CISTER

Research Centre in
Real-Time & Embedded
Computing Systems

Technical Report

On-board Deep Q-Network for UAV-assisted Online Power Transfer and Data Collection

Kai Li*

Wei Ni

Eduardo Tovar*

*CISTER Research Centre

CISTER-TR-190601

2019/06/02

On-board Deep Q-Network for UAV-assisted Online Power Transfer and Data Collection

Kai Li*, Wei Ni, Eduardo Tovar*

*CISTER Research Centre

Polytechnic Institute of Porto (ISEP P.Porto)

Rua Dr. António Bernardino de Almeida, 431

4200-072 Porto

Portugal

Tel.: +351.22.8340509, Fax: +351.22.8321159

E-mail: kaili@isep.ipp.pt, Wei.Ni@data61.csiro.au, emt@isep.ipp.pt

https://www.cister-labs.pt

Abstract

Unmanned Aerial Vehicles (UAVs) with Microwave Power Transfer (MPT) capability provide a practical means to deploy a large number of wireless powered sensing devices into areas with no access to persistent power supplies. The UAV can charge the sensing devices remotely and harvest their data. A key challenge is online MPT and data collection in the presence of on-board control of a UAV (e.g., patrolling velocity) for preventing battery drainage and data queue overflow of the sensing devices, while up-to-date knowledge on battery level and data queue of the devices is not available at the UAV. In this paper, an on-board deep Q-network is developed to minimize the overall data packet loss of the sensing devices, by optimally deciding the device to be charged and interrogated for data collection, and the instantaneous patrolling velocity of the UAV. Specifically, we formulate a Markov Decision Process (MDP) with the states of battery level and data queue length of sensing devices, channel conditions, and waypoints given the trajectory of the UAV; and solve it optimally with Q-learning. Furthermore, we propose the on-board deep Q-network that can enlarge the state space of the MDP, and a deep reinforcement learning based scheduling algorithm that asymptotically derives the optimal solution online, even when the UAV has only outdated knowledge on the MDP states. Numerical results demonstrate that the proposed deep reinforcement learning algorithm reduces the packet loss by at least 69.2%, as compared to existing non-learning greedy algorithms.

On-board Deep Q-Network for UAV-assisted Online Power Transfer and Data Collection

Kai Li[a], Wei Ni[b], Eduardo Tovar[a]

[a] *Real-Time and Embedded Computing Systems Research Centre, Portugal*
[b] *Commonwealth Scientific and Industrial Research Organization, Australia*

Abstract

Unmanned Aerial Vehicles (UAVs) with Microwave Power Transfer (MPT) capability provide a practical means to deploy a large number of wireless powered sensing devices into areas with no access to persistent power supplies. The UAV can charge the sensing devices remotely and harvest their data. A key challenge is online MPT and data collection in the presence of on-board control of a UAV (e.g., patrolling velocity) for preventing battery drainage and data queue overflow of the sensing devices, while up-to-date knowledge on battery level and data queue of the devices is not available at the UAV. In this paper, an on-board deep Q-network is developed to minimize the overall data packet loss of the sensing devices, by optimally deciding the device to be charged and interrogated for data collection, and the instantaneous patrolling velocity of the UAV. Specifically, we formulate a Markov Decision Process (MDP) with the states of battery level and data queue length of sensing devices, channel conditions, and waypoints given the trajectory of the UAV; and solve it optimally with Q-learning. Furthermore, we propose the on-board deep Q-network that can enlarge the state space of the MDP, and a deep reinforcement learning based scheduling algorithm that asymptotically derives the optimal solution online, even when the UAV has only outdated knowledge on the MDP states. Numerical results demonstrate that the proposed deep reinforcement learning algorithm reduces the packet loss by at least 69.2%, as compared to existing non-learning greedy algorithms.

Keywords: Unmanned aerial vehicle, microwave power transfer, online

Email address: kaili@isep.ipp.pt (Kai Li)

resource allocation, deep reinforcement learning, Markov decision process

1. Introduction

Wireless sensing devices operating on limited batteries have been airlifted and deployed in remote, human-unfriendly environments. Conventional terrestrial communication networks and persistent power supplies are unavailable or unreliable in such harsh environments. Unmanned Aerial Vehicles (UAVs) have been proposed to harvest data from the sensing devices, and offload command or software pitch, thanks to UAVs' excellent mobility and maneuverability, flexible deployment, and low operational costs [1, 2, 3]. Figure 1 depicts a typical real-time application using the UAV in rescue operations, where the wireless sensing device on the ground, typically running with limited battery, records critical information, such as locations, temperature, health conditions of rescuers, and oxygen supply. Each ground device generates data packets at an application-specific sampling rate, and put them into a data queue for future transmission.

A UAV can be employed to hover over the area of interest, collecting and ferrying the sensory data of the ground device. Since the ground device is constrained by a typically limited battery power which limits scalability and sustainability of the sensor network, Microwave Power Transfer (MPT) has been studied to enable energy harvesting in UAV-assisted data collection [4, 5]. Particularly, we consider a power-splitting MPT technique, e.g., Simultaneous Wireless Information and Power Transfer (SWIPT), where the UAV sends response messages to the ground device meanwhile charging its batteries [6, 7, 8]. Moreover, the ground device, equipped with a data communication antenna and a wireless power receiver, collects electrical energy conveyed by radio frequency signals while recovering the data from the signals. With SWIPT, every ground device only needs a single RF chain with a reduced hardware cost, since MPT and data transmission can work in the same radio frequency band.

In practical scenarios, energy harvesting and data transmission could be severely affected by movements of the UAV and time-varying channels. Moreover, the up-to-date knowledge about battery level and data queue length of the ground devices is not available at the UAV. Therefore, scheduling MPT and data collection online in the presence of on-board control of the UAV (e.g., patrolling velocity) for preventing battery drainage and data queue overflow is critical in UAV-assisted wireless powered sensor networks.

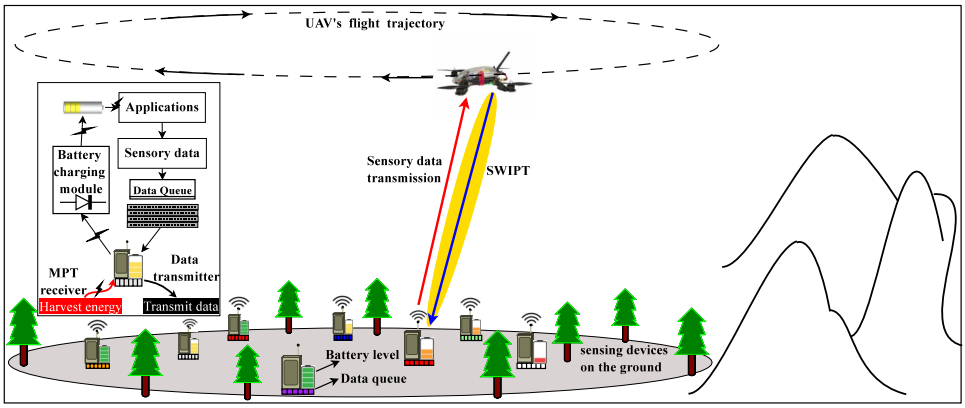


Figure 1: UAV-assisted MPT and data collection with the wireless powered ground devices.

In this paper, the problem is formulated as a Markov Decision Process (MDP) with the states of battery level and data queue length of the ground devices, channel conditions, and waypoints given the trajectory of the UAV, which can be optimally solved by a reinforcement learning approach, e.g., Q-learning. However, Q-learning suffers from the well-known curse of dimensionality, which is impractical for the resource allocation in the UAV-assisted online MPT and data collection due to a large number of states and actions. Furthermore, we propose an on-board deep Q-network that can enlarge the state and action space of the MDP to minimize the data packet loss of the entire system. A new Deep Reinforcement Learning based Scheduling Algorithm (DRL-SA) is developed, which derives the optimal solution online by taking current network state and action as the input and delivering a corresponding action-value function. DRL-SA learns an optimal resource allocation strategy asymptotically through online training at the on-board deep Q-network, where the selection of the ground device, modulation scheme, and instantaneous patrolling velocity of the UAV are jointly optimized based on the action-value function. DRL-SA utilizes an ϵ-greedy policy to balance the network cost minimization with respect to the knowledge already known with trying new actions to obtain knowledge unknown. Moreover, DRL-SA carries out experience replay [9] to significantly reduce expansion of the state space in which the algorithm's scheduling experiences at each time step are stored in a data set. DRL-SA is implemented using Keras deep learning library with Google TensorFlow as the backend engine. Numerical results

demonstrate that the proposed DRL-SA is able to reduce the packet loss by 69.2%, as compared to existing non-learning greedy algorithms.

In our earlier work [10, 11], scheduling strategies were studied with a focus on reducing packet loss in small-scale static wireless powered sensor networks in response to battery level and data queue statues of the ground devices. Due to the low-dimensional channel and device state spaces, the resource allocation problem can be solved by reinforcement learning or dynamic programming. However, in the UAV-assisted MPT and data collection, the mobility of the UAV with the varying patrolling velocity causes rapidly changing wireless channels. As a result, both the state space and the action space are exceedingly large and grow dramatically fast with the size of the network. This prevents conventional resource allocation approaches, such as [10] and [11], from scaling to the high-dimensional input spaces.

The rest of this paper is organized as follows. Section 2 presents related work on UAV-assisted data communication with power transfer. Section 3 studies system structure of UAV-assisted MPT and data collection, as well as the communication protocol design. In Section 4, a deep reinforcement learning algorithm is proposed to address the resource allocation problem for UAV-assisted online MPT and data collection. Numerical results and evaluation are presented in Section 5. Section 6 concludes the paper.

2. Related Work

In this section, we review the literature on data communication, trajectory planning, and power transfer in UAV networks.

2.1. UAV-assisted wireless communications

An energy-efficient UAV relaying scheme is studied in [12] to schedule the data transmission between the UAV and the sensor nodes while guaranteeing packet success rate. A practical and computationally efficient algorithm is designed to extend network lifetime, by decoupling energy balancing and modulation adaptation of the UAVs, and optimizing in an alternating manner. The authors in [13] implement passive scanning at the sensor nodes and periodic beaconing at the UAV to reduce network energy consumption. A non-cooperative game is constructed and equilibrium beaconing period durations are characterized for the UAVs. A learning algorithm is described to allow a UAV to discover the equilibrium beaconing strategy without observing the other UAVs' relaying schedules. Network outage probability is

analyzed in which a number of UAVs are used to relay the source signals to a data sink in a decode-and-forward manner [14]. The use of multiple antennas at the data sink is considered and their respective performance is also examined. The network outage problem can be decoupled into power allocation and trajectory planning subproblems [15]. An approximate solution that iteratively addresses the two subproblems is developed for approaching the minimum outage probability. In each iteration, the trajectory is planned according to the power control results obtained by the last iteration, and then the power control subproblem is solved given the UAV trajectory. A resource allocation algorithm is presented for improving network throughput while guaranteeing a seamless relaying transmission to the data sink outside network coverage via UAV [16]. By analyzing the outage probability, it is shown that non-orthogonal transmissions improve the performance of the UAV relaying network over orthogonal transmissions.

2.2. UAV trajectory planning

UAV trajectory planning is studied in [17, 18, 19, 20], where the UAV acts as a communication relay for connecting the sensor nodes. Network throughput and communication delay can be improved by the motion control of the UAV, e.g., heading, velocity, or radius of the flight trajectory. In [21, 22], the transmit power of the UAV and the trajectory planning are studied to increase network throughput over a finite time horizon. The power allocation exploits the predictable channel changes induced by the UAV's movement while the trajectory planning balances the throughput between the source-UAV and UAV-sink links. The UAV can also be used as a buffer-aided relay in free-space optical systems, which stores data emanating from some stationary sensor nodes for possible future delivery to other nodes [23]. Since an optical link between the two transceivers can be easily smeared by ambient atmospheric conditions, e.g., an intervening cloud, the UAV adjusts its altitude in such a way that would eliminate the cloud attenuation effect.

The existing resource allocation approaches in the UAV relaying network improve the performance based on power control and trajectory planning of the UAV. However, scheduling energy harvesting and data collection was yet to be considered.

2.3. UAV-assisted power transfer

Several studies have integrated MPT technologies into the UAV network. The UAV carrying an MPT transmitter is used to collect sensory data and

extend the lifetime of sensor networks in harsh terrains [24, 25]. With consideration of transferred power attenuation, residual energy and buffered data at the sensor node, the UAV is scheduled to selectively charge the nodes and collect their data. In [26], the UAV trajectory planning with the velocity control is exploited for charging all the sensor nodes in a fair fashion. The problem formulation and solution imply that the hovering location and duration can be designed to enhance the MPT efficiency. Moreover, machine learning techniques can be utilized to predict the UAV's trajectory and improve the energy harvesting efficiency [27]. In [8], SWIPT is introduced into the UAV relaying network, where the sensor node's energy limit can be alleviated by scavenging wireless energy from radio signals transmitted by the UAV. The network throughput is improved by adjusting the UAV's transmit power and the flight trajectory. Several MPT platforms are developed for the UAV to charge batteries of the sensor nodes remotely from the electric grid [28, 29, 30, 31, 32]. The lightweight design of hardware, control algorithms, and experiments are presented to verify the feasibility, reliability and efficiency of the UAV-assisted power transfer. However, the existing literature only focuses on improving the energy efficiency of MPT. The data loss caused by buffer overflows and poor channels is not considered.

3. System Model and Communication Protocol

In this section, we introduce the system model and the communication protocol of UAV-assisted online MPT and data collection. Notations used in the paper are summarized in Table 1.

3.1. System model

The network that we consider consists of I wireless powered ground devices in a remote area. The UAV that acts as a data collection node flies a predetermined circular trajectory for Z laps, where the number of laps is limited by the lifetime of the UAV's battery. Let $v_{\max}$ and $v_{\min}$ denote the maximum and minimum patrolling velocity of the UAV, respectively. The patrolling velocity at time slot t, i.e., $v^z(t)$, can be adjusted during the flight, where $v^z(t) \in [v_{\min}, v_{\max}]$ and $z \in [1, Z]$. The location of the UAV on its trajectory at t in lap z is denoted by $\zeta_z(t)$. The UAV is also responsible for remotely charging the ground devices using MPT. Specifically, receive beamforming is enabled at the UAV to enhance the received signal strength (RSS) and reduce bit error rate (BER). Device i ($\in [1, I]$) harvests energy

from the UAV to power its operations, e.g., sensing, computing and communication. In addition, multi-user beamforming techniques, e.g., zero-forcing beamforming, conjugate beamforming, and singular value decomposition, can be applied to UAV-assisted MPT and data collection. However, they are not considered in this work, due to their requirement of real-time feedback on channel state information.

The complex coefficient of the reciprocal wireless channel between the UAV and device i at t in lap z is $\mathbf{h}_i^z(t)$, which can be known by channel reciprocity. The modulation scheme of device i at t in lap z is denoted by $\phi_i^z(t)$. In particular, $\phi_i^z(t) = 1$, 2, and 3 indicates binary phase-shift keying (BPSK), quadrature-phase shift keying (QPSK), and 8 phase-shift keying (8PSK), respectively, and $\phi_i^z(t) \geq 4$ provides $2^{\phi_i^z(t)}$ quadrature amplitude modulation (QAM).

Suppose that the BER requirement is ε. Consider the generic Nakagami-m fading channel model, ε is given by [33]

$$\varepsilon \approx \frac{0.2}{\Gamma(m)}(\frac{m}{\bar{\gamma}_i})^m \Big[\frac{\Gamma(m, b_{\phi_i^z(t)}\gamma_i(\phi_i^z(t)))}{(b_{\phi_i^z(t)})^m} - \frac{\Gamma(m, b_{\phi_i^z(t)}\gamma_i(\phi_i^z(t)+1))}{(b_{\phi_i^z(t)})^m}\Big], \tag{1}$$

$$b_{\phi_i^z(t)} = \frac{m}{\bar{\gamma}_i} + \frac{3}{2(2^{\phi_i^z(t)}-1)}, \tag{2}$$

where $\Gamma(\cdot)$ is the Gamma function [34], and $\bar{\gamma}_i$ is the average SNR. $\gamma_i(\phi_i^z(t))$ is the SNR between device i and the UAV using $\phi_i^z(t)$, as given by

$$\gamma_i(\phi_i^z(t)) = \frac{\|\mathbf{h}_i^z(t)\|^2 P_i^z(t)}{\sigma_0^2}, \tag{3}$$

where $P_i^z(t)$ is the transmit power of device i, and σ_0^2 is noise power at the UAV. In particular, for illustration convenience, we consider a special case of the Nakagami-m model in this paper where $m = 1$ [35, 36, 37]. Note that the proposed deep reinforcement learning approach is generic, and can work with other Nakagami fading channel model with any m values. The required transmit power of the ground device depends on $\phi_i^z(t)$ and $\mathbf{h}_i^z(t)$, and can be given by [12]

$$P_i^z(t) \approx \frac{\kappa_2^{-1} \ln \frac{\kappa_1}{\varepsilon}}{\|\mathbf{h}_i^z(t)\|^2}(2^{\phi_i^z(t)} - 1), \tag{4}$$

where κ_1 and κ_2 are channel related constants.

According to [38], the MPT efficiency is jointly decided by the distance between the MPT transmitter and the receiver, and their antenna alignment. Therefore, the power transferred to device i at t in lap z via MPT is given by

$$\tilde{P}_i^z(t) = \omega(d_i^z(t), \theta_i^z(t)) P_{\mathrm{UAV}}^{tx} \|\mathbf{h}_i^z(t)\|^2, \tag{5}$$

where $\|\cdot\|$ stands for norm. $\omega(d_i^z(t), \theta_i^z(t))$ is the MPT efficiency factor given the distance $d_i^z(t)$ and the MPT transceiver alignment $\theta_i^z(t)$ between the ground device and the UAV. The transmit power of the UAV is fixed and set to be P_{UAV}^{tx}, so that the operations at the UAV can keep simple.

Each of the wirelessly powered ground devices harvests energy from the UAV. The rechargeable battery is finite with the capacity of E Joules, and the battery overflows if overcharged. Moreover, the battery readings are continuous variables with variance difficult to be traced in real-time. Therefore, to improve the mathematical tractability of the problem and for illustration convenience, the continuous battery is discretized into K levels, as $0 < \mathcal{E} < 2\mathcal{E} < \cdots < K\mathcal{E} = E$. $e_{i,z}(t) \in \{0, \mathcal{E}, 2\mathcal{E}, \cdots, K\mathcal{E}\}$ [39]. In other words, the battery level of the ground device is lower rounded to the closest discrete level.

We also assume that all the ground devices have the same battery size, data queue size, packet length, and the BER requirement. However, our proposed resource allocation approach can be extended to a heterogeneous network setting, where the complexity of the resource allocation problem may grow as the result of an increased number of MDP states.

3.2. Communication protocol

Figure 2 illustrates the communication protocol for UAV-assisted online MPT and data collection. Specifically, each communication frame that contains a number of time slots is allocated to the ground device for MPT and data transmission. The ground device selection is determined by the UAV, using DRL-SA which will be illustrated in Section 4. Then, the UAV broadcasts a short beacon message to the selected ground device for data transmission. On the reception of the ground device's data, the UAV is aware of the information for MPT, i.e., $d_i^z(t)$, $\theta_i^z(t)$, and $\mathbf{h}_i^z(t)$. Moreover, a control segment of the device's data packet contains $q_{i,z}(t)$ and $e_{i,z}(t)$. The overhead of this control segment is small. For example, consider $e_{i,z}(t)$ of 100 and

$q_{i,z}(t)$ of 100 packets, the overhead is only 12 bits, much smaller than the size of the data packet. Therefore, we assume that the transmission time and the energy consumption of the control segment are negligible.

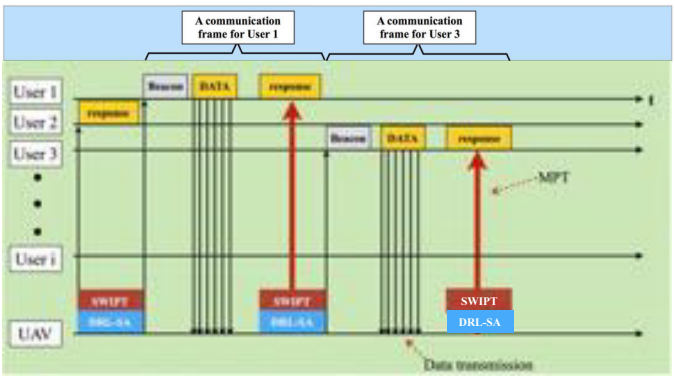


Figure 2: The communication protocol for UAV-assisted online MPT and data collection.

The UAV processes the received data packets online, and responds to the ground device's requests, e.g., providing network access services, or online information query. SWIPT is utilized to transmit the response to the ground device and charge its battery via MPT simultaneously. Meanwhile, DRL-SA is conducted by the UAV to schedule the other ground device for MPT and data transmission in the next communication frame.

4. Deep Reinforcement Learning for UAV-assisted online MPT and Data Collection

In this section, we first present the problem formulation based on MDP, and provide a reinforcement learning solution to the problem. Due to the curse-of-dimensionality of reinforcement learning, we propose a new deep reinforcement learning based scheduling algorithm to minimize overall data packet loss of the ground devices, by optimally deciding the device to be charged and interrogated for data collection, and the instantaneous patrolling velocity of the UAV. The modulation scheme of the ground device is also optimized to maximize the harvested energy. Finally, the optimality of the deep reinforcement learning approach is analyzed.

4.1. MDP formulation

The UAV takes the actions of the ground device selection, the modulation scheme, and the instantaneous patrolling velocity of the UAV. Each action depends on the current network state, i.e, the battery level $e_{i,z}(t)$ and queue length $q_{i,z}(t)$ of every device i, the channel quality $\mathbf{h}_i^z(t)$, and the location of the UAV $\zeta_z(t)$ along the flight trajectory. The actions also account for the potential influence on the future evolution of the network. Particularly, the current action that the UAV takes can affect the future battery level and queue length of every device and, in turn, influence the future actions to be taken. Such actions are a discrete-time stochastic control process which is partly random (due to the random and independent arrival/queueing process of sensory data at every device) and partly under the control of the decision-making UAV.

The action can be optimized in a sense that the optimality in regards of a specific metric, e.g., packet loss from queue overflows and unsuccessful data transmissions, is achieved in the long term over the entire stochastic control process (rather than myopically at an individual time slot). Motivated by this, we consider an MDP formulation for which the actions are chosen in each state to minimize a long-term objective. An MDP is defined by the quadruplet $\langle \mathcal{S}, \mathcal{A}, C\Big\{\mathcal{S}_\beta\Big|\mathcal{S}_\alpha, k\Big\}, \Pr\Big\{\mathcal{S}_\beta\Big|\mathcal{S}_\alpha, k\Big\}\rangle$, where $\mathcal{S}$ is the set of possible states; $\mathcal{A}$ is the set of actions; $C\Big\{\mathcal{S}_\beta\Big|\mathcal{S}_\alpha, k\Big\}$ is the immediate cost yielded when action k is taken at state $\mathcal{S}_\alpha$ and the following state changes to $\mathcal{S}_\beta$; and $\Pr\Big\{\mathcal{S}_\beta\Big|\mathcal{S}_\alpha, k\Big\}$ denotes the transition probability from state $\mathcal{S}_\alpha$ to state $\mathcal{S}_\beta$ when action k is taken.

The resource allocation problem of interest in UAV-assisted online MPT and data collection can be formulated as a discrete-time MDP, where each state $\mathcal{S}_\alpha$ collects the battery levels and queue lengths of the ground devices, the channel quality between the UAV and device i, and the location of the UAV, i.e., $\{(q_{i,z}(t), e_{i,z}(t), \mathbf{h}_i^z(t), \zeta_z(t)), i = 1, 2, ..., I; z = 1, 2, ..., Z\}$. The size of the state space, i.e., the number of such states, is $\Big((K+1)(D+1)\Big)IZ + HZ + VZ$, where H is the number of channel states, and V is the number of waypoints on the UAV's trajectory. The action $\mathcal{A}$ to be taken is to schedule one device to transmit data to the UAV at time slot t, while specifying the modulation of the device and the instantaneous patrolling velocity of the UAV, i.e., $\mathcal{A} \in \Big\{(i, \phi_i^z(t), v^z(t)) : i = 1, 2, ..., I; z = 1, 2, ..., Z; \phi_i^z(t) \in$

$\{1, 2, ..., \Phi\}; v_{\min} \leq v^z(t) \leq v_{\max}\Big\}$. The size of the action set is $IZ|v^z(t)|$, where $|v^z(t)|$ stands for the cardinality of the set $[v_{\min}, v_{\max}]$.

To illustrate the proposed MDP model, Figure 3 presents an example of transition diagram with 24 MDP states in one lap of the UAV's flight, where $I = 1$, $Z = 1$, $K = 1$, $D = 1$, $H = 2$ (e.g., -10dB, 30dB), and $V = 4$. The vertices stand for all possible states in MDP, i.e., $\{(q_{i,z}(t), e_{i,z}(t), \mathbf{h}_i^z(t), \zeta_z(t))\}$. The edges show the transition from each state to other states according to $\Pr\left\{\mathcal{S}_\beta \middle| \mathcal{S}_\alpha, k\right\}$. The state transition depends on the change of $\{(q_{i,z}(t), e_{i,z}(t), \mathbf{h}_i^z(t)\}$ of the ground device and $\zeta_z(t)$ along the trajectory of the UAV. In other words, the next state of $\{(q_{i,z}(t_1), e_{i,z}(t_1), \mathbf{h}_i^z(t_1), \zeta_z(t_1))\}$ can be one of the states at $\zeta_z(t_2)$, $\zeta_z(t_3)$, or $\zeta_z(t_4)$. For example, for t_1, the next state of $\{(q_{i,z}(t_1) = 1, e_{i,z}(t_1) = 1, \mathbf{h}_i^z(t_1) = -10\text{dB}, \zeta_z(t_1))\}$ can be $\{(q_{i,z}(t_2) = 2, e_{i,z}(t_2) = 2, \mathbf{h}_i^z(t_2) = -10\text{dB}, \zeta_z(t_2))\}$, if device i is selected, but the data collection is not successful; or $\{(q_{i,z}(t_2) = 1, e_{i,z}(t_2) = 2, \mathbf{h}_i^z(t_2) = -10\text{dB}, \zeta_z(t_2))\}$, if the data collection is successful. Note that Figure 3 gives a small-scale example of the transition of one of the states, i.e., $\{(q_{i,z}(t) = 1, e_{i,z}(t) = 1, \mathbf{h}_i^z(t) = -10\text{dB}, \zeta_z(t))\}$. The UAV's trajectory can have hundreds of waypoints and the model can contain over 8×10^5 MDP states, as configured in Section 5, which leads to an extremely complex state transition diagram.

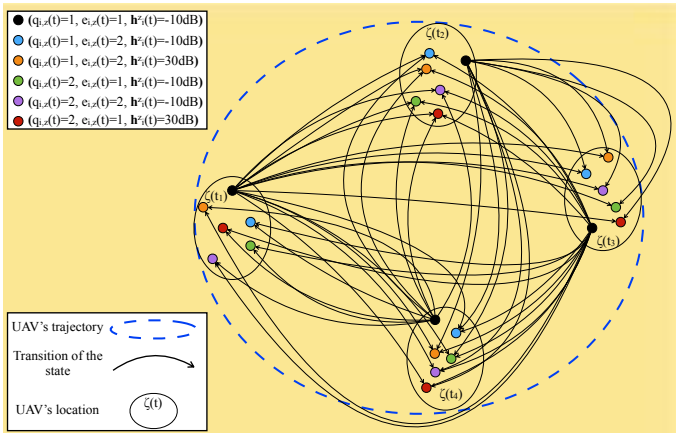


Figure 3: An example of a transition diagram of 24 MDP states, which has 4 waypoints on the UAV's trajectory.

The optimal policy in MDP can be determined by classical approaches, e.g., value iteration (computes and improves the action-value function esti-

mate iteratively) or policy iteration (redefines the policy at each step and computes the value according to this new policy). However, these two methods require that the transition probability and the cost of all states have been accurately known. In contrast, this paper is interested in a practical scenario where the UAV has no a-prior knowledge on $C\Big\{\mathcal{S}_\beta\Big|\mathcal{S}_\alpha,k\Big\}$ and $\Pr\Big\{\mathcal{S}_\beta\Big|\mathcal{S}_\alpha,k\Big\}$.

4.2. Q-learning

Q-learning, one of the reinforcement learning techniques, can obtain the optimal resource allocation when the transition and/or cost functions are unknown while minimizing the long-term expected accumulated discounted costs (i.e., the expected packet loss of the ground devices) [10]. We define the objective function of the resource allocation problem as $v(S_\alpha)$, which gives

$$v(S_\alpha)=\min_{\pi\in\Pi}\ \mathbb{E}^{\pi}_{\mathcal{S}_\alpha}\left\{\sum_{t=1}^{\infty}\delta^{t-1}C\Big\{\mathcal{S}_\beta\Big|\mathcal{S}_\alpha,k\Big\}\right\}, \tag{6}$$

where $\delta\in[0,1]$ is a discount factor for future states. $C\Big\{\mathcal{S}_\beta\Big|\mathcal{S}_\alpha,k\Big\}$ is the cost from state $\mathcal{S}_\alpha$ to $\mathcal{S}_\beta$ when action k is carried out. $\mathbb{E}^{\pi}_{\mathcal{S}_\alpha}\{\cdot\}$ denotes the expectation with respect to policy π and state $\mathcal{S}_\alpha$. Furthermore, the action-value function $Q\Big\{\mathcal{S}_\beta\Big|\mathcal{S}_\alpha,k\Big\}$ defines the expected cost after observing state $\mathcal{S}_\alpha$ and taking action k [40]. By performing the optimal action k^*, the optimal action-value function can be expressed as a combination of the expected cost and the minimum value of $Q\Big\{\mathcal{S}_{\beta'}\Big|\mathcal{S}_\beta,k'\Big\}$, where $\mathcal{S}_{\beta'}$ is the next state of $\mathcal{S}_\beta$, and k' is the next action of k. Thus, we have

$$\begin{aligned}Q^*\Big\{\mathcal{S}_\beta\Big|\mathcal{S}_\alpha,k^*\Big\}&=(1-\varrho)Q^*\Big\{\mathcal{S}_\beta\Big|\mathcal{S}_\alpha,k^*\Big\}+\\&\varrho\Big[C\Big\{\mathcal{S}_\beta\Big|\mathcal{S}_\alpha,k^*\Big\}+\delta\min\nolimits_{k'\in\mathcal{A}}Q\Big\{\mathcal{S}_{\beta'}\Big|\mathcal{S}_\beta,k'\Big\}\Big].\end{aligned} \tag{7}$$

where $\varrho\in(0,1]$ (a small positive fraction) indicates learning rate. As observed in (7), the convergence rate of $Q\Big\{\mathcal{S}_\beta\Big|\mathcal{S}_\alpha,k\Big\}$ to $Q^*\Big\{\mathcal{S}_\beta\Big|\mathcal{S}_\alpha,k^*\Big\}$ depends on the discount factor δ. Namely, the convergence rate of the action-value function increases with the discount factor δ.

However, Q-learning suffers from the well-known curse of dimensionality. Thus, Q-learning is impractical for the resource allocation problem in the

UAV-assisted online MPT and data collection due to a large state and action space in which the time needed for Q-learning to converge grows immeasurably. In addition, Q-learning is unstable when combined with non-linear approximation functions such as neural networks [41].

4.3. Deep reinforcement learning based approach

To circumvent the curse-of-dimensionality of reinforcement learning, we propose an on-board deep Q-network for the UAV to optimize the resource allocation online by approximating the optimal action-value function. As

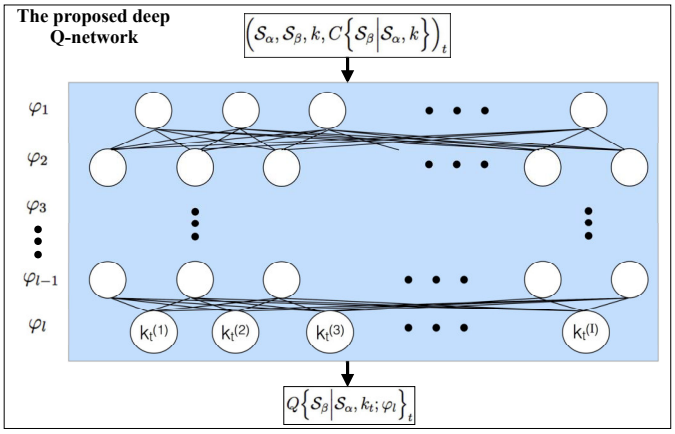


Figure 4: Schematic illustration of the proposed on-board deep Q-network.

shown in Figure 4, the action-value function in (7) is represented by the proposed on-board deep Q-network, which takes the current network state and action as the input and derives the corresponding action-value function. Given the total number of iterations Ω, $Q\left\{\mathcal{S}_\beta \middle| \mathcal{S}_\alpha, k; \varphi_l\right\}$ is approximated by adapting a set of weights φ_l, where $l \leq \Omega$. By optimizing φ_l, the outputs of the deep Q-network, i.e., the approximated $Q\left\{\mathcal{S}_\beta \middle| \mathcal{S}_\alpha, k; \varphi_l\right\}$, can be minimized.

In our on-board deep Q-network, experience replay is conducted to randomize over the states and the actions of MDP at each time-step t, i.e., $\left(\mathcal{S}_\alpha, \mathcal{S}_\beta, k, C\left\{\mathcal{S}_\beta \middle| \mathcal{S}_\alpha, k\right\}\right)_t$, thereby removing correlations in the observation sequence and smoothing over changes in the data distribution. Specifically, $\left(\mathcal{S}_\alpha, \mathcal{S}_\beta, k, C\left\{\mathcal{S}_\beta \middle| \mathcal{S}_\alpha, k\right\}\right)_t$ is stored in a data set $\mathbb{D}[\cdot]$, pooled over many episodes (where an episode ends when a terminal state is reached) into

$$\Gamma_l(\varphi_l) = \mathbb{E}_{\left(\mathcal{S}_\alpha,\mathcal{S}_\beta,k,C\left\{\mathcal{S}_\beta\middle|\mathcal{S}_\alpha,k\right\}\right)\sim\mathbb{D}}\left[\left(C\left\{\mathcal{S}_\beta\middle|\mathcal{S}_\alpha,k\right\}+\delta\min_{k'\in\mathcal{A}}Q\left\{\mathcal{S}_{\beta'}\middle|\mathcal{S}_\beta,k';\varphi_{l-1}\right\}-Q\left\{\mathcal{S}_\beta\middle|\mathcal{S}_\alpha,k;\varphi_l\right\}\right)^2\right], \tag{8}$$

$$\nabla\Gamma_l(\varphi_l) = \mathbb{E}_{\left(\mathcal{S}_\alpha,\mathcal{S}_\beta,k,C\left\{\mathcal{S}_\beta\middle|\mathcal{S}_\alpha,k\right\}\right)}\Big[\Big(C\left\{\mathcal{S}_\beta\middle|\mathcal{S}_\alpha,k\right\}+\delta\min_{k'\in\mathcal{A}}Q\left\{\mathcal{S}_{\beta'}\middle|\mathcal{S}_\beta,k';\varphi_{l-1}\right\}- Q\left\{\mathcal{S}_\beta\middle|\mathcal{S}_\alpha,k;\varphi_l\right\}\Big)\nabla_{\varphi_l}Q\left\{\mathcal{S}_\beta\middle|\mathcal{S}_\alpha,k;\varphi_l\right\}\Big]. \tag{9}$$

---

an experience replay memory. Moreover, samples (or minibatches) of the experience in the deep Q-network are accordingly updated during learning. The deep Q-network is trained by adjusting the weight φ_l at iteration l so as to minimize a sequence of loss functions $\Gamma_l(\varphi_l)$; see (8), where φ_l and φ_{l-1} are the weights at iterations l and $(l-1)$, respectively. At each iteration of minimizing $\Gamma_l(\varphi_l)$, the weight φ_{l-1} from iteration $(l-1)$ is fixed. Thus, the subproblem of learning $\Gamma_l(\varphi_l)$ at iteration l ($l \leq \Omega$) defines $C\left\{\mathcal{S}_\beta\middle|\mathcal{S}_\alpha,k\right\}+\delta\min_{k'\in\mathcal{A}}Q\left\{\mathcal{S}_{\beta'}\middle|\mathcal{S}_\beta,k';\varphi_{l-1}\right\}$. For each sample (or minibatch), the current weight φ_l, gradient descent is learned to derive weights φ_l, which iteratively computes the gradient $\nabla\Gamma_l(\varphi_l)$, and updates the neural network's weights to reach the global minimum. We differentiate $\Gamma_l(\varphi_l)$ with regards to φ_l, and obtain (9).

Algorithm 1 presents the proposed DRL-SA scheme, which optimizes the actions based on the deep Q-network to solve the online resource allocation problem. Specifically, U is a global parameter for counting the total number of updates to the UAV, rather than counting the updates from the local learner. The ϵ-greedy policy is utilized to balance the action-value function minimization based on the knowledge already known, as well as trying new actions to obtain knowledge unknown. In particular, the optimal action of determining $\phi_i^z(t)$ for maximizing the harvested energy (see the Appendix) is carried out once the optimal ground device is selected from the deep Q-network.

Note that the optimal solutions to a small-scale MDP problem of interest

can be solved by Q-learning in which the series of the optimal actions are obtained in a state machine, given the optimal decisions at each of the states. However, Q-learning is impractical for the complex resource allocation problem that contains a large state and action space, due to the well-known curse of dimensionality. As observed in Algorithm 1, our on-board deep Q-network maintains two separate Q-networks $Q\Big\{\mathcal{S}_\beta\Big|\mathcal{S}_\alpha, k; \varphi_l\Big\}$ and $Q\Big\{\mathcal{S}_\beta\Big|\mathcal{S}_\alpha, k; \varphi_{l-1}\Big\}$ with current weights φ_l and the old weights φ_{l-1}, respectively [42]. φ_l can be updated many times per time-step, and copied into φ_{l-1} after every U iterations. At every update iteration, the deep Q-network is trained to minimize the mean-squared Bellman error, by minimizing the loss function $\Gamma_l(\varphi_l)$. Therefore, the proposed DRL-SA can achieve the optimality asymptotically, with the growing size of the deep Q-network.

5. Numerical Results and Discussions

In this section, we first present network configurations and performance metrics. Then, we evaluate the network cost of the proposed DRL-SA scheme with regards to the network size, data queue length, and learning discount factors. Here, the network cost defines the amount of packet loss due to the data queue overflow and the failed transmission from the ground device to the UAV. We also show the UAV's velocity in terms of the network size, as well as the data queue length.

5.1. Implementation of DRL-SA

The simulation platform is an INSYS Server running 64-bit Ubuntu 16.04 LTS, with 4-core Intel i7-6700K 4GHz CPUs and 16G memory. DRL-SA is implemented in Python 3.5 by using Google TensorFlow [43] (the symbolic math library for numerical computation) with Keras [44] (the Python deep learning library). We develop the on-board deep Q-network in DRL-SA according to the following steps:

- Initialize the configurations. Each device generates 100 data packets, where the packet length is 128 bytes. The transmit power of the UAV is 100 milliwatts. ε is set to 0.05%, however, the value of ε can be configured depending on the traffic type and quality-of-service (QoS) requirement of the user's data, as well as the transmission capability of the UAV. Other simulation parameters are listed in Table 2.

- Set up the architecture of the deep Q-network. Three fully connected hidden layers are created by using *tensorflow.layers.dense(inputs, dimensionality of the output space, activation function)*. Then, *tensorflow.train.AdamOptimizer().minimize(loss function)* is called to minimize the loss function. In particular, the optimizer is imported from the Keras library.

- Build the memory for the experience replay. For online training the deep Q-network, the memory stores the learning outcomes, *a.k.a* experience at every step, using the quadruplet <*state, action, cost, next_state*>. The deep Q-network updates the memory by calling the function *memory.add_sample((state, action, cost, next_state))*, and retrieves the experiences by using *memory.sample(batch size)*.

- Create a deep Q-network agent, and kick off the learning. The agent is configured and compiled to take actions, and observe the cost and the new state. A TensorFlow session is created by implementing *tensorflow.Session()* to execute the learning and evaluate the learning progress.

For performance comparison, the proposed DRL-SA scheme is compared with two other on-board online scheduling policies as

- Random scheduling policy (RSA). The UAV randomly schedules one ground device to collect data and transfer power at each time slot. The resource allocation is independent of battery and data queue of the ground device, channel variation, or UAV's trajectory.

- Longest queue scheduling policy (LQSA). This is a greedy algorithm, where the scheduling is based on the data queue length of the ground device. The device with the longest queue is given the highest priority to transmit data and harvest power. For LQSA, it is assumed that the up-to-date information of the data queue length at each device is known by the UAV.

Both RSA and LQSA are implemented in Matlab. Their scheduling policies are not obtained by the deep Q-network. Note that the existing learning-based approaches, e.g., QSA [10] or SARSA [45], can only solve the small-scale scheduling problem, and do not apply to the resource allocation in UAV-assisted online MPT and data collection. In addition, the resource allocation

optimization schemes, e.g., EHMDP [11] or EACH [46], require the a-prior knowledge about the network and have to be conducted in offline.

5.2. Performance evaluation

5.2.1. Network size

We assess the performance of DRL-SA when the number of ground devices enlarges from 50 to 200. Figure 5 shows the network cost at each episode, given $I = 120, 150$ and 180. The network cost of DRL-SA is high at the beginning of the learning process. With an increasing number of episodes, the network cost drops significantly until it reaches a relatively stable value. It confirms the fact that deep Q learning gradually converges after a number of episodes. Moreover, the average network cost is smaller when the number of ground devices is 120, as compared to the cases where the number of ground devices is 150 and 180, since the increasing number of ground devices leads the data queues to increasingly overflow.

Figure 6(a) studies the network cost with an increasing number of ground devices, where the data queue length of DRL-SA is set to 50 or 10. In general, the proposed DRL-SA is able to reduce the network cost to a greater extent than RSA and LQSA. Particularly, when $I = 200$ and $D = 10$, the maximum network cost of DRL-SA is lower than RSA and LQSA by around 82.8% and 69.2%, respectively. The performance gains keep growing with I. The reason is that DRL-SA learns the ground devices' energy consumption and data queue states, so that the scheduling of MPT and data communications can minimize the data packet loss of the entire network.

Figure 6(b) illustrates the packet loss rate, which is the ratio of the network cost and the total number of data packets of all the ground devices. Specifically, DRL-SA has a similar packet loss rate to LQSA, when there are 50 devices in the network. However, from $I = 80$ to 200, DRL-SA outperforms the two non-learning-based algorithms. In particular, when $I = 200$, DRL-SA with $D = 10$ achieves 53%, and 25% lower packet loss rates than RSA and LQSA, respectively. In Figure 6(b), we also see that the packet loss rate of DRL-SA only slightly grows about 2% from $I = 50$ to 200. In other words, adding more devices to the network does not result in a critical data packet loss. This is because DRL-SA takes the actions to adapt the instantaneous velocity of the UAV to ensure the connection time and channel quality between the device and the UAV, hence minimizing the data packet loss. As shown in Figure 6(c), the patrolling velocity raises with an increase of I. This is reasonable because the UAV needs to transfer power

and collect data from more devices in order to reduce their data queue overflow. Furthermore, it is also observed that increasing the data queue length of the devices downgrades the patrolling velocity of the UAV. The reason is that a larger data queue can hold more packets. This allows an extended data transmission time between the ground device and the UAV and in turn, reduces the patrolling velocity.

5.2.2. Data queue length of ground devices

We consider different data queue lengths of the ground devices, i.e., D, where the number of ground devices is set to 140 or 300. Figures 6(d) and (e) depict the network cost and packet loss rate with respect to the maximum data queue length of the ground devices, respectively. We observe that DRL-SA achieves lower network costs and packet loss rates than RSA and LQSA, while DRL-SA outperforms RSA with substantial gains of 82.7% and 69% when $D = 20$ and $I = 140$. Moreover, given $I = 300$, from $D = 20$ to $D = 60$, the network cost and packet loss rate of DRL-SA drop by 62.2% and 13.2%. This confirms that DRL-SA significantly reduces data queue overflow for all the ground devices when enlarging their data queue length. In terms of the patrolling velocity of the UAV, it is observed in Figure 6(f) that DRL-SA reduces the velocity from 18.5 m/s to 14.1 m/s, and from 8.1 m/s to 6.2 m/s, given 300 and 140 ground devices in the network, respectively.

Figure 6 implies a tradeoff between the data packet loss and the battery lifetime of the UAV. Specifically, an increase of the network size, or a decrease of data queue length, results in a growth of the packet loss due to the data queue overflow. In this case, the patrolling velocity of the UAV can be increased so that more ground devices harvest energy and transmit data, which reduces the packet loss of all the devices. However, accelerating the patrolling velocity speeds up draining the energy of the UAV's battery, and reduces the lifetime of the network. Therefore, the network size and the data queue length need to be balanced, so as to maintain a sustainable UAV-assisted network while reducing the data packet loss.

5.2.3. The actions of the UAV

To further reveal the impact of network size and data queue length on the actions of the UAV, Figure 7 demonstrates the patrolling velocity allocated by the proposed DRL-SA, with respect to the episodes, given $(I = 50, D = 60)$ and $(I = 200, D = 20)$. We can see that DRL-SA with $(I = 200, D = 20)$ allocates 8.7 m/s higher patrolling velocity to the UAV than the one with $(I =$

$50, D = 60$) on average. This confirms that the patrolling velocity drops with a decrease of network size or growth of data queue length, due to the data queue overflow, as explained in Figure 6. Moreover, the result in Figure 7 also presents that DRL-SA allocates the patrolling velocity adapting to the time-varying network state. As observed, the patrolling velocity allocation converges with an increase in the learning time (i.e., episodes).

5.2.4. Discount factor of learning

Since DRL-SA utilizes deep Q learning to approximate the Q function with asymptotic convergence, the convergence time can be affected by the discount factor in the learning process. Figure 8 plots the network cost of DRL-SA with regards to the episodes given the discount factor $\delta = 0.1, 0.5$ and 0.99. I and D are set to 300 ground devices and 20, respectively. The other settings are provided in Section 5.1.

As observed, DRL-SA has a high network cost at the beginning of the learning. The performance improves with the increase of the episodes due to deep reinforcement learning. In particular, the network cost of DRL-SA with $\delta = 0.99$ quickly falls to 0 from episode 1 to episode 359, and the performance remains relatively stable afterward. However, the convergence of DRL-SA with $\delta = 0.5$ or 0.1 requires more than 475 or 500 episodes. Therefore, the result in Figure 8 indicates that the convergence rate of DRL-SA grows with δ. In other words, a high discount factor of learning accelerates the learning process of the deep Q-network, as confirmed by (7).

6. Conclusion

In this paper, we focus on online MPT and data collection in the presence of on-board control of the patrolling velocity of the UAV, for preventing battery drainage and data queue overflow of the sensing devices. The problem is formulated as the MDP, with the states of battery level and data queue length of the ground devices, channel conditions, and waypoints given the trajectory of the UAV. We propose an on-board deep Q-network that can enlarge the state and action space of the MDP to minimize the data packet loss of the entire system. Based on deep reinforcement learning, DRL-SA is developed to learn the optimal resource allocation strategy asymptotically through online training at the on-board deep Q-network, where the selection of the ground device, modulation scheme, and instantaneous patrolling velocity of the UAV are jointly optimized. Moreover, DRL-SA carries out

experience replay to reduce expansion of the state space in which the algorithm's scheduling experiences at each time step are stored in a data set.

The proposed DRL-SA scheme is implemented by using Keras deep learning library with Google TensorFlow as the backend engine. Numerical results demonstrate that DRL-SA reduces packet loss by at least 69.2%, as compared to the existing non-learning greedy algorithms.

Acknowledgements

This work was partially supported by National Funds through FCT/MCTES (Portuguese Foundation for Science and Technology), within the CISTER Research Unit (CEC/04234); also by the Operational Competitiveness Programme and Internationalization (COMPETE 2020) through the European Regional Development Fund (ERDF) and by national funds through the FCT, within project POCI-01-0145-FEDER-029074 (ARNET).

Appendix A. [Optimizing $\phi_i^z(t)$]

To minimize the packet loss stemming from insufficient energy, $\phi_i^z(t)$ of device i is to be chosen to maximize the energy harvested during a contact time with the UAV, with a length of $\widehat{T}_i^z(t)$. The optimal modulation of the ground device, $\phi_i^z(t)^\star$, is independent of the battery level and the queue length. This is because $\phi_i^z(t)^\star$ is selected to maximize the increase of the battery level at device i, under the bit error rate requirement ϵ_i for the packet transmitted. As a result, $\phi_i^z(t)$ can be decoupled from $\mathcal{A}$, and optimized in prior by

$$\phi_i^z(t) = \arg \max_{\phi=1,\cdots,\Phi} \left\{ (\widehat{T}_i^z(t) - \frac{B}{\phi W}) P_i^z(t) - \frac{B}{\phi W} P_i^z(\phi) \right\}, \tag{A.1}$$

the right-hand side (RHS) of which, by substituting (5) and (4), can be rewritten as

$$\max_{\phi=1,\cdots,\Phi} \left\{ (\widehat{T}_i^z(t) - \frac{B}{\phi W}) \omega(d_i^z(t), \theta_i^z(t)) P_{\text{UAV}}^{tx} \|\mathbf{h}_i^z(t)\|^2 - \right.$$
$$\left. \frac{B\kappa_2^{-1} \ln(\frac{\kappa_1}{\epsilon})}{\|\mathbf{h}_i^z(t)\|^2 \phi W} \left(2^\phi - 1\right) \right\}, \tag{A.2}$$

where W is the bandwidth of the uplink data transmission, $\frac{1}{W}$ is the duration of an uplink symbol, and $\frac{B}{\phi W}$ is the duration of uplink data transmission.

$(\widehat{T}_i^z(t) - \frac{B}{\phi W})$ is the rest of the time slots used for downlink WPT, and $\widehat{T}_i^z(t)$ is the contact time between the ground device and the UAV in the time slot, which is affected by the patrolling velocity of the UAV. Thus, we have

$$\widehat{T}_i^z(t) = \frac{2\sqrt{(d_i^z(t))^2 - (b^z)^2}}{v^z(t)}, \tag{A.3}$$

where b^z is the altitude of the UAV at lap z. We assume that the UAV maintains the same altitude and the same heading in each lap.

By using the first-order necessary condition of the optimal solution, we have

$$\frac{d}{d\phi}((\widehat{T}_i^z(t) - \frac{B}{\phi W})\omega(d_i^z(t), \theta_i^z(t))P_{\text{UAV}}^{tx}\|\mathbf{h}_i^z(t)\|^2 - \frac{B\kappa_2^{-1}\ln(\frac{\kappa_1}{\epsilon})}{\|\mathbf{h}_i^z(t)\|^2\phi W}\big(2^\phi - 1\big)) = 0, \tag{A.4}$$

$$\phi^{-2}\frac{B}{W}\omega(d_i^z(t), \theta_i^z(t))P_{\text{UAV}}^{tx}\|\mathbf{h}_i^z(t)\|^2 - \frac{B\kappa_2^{-1}\ln(\frac{\kappa_1}{\epsilon})}{\|\mathbf{h}_i^z(t)\|^2 W}\cdot (\phi^{-1}2^\phi \text{In}2 - \phi^{-2}2^\phi) - \frac{B\kappa_2^{-1}\ln(\frac{\kappa_1}{\epsilon})}{\|\mathbf{h}_i^z(t)\|^2 W}\phi^{-2} = 0. \tag{A.5}$$

The ϕ values are then given as follows:

$$\phi 2^\phi \text{In}2 - 2^\phi = \frac{B}{W}\omega(d_i^z(t), \theta_i^z(t))P_{\text{UAV}}^{tx}\|\mathbf{h}_i^z(t)\|^2\frac{\|\mathbf{h}_i^z(t)\|^2 W}{B\kappa_2^{-1}\ln(\frac{\kappa_1}{\epsilon})} - 1. \tag{A.6}$$

Since the left-hand side (LHS) of (A.6) monotonically increases with ϕ, the optimal value $\phi^\star$ can be obtained by applying a bisection search method, and evaluating the two closest integers about the fixed point of the bisection method [47]. Specifically, $\phi_- = 1$ and $\phi_+ = \Phi$ are initialized. Each iteration of the bisection method contains 4 steps applied over the range of $\phi = [1, \Phi]$, as follows.

- The midpoint of the modulation interval $[\phi_-; \phi_+]$ is calculated, which gives $\phi_{\text{mid}} = \frac{\phi_- + \phi_+}{2}$.
- Substitute ϕ_{mid} into (A.6) to obtain the function value $f(\phi_{\text{mid}})$.

- If the convergence is attained (that is, the modulation interval or $|f(\phi_{\text{mid}})|$ cannot be further reduced), return ϕ_{mid} and stop the iteration.

- Replace either $(\phi_-, f(\phi_-))$ or $(\phi_+, f(\phi_+))$ with $(\phi_{\text{mid}}, f(\phi_{\text{mid}}))$.

[1] T. Tomic, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I. L. Grixa, F. Ruess, M. Suppa, D. Burschka, Toward a fully autonomous UAV: Research platform for indoor and outdoor urban search and rescue, IEEE robotics & automation magazine 19 (3) (2012) 46–56.

[2] C. Luo, P. Ward, S. Cameron, G. Parr, S. McClean, Communication provision for a team of remotely searching UAVs: A mobile relay approach, in: Globecom Workshops (GC Wkshps), IEEE, 2012, pp. 1544–1549.

[3] S. Waharte, N. Trigoni, Supporting search and rescue operations with UAVs, in: International Conference on Emerging Security Technologies (EST), IEEE, 2010, pp. 142–147.

[4] H. Wang, G. Ding, F. Gao, J. Chen, J. Wang, L. Wang, Power control in UAV-supported ultra dense networks: Communications, caching, and energy transfer, IEEE Communications Magazine 56 (6) (2018) 28–34.

[5] Y. Zeng, R. Zhang, T. J. Lim, Wireless communications with unmanned aerial vehicles: opportunities and challenges, IEEE Communications Magazine 54 (5) (2016) 36–42.

[6] T. D. P. Perera, D. N. K. Jayakody, S. K. Sharma, S. Chatzinotas, J. Li, Simultaneous wireless information and power transfer (SWIPT): Recent advances and future challenges, IEEE Communications Surveys & Tutorials 20 (1) (2017) 264–302.

[7] S. Yin, J. Tan, L. Li, UAV-assisted cooperative communications with wireless information and power transfer, arXiv preprint arXiv:1710.00174.

[8] S. Yin, Y. Zhao, L. Li, UAV-assisted cooperative communications with time-sharing SWIPT, in: 2018 IEEE International Conference on Communications (ICC), IEEE, 2018, pp. 1–6.

[9] L.-J. Lin, Reinforcement learning for robots using neural networks, Tech. rep., Carnegie-Mellon Univ Pittsburgh PA School of Computer Science (1993).

[10] K. Li, W. Ni, M. Abolhasan, E. Tovar, Reinforcement learning for scheduling wireless powered sensor communications, IEEE Transactions on Green Communications and Networking.

[11] K. Li, W. Ni, L. Duan, M. Abolhasan, J. Niu, Wireless power transfer and data collection in wireless sensor networks, IEEE Transactions on Vehicular Technology 67 (3) (2018) 2686–2697.

[12] K. Li, W. Ni, X. Wang, R. P. Liu, S. S. Kanhere, S. Jha, Energy-efficient cooperative relaying for unmanned aerial vehicles, IEEE Transactions on Mobile Computing (6) (2016) 1377–1386.

[13] S. Koulali, E. Sabir, T. Taleb, M. Azizi, A green strategic activity scheduling for UAV networks: A sub-modular game perspective, IEEE Communications Magazine 54 (5) (2016) 58–64.

[14] X. Li, Y. D. Zhang, Multi-source cooperative communications using multiple small relay UAVs, in: IEEE GLOBECOM Workshops, 2010, pp. 1805–1810.

[15] S. Zhang, H. Zhang, Q. He, K. Bian, L. Song, Joint trajectory and power optimization for UAV relay networks, IEEE Communications Letters 22 (1) (2018) 161–164.

[16] J. Baek, S. I. Han, Y. Han, Optimal resource allocation for non-orthogonal transmission in UAV relay systems, IEEE Wireless Communications Letters 7 (3) (2018) 356–359.

[17] Z. M. Fadlullah, D. Takaishi, H. Nishiyama, N. Kato, R. Miura, A dynamic trajectory control algorithm for improving the communication throughput and delay in UAV-aided networks, IEEE Network 30 (1) (2016) 100–105.

[18] D. H. Choi, S. H. Kim, D. K. Sung, Energy-efficient maneuvering and communication of a single UAV-based relay, IEEE Transactions on Aerospace and Electronic Systems 50 (3) (2014) 2320–2327.

[19] F. Jiang, A. L. Swindlehurst, Optimization of UAV heading for the ground-to-air uplink, IEEE Journal on Selected Areas in Communications 30 (5) (2012) 993–1005.

[20] P. Zhan, K. Yu, A. L. Swindlehurst, Wireless relay communications with unmanned aerial vehicles: Performance and optimization, IEEE Transactions on Aerospace and Electronic Systems 47 (3) (2011) 2068–2085.

[21] Q. Wu, Y. Zeng, R. Zhang, Joint trajectory and communication design for multi-UAV enabled wireless networks, IEEE Transactions on Wireless Communications 17 (3) (2018) 2109–2121.

[22] Y. Zeng, R. Zhang, T. J. Lim, Throughput maximization for UAV-enabled mobile relaying systems, IEEE Transactions on Communications 64 (12) (2016) 4983–4996.

[23] W. Fawaz, C. Abou-Rjeily, C. Assi, UAV-aided cooperation for FSO communication systems, IEEE Communications Magazine 56 (1) (2018) 70–75.

[24] Y. Pang, Y. Zhang, Y. Gu, M. Pan, Z. Han, P. Li, Efficient data collection for wireless rechargeable sensor clusters in harsh terrains using UAVs, in: Global Communications Conference (GLOBECOM), IEEE, 2014, pp. 234–239.

[25] J. Johnson, E. Basha, C. Detweiler, Charge selection algorithms for maximizing sensor network life with UAV-based limited wireless recharging, in: International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), IEEE, 2013, pp. 159–164.

[26] J. Xu, Y. Zeng, R. Zhang, UAV-enabled wireless power transfer: Trajectory design and energy optimization, IEEE Transactions on Wireless Communications.

[27] S. Jeong, J. Bito, M. M. Tentzeris, Design of a novel wireless power system using machine learning techniques for drone applications, in: Wireless Power Transfer Conference (WPTC), IEEE, 2017, pp. 1–4.

[28] X. He, J. Bito, M. M. Tentzeris, A drone-based wireless power transfer and communications platform, in: Wireless Power Transfer Conference (WPTC), IEEE, 2017, pp. 1–4.

[29] C. Wang, Z. Ma, Design of wireless power transfer device for UAV, in: International Conference on Mechatronics and Automation (ICMA), IEEE, 2016, pp. 2449–2454.

[30] S. Chen, Y. Shu, B. Yu, C. Liang, Z. Shi, J. Chen, Mobile wireless charging and sensing by drones, in: International Conference on Mobile Systems, Applications, and Services Companion (MobiSys), ACM, 2016, pp. 99–99.

[31] A. Mittleider, B. Griffin, C. Detweiler, Experimental analysis of a UAV-based wireless power transfer localization system, in: Experimental Robotics, Springer, 2016, pp. 357–371.

[32] B. Griffin, C. Detweiler, Resonant wireless power transfer to ground sensors from a UAV, in: International Conference on Robotics and Automation (ICRA), IEEE, 2012, pp. 2660–2665.

[33] M.-S. Alouini, A. J. Goldsmith, Adaptive modulation over nakagami fading channels, Wireless Personal Communications 13 (1-2) (2000) 119–143.

[34] I. S. Gradshteyn, I. M. Ryzhik, Table of integrals, series, and products, Academic press, 2014.

[35] K. Li, W. Ni, X. Wang, R. P. Liu, S. S. Kanhere, S. Jha, EPLA: Energy-balancing packets scheduling for airborne relaying networks, in: International Conference on Communications (ICC), IEEE, 2015, pp. 6246–6251.

[36] X. Wang, K. Li, S. S. Kanhere, D. Li, X. Zhang, E. Tovar, PELE: Power efficient legitimate eavesdropping via jamming in UAV communications, in: International Wireless Communications and Mobile Computing Conference (IWCMC), IEEE, 2017, pp. 402–408.

[37] K. Li, N. Ahmed, S. S. Kanhere, S. Jha, Reliable transmissions in AWSNs by using O-BESPAR hybrid antenna, Pervasive and Mobile Computing 30 (2016) 151–165.

[38] K. Li, C. Yuen, S. Jha, Fair scheduling for energy harvesting WSN in smart city, in: SenSys, ACM, 2015, pp. 419–420.

[39] K.-H. Liu, Selection cooperation using RF energy harvesting relays with finite energy buffer, in: Wireless Communications and Networking Conference (WCNC), IEEE, 2014, pp. 2156–2161.

[40] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al., Human-level control through deep reinforcement learning, Nature 518 (7540) (2015) 529.

[41] J. N. Tsitsiklis, B. Van Roy, An analysis of temporal-difference learning with function approximation, IEEE Transactions on Automatic Control 42 (5) (1997) 674–690.

[42] M. van Otterlo, M. Wiering, Reinforcement learning and markov decision processes, in: Reinforcement Learning, Springer, 2012, pp. 3–42.

[43] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al., Tensorflow: a system for large-scale machine learning, in: OSDI, Vol. 16, 2016, pp. 265–283.

[44] F. Chollet. Keras: Deep learning library for theano and tensorflow [online] (2019) [cited 2019].

[45] D. T. Hoang, D. Niyato, P. Wang, D. I. Kim, L. B. Le, Optimal data scheduling and admission control for backscatter sensor networks, IEEE Transactions on Communications 65 (5) (2017) 2062–2077.

[46] Y. Zhang, S. He, J. Chen, Y. Sun, X. S. Shen, Distributed sampling rate control for rechargeable sensor nodes with limited battery capacity, IEEE Transactions on Wireless Communications 12 (6) (2013) 3096–3106.

[47] D. Estep, The bisection algorithm, Practical Analysis in One Variable (2002) 165–177.

Table 1: The list of fundamental variables defined in system model

Notation	**Definition**
I	number of wireless powered ground devices
$v_{\max}$, $v_{\min}$	the maximum and minimum velocity of the UAV
Z	number of laps the UAV patrols
$P_i^z(t)$	transmit power of device i
$\tilde{P}_i^z(t)$	power transferred to device i
P_{UAV}^{tx}	transmit power of the UAV
$\zeta_z(t)$	location of the UAV on its trajectory
$\mathbf{h}_i^z(t)$	channel gain between device i and the UAV
$q_{i,z}(t)$	queue length of device i
D	maximum queue length of the ground device
$\phi_i^z(t)$	modulation scheme of device i
Φ	the highest modulation order
γ_i	SNR between device i and the UAV
$\omega(d_i^z(t), \theta_i^z(t))$	MPT efficiency factor
$e_{i,z}(t)$	battery level of device i
E	battery capacity of the ground device
K	the highest battery level of the ground device
B	number of bits of the data packet
ε	required BER of the channel between the ground device and the UAV
$\widehat{T}_i^z(t)$	contact time between device i and the UAV
$\mathcal{A}$	action set of MDP
δ	discount factor for future states
l	learning iteration in the deep Q-network
φ_l	learning weight in the deep Q-network

Algorithm 1 Deep Reinforcement Learning based Scheduling Algorithm

1. Initialize:
$\mathcal{S}_\alpha \in \mathcal{S}$, $k \in \mathcal{A}$, ϱ, and $w \rightarrow Q\Big\{\mathcal{S}_\beta\Big|\mathcal{S}_\alpha, k\Big\}$.
Random weights $\varphi_l \rightarrow Q\Big\{\mathcal{S}_\beta\Big|\mathcal{S}_\alpha, k; \varphi_l\Big\}$.
Weights $\varphi_{l-1} = \varphi_l \rightarrow$ target deep Q-network $\hat{Q}\Big\{\mathcal{S}_\beta\Big|\mathcal{S}_\alpha, k; \varphi_{l-1}\Big\}$.
Learning time $\rightarrow t_{\text{learning}}$.
2. Learning:
for *episode* $= 1 \rightarrow M$ **do**
  Start state $\mathcal{S}_\alpha \rightarrow \mathcal{S}_1$ and accordingly update $\varphi_l \rightarrow \varphi_1$.
  for $t = 1 \rightarrow t_{\text{learning}}$ **do**
    if Probability ϵ **then**
      Select a random action k_t.
    else
      $k_t \rightarrow \text{argmin}_k Q\Big\{\mathcal{S}_\beta\Big|\mathcal{S}_\alpha, k; \varphi_l\Big\}$
    end if
    Execute action k_t in the environment.
    Obtain the cost $C\Big\{\mathcal{S}_\beta\Big|\mathcal{S}_\alpha, k_t\Big\}$ and the next state $\mathcal{S}_\beta$ at $t+1$.
    Store transition $\Big(\mathcal{S}_\alpha, \mathcal{S}_\beta, k_t, C\Big\{\mathcal{S}_\beta\Big|\mathcal{S}_\alpha, k_t\Big\}\Big)_t \rightarrow \mathbb{D}[\cdot]$.
    Sample random minibatch $\Big(\mathcal{S}_\alpha, \mathcal{S}_\beta, k, C\Big\{\mathcal{S}_\beta\Big|\mathcal{S}_\alpha, k\Big\}\Big)_l$ from $\mathbb{D}[\cdot]$.
    if $\mathcal{S}_\beta$ terminates at step $l+1$ **then**
      Set $y_l \rightarrow C\Big\{\mathcal{S}_\beta\Big|\mathcal{S}_\alpha, k_t\Big\}_l$.
    else
      Set $y_l \rightarrow C\Big\{\mathcal{S}_\beta\Big|\mathcal{S}_\alpha, k_t\Big\}_l + \delta \min_{k' \in \mathcal{A}} \hat{Q}\Big\{\mathcal{S}_{\beta'}\Big|\mathcal{S}_\beta, k'; \varphi_{l-1}\Big\}$.
    end if
    Derive the loss function $\Gamma_l(\varphi_l) \rightarrow (y_l - Q\Big\{\mathcal{S}_\beta\Big|\mathcal{S}_\alpha, k_t; \varphi_l\Big\}_t)^2$.
    Perform (9) $\rightarrow$ gradients with respect to φ_l.
    if There are U number of learning updates. **then**
      Synchronize $\varphi_{l-1} \rightarrow \varphi_l$.
      Reset $\hat{Q}\Big\{\mathcal{S}_\beta\Big|\mathcal{S}_\alpha, k; \varphi_{l-1}\Big\} \rightarrow Q\Big\{\mathcal{S}_\beta\Big|\mathcal{S}_\alpha, k_t; \varphi_l\Big\}_t$.
    end if
  end for
end for
3. Modulation:
Determining $\phi_k^z(t)^\star$ for the scheduled ground device k (see Appendix).

Table 2: TensorFlow configurations

Parameters	**Values**
Number of ground devices	50 $\backsim$ 200
Queue length	20 $\backsim$ 60
Energy levels	50
Number of UAV's way-points	100
Discount factor	0.99
Learning rate	0.0001
Replay memory size	5000
Batch size	32
Number of steps	1000
Number of episodes	500

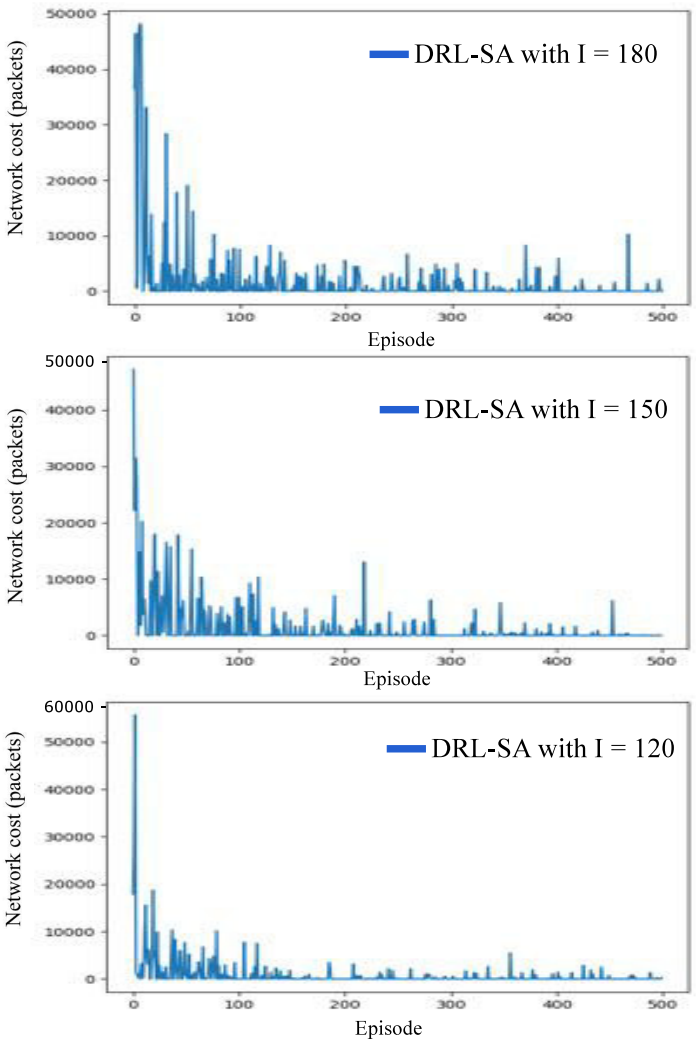
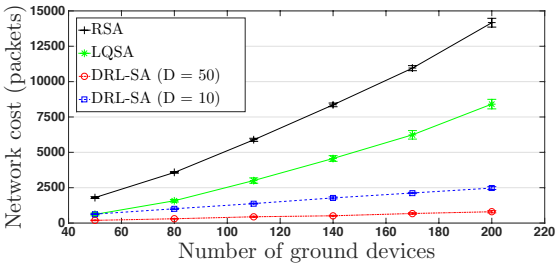
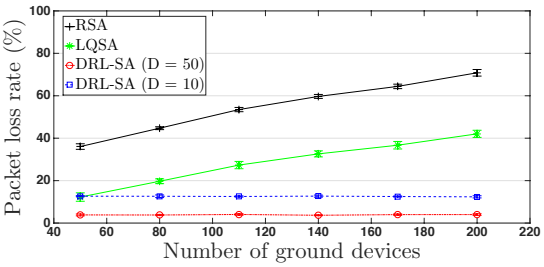


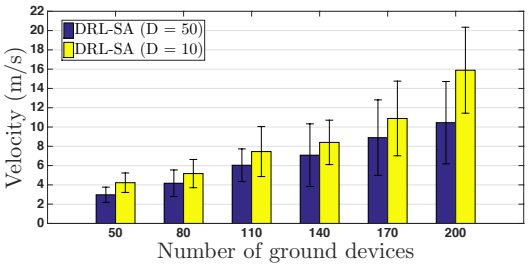
Figure 5: Network cost at the episode in terms of the different number of ground devices, i.e., I.



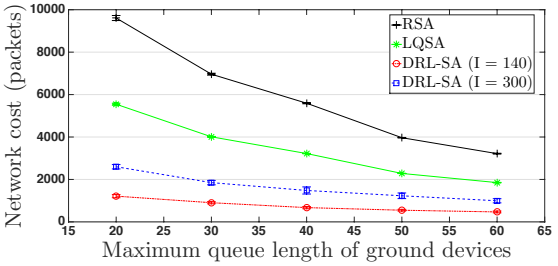
(a) Network cost with regards to I



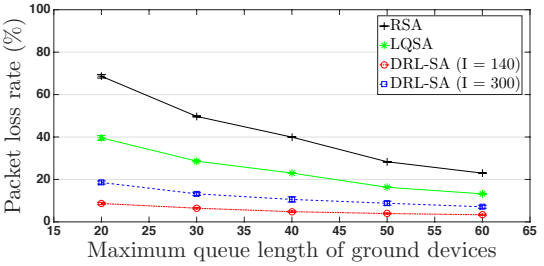
(b) Packet loss rate with regards to I



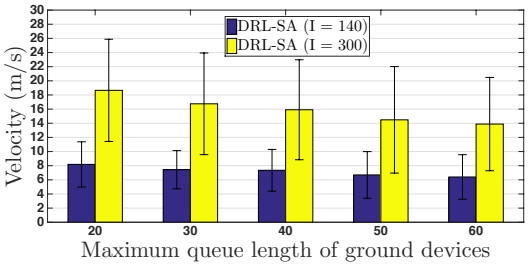
(c) Patrolling velocity with regards to I



(d) Network cost with regards to D



(e) Packet loss rate with regards to D



(f) Patrolling velocity with regards to D

Figure 6: A comparison of network cost and packet loss rate by DRL-SA and the typical scheduling strategies, where the error bars show the standard deviation over 20 runs. The patrolling velocity of the UAV given the different number of ground devices and data queue length is also presented.

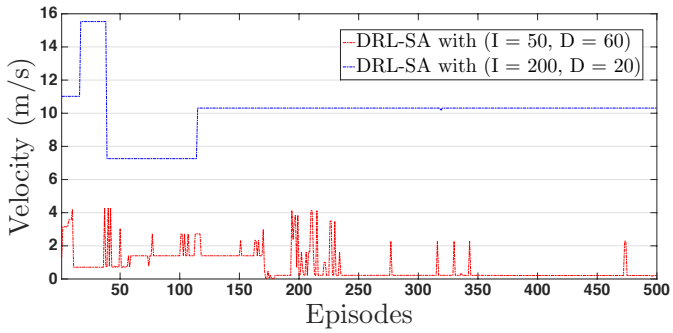


Figure 7: The patrolling velocity of the UAV with respect to the episodes, given $(I = 50, D = 60)$ and $(I = 200, D = 20)$.

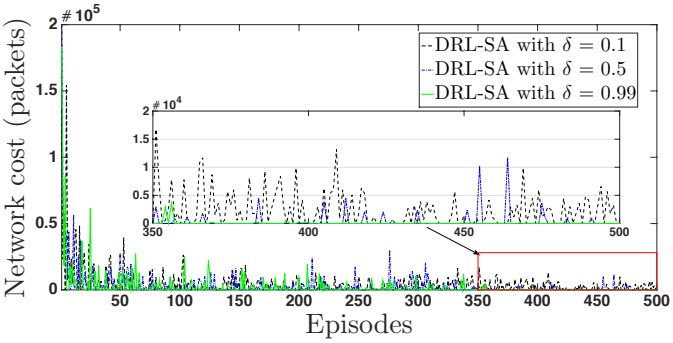


Figure 8: Network cost of DRL-SA with regards to the episodes given the discount factor $\delta = 0.1, 0.5$ and 0.99.