is noted that $r_{n,t}$ can be expressed as

$$\begin{aligned} r_{n,t} &= \log_2\left(1 + \frac{c_{n,t}P_{n,t}10^{-\frac{L(\theta_{n,t},d_{n,t})}{10}}}{\sum_{n'=1,n'\neq n}^{N} c_{n',t}P_{n',t}10^{-\frac{L(\theta_{n',t},d_{n',t})}{10}} + \sigma^2}\right) \\ &= \log_2\left(\sum_{n=1}^{N} c_{n,t}P_{n,t}10^{-\frac{L(\theta_{n,t},d_{n,t})}{10}} + \sigma^2\right) - \widetilde{r}_{n,t}, \ \forall n \in \mathcal{N}, \end{aligned} \tag{22}$$

where

$$\widetilde{r}_{n,t} = \log_2\Big(\sum_{n'=1,n'\neq n}^{N} c_{n',t}P_{n',t}10^{-\frac{L(\theta_{n',t},d_{n',t})}{10}} + \sigma^2\Big), \ \forall n \in \mathcal{N}. \tag{23}$$

In order to solve the above non-convex constraint of (21b), we apply the successive convex approximation (SCA) to calculate the value of $\widetilde{r}_{n,t}$. Specifically, we define $\boldsymbol{P}^k = \{P^k_{n,t}, \forall n \in \mathcal{N}\}$ as the given transmission power of UEs in TS t

Algorithm 1 The proposed DQN algorithm

```
Initialize evaluation network, target network with parameters θ;
Initialize experience replay memory with size M^max;
for Episode = 1,2,...,E^max do
    Initialize state s_t = [X_0, Y_0, Z_0, e^max];
    for TS = 1,2,...T do
        Obtain s_t;
        ε_t = rand(0,1);
        if ε_t ≤ ε then
            a_t = argmax Q(s_t, a_t);
        else
            Select a random action a_t from A_t;
        end if
        Execute a_t;
        Obtain z_t according to Algorithm.2;
        Obtain s_{t+1};
        Store transition {s_t, a_t, z_t, s_{t+1}} into experience replay memory;
        if the learning process starts then
            Randomly sample K transitions from memory;
            Obtain loss value according to (19);
            Perform a gradient descent step on loss value with respect to the network parameters θ;
            Update evaluation network;
            Update target network periodically;
        end if
    end for
end for
```

in the k-th iteration. Inspired by[23], any concave function can be globally upper-bounded by its first-order Taylor expansion at any point. Hence, by given $\boldsymbol{P}^k$, one has

$$\begin{aligned}\widetilde{r}_{n,t} &= \log_2\left(\sum_{n'=1,n'\neq n}^{N} c_{n',t}P_{n',t}10^{-\frac{L(\theta_{n',t},d_{n',t})}{10}}+\sigma^2\right)\\ &\leq \sum_{n'=1,n'\neq n}^{N}\frac{c_{n',t}10^{-\frac{L(\theta_{n',t},d_{n',t})}{10}}\log_2(e)}{\sum_{l=1,l\neq n}^{N}c_{l,t}P_{n',t}^{k}10^{-\frac{L(\theta_{l,t},d_{l,t})}{10}}+\sigma^2}(P_{n',t}-P_{n',t}^{k})\\ &\quad+\log_2\left(\sum_{n'=1,n'\neq n}^{N}c_{n',t}P_{n',t}^{k}10^{-\frac{L(\theta_{n',t},d_{n',t})}{10}}+\sigma^2\right)\triangleq \widetilde{r}_{n,t}^{up}.\end{aligned} \tag{24}$$

With any given local point $\boldsymbol{P}^k$ and the upper bound $\widetilde{r}_{n,t}^{up}$, Problem (21) can be transformed into

$$\max_{\eta^k,\boldsymbol{P}} \eta^k, \tag{25a}$$

subject to:

$$f_t\cdot C_t\cdot\sum_{n=1}^{N}c_{n,t}\left(\log_2\Big(\sum_{n=1}^{N}c_{n,t}P_{n,t}10^{-\frac{L(\theta_{n,t},d_{n,t})}{10}}+\sigma^2\Big)-\widetilde{r}_{n,t}^{up}\right)\geq\eta^k,\ \forall n\in\mathcal{N}, \tag{25b}$$

$$0\leq P_{n,t}\leq P^{max},\ \forall n\in\mathcal{N}. \tag{25c}$$

One can see that the above problem is now been converted to the convex optimization, which can be solved efficiently by the standard convex optimization solver, e.g., CVX[23]. Then, we provide the pseudo code in Algorithm 2.

Algorithm 2 The proposed convex optimization based algorithm

```
Obtain a_t according to the DQN network;
Execute a_t;
Obtain f_t, C_t according to Eq. (14) and Eq. (9);
Initialize P^0;
k = 0;
repeat
    Solve Problem (25) for given P^k;
    Denote the optimal solution as P^{k+1};
    k = k + 1;
until The convergence is achieved
```

As shown in Algorithm 2, we first obtain the state of UAV s_t, execute a_t and obtain f_t and C_t. Then, we initialize $\boldsymbol{P}^0$, and solve Problem (25) for given $\boldsymbol{P}^k$. Next, we repeat the process until the convergence is achieved.

4 Simulation Result

In this section, we evaluate the performance of proposed DQN and convex optimization based algorithm. The simulation is executed by using Python 3.7, Tensorflow 1.15[24]. CVXPY 1.0.24[25] is used in the convex optimization based algorithm. We deploy two fully-connected hidden layer with $[400\times 300]$ neurons in DQN networks. The learning rate is 0.001 and RMSOptimizer is used to update DQN networks. We set the target area to be a square with side length $l^{max}=400$ m and 30 UEs are randomly distributed in the target area. In each training episode, the UAV always starts from the same initial point, i.e., $[X_0,Y_0,Z_0]=[5,5,70]$. In each TS, once the UE is covered by UAV, UAV starts data collection from the UE. More parameters can be found in Table. 2.

We first analyze the overall reward achieved by DQN algorithm in each training episode (i.e., 20 TSs) in Fig. 3, from which, we observe that the overall reward remains negative at

Table 2 Parameter Setting.

Parameter	Description	Parameter	Description
N	30	ω^{max}	30 m
l^{max}	400 m	N_α	6
N_β	5	N_ω	4
Z_{min}	50 m	Z^{max}	100 m
P^{max}	0.1 W	ε	0.9
e^{max}	200 KJ	T^{max}	1 s
θ'	$\frac{\pi}{4}$	P_0	79.85
P_1	88.63	V_r	120
V_0	4.03	d_0	0.6
ρ	1.225	s	0.05
R_b	0.4 m	η_{LoS}	1.6 dB
η_{NLoS}	23 dB	f_c	2.5 GHz
c	3×10^8 m/s	a	12.08
b	0.11	σ^2	-100 dBm
γ	0.99	K	256
M^{max}	10^5	p	2

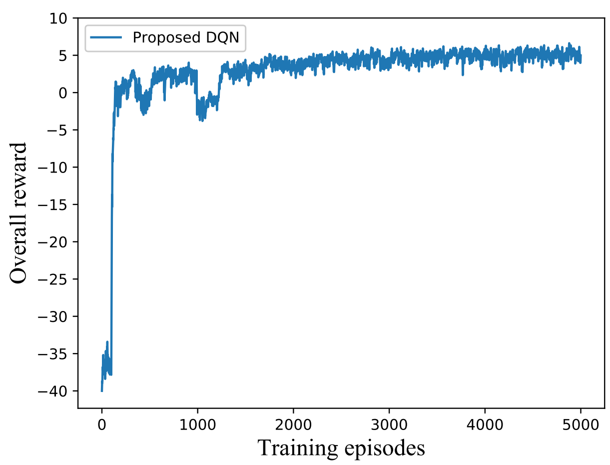


Figure 3 Overall reward versus training episodes.

the beginning. This is because the UAV always flies out of the target area, which means the penalty is always incurred. When the learning process starts, the agent learns to optimize the UAV trajectory from the exploration process and the DQNs start converging, which increases the overall reward. Once the convergence is achieved, the overall reward remains about 5, which shows the best UAV trajectory and transmission power of each UE are obtained.

After adequate training, the model and their parameters are saved for testing. We analyse the performance of the proposed DQN algorithm during the testing procedure in Fig. 4. Specifically, we first evaluate the accumulated fairness in different numbers of TSs in Fig. 4(a), from which we can observe that the accumulated fairness keeps rising from 0 and stabilizes at 7. In Fig. 4(b), one can see that the accumulated coverage increases from 0 to 6 eventually. Then, we evaluate the accumulated data rate (bps/Hz) of UEs served by UAV in Fig. 4(c), from which we observe that the data rate keeps rising with the increase of the number of TSs. It reaches about 4 bps/Hz finally. Overall, one can see from Fig. 4 that our proposed DQN algorithm can learn from experience and reach the considerable performance.

Then, for comparison, we present the following baseline algorithms:

- Random: In each TS, UAV randomly selects a horizontal angle value α_t, a vertical angle value β_t and distance value ω_t from the action set A_t. Additionally, it randomly selects the power control $P_{n,t}$ for each UE. It is worth mentioning that the UAV is restricted to the target area.
- Maximum rate: In each TS, the UAV always selects the action a_t from A_t that can maximize the instantaneous data rate, which is defined as

$$a_t = \max_{a_t} R_t|_{a_t \in A_t}. \tag{26}$$

Note that in this solution, the UEs served by UAV always transmit their data with maximal power consumption as P^{max}.

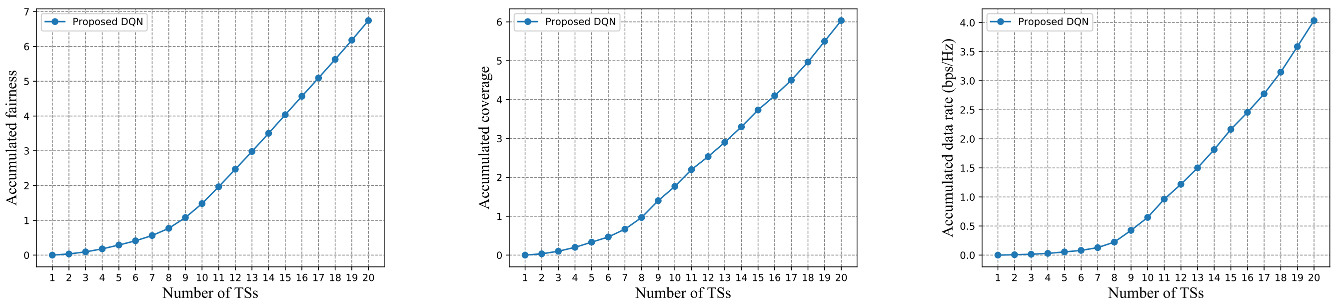
- Maximum reward: In each TS, the UAV selects the action of UAV a_t that can maximize the reward.

Similar as before, the maximum transmission power P^{max} is applied for each UE.

Then, we evaluate the performance of the proposed DQN algorithm and the above baseline solutions in different number of TSs in Fig. 5. It is worth mentioning that it is quite challenging to achieve the best solution in all three factors, i.e., fairness, coverage and data rate at the same time. On one hand, the UAV will keep flying for serving different UEs for maximizing the fairness, which will inevitably reduce the data rate and consume more energy of UAV. On the other hand, the UAV will tend to stay at the location that can maximize the data rate, which will have a negative effect on fairness and coverage. Besides, maximizing the number of UEs served by the UAV will lead to severe interference between UEs, which will also reduce the overall data rate. However, as our objective is to maximize the overall reward consisting of all the three factors. Our proposed solution can achieve the best performance in this regard and will be shown below.

First, in Fig. 5(a), we analyse the impact of the number of TSs on fairness. One can observe that the proposed DQN algorithm outperforms other baselines in all the examined cases. It can always achieve the fairness above 0.35, where as the other three algorithms can only achieve fairness below 0.2.

Then, in Fig. 5(b), one can see that in terms of coverage, our proposed DQN-based solution performs the best, which



(a) (b) (c)

Figure 4 The accumulated (a) fairness, (b) coverage and (c) data rate over one episode during testing.

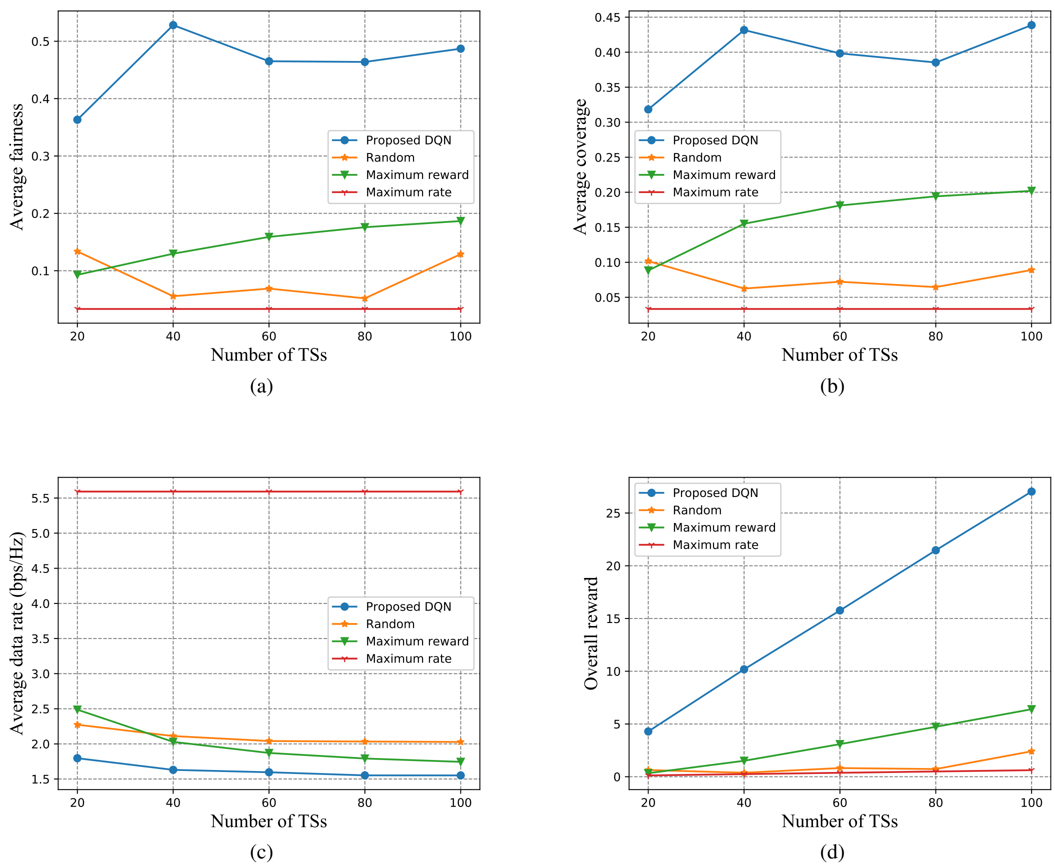


Figure 5 The average (a) fairness, (b) coverage, (c) data rate, and (d) overall reward versus different number of TSs which UAV possesses.

can reach close to 0.45, However, other benchmark algorithms can reach at most around 0.2.

Furthermore, we evaluate the performance in terms of average data rate of UEs served by UAV in Fig. 5(c). One observes that the "maximum rate" solution has the best performance, as it aims at maximizing the data rate of the users, while "random" and "maximum reward" perform slightly better than our proposed DQN. The explanation is that the UAV controlled by "maximum rate" only serves a few UEs, which will lead to lower interference between UEs, however, it cannot guarantee the coverage and fairness, as shown before. Our proposed DQN-based solution, as it will also consider the coverage, and it may serve several UEs at the same time, resulting in lower data rate due to interference among different UEs.

Then, as shown in Fig. 5(d), we depict the overall reward achieved by the DQN-based solution and other baselines in a single episode with respect to different number of TSs. One can observe that with the increase of the number of TSs, the overall reward of all algorithms increase. The proposed DQN has the best performance, as expected. Other benchmark algorithms have lower performance, as they only focus on one factor, such as data rate.

5 Conclusion

In this paper, we have considered the UAV-aided emergency communications, where the UAV is deployed in the case that the ground base station is destroyed. We propose a DRL based DQN algorithm to optimize the UAV trajectory. Additionally, we present a convex optimization based algorithm to optimize the power transmission of UEs served by UAV. Simulation results show that the proposed algorithm can achieve the considerable performance gain over the existing algorithms in terms of fairness, the total number of UEs served and the overall data rate of UEs.

References

[1] ZENG Y, WU Q, ZHANG R. Accessing from the sky: A tutorial on UAV communications for 5G and beyond [J/OL]. Proceedings of the IEEE, 2019, 107(12):2327-2375. DOI: 10.1109/JPROC.2019.2952892.

[2] WANG L, WANG K, PAN C, et al. Multi-agent deep reinforcement learning based trajectory planning for multi-UAV assisted mobile edge computing[J]. IEEE Transactions on Cognitive Communications and Networking, 2020:1-12.

[3] WANG L, HUANG P, WANG K, et al. RL-based user association and resource allocation for multi-UAV enabled MEC[C]//2019 15th International Wireless Communications Mobile Computing Conference (IWCMC). Piscataway: IEEE Press, 2019: 741-746.

[4] DU Y, YANG K, WANG K, et al. Joint resources and workflow scheduling in UAV-enabled wirelessly-powered MEC for IoT systems[J/OL]. IEEE Transactions on Vehicular Technology, 2019, 68(10):10187-10200. DOI: 10.1109/TVT.2019.2935877.

[5] WU Q, ZHANG R. Common throughput maximization in UAV-enabled OFDMA systems with delay consideration[J]. IEEE Transactions on Communications, 2018, 66(12):6614-6627.

[6] YANG Z, PAN C, SHIKH-BAHAEI M, et al. Joint altitude, beamwidth, location, and bandwidth optimization for UAV-enabled communications[J/OL]. IEEE Communications Letters, 2018, 22(8):1716-1719. DOI: 10.1109/LCOMM.2018.2846241.

[7] ZENG Y, ZHANG R, LIM T. Throughput maximization for UAV-enabled mobile relaying systems[J]. IEEE Transactions on Communications, 2016, 64(12):4983-4996.

[8] ZENG Y, ZHANG R. Energy-efficient UAV communication with trajectory optimization[J]. IEEE Transactions on Wireless Communications, 2017, 16(6):3747-3760.

[9] AL-HOURANI A, KANDEEPAN S, LARDNER S. Optimal LAP altitude for maximum coverage[J]. IEEE Wireless Communications Letters, 2014, 3(6):569-572.

[10] LYU J, ZENG Y, ZHANG R. Cyclical multiple access in UAV-aided communications: A throughput-delay tradeoff[J]. IEEE Wireless Communications Letters, 2016, 5 (6):600-603.

[11] ZHAO N, CHENG F, YU F R, et al. Caching UAV assisted secure transmission in hyper-dense networks based on interference alignment[J]. IEEE Transactions on Communications, 2018, 66(5):2281-2294.

[12] YANG Z, PAN C, WANG K, et al. Energy efficient resource allocation in UAV-enabled mobile edge computing networks[J]. IEEE Transactions on Wireless Communications, 2019, 18(9):4576-4589.

[13] MOZAFFARI M, SAAD W, BENNIS M, et al. Unmanned aerial vehicle with underlaid device-to-device communications: Performance and tradeoffs[J]. IEEE Transactions on Wireless Communications, 2016, 15(6): 3949-3963.

[14] MOZAFFARI M, SAAD W, BENNIS M, et al. Mobile internet of things: Can UAVs provide an energy-efficient mobile architecture?[C/OL]//2016 IEEE Global Communications Conference (GLOBECOM). Piscataway: IEEE Press, 2016: 1-6. DOI: 10.1109/GLOCOM .2016.7841993.

[15] HUANG P, WANG Y, WANG K, et al. Differential evolution with a variable population size for deployment optimization in a UAV-assisted IoT data collection system[J/OL]. IEEE Transactions on Emerging Topics in Computational Intelligence, 2019:1-12. DOI: 10.1109/TETCI.2019.2939373.

[16] SUTTON R S, BARTO A G. Reinforcement learning: An introduction[M]. [S.l.]: MIT press, 2018.

[17] WATKINS C J, DAYAN P. Q-learning[J]. Machine learning, 1992, 8(3-4):279-292.

[18] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Human-level control through deep reinforcement learning[J]. Nature, 2015, 518(7540):529-533.

[19] ZENG Y, XU J, ZHANG R. Energy minimization for wireless communication with rotary-wing UAV[J]. IEEE Transactions on Wireless Communications, 2019, 18(4):2329-2345.

[20] JAIN R, DURRESI A, BABIC G. Throughput fairness index: An explanation[C]//ATM Forum contribution: volume 99. [S.l.: s.n.], 1999.

[21] LIU C H, CHEN Z, ZHAN Y. Energy-efficient distributed mobile crowd sensing: A deep learning approach[J]. IEEE Journal on Selected Areas in Communications, 2019, 37(6):1262-1276.

[22] WU Q, ZENG Y, ZHANG R. Joint trajectory and communication design for multi-UAV enabled wireless networks[J]. IEEE Transactions on Wireless Communications, 2018, 17(3):2109-2121.

[23] BOYD S, BOYD S P, VANDENBERGHE L. Convex optimization[M]. [S.l.]: Cambridge university press, 2004.

[24] ABADI M, AGARWAL A, BARHAM P, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems[Z]. [S.l.: s.n.], 2016.

[25] DIAMOND S, BOYD S. CVXPY: A Python-embedded modeling language for convex optimization[J]. Journal of Machine Learning Research, 2016, 17(83):1-5.

About the Authors

Liang Wang received his B.Eng. degree in 2014 and MSc. degree in 2015. He is currently working towards the Ph.D. degree in computer science with Northumbria University, Newcastle upon Tyne, U.K. His research interests include UAV communication, mobile edge computing, and machine learning. (Email: liang.wang@northumbria.ac.uk)

Kezhi Wang [corresponding author] received his B.E. and M.E. degrees in School of Automation from Chongqing University, China, in 2008 and 2011, respectively. He received his Ph.D. degree in Engineering from the University of Warwick, U.K. in 2015. He was a senior research officer in University of Essex, U.K. from 2015-2017. Currently he is a Senior Lecturer with Department of Computer and Information Sciences at Northumbria University, U.K. His research interests include mobile edge computing, intelligent reflection surface (IRS) and machine learning. (Email: kezhi.wang@northumbria.ac.uk)

Cunhua Pan received the B.S. and Ph.D. degrees from the School of Information Science and Engineering, Southeast University, Nanjing, China, in 2010 and 2015, respectively. From 2015 to 2016, he was a Research Associate at the University of Kent, U.K. He held a post-doctoral position at Queen Mary University of London, U.K., from 2016 and 2019, where he is currently a Lecturer.

His research interests mainly include intelligent reflection surface (IRS), machine learning, UAV, Internet of Things, and mobile edge computing. He serves as a TPC member for numerous conferences, such as ICC and GLOBECOM, and the Student Travel Grant Chair for ICC 2019. He also serves as an Editor of IEEE Wireless Communication Letters, IEEE Communication Letters and IEEE ACCESS. (Email: c.pan@qmul.ac.uk)

Xiaomin Chen received the Ph.D. degree in Mathematics from Hamilton Institute, National University of Ireland Maynooth in Nov 2012. From Dec 2012 to Aug 2013, she worked as a Postdoc Research Fellow in Hamilton Institute, NUIM. In 2014, she moved to the UK and worked as a Research Associate in the Department of Computer Science, Loughborough University from Sep 2014 to Aug 2015. In Jan 2016, she joined the Department of Computer and Information Sciences, Northumbria University, UK as a senior lecturer. Her current research lies in wireless networking, resource allocation and optimization, network-level coding, network security. (Email: xiaomin.chen@northumbria.ac.uk)

Nauman Aslam received the Ph.D. degree in engineering mathematics from Dalhousie University, Halifax, NS, Canada, in 2008. He is currently a Professor with the Department of Computer Science and Digital Technologies, Northumbria University, Newcastle upon Tyne, U.K. Prior to joining Northumbria University, he was an Assistant Professor with Dalhousie University. His research interests include wireless sensor network, energy efficiency, security, and WSN health applications. (Email: nauman.aslam@northumbria.ac.uk)