3D UAV Trajectory and Data Collection Optimisation via Deep Reinforcement Learning

Khoi Khac Nguyen, *Student Member, IEEE,* Trung Q. Duong, *Fellow, IEEE,* Tan Do-Duy, *Member, IEEE,* Holger Claussen, *Fellow, IEEE,* and Lajos Hanzo, *Fellow, IEEE,*

Abstract—**Unmanned aerial vehicles (UAVs) are now beginning to be deployed for enhancing the network performance and coverage in wireless communication. However, due to the limitation of their on-board power and flight time, it is challenging to obtain an optimal resource allocation scheme for the UAV-assisted Internet of Things (IoT). In this paper, we design a new UAV-assisted IoT system relying on the shortest flight path of the UAVs while maximising the amount of data collected from IoT devices. Then, a deep reinforcement learning-based technique is conceived for finding the optimal trajectory and throughput in a specific coverage area. After training, the UAV has the ability to autonomously collect all the data from user nodes at a significant total sum-rate improvement while minimising the associated resources used. Numerical results are provided to highlight how our techniques strike a balance between the throughput attained, trajectory, and the time spent. More explicitly, we characterise the attainable performance in terms of the UAV trajectory, the expected reward and the total sum-rate.**

Keywords- UAV-assisted wireless network, trajectory, data collection, and deep reinforcement learning.

I. Introduction

Given the agility of unmanned aerial vehicles (UAVs), they are capable of supporting compelling applications and are beginning to be deployed more broadly. Recently, the UK and Chile authorities proposed to deliver medical support and other essential supplies by using UAVs to vulnerable people in response to Covid-19 [1], [2]. In [3], the authors used UAVs for image collection and high-resolution topography exploration. However, given the several limitations of on-board power level and the ability to adapt to changes in the environment, UAVs may not be fully autonomous and can only operate for short flight-durations, unless remote laser-charging is used [4]. Moreover, due to some challenging tasks such as topographic surveying, data collection or obstacle avoidance, the existing UAV technologies cannot operate in an optimal manner.

Wireless networks supported by UAVs constitute a promising technology for enhancing the network performance [5]. The applications of UAVs in wireless networks span across diverse research fields, such as wireless sensor networks (WSNs) [6], caching [7], heterogeneous cellular networks [8], massive multiple-input multiple-output (MIMO) [9], disaster communications [10], [11] and device-to-device communications (D2D) [12]. For example, in [13], UAVs were deployed to provide network coverage for people in remote areas and disaster zones. UAVs were also used for collecting data in a WSN [6]. Nevertheless, the benefits of UAV-aided wireless communication are critically dependent on the limited on-board power level. Thus, the resource allocation of UAV-aided wireless networks plays a pivotal role in approaching the optimal performance. Yet, the existing contributions typically assume having static environment [10], [11], [14] and often ignore the stringent flight time constraints in real-life applications [6], [8], [15].

Machine learning has recently been proposed for the intelligent support of UAVs and other devices in the network [9], [16]–[24]. Reinforcement learning (RL) is capable of searching for an optimal policy by trial-and-error learning. However, it is challenging for model-free RL algorithms, such as Q-learning to obtain an optimal strategy, while considering a large state and action space. Fortunately, with the emerging neural networks, the sophisticated combination of RL and deep learning, namely deep reinforcement learning (DRL) is eminently suitable for solving high-dimensional problems. Hence, DRL algorithms have been widely applied in fields such as robotics [25], business management [26] and gaming [27]. Recently, DRL has also become popular in solving diverse problems in wireless networks thanks to their decision-making ability and flexible interaction with the environment [7], [9], [18]–[24], [28]–[30]. For example, DRL was used for solving problems in the areas of resource allocation [18], [19], [29], navigation [9], [31] and interference management [22].

A. Related Contributions

UAV-aided wireless networks have also been used for machine-to-machine communications [32] and D2D scenarios in 5G [14], [33], but the associated resource allocation problems remain challenging in real-life applications. Several techniques have been developed for solving resource allocation problems [18], [19], [31], [34]–[36]. In [34], the authors have conceived a multi-beam UAV communications and a cooperative interference cancellation scheme for maximising

Khoi Khac Nguyen and Trung Q. Duong are with the School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, Belfast BT7 1NN, U.K. (e-mail: {knguyen02,trung.q.duong}@qub.ac.uk).

Tan Do-Duy is with Ho Chi Minh City University of Technology and Education, Vietnam (e-mail: tandd@hcmute.edu.vn).

Holger Claussen is with Tyndall National Institute, Dublin, Ireland (e-mail: holger.claussen@tyndall.ie).

Lajos Hanzo is with the School of Electronics and Computer Science, University of Southampton, Southampton, SO17 1BJ, U.K. (e-mail: lh@ecs.soton.ac.uk).

The work of T. Q. Duong was supported by the U.K. Royal Academy of Engineering (RAEng) under the RAEng Research Chair and Senior Research Fellowship scheme Grant RCSRF2021\11\41. L. Hanzo would like to acknowledge the financial support of the Engineering and Physical Sciences Research Council projects EP/P034284/1 and EP/P003990/1 (COALESCE) as well as of the European Research Council's Advanced Fellow Grant QuantCom (Grant No. 789028)

the uplink sum-rate received from multiple UAVs by the base stations (BS) on the ground. The UAVs were deployed as access points to serve several ground users in [35]. Then, the authors proposed successive convex programming for maximising the minimum uplink rate gleaned from all the ground users. In [31], the authors characterised the trade-off between the ground terminal transmission power and the specific UAV trajectory both in a straight and in a circular trajectory.

The issues of data collection, energy minimisation, and path planning have been considered in [23], [32], [37]–[45]. In [38], the authors minimised the energy consumption of the data collection task considered by jointly optimising the sensor nodes' wakeup schedule and the UAV trajectory. The authors of [39] proposed an efficient algorithm for joint trajectory and power allocation optimisation in UAV-assisted networks to maximise the sum-rate during a specific length of time. A pair of near-optimal approaches for optimal trajectory was proposed for a given UAV power allocation and power allocation optimisation for a given trajectory. In [32], the authors introduced a communication framework for UAV-to-UAV communication under the constraints of the UAV's flight speed, location uncertainty and communication throughput. Then, a path planning algorithm was proposed for minimising the associated completion time task while balancing the performance by computational complexity trade-off. However, these techniques mostly operate in offline modes and may impose excessive delay on the system. It is crucial to improve the decision-making time for meeting the stringent requirements of UAV-assisted wireless networks.

Again, machine learning has been recognised as a powerful tool of solving the high-dynamic trajectory and resource allocation problems in wireless networks. In [36], the authors proposed a model based on the classic k-means algorithm for grouping the users into clusters and assigned a dedicated UAV to serve each cluster. By relying on their decision-making ability, DRL algorithms have been used for lending each node some degree of autonomy [7], [18]–[21], [28], [29], [46]. In [28], an optimal DRL-based channel access strategy to maximise the sum rate and α-fairness was considered. In [18], [19], we deployed DRL techniques for enhancing the energy-efficiency of D2D communications. In [21], the authors characterised the DQL algorithm for minimising the data packet loss of UAV-assisted power transfer and data collection systems. As a further advance, caching problems were considered in [7] to maximise the cache success hit rate and to minimise the transmission delay. The authors designed both a centralised and a decentralised system model and used an actor-critic algorithm to find the optimal policy.

DRL algorithms have also been applied for path planning in UAV-assisted wireless communications [9], [22]–[24], [30], [47]. In [22], the authors proposed a DRL algorithm based on the echo state network of [48] for finding the flight path, transmission power and associated cell in UAV-powered wireless networks. The so-called deterministic policy gradient algorithm of [49] was invoked for UAV-assisted cellular networks in [30]. The UAV's trajectory was designed for maximising the uplink sum-rate attained without the knowledge of the user location and the transmit power. Moreover, in [9], the authors used the DQL algorithm for the UAV's navigation based on the received signal strengths estimated by a massive MIMO scheme. In [23], Q-learning was used for controlling the movement of multiple UAVs in a pair of scenarios, namely for static user locations and for dynamic user locations under a random walk model. However, the aforementioned contributions have not addressed the joint trajectory and data collection optimisation of UAV-assisted networks, which is a difficult research challenge. Furthermore, these existing works mostly neglected interference, 3D trajectory and dynamic environment.

B. Contributions and Organisation

A novel DRL-aided UAV-assisted system is conceived for finding the optimal UAV path for maximising the joint reward function based on the shortest flight distance and the uplink transmission rate. We boldly and explicitly contrast our proposed solution to the state-of-the-art in Table I. Our main contributions are further summarised as follows:

- In our UAV-aided system, the maximum amount of data is collected from the users with the shortest distance travelled.
- Our UAV-aided system is specifically designed for tackling the stringent constraints owing to the position of the destination, the UAV's limited flight time and the communication link's realistic constraints. The UAV's objective is to find the optimal trajectory for maximising the total network throughput, while minimising its distance travelled.
- Explicitly, these challenges are tackled by conceiving bespoke DRL techniques for solving the above problem. To elaborate, the area is divided into a grid to enable fast convergence. Following its training, the UAV can have the autonomy to make a decision concerning its next action at each position in the area, hence eliminating the need for human navigation. This makes our UAV-aided system more reliable, practical and optimises the resource requirements.
- A pair of scenarios are considered relying either on three or five clusters for quantifying the efficiency of our novel DRL techniques in terms of both the sum-rate, the trajectory and the associated time. A convincing 3D trajectory visualisation is also provided.
- Finally, but most importantly, it is demonstrated that our DRL techniques approach the performance of the optimal "genie-solution" associated with the perfect knowledge of the environment.

Although the existing DRL algorithms have been well exploited in wireless networks, it is challenging to apply to current scenarios due to stringent constraints of the considered system, such as UAV's flying time, transmission distance, and mobile users. As with the DQL and dueling DQL algorithm, we discretise the flying path into grid and the UAV only needs to decide the action in a finite action space. With the finite state and action space, the neural networks can be easily trained and deployed for online phase. With other

TABLE I
A COMPARISON WITH EXISTING LITERATURE

	[37]	[6]	[21]	[23]	[40]	[9]	[41]	[47]	[42]	[43]	Our work
3D trajectory				✓			✓			✓	✓
Sum-rate maximisation	✓		✓	✓				✓	✓	✓	✓
Time minimisation		✓	✓				✓		✓		✓
Dynamic environment				✓	✓			✓			✓
Unknown users											✓
Reinforcement learning			✓	✓		✓		✓			✓
Deep neural networks			✓			✓		✓			✓

existing RL algorithm, we have tried and found out that some of them are not effective in solving our proposed problem. Meanwhile, the continuous solver RL algorithms, e.g., deep deterministic policy gradient (DDPG) and proximal policy optimisation (PPO), are not suitable and so challenging for the trade-off problem. Therefore, in this paper, we propose the DQL and dueling DQL algorithm to obtain the optimal trade-off in total achievable sum-rate and trajectory. As such, we can transferred a real-life application into a digital environment for optimisation and solve it efficiently.

The rest of our paper is organised as follows. In Section II, we describe our data collection system model and the problem formulation of IoT networks relying on UAVs. Then, the mathematical background of the DRL algorithms is presented in Section III. Deep Q-learning (DQL) is employed for finding the best trajectory and for solving our data collection problem in Section IV. Furthermore, we use the dueling DQL algorithm of [50] for improving the system performance and convergence speed in Section V. Next, we characterise the efficiency of the DRL techniques in Section VI. Finally, in Section VII, we summarise our findings and discuss our future research.

II. System Model and Problem Formulation

Consider a system consisting of a single UAV and M groups of users, as shown in Fig. 1, where the UAV relying on a single antenna visits all clusters to cover all the users. The 3D coordinate of the UAV at time step t is defined as $X^t = (x_0^t, y_0^t, H_0^t)$. Each cluster consists of K users, which are unknown and distributed randomly within the coverage radius of C. The users are moving following the random walk model with the maximum velocity v. The position of the kth user in the mth cluster at time step t is defined as $X_{m,k}^t = (x_{m,k}^t, y_{m,k}^t)$. The UAV's objective is to find the best trajectory while covering all the users and to reach the dock upon completing its mission.

A. Observation model

The distance from the UAV to user k in cluster m at time step t is given by:

$$d_{m,k}^t = \sqrt{(x_0^t - x_{m,k}^t)^2 + (y_0^t - y_{m,k}^t)^2 + {H_0^t}^2}. \tag{1}$$

We assume that the communication channels between the UAV and users are dominated by line-of-sight (LoS) links; thus the channel between the UAV and the kth user in the

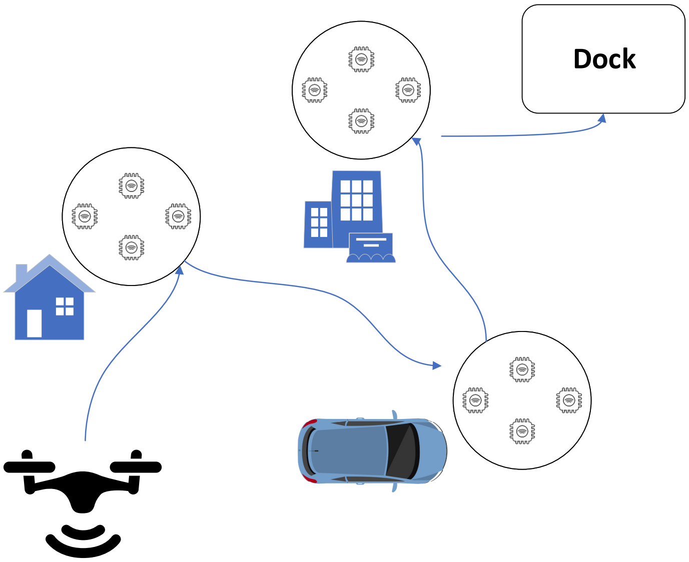


Fig. 1. System model of UAV-aided IoT communications.

mth cluster at time step t follows the free-space path loss model, which is represented as

$$\begin{aligned} h_{m,k}^t &= \beta_0 {d_{m,k}^t}^{-2} \\ &= \frac{\beta_0}{(x_0^t - x_{m,k}^t)^2 + (y_0 - y_{m,k}^t)^2 + {H_0^t}^2}, \end{aligned} \tag{2}$$

where the channel's power gain at a reference distance of $d = 1m$ is denoted by β_0.

The achievable throughput from the kth user in the mth cluster to the UAV at time t if the user satisfies the distance constraint is defined as follows:

$$R_{m,k}^t = \\ B \log_2 \left(1 + \frac{p_{m,k}^t h_{m,k}^t}{\sum_{i \neq m}^M \sum_j^K p_{i,j}^t h_{i,j}^t + \sum_{u \neq k}^K p_{m,u}^t h_{m,u}^t + \alpha^2} \right), \forall m, k, \tag{3}$$

where B and α^2 are the bandwidth and the noise power, respectively; $p_{m,k}$ is the transmit power at the kth user in the mth cluster. Then the total sum-rate over the T time step from the kth user in cluster m to the UAV is given by:

$$R_{m,k} = \int_0^T R_{m,k}^t dt, \forall m, k. \tag{4}$$

B. Game formulation

Both the current location and the action taken jointly influence the rewards obtained by the UAV; thus the trial-and-error based learning task of the UAV satisfies the Markov property. We formulate the associated Markov decision process (MDP) [51] as a 4 tuple $< \mathcal{S}, \mathcal{A}, \mathcal{P}_{ss'}, \mathcal{R} >$, where $\mathcal{S}$ is the state space of the UAV, $\mathcal{A}$ is the action space; $\mathcal{R}$ is the expected reward of the UAV and $\mathcal{P}_{ss'}$ is the probability of transition from state s to state s', where we have $s' = s^{t+1}|s = s^t$. Through learning, the UAV can find the optimal policy $\pi^* : \mathcal{S} \to \mathcal{A}$ for maximising the reward $\mathcal{R}$. As the definition of RL, the UAV does not have any knowledge about the environment. We transfer a real-life application of the data collection in the UAV-assisted IoT networks into a digital form. Thus, the UAV only has local information and the state is defined by the position of UAV. We have also discretised the state and action space for learning. More particularly, we formulate the trajectory and data collection game of UAV-aided IoT networks as follows:

- *Agent*: The UAV acts like an agent interacting with the environment to find the peak of the reward.
- *State space*: We define the state space by the position of UAV as
$$\mathcal{S} = \{x, y, H\}. \quad (5)$$
At time step t, the state of the UAV is defined as $s^t = (x^t, y^t, H^t)$.
- *Action space*: The UAV at state s^t can choose an action a^t of the action space by following the policy at time-step t. By dividing the area into a grid, we can define the action space as follows:
$$\begin{aligned} \mathcal{A} = \{&\text{left,right, forward, backward,} \\ &\text{upward, downward, hover}\}. \end{aligned} \quad (6)$$
The UAV moves in the environment and begins collecting information when the users are in the coverage of the UAV. When the UAV has sufficient information $R_{m,k} \geq r_{min}$ from the kth user in the mth cluster, that user will be marked as collected in this mission and may not be visited by the UAV again.
- *Reward function*: In joint trajectory and data collection optimisation, we design the reward function to be dependent on both the total sum-rate of ground users associated with the UAV plus the reward gleaned when the UAV completes one route, which is formulated as follows:
$$R = \frac{\beta}{MK} \left(\sum_{m}^{M} \sum_{k}^{K} P(m,k) R_{m,k} \right) + \zeta R_{plus}, \quad (7)$$
where β and ζ are positive variables that represent the trade-off between the network's sum-rate and UAV's movement, which will be described in the sequel. Here, $P(m,k) \in \{0,1\}$ indicates whether or not user k of cluster m is associated with the UAV; R_{plus} is the acquired reward when the UAV completes a mission by reaching the final destination. On the other hand, the term $\frac{\sum_m^M \sum_k^K P(m,k) R_{m,k}}{MK}$ defines the average throughput of all users.
- *Probability*: We define $\mathcal{P}_{s^t s^{t+1}}(a^t, \pi)$ as the probability of transition from state s^t to state s^{t+1} by taking the action a^t under the policy π.

At each time step t, the UAV chooses the action a^t based on its local information to obtain the reward r^t under the policy π. Then the UAV moves to the next state s^{t+1} by taking the action a^t and starts collecting information from the users if any available node in the network satisfies the distance constraint. Meanwhile, the users in clusters also move to new positions following the random walk model with velocity v. Again, we use the DRL techniques to find the optimal policy π^* for the UAV to maximise the reward attained (7). Following the policy π, the UAV forms a chain of actions $(a^0, a^1, \ldots, a^t, \ldots, a^{final})$ to reach the landing dock.

Our target is to maximise the reward expected by the UAV upon completing a single mission during which the UAV flies from the initial position over the clusters and lands at the destination. Thus, we design the trajectory reward R_{plus} when the UAV reaches the destination in two different ways. Firstly, the binary reward function is defined as follows:

$$R_{plus} = \begin{cases} 1 & , \quad X_{final} \in X_{target} \\ 0 & , \quad \text{otherwise.} \end{cases}, \quad (8)$$

where X_{final} and X_{target} are the final position of UAV and the destination, respectively. However, the UAV has to move a long distance to reach the final destination. It may also be trapped in a zone and cannot complete the mission. These situations lead to increased energy consumption and reduced convergence. Thus, we consider the value of R^t_{plus} in a different form by calculating the horizontal distance between the UAV and the final destination at time step t, yielding:

$$R^t_{plus} = \begin{cases} 1 & , \quad X_{final} \in X_{target} \\ \Big(\exp(d_{targ})\Big)^{-1} & , \quad \text{otherwise,} \end{cases} \quad (9)$$

where $d_{targ} = \sqrt{(x_{target} - x^t_0)^2 + (y_{target} - y^t_0)^2}$ is the distance from the UAV to the landing dock.

When we design the reward function as in (9), the UAV is motivated to move ahead to reach the final destination. However, one of the disadvantages is that the UAV only moves forward. Thus, the UAV is unable to attain the best performance in terms of its total sum-rate in some environmental settings. We compare the performance of the two trajectory reward function definitions in Section VI to evaluate the pros and cons of each approach.

In our work, we optimise the 3D trajectory of the UAV and data collection for the IoT network. Particularly, we have design the reward function by a trade-off game between the achievable sum-rate and the trajectory. Denote the flying path of the UAV from the initial point to final position by $X = (X_0, X_1, \ldots, X_{final})$, the agent needs to learn by iterating with the environment to find an optimal X. We have defined a trade-off value β and ζ to make our approach more adaptive and flexible. By modifying the value of β/ζ , the UAV adapts to several scenarios: a) fast deployment for emergency services, b) maximising the total sum-rate, and c) maximising the number of connections between the UAV and users. Depending on the specific problems, we can adjust the value of the trade-off parameters β, ζ to achieve the best performance. Thus, the game formulation is defined as follows:

$$\begin{aligned}\max R = \quad & \frac{\beta}{MK}\left(\sum_{m}^{M}\sum_{k}^{K} P(m,k) R_{m,k}\right) + \zeta R_{plus}, \\ s.t. \quad & X_{final} = X_{target}, \\ & d_{m,k} \leq d_{cons}, \\ & R_{m,k} \geq r_{min}, \\ & P(m,k) \in \{0,1\}, \\ & T \leq T_{cons} \\ & \beta \geq 0,\ \zeta \geq 0,\end{aligned} \tag{10}$$

where T and T_{cons} are the number of steps that the UAV takes in a single mission and the maximum number of UAV's steps given its limited power, respectively. The term $X_{final} = X_{target}$ denotes the completed flying route when the final position of the UAV belongs to the destination zone. We have designed the reward function following this constraint with two functions: binary reward function in (8) and exponential reward function in (9). The term $d_{m,k} \leq d_{cons}, R_{m,k} \geq r_{min}, P(m,k) \in \{0,1\}$ denote the communication constraint. Particularly, the distance constraint $d_{m,k} \leq d_{cons}$ indicates that the served (m,k)-user has a satisfying distance to the UAV. $P(m,k) \in \{0,1\}$ indicates whether or not user k of cluster m is associated with the UAV. $R_{m,k} \geq r_{min}$ denotes the minimum information collected during the flying path. All the constraints are integrated into the reward functions in the RL algorithm. The term $T \leq T_{cons}$ denotes the constraint about the flying time. Consider the maximum flying time is T_{cons}, the UAV needs to complete a route by reaching the destination zone before T_{cons}. If the UAV can not complete a route before T_{cons}, the $R_{plus} = 0$ as we defined in (8) and (9). We have the trade-off value in reward function $\beta \geq 0,\ \zeta \geq 0$. Those stringent constraints, such as the transmission distance, position and flight time make the optimisation problem more challenging. Thus, we propose DRL techniques for the UAV in order to attain optimal performance.

III. Preliminaries

In this section, we introduce the fundamental concept of Q-learning, where the so-called value function is defined by a reward of the UAV at state s^t as follows:

$$V(s,\pi) = \mathbb{E}\left[\sum_{t}^{T} \gamma \mathcal{R}^t(s^t,\pi) | s_0 = s\right], \tag{11}$$

where $\mathbb{E}[.]$ represents an average of the number of samples and $0 \leq \gamma \leq 1$ denotes the discount factor. In a finite game, there is always an optimal policy π^* that satisfies the Bellman optimality equation [52]

$$\begin{aligned} V^*(s,\pi) &= V(s,\pi^*) \\ &= \max_{a \in \mathcal{A}}\left[\mathbb{E}\left[\mathcal{R}^t(s^t,\pi^*)\right] + \gamma \sum_{s' \in S} \mathcal{P}_{ss'}(a,\pi^*) V(s',\pi^*)\right]. \end{aligned} \tag{12}$$

The action-value function is obtained, when the agent at state s^t takes action a^t and receives the reward r^t under the agent policy π. The optimal Q-value can be formulated as:

$$Q^*(s,a,\pi) = \mathbb{E}\left[\mathcal{R}^t(s^t,\pi^*)\right] + \gamma \sum_{s' \in S} \mathcal{P}_{ss'}(a,\pi^*) V(s',\pi^*). \tag{13}$$

The optimal policy π^* can be obtained from $Q^*(s,a,\pi)$ as follows:

$$V^*(s,\pi) = \max_{a \in \mathcal{A}} Q(s,a,\pi). \tag{14}$$

From (13) and (14), we have

$$\begin{aligned} Q^*(s,a,\pi) &= \mathbb{E}\left[\mathcal{R}^t(s^t,\pi^*)\right] + \gamma \sum_{s' \in S} \mathcal{P}_{ss'}(a,\pi^*) \max_{a' \in \mathcal{A}} Q(s',a',\pi), \\ &= \mathbb{E}\left[\mathcal{R}^t(s^t,\pi^*) + \gamma \max_{a' \in \mathcal{A}} Q(s',a',\pi)\right], \end{aligned} \tag{15}$$

where the agent takes the action $a' = a^{t+1}$ at state s^{t+1}.

Through learning, the Q-value is updated based on the available information as follows:

$$\begin{aligned} Q(s,a,\pi) = \ & Q(s,a,\pi) \\ & + \alpha\left[\mathcal{R}^t(s^t,\pi^*) + \gamma \max_{a' \in \mathcal{A}} Q(s',a',\pi) - Q(s,a,\pi)\right], \end{aligned} \tag{16}$$

where α denotes the updated parameter of the Q-value function.

In RL algorithms, it is challenging to balance the *exploration* and *exploitation* for appropriately selecting the action. The most common approach relies on the ϵ-greedy policy for the action selection mechanism as follows:

$$a = \begin{cases} \arg\max Q(s,a,\pi) & \text{with} \quad \epsilon \\ \quad \text{randomly} & \text{if} \quad 1-\epsilon. \end{cases} \tag{17}$$

Upon assuming that each episode lasts T steps, the action at time step t is a^t that is selected by following the ϵ-greedy policy as in (17). The UAV at state s^t communicates with the user nodes from the ground if the distance constraint of $d_{m,k} \leq d_{cons}$ is satisfied. Following the information transmission phase, the user nodes are marked as collected users and may not be revisited later during that mission. Then, after obtaining the immediate reward $r(s^t,a^t)$ the agent at state s^t takes action a^t to move to state s^{t+1} as well as to update the Q-value function in (16). Each episode ends when the UAV reaches the final destination and the flight duration constraint is satisfied.

IV. An Effective Deep Reinforcement Learning Approach for UAV-assisted IoT Networks

In this section, we conceive the DQL algorithm for trajectory and data collection optimisation in UAV-aided IoT networks. However, Q-learning technique typically falters for large state and action spaces due to its excessive Q-table size. Thus, instead of applying the Q-table in Q-learning, we use deep neural networks to represent the relationship between the action and state space. Furthermore, we employ a pair of techniques for stabilising the neural network's performance in our DQL algorithm as follows:

- *Experience replay buffer*: Instead of using current experience, we use a so-called replay buffer $\mathcal{B}$ to store the transitions (s, a, r, s') for supporting the neural network in overcoming any potential instability. When the buffer $\mathcal{B}$ is filled with the transitions, we randomly select a mini-batch of K samples for training the networks. The finite buffer size of $\mathcal{B}$ allows it to be always up-to-date, and the neural networks learn from the new samples.
- *Target networks*: If we use the same network to calculate the state-action value Q and the target network, the network can be shifted dramatically in the training phase. Thus, we employ a target network Q' for the target value estimator. After a number of iterations, the parameters of the target network Q' will be updated by the network Q.

The UAVs start from the initial position and interact with the environment to find the proper action in each state. The agent chooses the action a^t following current policy at state s^t. By execute the action a^t, the agent receives the response from the environment with reward r^t and new state s^{t+1}. After each time step, the UAVs have new positions and the environment is changed with moving users. The obtained transitions are stored into a finite memory buffer and used for training the neural networks.

The neural network parameters are updated by minimising the loss function defined as follows:

$$\mathbb{L}(\theta) = \mathbb{E}_{s,a,r,s'}\left[\left(y^{DQL} - Q(s,a;\theta)\right)^2\right], \qquad (20)$$

where θ is a parameter of the network Q and we have

$$y = \begin{cases} r^t & \text{if terminated at } s^{t+1} \\ r^t + \gamma \max_{a' \in \mathcal{A}} Q'(s', a'; \theta') & \text{otherwise.} \end{cases} \qquad (21)$$

The details of the DQL approach in our joint trajectory and data collection trade-off game designed for UAV-aided IoT networks are presented in Algorithm 1 where L denotes the number of episode. Moreover, in this paper, we design the reward obtained in each step to assume one of two different forms and compare them in our simulation results. Firstly, we calculate the difference between the current and the previous reward of the UAV as follows:

$$r_1^t(s^t, a^t) = r^t(s^t, a^t) - r^{t-1}(s^{t-1}, a^{t-1}). \qquad (22)$$

Secondly, we design the total episode reward as the accumulation of all immediate rewards of each step within one episode as

$$r_2^t(s^t, a^t) = \sum_{i=0}^{t} r_1^t(s^t, a^t). \qquad (23)$$

Algorithm 1 The deep Q-learning algorithm for trajectory and data collection optimisation in UAV-aided IoT networks.

```
Initialise the network $Q$ and the target network $Q'$ with
    the random parameters $\theta$ and $\theta'$, respectively
Initialise the replay memory pool $\mathcal{B}$
for episode = $1, \ldots, L$ do
    Receive initial observation state $s^0$
    while $X_{final} \notin X_{target}$ or $T \leq T_{cons}$ do
        Obtain the action $a^t$ of the UAV according to the
            $\epsilon$-greedy mechanism (17)
        Execute the action $a^t$ and estimate the reward $r^t$
            according to (7)
        Observe the next state $s^{t+1}$
        Store the transition $(s^t, a^t, r^t, s^{t+1})$ in the replay
            buffer $\mathcal{B}$
        Randomly select a mini-batch of $K$ transitions
            $(s^k, a^k, r^k, s^{k+1})$ from $\mathcal{B}$
        Update the network parameters using gradient de-
            scent to minimise the loss
            $\mathbb{L}(\theta) = \mathbb{E}_{s,a,r,s'}\left[\left(y^{DQL} - Q(s,a;\theta)\right)^2\right]$,   (18)
        The gradient update is
            $\nabla_\theta \mathbb{L}(\theta) = \mathbb{E}_{s,a,r,s'}\left[\left(y^{DQL} - Q(s,a;\theta)\right)\nabla_\theta Q(s,a;\theta)\right]$,   (19)
        Update the state $s^t = s^{t+1}$
        Update the target network parameters after a number
            of iterations as $\theta' = \theta$
    end while
end for
```

V. Deep Reinforcement Learning Approach for UAV-assisted IoT networks: A Dueling Deep Q-learning Approach

According to Wang *et. al.* [50], the standard Q-learning algorithm often falters due to the over-supervision of all the state-action pairs. On the other hand, it is unnecessary to estimate the value of each action choice in a particular state. For example, in our environment setting, the UAV has to consider moving either to the left or to the right when it hits the boundaries. Thus, we can improve the convergence speed by avoiding visiting all state-action pairs. Instead of using Q-value function of the conventional DQL algorithm, the dueling neural network of [50] is introduced for improving the convergence rate and stability. The so-called advantage function $A(s, a) = Q(s, a) - V(s)$ related both to the value function and to the Q-value function describes the importance of each action related to each state.

The idea of a dueling deep network is based on a combination of two streams of the value function and the advantage function used for estimating the single output Q-function. One of the streams of a fully-connected layer estimates the value function $V(s; \theta_V)$, while the other stream outputs a vector $A(s, a; \theta_A)$, where θ_A and θ_V represent the parameters of the two networks. The Q-function can be obtained by combining the two streams' outputs as follows:

$$Q(s, a; \theta, \theta_A, \theta_V) = V(s; \theta_V) + A(s, a; \theta_A). \qquad (27)$$

Equation (27) applies to all (s, a) instances; thus, we have to replicate the scalar $V(s; \theta_V)$, $|\mathcal{A}|$ times to form a matrix. However, $Q(s, a; \theta, \theta_A, \theta_V)$ is a parameterised estimator of

Algorithm 2 The dueling deep Q-learning algorithm for trajectory and data collection optimisation in UAV-aided IoT networks.

```
Initialise the network $Q$ and the target network $Q'$ with the random parameters, $\theta$ and $\theta'$, respectively
Initialise the replay memory pool $\mathcal{B}$
for episode = $1, \dots, L$ do
    Receive the initial observation state $s^0$
    while $X_{final} \notin X_{target}$ or $T \leq T_{cons}$ do
        Obtain the action $a^t$ of the UAV according to the $\epsilon$-greedy mechanism (17)
        Execute the action $a^t$ and estimate the reward $r^t$ according to (7)
        Observe the next state $s^{t+1}$
        Store the transition $(s^t, a^t, r^t, s^{t+1})$ in the replay buffer $\mathcal{B}$
        Randomly select a mini-batch of $K$ transitions $(s^k, a^k, r^k, s^{k+1})$ from $\mathcal{B}$
        Estimate the Q-value function by combining the two streams as follows:
            $Q(s, a; \theta, \theta_A, \theta_V) = V(s; \theta_V) + \left(A(s, a; \theta_A) - \frac{1}{|\mathcal{A}|} \sum_{a'} A(s, a'; \theta_A)\right).$    (24)
        Update the network parameters using gradient descent to minimise the loss
            $\mathbb{L}(\theta) = \mathbb{E}_{s,a,r,s'} \left[ \left( y^{DuelingDQL} - Q(s, a; \theta, \theta_A, \theta_V) \right)^2 \right],$    (25)
        where
            $y^{DuelingDQL} = r^t + \gamma \max_{a' \in \mathcal{A}} Q'(s', a'; \theta', \theta_A, \theta_V).$    (26)
        Update the state $s^t = s^{t+1}$
        Update the target network parameters after a number of iterations as $\theta' = \theta$
    end while
end for
```

the true Q-function; thus, we cannot uniquely recover the value function V and the advantage function A. Therefore, (27) results in poor practical performances when used directly. To address this problem, we can map the advantage function estimator to have no advantage at the chosen action by combining the two streams as follows:

$$Q(s, a; \theta, \theta_A, \theta_V) = V(s; \theta_V) + \left(A(s, a; \theta_A) - \max_{a' \in |\mathcal{A}|} A(s, a'; \theta_A)\right). \tag{28}$$

Intuitively, for $a^* = \arg\max_{a' \in \mathcal{A}} Q(s, a'; \theta, \theta_A, \theta_V) = \arg\max_{a' \in \mathcal{A}} A(s, a'; \theta_A)$, we have $Q(s, a^*; \theta, \theta_A, \theta_V) = V(s; \theta_V)$. Hence, the stream $V(s; \theta_V)$ estimates the value function and the other streams is the advantage function estimator. We can transform (28) using an average formulation instead of the *max* operator as follows:

$$Q(s, a; \theta, \theta_A, \theta_V) = V(s; \theta_V) + \left(A(s, a; \theta_A) - \frac{1}{|\mathcal{A}|} \sum_{a'} A(s, a'; \theta_A)\right). \tag{29}$$

Now, we can solve the problem of identifiability by subtracting the mean as in (29). Based on (29), we propose a dueling DQL algorithm for our joint trajectory and data collection problem in UAV-assisted IoT networks relying on Alg. 2. Note that estimating $V(s; \theta_V)$ and $A(s, a; \theta_A)$ does not require any extra supervision and they will be computed automatically.

TABLE II
SIMULATION PARAMETERS.

Parameters	Value
Bandwidth (W)	1 MHz
UAV transmission power	5 W
The start position of UAV	$(0, 0, 200)$
Discounting factor	$\gamma = 0.9$
Max number of users per cluster	10
Noise power	$\alpha^2 = -110dBm$
The reference channel power gain	$\beta_0 = -50dB$
Path-loss exponent	2

VI. Simulation Results

In this section, we present our simulation results characterising the joint optimisation problem of UAV-assisted IoT networks. To highlight the efficiency of our proposed model and the DRL methods, we consider a pair of scenarios: a simple having three clusters, and a more complex one with five clusters in the coverage area. We use Tensorflow 1.13.1 [53] and the Adam optimiser of [54] for training the neural networks. In this paper, we set the maximum value of β/ζ not too large because we prefer the completion of a mission. The maximum value is set to $\beta/\zeta = 4/1$. All the other parameters are provided in Table II.

In Fig. 2, we present the trajectory obtained after training using the DQL algorithm in the 5-cluster scenario. The green circle and blue dots represent the clusters' coverage and the user nodes, respectively. The red line and black line in the figure represent the UAV's trajectory based on our method in (8) and (9), respectively. The UAV starts at $(0, 0)$, visits about 40 users, and lands at the destination that is denoted by the black star. In a complex environment setting, it is challenging to expect the UAV to visit all users, while satisfying the flight-duration and power level constraints.

A. Expected reward

We compare our proposed algorithm with opitimal performance and the Q-learning algorithm. However, to achieve the optimal results, we have defined some assumptions of knowing the IoT's position and unlimited power level of the UAV. For purposes of comparison, we run the algorithm five times in five different environmental settings and take the average to draw

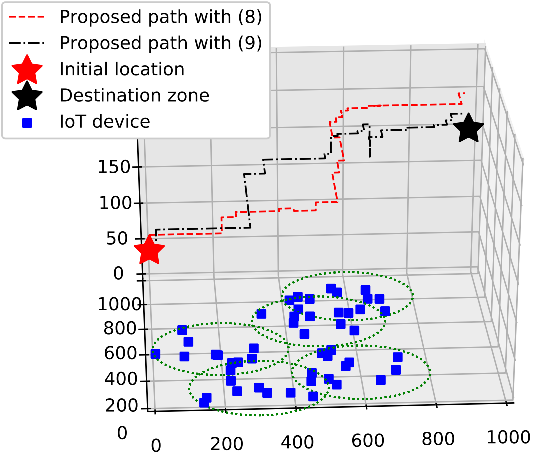
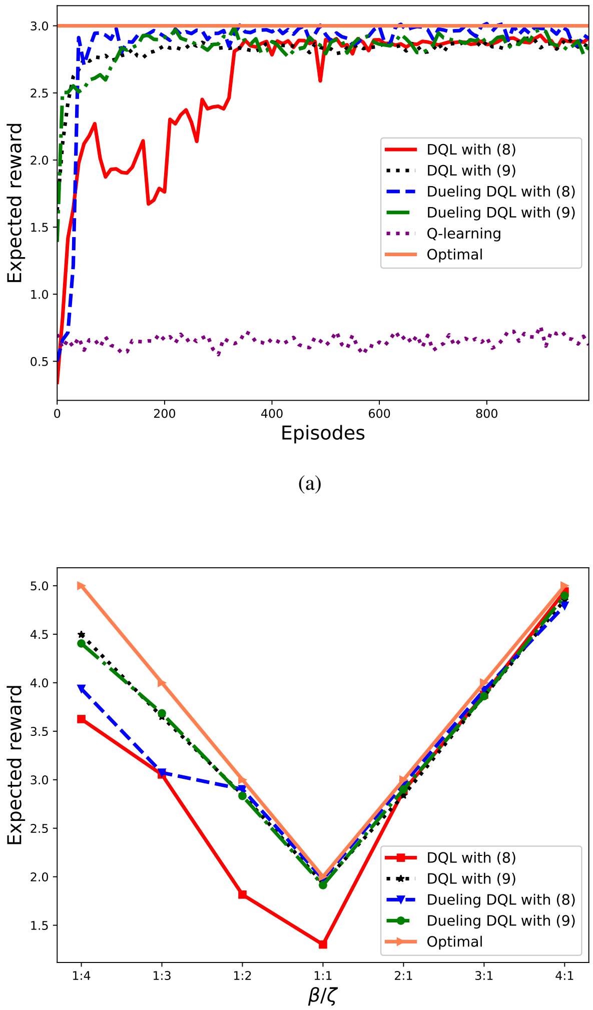


Fig. 2. Trajectory obtained by using our dueling DQL algorithm



(a)

(b)

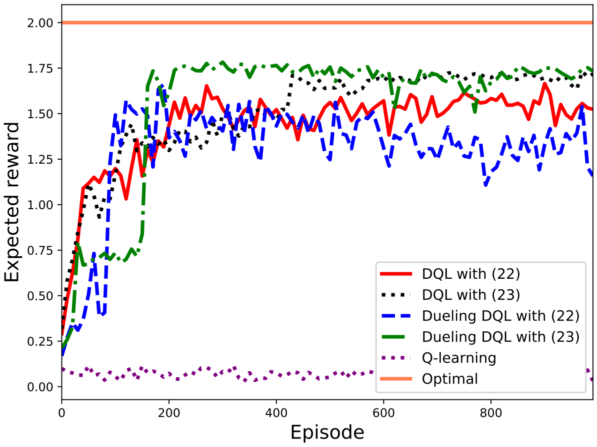
Fig. 3. The performance when using the DQL and dueling DQL algorithms with 3 clusters while considering different β/ζ values

the figures. Firstly, we compare the reward obtained following (7). Let us consider the 3-cluster scenario and $\beta/\zeta = 2 : 1$ in Fig. 3a, where the DQL and the dueling DQL algorithms using the exponential function (9) reach the best performance. When using the exponential trajectory design function (9), the performance converges faster than that of the DQL and of the dueling DQL methods using the binary trajectory function (8). The performance of using the Q-learning algorithm is worst. In addition, in Fig. 3b, we compare the performance of the DQL and dueling DQL techniques using different β/ζ values. The average performance of the dueling DQL algorithm is better than that of the DQL algorithm. Furthermore, the results of using the exponential function (9) are better than that of the ones using the binary function (8). When the value of $\beta/\zeta \geq 1 : 2$, the performance achieved by the DQL and dueling DQL algorithm close to the optimal performance.

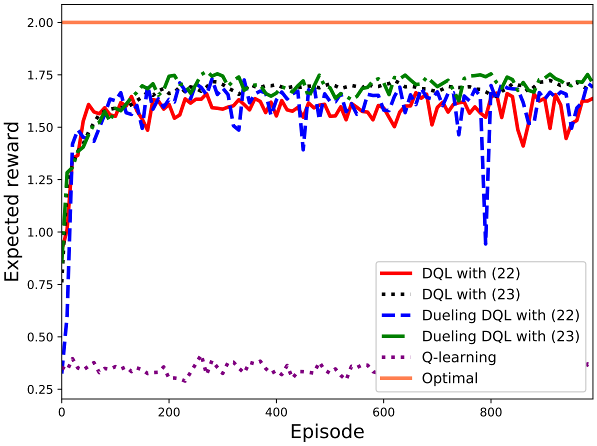
Furthermore, we compare the rewards obtained by the DQL and dueling DQL algorithms in complex scenarios with 5 clusters and 50 user nodes in Fig. 4. The performance of using the episode reward (23) is better than that using the immediate reward (22) in both trajectory designs relying on the DQL and dueling DQL algorithms. In Fig. 4a, we compare the performance in conjunction with the binary trajectory design while in Fig. 4b the exponential trajectory design is considered. or $\beta/\zeta = 1 : 1$, the rewards obtained by the DQL and dueling DQL are similar and stable after about 400 episodes. When using the exponential function (9), the dueling DQL algorithm reaches the best performance and close to the optimal performance. Moreover, the convergence of the dueling DQL technique is faster than that of the DQL algorithm. In both reward definitions, the Q-learning with (22) shows the worst performance.

In Fig. 5, we compare the performance of the DQL and of the dueling DQL algorithms while considering different β/ζ parameter values. The dueling DQL algorithm shows better performance for all the β/ζ pair values, exhibiting better rewards. Additionally, when using the exponential function (9), both proposed algorithms show better performance than the ones using the binary function (8) if $\beta/\zeta \leq 1 : 1$, but it becomes less effective when β/ζ is set higher. Again, we achieve a near-optimal solution while we consider a complex environment without knowing the IoT nodes' position and mobile users. It is challenging to expect the UAV to visit all IoT nodes with limited flying power and duration.

We compare the performance of the DQL and of the dueling DQL algorithm using different reward function setting in Fig. 6 and in Fig. 7, respectively. The DQL algorithm reaches the best performance when using the episode reward (23) in Fig. 6a while the fastest convergence speed can be achieved by using the exponential function (9). When $\beta/\zeta \geq 1 : 1$, the DQL algorithm relying on the episode function (23) outperforms the ones using the immediate reward function (22) in Fig. 6b. The reward (7) using the exponential trajectory design (9) has a better performance than that using the binary



(a) With (8)



(b) With (9)

Fig. 4. The expected reward when using the DQL and dueling DQL algorithms with 5-cluster scenario

trajectory design (8) for all the β/ζ values. The similar results are shown when using the dueling DQL algorithm in Fig. 7. The immediate reward function (22) is less effective than the episode reward function (23).

B. Throughput comparison

In (7), we consider two elements: the trajectory cost and the average throughput. In order to quantify the communication efficiency, we compare the total throughput in different scenarios. In Fig. 8, the performances of the DQL algorithm associated with several β/ζ values are considered while using the binary trajectory function (8), the episode reward (23) and 3 clusters. The throughput obtained for $\beta/\zeta = 1:1$ is higher than that of the others and when β increases, the performance degrades. However, when comparing with the Fig. 3b, we realise that in some scenarios the UAV was stuck and could not find the way to the destination. That leads to increased flight time spent and distance travelled. More details are shown in Fig. 8b, where we compare the expected throughput of both the DQL and dueling DQL algorithms. The best throughput is achieved when using the dueling DQL algorithm with $\beta/\zeta = 1:1$ in conjunction with (8), which is higher than the peak of the DQL method with $\beta/\zeta = 1:2$.

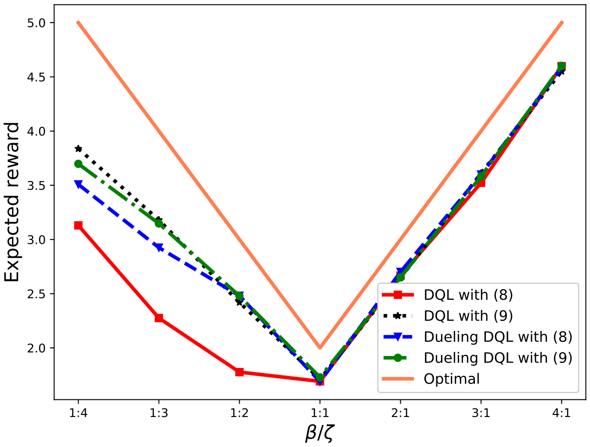
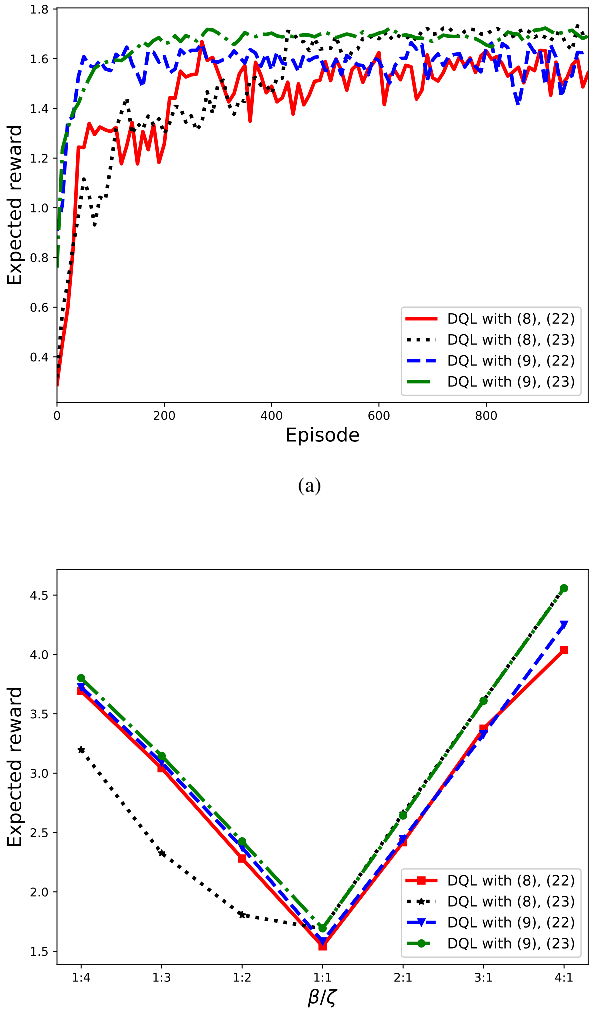


Fig. 5. The performance when using the DQL and dueling DQL algorithms with 5 clusters and different β/ζ values

In Fig. 9, we compare the throughput of different techniques in the 5-cluster scenario. Let us now consider the binary trajectory design function (8) in Fig. 9a, where the DQL algorithm achieves the best performance using $\beta/\zeta = 1:1$ and $\beta/\zeta = 2:1$. There is a slight difference between the DQL method having different settings, when using exponential the trajectory design function (9), as shown in Fig. 9b.

In Fig. 10 and Fig. 11, we compare the throughput of different β/ζ pairs. The DQL algorithm reaches the optimal throughput with the aid of trial-and-learn methods, hence it is important to carefully design the reward function to avoid excessive offline training. As shown in Fig. 10, the DQL and dueling DQL algorithm exhibit reasonable stability for several $\beta/\zeta \leq 1:1$ pairs as well as reward functions. While we can achieve the similar expected reward with different reward setting in Fig. 6, the throughput is degraded when the β/ζ increases. In contrast, with higher β values, the UAV can finish the mission faster. It is a trade-off game when we can choose an approximate β/ζ value for our specific purposes. When we employ the DQL and the dueling DQL algorithms with the episode reward (23), the throughput attained is higher than that using the immediate reward (22) with different β/ζ values.

Furthermore, we compare the expected throughput of the DQL and of the dueling DQL algorithm when using the exponential trajectory design (9) in Fig. 11a and the episode reward (23) in Fig. 11b. In Fig. 11a, the dueling DQL method outperforms the DQL algorithm for almost all β/ζ values in both function (22) and (23). When we use the episode reward (23), the obtained throughput are stable with different β/ζ values. The throughput attained by using the exponential function (9) is lower than that using the binary trajectory (8) and by using the episode reward (23) is higher than that



(a)

(b)

Fig. 6. The expected reward when using the DQL algorithm with 5 clusters and different reward function settings

using the immediate reward (22). We can achieve the best performance when using the dueling DQL algorithm with (9) and (23). However, in some scenarios, we can achieve the better performance with different algorithmic setting as we can see in Fig. 8b and Fig. 10a. Thus, there is a trade-off governing the choice of the algorithm and function design.

C. Parametric Study

In Fig. 12, we compare the performance of our DQL technique using different *exploration* parameters γ and ϵ values in our ϵ-greedy method. The DQL algorithm achieves the best performance with the discounting factor of $\gamma = 0.9$ and $\epsilon = 0.9$ in the 5-cluster scenario of Fig. (12). Balancing the *exploration* and *exploitation* as well as the action chosen is quite challenging, in order to maintain a steady performance of the DQL algorithm. Based on the results of Fig. 12, we opted for $\gamma = 0.9$ and $\epsilon = 0.9$ for our algorithmic setting.

Next, we compare the expected reward of different mini-batch sizes, K. In the 5-cluster scenario of Fig. 13, the DQL achieves the optimal performance with a batch size of $K = 32$. There is a slight difference in terms of convergence speed with batch size $K = 32$ is the fastest. Overall, we set the mini-batch size to $K = 32$ for our DQL algorithm.

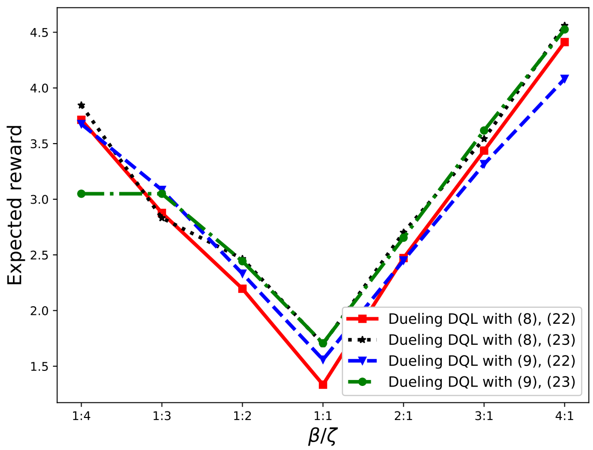


Fig. 7. The performance when using the dueling DQL with 5 clusters, and different β/ζ values

Fig. 14 shows the performance of the DQL algorithm with different learning rates in updating the neural networks parameters while considering the scenarios of 5 clusters. When the learning rate is as high as $\alpha = 0.01$, the pace of updating the network may result the fluctuating performance. Moreover, when $\alpha = 0.0001$ or $\alpha = 0.00001$ the convergence speed is slower and may be stuck in a local optimum instead reaching the global optimum. Thus, based on our experiments, we opted for the learning rate of $\alpha = 0.001$ for the algorithms.

VII. Conclusion

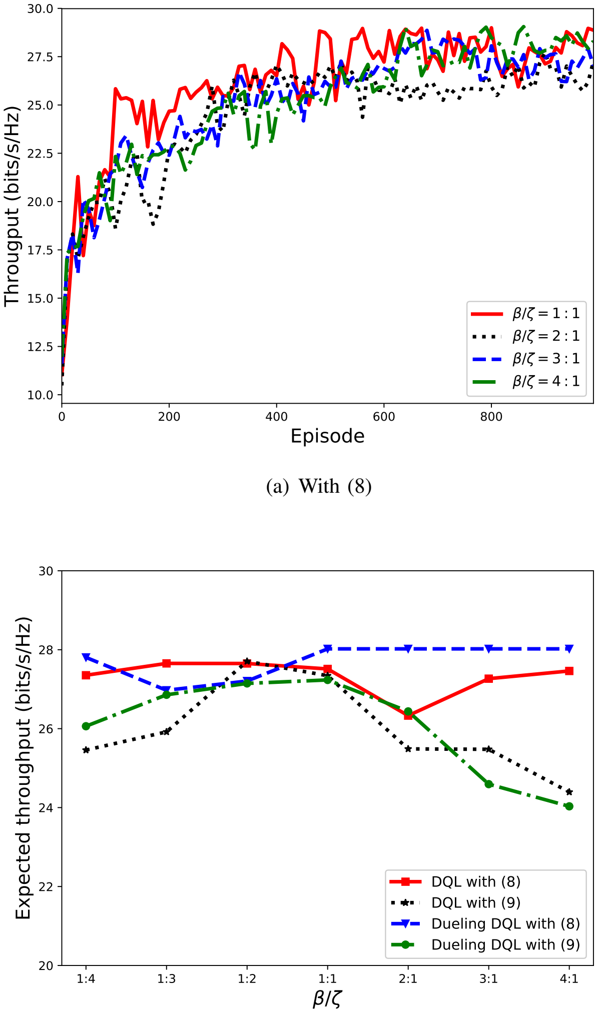
In this paper, the DRL technique has been proposed jointly optimising the flight trajectory and data collection performance of UAV-assisted IoT networks. The optimisation game has been formulated to balance the flight time and total throughput while guaranteeing the quality-of-service constraints. Bearing in mind the limited UAV power level and the associated communication constraints, we proposed a DRL technique for maximising the throughput while the UAV has to move along the shortest path to reach the destination. Both the DQL and dueling DQL techniques having a low computational complexity have been conceived. Our simulation results showed the efficiency of our techniques both in simple and complex environmental settings.

References

[1] "Drone trial to help Isle of Wight receive medical supplies faster during COVID19 pandemic." [Online]. Available: https://www.southampton.ac.uk/news/2020/04/drones-covid-iow.page

[2] "This Chilean community is using drones to deliver medicine to the elderly." [Online]. Available: https://www.weforum.org/agenda/2020/04/drone-chile-covid19/

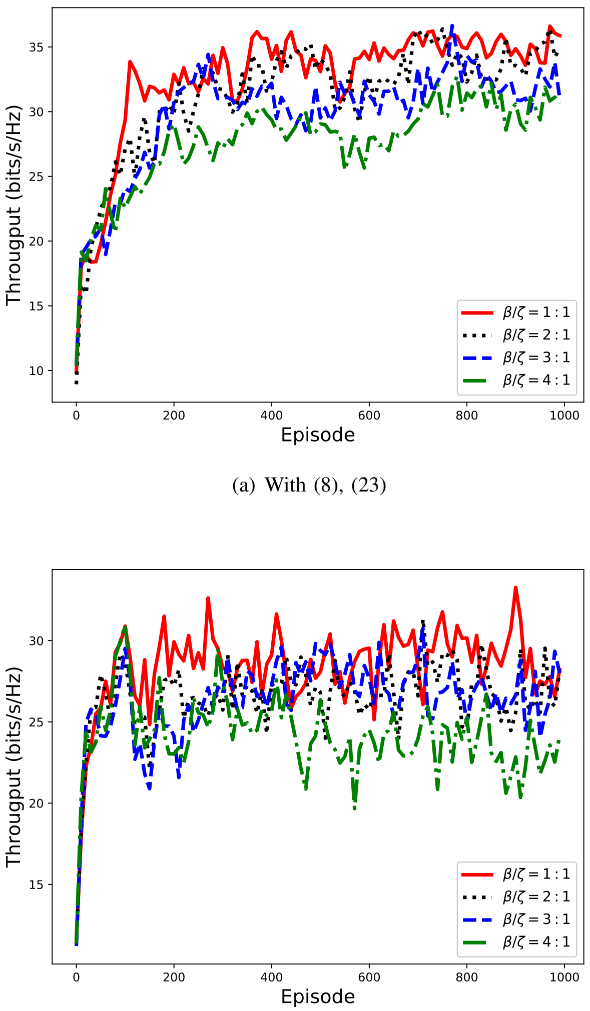
[3] M. Gao, X. Xu, Y. Klinger, J. van der Woerd, and P. Tapponnier, "High-resolution mapping based on an unmanned aerial vehicle (UAV) to capture paleoseismic offsets along the Altyn-Tagh fault, China," *Sci. Rep.*, vol. 7, no. 1, pp. 1–11, Aug. 2017.



(a) With (8)

(b)

Fig. 8. The network's sum-rate when using the DQL and dueling DQL algorithms with 3 clusters



(a) With (8), (23)

(b) With (9), (23)

Fig. 9. The obtained total throughput when using the DQL algorithm with 5 clusters

[4] Q. Liu, J. Wu, P. Xia, S. Zhao, Y. Yang, W. Chen, and L. Hanzo, "Charging unplugged: Will distributed laser charging for mobile wireless power transfer work?" *IEEE Vehicular Technology Magazine*, vol. 11, no. 4, pp. 36–45, Dec. 2016.

[5] H. Claussen, "Distributed algorithms for robust self-deployment and load balancing in autonomous wireless access networks," in *Proc. IEEE Int. Conf. on Commun. (ICC)*, vol. 4, Istanbul, Turkey, Jun. 2006, pp. 1927–1932.

[6] J. Gong, T.-H. Chang, C. Shen, and X. Chen, "Flight time minimization of UAV for data collection over wireless sensor networks," *IEEE J. Select. Areas Commun.*, vol. 36, no. 9, pp. 1942–1954, Sept. 2018.

[7] C. Zhong, M. C. Gursoy, and S. Velipasalar, "Deep reinforcement learning-based edge caching in wireless networks," *IEEE Trans. Cogn. Commun. Netw.*, vol. 6, no. 1, pp. 48–61, Mar. 2020.

[8] H. Wu, Z. Wei, Y. Hou, N. Zhang, and X. Tao, "Cell-edge user offloading via flying UAV in non-uniform heterogeneous cellular networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 4, pp. 2411–2426, Apr. 2020.

[9] H. Huang *et al.*, "Deep reinforcement learning for UAV navigation through massive MIMO technique," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 1117–1121, Jan. 2020.

[10] T. Q. Duong, L. D. Nguyen, H. D. Tuan, and L. Hanzo, "Learning-aided realtime performance optimisation of cognitive UAV-assisted disaster communication," in *Proc. IEEE Global Communications Conference (GLOBECOM)*, Waikoloa, HI, USA, Dec. 2019.

[11] T. Q. Duong, L. D. Nguyen, and L. K. Nguyen, "Practical optimisation of path planning and completion time of data collection for UAV-enabled disaster communications," in *Proc. 15th Int. Wireless Commun. Mobile Computing Conf. (IWCMC)*, Tangier, Morocco, Jun. 2019, pp. 372–377.

[12] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Unmanned aerial vehicle with underlaid device-to-device communications: Performance and tradeoffs," *IEEE Trans. Wireless Commun.*, vol. 15, no. 6, pp. 3949–3963, Jun. 2016.

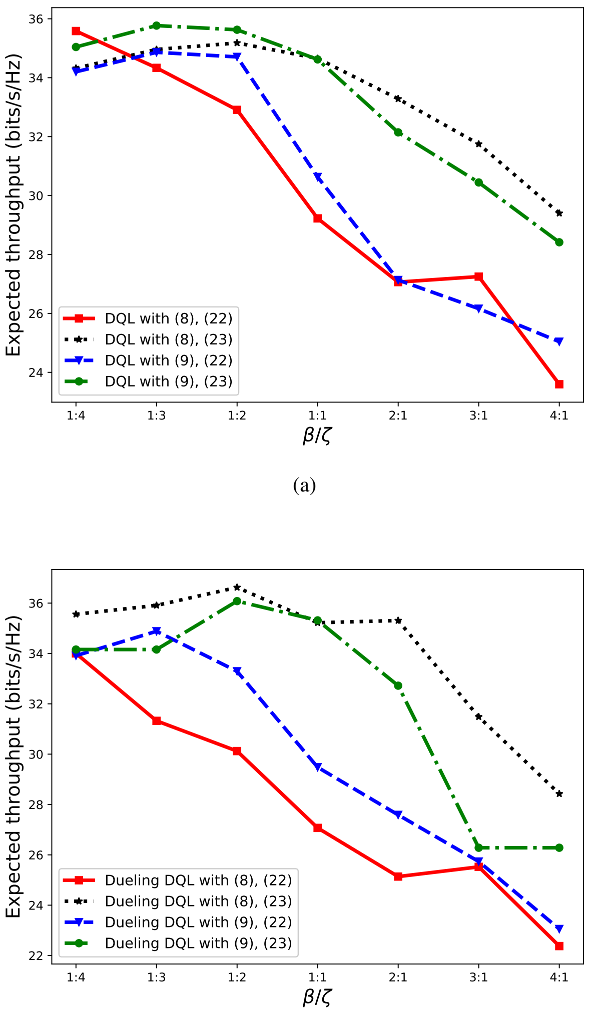
[13] L. D. Nguyen, A. Kortun, and T. Q. Duong, "An introduction of real-time embedded optimisation programming for UAV systems under disaster communication," *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems*, vol. 5, no. 17, pp. 1–8, Dec. 2018.

[14] M.-N. Nguyen, L. D. Nguyen, T. Q. Duong, and H. D. Tuan, "Real-time optimal resource allocation for embedded UAV communication systems," *IEEE Wireless Commun. Lett.*, vol. 8, no. 1, pp. 225–228, Feb. 2019.

[15] X. Li, H. Yao, J. Wang, X. Xu, C. Jiang, and L. Hanzo, "A near-optimal UAV-aided radio coverage strategy for dense urban areas," *IEEE Trans. Veh. Technol.*, vol. 68, no. 9, pp. 9098–9109, Sept. 2019.

[16] H. Zhang and L. Hanzo, "Federated learning assisted multi-UAV networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 14 104–14 109, Nov. 2020.

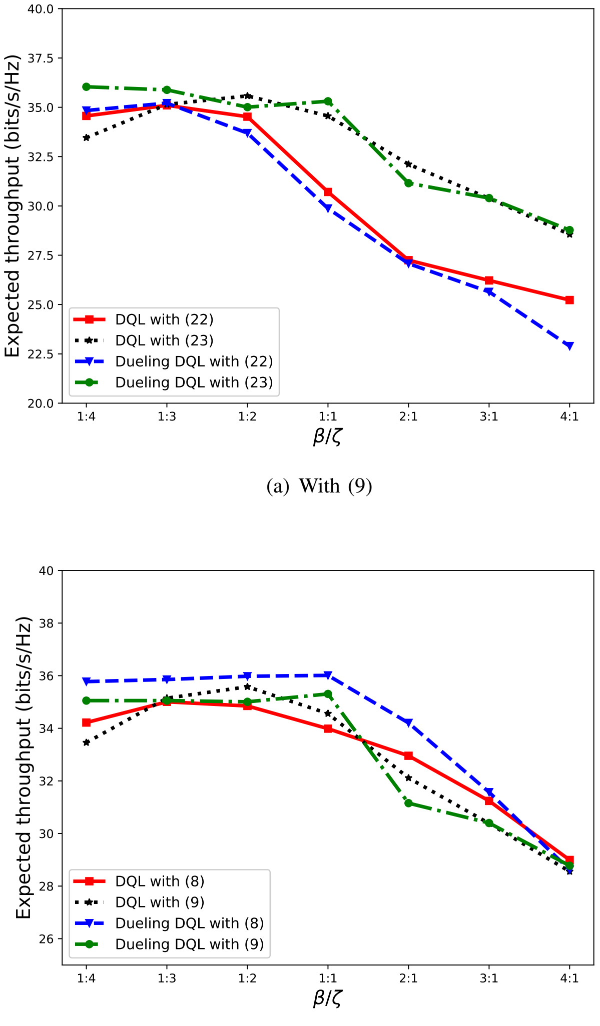
[17] X. Liu, Y. Liu, Y. Chen, and L. Hanzo, "Trajectory design and power control for multi-UAV assisted wireless networks: A machine learning approach," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 7957–7969, Aug. 2019.



(a)

(b)

Fig. 10. The obtained throughput when using the DQL and dueling DQL algorithms in 5-cluster scenario



(a) With (9)

(b) With (23)

Fig. 11. The expected throughput when using the DQL and dueling DQL algorithms with 5 clusters

[18] K. K. Nguyen, T. Q. Duong, N. A. Vien, N.-A. Le-Khac, and L. D. Nguyen, “Distributed deep deterministic policy gradient for power allocation control in D2D-based V2V communications,” *IEEE Access*, vol. 7, pp. 164 533–164 543, Nov. 2019.

[19] K. K. Nguyen, T. Q. Duong, N. A. Vien, N.-A. Le-Khac, and N. M. Nguyen, “Non-cooperative energy efficient power allocation game in D2D communication: A multi-agent deep reinforcement learning approach,” *IEEE Access*, vol. 7, pp. 100 480–100 490, Jul. 2019.

[20] K. K. Nguyen, N. A. Vien, L. D. Nguyen, M.-T. Le, L. Hanzo, and T. Q. Duong, “Real-time energy harvesting aided scheduling in UAV-assisted D2D networks relying on deep reinforcement learning,” *IEEE Access*, vol. 9, pp. 3638–3648, Dec. 2021.

[21] K. Li, W. Ni, E. Tovar, and A. Jamalipour, “On-board deep Q-network for UAV-assisted online power transfer and data collection,” *IEEE Trans. Veh. Technol.*, vol. 68, no. 12, pp. 12 215–12 226, Dec. 2019.

[22] U. Challita, W. Saad, and C. Bettstetter, “Interference management for cellular-connected UAVs: A deep reinforcement learning approach,” *IEEE Trans. Wireless Commun.*, vol. 18, no. 4, pp. 2125–2140, Apr. 2019.

[23] 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.

[24] C. Wang, J. Wang, Y. Shen, and X. Zhang, “Autonomous navigation of UAVs in large-scale complex environments: A deep reinforcement learning approach,” *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 2124–2136, Mar. 2019.

[25] S. Gu, E. Holly, T. Lillicrap, and S. Levine, “Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates,” in *Proc. IEEE International Conf. Robot. Autom. (ICRA)*, May 2017, pp. 3389–3396.

[26] Q. Cai, A. Filos-Ratsikas, P. Tang, and Y. Zhang, “Reinforcement mechanism design for fraudulent behaviour in e-commerce,” in *Proc. Thirty-Second AAAI Conf. Artif. Intell.*, 2018.

[27] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, “Playing Atari with deep reinforcement learning,” 2013. [Online]. Available: arXivpreprintarXiv:1312.5602

[28] Y. Yu, T. Wang, and S. C. Liew, “Deep-reinforcement learning multiple access for heterogeneous wireless networks,” *IEEE J. Select. Areas Commun.*, vol. 37, no. 6, pp. 1277–1290, Jun. 2019.

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

[30] 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.

[31] D. Yang, Q. Wu, Y. Zeng, and R. Zhang, “Energy tradeoff in ground-to-UAV communication via trajectory design,” *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6721–6726, Jul. 2018.

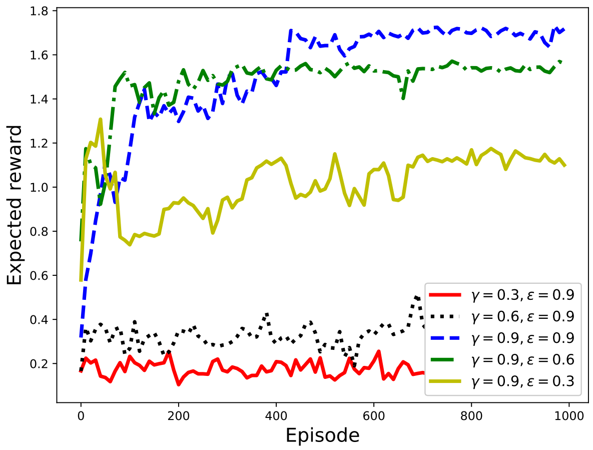


Fig. 12. The performance when using the DQL algorithm with different discount factors, γ, and exploration factors, ϵ

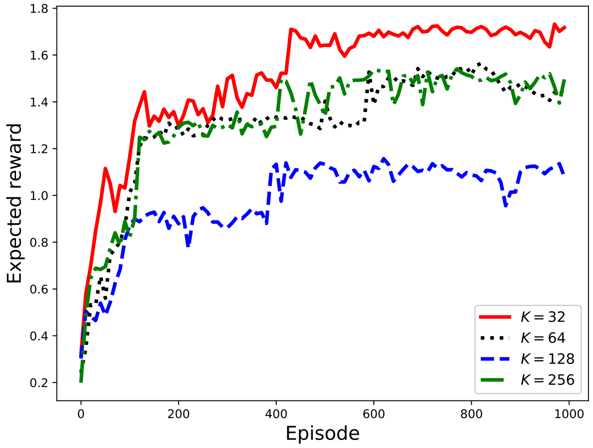


Fig. 13. The performance when using the DQL algorithm in 5-cluster scenario and different batch sizes, K

[32] H. Wang, J. Wang, G. Ding, J. Chen, F. Gao, and Z. Han, “Completion time minimization with path planning for fixed-wing UAV communications,” *IEEE Trans. Wireless Commun.*, vol. 18, no. 7, pp. 3485–3499, Jul. 2019.

[33] H. T. Nguyen, H. D. Tuan, T. Q. Duong, H. V. Poor, and W.-J. Hwang, “Joint D2D assignment, bandwidth and power allocation in cognitive UAV-enabled networks,” *IEEE Trans. Cogn. Commun. Netw.*, vol. 6, no. 3, pp. 1084–1095, Sept. 2020.

[34] L. Liu, S. Zhang, and R. Zhang, “Multi-beam UAV communication in cellular uplink: Cooperative interference cancellation and sum-rate maximization,” *IEEE Trans. Wireless Commun.*, vol. 18, no. 10, pp. 4679–4691, Oct. 2019.

[35] L. Xie, J. Xu, and R. Zhang, “Throughput maximization for UAV-enabled wireless powered communication networks,” *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1690–1703, Apr. 2019.

[36] L. D. Nguyen, K. K. Nguyen, A. Kortun, and T. Q. Duong, “Real-time deployment and resource allocation for distributed UAV systems in disaster relief,” in *Proc. IEEE 20th International Workshop on Signal Processing Advances in Wireless Commun. (SPAWC)*, Cannes, France, Jul. 2019, pp. 1–5.

[37] Q. Wu, Y. Zeng, and R. Zhang, “Joint trajectory and communication design for multi-UAV enabled wireless networks,” *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 2109–2121, Mar. 2018.

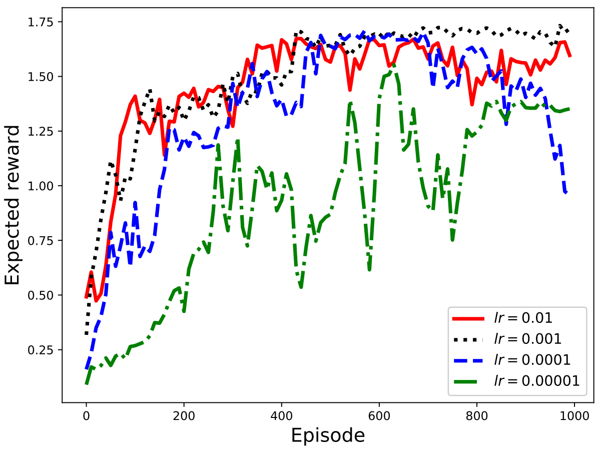


Fig. 14. The performance when using DQL algorithm with different learning rate, lr

[38] C. Zhan, Y. Zeng, and R. Zhang, “Energy-efficient data collection in UAV enabled wireless sensor network,” *IEEE Wireless Commun. Lett.*, vol. 7, no. 3, pp. 328–331, Jun. 2018.

[39] H. Wang, G. Ren, J. Chen, G. Ding, and Y. Yang, “Unmanned aerial vehicle-aided communications: Joint transmit power and trajectory optimization,” *IEEE Wireless Commun. Lett.*, vol. 7, no. 4, pp. 522–525, Aug. 2018.

[40] Z. Wang, R. Liu, Q. Liu, J. S. Thompson, and M. Kadoch, “Energy-efficient data collection and device positioning in UAV-assisted IoT,” *IEEE Internet Things J.*, vol. 7, no. 2, pp. 1122–1139, Feb. 2020.

[41] J. Li *et al.*, “Joint optimization on trajectory, altitude, velocity, and link scheduling for minimum mission time in UAV-aided data collection,” *IEEE Internet Things J.*, vol. 7, no. 2, pp. 1464–1475, Feb. 2020.

[42] M. Samir, S. Sharafeddine, C. M. Assi, T. M. Nguyen, and A. Ghrayeb, “UAV trajectory planning for data collection from time-constrained IoT devices,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 34–46, Jan. 2020.

[43] M. Hua, L. Yang, Q. Wu, and A. L. Swindlehurst, “3D UAV trajectory and communication design for simultaneous uplink and downlink transmission,” *IEEE Trans. on Commun.*, vol. 68, no. 9, pp. 5908–5923, Sept. 2020.

[44] C. Zhan and Y. Zeng, “Aerial–ground cost tradeoff for multi-UAV-enabled data collection in wireless sensor networks,” *IEEE Trans. on Commun.*, vol. 68, no. 3, pp. 1937–1950, Mar. 2020.

[45] S. Zhang and R. Zhang, “Radio map-based 3D path planning for cellular-connected UAV,” *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1975–1989, Mar. 2021.

[46] Y. Zeng, X. Xu, S. Jin, and R. Zhang, “Simultaneous navigation and radio mapping for cellular-connected UAV with deep reinforcement learning,” *IEEE Trans. Wireless Commun.*, vol. 20, no. 7, pp. 4205–4220, Jul. 2021.

[47] M. Samir, C. Assi, S. Sharafeddine, D. Ebrahimi, and A. Ghrayeb, “Age of information aware trajectory planning of UAVs in intelligent transportation systems: A deep learning approach,” *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 12 382–12 395, Nov. 2020.

[48] H. Jaeger, “The “echo state” approach to analysing and training recurrent neural networks-with an erratum note,” *” GMD - German National Research Institute for Computer Science, Tech. Rep.*, vol. 148, no. 34, p. 13, Jan. 2010.

[49] T. P. Lillicrap *et al.*, “Continuous control with deep reinforcement learning,” in *Proc. 4th International Conf. on Learning Representations (ICLR)*, 2016.

[50] Z. Wang, T. Schaul, M. Hessel, H. van Hasselt, M. Lanctot, and N. de Freitas, “Dueling network architectures for deep reinforcement learning,” 2015. [Online]. Available: arXivpreprintarXiv:1511.06581

[51] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., 1994.

[52] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Athena Scientific Belmont, MA, 1995, vol. 1, no. 2.

[53] M. Abadi *et al.*, "Tensorflow: A system for large-scale machine learning," in *Proc. 12th USENIX Sym. Opr. Syst. Design and Imp. (OSDI 16)*, Nov. 2016, pp. 265–283.

[54] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," 2014. [Online]. Available: arXivpreprintarXiv:1412.6980

Khoi Khac Nguyen (Student Member, IEEE) was born in Bac Ninh, Vietnam. He received his B.S. degree in information and communication technology from the Hanoi University of Science and Technology (HUST), Vietnam in 2018. He is working towards his Ph.D. degree with the School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, U.K. His research interests include machine learning and deep reinforcement learning for real-time optimisation in wireless networks, reconfigurable intelligent surfaces, unmanned air vehicle (UAV) communication and massive Internet of Things (IoTs).

Trung Q. Duong (Fellow, IEEE) is a Chair Professor of Telecommunications at Queen's University Belfast (UK), where he was a Lecturer (Assistant Professor) (2013-2017), a Reader (Associate Professor) (2018-2020), and Full Professor from August 2020. He also holds a prestigious Research Chair of Royal Academy of Engineering. His current research interests include wireless communications, machine learning, realtime optimisation, and data analytic.

Dr. Duong currently serves as an Editor for the IEEE Transactions on Wireless Communications, IEEE Transactions on Vehicular Technology, IEEE Wireless Communications Letters, and an Executive Editor for IEEE Communications Letters. He has served as an Editor/Guest Editor for IEEE Transactions on Communications, IEEE Wireless Communications, IEEE Communications Magazines, IEEE Communications Letters, and IEEE Journal on Selected Areas in Communications. He was awarded the Best Paper Award at the IEEE Vehicular Technology Conference (VTC-Spring) in 2013, IEEE International Conference on Communications (ICC) 2014, IEEE Global Communications Conference (GLOBECOM) 2016 and 2019, IEEE Digital Signal Processing Conference (DSP) 2017, and International Wireless Communications & Mobile Computing Conference (IWCMC) 2019. He is the recipient of prestigious Royal Academy of Engineering Research Fellowship (2015-2020) and has won a prestigious Newton Prize 2017. He is a Fellow of IEEE (2022 Class).

Tan Do-Duy (Member, IEEE) received his B.S. degree from Ho Chi Minh City University of Technology (HCMUT), Vietnam, and M.S. degree from Kumoh National Institute of Technology, Korea, in 2010 and 2013, respectively. He received his Ph.D. degree from Autonomous University of Barcelona, Spain, in 2019. He is currently with the Department of Computer and Communication Engineering, Ho Chi Minh City University of Technology and Education (HCMUTE) in Vietnam as an Assistant Professor. His main research interests include wireless cooperative communications, real-time optimisation for resource allocation in wireless networks, and coding applications for wireless communications.

Holger Claussen (Fellow, IEEE) is Head of the Wireless Communications Laboratory at Tyndall National Institute, and Research Professor at University College Cork, where he is building up research teams in the area of RF, Access, Protocols, AI, and Quantum Systems to invent the future of Wireless Communication Networks. Previously he led the Wireless Communications Reserach Department of Nokia Bell Labs located in Ireland and the US. In this role, he and his team innovated in all areas related to future evolution, deployment, and operation of wireless networks to enable exponential growth in mobile data traffic and reliable low latency communications. His research in this domain has been commercialised in Nokia's (formerly Alcatel-Lucent's) Small Cell product portfolio and continues to have significant impact. He received the 2014 World Technology Award in the individual category Communications Technologies for innovative work of "the greatest likely long-term significance". Prior to this, Holger directed research in the areas of self-managing networks to enable the first large scale femtocell deployments. Holger joined Bell Labs in 2004, where he began his research in the areas of network optimisation, cellular architectures, and improving energy efficiency of networks. Holger received his Ph.D. degree in signal processing for digital communications from the University of Edinburgh, United Kingdom in 2004. He is author of the book "Small Cell Networks", more than 130 journal and conference publications, 78 granted patent families, and 46 filed patent applications pending. He is Fellow of the IEEE, Fellow of the World Technology Network, and member of the IET.

Lajos Hanzo (Fellow, IEEE) received his Master degree and Doctorate in 1976 and 1983, respectively from the Technical University (TU) of Budapest. He was also awarded the Doctor of Sciences (DSc) degree by the University of Southampton (2004) and Honorary Doctorates by the TU of Budapest (2009) and by the University of Edinburgh (2015). He is a Foreign Member of the Hungarian Academy of Sciences and a former Editor-in-Chief of the IEEE Press. He has served several terms as Governor of both IEEE ComSoc and of VTS. He has published 2000+ contributions at IEEE Xplore, 19 Wiley-IEEE Press books and has helped the fast-track career of 123 PhD students. Over 40 of them are Professors at various stages of their careers in academia and many of them are leading scientists in the wireless industry. He is also a Fellow of the Royal Academy of Engineering (FREng), of the IET and of EURASIP. He was bestowed upon the Eric Sumner Field Award.