



CISTER

Research Centre in
Real-Time & Embedded
Computing Systems

Journal Paper

Joint Flight Cruise Control and Data Collection in UAV-aided Internet of Things: An Onboard Deep Reinforcement Learning Approach

Early Access Article

Kai Li*

Wei Ni

Eduardo Tovar*

Mohsen Guizani

*CISTER Research Centre

CISTER-TR-200901

2020/08/24

Joint Flight Cruise Control and Data Collection in UAV-aided Internet of Things: An Onboard Deep Reinforcement Learning Approach

Kai Li*, Wei Ni, Eduardo Tovar*, Mohsen Guizani

*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: kai@isep.ipp.pt, Wei.Ni@data61.csiro.au, emt@isep.ipp.pt

<https://www.cister-labs.pt>

Abstract

Employing Unmanned Aerial Vehicles (UAVs) as aerial data collectors in Internet-of-Things (IoT) networks is a promising technology for large-scale environment sensing. A key challenge in UAV-aided data collection is that UAV maneuvering gives rise to buffer overflow at the IoT node and unsuccessful transmission due to lossy airborne channels. This paper formulates a joint optimization of flight cruise control and data collection schedule to minimize network data loss as a Partial Observable Markov Decision Process (POMDP), where the states of individual IoT nodes can be obscure to the UAV. The problem can be optimally solvable by reinforcement learning, but suffers from curse-of-dimensionality and becomes rapidly intractable with the growth in the number of IoT nodes. In practice, a UAV-aided IoT network contains a large number of network states and actions in POMDP while the up-to-date knowledge is not available at the UAV. We propose an onboard Deep Q-Network based Flight Resource Allocation Scheme (DQN-FRAS) to optimize the online flight cruise control of the UAV and data scheduling given outdated knowledge on the network states. Numerical results demonstrate that DQN-FRAS reduces the packet loss by over 51%, as compared to existing non-learning heuristics.

Joint Flight Cruise Control and Data Collection in UAV-aided Internet of Things: An Onboard Deep Reinforcement Learning Approach

Kai Li, *Senior Member, IEEE*, Wei Ni, *Senior Member, IEEE*, Eduardo Tovar, *Member, IEEE*, and Mohsen Guizani, *Fellow, IEEE*

Abstract—Employing Unmanned Aerial Vehicles (UAVs) as aerial data collectors in Internet-of-Things (IoT) networks is a promising technology for large-scale environment sensing. A key challenge in UAV-aided data collection is that UAV maneuvering gives rise to buffer overflow at the IoT node and unsuccessful transmission due to lossy airborne channels. This paper formulates a joint optimization of flight cruise control and data collection schedule to minimize network data loss as a Partial Observable Markov Decision Process (POMDP), where the states of individual IoT nodes can be obscure to the UAV. The problem can be optimally solvable by reinforcement learning, but suffers from curse-of-dimensionality and becomes rapidly intractable with the growth in the number of IoT nodes. In practice, a UAV-aided IoT network contains a large number of network states and actions in POMDP while the up-to-date knowledge is not available at the UAV. We propose an onboard Deep Q-Network based Flight Resource Allocation Scheme (DQN-FRAS) to optimize the online flight cruise control of the UAV and data scheduling given outdated knowledge on the network states. Numerical results demonstrate that DQN-FRAS reduces the packet loss by over 51%, as compared to existing non-learning heuristics.

Index Terms—Flight cruise control, Communication decisions, Unmanned aerial vehicles, Internet-of-Things, Deep reinforcement learning

I. INTRODUCTION

Recent advances in energy harvesting techniques enable Internet-of-Things (IoT) networks to contain a large number of low-cost sensing devices with energy harvesting capabilities [1], [2]. The IoT node is equipped with solar panels, wind power generators, or wireless power receiver to harvest energy from ambient resources for opportunistically charging its battery [3]. Sensory data are generated by the IoT node at an application-specific rate, and they are stored in a data queue for future transmission. Data collection in a large-scale IoT network is difficult since the

nodes can be dispersedly deployed in separate areas where communication facilities are not available [4].

Due to flexible deployment, low operational costs, and excellent maneuverability, Unmanned Aerial Vehicle (UAV), *a.k.a.* drone, is envisioned to provide a promising paradigm for data collection in IoT networks [5], [6]. The UAV can move sufficiently close to the IoT node to improve wireless signal strength, leveraging a short-distance line-of-sight (LoS) communication link between the UAV and the IoT node [7]. In fact, several pilot UAV projects have been launched by the leading ICT tycoons. For example, a joint project is funded by SoftBank company partnered with NASA and U.S. aerospace company AeroVironment for connecting 5G networks and the Internet of Things [8]. Initial trials of deploying UAVs are made by Facebook and Google to enhance the communication services for the terrestrial users in the cellular networks [9], [10]. The 3rd Generation Partnership Project (3GPP) also studies capability of the enhanced Long Term Evolution (LTE) support for UAVs [11].

Figure 1 depicts a typical UAV-aided IoT networks, where the IoT nodes on the ground are deployed for environmental monitoring, architecture surveillance, public safety, intelligent transportation, or logistics automation. The cooperative secure network codes first proposed in [12] are able to support stable and reliable data communications for IoT nodes with differential QoS guarantee. The UAV equipped with a high-capacity battery, wireless radio and onboard processors hovers over the area of interest. Flight cruise of the UAV is adapted for collecting and ferrying the sensory data of the IoT nodes given limited radio coverage of the UAV and the IoT nodes [13], [14]. In particular, the data transmission the IoT nodes is dramatically influenced by time-varying airborne lossy channels due to the flight cruise control at the UAV.

Battery energy of the IoT nodes can be greatly different from each other, since the amount of harvested energy is impacted by natural conditions, e.g., weather or wireless interference from existing wireless networks. Scheduling the IoT node with poor channel condition or low battery to transmit data gives rise to packet reception errors or buffer overflow at other nodes. In practice, the battery energy levels, data queue lengths, and channel conditions of all the IoT nodes are not available or can only be partially observed by the UAV. Therefore, online flight resource

K. Li and E. Tovar are with Real-Time and Embedded Computing Systems Research Centre (CISTER), 4249-015 Porto, Portugal (E-mail: {kai,emt}@isep.ipp.pt).

W. Ni is with the Digital Productivity and Services Flagship, Commonwealth Scientific and Industrial Research Organization (CSIRO), Sydney, Australia (E-mail: wei.ni@data61.csiro.au).

M. Guizani is with Computer Science and Engineering Department, Qatar University, Qatar (E-mail: mguizani@ieee.org).

Copyright (c) 20xx IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubpermissions@ieee.org.

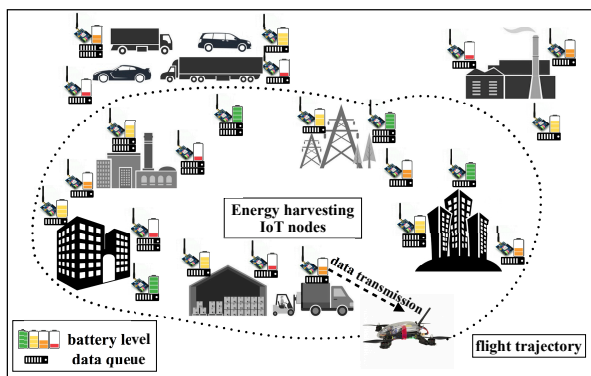


Fig. 1: In UAV-aided IoT networks, the UAV hovers over the area of interest and adapts its flight cruise for the data collection. The IoT node can harvest energy from ambient natural resources to charge its battery.

allocation, i.e., flight cruise control of the UAV and the IoT node selection for data collection, for preventing data lost resulting from data queue overflow and fading channels is crucial in UAV-aided IoT networks.

In this paper, the flight resource allocation in the UAV-aided IoT network is formulated as a Partial Observable Markov Decision Process (POMDP), where network states consist of battery levels and queue lengths of the IoT nodes, channel conditions, Time To be Alive (TTA), and waypoints of the UAV. As a POMDP evolves over time as a dynamical process, the UAV does not have complete knowledge about network states at each waypoint along the flight cruise. Instead, the UAV has the observations generated over time from the environment, providing clues of the actual underlying states (hence the term partially observable). A reinforcement learning approach, e.g., Q-learning proposed by Watkins [15], can obtain the optimal decisions or actions given the small state and action space in POMDP. Unfortunately, the state and action space in the UAV-aided IoT network is typically large, thus, Q-learning that suffers from the well-known curse-of-dimensionality [16] is impractical to solve the online flight resource allocation.

To alleviate the POMDP with large state and action spaces, an onboard Deep Q-Network based Flight Resource Allocation Scheme (DQN-FRAS) is proposed, which combines reinforcement learning with deep neural networks, to asymptotically minimize the data lost. The onboard DQN-FRAS jointly optimizes the flight cruise of the UAV and data transmission scheduling through online training actions of the UAV, i.e., the instantaneous waypoints, the selection of the IoT node, and the transmit power allocation. In particular, any position in the environment can be determined as a waypoint of the UAV, which results in a massive action space for training DQN and high complexity of deep reinforcement learning. To address this issue, DQN-FRAS significantly reduces the number of waypoints in the action space by restricting heading directions of the UAV at each waypoint. Given a number of potential heading directions, the next waypoint of the UAV is a position along one of

the directions while it is confined by the maximum speed of the UAV.

An ϵ -greedy policy is carried out in DQN-FRAS to balance exploitation based on the packet loss already obtained with trying new flight resource allocation to explore the unknown knowledge of the network packet loss. In addition, experience replay [17] is utilized to store the flight resource allocation experiences at each time step, which significantly reduces expansion of the state space.

The main contributions can be summarized as follows:

- We study a fundamental challenge in UAV-aided IoT networks, where the UAV maneuvering gives rise to buffer overflow at the IoT node and unsuccessful transmission due to lossy airborne channels.
- The flight cruise control and data collection schedule in the UAV-aided IoT network are formulated as a POMDP, where the UAV does not have complete knowledge about network states at each waypoint along the flight cruise.
- DQN-FRAS explores the deep reinforcement learning to jointly optimize the online flight cruise control and data collection scheduling given outdated knowledge on the network states.
- To verify our design, we implement DQN-FRAS based on Python library Keras running on top of Google TensorFlow that is one of the most popular platforms for deep neural networks. Numerical results demonstrate that the proposed DQN-FRAS achieves at least 51% reduction in the overall packet loss, as compared to existing non-learning heuristics.

The rest of this paper is organized as follows. Section II presents related work on path planning and data communications in UAV networks. Section III studies system model and data collection protocol in UAV-aided IoT Networks. In Section IV, we propose onboard DQN-FRAS for the flight cruise control and data collection scheduling. Numerical results are analyzed in Section V, and Section VI concludes the paper.

II. LITERATURE REVIEW

This section presents the literature on the trajectory planning and the UAV relay communications.

A. Trajectory planning of the UAV

In [18], the UAV is employed to relay data for the ground nodes in a large area. Since moving close to some of the ground nodes prolongs the transmission latency of the other nodes, the flight trajectory of the UAV is designed to balance the transmission latency of the ground nodes, and reduce the energy consumption of the UAV. The ground node consumes less energy on the data transmission if the UAV can fly closer to it [19]. The UAV consumes extra propulsion energy due to such movements. The tradeoff between the transmission energy consumption of the ground nodes and the propulsion energy consumption of the UAV is characterized by the trajectory design. A UAV-enabled multicasting system is presented in [20], where the UAV

disseminates a common file to multiple ground nodes. The flight trajectory is planned to reduce completion time of the mission, and increase the successful packet delivery probability at the ground nodes. A UAV-enabled wireless power transfer system is presented in [21], where a UAV-mounted energy transmitter is employed to transfer wireless energy to the ground nodes. Given a specific charging period, the UAV's trajectory is adapted to guarantee the minimum harvested energy of the ground nodes under the maximum speed constraint of the UAV. In [22], the minimum average throughput of the ground nodes is improved by planning the flight trajectory and scheduling the downlink transmission from the UAV to the ground nodes. The transmit power of the UAV is adjusted along the trajectory to alleviate the co-channel interference with the other ground nodes.

For distributed estimation of unknown data in UAV-aided sensor networks, the UAV's trajectory is studied to enlarge coverage areas of the data collection subject to the flight duration and the maximum cruising speed [23]. Since the trajectory planning problem is non-convex and NP-hard, the trajectory is assumed to contain connected line segments only, which is solved by the traveling salesman problem method. For protecting the communication from eavesdropping attacks, the UAV trajectory and transmit power are exploited to enhance secrecy rate of the ground nodes, without location information of the eavesdroppers [24]. The UAV with data queue length above a threshold experiences traffic congestion resulting in communication delay [25]. To alleviate the congestion at the UAV, a trajectory control algorithm is developed to update the flight cruise based on the traffic congestion states of the communication link. In [26], a trajectory planning algorithm is presented to adjust heading of the UAV to reduce the channel interference. Given a constant cruising speed, the approximate sum rate is improved by controlling the heading while guaranteeing fairness on the data rate of the ground nodes.

Deep reinforcement learning is applied to UAV-to-Ground communications [27], where the trajectory design is modeled as a Markov decision process. A sensing and transmission protocol is designed to decide to transmit the sensory data of the UAV to the ground nodes through the UAV-to-Ground link or the cellular network, by running deep reinforcement learning. The authors of [28] focus on a UAV-assisted wireless network, where a UAV collects the status information of energy-constrained ground nodes. Deep reinforcement learning is used to decide on the UAV's flight trajectory and schedule the transmissions of the ground nodes to minimize the age-of-information at the UAV. In [29], an interference-aware trajectory planning scheme is studied in a network of cellular-connected UAVs. A non-cooperative game is formulated to balance the energy consumption of the UAV, transmission latency and interference, where deep reinforcement learning learns the trajectory and transmit power of the UAV. These existing trajectory design techniques in UAV-aided IoT networks improve network throughput, timeliness, or energy efficiency of the UAV. However, the data loss caused by buffer overflows at the IoT nodes and poor channel conditions is

not considered.

B. Data communications in UAV networks

The authors in [30] study deep reinforcement learning in the UAV network to collect data from the ground sensors while preventing buffer overflows of the ground sensors. Given a fixed flight cruise, the UAV equipped with a wireless power transmitter selectively charges the ground sensors within radio coverage of the UAV to extend network lifetime. In [31], a deep reinforcement learning based deployment strategy is developed to reduce propulsion energy consumption of the UAVs while improving network coverage and connectivity. Li *et al.* investigate an energy-efficient relay scheduling scheme in the UAV network to extend the network lifetime [32]. A practical data scheduling algorithm is presented by decoupling energy consumption balancing and transmit rate adaptation, and performing these two parts in an alternating manner.

The authors of [33] focus on the UAV with caches that store frequently required contents to relieve the pressure of backhaul at peak time. Precoding and decoding matrices are designed for managing interference alignment of the ground nodes. To secure the data transmission of the legitimate nodes, the idle ground nodes transmit jamming signal to disrupt the potential eavesdropping attacks. In [34], an adversary UAV is considered to launch multiple attacks against data communications of the legitimate UAV. A deep Q-learning based transmit power allocation strategy is studied for the legitimate UAV to detect the attack mode and adjust the transmit power against the attack. For communication surveillance in UAV networks, the legitimate UAV is studied to overhear the communication of the suspicious UAVs and track their flight trajectories [35]. The transmit power of the legitimate UAV is adjusted to enhance the legitimate eavesdropping performance.

Deep reinforcement learning is applied to the channel and power allocation of a UAV-aided IoT system in [36]. By conducting an actor-critic strategy of deep reinforcement learning, the UAV allocates the channels and transmit power to the IoT nodes to improve the energy efficiency of the IoT network. In [37], machine learning is applied to the predictive deployment of the UAV to provide on-demand wireless services to cellular users. The machine learning predicts network congestions and deploys the UAV to the hotspots, while reducing the overall power consumption of communication and mobility. These communication protocols are designed, given the flight cruise of the UAV. Different from these existing studies [36], [37], we jointly optimize the online flight cruise control of the UAV and the transmission schedule of the IoT nodes in this paper.

III. SYSTEM MODEL OF UAV-AIDED DATA COLLECTION

In this section, we introduce system model of UAV-aided data collection in IoT networks. Table I lists notations of fundamental variables used in the paper.

TABLE I: The list of fundamental variables

Notation	Definition
N	number of energy harvesting IoT nodes, $i \in [1, N]$
S	number of laps of the flight cruise, $s \in [1, S]$
$P_i^s(t)$	transmit power of node i at t in the s -th lap
W	total waypoints on the flight cruise
$h_{i,s}(t)$	channel gain of the link between the UAV and i
$q_{i,s}(t)$	queue length of the IoT node
L	buffer size of the IoT node
$r_i^s(t)$	modulation of node i
R	the highest modulation order
$\eta_i^s(t)$	SNR between the UAV and the IoT node
$b_{i,s}(t)$	battery level of node i
B	battery capacity of the IoT node
τ_i	TTA value of IoT node i
\mathcal{S}	network states in POMDP
\mathcal{A}	actions in POMDP
κ	discount factor
ω	learning weight in DQN-FRAS
J	number of learning iterations in DQN-FRAS
φ_j	learning weight at the j th iteration in the DQN

A. System model

We consider N energy harvesting single-antenna IoT nodes airlifted to a remote area. A multi-antenna UAV is employed to fly over the area of interest to collect data from the IoT nodes, where the flight cruise contains S laps. High-capacity battery or solar powered UAVs have been developed [38], which leads to a prolonged UAV lifetime and subsequently a large value of S . The received signal strength (RSS) of the airborne channel between the UAV and the IoT node can be enhanced by using (coherent) beamforming techniques at the UAV.

Let $p_{\text{uav}}^s(t) = (x_{\text{uav}}^s(t), y_{\text{uav}}^s(t), z_{\text{uav}}^s(t))$ denote the waypoint of the UAV at time t . We have $p_{\text{uav}}^s(0) = p_{\text{uav}}^s(T)$, indicating that the UAV returns to the starting point at the end of each lap. The distance from the UAV to node i at time t in lap s can be expressed as

$$d_i^s(t) = \|p_{\text{uav}}^s(t) - p_i\| = \sqrt{(x_{\text{uav}}^s(t) - x_i)^2 + (y_{\text{uav}}^s(t) - y_i)^2 + (z_{\text{uav}}^s(t))^2}, \quad (1)$$

where $p_i = (x_i, y_i, 0)$ is the location of the IoT node.

Let $h_{i,s}(t)$ denote the complex channel coefficient between a particular antenna of the UAV and the IoT node, and can be measured at the UAV. The channels between the UAV and the IoT nodes are dominated by the LoS propagation. Thus, the channel coefficients between the UAV and node i at t in lap s follow the free-space path loss model, which can be expressed as

$$h_{i,s}(t) = \frac{P_0}{d_i^s(t)^2}, \quad (2)$$

where P_0 is a reference transmit power of the IoT node at the distance $d_i^s(t) = 1$ m, and the Doppler effect caused by the UAV mobility is assumed to be well compensated at the receiver. Furthermore, the SNR of the channel between the UAV and node i can be given by

$$\eta_i^s(t) = \frac{h_{i,s}(t)P_i^s(t)}{\sigma_0^2}, \quad (3)$$

where $P_i^s(t)$ is the real-time transmit power of the IoT node, and σ_0^2 is noise power at the UAV.

Let $r_i^s(t)$ denote the modulation scheme that is allocated to the IoT node. Given $r_i^s(t)$ and $h_{i,s}(t)$, the required transmit power of node i at t in lap s can be given by [32]

$$P_i^s(t) \approx \frac{\varrho_2^{-1} \ln \frac{\varrho_1}{\epsilon}}{h_{i,s}(t)} (2^{r_i^s(t)} - 1), \quad (4)$$

where ϱ_1 and ϱ_2 are constants, and the required bit error rate (BER) between the UAV and the IoT node is ϵ .

The IoT node i has the data queue length of $q_{i,s}(t)$, and $q_{i,s}(t) \leq L$, where L is the buffer size. The battery of the IoT node is rechargeable and the capacity has B Joules. Each of the IoT nodes harvests energy from their ambient environment, e.g., using solar panels, wind power generators or wireless power transfer, to energize its operations, e.g., computing and communication. Battery readings are time-varying continuous variables that are difficult to be obtained in the real-time flight resource allocation. For illustration convenience and mathematical tractability, the battery readings at the IoT node are discretized into e levels, as $0 < \mathcal{B} < 2\mathcal{B} < \dots < e\mathcal{B} = B$. Accordingly, we have $b_{i,s}(t) \in \{0, \mathcal{B}, 2\mathcal{B}, \dots, e\mathcal{B}\}$ [39], which quantizes the battery of the IoT node to the closest lower level. In addition, overcharging the IoT node causes its battery overflows given the maximum battery level of B .

To simplify the system model, we consider that the IoT network is homogenous where the IoT nodes have the same battery capacity and buffer size. However, the proposed DQN-FRAS framework can be extended to a heterogeneous IoT network, where the complexity of the resource allocation problem may grow as the result of an increased number of possible network states.

B. UAV-aided data collection

Figure 2 presents the data collection protocol for the UAV-aided IoT network. Specifically, each communication frame consists of a number of equally divided time slots, in which one IoT node is scheduled to transmit one data packet in each slot. The proposed DQN-FRAS is carried out on the UAV to determine instantaneously the next waypoint for the flight cruise, select an IoT node to capture its sensory data, allocate the transmit power of the selected node. The details will be investigated in the next section. A short beacon message that notifies the selected node for data collection is broadcasted by the UAV at the beginning of the communication frame. The state information of the selected IoT node, i.e., $b_{i,s}(t)$ and $q_{i,s}(t)$, can be put in a control segment of the data packet that is transmitted to the UAV. The channel SNR $\eta_i^s(t)$ is measured by the UAV based on the reception of the data packet. Moreover, the UAV collects and processes the received data packets online, while $b_{i,s}(t)$, $q_{i,s}(t)$ and $\eta_i^s(t)$ are used for training the onboard DQN-FRAS. With the updated state information, DQN-FRAS controls the flight cruise and schedules the next IoT node in the following communication frame. Additionally, the UAV-aided data collection protocol has

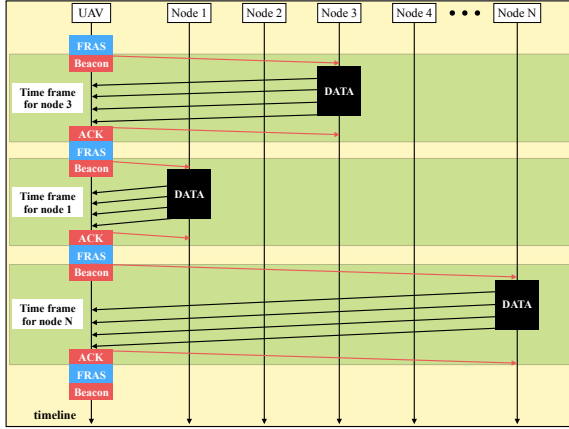


Fig. 2: Data collection protocol of the UAV-aided IoT network. In each communication frame, one node is selected for data transmission according to the policy of DQN-FRAS, which will be investigated in Section IV. The ID of the selected IoT node fits in a short beacon message that is broadcasted by the UAV.

small control segment overheads. For example, consider $b_{i,s}(t)$ of 100 and $q_{i,s}(t)$ of 100 packets, the overhead in each data packet has 12 bits, much smaller than the data payload that is typically several kilobits.

In each communication frame, only one IoT node is selected by the UAV to capture data. To estimate the battery energy and queue length of unselected IoT nodes, the TTA value, denoted by τ_i , is recorded and updated at the UAV for node i ($i \in [1, N]$). τ_i defines the time span when the UAV collects a new data packet from node i . Moreover, τ_i returns to 0 when a new packet is collected from node i .

IV. PROBLEM FORMULATION AND DQN-FRAS

In this section, we study a POMDP formulation for the joint flight cruise control and data collection problem. The onboard DQN-FRAS is proposed to minimize packet loss of the IoT nodes. Moreover, the action space for training DQN-FRAS is greatly reduced by the constraint on heading directions and cruising speeds of the UAV.

A. POMDP formulation

The joint flight cruise control and data collection are formulated as a POMDP since the knowledge of battery levels, data buffer length of the IoT nodes, and channel quality, can only be partially observed by the UAV due to the limited radio coverage and movements of the UAV.

Let \mathcal{S}_α denote network state α , which is composed of the battery level $b_{i,s}(\alpha)$ and data queue length $q_{i,s}(\alpha)$ of every IoT node i , the current location of the UAV $p_{uav}^s(\alpha)$, the TTA value of the IoT node $\tau_i(\alpha)$, and the channel gain $h_{i,s}(\alpha)$. At \mathcal{S}_α , the UAV decides on its next waypoint $p_{uav}^s(\beta)$ on the flight cruise, selects the next IoT node i_β^s to capture data, and allocates $P_i^s(\beta)$. We denote $\mathcal{A}_{\mathcal{S}_\alpha} \subseteq \mathcal{A}$ as the actions of the UAV at state \mathcal{S}_α , where \mathcal{A} is the complete

set of all the actions in POMDP. Therefore, we have

$$\mathcal{S}_\alpha = \{b_{i,s}(\alpha), q_{i,s}(\alpha), p_{uav}^s(\alpha), \tau_i(\alpha), h_{i,s}(\alpha)\}; \quad (5)$$

$$\mathcal{A}_{\mathcal{S}_\alpha} = \{p_{uav}^s(\beta), i_\beta^s, P_i^s(\beta) | \mathcal{S}_\alpha\}, \quad (6)$$

where $i \in [1, N]$, $s \in [1, S]$. $\mathcal{A}_{\mathcal{S}_\alpha}$ also accounts for potential influence on the future evolution of the network. Particularly, the decision of $\mathcal{A}_{\mathcal{S}_\alpha}$ can affect the future network states and, in turn, impact the future actions of cruise control and data transmission scheduling. The actions of the UAV can be described as a discrete-time stochastic control process, which is synthetically determined by the random data arrival or queueing status at the IoT node and the cruise control and node selection decisions taken by the UAV. The state transition probability is denoted by $\Pr\{\mathcal{S}_\beta | \mathcal{S}_\alpha, \mathcal{A}_{\mathcal{S}_\alpha}\}$ given the action $\mathcal{A}_{\mathcal{S}_\alpha}$.

Since $b_{i,s}(\alpha)$, $q_{i,s}(\alpha)$ and $h_{i,s}(\alpha)$ in \mathcal{S}_α are not fully observable at the UAV, a partial observation of the network states is defined as $z_s(\alpha)$, which gives

$$z_s(\alpha) = \{b_{i,s}(\alpha) + \tilde{b}_{i,s}, q_{i,s}(\alpha) + \tilde{q}_{i,s}, h_{i,s}(\alpha) + \tilde{h}_{i,s}\} \quad (7)$$

where $\tilde{b}_{i,s}$, $\tilde{q}_{i,s}$, and $\tilde{h}_{i,s}$ represent random measurement errors. A belief state in POMDP, denoted by χ , presents the posterior distribution of the underlying network state, which is updated using Bayes rule given the observations. Given the current belief-state χ , the objective of POMDP is to find an optimal policy π^* which minimizes a long-term overall network packet loss when following a sequence of actions of the UAV and observations. The accumulated discounted return is the sum of the discounted network cost after executing every action, which is $\sum_{t=1}^{\infty} \kappa^{t-1} C_t$, where C_t is the immediate cost received at particular time step t for taking action $\mathcal{A}_{\mathcal{S}_\alpha}$. $\kappa \in [0, 1]$ is a discount factor for future states. The objective function of the formulated POMDP is the expected return from the belief state χ according to policy π , which defines $\Phi^\pi(\chi) = \min_{\pi \in \Pi} \mathbb{E}^\pi \left\{ \sum_{t=1}^{\infty} \kappa^{t-1} C_t \mid \mathcal{A}_{\mathcal{S}_\alpha}, \chi \right\}$. Therefore, the optimal policy for POMDP is the one which minimizes the objective, $\pi^*(\chi) = \arg \min_{\pi} \Phi^\pi(\chi)$.

In terms of the state space, the total number of states in POMDP has $(BLH)^N(W-1)S$, where the flight cruise has W number of waypoints, and the UAV returns to the starting point at the end of each lap. H is the number of discretized channel states. Given the actions of the UAV, i.e., $\mathcal{A}_{\mathcal{S}_\alpha} = \{p_{uav}^s(\beta), i_\beta^s, P_i^s(\beta) | \mathcal{S}_\alpha\}$, the size of the action space in POMDP is $|\mathcal{A}_{\mathcal{S}_\alpha}| = (W-2)RNS$, where R is the highest modulation order.

Consider a simplified example of the POMDP, where $N = 2$, $W = 4$, $B = 2$, $L = 2$, $S = 2$ and $H = 2$. When the UAV moves to the next waypoint (x_2, y_2) and no energy is harvested by node i , the next state of $\mathcal{S}_\alpha = \{b_{i,1}(\alpha) = 1, q_{i,1}(\alpha) = 1, p_{uav}^s(\alpha) = (x_1, y_1), \tau_i(\alpha)\}$ can be $\{b_{i,1}(\beta) = 1, q_{i,1}(\beta) = 2, p_{uav}^s(\beta) = (x_2, y_2), \tau_i(\beta)\}$ with two possible state transitions, i.e., (i) node i is scheduled to transmit data, but link failure due to poor SNR; (ii) node i is not selected and a new packet arrives at node i . The next state \mathcal{S}_β can also be $\{b_{i,1}(\beta) = 2, q_{i,1}(\beta) = 1, p_{uav}^s(\beta) = (x_3, y_3), \tau_i(\beta)\}$, where the UAV

moves to (x_3, y_3) and node i successfully harvests energy. The two possible state transitions are (i) node i is selected, and the data transmission is successful; (ii) node i is not selected and there is no new packet arrival. Note that this gives a small-scale example of the state transition with two possible trajectories. In practice, the UAV can have thousands of possible trajectories, which generates about 1.2×10^{12} POMDP states. This leads to an extremely large state and action space in POMDP, as well as a complex state transition diagram.

B. Q-learning Optimization

It is difficult to optimize a POMDP by using the optimization theory. One-shot optimization techniques, such as linear programming and convex optimization [40], would require the a-priori knowledge of the system across the entire time horizon, and could only produce offline solutions. Stochastic optimization, such as Lyapunov optimization [41] and stochastic gradient descent (SGD), does produce asymptotically optimal online solutions in the absence of the a-priori knowledge, but would only be effective under stationary settings (as opposed to an MDP or POMDP). Dynamic programming (DP) is a typical solver of MDPs, which is not suitable for POMDPs due to the lack of the complete knowledge of the system (i.e., the network state) at a decision-making moment.

Since $C\{\mathcal{S}_\beta|\mathcal{S}_\alpha, \mathcal{A}_{\mathcal{S}_\alpha}\}$ and $\Pr\{\mathcal{S}_\beta|\mathcal{S}_\alpha, \mathcal{A}_{\mathcal{S}_\alpha}\}$ are unknown in real-time UAV's flight, Q-learning can be considered to optimally solve the flight resource allocation problem in $\Phi(\mathcal{S}_\alpha)$. Q-learning aims to minimize the long-term expected packet loss of the ground IoT nodes without the transition and/or cost functions. Specifically, $Q\{\mathcal{S}_\beta|\mathcal{S}_\alpha, \mathcal{A}_{\mathcal{S}_\alpha}\}$ defines a Q-function that gives the expected accumulated network cost, i.e., overall data loss, after observing \mathcal{S}_α and taking action $\mathcal{A}_{\mathcal{S}_\alpha}$. Moreover, $Q\{\mathcal{S}_\beta|\mathcal{S}_\alpha, \mathcal{A}_{\mathcal{S}_\alpha}\}$ is minimized by optimally determining the next waypoint of the UAV, selecting the IoT node, and allocating the transmit power. Thus, the optimal Q-function, denoted by $Q^*\{\mathcal{S}_\beta|\mathcal{S}_\alpha, \mathcal{A}_{\mathcal{S}_\alpha}^*\}$, is expressed in the form of the expected packet loss at \mathcal{S}_α and the minimum value of Q-function over all future states, which is

$$Q^*\{\mathcal{S}_\beta|\mathcal{S}_\alpha, \mathcal{A}_{\mathcal{S}_\alpha}^*\} = (1 - \omega)Q^*\{\mathcal{S}_\beta|\mathcal{S}_\alpha, \mathcal{A}_{\mathcal{S}_\alpha}^*\} + \omega[C\{\mathcal{S}_\beta|\mathcal{S}_\alpha, \mathcal{A}_{\mathcal{S}_\alpha}^*\} + \kappa \min_{\mathcal{A}_{\mathcal{S}_\beta} \in \mathcal{A}} Q\{\mathcal{S}_{\beta'}|\mathcal{S}_\beta, \mathcal{A}_{\mathcal{S}_\beta}\}]. \quad (8)$$

where $\mathcal{S}_{\beta'}$ is the next state of \mathcal{S}_β when the action $\mathcal{A}_{\mathcal{S}_\beta}$ is carried out. The convergence rate of $Q\{\mathcal{S}_\beta|\mathcal{S}_\alpha, \mathcal{A}_{\mathcal{S}_\alpha}\}$ optimization increases with $\omega \in (0, 1]$ which denotes the learning rate.

It has been known that Q-learning suffers from the curse-of-dimensionality, where the state and action spaces have to be maintained in a small scale. However, the UAV-aided IoT network typically contains a large number of potential waypoints and IoT nodes, which leads to the extremely large state and/or action spaces.

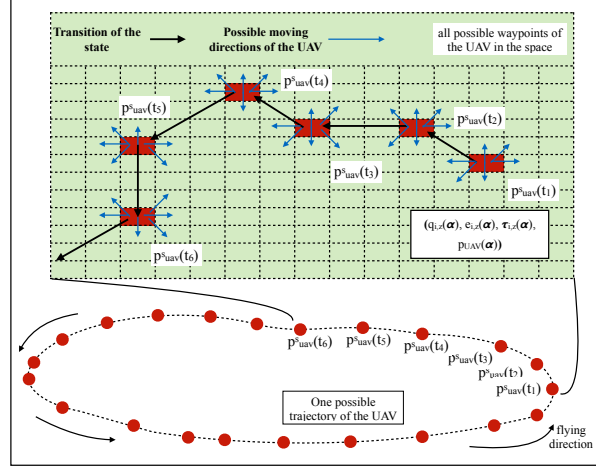


Fig. 3: The blocks stand for possible waypoints of the UAV in space. At each waypoint, the UAV takes actions of moving along one of the 7 possible directions to the next waypoint that is determined by DQN-FRAS.

C. DQN-FRAS

Deep reinforcement learning, e.g., DQN, is the asymptotic approximation of learning to circumvent the curse-of-dimensionality of Q-learning and is suitable for systems with large state and action spaces. In particular, DQNs utilize deep neural networks to map between the state-action combinations and the Q-values. In this sense, a DQN is suitable for our considered problem which has a large state space consisting of all the possible waypoints of the UAV, and the battery levels, queue lengths and channel conditions of all the ground IoT devices.

To overcome the scalability issue of the training space in Q-learning, we firstly reduce the action space by restricting the heading directions of the UAV. As shown in Figure 3, the UAV at each waypoint moves along one of the 7 possible heading directions to the next waypoint while the UAV is not allowed to fly backward. The blocks stand for the possible waypoints of the UAV in space. In particular, the furthest position where the UAV can move along the heading direction is confined by the maximum cruising speed. Furthermore, we develop the onboard DQN-FRAS based on deep reinforcement learning to optimize the online flight cruise control and data transmission scheduling with the large state and action spaces. Figure 4 illustrates the DQN-FRAS architecture, in which onboard deep reinforcement learning trains and optimizes the actions of the UAV to address the online flight cruise control, IoT node selection, and transmit power of the selected node. Since every action can stimulate a new state transition, making the learning process rather slow, experience replay is deployed in DQN-FRAS to store historical data in the onboard memory of the UAV, as shown in Figure 4. The network states and actions are randomized to remove correlations of the observations.

DQN-FRAS implements the replay memory $\mathbb{D}[\cdot]$ to keep $(\mathcal{S}_\alpha, \mathcal{S}_\beta, \mathcal{A}_{\mathcal{S}_\alpha}, C\{\mathcal{S}_\beta|\mathcal{S}_\alpha, \mathcal{A}_{\mathcal{S}_\alpha}\})$ at each state, pooled over many episodes (where an episode ends when a terminal

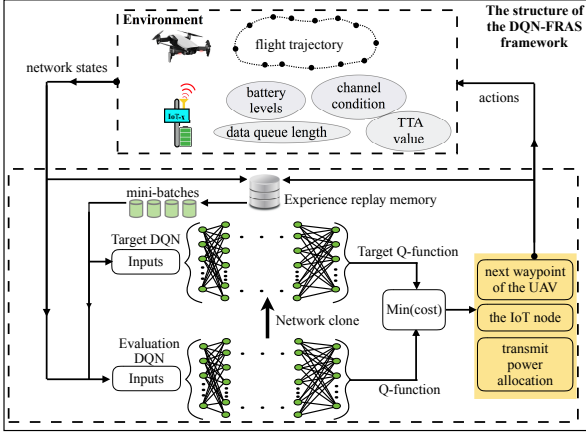


Fig. 4: An illustration of the DQN-FRAS architecture, where deep reinforcement learning with experience relay is carried out at the UAV to optimize its actions.

state is reached). Moreover, the samples (or minibatches) of the experience in DQN-FRAS are accordingly updated by learning to minimize a sequence of the loss function $\Gamma(\varphi_j)$ which is

$$\Gamma(\varphi_j) = \mathbb{E}_{(\mathcal{S}_\alpha, \mathcal{S}_\beta, \mathcal{A}_{\mathcal{S}_\alpha}, C\{\mathcal{S}_\beta|\mathcal{S}_\alpha, \mathcal{A}_{\mathcal{S}_\alpha}\}) \sim \mathbb{D}[\mathcal{S}_\alpha, \mathcal{S}_\beta, \mathcal{A}_{\mathcal{S}_\alpha}, C\{\mathcal{S}_\beta|\mathcal{S}_\alpha, \mathcal{A}_{\mathcal{S}_\alpha}\}]}, \quad (9)$$

where φ_j is the weight at iteration j . $\mathbb{D}[\mathcal{S}_\alpha, \mathcal{S}_\beta, \mathcal{A}_{\mathcal{S}_\alpha}, C\{\mathcal{S}_\beta|\mathcal{S}_\alpha, \mathcal{A}_{\mathcal{S}_\alpha}\}]$ denotes the experience play storage, and is given in (10).

In Figure 4, DQN-FRAS maintains two separate neural networks at the UAV, an evaluation DQN and a target DQN. Specifically, the evaluation DQN takes \mathcal{S}_α from the environment and $\mathcal{A}_{\mathcal{S}_\alpha}$ from replay memory as the input. The corresponding $Q\{\mathcal{S}_\beta|\mathcal{S}_\alpha, \mathcal{A}_{\mathcal{S}_\alpha}\}$ is calculated at the UAV to determine the action at the future state \mathcal{S}_β . Furthermore, the target DQN in DQN-FRAS generates a target Q-function $\hat{Q}\{\mathcal{S}_\beta|\mathcal{S}_\alpha, \mathcal{A}_{\mathcal{S}_\alpha}\}$ in Q-learning update at the UAV. The target DQN has same structure as the evaluation DQN but different parameters, since the input of \mathcal{S}_α and $\mathcal{A}_{\mathcal{S}_\alpha}$ to the target DQN is the learning experience from the replay memory.

In the evaluation DQN, a set of learning weights φ_j in $Q\{\mathcal{S}_\beta|\mathcal{S}_\alpha, \mathcal{A}_{\mathcal{S}_\alpha}; \varphi_j\}$ are adapted over multiple iterations to approximate the target Q-function, where $j \leq J$ and J is the number of learning steps. Moreover, the parameters of the target DQN are cloned from the evaluation DQN every J steps, in the other words, J counts the number of learning updates in DQN-FRAS.

Based on the DQN-FRAS architecture, Algorithm 1 demonstrates the deep reinforcement learning for the on-

Algorithm 1 Deep Q-Network based Flight Resource Allocation Scheme (DQN-FRAS)

```

1: 1. Initialize:
2:  $\mathcal{S}_\alpha \in \mathcal{S}$ ,  $\mathcal{A}_{\mathcal{S}_\alpha} \in \mathcal{A}$ ,  $\varphi_1$ , and  $\omega$ .
3: Random weights  $\varphi_j \rightarrow Q\{\mathcal{S}_\beta|\mathcal{S}_\alpha, \mathcal{A}_{\mathcal{S}_\alpha}; \varphi_j\}$ .
4: Weights  $\varphi_{j-1} = \varphi_j \rightarrow \hat{Q}\{\mathcal{S}_\beta|\mathcal{S}_\alpha, \mathcal{A}_{\mathcal{S}_\alpha}; \varphi_{j-1}\}$ .
5: Replay memory  $\mathbb{D}[\cdot] \rightarrow$  capacity  $F$ .
6: 2. DQN:
7: for  $episode = 1 \rightarrow M$  do
8:   if Probability  $\zeta$  then
9:     Determine a random waypoint for the UAV, select a random IoT node for data collection, and allocate a random transmit power of the node  $\rightarrow \mathcal{A}_{\mathcal{S}_\alpha}$ .
10:   else
11:      $\mathcal{A}_{\mathcal{S}_\alpha} \rightarrow \operatorname{argmin}_{\mathcal{A}_{\mathcal{S}_\alpha}} C\{\mathcal{S}_\beta|\mathcal{S}_\alpha, \mathcal{A}_{\mathcal{S}_\alpha}\}$ .
12:   end if
13:   The UAV conducts action  $\mathcal{A}_{\mathcal{S}_\alpha}$  in the environment.
14:   Obtain  $C\{\mathcal{S}_\beta|\mathcal{S}_\alpha, \mathcal{A}_{\mathcal{S}_\alpha}\}$  and  $\mathcal{S}_\beta$ .
15:   Store  $(\mathcal{S}_\alpha, \mathcal{S}_\beta, \mathcal{A}_{\mathcal{S}_\alpha}, C\{\mathcal{S}_\beta|\mathcal{S}_\alpha, \mathcal{A}_{\mathcal{S}_\alpha}\})$  into  $F$ .
16:   Sample random minibatch of transition  $(\mathcal{S}_\alpha, \mathcal{S}_\beta, \mathcal{A}_{\mathcal{S}_\alpha}, C\{\mathcal{S}_\beta|\mathcal{S}_\alpha, \mathcal{A}_{\mathcal{S}_\alpha}\})$  from  $\mathbb{D}[\cdot]$ .
17:   if  $\mathcal{S}_\beta$  terminates at step  $j + 1$  then
18:     Set  $c_j \rightarrow C\{\mathcal{S}_\beta|\mathcal{S}_\alpha, \mathcal{A}_{\mathcal{S}_\alpha}\}$ .
19:   else
20:     Set  $c_j \rightarrow C\{\mathcal{S}_\beta|\mathcal{S}_\alpha, \mathcal{A}_{\mathcal{S}_\alpha}\} + \kappa \min_{\mathcal{A}_{\mathcal{S}_\beta} \in \mathcal{A}} \hat{Q}\{\mathcal{S}_{\beta'}|\mathcal{S}_\beta, \mathcal{A}_{\mathcal{S}_\beta}; \varphi_{j-1}\}$ .
21:   end if
22:   Derive  $\Gamma(\varphi_j) \rightarrow (c_j - Q\{\mathcal{S}_\beta|\mathcal{S}_\alpha, \mathcal{A}_{\mathcal{S}_\alpha}; \varphi_j\})^2$ .
23:   if There are  $J$  number of learning steps. then
24:     Synchronize  $\varphi_{j-1} \rightarrow \varphi_j$ .
25:      $\hat{Q}\{\mathcal{S}_\beta|\mathcal{S}_\alpha, \mathcal{A}_{\mathcal{S}_\alpha}; \varphi_{j-1}\} \rightarrow Q\{\mathcal{S}_\beta|\mathcal{S}_\alpha, \mathcal{A}_{\mathcal{S}_\alpha}; \varphi_j\}$ .
26:   end if
27: end for
28: 3. Output:
29: Next waypoint of the UAV  $\rightarrow p_{\text{uav}}^*$ . Schedule the IoT node  $i$  for data collection, and allocate  $P_i^*$ .

```

board DQN-FRAS, which is carried out at the UAV. The starting state and learning time are initialized to \mathcal{S}_α and t_{learning} , respectively. A ζ -greedy strategy is carried out. Specifically, with the probability of $1 - \zeta$, the UAV conducts action $\mathcal{A}_{\mathcal{S}_\alpha}$ which minimizes $C\{\mathcal{S}_\beta|\mathcal{S}_\alpha, \mathcal{A}_{\mathcal{S}_\alpha}\}$. Otherwise, the UAV randomly determines its next waypoint, the IoT node for data collection, and transmit power of the selected node. The next state is updated to $\mathcal{A}_{\mathcal{S}_\beta}$. Consequently, the approximated outputs of DQN-FRAS, i.e., $Q\{\mathcal{S}_\beta|\mathcal{S}_\alpha, \mathcal{A}_{\mathcal{S}_\alpha}; \varphi_j\}$, can be minimized by performing φ_j^* at the j th learning step.

In addition, once the optimal waypoint of the UAV is

$$\mathbb{D}[\mathcal{S}_\alpha, \mathcal{S}_\beta, \mathcal{A}_{\mathcal{S}_\alpha}, C\{\mathcal{S}_\beta|\mathcal{S}_\alpha, \mathcal{A}_{\mathcal{S}_\alpha}\}] = \mathbb{D}[(C\{\mathcal{S}_\beta|\mathcal{S}_\alpha, \mathcal{A}_{\mathcal{S}_\alpha}\} + \kappa \min_{\mathcal{A}_{\mathcal{S}_\beta} \in \mathcal{A}} Q\{\mathcal{S}_{\beta'}|\mathcal{S}_\beta, \mathcal{A}_{\mathcal{S}_\beta}; \varphi_{j-1}\} - Q\{\mathcal{S}_\beta|\mathcal{S}_\alpha, \mathcal{A}_{\mathcal{S}_\alpha}; \varphi_j\})^2] \quad (10)$$

determined and the IoT node is selected by DQN-FRAS, the optimal action of allocating the transmit power $P_i^s(t)^*$ (refers to [42]) is carried out.

The typical reinforcement learning approaches, such as Q-learning, can solve the small-scale resource allocation optimization. However, Q-learning suffers from the well-known curse of dimensionality, which is impractical for the complex cruise control and data collection problem. As observed in Figure 4 and the implementation in Algorithm 1, the proposed DQN-FRAS synchronously maintains two separate Q-networks onboard the UAV, i.e., a target DQN (that is $Q\{\mathcal{S}_\beta|\mathcal{S}_\alpha, k; \varphi_j\}$) and an evaluation DQN (that is $\hat{Q}\{\mathcal{S}_\beta|\mathcal{S}_\alpha, \mathcal{A}_{\mathcal{S}_\alpha}; \varphi_{j-1}\}$). Every J steps, φ_{j-1} that is repeatedly updated at each episode is synchronized to φ_j . Moreover, $\Gamma(\varphi_j)$ is minimized by training DQN-FRAS, which achieves the minimum mean-squared Bellman error. Therefore, the optimality of DQN-FRAS can be achieved asymptotically with the growing size of state and action spaces.

D. Complexity, convergence, and energy

The computational complexity of the proposed DQN-FRAS depends on the state space and the action space in the training phase. The state space of the proposed POMDP is $(BLH)^N(W-1)S$. The size of the action space is $|\mathcal{A}_{\mathcal{S}_\alpha}|$. As a result, the complexity for of DQN-FRAS is $\mathcal{O}((BLH)^N(W-1)S|\mathcal{A}_{\mathcal{S}_\alpha}|)$ in the training phase. It is worth mentioning that the initial training can be potentially conducted offline by using synthetic data or previously captured experimental data, as suggested in [43]. During the online training stage, the UAV updates the onboard deep neural network once per waypoint. At each waypoint, the instant complexity of the DQN-FRAS only depends on the number of layers and the number of nodes per layer in the deep neural network.

According to (8), $Q\{\mathcal{S}_\beta|\mathcal{S}_\alpha, \mathcal{A}_{\mathcal{S}_\alpha}; \varphi_j\}$ in the DQN of the proposed DQN-FRAS technique approximates the optimal Q-function $Q^*\{\mathcal{S}_\beta|\mathcal{S}_\alpha, \mathcal{A}_{\mathcal{S}_\alpha}^*\}$, and the approximation is expected to be increasingly accurate by iteratively updating the weights of the DQN, i.e., φ_j . Here, j indicates the j -th iteration of the DQN-FRAS. Recall that $C\{\mathcal{S}_\beta|\mathcal{S}_\alpha, \mathcal{A}_{\mathcal{S}_\alpha}\}$ is the network cost when the action $\mathcal{A}_{\mathcal{S}_\alpha}$ is carried out at state \mathcal{S}_α , and $\mathcal{S}_{\beta'}$ is the next state of \mathcal{S}_β . We have:

$$Q\{\mathcal{S}_\beta|\mathcal{S}_\alpha, \mathcal{A}_{\mathcal{S}_\alpha}\} \geq C\{\mathcal{S}_\beta|\mathcal{S}_\alpha, \mathcal{A}_{\mathcal{S}_\alpha}\} + \hat{Q}^*\{\mathcal{S}_{\beta'}|\mathcal{S}_\beta, \mathcal{A}_{\mathcal{S}_\beta}; \varphi_j^*\}, \quad (11)$$

where

$$\varphi_j^* = \arg \min Q\{\mathcal{S}_\beta|\mathcal{S}_\alpha, \mathcal{A}_{\mathcal{S}_\alpha}; \varphi_j\}, \quad (12)$$

and the inequality in (11) is because the target Q-function at the next state \mathcal{S}_β , i.e., $\hat{Q}^*\{\mathcal{S}_{\beta'}|\mathcal{S}_\beta, \mathcal{A}_{\mathcal{S}_\beta}; \varphi_j^*\}$, is minimized by optimizing the weight φ_j^* so that the cost can increasingly approach the minimum of $Q\{\mathcal{S}_\beta|\mathcal{S}_\alpha, \mathcal{A}_{\mathcal{S}_\alpha}\}$ at state \mathcal{S}_α . Thus, the right-hand-side of (11) is smaller than $Q\{\mathcal{S}_\beta|\mathcal{S}_\alpha, \mathcal{A}_{\mathcal{S}_\alpha}\}$.

Define the initial loss function (before the first iteration of the DQN-FRAS) as $\Gamma(\varphi_0) = Q\{\mathcal{S}_\beta|\mathcal{S}_\alpha, \mathcal{A}_{\mathcal{S}_\alpha}; \varphi_0\}$. The

loss function of the DQN-FRAS at the $(j+1)$ -th iteration, i.e., $\Gamma(\varphi_{j+1})$, can be derived based on the loss function at the j -th iteration, as given by

$$\Gamma(\varphi_{j+1}) = C\{\mathcal{S}_\beta|\mathcal{S}_\alpha, \mathcal{A}_{\mathcal{S}_\alpha}\} + \omega\Gamma(\varphi_j) + (1-\omega)\hat{Q}\{\mathcal{S}_{\beta'}|\mathcal{S}_\beta, \mathcal{A}_{\mathcal{S}_\beta}; \varphi_j\}, \quad (13)$$

where ω is the learning rate. Since $Q^*\{\mathcal{S}_\beta|\mathcal{S}_\alpha, \mathcal{A}_{\mathcal{S}_\alpha}^*\}$ is the optimal solution to (13), we have [44]

$$Q^*\{\mathcal{S}_\beta|\mathcal{S}_\alpha, \mathcal{A}_{\mathcal{S}_\alpha}^*\} = C\{\mathcal{S}_\beta|\mathcal{S}_\alpha, \mathcal{A}_{\mathcal{S}_\alpha}\} + \omega Q^*\{\mathcal{S}_{\beta'}|\mathcal{S}_\beta, \mathcal{A}_{\mathcal{S}_\beta}; \varphi_j^*\} + (1-\omega)\hat{Q}^*\{\mathcal{S}_{\beta'}|\mathcal{S}_\beta, \mathcal{A}_{\mathcal{S}_\beta}; \varphi_j^*\}, \quad (14)$$

$\Gamma(\varphi_{j+1}) \leq \Gamma(\varphi_j)$ and, therefore, $\lim_{j \rightarrow \infty} \Gamma(\varphi_j) = Q^*\{\mathcal{S}_\beta|\mathcal{S}_\alpha, \mathcal{A}_{\mathcal{S}_\alpha}^*\}$. In other words, the loss function is minimized by training φ_j for a sufficient number of iterations. The DQN-FRAS eventually converges to the optimal Q-function $Q^*\{\mathcal{S}_\beta|\mathcal{S}_\alpha, \mathcal{A}_{\mathcal{S}_\alpha}^*\}$, as also verified numerically in Section V.

UAVs are becoming increasingly less restrictive in terms of energy due to new advancements of battery and energy harvesting technologies. For example, a fixed-wing UAV of a considerable size can be equipped with lightweight high-capacity rechargeable batteries and solar panels on top of its wings. The UAV can fly over a long distance and hover in the air for an extended period [45], [46]. On the other hand, the energy consumption of deep reinforcement learning has been dramatically reduced to make the online control of the UAVs possible, as shown in the recent literature [31], [47]. Energy-efficient hardware platforms, such as FPGA, have been reported to implement deep neural networks [48]. Special-purpose ASIC chips have been designed and fabricated for different deep learning applications [49]. In addition, the proposed DQN-FRAS can be potentially trained offline first and then refined online, as suggested in [43]. By the means, the UAV only needs to update the DQN once per waypoint. From all these aspects, the computational complexity of the proposed DQN-FRAS is practically affordable in future UAV platforms.

V. PERFORMANCE EVALUATION

In this section, we first demonstrate the implementation of the proposed DQN-FRAS framework on Google TensorFlow (the symbolic math library for numerical computation) [50]. Then, the numerical results are presented to analyze the packet loss in terms of the flight cruise training, number of IoT nodes, and the data queue length.

A. TensorFlow settings

N number of IoT nodes are randomly deployed, where N increases from 10 to 80. Each IoT node has the maximum discretized battery capacity $B = 150$, the highest modulation $R = 5$, and the maximum transmit power 100 milliwatts. The data queue length L increases from 20 to 100. The data packets are generated by the IoT node per time slot according to poisson random distribution, and put

TABLE II: Configuration of simulations

Parameters	Values
Number of IoT nodes (N)	10 ~ 80
Battery capacity (B)	150
The highest modulation order (R)	5
Buffer size (L)	20 ~ 100
Total waypoints (W)	10 ~ 80
Required BER (ϵ)	0.05%
Number of laps of the flight (S)	20
Number of learning iterations (J)	10 ~ 600
The discount factor (κ)	0.99
The learning weight (ω)	0.00025
Capacity of the replay memory (F)	10000

into the queue. Data payload of each packet has 128 bytes, and $\epsilon = 0.05\%$.

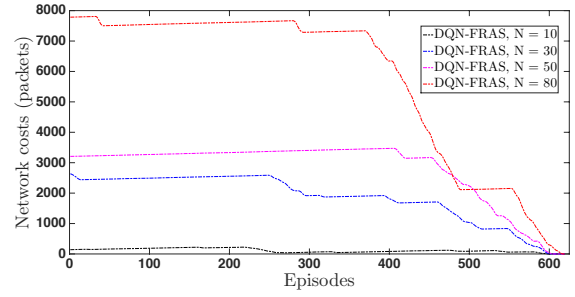
Moreover, the region of interest is set to be a square area with a size of 1000 x 1000 meters, where the IoT nodes are distributed in the targeted region. Following Fig. 3, the targeted region can be evenly divided to 80 blocks at most, each of which has a unit size. Hence, the UAV has $W = 80$ waypoints in the region. We set the communication range of the UAV to be 50 meters given a fixed flight altitude, i.e., $z_{\text{uav}}^s(t) = 20$ meters, unless otherwise specified. The starting point of the UAV is initialized to $p_{\text{uav}}^1(0) = (x_{\text{uav}}^1(0), y_{\text{uav}}^1(0), z_{\text{uav}}^1(0))$. The total number of laps that the UAV patrols is $S = 20$.

DQN-FRAS is implemented in Python 3.5 on TensorFlow with Keras [51] (the Python deep learning library). A Jonsbo UMX4 workstation [52] with an NVIDIA GeForce GTX Titan X GPU based on 64-bit Ubuntu 18.04 is used for the TensorFlow setup. Deep reinforcement learning trains DQN-FRAS for 900 episodes, each of which has 500 epochs. The discount factor and learning rate are set to $\kappa = 0.99$ and $\omega = 0.00025$, respectively. In terms of our implementation, DQN-FRAS trains the flight resource allocation by creating a session `tf.Session()` in TensorFlow. The deep neural network is built by defining a `neural_network_model()` function, where the network state holder and the Q-function holder are initialized by `tf.placeholder()`. The loss function $\Gamma(\varphi_j)$ is constructed by `tf.losses.mean_squared_error(Q-function holder, neural_network_model)`, and minimized by `tf.train.AdamOptimizer().minimize(\Gamma(\varphi_j))`.

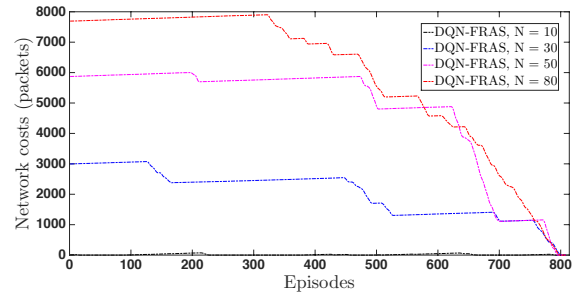
The experience replay memory with capacity $F = 10000$ is created by `Memory(F)` in Tensorflow, and stores the learning experiences at every step by calling `memory.add_sample(current_state, actionsCruiseControlNodeSelection, packet_loss, next_states_array)`. The experiences can be retrieved by `memory.retrieve(minibatch)`, where the minibatch size is 32. The simulation parameters are listed in Table II.

B. Numerical results of DQN-FRAS

1) *DQN-FRAS training*: Figures 5(a) and 5(b) show the network cost, i.e., packet loss of the IoT nodes, where each IoT node generates 100 data packets and κ is set to 0.99 and 0.7, respectively. DQN-FRAS has a high network packet loss at the first 300 episodes during the training given $N =$



(a) Network cost while training DQN-FRAS, where $\kappa = 0.99$.



(b) Network cost while training DQN-FRAS, where $\kappa = 0.7$.

Fig. 5: Network cost (i.e., the overall packet loss) in terms of the number of IoT nodes, where κ is set to 0.99 or 0.7.

10, 30, 50, or 80. Then, the overall packet loss substantially drops as the flight cruise and node selection strategy is optimally trained by the onboard DQN. In particular, the packet loss of DQN-FRAS with $\kappa = 0.99$ falls to 0 in 638 episodes. However, the convergence of DQN-FRAS with $\kappa = 0.7$ requires more than 188 episodes. This can be confirmed by (8), namely, the duration of training the onboard DQN can be reduced by a high discount factor.

2) *Flight cruise training at the UAV*: Figure 6 presents flight trajectories of the UAV that is trained by the proposed DQN-FRAS scheme with regards to $W = 10, 50$, or 80. Specifically, Figures 6(a), (b), and (c) show the trajectories at the beginning of training given $J = 10$. DQN-FRAS determines W waypoints and calculates the Q-function based on the observed states on the UAV (i.e., waypoints, channel conditions) and the IoT nodes (i.e., data queues, battery levels, TTA values). Since the experience in the replay memory is not sufficient, the flight cruise of the UAV is hardly optimized given a small number of learning iterations. Figures 6(d), (e), and (f) plot the trajectories of the UAV, where J increases to 600. By taking advantage of the experience replay, the target DQN and evaluation DQN in DQN-FRAS are adequately trained to minimize the loss functions.

3) *Network packet loss rate*: For performance comparison, the proposed DQN-FRAS scheme is compared with two other state-of-the-art trajectory planning and scheduling approaches as Random Walk and Random Scheduling (RWRS) policy and Channel-Aware Walk and Round-Robin

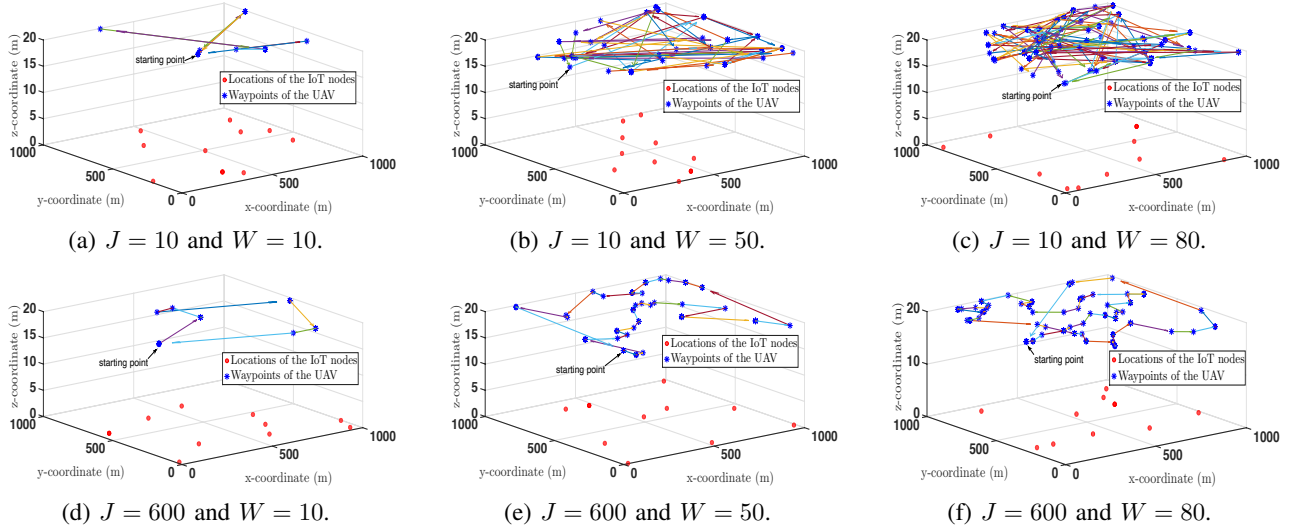


Fig. 6: Training the flight cruise of the UAV with regards to different number of learning iterations and waypoints.

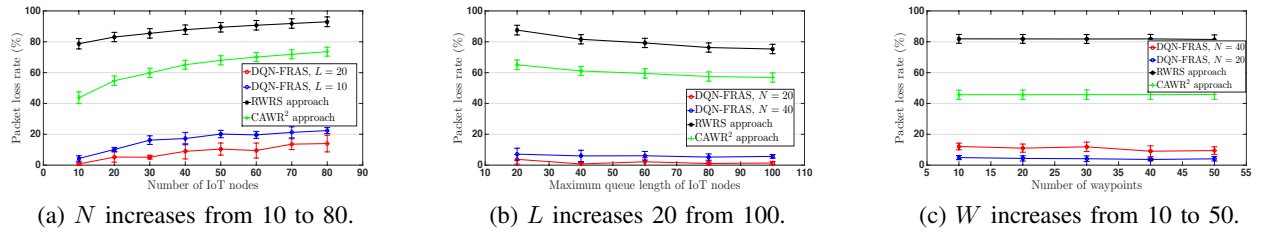


Fig. 7: Packet loss rate with regards to number of IoT nodes, data queue sizes, and waypoints. Each error bar presents the standard deviation over 30 experiments.

Scheduling (CAWR²) policy.

- RWRS randomly determines the next waypoint of the UAV while one of the IoT nodes in the communication range is scheduled to transmit data at each waypoint. The flight cruise and node selection are independent of UAV's current location, battery, TTA and data queue of the IoT node, or channel variation.
- CAWR² sets the next waypoint to the location with the highest RSS, where a-prior knowledge on the channels in the target field is assumed to be known to the UAV. At each waypoint, CAWR² schedules the data transmission of the IoT nodes in a round robin order.

Figure 7 studies packet loss rate of RWRS, CAWR², and the proposed DQN-FRAS scheme with an increasing number of IoT nodes, data queue sizes, or waypoints. In particular, the settings of L and N of RWRS and CAWR² in Figure 7 are fixed to 20 and 40, respectively.

In Figure 7(a), the packet loss rate of RWRS, CAWR², and DQN-FRAS grows with the IoT network size since more IoT nodes have to buffer their data while one node is scheduled to transmit data. Significantly, DQN-FRAS with $L = 10$ achieves about 69% and 51% lower than RWRS and CAWR². This is because DQN-FRAS jointly optimizes the movements of the UAV and the selection of the IoT node

to minimize the buffer overflow and failed transmission. In addition, the packet loss rate of DQN-FRAS with $L = 20$ is 4% lower than the one with $L = 10$. This confirms the fact that extending the queuing space at the IoT node can reduce the buffer overflow.

Figure 7(b) depicts the packet loss rate of DQN-FRAS when data queue size L is extended from 20 to 100. It is observed that DQN-FRAS reduces the packet loss rate to a greater extent than RWRS and CAWR². In particular, given $N = 40$ and $L = 20$, DQN-FRAS outperforms RWRS and CAWR² on the packet loss rate by 76% and 54%, respectively. Although extending the data queue size slightly reduces the packet loss of RWRS and CAWR², DQN-FRAS still achieves 66% and 48% lower packet loss rate when $L = 100$.

Figure 7(c) shows the packet loss rate in regards to the number of waypoints on the flight cruise. Given $N = 40$ or 20, the packet loss rate of DQN-FRAS decreases to 12% or 7% when W increases from 10 to 50. On the contrary, the performance of RWRS and CAWR² is hardly improved by adding the number of waypoints to the flight cruise. The reason is that DQN-FRAS optimizes the future waypoints of the UAV at every location by taking advantage of the learning experience in the replay memory. Adding

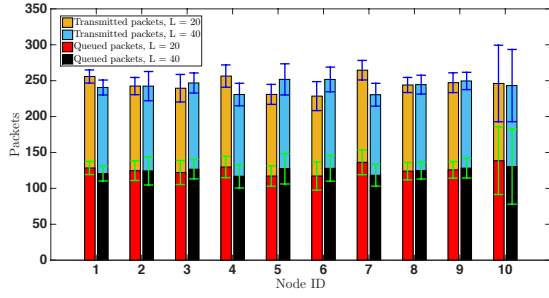


Fig. 8: The number of packets generated and transmitted by the IoT nodes given $N = 10$ and $W = 10$, where the standard deviation is calculated based on 30 experiments.

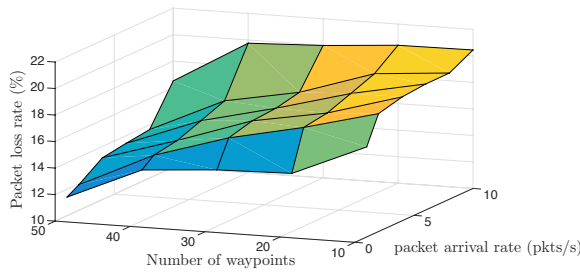


Fig. 9: Packet loss rate achieved by DQN-FRAS with regards to the number of waypoints and data traffic of the IoT nodes.

waypoints on the flight cruise extends the action space and learning experiences in DQN-FRAS.

4) Queued and transmitted packets with DQN-FRAS:

Figure 8 illustrates the number of generated (stored in the data queue) and transmitted packets with DQN-FRAS, where $N = 10$. It can be observed that most of the packets in the data queue of the IoT node are successfully transmitted to the UAV. This is benefited from the joint optimization of flight cruise and data capture in DQN-FRAS. Interestingly, we also see that the number of transmitted packets of the IoT nodes are similar to each other given a specific data queue size. This is because TTA values of the IoT nodes ensure that DQN-FRAS optimizes the flight cruise of the UAV to cover all the IoT nodes, thus minimizing the network cost of each of them.

5) Cruise control of the UAV and data traffic of the IoT nodes:

In this case, we study the impact of waypoints of the UAV and the average packet arrival rate of the IoT nodes on the proposed DQN-FRAS scheme. Figure 9 plots the packet loss rate achieved by DQN-FRAS with regards to the number of waypoints and the average packet arrival rate, where $N = 20$ and $L = 20$. Given a specific number of waypoints, the packet loss rate of DQN-FRAS grows with the average packet arrival rate at the IoT nodes. This is because increasing the packet arrival rate at the IoT nodes gives rise to more data packets queuing in the buffer. As one of the IoT nodes can be selected at each time slot for data transmission, the other nodes have to buffer the newly arrived data, which results in data queue overflows. On the other hand, the packet loss rate can be reduced by adding

waypoints to the flight cruise of the UAV. Given the average packet arrival rate of 10, the packet loss rate drops from around 20.5% to 17.5% when the number of waypoints increases by five times.

Essentially, Figure 9 implies a tradeoff between the data traffic at the ground IoT network and the flight control at the UAV. Specifically, the high packet arrival rate expedites the buffer overflows in UAV-aided IoT networks and in turn, increasing the packet loss rate. However, extending the number of waypoints on the flight cruise of the UAV allows to schedule the IoT nodes with higher channel condition to reduce the buffer overflow. Therefore, the number of waypoints and the data traffic of the IoT nodes need to be balanced, so as to achieve the minimized packet loss rate.

VI. CONCLUSION

This paper studies the joint flight cruise control and data collection scheduling in the UAV-aided IoT network. The flight resource allocation is formulated as a POMDP to minimize the data lost due to buffer overflows at the IoT nodes and fading airborne channels. Given the large state and action spaces, onboard DQN-FRAS is proposed to optimally determine the instantaneous waypoints for the flight cruise control, the selection of the IoT node for the data collection, and the transmit power of the selected IoT node. For accelerating the training of the onboard DQN-FRAS, the potential next waypoints of the UAV in the action space are further reduced by the restricted heading direction and the maximum cruising speed. Furthermore, the proposed DQN-FRAS is implemented by using Keras deep learning library with Google TensorFlow. Numerical results demonstrate that the trajectories of the UAV can be sufficiently trained to minimize the loss function by extending the learning iterations.

ACKNOWLEDGEMENTS

This work was partially supported by National Funds through FCT/MCTES (Portuguese Foundation for Science and Technology), within the CISTER Research Unit (UIDB/04234/2020); also by the Operational Competitiveness Programme and Internationalization (COMPETE 2020) under the PT2020 Partnership Agreement, through the European Regional Development Fund (ERDF), and by national funds through the FCT, within project(s) POCI-01-0145-FEDER-029074 (ARNET).

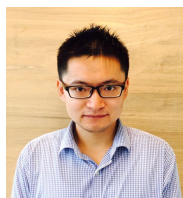
REFERENCES

- [1] O. B. Akan, O. Cetinkaya, C. Koca, and M. Ozger, "Internet of hybrid energy harvesting things," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 736–746, 2017.
- [2] P. Kamalinejad, C. Mahapatra, Z. Sheng, S. Mirabbasi, V. C. Leung, and Y. L. Guan, "Wireless energy harvesting for the internet of things," *IEEE Communications Magazine*, vol. 53, no. 6, pp. 102–108, 2015.
- [3] M. L. Ku, W. Li, Y. Chen, and K. R. Liu, "Advances in energy harvesting communications: Past, present, and future challenges," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1384–1412, 2015.

- [4] N. H. Motlagh, T. Taleb, and O. Arouk, "Low-altitude unmanned aerial vehicles-based internet of things services: Comprehensive survey and future perspectives," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 899–922, 2016.
- [5] N. H. Motlagh, M. Bagaa, and T. Taleb, "UAV-based IoT platform: A crowd surveillance use case," *IEEE Communications Magazine*, vol. 55, no. 2, pp. 128–134, 2017.
- [6] H. Menouar, I. Guvenc, K. Akkaya, A. S. Uluagac, A. Kadri, and A. Tuncer, "UAV-enabled intelligent transportation systems for the smart city: Applications and challenges," *IEEE Communications Magazine*, vol. 55, no. 3, pp. 22–28, 2017.
- [7] W. Fawaz, C. Abou-Rjeily, and C. Assi, "UAV-aided cooperation for FSO communication systems," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 70–75, 2018.
- [8] Softbank hopes its solar internet drone will soar where facebook's and google's sank. [Online]. Available: <https://spectrum.ieee.org/tech-talk/telecom/internet/softbank-hopes-its-solar-internet-drone-will-soar-where-facebooks-and-googles-sank>
- [9] Facebook takes flight. [Online]. Available: <https://www.theverge.com/a/mark-zuckerberg-future-of-facebook/aquila-drone-internet>
- [10] Project loon. [Online]. Available: <https://www.loon.com/>
- [11] 3GPP: study on enhanced support for aerial vehicles. [Online]. Available: <http://www.3gpp.org/dynareport/36777.htm>
- [12] J. Chen, R. Zhang, L. Song, Z. Han, and B. Jiao, "Joint relay and jammer selection for secure two-way relay networks," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 1, pp. 310–320, 2011.
- [13] L. Xie, J. Xu, and R. Zhang, "Throughput maximization for UAV-enabled wireless powered communication networks," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1690–1703, 2018.
- [14] H. Wang, G. Ding, F. Gao, J. Chen, J. Wang, and L. Wang, "Power control in UAV-supported ultra dense networks: Communications, caching, and energy transfer," *IEEE Communications Magazine*, vol. 56, no. 6, pp. 28–34, 2018.
- [15] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3–4, pp. 279–292, 1992.
- [16] L. Prashanth and S. Bhatnagar, "Reinforcement learning with function approximation for traffic signal control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 412–421, 2010.
- [17] J. O'Neill, B. Pleydell-Bouverie, D. Dupret, and J. Csicsvari, "Play it again: reactivation of waking experience and memory," *Trends in neurosciences*, vol. 33, no. 5, pp. 220–229, 2010.
- [18] Q. Wu, L. Liu, and R. Zhang, "Fundamental trade-offs in communication and trajectory design for UAV-enabled wireless network," *IEEE Wireless Communications*, vol. 26, no. 1, pp. 36–44, 2019.
- [19] D. Yang, Q. Wu, Y. Zeng, and R. Zhang, "Energy tradeoff in ground-to-UAV communication via trajectory design," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 7, pp. 6721–6726, 2018.
- [20] Y. Zeng, X. Xu, and R. Zhang, "Trajectory design for completion time minimization in UAV-enabled multicasting," *IEEE Transactions on Wireless Communications*, vol. 17, no. 4, pp. 2233–2246, 2018.
- [21] J. Xu, Y. Zeng, and R. Zhang, "UAV-enabled wireless power transfer: Trajectory design and energy optimization," *IEEE Transactions on Wireless Communications*, vol. 17, no. 8, pp. 5092–5106, 2018.
- [22] Q. Wu, Y. Zeng, and R. Zhang, "Joint trajectory and communication design for multi-UAV enabled wireless networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 3, pp. 2109–2121, 2018.
- [23] C. Zhan, Y. Zeng, and R. Zhang, "Trajectory design for distributed estimation in UAV-enabled wireless sensor network," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 10, pp. 10 155–10 159, 2018.
- [24] M. Cui, G. Zhang, Q. Wu, and D. W. K. Ng, "Robust trajectory and transmit power design for secure UAV communications," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 9, pp. 9042–9046, 2018.
- [25] Z. M. Fadlullah, D. Takaishi, H. Nishiyama, N. Kato, and R. Miura, "A dynamic trajectory control algorithm for improving the communication throughput and delay in UAV-aided networks," *IEEE Network*, vol. 30, no. 1, pp. 100–105, 2016.
- [26] F. Jiang and A. L. Swindlehurst, "Optimization of UAV heading for the ground-to-air uplink," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 5, pp. 993–1005, 2012.
- [27] F. Wu, H. Zhang, J. Wu, and L. Song, "Cellular UAV-to-device communications: Trajectory design and mode selection by multi-agent deep reinforcement learning," *IEEE Transactions on Communications*, 2020.
- [28] M. A. Abd-Elmagid, A. Ferdowsi, H. S. Dhillon, and W. Saad, "Deep reinforcement learning for minimizing age-of-information in UAV-assisted networks," in *IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.
- [29] U. Challita, W. Saad, and C. Bettstetter, "Interference management for cellular-connected UAVs: A deep reinforcement learning approach," *IEEE Transactions on Wireless Communications*, vol. 18, no. 4, pp. 2125–2140, 2019.
- [30] K. Li, W. Ni, E. Tovar, and A. Jamalipour, "On-board deep Q-network for UAV-assisted online power transfer and data collection," *IEEE Transactions on Vehicular Technology*, 2019.
- [31] C. H. Liu, Z. Chen, J. Tang, J. Xu, and C. Piao, "Energy-efficient UAV control for effective and fair communication coverage: A deep reinforcement learning approach," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 9, pp. 2059–2070, 2018.
- [32] K. Li, W. Ni, X. Wang, R. P. Liu, S. S. Kanhere, and S. Jha, "Energy-efficient cooperative relaying for unmanned aerial vehicles," *IEEE Transactions on Mobile Computing*, no. 6, pp. 1377–1386, 2016.
- [33] N. Zhao, F. Cheng, F. R. Yu, J. Tang, Y. Chen, G. Gui, and H. Sari, "Caching UAV assisted secure transmission in hyper-dense networks based on interference alignment," *IEEE Transactions on Communications*, vol. 66, no. 5, pp. 2281–2294, 2018.
- [34] L. Xiao, X. Lu, D. Xu, Y. Tang, L. Wang, and W. Zhuang, "UAV relay in VANETs against smart jamming with reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 5, pp. 4087–4097, 2018.
- [35] K. Li, R. C. Voicu, S. S. Kanhere, W. Ni, and E. Tovar, "Energy efficient legitimate wireless surveillance of UAV communications," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2283–2293, 2019.
- [36] Y. Cao, L. Zhang, and Y.-C. Liang, "Deep reinforcement learning for channel and power allocation in UAV-enabled IoT systems," in *IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.
- [37] Q. Zhang, M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Machine learning for predictive on-demand deployment of UAVs for wireless communications," in *IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2018, pp. 1–6.
- [38] S. Morton, R. D'Sa, and N. Papanikolopoulos, "Solar powered UAV: Design and experiments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 2460–2466.
- [39] H. Cotuk, K. Bicakci, B. Tavli, and E. Uzun, "The impact of transmission power control strategies on lifetime of wireless sensor networks," *IEEE Transactions on Computers*, vol. 63, no. 11, pp. 2866–2879, 2013.
- [40] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.
- [41] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synthesis Lectures on Communication Networks*, vol. 3, no. 1, pp. 1–211, 2010.
- [42] K. Li, W. Ni, M. Abolhasan, and E. Tovar, "Reinforcement learning for scheduling wireless powered sensor communications," *IEEE Transactions on Green Communications and Networking*, 2018.
- [43] W. Koch, R. Mancuso, R. West, and A. Bestavros, "Reinforcement learning for UAV attitude control," *ACM Transactions on Cyber-Physical Systems*, vol. 3, no. 2, pp. 1–21, 2019.
- [44] B. Luo, Y. Yang, and D. Liu, "Adaptive Q-learning for data-based optimal output regulation with experience replay," *IEEE transactions on cybernetics*, vol. 48, no. 12, pp. 3337–3348, 2018.
- [45] HAPS Mobile, "Softbank corp. develops aircraft that delivers telecommunications connectivity from the stratosphere," April 2019. [Online]. Available: https://www.softbank.jp/en/corp/news/press/sbkk/2019/20190425_02/
- [46] Z. Liu, "Chinese solar-powered drone morning star spreads its wings in successful test flight," October 2018. [Online]. Available: <https://www.scmp.com/news/china/military/article/2171081/chinese-solar-powered-drone-spreads-its-wings-successful-test>
- [47] C. H. Liu, X. Ma, X. Gao, and J. Tang, "Distributed energy-efficient multi-UAV navigation for long-term communication coverage by deep reinforcement learning," *IEEE Transactions on Mobile Computing*, vol. 19, no. 6, pp. 1274–1285, 2019.
- [48] J. Qiu, J. Wang, S. Yao, K. Guo, B. Li, E. Zhou, J. Yu, T. Tang, N. Xu, S. Song *et al.*, "Going deeper with embedded fpga platform for convolutional neural network," in *Proceedings of the 2016*

ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, 2016, pp. 26–35.

- [49] W. Dai and D. Berleant, “Benchmarking contemporary deep learning hardware and frameworks: A survey of qualitative metrics,” in *IEEE First International Conference on Cognitive Machine Intelligence (CogMI)*. IEEE, 2019, pp. 148–155.
- [50] 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 *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 2016, pp. 265–283.
- [51] F. Chollet *et al.*, “Keras: Deep learning library for theano and tensorflow,” URL: <https://keras.io/k>, vol. 7, no. 8, p. T1, 2015.
- [52] G. Darksaber. (2017, January) Jonsbo UMX4 review. [Online]. Available: <https://www.techpowerup.com/review/jonsbo-umx4/>.



Kai Li (S’09–M’14–SM’20) received the B.E. degree from Shandong University, China, in 2009, the M.S. degree from The Hong Kong University of Science and Technology, Hong Kong, in 2010, and the Ph.D. degree in Computer Science from The University of New South Wales, Sydney, Australia, in 2014. Currently he is a senior research scientist and project leader at Real-Time and Embedded Computing Systems Research Centre (CISTER), Portugal. He is also a research fellow with Carnegie Mellon Portugal

Research Program. Prior to this, Dr. Li was a postdoctoral research fellow at The SUTD-MIT International Design Centre, The Singapore University of Technology and Design, Singapore (2014–2016). He was a visiting research assistant at ICT Centre, CSIRO, Australia (2012–2013). From 2010 to 2011, he was a research assistant at Mobile Technologies Centre with The Chinese University of Hong Kong. His research interests include vehicular communications and security, resource allocation optimization, Cyber-Physical Systems, Internet of Things (IoT), human sensing systems, sensor networks and UAV networks.

Dr. Li serves as the Associate Editor for IEEE Access Journal, the Demo Co-chair for ACM/IEEE IPSN 2018, the TPC member of MSN’19, EWSN’19, IEEE PerCom’19, Globecom’18, MASS’18, VTC-Spring’18, Globecom’17, VTC’17, and VTC’16.



Wei Ni (M’09–SM’15) received the B.E. and Ph.D. degrees in Electronic Engineering from Fudan University, Shanghai, China, in 2000 and 2005, respectively. Currently, he is a Group Leader and Principal Research Scientist at CSIRO, Sydney, Australia, and an adjunct professor at the University of Technology Sydney and an Honorary Professor at Macquarie University, Sydney. Prior to this, he was a Postdoctoral Research Fellow at Shanghai Jiaotong University from 2005 – 2008; Deputy Project Manager at

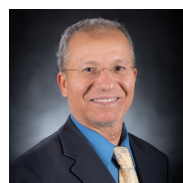
the Bell Labs, Alcatel/Alcatel-Lucent from 2005 – 2008; and Senior Researcher at Devices R&D, Nokia from 2008 – 2009. His research interests include signal processing, stochastic optimization, as well as their applications to network efficiency and integrity.

Dr. Ni is the Chair of IEEE Vehicular Technology Society (VTS) New South Wales (NSW) Chapter since 2020 and an Editor of IEEE Transactions on Wireless Communications since 2018. He served first the Secretary and then Vice-Chair of IEEE NSW VTS Chapter from 2015 – 2019, Track Chair for VTC-Spring 2017, Track Co-chair for IEEE VTC-Spring 2016, Publication Chair for BodyNet 2015, and Student Travel Grant Chair for WPMC 2014.



Eduardo Tovar was born in 1967 and has received the Licenciante, MSc and PhD degrees in electrical and computer engineering from the University of Porto, Porto, Portugal, in 1990, 1995 and 1999, respectively. Currently he is Professor in the Computer Engineering Department at the School of Engineering (ISEP) of Polytechnic Institute of Porto (IPP), where he is also engaged in research on real-time distributed systems, wireless sensor networks, multiprocessor systems, cyber-physical systems and industrial communication systems.

He heads the CISTER Research Unit, an internationally renowned research centre focusing on RTD in real-time and embedded computing systems. He is deeply engaged in research on real-time distributed systems, multiprocessor systems, cyber-physical systems and industrial communication systems. He is currently the Vice-chair of ACM SIGBED (ACM Special Interest Group on Embedded Computing Systems) and was for 5 years, until December 2015, member of the Executive Committee of the IEEE Technical Committee on Real-Time Systems (TC-RTS). Since 1991 he authored or co-authored more than 150 scientific and technical papers in the area of real-time and embedded computing systems, with emphasis on multiprocessor systems and distributed embedded systems. Eduardo Tovar has been consistently participating in top-rated scientific events as member of the Program Committee, as Program Chair or as General Chair. Notably he has been program chair/co-chair for ECRTS 2005, IEEE RTCSA 2010, IEEE RTAS 2013 or IEEE RTCSA 2016, all in the area of real-time computing systems. He has also been program chair/co-chair of other key scientific events in the area of architectures for computing systems and cyber-physical systems as is the case of ARCS 2014 or the ACM/IEEE ICCPS 2016 or in the area of industrial communications (IEEE WFCS 2014).



Mohsen Guizani (S’85–M’89–SM’99–F’09) received the B.S. (with distinction) and M.S. degrees in electrical engineering, the M.S. and Ph.D. degrees in computer engineering from Syracuse University, Syracuse, NY, USA, in 1984, 1986, 1987, and 1990, respectively. He is currently a Professor and the ECE Department Chair at the University of Idaho, USA. Previously, he served as the Associate Vice President of Graduate Studies, Qatar University, Chair of the Computer Science Department, Western

Michigan University, and Chair of the Computer Science Department, University of West Florida. He also served in academic positions at the University of Missouri-Kansas City, University of Colorado-Boulder, Syracuse University, and Kuwait University. His research interests include wireless communications and mobile computing, computer networks, mobile cloud computing, security, and smart grid. He currently serves on the editorial boards of several international technical journals and the Founder and the Editor-in-Chief of Wireless Communications and Mobile Computing journal (Wiley). He is the author of 9 books and more than 500 publications in refereed journals and conferences. He guest edited a number of special issues in IEEE journals and magazines. He also served as a member, Chair, and General Chair of a number of international conferences. He received the teaching award multiple times from different institutions as well as the best Research Award from three institutions. He received the 2017 IEEE ComSoc Recognition Award for his contribution to Wireless Communications. He was the Chair of the IEEE Communications Society Wireless Technical Committee and the Chair of the TAOS Technical Committee. He served as the IEEE Computer Society Distinguished Speaker from 2003 to 2005. He is a Fellow of IEEE and a Senior Member of ACM.