Joint Communication Scheduling and Velocity Control in Multi-UAV-Assisted Sensor Networks: A Deep Reinforcement Learning Approach

Yousef Emami, *Student Member, IEEE*, Bo Wei , Kai Li , *Senior Member, IEEE*, Wei Ni, *Senior Member, IEEE*, and Eduardo Tovar, *Member, IEEE*

***Abstract*—Recently, Unmanned Aerial Vehicle (UAV) swarm has been increasingly studied to collect data from ground sensors in remote and hostile areas. A key challenge is the joint design of the velocities and data collection schedules of the UAVs, as inadequate velocities and schedules would lead to failed transmissions and buffer overflows of sensors and, in turn, significant packet losses. In this paper, we optimize jointly the velocity controls and data collection schedules of multiple UAVs to minimize data losses, adapting to the battery levels, queue lengths and channel conditions of the ground sensors, and the trajectories of the UAVs. In the absence of the up-to-date knowledge of the ground sensors' states, a Multi-UAV Deep Reinforcement Learning based Scheduling Algorithm (MADRL-SA) is proposed to allow the UAVs to asymptotically minimize the data loss of the system under the outdated knowledge of the network states at individual UAVs. Numerical results demonstrate that the proposed MADRL-SA reduces the packet loss by up to 54% and 46% in the considered simulation setting, as compared to an existing DRL solution with single-UAV and non-learning greedy heuristic, respectively.**

***Index Terms*—Unmanned aerial vehicles, communication scheduling, velocity control, multi-UAV deep reinforcement learning, deep Q-Network.**

I. Introduction

THANKS to excellent mobility and maneuverability, unmanned Aerial Vehicles (UAVs) are used in many civilian and commercial applications, e.g., weather monitoring, traffic control, package delivery [1] and crops monitoring [2]. UAVs are also employed to relay data for the ground sensors in harsh environment, such as natural disaster monitoring [3], border surveillance [4] and emergency assistance [5]. We consider a remote environment, where ground sensors are deployed beyond the reach of any terrestrial gateways and have no persistent power supply. A UAV can physically approach each individual ground sensor. The short, line-of-sight (LoS)-dominant communication link between the UAV and a ground sensor enjoys a significant channel gain and supports high-speed data transmission. In this sense, employing the UAVs to collect data can improve the network throughput and extend the coverage range beyond terrestrial gateways. Fig. 1 depicts a typical multi-UAV-assisted wireless sensor network (MA-WSN), where the ground sensors are deployed to monitor temperature and humidity of croplands. Sensory data are generated by the ground sensors and are stored in a data queue for future transmission to the UAVs. UAVs are employed to hover over the cropland, where the UAV can move sufficiently close to each ground sensor, exploiting short-distance LoS communication links, for collecting the data.

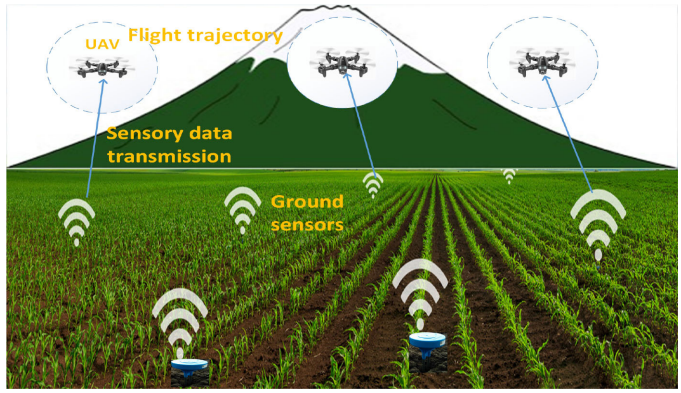


Fig. 1. Multi-UAV assisted wireless sensor networks, where UAVs are employed to collect sensory data of the ground sensors.

In MA-WSN, the ground sensors undergo random data arrivals, since the data generation experiences a random environmental change of the temperature and humidity. As depicted in Fig. 1, the UAVs are employed to hover over a cropland, where a UAV can move sufficiently close to a ground sensor, exploiting short-distance line-of-sight (LoS) communication links

Yousef Emami, Kai Li, and Eduardo Tovar are with the Real-Time and Embedded Computing Systems Research Centre (CISTER), 4249-015 Porto, Portugal (e-mail: emami@isep.ipp.pt; kaili@isep.ipp.pt; emt@isep.ipp.pt).

Bo Wei is with the Department of Computer and Information Sciences, Northumbria University, Newcastle upon Tyne NE1 8ST, U.K. (e-mail: bo.wei@northumbria.ac.uk).

Wei Ni is with Digital Productivity and Services Flagship, Common-wealth Scientific and Industrial Research Organization (CSIRO), Sydney VIC3169, Australia (e-mail: Wei.Ni@data61.csiro.au).

for data collection. However, selecting a ground sensor for data collection may lead to buffer overflows at other sensors, if those sensors' buffers are already full while new data keeps arriving. Moreover, the transmissions of ground sensors which are far away from the UAVs and experience poor channel conditions are prone to errors at the UAVs. The slow mobility of a UAV can give rise to buffer overflows of the ground sensors since newly arrived data is not promptly collected by the UAV. Adequately scheduling data collection coupling with onboard velocity of the UAVs is critical to data queue overflow and communication failure. In addition, the joint velocity control and sensor selection need to be coordinated between participating UAVs. However, it is difficult for the UAVs to share their velocities and selected sensors with each other in real-time due to limited radio coverage and fast movements.

In this paper, joint communication schedule and velocities control of multiple UAVs are formulated as a multi-agent Markov Decision Process (MMDP), which aims to minimize packet loss resulting from buffer overflows and communication failures. In particular, the ground sensor records the visit time whenever a UAV schedules the sensor to transmit data. Moreover, the visiting records of the sensor is shared to the UAV, which is used as an evidence of the other UAVs' communication schedules. The network state of MMDP contains battery levels and data queue lengths of the ground sensors, channel conditions, visit time and the waypoints along the trajectory of the UAVs. The UAVs take actions of selecting the ground sensors for data transmissions, determining modulation schemes and adjusting the patrol velocities. In practice, the up-to-date knowledge of the battery level and data queue length of the ground sensors is not available at the UAVs. Thus, multi-UAV Q-learning can be a solution to training the actions of the UAVs. Since the trajectory of each UAV can consist of a large number of waypoints, the velocity control of the UAVs along the trajectories results in a massive state and action space and high complexity of multi-UAV Q-learning. The main contributions of this paper can be summarized as follows:

- We formulate the problem of joint velocity control and data collection scheduling as an MMDP to minimize the overall packet loss resulting from the buffer overflow and channel fading. To deal with the large state and action spaces, we propose Multi-UAV Deep Reinforcement Learning based Scheduling Algorithm (MADRL-SA) based on Deep Q-networks (DQN) to optimize the selection of the ground sensor, instantaneous patrol velocity of the UAVs and modulation scheme. The UAVs also carry out experience replay to make learning efficient by breaking the correlation between consecutive samples.
- In practice, the online decisions of the UAVs in flight are unknown to each other, which may result in incomplete training of MADRL-SA. To train the actions of a UAV according to the actions of the other UAVs, a local action recording process is developed, where each ground sensor records historical visits of all UAVs. The UAV that schedules the ground sensor to transmit data receives the records, which contains the past scheduling decisions of the other UAVs.

The rest of this paper is organized as follows. Section II reviews the related work on multi-UAV systems. Section III presents the system model. The joint optimization of the velocity control and communication schedule is formulated in Section IV. In Section V, preliminaries is presented then multi-UAV DQN is developed and a new MADRL-SA scheme is designed to optimize the decision process of the MMDP, thereby optimizing the patrol velocities as well as the transmission schedule of the ground sensors. Performance evaluation is presented in Section VI. This paper is concluded in Section VII.

II. Related Work

This section presents the literature on resource allocation and scheduling in multi-UAV systems.

A. Resource Allocation in UAV Networks

The work in [6] develops a framework for trajectory control, user association, and power control in multi-UAV enabled wireless networks. Communication throughput gains can be obtained by mobile UAVs over static UAVs/fixed terrestrial base stations, by exploiting the design degree of freedom via UAV trajectory adjustment. A general mixed integer nonlinear program formulation for a multi-UAV network is presented in [7] to adjust the communication and the computational energy. [8] explores a multi-UAV-aided relaying system, where UAV relays aim to establish communication between senders and receivers and to improve the rate between the pair of sender and receiver, the UAVs' positions are adjusted and resource allocations are conducted. In [9], a cooperative framework designed which allowed the formation of a network between the aerial and the ground nodes. Their approach provides continuous connectivity, enhanced lifetime, and improved coverage in the UAV coordinated WSNs and laid the foundation of guided network formations between the UAVs and the ad hoc networks on the ground. A framework is developed in [10] to improve energy efficiency in deadline-based WSN data collection with multiple UAVs. In [11], the mission completion time is adjusted for multi-UAV-enabled data collection. An energy-efficient transmission scheduling scheme of UAVs in a cooperative relaying network is developed in [12] such that the maximum energy consumption of all the UAVs is minimized, in which an applicable sub-optimal solution is developed and the energy could be saved up to 50% via simulations. In [13] a UAV is used to collect data from time-constrained Internet of Things (IoT) devices. The UAV trajectory and radio resource allocation are adjusted to collect data from IoT devices adapting to their deadline.

B. DRL Approaches

In [14], a single agent DQN for UAV-assisted online power transfer and data collection is developed. However, in most situations, multiple UAVs are needed to interact with each other to solve a resource allocation problem. In [15], online velocity control and data capture are studied in UAV-enabled IoT networks. DQN is developed in the presence of outdated

knowledge to determine the patrolling velocity and data transmission schedule of the IoT node. In [16], the joint flight cruise control and data collection scheduling in the UAV-aided IoT network is formulated as a POMDP to minimize the data lost due to buffer overflows at the IoT nodes and fading airborne channels. A UAV-assisted IoT communication is investigated in [17] where by applying multi-agent DRL a resource allocation scheme adapting to bandwidth, throughput, and interference is obtained.

A wireless powered communication network is developed in [18] where multiple UAVs provide energy supply and communication services to IoT devices. They used a multi-UAV DQN based approach to improve throughput by jointly adjusting UAVs' path design and time resource assignment. They follow an independent learner approach without cooperation between UAVs. In [19], the authors consider long-term, long-distance sensing tasks in a smart city scenario where UAVs make decisions based on DQN for energy-efficient data collection. An energy-saving DRL-based UAV control strategy is developed in [20] to enhance the energy efficiency and communication coverage. They used deep deterministic policy gradient method and take into account communications coverage, fairness, energy consumption and connectivity. In [21], the dueling DQN is employed to adjust the UAV deployment in the multi-UAV wireless networks so that downlink capacity is to be enhanced while covering all ground terminals. They modeled the problem as a constrained MDP problem.

The multi-agent reinforcement learning (MARL) framework is developed in [22] to investigate the dynamic resource allocation problem in UAV networks. A Q-learning based algorithm is developed to enhance the long-term rewards where each UAV runs Q-learning algorithm and automatically selects its communication mode, power levels and sub-channels in concurrent manner. [23] studies spectrum sharing among a network of UAVs. A relaying service is realized by team of UAVs to serve primary users on the ground aiming to gain spectrum access consequently. The gained spectrum belongs to not only UAV relay but also other UAVs that perform the sensing task. The problem is formulated as deterministic MMDP and distributed Q-learning is utilized to solve it.

[24] develops the DRL algorithm based on echo state network cells to find an interference-aware path and allocate resources to the UAVs. The developed scheme reduces wireless latency and improves energy efficiency. The work in [25] adjusts trajectory and power control in multiple UAVs scenarios to enhance the users' throughput and satisfying the users' rate requirement.

The proposed MADRL-SA is different from the MARL framework [22]. In MARL, the UAVs follow an independent learner paradigm, while in MADRL-SA the UAVs cooperate to minimize the packet loss. Moreover, MADRL-SA is for practical scenarios and utilizes DQN unlike MARL which utilizes Q-learning. The work in [14] follows a single UAV approach, while MADRL-SA follows a multi-UAV approach with the merits of scalability and robustness. Our work focuses on minimizing the packet loss and provide velocity control while the work in [19] focuses on energy efficiency and neglect the velocity control, also the UAVs act independently.

III. System Model

The network contains J ground sensors and I UAVs. Our study focuses on the joint velocity control and communication scheduling under preconfigured UAV trajectories. The UAVs fly along pre-determined trajectories which consist of a large number of waypoints to cover all the ground sensors in the field. The trajectories of the UAVs can be predesigned according to the required network capacity [26], coverage [14], or the UAVs' propulsion energy consumption [27]. The optimization of UAV trajectories has been widely studied in the literature [28]–[30]. The proposed MADRL-SA is generic to any given trajectory.

The channel coefficient between the UAV i ($\in [1, I]$) and device j ($\in [1, J]$) at t is $h_j^i(t)$, which can be known by channel reciprocity. The modulation scheme of device j at t is denoted by $\phi_j(t)$. In particular, $\phi_j(t)$= 1, 2, and 3 indicates binary phase-shift keying (BPSK), quadrature-phase shift keying (QPSK), and 8 phase-shift keying (8PSK), respectively, and $\phi_j(t) \geq 4$ provides $2^{\phi_j(t)}$ quadrature amplitude modulation (QAM).

Let $h_j^i(t)$ denote channel gain between ground sensor j and UAV i. The transmit power of the ground sensor, denoted by $P_j^i(t)$, is [31]

$$P_j^i(t) = \frac{ln\frac{k_1}{\epsilon}}{k_2 h_j^i(t)^2}\left(2^{\phi_j(t)} - 1\right) \tag{1}$$

where k_1 and k_2 are channel constants, and ϵ denotes the required bit error rate (BER) of the channel. We consider that UAV i moves in low attitude for data collection, where the probability of LoS communication between UAV i and ground sensor j can be

$$Pr_{LoS}(\varphi_j^i) = \frac{1}{1 + aexp(-b[\varphi_j^i - a])} \tag{2}$$

where a and b are constants, and φ_j^i denotes the elevation angle between UAV i and ground sensor j. Furthermore, path loss of the channel between UAV i and device j can be obtained by

$$\gamma_j^i = Pr_{LoS}(\varphi_j^i)(\eta_{LoS} - \eta_{NLoS}) + 20log(r\sec(\varphi_j^i)) + 20log(\lambda) + 20log\left(\frac{4\pi}{v_c}\right) + \eta_{NLoS} \tag{3}$$

where r denotes the radius of the radio coverage of UAV i, λ is the carrier frequency, and v_c is the speed of light. η_{LoS} and η_{NLoS} represent the excessive path losses of LoS or non-LoS, respectively [32]. Please See Appendix A.

A. Communication Protocol

Fig. 2 shows the data collection protocol for the MA-WSN. Specifically, the proposed MADRL-SA operates onboard at the UAVs to determine their velocities and sensor selection, and allocate the modulation scheme for the selected sensors. The details of MADRL-SA will be provided in the next section. Next, the UAV broadcasts a short beacon message which contains the ID of the selected sensor. Upon the receipt of the beacon message, the selected sensor transmits its data packets to the UAV, along with the state information of $e_j(t)$, $q_j(t)$, and TVR_p

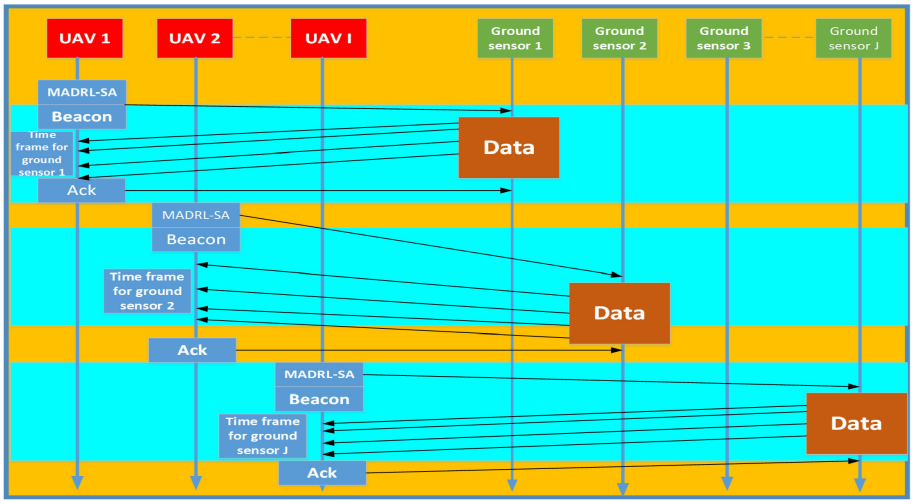


Fig. 2. Data communication protocol for the MA-WSN. In each communication frame, MADRL-SA is conducted at the UAVs to determine the velocity and sensor selection while allocating the modulation scheme for the selected sensor.

in the control segment of the data packet. Once the data is correctly received, the UAV sends an acknowledgment to the ground sensor.

IV. Problem Formulation

In this section, we present the problem formulation.

A. Optimization Formulation

Let $\kappa_j^i(t)$ be the binary indicator of ground sensor j being selected by UAV i for data transmission at time t. If ground sensor j is scheduled by UAV i at time t, $\kappa_j^i(t) = 1$; otherwise, $\kappa_j^i(t) = 0$. The joint optimization of UAV velocity and communication schedule aims to minimize the packet loss of all the ground sensors, as given by

Optimization problem:

$$\min_{\kappa_j^i(t), v_i(t), P_j^i(t)} \Sigma_{i=1}^{I}\Sigma_{j=1}^{J} f_{ij}(\kappa_j^i(t), v_i(t), P_j^i(t)) + \Sigma_{j=1}^{J} g_j(\kappa_j^i(t))) \quad (4)$$

subject to:

$$0 \le P_j^i(t)\kappa_j^i(t) \le P_{max}, \quad (5)$$

where

$$f_{ij}(\kappa_j^i(t), v_i(t), P_j^i(t)) = \begin{cases} 1, & \text{if } (\kappa_j^i(t)=1) \text{ \& } (h_j^i(t)<=h_{th}) \text{ \& } (v_i(t)<=v_{max}); \\ 0, & \text{otherwise,} \end{cases}$$

and

$$g_j(\kappa_j^i(t)) = \begin{cases} 1, & \text{if } (q_j(t) > D) \text{ \& } (\kappa_j^i(t) = 0); \\ 0, & \text{otherwise,} \end{cases}$$

Constraint (5) ensures that the transmit power of the scheduled ground sensor does not exceed the maximum transmit power P_{max}.

B. MMDP Formulation

MMDP can be defined by the tuple $\{I,\ \{S_{\alpha,i}\},\ \{a_i\},\ \mathrm{C}\{S_\beta|S_\alpha, a\},\ \Pr\{S_\beta|S_\alpha, a\}\}$

1) I is the number of agents, i.e., UAVs.
2) $S_{\alpha,i}$ is the network state observed by agent i ($i \in I$). $S_{\alpha,i}$ comprises: channel quality $h_j^i(t)$, battery level $e_j(t)$, queue length $q_j(t)$, visit time TVR_p and the location of UAV $\zeta_i(t)$,
i.e., $S_{\alpha,i} = \{(h_j^i(t), e_j(t), q_j(t), TVR_p, \zeta_i(t)), \mathrm{i}=1,2,...\mathrm{I}\}$. In particular, each ground sensor maintains a list of visiting time of the agents. Joint state of all the agents is denoted S_α, where $S_\alpha = S_{\alpha,1} \times \ldots\ldots \times S_{\alpha,I}$.
3) a_i represents the action of agent i. a_i is to schedule one sensor to transmit data to the UAV, determine the modulation and the instantaneous patrol velocity of the UAV, i.e., $a_i = \{(j, \phi_j(t), v(t)), i = 1, 2, \ldots I\}$. Joint action a which consists of the actions of all the agents is $a = a_1 \times \ldots\ldots \times a_I$. The size of action space is $J\Phi \mid v(t) \mid$, where Φ is the highest modulation order and $\mid v(t) \mid$ stands for the cardinality of the set $[v_{min}, v_{max}]$.
4) $\mathrm{C}\{S_\beta|S_\alpha, a\}$ is the network cost yielded when joint action a is taken at joint state S_α and the following joint state changes to S_β. The network cost is the packet loss of the ground sensors.
5) $\Pr\{S_\beta|S_\alpha, a\}$ denotes the transition probability from joint state S_α to joint state S_β when joint action a is taken.

C. Transition Probability

The transition probability of the MMDP, from S_α to S_β can be given by

$$\Pr\{S_\beta|S_\alpha\} = \Pi_{i=1}^{I}(\Pr\{(e_{\beta,j}, q_{\beta,j}, h_{\beta,j}, \zeta_{\beta,j}) |(e_{\alpha,j}, q_{\alpha,j}, h_{\alpha,j}, \zeta_{\alpha,j}), j \in a_i\}_i \times \Pi_{k=1}^{K} \Pr\{(e_{\beta,k}, q_{\beta,k}, h_{\beta,k}, \zeta_{\beta,k}) |(e_{\alpha,k}, q_{\alpha,k}, h_{\alpha,k}, \zeta_{\alpha,k}, k \neq a_i; i \in [1, I]\}_i) \tag{6}$$

Specifically, the state transition probability presented in (6) consists of two parts. The first part, i.e., $\Pr\{(e_{\beta,j}, q_{\beta,j}, h_{\beta,j}, \zeta_{\beta,j})|(e_{\alpha,j}, q_{\alpha,j}, h_{\alpha,j}, \zeta_{\alpha,j}), j \in a_i\}$is the state transition probability from S_α to S_β in terms of the selected ground sensor ($j \in a_i$). Let K denote the total number of unselected ground sensors. The second part, i.e.,

$\Pi_{k=1}^{K} \Pr\{(e_{\beta,k}, q_{\beta,k}, h_{\beta,k}, \zeta_{\beta,k})|(e_{\alpha,k}, q_{\alpha,k}, h_{\alpha,k}, \zeta_{\alpha,k}, k \neq a_i; i \in [1, I]\}$ is the probability from S_α to S_β in terms of the unselected ground sensors, where $k \neq a_i; i \in [1, I]$ indicates the sensors that are not selected by any of the I agents.

Let $d_{i,j}$ denotes the distance between ground sensor *j* and UAV *i*, $v(t)$ is velocity of the UAV, $R(t)$ is the data rate of the ground sensor and λ is the packet arrival probability. The state transition probability of the selected sensor j, which is specified in (5), depends on the following possible transitions.

1) Packet transmission is successful due to the good channel quality, i.e., $h_{\beta,j} > h_{\alpha,j}$ and low velocity. There is no packet arrival, the data queue of the selected node decreases, i.e., $q_{\beta,j} = q_{\alpha,j} - 1$. The state transition probability is $(1-\epsilon)^{\frac{2d_{i,j}R(t)}{v(t)}}(1-\lambda)$.
2) Packet transmission is failed due to the poor channel quality, i.e., $h_{\beta,j} < h_{\alpha,j}$ and high velocity. A new data packet is generated and buffered, the data queue of the selected node increases, i.e., $q_{\beta,j} = q_{\alpha,j} + 1$. The state transition probability is $(1-(1-\epsilon)^{\frac{2d_{i,j}R(t)}{v(t)}})\lambda$.
3) Packet transmission is successful due to the good channel quality, i.e., $h_{\beta,j} > h_{\alpha,j}$ and low velocity. A new data packet is generated and buffered, the data queue of the selected node remains unchanged, i.e., $q_{\beta,j} = q_{\alpha,j}$. The state transition probability is $(1-\epsilon)^{\frac{2d_{i,j}R(t)}{v(t)}}\lambda$.
4) Packet transmission is failed due to the poor channel quality, i.e., $h_{\beta,j} < h_{\alpha,j}$ and high velocity. There is no packet arrival, the data queue of the selected node remains unchanged, i.e., $q_{\beta,j} = q_{\alpha,j}$. The state transition probability is $(1-(1-\epsilon)^{\frac{2d_{i,j}R(t)}{v(t)}})(1-\lambda)$.

Due to the packet transmission, the battery level of the selected sensor decreases by Δe. See Eq. (7) and (8) shown at the bottom of this page.

(8) corresponds to the unselected sensors with two different cases. The first case corresponds to the case when queue of the ground sensor increases, i.e., $q_{\beta,k} = q_{\alpha,k} + 1$ due to a new packet arrival, i.e., λ. The second case gives that the data queue remains unchanged, i.e., $q_{\beta,k} = q_{\alpha,k}$ since there is no packet arrival, i.e., $(1-\lambda)$.

By solving the formulated MDP, e.g., by using dynamic programming techniques, the optimal solution with complete states could be achieved, which could be used for performance benchmarking in multi-UAV-assisted wireless sensor networks. Unfortunately, dynamic programming (and the MDP formulation) suffers from the well-known curse-of-dimensionality, and incurs a prohibitive complexity and intractability, which can be noted in Appendix B. Please See Appendix B.

V. MULTI-UAV PERSPECTIVE

A. Preliminaries

Reinforcement Learning (RL) is a major branch of machine learning, where an agent learns to behave in an environment by performing actions and observing the associated results [33]. RL can be applied to solve MMDPs with unknown transition probabilities. In an RL process, an agent observes its current state, takes an action, and receives its immediate cost together with its new state. The observed information, i.e., the immediate cost and new state, is used to adjust the agent's policy. This process repeats until the agent's policy approaches the optimal policy [34]. Q-learning [35] is the most popular RL paradigm which can be used to calculate the Q-functions and decide the optimal policy. It gives an agent the ability to act optimally in an MMDP setup. Agents implementing Q-learning update their Q-values according to the following update rule

$$\Pr\{(e_{\beta,j}, q_{\beta,j}, h_{\beta,j}, \zeta_{\beta,j})|(e_{\alpha,j}, q_{\alpha,j}, h_{\alpha,j}, \zeta_{\alpha,j}), j \in a_i\} = \begin{cases} (1-\epsilon)^{\frac{2d_{i,j}R(t)}{v(t)}}(1-\lambda) & \text{if } e_{\beta,j} = e_{\alpha,j}\text{-}\Delta e \text{ and } q_{\beta,j} = q_{\alpha,j} - 1 \text{and } h_{\beta,j} > h_{\alpha,j} \\ (1-(1-\epsilon)^{\frac{2d_{i,j}R(t)}{v(t)}})\lambda & \text{if } e_{\beta,j} = e_{\alpha,j}\text{-}\Delta e \text{ and } q_{\beta,j} = q_{\alpha,j} + 1 \text{and } h_{\beta,j} < h_{\alpha,j} \\ (1-\epsilon)^{\frac{2d_{i,j}R(t)}{v(t)}}\lambda & \text{if } e_{\beta,j} = e_{\alpha,j}\text{-}\Delta e \text{ and } q_{\beta,j} = q_{\alpha,j} \text{ and} h_{\beta,j} > h_{\alpha,j} \\ (1-(1-\epsilon)^{\frac{2d_{i,j}R(t)}{v(t)}})(1-\lambda) & \text{if } e_{\beta,j} = e_{\alpha,j}\text{-}\Delta e \text{ and } q_{\beta,j} = q_{\alpha,j} \text{ and} h_{\beta,j} < h_{\alpha,j} \end{cases} \tag{7}$$

$$\Pr\{(e_{\beta,k}, q_{\beta,k}, h_{\beta,k}, \zeta_{\beta,k})|(e_{\alpha,k}, q_{\alpha,k}, h_{\alpha,k}, \zeta_{\alpha,k}, k \neq a_i; i \in [1, I]\} = \begin{cases} \lambda & \text{if } e_{\beta,k} = e_{\alpha,k} \text{ and } q_{\beta,k} = q_{\alpha,k} + 1 \\ 1-\lambda & \text{if } e_{\beta,k} = e_{\alpha,k} \text{ and } q_{\beta,k} = q_{\alpha,k} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

$$Q_i(S_{\beta,i}|S_{\alpha,i},a_i) = (1-\nu)Q_i(S_{\beta,i}|S_{\alpha,i},a_i)+ \nu(C(S_{\beta,i}|S_{\alpha,i},a_i) + \gamma \min_{a'_i} Q_i(S_{\beta',i}|S_{\beta,i},a'_i) \quad (9)$$

where $\nu \in (0,1]$ is the learning rate, $S_{\beta',i}$ is the next state, and a'_i is the next action. The trajectory of each UAV can consist of a large number of waypoints. The velocity control of the UAVs along the trajectories can result in a very large state and action space. In this case, Q-learning suffers from the well-known curse of dimensionality [36]. To circumvent this impasse, we propose a new deep Q-network to optimize the velocity control online by approximating the optimal action-value function.

B. Multi-Uav Dqn

In practice, the UAVs have a common goal and able to perform actions. Meanwhile, the environment is unknown for the UAVs, and they receive outdated knowledge of the network states. Therefore, statistical methodologies can not be applied due to lack of real-time knowledge and the size of the state space. RL could be the best paradigm in such situation given uncertain environment, action and goal. In the joint problem of communication scheduling and velocity control with I UAVs, choice of actions by one UAV impact those of other UAVs. Each UAV interacts with an unknown environment to learn a policy. In learning process, the current environment state is partially observed by UAV i then by following its policy takes an action, this action is dependent on the past actions of other UAVs on the scheduled ground sensor a_u^{t-1}, and consequently obtains the cost and the new environment state. Then, UAV i utilizes the gathered data to optimize its policy. UAV i interacts with the environment, performs the action and optimizes its policy for many iterations to converge to the optimal policy. In multi-UAV setting, each UAV is designed to find an optimal policy π_i for minimizing the long-term expected accumulated discounted costs. The actions of the UAVs i.e., velocities control, modulation allocation and ground sensor selection are optimized to minimize the packet losses resulting from buffer overflows and failed transmissions of the sensors. The actions are defined as a tuple $a =< \ldots >$ which consists of the optimization variables for all the UAVs. The action space in the MDP contains all the UAVs' decisions. The decisions of velocity control and sensor selection are independently made by the UAVs. The action of each UAV not only determines its future state, but also influences the actions of the other UAVs. Therefore, a formulation of multi-agent MDP and multi-agent DQN optimizes the actions of multiple decision makers, i.e., UAVs.

From the perspective of UAV i, the accumulated cost by executing action a_i dependent on a_u^{t-1} at the current environment state s on the basis of policy π_i can be represented by

$$Q_i^{\pi_i}(S_{\alpha,i},a_i,a_u^{t-1}) = E[\Sigma_{t=0}^{\infty}\gamma^t C(S_{\beta,i}|S_{\alpha,i},a_i,a_u^{t-1})] \quad (10)$$

where $\gamma \in [0, 1]$ is the discount factor. Each UAV aims to learn the optimal Q-value or the optimal policy. The Q-value for agent i is updated as follow:

$$Q_i(S_{\beta,i}|S_{\alpha,i},a_i,a_u^{t-1}) = (1-\nu)Q_i(S_{\beta,i}|S_{\alpha,i},a_i,a_u^{t-1}) + \nu(C(S_{\beta,i}|S_{\alpha,i},a_i,a_u^{t-1}) + \gamma \min_{a'_i} Q_i(S_{\beta',i}|S_{\beta,i},a'_i,a'_u) \quad (11)$$

where $\nu \in (0,1]$ is the learning rate, $S_{\beta',i}$ is the next state and a'_i the next action. Multi-UAV Q-learning cannot deal with the exponential growth of states and actions for the resource allocation problem in the MA-WSN. This is known as the curse of dimensionality. Multi-UAV DQN circumvent the curse-of-dimensionality of the problem. It represents the action-value function of each agent with a deep neural network parameterized by θ^{Q_i}. For each UAV, θ^{Q_i} is learned by sampling transition from the replay memory and minimizing the squared temporal difference error:

$$\Gamma(\theta^{Q_i}) = y_i - Q_i\{S_{\beta,i} \mid S_{\alpha,i},a_i,a_u^{t-1};\theta^{Q_i}\}. \quad (12)$$

where y_i is the target Q-value which is set as a label and can be denoted by

$$y_i = C\{S_{\beta,i} \mid S_{\alpha,i},a_i,a_u^{t-1}\}+ \gamma \min_{a'_i} Q'_i\{S_{\beta,i'} \mid S_{\beta,i},a'_i,a'_u;\theta^{Q'_i}\} \quad (13)$$

Multi-UAV DQN use target network and experience replay for each UAV to guarantee stability. In multi-UAV DQN, experience replay is used to remove correlations in the observation sequence and smoothing over changes in the data distribution by randomizing over the states and the actions of MMDP at each time-step *t*. The provided multi-UAV DQN formulation is effective and promising for computing multi-UAV policies, in contrast to the traditional approaches for solving MMDP, it does not fail to deal with enormous size and complexity.

C. Proposed MADRL-SA

We present a multi-UAV version of DQN called MADRL-SA, MADRL-SA realizes cooperation between UAVs, by enabling them to learn the scheduling decisions of each other.

According to Fig. 3, MADRL-SA has three UAVs, and each UAV is equipped with a classical DQN algorithm and learns through interaction by environment. As can be seen in Fig. 3, UAV 3 performs its action and schedules a ground sensor, then receives its visiting record and consequently calculates the time differences $\delta[]$ between its visiting time(t) and TVR_p. $\delta[]$ is augmented to state and utilized in the learning process. Therefore, each UAV learns to coordinate its action. The UAVs that visited the same ground sensor would learn to improve their scheduling process based on computed timing information. For example, if the computed time differences are large the UAV is encouraged to schedule the ground sensor for the next time. Overall, our goal is to allow different UAVs schedule different ground sensors (other ground sensors may have buffer overflow probability) and if a ground sensor recently visited by an UAV no other UAV visits that ground sensor. The proposed scheme is described in Algorithm 1, which optimizes the actions based on the multi-UAV DQN to solve the online resource allocation problem.

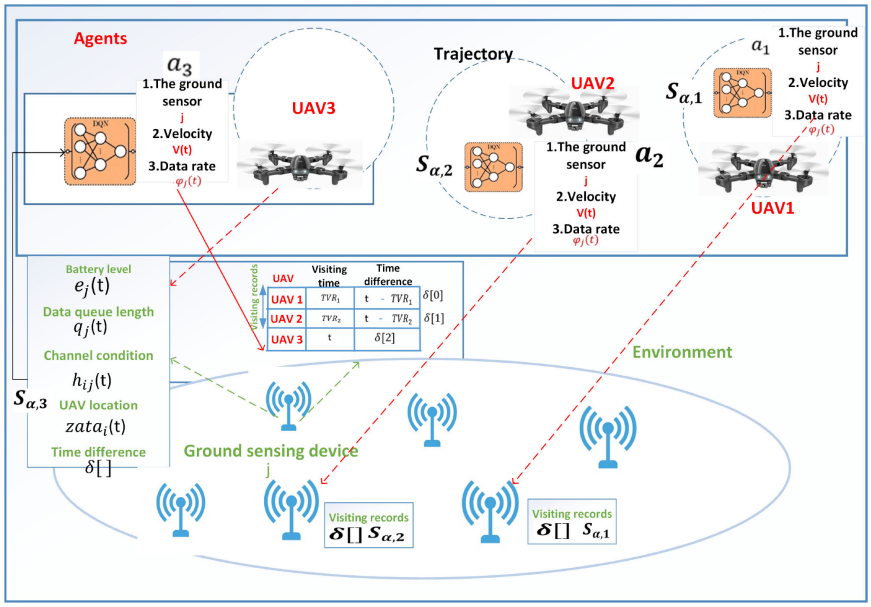


Fig. 3. An overview of MADRL-SA, where the UAVs observe the current environment state by following their policy take actions.

Overall, two separate Q-networks are maintained with each UAV, Q-network: $Q_i\{S_{\beta,i} \mid S_{\alpha,i}, a_i, a_u^{t-1}; \theta^{Q_i}\}$ and target network: $Q_i'\{S_{\beta,i'} \mid S_{\beta,i}, a_i', a_u'; \theta^{Q_i'}\}$, with weights θ^{Q_i} and $\theta^{Q_i'}$ respectively. At first step, Q-network and associated target of each UAV are initialized and then learning is ignited. Each UAV samples its state and computes its local state $S_{\alpha,i}$ including $\delta[]$. Each UAV receives the local state $S_{\alpha,i}$ and selects a random action with probability ε or exploits its knowledge and produce its action. Each UAV executes the selected action and computes the vector of δ using t and TVR_p; then corresponding cost and next state including $\delta[]$ are sampled. Then the associated transition $(S_{\alpha,i}, S_{\beta,i}, a_i, C)$ is stored. θ^{Q_i} is learned by sampling batches of transitions from the replay memory and minimizing the squared temporal difference error:

$$\Gamma(\theta^{Q_i}) = y_i - Q_i\{S_{\beta,b} \mid S_{\alpha,b}, a_b, a_{ub}; \theta^{Q_i}\} \tag{14}$$

where

$$y_i = C\{S_{\beta,b} \mid S_{\alpha,b}, a_b, a_{ub}\} + \gamma \min_{a_b'} Q_i'\{S_{\beta,b'} \mid S_{\beta,b}, a_b', a_{ub}'; \theta^{Q_i'}\} \tag{15}$$

finally for each agent the parameters of a Q-network θ^{Q_i} copied into those of target network $\theta^{Q_i'}$ after a constant number of iterations. The proposed MADRL-SA can be readily repurposed to support different objective functions. For example, it can be potentially repurposed to maximize the energy efficiency, which is the ratio of network throughput to the energy consumption.

D. Energy and Feasibility

UAVs are becoming increasingly less restrictive in terms of energy due to new advancements of battery and energy harvesting technologies. For example, Atlantik Solar has developed an autonomous, solar-powered drone (UAV) that can fly up to 10 days continuously. A ground sensor can be equipped with solar panels, wind power generators or other energy harvesting mechanisms to harvest renewable energy from ambient resources and recharge its battery.

The UAVs select the optimal sensors to transmit data and allocate their modulation schemes, by learning the states of the ground sensors. The selected sensor uses the allocated modulation to transmit data to the UAV, while updating the visiting time of the UAV. In particular, the historical record of the visiting time typically has a small size. Consider 100 UAVs, the size of the historical record at the sensor is just seven bits. The time for updating the record is negligible. Also, the sensors only need to synchronize with the UAVs the recent historical record of visits. The overhead is small. Therefore, the proposed deep reinforcement learning based data collection requires a small amount of computation at the sensors, which is feasible and practical in real-world Multi-UAV-Assisted WSNs.

VI. Numerical Results and Discussions

In this section, we first investigate complexity of MADRL-SA, then we present network configurations and performance metrics. Next, we evaluate the network cost of the proposed MADRL-SA scheme with regard to the network size and varying number of UAVs. Here, the network cost defines the amount of packet loss due to the data queue overflow and the failed transmission from the ground sensor to the UAV.

A. Complexity of MADRL-SA

The time complexity for training each network Q_i that has Z layers with z_i neurons per layer is given by

$$\mathcal{O}(MT \times (\Sigma_{i=1}^{Z-1} z_i z_{i+1})) \tag{16}$$

where M is the number of episodes and T is the number of iterations. Therefore, the time complexity of MADRL-SA with

Algorithm 1: MADRL-SA.

1.Initialize:

Randomly initialize the networks
$Q_i\{S_{\beta,i} \mid S_{\alpha,i}, a_i, a_u^{t-1}; \theta^{Q_i}\}$ with θ^{Q_i}
Initialize target networks Q_i' with weights $\theta^{Q_i'} = \theta^{Q_i}$
$\forall i \in (1, I)$

2.Learning:

for *episode=1 to M* **do**
  Obtain state $S_{\alpha,i}$
  for *t=1 to T* **do**
    if(Probability ε)
    Select a random action a_i
    else
    $a_i = argmin_{a_i} Q_i\{S_{\beta,i} \mid S_{\alpha,i}, a_i, a_u^{t-1}; \theta^{Q_i}\}$
    end
    Execute action a_i in the environment
    Receive the visiting record
    for *p=1 to I* **do**
      if(i==p)
      $\delta[p]$=t
      else
      $\delta[p]$=t - TVR_p
      end
    end
    Obtain the cost function $C_{t,i} = \{S_{\beta,i} \mid S_{\alpha,i}, a_i, a_u^{t-1}\}$ and the next state $S_{\beta,i}$ at $t+1$
    Store Transition $(S_{\alpha,i}, S_{\beta,i}, a_i, C_{t,i})$
    Sample random minibatch $(S_{\alpha,b}, S_{\beta,b}, a_b, C_{t,b})$
    $y_i = C\{S_{\beta,b} \mid S_{\alpha,b}, a_b, a_{ub}\} + \gamma min_{a_b'} Q_i'\{S_{\beta,b'} \mid S_{\beta,b}, a_b', a_{ub}'; \theta^{Q_i'}\}$
    Derive the loss function
    $\Gamma(\theta_i^Q) = y_i - Q_i\{S_{\beta,b} \mid S_{\alpha,b}, a_b, a_{ub}; \theta^{Q_i}\}$
    Update the target networks.
    $\theta^{Q_i'} = \theta^{Q_i}$

    $S_\alpha = S_\beta$
  end
end

I networks of Q_i is given by

$$\mathcal{O}(I \times MT \times (\Sigma_{i=1}^{Z-1} z_i z_{i+1})) \tag{17}$$

The case of an equal number of neurons in each layer, the time complexity can be written as

$$\mathcal{O}(MT \times (I \times MT \times Z - 1 \times z^2) = \mathcal{O}(I \times MT \times z^2) \tag{18}$$

B. Implementation of MADRL-SA

J number of ground sensors are randomly deployed, where J increases from 20 to 120. Each ground sensor has the maximum discretized battery capacity 50 Joules, the highest modulation = 5, and the maximum transmit power 100 milliwatts. For calculating $P_j^i(t)$ of the ground sensor, the two channel constants, k_1 and k_2 are set to 0.2 and 3, respectively. The required BER is 0.05, and the carrier frequency is 2000 MHz. ε is set to 0.05. However, the value of ε can be configured based 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 II. Moreover, the region of interest is set to be a square area with a size of 1000 x 1000 meters, where the ground sensors are distributed in the targeted region. MADRL-SA is implemented in Python 3.5 using Pytorch (the Python deep learning library). A Lenovo Workstation running 64-bit Ubuntu 16.04 LTS, with Intel Core i5-7200 U CPU @ 2.50 GHz 4 and 8 G memory is used for the PyTorch setup. Deep reinforcement learning trains MADRL-SA for 1000 episodes. The discount factor and learning rate are set to 0.99 and 0.001, respectively. We use 2-layer fully connected neural network for each agent, which includes 400 and 300 neurons in the first and second layers, respectively. We utilize the rectified linear unit (ReLU) function for the activation function. The experience replay memory with the size of 10^6 is created for each agent to store the learning outcomes in the format of a quadruplet <state, action, cost, next state>. The memory is updated by calling the function replay buffer$_i$.add((state, action, cost, next state)), and retrieves the experiences by using replay buffer$_i$.sample(batch size).

For performance evaluation, the proposed MADRL-SA is compared with Random scheduling policy (RSA), Channel scheduling policy (CHSA) and DRL-SA [14] algorithms.

- RSA randomly determines the velocities of the UAVs at each waypoint, and one of the ground sensors within the communication range of the UAV is randomly selected to

TABLE I
NOTATION AND DEFINITION

Notation	Definition
J	number of ground sensors
I	number of UAVs
a_u^{t-1}	past actions of other UAVs on a ground sensor
a_i	action of UAV i
$S_{\alpha,i}$	state of UAV i
$S_{\beta,i}$	next state of UAV i
$P_j^i(t)$	transmit power between device j and UAV i
$h_j^i(t)$	channel gain between device j and UAV i
$\zeta_i(t)$	location of the UAV on its trajectory
$v(t)$	velocity of the UAV
v_{max}, v_{min}	the maximum and minimum velocity of the UAV
$e_j(t)$	battery level of device j
$q_j(t)$	queue length of device j
TVR_p	Time of each visiting record
D	maximum queue length of ground sensor
$\phi_j(t)$	modulation scheme of device j
γ	discount factor for future states
θ	learning weight in deep Q-network

TABLE II
PYTORCH CONFIGURATION

Parameters	Values
Number of ground sensors	20-120
Queue length	40
Energy levels	50
Discount factor	0.99
Learning rate	0.001
Replay memory size	10^6
Batch size	100
Number of episodes	1000

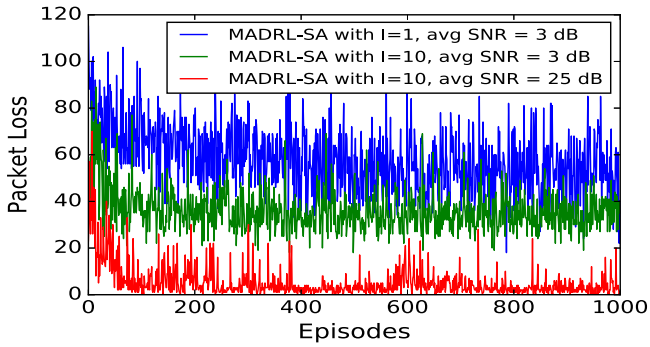


Fig. 4. The network cost at each episode of two versions of MADRL-SA with I = 10 and DRL-SA.

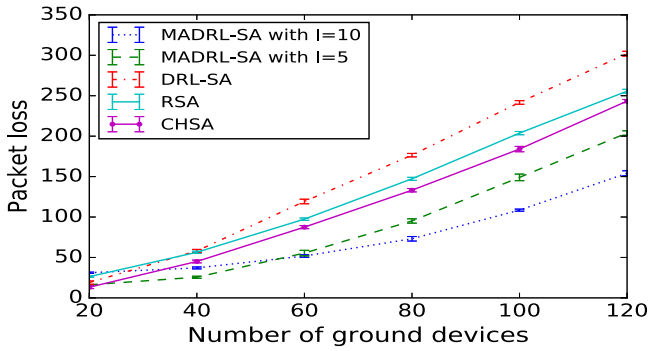


Fig. 5. Comparison of packet loss by MADRL-SA and the baselines in term of ground sensors.

transmit data. The velocity control and sensor selection are independent of the batteries, data queue lengths of the ground sensors, channel variation, and UAVs' positions.

- CHSA allows the UAVs to move with the minimum velocity and schedule the ground sensors based on their channel quality. Each UAV sends beacons along the trajectory. Based on the sensors' replies to the beacons, the UAV measures the channel gains. The ground sensor with the highest channel gain is selected to transmit.
- DRL-SA enables a single-agent DQN, where each UAV leverages DQN to learn the optimal velocity control and sensor selection strategy based on the data queue length, energy level, channel variation and UAV's positions. The selection of the ground sensor, modulation scheme, and velocity of the UAV is jointly optimized (independently of the rest of the UAVs).

C. Performance Evaluation

Fig. 4 depicts the convergence of MADRL-SA with I = 10 for low and high SNR cases and DRL-SA. MADRL-SA with I = 10 and high SNR show the best performance since it reduce the overflow cost as well as the fading cost due to good SNR. MADRL-SA with I = 10 and low SNR outperform the DRL-SA which has the highest network cost. The reason is that when multiple UAVs act it results in the reduction of overflow cost.

Fig. 5 depicts the network cost of MADRL-SA (data queue length = 40) and the baselines in term of ground sensors. MADRL-SA with I = 5 and I = 10 achieves a lower network

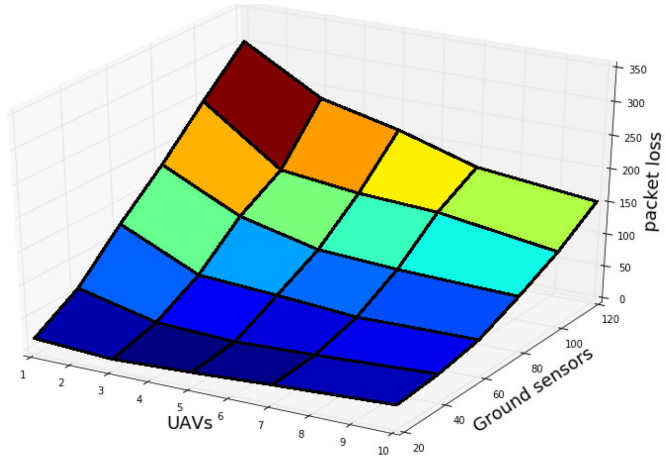


Fig. 6. Trade-off between the number of UAVs and ground sensors.

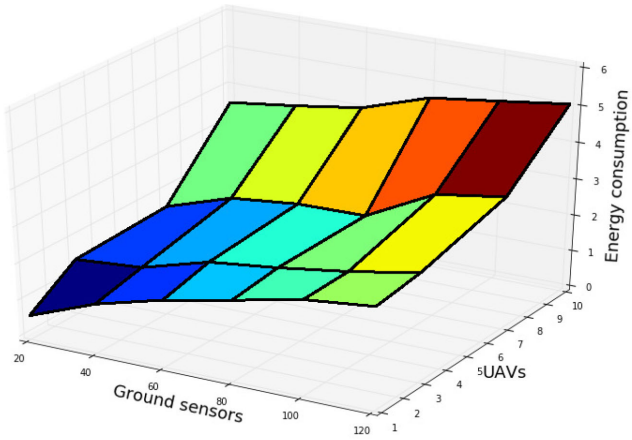


Fig. 7. Energy consumption of ground sensors.

cost in comparison to CHSA. The network cost of MADRL-SA with I = 5 is lower than that of CHSA. Overall, MADRL-SA with I = 5 and I = 10 outperforms CHSA. Particularly, when J = 100 the packet loss of MADRL-SA with I = 5 and I = 10 is lower than CHSA by around 21% and 40%, respectively.

Fig. 6 shows the trade-off between the number of ground sensors and UAVs. Specifically, a large number of ground sensors expedites the buffer overflows in MA-WSN and in turn, increases the packet loss. On the other hand, increasing the number of UAVs allows the ground sensors to be scheduled in parallel, hence reducing the buffer overflow. A balance needs to be struck between the numbers of UAVs and ground sensors to minimize the packet loss.

Fig. 7 shows the energy consumption of the ground sensors by varying the number of ground sensors and UAVs. For a given number of UAVs, the energy consumption of the network increases with the number of ground sensors. On the other hand, the increasing number of UAVs helps increase the number of ground sensors scheduled to transmit data, hence raising the energy consumption of the ground sensor network.

Fig. 8 show the velocities and trajectories of different UAVs for the MADRL-SA with I = 7. Fig. 8(a) demonstrates the velocity of 7 UAVs given 20 waypoints. The color bar shows

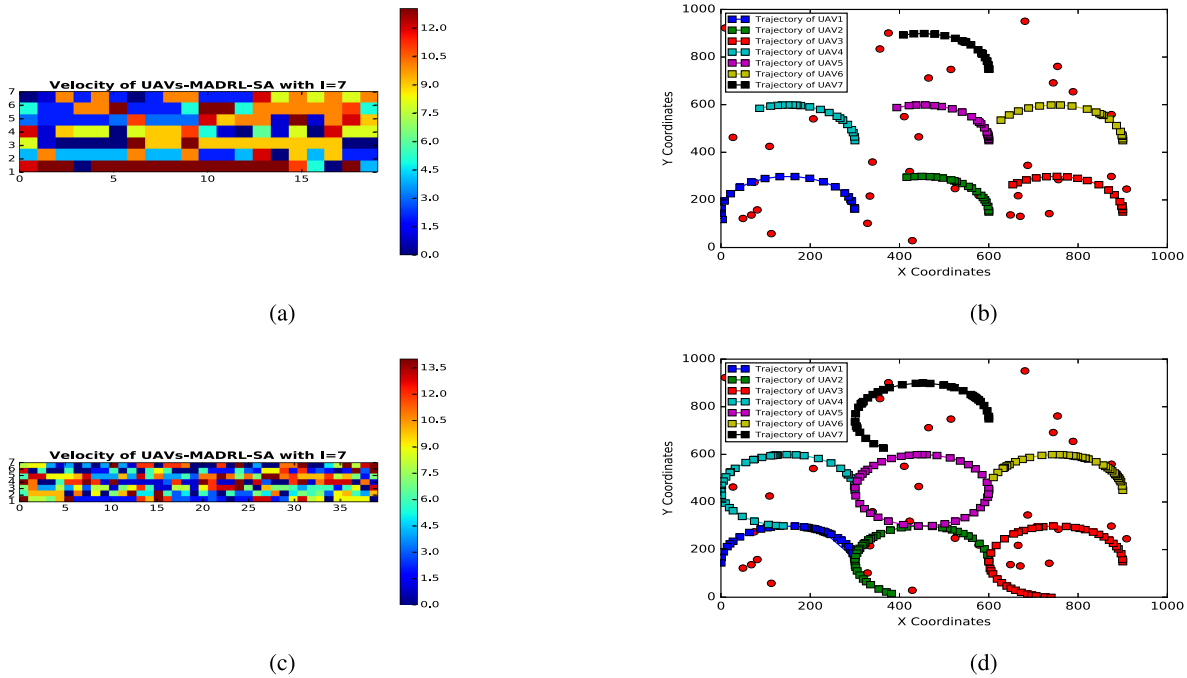


Fig. 8. Velocities and trajectories of MADRL-SA with I = 7. (a) and (b)velocity and trajectory given number of waypoints = 20. (c) and (d) velocity and trajectory given number of waypoints=40

the range of values for velocity and color map shows the actual velocity of each UAV for each waypoint in color format. As can be seen UAV 2 moves with the lowest velocity as confirmed by its small trajectory in Fig. 8(b). In contrast, UAV 1 moves with the highest velocity as confirmed by its trajectory. Overall, for waypoints 1-12, UAV 3-7 move with the lowest velocity witnessing subtle changes. After these waypoints the velocity of these UAVs is increasing.

Fig. 8(c) is similar to Fig. 8(a) except that number of waypoints is increased to 40. Overall, the pattern for all UAVs except UAV 5 is almost similar and all of them move with low or moderate velocity witnessing high velocity at some points, this can be confirmed by their associated trajectories in Fig. 8(d). UAV 5 moves smoothly before waypoint 20. After this point its velocity start increasing and hence a full trajectory is shaped as can be seen in Fig. 8(d).

Fig. 9 evaluates the network cost with the increasing number of UAVs, where the buffer size of MADRL-SA is set to 20 or 40 and the number of ground sensors is 40. For MADRL-SA with buffer size of 40, increasing the number of UAVs from 3 to 10 leads to a reduction of the packet loss by 68%. In contrast, when the buffer size is 20, a reduction of 77% in the packet loss is witnessed. Fig. 9 also shows that MADRL-SA significantly outperforms RSA by 80% when the buffer size is 40, and by 34% when the buffer size is 20.

Fig. 10 demonstrates the training performance with varied learning rates(lr). After few episodes in the beginning, the network cost have an obvious tendency to decrease and converge in the case of lr = 1e-3 and lr = 5e-4. Nevertheless, the algorithm may converge to a local optimum in case of large learning rate, this situation can be seen in the case of lr = 1e-1 and lr = 1e-2.

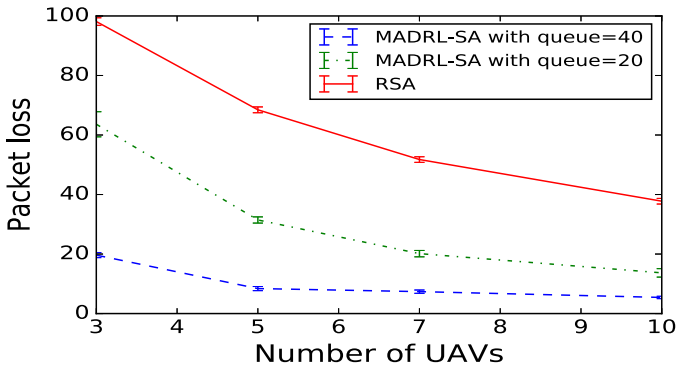


Fig. 9. The network cost with an increasing number of UAVs, where the data queue length of MADRL-SA is set to 20 and 40 and number of ground sensors is 40.

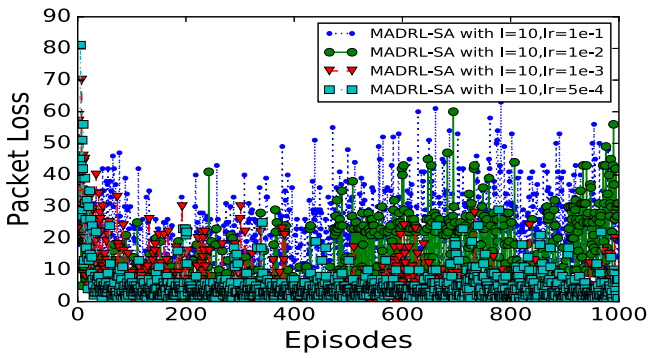


Fig. 10. The training performance with varied learning rates.

VII. Conclusion

In this paper, we study the joint flight cruise control and data collection scheduling in the MA-WSN. We formulate the problem using MMDP to minimize the packet loss due to buffer overflows at the ground sensors and fading airborne channels. We propose MADRL-SA to solve the formulated MMDP, where all UAVs utilize DQN to conduct respective decisions. In MADRL-SA, the UAVs acting as agents learn the underlying patterns of the data and energy arrivals at all the ground sensors as well as the scheduling decisions of the other UAVs. We conduct simulation using PyTorch deep learning library and results reveal that the proposed MADRL-SA for MA-WSN reduces packet loss by up to 54% and 46%, as compared to the single agent case and existing non-learning greedy algorithm, respectively. The joint online optimization of the trajectories, speed control, and communication schedules will be our future work, where we may consider other learning techniques to capture continuity and smoothness of the trajectories and address issues, such as collision avoidance.

Appendix A

The path loss of the LoS link is given by

$$PL_{LOS} = 20\log d + 20\log f + 20\log\left(\frac{4\pi}{c}\right) + \eta_{LOS} \quad (19)$$

The path loss of the non-LoS link is given by

$$PL_{NLOS} = 20\log d + 20\log f + 20\log\left(\frac{4\pi}{c}\right) + \eta_{NLOS} \quad (20)$$

The LoS probability is given by

$$Pr_{LOS} = \frac{1}{1 + aexp(-b[\varphi_j^i - a])} \quad (21)$$

Then, the NLoS probability is

$$Pr_{NLOS} = 1 - Pr_{LOS} \quad (22)$$

The expectation of the path loss between UAV i and device j can be obtained by

$$\gamma_j^i = Pr_{LOS} \times PL_{LOS} + Pr_{NLOS} \times PL_{NLOS} \quad (23)$$

By substituting (22) into (23), we have

$$\gamma_j^i = Pr_{LOS}(PL_{LOS} - PL_{NLOS}) + PL_{NLOS} \quad (24)$$

Substituting (19), (20), (21) into (24) leads to

$$\gamma_j^i = \frac{(\eta_{LOS} - \eta_{NLOS})}{1 + aexp(-b[\varphi_j^i - a])} + 20\log d + 20\log f + 20\log\left(\frac{4\pi}{c}\right) + \eta_{NLOS} \quad (25)$$

Rewriting 25 in term of φ_j^i and r, we finally obtain

$$\gamma_j^i = \frac{(\eta_{LOS} - \eta_{NLOS})}{1 + aexp(-b[\varphi_j^i - a])} + 20log(r\sec(\varphi_j^i)) + 20\log(\lambda) + 20\log\left(\frac{4\pi}{c}\right) + \eta_{NLOS} \quad (26)$$

Appendix B

Let ϵ denote the bit error rate, L denote the data packet length and λ denote the packet arrival probability. Depending on the transmission status and arrival pattern, four transitions may happen as presented in (7):

1) In the first case, the packet transmission is successful $(1-\epsilon)^L$ and there is no packet arrival $(1-\lambda)$. The probability of such transition is $(1-\epsilon)^L \times (1-\lambda)$. Given L=R(t)*T where T is the conversation time of UAV i and ground sensor j, and $T = \frac{2d_{i,j}}{v(t)}$. We have $L = \frac{2d_{i,j}R(t)}{v(t)}$ by substituting T into L. Therefore, the transition probability of the first case is $(1-\epsilon)^{\frac{2d_{i,j}R(t)}{v(t)}}(1-\lambda)$.
2) In the second case, the packet transmission is not successful $(1-(1-\epsilon)^L)$ and there is packet arrival λ. The probability of such transition is $(1-(1-\epsilon)^L) \times \lambda$. By substituting T into L, we have $L = \frac{2d_{i,j}R(t)}{v(t)}$. Therefore, the transition probability of the second case is $(1-(1-\epsilon)^{\frac{2d_{i,j}R(t)}{v(t)}})\lambda$.
3) In the third case, the packet transmission is successful $(1-\epsilon)^L$ and there is packet arrival λ. The probability of such transition is $(1-\epsilon)^L \times \lambda$. By substituting T into L, we have $L = \frac{2d_{i,j}R(t)}{v(t)}$. Therefore, the transition probability of the third case is $(1-\epsilon)^{\frac{2d_{i,j}R(t)}{v(t)}}\lambda$.
4) In the fourth case, the packet transmission is not successful $(1-(1-\epsilon)^L)$ and there is no packet arrival $(1-\lambda)$. The probability of such transition is $(1-(1-\epsilon)^L) \times (1-\lambda)$. We have $L = \frac{2d_{i,j}R(t)}{v(t)}$. Therefore, the transition probability of the fourth case is $(1-(1-\epsilon)^{\frac{2d_{i,j}R(t)}{v(t)}})(1-\lambda)$.

(8) investigates the transmission probabilities for unselected ground sensors. These ground sensors do not transmit data. In this case, the ground sensors either receive packet with transition probability λ or no packet is received with transition probability $1-\lambda$.

References

[1] H. Shakhatreh *et al.*, "Unmanned aerial vehicles (UAVs): A survey on civil applications and key research challenges," *IEEE Access*, vol. 7, pp. 48572–48634, 2019.

[2] J. Kim, S. Kim, C. Ju, and H. I. Son, "Unmanned aerial vehicles in agriculture: A review of perspective of platform, control, and applications," *IEEE Access*, vol. 7, pp. 105 100–105 115, 2019.

[3] N. Zhao *et al.*, "UAV-assisted emergency networks in disasters," *IEEE Wireless Commun.*, vol. 26, no. 1, pp. 45–51, Feb. 2019.

[4] H. Shakhatreh *et al.*, "Unmanned aerial vehicles (UAVs): A survey on civil applications and key research challenges," *IEEE Access*, vol. 7, pp. 48572–48634, 2019.

[5] Y. Gao, X. Chen, J. Yuan, Y. Li, and H. Cao, "A data collection system for environmental events based on unmanned aerial vehicle and wireless sensor networks," in *Proc. IEEE 4th Inform. Technol., Netw., Electron. Automat. Control Conf.*, 2020, pp. 2175–2178.

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

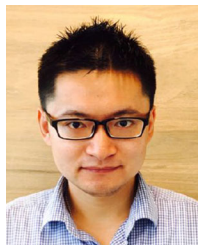
[7] M. Thammawichai, S. P. Baliyarasimhuni, E. C. Kerrigan, and J. B. Sousa, "Optimizing communication and computation for multi-UAV information gathering applications," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 54, no. 2, pp. 601–615, Apr. 2018.

[8] Q. Chen, "Joint position and resource optimization for multi-UAV-aided relaying systems," *IEEE Access*, vol. 8, pp. 10403–10415, 2020.

[9] V. Sharma, I. You, and R. Kumar, "Energy efficient data dissemination in multi-UAV coordinated wireless sensor networks," *Mobile Inform. Syst.*, Hindawi, vol. 2016, 2016.
[10] A. T. Albu-Salih and S. A. H. Seno, "Energy-efficient data gathering framework-based clustering via multiple UAVs in deadline-based wsn applications," *IEEE Access*, vol. 6, pp. 72 275–72 286, 2018.
[11] C. Zhan and Y. Zeng, "Completion time minimization for multi-UAV-enabled data collection," *IEEE Trans. Wireless Commun.*, vol. 18, no. 10, pp. 4859–4872, Oct. 2019.
[12] K. Li, W. Ni, X. Wang, R. P. Liu, S. S. Kanhere, and S. Jha, "Energy-efficient cooperative relaying for unmanned aerial vehicles," *IEEE Trans. Mobile Comput.*, vol. 15, no. 6, pp. 1377–1386, Jun. 2016.
[13] M. Samir, S. Sharafeddine, C. M. Assi, T. M. Nguyen, and A. Ghrayeb, "UAV trajectory planning for data collection from time-constrained IoT devices," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 34–46, Jan. 2020.
[14] K. Li, W. Ni, E. Tovar, and A. Jamalipour, "On-board deep q-network for UAV-assisted online power transfer and data collection," *IEEE Trans. Veh. Technol.*, vol. 68, no. 12, pp. 12 215–12 226, Dec. 2019.
[15] K. Li, W. Ni, E. Tovard, and A. Jamalipour, "Online velocity control and data capture of drones for the Internet of Things: An onboard deep reinforcement learning approach," *IEEE Veh. Technol. Mag.*, vol. 16, no. 1, pp. 49–56, Mar. 2021.
[16] K. Li, W. Ni, E. Tovar, and M. Guizani, "Joint flight cruise control and data collection in UAV-aided Internet of Things: An onboard deep reinforcement learning approach," *IEEE Internet Things J.*, vol. 8, no. 12, pp. 9787–9799, Jun. 2021.
[17] Y. Y. Munaye, R. T. Juang, H. P. Lin, and G. B. Tarekegn, "Resource allocation for multi-UAV assisted IoT networks: A deep reinforcement learning approach," in *Proc. Int. Conf. Pervasive Artif. Intell.*, 2020, pp. 15–22.
[18] J. Tang, J. Song, J. Ou, J. Luo, X. Zhang, and K.-K. Wong, "Minimum throughput maximization for multi-UAV enabled WPCN: A deep reinforcement learning method," *IEEE Access*, vol. 8, pp. 9124–9132, 2020.
[19] B. Zhang, C. H. Liu, J. Tang, Z. Xu, J. Ma, and W. Wang, "Learning-based energy-efficient data collection by unmanned vehicles in smart cities," *IEEE Trans. Ind. Informat.*, vol. 14, no. 4, pp. 1666–1676, Apr. 2018.
[20] 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 J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 2059–2070, Sep. 2018.
[21] Q. Wang, W. Zhang, Y. Liu, and Y. Liu, "Multi-UAV dynamic wireless networking with deep reinforcement learning," *IEEE Commun. Lett.*, vol. 23, no. 12, pp. 2243–2246, Dec. 2019.
[22] J. Cui, Y. Liu, and A. Nallanathan, "Multi-agent reinforcement learning-based resource allocation for UAV networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 2, pp. 729–743, Feb. 2020.
[23] A. Shamsoshoara, M. Khaledi, F. Afghah, A. Razi, and J. Ashdown, "Distributed cooperative spectrum sharing in UAV networks using multi-agent reinforcement learning," in *Proc. 16th IEEE Annu. Consum. Commun. Netw. Conf.*, 2019, pp. 1–6.
[24] U. Challita, W. Saad, and C. Bettstetter, "Deep reinforcement learning for interference-aware path planning of cellular-connected UAVs," in *Proc. IEEE Int. Conf. Commun.*, 2018, pp. 1–7.
[25] X. Liu, Y. Liu, Y. Chen, and L. Hanzo, "Trajectory design and power control for multi-UAV assisted wireless networks: A machine learning approach," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 7957–7969, Aug. 2019.
[26] D. H. Choi, S. H. Kim, and D. K. Sung, "Energy-efficient maneuvering and communication of a single UAV-based relay," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 50, no. 3, pp. 2320–2327, Jul. 2014.
[27] Y. Zeng and R. Zhang, "Energy-efficient UAV communication with trajectory optimization," *IEEE Trans. Wireless Commun.*, vol. 16, no. 6, pp. 3747–3760, Jun. 2017.
[28] N. Zhao, Z. Liu, and Y. Cheng, "Multi-agent deep reinforcement learning for trajectory design and power allocation in multi-UAV networks," *IEEE Access*, vol. 8, pp. 139 670–139 679, 2020.
[29] J. Hu, H. Zhang, L. Song, R. Schober, and H. V. Poor, "Cooperative internet of UAVs: Distributed trajectory design by multi-agent deep reinforcement learning," *IEEE Trans. Commun.*, vol. 68, no. 11, pp. 6807–6821, Nov. 2020.
[30] L. Wang, K. Wang, C. Pan, W. Xu, N. Aslam, and L. Hanzo, "Multi-agent deep reinforcement learning-based trajectory planning for multi-UAV assisted mobile edge computing," *IEEE Trans. Cogn. Commun. Netw.*, vol. 7, no. 1, pp. 73–84, Mar. 2021.
[31] K. Li, Y. Emami, W. Ni, E. Tovar, and Z. Han, "Onboard deep deterministic policy gradients for online flight resource allocation of UAVs," *IEEE Netw. Lett.*, vol. 2, no. 3, pp. 106–110, Sep. 2020.
[32] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal LAP altitude for maximum coverage," *IEEE Wireless Commun. Lett.*, vol. 3, no. 6, pp. 569–572, Dec. 2014.
[33] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.
[34] N. C. Luong *et al.*, "Applications of deep reinforcement learning in communications and networking: A survey," *IEEE Commun. Surv. Tut.*, vol. 21, no. 4, pp. 3133–3174, 2019.
[35] W. Qiang and Z. Zhongli, "Reinforcement learning model, algorithms and its application," in *Proc. Int. Conf. Mechatronic Sci., Electric Eng. Comput.*, 2011, pp. 1143–1146.
[36] A. Gosavi, "Reinforcement learning: A tutorial survey and recent advances," *INFORMS J. Comput.*, vol. 21, no. 2, pp. 178–192, Apr. 2009.

Yousef Emami (Student Member, IEEE) received the B.Sc. degree in computer engineering from the Islamic Azad University of Dezful, Dezful, Iran, in 2007, and the M.Sc. degree in computer networks from the Shiraz University of Technology, Shiraz, Iran, in 2015. He is currently working toward the Ph.D. degree with the University of Porto and Researcher with Real-Time and Embedded Computing Systems Research Centre (CISTER). His main research interests include wireless communications, UAV networks, reinforcement learning, and game theory.

Bo Wei received the Ph.D. degree in computer science and engineering from the University of New South Wales, Australia, in 2015. He has been a Senior Lecturer with the Department of Computer and Information Sciences, Northumbria University. He was a Lecturer and then a Senior Lecturer in computer science with Teesside University. Before joining Teesside, he was a Postdoctoral Research Assistant with University of Oxford. His research interests include mobile computing, Internet of Things, and wireless sensor networks.



Kai Li (Senior Member, IEEE) 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. He is currently a Senior Research Scientist with Real-Time and Embedded Computing Systems Research Centre (CISTER), Portugal. He is also a CMU-Portugal Research Fellow, which is jointly supported by Carnegie Mellon University, U.S., and The Foundation for Science and Technology (FCT), Portugal. Prior to this, Dr. Li was a Postdoctoral Research Fellow with The SUTD-MIT International Design Centre, The Singapore University of Technology and Design, Singapore (2014–2016). He was a Visiting Research Assistant with ICT Centre, CSIRO, Australia (2012–2013). From 2010 to 2011, he was a Research Assistant with Mobile Technologies Centre with The Chinese University of Hong Kong. His research interests include machine learning, vehicular communications and security, resource allocation optimization, cyber-physical systems, Internet of Things (IoT), and UAV networks. Dr. Li has been serving as the Associate Editor for *Elsevier Ad Hoc Networks Journal* and IEEE ACCESS JOURNAL, and the Demo Co-Chair for ACM/IEEE IPSN 2018.

Wei Ni (Senior Member, IEEE) received the B.E. and Ph.D. degrees in electronic engineering from Fudan University, Shanghai, China, in 2000 and 2005, respectively. He is currently a Group Leader and Principal Research Scientist with CSIRO, Sydney, Australia, and an Adjunct Professor with the University of Technology Sydney and an Honorary Professor with Macquarie University, Sydney. Prior to this, he was a Postdoctoral Research Fellow with Shanghai Jiaotong University from 2005–2008, Deputy Project Manager with the Bell Labs, Alcatel/Alcatel-Lucent from 2005–2008, and Senior Researcher with 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 the Editor of IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS since 2018. He was 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 (Member, IEEE) was born in 1967. He received the Licentiate, M.Sc. and Ph.D. degrees in electrical and computer engineering from the University of Porto, Porto, Portugal, in 1990, 1995, and 1999, respectively. He is currently a Professor with the Computer Engineering Department, 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. Since 1991, he authored or coauthored 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. 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 five years, until December 2015, member of the Executive Committee of the IEEE Technical Committee on Real-Time Systems (TC-RTS). Dr. 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).