Trajectory Planning of UAV in Wireless Powered IoT System Based on Deep Reinforcement Learning

Jidong Zhang*, Yu Yu†, Zhigang Wang*, Shaopeng Ao†,
Jie Tang†, Xiuyin Zhang† and Kai-Kit Wong‡

*Guangdong Communictions and Networks Institute, China,
†School of Electronic and Information Engineering, South China University of Technology, China,
‡Department of Electronic and Electrical Engineering, University College London, UK

***Abstract*—In this paper, a UAV-assisted wireless powered communication system for IoT network is studied. Specifically, the UAV performs as base station (BS) to collect the sensory information of the IoT devices as well as to broadcast energy signals to charge them. Considering the devices' limited data storage capacity and battery life, we propose a multi-objective optimization problem that aims to minimize the average data buffer length, maximize the residual battery level of the system and avoid data overflow and running out of battery of devices. Since the services requirements of IoT devices are dynamic and uncertain and the system can not be full observed by the UAV, it is challenging for UAV to achieve trajectory planning. In this regard, a deep Q network (DQN) is applied for UAV's flight control. Simulation results indicate that the DQN-based algorithm provides an efficient UAV's flight control policy for the proposed optimization problem.**

I. Introduction

In the recent years, the Internet of Things (IoT) has drawn the attention of the research community. It tends to shift services and applications towards wider integration and accessibility [1]. In 5G enabled IoT system, massive IoT devices are widely deployed to observe different physical processes. The quality-of-service (QoS) for many real-time applications, e.g., event monitoring and predication for health safety [2] [3], is restricted by the freshness of information. However, the existing cellular networks are unable to support the massive IoT communications with wide coverage, ultra low latency and low deployment costs [4]. Besides, considering the limited capabilities of both data storage and energy capacity of the IoT devices, timely delivery of collected information and charging are essential for many realtime IoT devices. It is of great challenge for 5G network to meet the diverse requirements of the IoT.

The recent advance in RF (radio frequency)-enabled wireless energy transfer (WET) technology provides an attractive solution by powering wireless devices with continuous and stable energy over the air [5]. WET is based on a dedicated energy source and is more stable and controllable than renewable sources, such as wind and solar [6]. Enabled by WET, wireless powered communication (WPC), where wireless devices use harvested energy for communication, is envisioned as an important building block of 5G enabled IoT system. It can be applied in massive IoT devices with much lower maintenance cost and enhanced flexibility in practical deployment [7].

Benefiting from the deployment flexibility and maneuverability, unmanned aerial vehicles (UAVs) are introduced into the next-generation wireless systems as a mobile base station to enhance coverage, capacity, reliability, and energy efficiency [8] [9] [10]. Applied with WET technology, they can operate as flying mobile data collector and/or charger for wildly distributed IoT devices. For example, in [11], UAV was applied in a backscatter communication system as a mobile data relay to collect data from terrestrial tags and then uplink the information to base station. In [12], a UAV-enabled WPC network with two UAVs and two ground IoT-devices was considered. The UAVs was dispatched to periodically charge the IoT-devices in the downlink and collect information in the uplink.

As the mobile data collector and charger of IoT system, UAV is required to achieve real-time control in uncertain and dynamic IoT network environments. Reinforcement learning (RL) [13], in which an agent learns its optimal policy through interaction with its environment, has been efficiently used in the optimization decision of network entities. To deal with complex and large-scale networks, deep learning [14] has been prevailing in RL to improve the learning speed and the decision-making ability. Deep Q learning (DQL) is one of the successes. In DQN, deep neural network (DNN) was applied in the Q-learning algorithm so that it can efficiently obtain an optimal policy even the state space and action space are large. It is an emerging tool to effectively address various problems and challenges in the areas of communications and networking, e.g. for dynamic network access and adaptive data rate control [15], wireless caching and data offloading [16].

There has been many research works on the application of DRL in UAV networks and/or IoT networks in recent years. In [17] and [18], the authors did not consider the different priorities of different PoIs. It has been improved in [19], but the data distribution and priority requirements was assumed to be certain and published at the beginning. The author of [20] studied an online data processing network, where the data generation model and the freshness real-time update process have been presented. The observations of surrounding environment was taken as input of a convolutional neural networks(CNN) network which was trained to predict the reward of each action and an online path planning algorithm based on DQN was developed. Along with the age of information (AoI) for the

collected data, in [21], stable and reliable energy supply has also been considered. The author considered a freshness-aware wireless powered communication systems in which multiple source nodes were responsible for sending update packets to a common destination node and harvest energy from the destination to enable a self-perpetuating operation. The battery level at each source node and the AoI values for different processes at the destination were updated in real time and a DRL algorithm was proposed to solve the weighted sum-AoI minimization problem. Compared to these related work, the differences as well as contribution of our work are summarized as follow:

- We consider a UAV-assisted wireless powered IoT system in which the data transfer and charging priority requirements of different devices are different and updated dynamically in different patterns over time.
- We propose a multi-objective optimization problem that aims to minimize the average data buffer length, maximize the residual battery level of the system and avoid devices' data overflow and running out of battery. The trajectory planning of UAV jointly considers the service requirements of data transfer and charging and reward coefficients are introduced to guide the optimization direction.
- We use DQN algorithm to enable the trajectory planning of the UAV. Since the global network information is challenging to be precisely known in advance, rather than including a large number of network information in state space as above mentioned work did, we extract several features which reflect the status of the system as the input of the network. The simulation results illustrate that our algorithm can achieve efficient coverage so as to prevent battery drainage and data queue overflow of the IoT devices. Furthermore, the influence of the degrees of freedom of the action space to the result is shown.

The rest of this paper is organized as follows. Section II presents the system model and problem definition. The DQN-based trajectory planning strategy is introduced in Section III. The results of the simulations are presented in Section IV, and finally, conclusion is provided in Section V.

II. System Model and Problem Formulation

In this section, we first describe the UAV-assisted wireless powered communication system model and then mathematically formulate the optimization problem.

A. System Description

We consider a wireless powered IoT communication system with a UAV and a certain amount of IoT devices randomly distributed in a certain geographical region. The UAV is deployed as mobile data collector and energy transmitter to collect data and charge the IoT devices. We assume that the UAV flies horizontally at a certain altitude to provide data collection and charging service and is aware of its own location. The horizontal position of the UAV is denoted as $[x(t), y(t)]$, where $t = 1, 2, ..., T$ represents the number of time slots. Since we assume it flies at a certain altitude, its hovering altitude is omitted here. $v(t) \in [0, v_{\max}]$ indicates its flying speed, where $v_{\max}$ is the maximum flying speed. $\theta(t) \in [0, 2\pi]$ is the flying direction (i.e., angle). The UAV is restricted to a certain geographical region. Once it tries to fly out of range, it will be forced to hover at the original location. $N_f(t)$ is given to record the number of times that the UAV tries to fly out of bounds until time slot t. We suppose the maximum coverage radius of the UAV as D.

Let $\mathcal{K} \triangleq \{k = 1, 2, \cdots, K\}$ be a set of IoT devices. As for IoT device $k \in \mathcal{K}$, let $\lambda_k(t)$ represents its data generation rate per time slot, where $\lambda_k(t)$ obeys the poisson distribution, and varies from device to device. $b_k^d(t)$ denotes the data length of data buffer at time slot t. It is updated at the beginning of each time slot according to

$$b_k^d(t+1) = b_k^d(t) + \lambda_k(t), \tag{1}$$

$b_k^d(t) \in [0, b_{\max}^d]$, where $b_{\max}^d$ is the data buffer storage capacity. Here we assume $b_{\max}^d$ is same for all the sensor devices. When $b_k^d(t)$ is beyond $b_{\max}^d$, it means that the new gathered data can not be put into the data buffer and will be dropped. It may lead to the loss of important information. We use $N_d(t)$ to represent the number of devices with data overflow at time slot t. Once $b_k^d(t) > b_{\max}^d$, we set $b_k^d(t) = b_{\max}^d$ and increase $N_d(t)$ by 1.

As for charging requirements, $b_k^e(t)$ denotes the residual energy in the battery of device k at time slot t and $\mu_k(t)$ denotes its energy consumption rate per time slot. $\mu_k(t)$ of different device is different. The specific values at different time slot obeys the gaussian distribution. It is updated according to

$$b_k^e(t+1) = b_k^e(t) - \mu_k(t), \tag{2}$$

where $b_k^e(t)$ is upper bounded by the capacity of the battery $b_{\max}^e(t)$. We use $N_e(t)$ to represent the number of devices that run out of battery at t. Once $b_k^e(t) \leq 0$, we set $b_k^e(t) = 0$ and increment $N_e(t)$ correspondingly.

B. Problem Formulation

Based on the above system model, we aim to achieve the flight control of the UAV to provide efficient and timely data collection and charging services for IoT devices. More specifically, every step the UAV takes tries to get the IoT devices with high demand for services into its coverage so as to reduce the overall data buffer level and improve the residual battery level of the system as well as to avoid the data overflow and battery exhaustion of any IoT device. At the beginning of each time slot, the UAV updates its position according to local observations. Hovering at the new position, the UAV broadcasts the information to notify the IoT devices within its coverage area to transmit their buffered data. After the information is uploaded, the data length of data buffer is updated as

$$b_k^d(t) = 0, \quad \forall \Delta d_k(t) \leq D, \tag{3}$$

where $\Delta d_k(t)$ is the distance between the UAV and the device k. While the devices transmitting their gathered data to the UAV by the uplink access, the information of their battery level will be sent to the UAV at the same time to ask for

charging. We assume that the devices will be fully charged. Then the residual battery is updated as

$$b_k^e(t) = b_{\max}^e(t), \quad \forall \Delta d_k(t) \leq D. \tag{4}$$

The average data buffer length $B_{avg}(t)$ and the the residual battery level $E_{avg}(t)$ of the whole system at time slot t are given as

$$B_{avg}(t) = \frac{1}{K}\sum_{k=1}^{K}\frac{b_k^d(t)}{b_{\max}^d}. \tag{5}$$

$$E_{avg}(t) = \frac{1}{K}\sum_{k=1}^{K}\frac{b_k^e(t)}{b_{\max}^e}. \tag{6}$$

According to our goal, we aim to find a control strategy to achieve UAV's trajectory planning to 1) minimize the average data buffer length; 2) maximize the residual battery level; 3) minimize the total number of devices that have data overflow and ran out of energy. To sum up, the multi-objective optimization problem is formulated as follows:

$$\mathbf{P1}: \max_{[v(t),\theta(t)]} \begin{bmatrix} -B_{avg}(t) \\ E_{avg}(t) \\ -N_d(t) \\ -N_e(t) \end{bmatrix}^{\mathrm{T}} \tag{7}$$

$$\text{s.t.} \quad v(t) \in [0, v_{\max}], \quad \forall t = 1, 2, \cdots, T. \tag{8}$$

$$\theta(t) \in [0, 2\pi], \quad \forall t = 1, 2, \cdots, T. \tag{9}$$

Since the data collection rate and power consumption rate of the IoT devices change dynamically over time and have a certain randomness, the status of network environments is uncertain and stochastic. Besides, the environment state can't be full observed by the UAV. Conventional optimization algorithms like convex optimization are not applicable for this problem, neither is conventional dynamic programming method which is a model based approach. To this end, we come up with a DQN-based strategy to solve this sophisticated optimization problem.

III. UAV TRAJECTORY PLANNING BASED ON DQN ALGORITHM

In this section, we will propose a DQN-based reinforcement learning framework, in which the UAV will learn and build knowledge about the IoT networking environment. And then provide a flight control policy to solve the proposed multi-objective optimization problem.

A. Preliminaries

Here we first provide a clear and simple account of the key ideas and algorithms of RL [13]. It is based on an autonomous agent observing a state $s_t \in \mathcal{S}$ from its environment at time t, where $\mathcal{S}$ is the set of states. At each discrete decision epoch t, the agent observes state s_t, interacts with the environment by executing an action $a_t \in \mathcal{A}$, and then receives a reward r_t. Then the agent and the environment will go to a new state s_{t+1} accordingly. $\mathcal{A}$ is the set of actions. The policy is defined as a mapping from state s_t to a probability of choosing action a_t, which can be represented as

$$\pi : \mathcal{S} \to \mathcal{P}(\mathcal{A}). \tag{10}$$

The sum of discounted future reward as return is defined as

$$R_t = \sum_{i=t}^{T}\gamma^{(i-t)} r\left(s_i, a_i\right), \tag{11}$$

where $r(\cdot)$ is the reward function and $\gamma \in [0, 1]$ is a discount factor. γ determines the effect of future rewards to current caching decisions. A lower value of γ places more emphasis on immediate rewards.The objective of RL is to find the optimal policy π which maps a state to an action to maximize the average long-term cumulative reward. The optimal policy can be represented as

$$\pi^* = \arg\max_{\pi} \mathbb{E}\left[R_t|\pi\right]. \tag{12}$$

The action-value function is used to describe the expected return after taking an action a_t in state s_t and thereafter following policy π:

$$Q^{\pi}\left(s_t, a_t\right) = \mathbb{E}_{r_{i\geq t}, s_{i>t}\sim E, a_{i>t}\sim\pi}\left[R_t|s_t, a_t\right]. \tag{13}$$

The commonly used off-policy algorithm, Q-learning, uses the greedy policy $\mu(s) = \arg\max_a Q(s, a)$ to find the optimal policy [22]. θ^Q denotes the parameterized function approximators, which is optimized by minimizing the loss:

$$L\left(\theta^Q\right) = \mathbb{E}\left[\left(Q\left(s_t, a_t|\theta^Q\right) - y_t\right)^2\right], \tag{14}$$

where y_t is the target value and is given by

$$y_t = r\left(s_t, a_t\right) + \gamma Q\left(s_{t+1}, \mu\left(s_{t+1}\right)|\theta^Q\right). \tag{15}$$

In practical application, realistic tasks tend to have large state space. It is difficult for traditional reinforcement learning to deal with high-dimension input. Based on Q-table, Q-learning will become impractical with high-dimensional sensory inputs. For this case, DQN, in which the deep neural network function approximators are used to estimate the action-value function was developed [23]. The combination of reinforcement learning and deep neural networks has made it possible for the agent to learn concepts in a more primitive state. Besides, DQN comes up with two key techniques to overcome the instability and difficulty of convergence of reinforcement learning. One of the key ideas is experience relay and the other is target network. Random sampling from the replay buffer removes correlations in the observation sequence and allows the algorithm to learn across a set of uncorrelated transitions. Which more, the target network parameters are updated periodically by having them slowly track the learned networks to improve the stability of learning.

B. Proposed DQN-based UAV trajectory planning algorithm

To enable the DQN algorithm in the proposed optimization problem, the state space, action space and reward function are defined as follow:

- State space:

$$\mathcal{S} \triangleq \{\mathbf{s}_t\} = \{[\Delta d_x(t), \Delta d_y(t), x(t), y(t), N_f(t), N_d(t), N_e(t)]\}. \tag{16}$$

We assume that the UAV is aware of the location of the IoT device with the highest service requirement priority and set it as the target location. $[\Delta d_x(t), \Delta d_y(t)]$ is the distance between the UAV and the target device. Besides, the state consists of the location of the UAV as well as the number of times it has continuously attempted to fly out of the designated area, the number of devices with data overflow and running out of the battery of the IoT network.

- Action space:

$$\mathcal{A} \triangleq \{\mathbf{a}_t\} = \{[v(t), \theta(t)]\}. \tag{17}$$

The action consists of the flying speed and angle of the UAV. Both of the flying speed and angle are uniformly discretized into m values within their range, where m is the number of degrees of freedom.

- Reward function: On one hand, we reward the UAV based on service requirements priority of devices within its coverage. The service requirement priority not only consider the data buffer length and the residual battery level of the IoT devices in the current slot, but also take their trend into account. The service requirement priority of device k to upload data and charging at time slot t are defined as $Q_k^d(t)$ and $Q_k^e(t)$ respectively, which are given as

$$Q_k^d(t) = \lambda_k(t) \frac{b_k^d(t)}{b_{\max}^d(t)}. \tag{18}$$

$$Q_k^e(t) = \mu_k(t) \frac{b_{\max}^e(t) - b_k^e(t)}{b_{\max}^e(t)}. \tag{19}$$

On the other hand, we punish the UAV for its flying out of the designated area and leading to data overflow and battery expiry of devices. Based on these setting, the reward at time slot t is set as

$$\begin{aligned} r_t = & r_d \sum_{k, \forall \Delta d_k(t) \le D} Q_k^d(t) + r_e \sum_{k, \forall \Delta d_k(t) \le D} Q_k^e(t) \\ & - p_f N_f(t) - p_d N_d(t) - p_e N_e(t), \end{aligned} \tag{20}$$

where r_d and r_e are the reward coefficients of data collection and charging respectively. And p_f, p_d and p_e are the corresponding weight parameters.

The complete algorithm to solve the proposed optimization problem for UAV trajectory planning is summarized in Algorithm 1.

Algorithm 1 DQN-based Algorithm For UAV trajectory planning

```
Initialize action-value function Q with random weights θ;
Initialize target action-value function Q̂ with weights θ' = θ;
Initialize replay buffer B to capacity N;
Initialize ϵ for action exploration;
for episode := 1, ··· , M do
    Initialize the environment and receive initial state s_1 according to (16);
    for t := 1, ··· , T do
        Update the system status according to (1)(2) and record N_d(t), N_e(t);
        With probability ϵ select a random action a_t;
        if a_t will lead the UAV fly out of the designated area then
            a_t = [0, 0], N_f(t) += 1;
        end if
        Execute action a_t, observe reward r_t according to
        (20) and observe new state s_{t+1} accordingly;
        Store transition (s_t, a_t, r_t, s_{t+1}) in B;
        Randomly sample a minibatch of N transitions from B ;
        Train the network and update parameters θ using gradient descent;
        Every C steps update θ' = θ;
    end for
end for
```

TABLE I: Network configurations

Parameters	Values
Number of training episodes	1200
Number of time steps	500
Learning rate	0.01
Reward discount	0.9
Replay memory size	500
Batch size	32
Exploration probability	0.9
network structure	[400,300]

IV. SIMULATION RESULTS AND DISCUSSION

Simulation are carried out to evaluate the performance of our proposed algorithm. We assume that 50 IoT devices are randomly distributed in a 400×400 units area. Their data accumulation rate $\lambda_k(t)$ obey poisson distribution with different expectations, which are randomly assigned from the set $\{4, 8, 14, 18\}$. Their energy consumption rate $\mu_k(t)$ obey gaussian distribution with different expectations and same variance. Expectations are randomly assigned from the set $\{0.2, 0.4, 0.6, 0.8\}$ and the variance is set to 0.05. The capacity of data buffer $b_{\max}^d$ is set as 2500 units and the battery capacity $b_{\max}^e$ is set as 150 units. At the beginning of each task (e.g.,episode), the buffered data length $b_k^d(1)$ and the residual battery level $b_k^e(1)$ of the devices are randomly initialized in the interval $[0, b_{\max}^d]$ and $[0, b_{\max}^e]$ respectively. And the UAV hovers at a random position in the designated area. The radius of UAV coverage is 10 units, and the maximum flying speed $v_{\max} = 40$ units. The reward parameters: $r_d = 100, r_e = 100, p_f = p_d = p_e = 5$. Once the UAV gets the target device into its coverage, r_t will add 500 in addition. Our simulation runs were performed with Tensorflow 2.1.0 and Python 3.7. The detail configurations setting of the DQN network are given

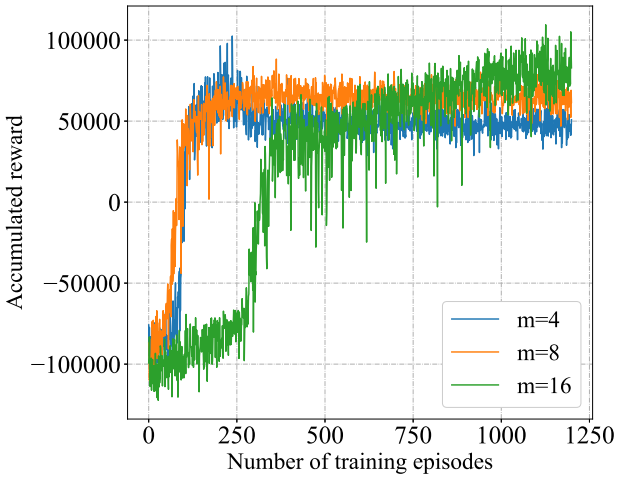

Fig. 1: Training curves tracking the agent's accumulated reward with respect to the number of degrees of freedom.

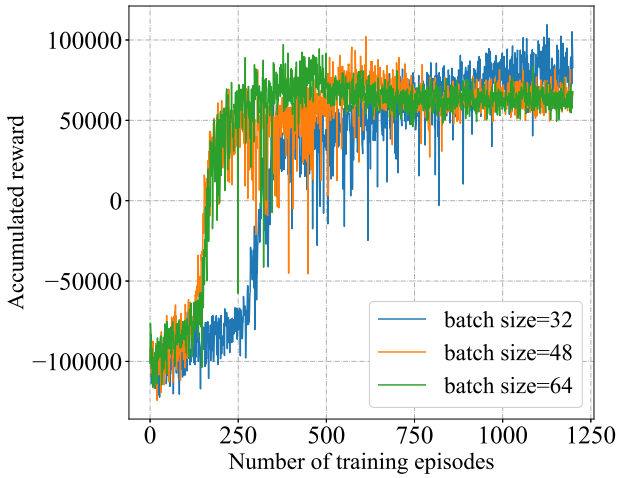

Fig. 2: Training curves tracking the agent's accumulated reward with respect to batch size.

in TABLE I.

First we present the training curves that tracking the agent's accumulated reward, as shown in Fig. 1. The number of degrees of freedom of the action space is set as $4, 8, 16$. From Fig. 1, we can observe that the accumulated reward fluctuates at very low values at first, then rises rapidly, and finally converges steadily at a high level when $m = 4, 8$. The reason for the change is that at first training episodes, UAV is in the experience stage. Without any knowledge of the environment, the action is almost randomly chosen. When the UAV has accumulated enough samples, it begins to use the accumulated sample to train network, and finally obtains the flight control policy. Besides, we can see that the convergence rate decreases with the increase of m. For $m = 16$, the accumulated reward didn't convergence after 1200 training episodes. The reason is that the number of actions increases exponentially with the number of degrees of freedom. With the same experiment parameter setting, larger action spaces are more difficult to explore efficiently. Fig. 2 shows the influence of the batch size to convergence rate. We consider the case that $m = 16$ and the batch size is set as $32, 48$ and 64. We can see that increasing batch size can accelerate the rate of convergence. In terms of the training result after convergence, it can be observed that the finally accumulated reward of $m = 8, 16$ are almost the same and both of them are larger than the case that $m = 4$.

In order to verify the effectiveness of our proposed algorithm as well as to further analyze the influence of different degrees of freedom of the action space on the performance of trajectory planning policy, we present the training curves of the optimization objectives of the proposed problem, as shown in Fig. 3 and Fig. 4. It can be observed in Fig. 3 that for all cases, the average data buffer length of devices goes from more than 0.7 to about 0.4 and the residual battery level goes from less than 0.35 to about 0.6. Besides, Fig. 4 shows that the average number of devices with data overflow and the average number of devices out of battery of each episodes converge to less than 5 in the end. It is clearly that the proposed DQN-based algorithm has provided an efficient solution to

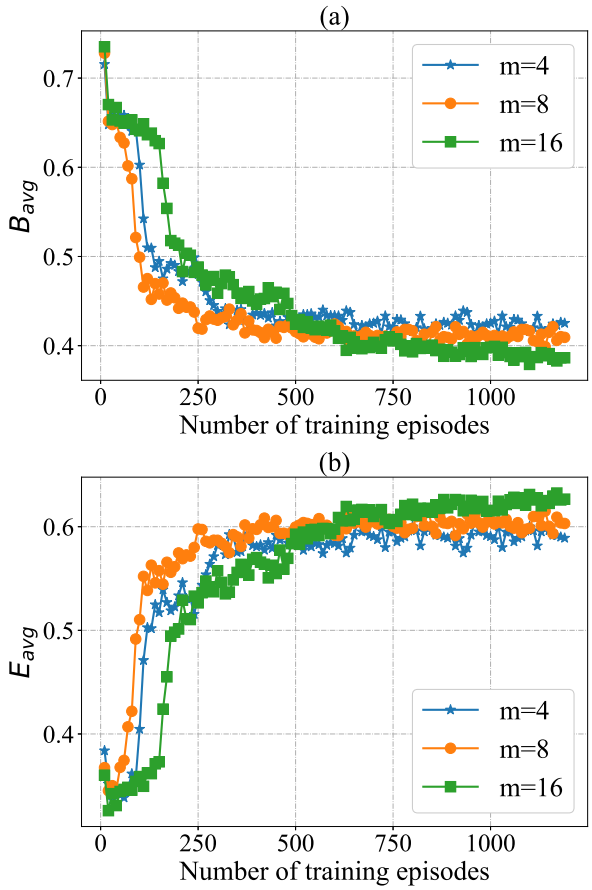

Fig. 3: The training curves of (a) Average data buffer length, (b) Average residual battery level.

the optimization problem **P1**. Considering the influence of the number of degrees of freedom of the action space, we can see that the greater the degree of freedom, the better performance the trajectory planning policy can achieve. Even though the finally accumulated reward of $m = 8$ and $m = 16$ are almost the same, as shown in Fig. 1 and Fig. 2, the case that $m = 16$ can achieve the better performance in term of

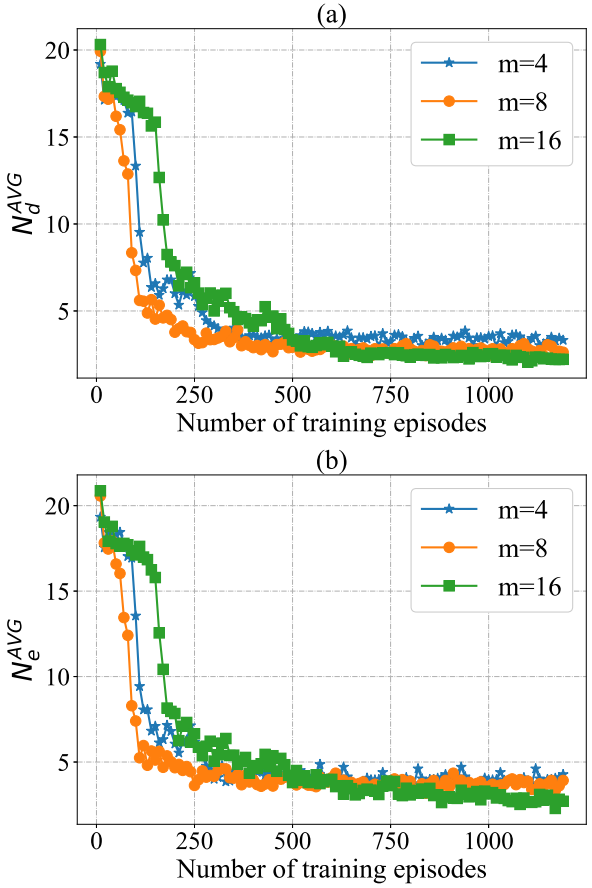

Fig. 4: The training curves of (a) Average number of devices with data overflow, (b) Average number of devices out of battery.

all the optimization objectives. The reason is that with greater degree of freedom of the action space, the UAV has higher freedom of movement and can achieve finer control.

V. Conclusion

In this paper, we are committed to realizing flight trajectory control of the UAV in the wireless powered IoT networks. An IoT system with devices that has dynamic and uncertain data transmission requirements and charging requirements and a UAV served as a data collector and charger is developed. Our target is to minimize the average data buffer length, maximize the residual battery level of the system and avoid devices' data overflow and running out of battery. To handle this complex and challenging problem, a DQN-based algorithm is provided to achieve the flight control policy for the UAV. Numerical results prove the validity and convergence of our algorithm. Besides, the influence of the degrees of freedom of the action space to the performance of policy has been studied.

References

[1] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for Smart Cities," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, 2014.

[2] C. W. Tsai, C. F. Lai, M. C. Chiang, and L. T. Yang, "Data Mining for Internet of Things: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 77–97, 2014.

[3] M. A. Abd-Elmagid, A. Ferdowsi, H. S. Dhillon, and W. Saad, "Deep Reinforcement Learning for Minimizing Age-of-Information in UAV-assisted Networks," *CoRR*, vol. abs/1905.02993, 2019. [Online]. Available: http://arxiv.org/abs/1905.02993

[4] Z. Dawy, W. Saad, A. Ghosh, J. G. Andrews, and E. Yaacoub, "Toward Massive Machine Type Cellular Communications," *IEEE Wireless Communications*, vol. PP, no. 99, pp. 2–10, 2015.

[5] L. R. Varshney, "Transporting information and energy simultaneously," in *2008 IEEE International Symposium on Information Theory*, July 2008, pp. 1612–1616.

[6] S. Bi, C. K. Ho, and Z. Rui, "Wireless Powered Communication: Opportunities and Challenges," *IEEE Communications Magazine*, vol. 53, no. 4, pp. 117–125, 2014.

[7] S. Bi, Y. Zeng, and R. Zhang, "Wireless powered communication networks: an overview," *IEEE Wireless Communications*, vol. 23, no. 2, pp. 10–18, April 2016.

[8] M. Mozaffari, W. Saad, M. Bennis, Y. Nam, and M. Debbah, "A Tutorial on UAVs for Wireless Networks: Applications, Challenges, and Open Problems," *IEEE Communications Surveys Tutorials*, vol. 21, no. 3, pp. 2334–2360, thirdquarter 2019.

[9] F. Cheng, S. Zhang, Z. Li, Y. Chen, N. Zhao, R. Yu, and V. C. M. Leung, "UAV Trajectory Optimization for Data Offloading at the Edge of Multiple Cells," *IEEE Transactions on Vehicular Technology*, vol. 67, pp. 6732–6736, 2018.

[10] W. Feng, J. Tang, Y. Yu, J. Song, N. Zhao, G. Chen, K.-K. Wong, and J. Chambers, "UAV-Enabled SWIPT in IoT Networks for Emergency Communications," *IEEE Wireless Communications*, 2020.

[11] S. Yang, Y. Deng, X. Tang, Y. Ding, and J. Zhou, "Energy Efficiency Optimization for UAV-assisted Backscatter Communications," *IEEE Communications Letters*, p. 1–1, 2019. [Online]. Available: http://dx.doi.org/10.1109/lcomm.2019.2931900

[12] L. Xie, J. Xu, and Y. Zeng, "Common Throughput Maximization for UAV-Enabled Interference Channel with Wireless Powered Communications," 2019.

[13] R. Sutton and A. Barto, *Reinforcement Learning:An Introduction*. MIT Press, 1998.

[14] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[15] X. Hu, S. Liu, R. Chen, W. Wang, and C. Wang, "A Deep Reinforcement Learning-Based Framework for Dynamic Resource Allocation in Multibeam Satellite Systems," *IEEE Communications Letters*, vol. 22, no. 8, pp. 1612–1615, Aug 2018.

[16] J. Wang, C. Xu, Y. Huangfu, R. Li, Y. Ge, and J. Wang, "Deep Reinforcement Learning for Scheduling in Cellular Networks," *CoRR*, vol. abs/1905.05914, 2019. [Online]. Available: http://arxiv.org/abs/1905.05914

[17] 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 Journal on Selected Areas in Communications*, vol. 36, no. 9, pp. 2059–2070, Sep. 2018.

[18] C. H. Liu, X. Ma, X. Gao, and J. Tang, "Distributed Energy-Efficient Multi-UAV Navigation for Long-Term Communication Coverage by Deep Reinforcement Learning," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2019.

[19] B. Zhang, C. H. Liu, J. Tang, Z. Xu, J. Ma, and W. Wang, "Learning-Based Energy-Efficient Data Collection by Unmanned Vehicles in Smart Cities," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1666–1676, April 2018.

[20] S. Wan, J. Lu, P. Fan, and K. B. Letaief, "Towards Big data processing in IoT: Path Planning and Resource Management of UAV Base Stations in Mobile-Edge Computing System," *CoRR*, vol. abs/1906.05023, 2019. [Online]. Available: http://arxiv.org/abs/1906.05023

[21] M. A. Abd-Elmagid, H. S. Dhillon, and N. Pappas, "A Reinforcement Learning Framework for Optimizing Age-of-Information in RF-powered Communication Systems," 2019.

[22] C. Watkins, J. Christopher, and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[23] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. A. Riedmiller, "Playing atari with deep reinforcement learning," *CoRR*, vol. abs/1312.5602, 2013. [Online]. Available: http://arxiv.org/abs/1312.5602