Wireless Power and Energy Harvesting Control in IoD by Deep Reinforcement Learning

Jingjing Yao, *Student Member, IEEE*, and Nirwan Ansari, *Fellow, IEEE*

***Abstract*—Internet of drones (IoD), which deploys several drones in the air to collect ground information and send them to the IoD gateway for further processing, can be applied in traffic surveillance and disaster rescue. The performance of IoD is greatly affected by drones' battery capacities. We hence utilize the energy harvesting technology to charge the batteries and the wireless power control to adjust the drone wireless transmission power in order to address this challenge. In our work, we investigate the joint optimization of power control and energy harvesting control to determine each drone's transmission power and the transmitted energy from the charging station in time-varying IoD networks. Our objective is to minimize the long-term average system energy cost constrained by the drones' battery capacities and quality of service (QoS) requirements. A Markov Decision Process (MDP) is formulated to characterize the power and energy harvesting control process in time-varying IoD networks. A modified actor-critic reinforcement learning algorithm is then proposed to tackle our problem and its performance is demonstrated via extensive simulations.**

***Index Terms*—Internet of drones (IoD), power control, energy scheduling, energy harvesting, deep reinforcement learning, actor-critic, quality of service (QoS).**

I. Introduction

Internet of things (IoT) interconnects billions of smart devices to exchange network information and provisions various applications such as smart industry, city, transportation, and healthcare [1]. Drones, also known as unmanned aerial vehicles (UAVs), are used for a growing number of purposes such as environment monitoring, disaster investigation and surveillance [2], [3]. Internet of drones (IoD) utilizes drones as the IoT devices to provision applications such as object tracking and disaster rescue [4]. One of the fundamental IoD applications is the sensing service (i.e., data collection service) where several drones are deployed in the air to collect the environmental information (e.g., temperature, air pollutant index, and ground pictures). The collected data are then sent to the IoD gateway for further processing.

IoD has been utilized for a growing number of applications. However, its performance is greatly limited by the drone's battery capacity. Many IoD applications are deployed in hard-to-reach or hazardous areas, and so it is impractical to replace batteries. To address this challenge, energy-efficient IoD communication system should be designed to reduce the drone's energy consumption. Adjusting the drone's wireless transmission power can help reduce the energy consumption of transmitting the collected IoD data [5]. It is thus important to investigate the wireless power control.

Energy harvesting can also be a good alternative to prolong the lifetime of drone batteries [6]. Energy harvesting may use the ambient energy sources, e.g., solar and wind, to harvest energy. However, the ambient sources may not guarantee the quality of service (QoS) requirements (e.g., minimum data transmission time) because they are random and uncertain. Hence, controllable energy sources, e.g., radio frequency (RF) signals from a power station, can be considered to supply energy on demand [7]. We hence utilize a charging station to charge drone batteries to help maintain drones' operations, where the radio signals are sent from the charging station to carry energy in the form of electromagnetic radiation. Then, the energy harvesting device of each drone converts the radio signals to its battery energy [8]. The harvested energy depends on the transmitted energy of the charging station and the path loss between drones and the charging station [9]. Hence, energy harvesting control, which determines the transmitted energy from the charging station, is important to be investigated.

The dynamic and time-varying IoD networks also pose a great challenge of wireless power control and energy harvesting control. The varying network states (e.g., collected data, battery level, and QoS requirement) at different time epochs require different power control and energy harvesting control policies to achieve the optimum performance. In order to characterize the time-varying IoD network, a Markov decision process (MDP) can be utilized [10]. It is usually difficult to obtain the complete and accurate information of the MDP model in the unknown and dynamic IoD networks. We hence design a deep reinforcement learning algorithm to address the MDP model, i.e., the sequential decision-making problem in time-varying IoD networks [11]. Specifically, the reinforcement learning is a learning process of trials and errors that interacts with the network environment by observing network states (i.e., collected data and battery levels) and then taking actions (i.e., determining the wireless power control and energy harvesting control policies). Deep reinforcement learning uses the deep neural networks (DNNs) to approximate the state-action values in measuring the possible system cost brought by each state-action pair [11].

Motivated by the above analysis, we investigate the wireless power control and energy harvesting control in time-varying IoD networks for the sensing service. Specifically, we try to optimize each drone's wireless transmission power and the transmitted energy from the charging station to each drone at

J. Yao and N. Ansari are with the Advanced Networking Laboratory, Helen and John C. Hartmann Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA (e-mail: jy363@njit.edu; nirwan.ansari@njit.edu). This work was supported in part by the U.S. National Science Foundation under Grant No. CNS-1814748.

each time epoch with the objective to minimize the long-term average system energy cost constrained by the drone battery capacities and the QoS requirements. We further adopt an MDP to model this problem and propose a deep reinforcement learning algorithm to solve it.

The rest of this paper is organized as follows. Section II summarizes the related works. Our energy harvesting aided IoD architecture is described in Section III. Then, the joint optimization of the wireless data transmission power control and energy harvesting control is formulated in Section IV. In Section V, we propose a deep reinforcement learning algorithm to address the joint optimization problem. The simulation results are presented and analyzed in Section VI. Finally, Section VII concludes this paper.

II. Related Works

The IoD network was proposed in [4], in which a conceptual model of the IoD system's organization, the features, and the implementation are described in detail. The authors also demonstrated that the IoD network can be applied for package delivery, disaster rescue, and traffic surveillance. Wazid *et al.* [12] proposed a lightweight user authentication scheme in an IoD network for the users to access data from drones and demonstrated that their scheme provides better security than existing schemes. Bera *et al.* [13] proposed a blockchain based secure framework for data management in IoD networks, which provides better security and also incurs less communication and computation overheads. Yao and Ansari [14] designed an online algorithm to address the joint optimization of task allocation and flying speed control in an IoD network to minimize the drone's journey completion time during which a drone generates computing tasks, offloads them to a fog node, and visits different locations of interest.

Energy harvesting is a prominent technology to charge batteries. Altinel *et al.* [15] proposed Markov energy model to analyze the energy outage, shortage and service loss probabilities of energy harvesting aided communication systems. Nguyen *et al.* [16] designed an energy-harvesting-aware routing protocol for IoT networks to improve the lifetime of IoT devices under variable traffic load and energy availability conditions. Yao and Ansari [17] proposed a Stackelberg game in cached-enabled energy-harvesting-aided IoT networks to incentivize the charging station to transmit energy to the IoT devices. Jawad *et al.* [18] utilized the magnetic resonant coupling (MRC) technology for the wireless power transfer to charge the drone batteries. They demonstrated that the battery life of the drone was extended from 30 to 851 minutes.

The above works do not consider wireless power control in reducing the energy consumption of IoT/IoD networks. Yao and Ansari [19] jointly optimized the power control and fog resource provisioning in fog-aided IoT networks to minimize the system cost while guaranteeing QoS requirements. Lee and Hong [20] proposed a power control scheme for secure device-to-device communication in IoT networks to improve system energy efficiency. Mach and Becvar [21] proposed a distributed power control algorithm to increase the delivery ratio of computation results constrained by the QoS requirements in mobile edge networks. Yao and Ansari [5] investigated the power control in IoD networks for the data collection service to minimize the drone's energy consumption while satisfying the QoS requirement. However, none of the above works consider the joint optimization of power control and energy harvesting control in IoD networks. Challita *et al.* [22] proposed a deep reinforcement learning algorithm to optimize the transmission power, path, and cell association of each UAV to minimize the interference level and the wireless transmission delay in multi-UAV-aided networks. Pace *et al.* [23] proposed a cognitive transmission power control scheme in IoT networks by a muli-agent Q-learning algorithm where each IoT sensor learns its own power control policy.

Deep reinforcement learning has been utilized in time-varying IoT/IoD networks to improve the performance of network strategies [24]. Lei *et al.* [25] proposed a joint computation offloading and multiuser scheduling problem in IoT edge system to minimize the average weighted sum of delay and power consumption. They further designed a deep reinforcement learning algorithm to solve this joint optimization problem. Yao and Ansari [26] investigated the content placement problem in time-varying cache-enabled IoT networks to minimize the data transmission delay constrained by the cache storage capacity and IoT data freshness. Liu *et al.* [27] proposed a data collection and secure sharing scheme by combining Ethereum blockchain and deep reinforcement learning to create a reliable and safe IoT environment. However, none of the above works consider utilizing deep reinforcement learning to solve the power control in energy harvesting aided IoD networks.

Our preliminary results of wireless power control in energy harvesting aided IoD networks by deep reinforcement learning was presented at ICC2020 [28]. We extend our preliminary work by additionally consider the energy harvesting control (i.e., determining the amount of transmitted energy to each drone) to further reduce our system energy cost. In this work, we investigate the joint optimization of wireless power control and energy harvesting control in time-varying IoD networks to minimize the long-term average system cost constrained by the drone battery capacities and QoS requirements. A deep reinforcement learning algorithm is proposed to solve this joint optimization problem.

III. System Model

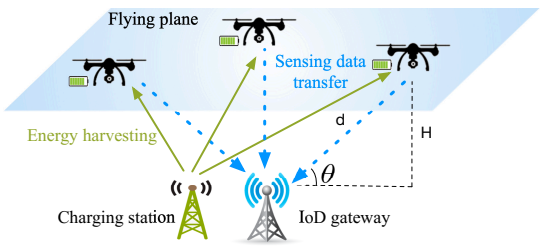

Fig. 1. Data collection in energy harvesting aided IoD.

Consider our system model with N drones hovering in the flying plane at the height of H, as shown in Fig. 1. We denote

the set of drone indexes as $\mathcal{N} = \{1, 2, ..., N\}$. The drones sense the environmental data (e.g., pictures and videos) at different locations and send them to the IoD gateway. The IoD gateway can further process the sensed data and send them to a monitor or users who request these data. Owing to the limited drone battery capacities, a charging station is utilized to charge the drone batteries in order to support their operations. Specifically, the drone battery can harvest energy by converting the received radio frequency (RF) signals from the charging station to power [29]. The charging station can use the license-free frequency bands (e.g., 915 *MHz* [30] and 5 *GHz* [31]) for energy transfer and provide controllable energy.

We assume the network operates at discrete time epochs and the network states remain static within a time epoch but vary over different ones [32]. At each time epoch, the IoD gateway determines each drone's wireless transmission power to transmit its sensed data and the transmitted energy from the charging station to each drone. In our work, we characterize the QoS requirement as the minimum data transmission time [19].

A. Drone Data Transmission Delay

The drone's data transmission rate depends on the wireless channel between the drone and the IoD gateway. In our work, we adopt the widely used probability model which assumes that the signal between the drone the IoD gateway is either Line-of-Sight (LoS) with probability Pr^{LoS} or Non-Line-of-Sight (NLoS) with probability Pr^{NLoS} [33]. The probabilities Pr^{LoS} and Pr^{NLoS} can be calculated by

$$Pr^{LoS} = \frac{1}{1 + \alpha \exp(-\beta[\frac{180}{\pi} \arcsin(\frac{H}{d}) - \alpha])}, \quad (1)$$

and

$$Pr^{NLoS} = 1 - Pr(LoS), \quad (2)$$

respectively. α and β are environment-related (e.g., rural and urban) constants [33], H is the drone's flying height, and d is the distance between the drone and the IoD gateway, as shown in Fig. 1. The path losses for LoS signals PL^{LoS} and NLoS signals PL^{NLoS} are calculated by [33]

$$PL^{LoS} = 20 \log_{10}(\frac{4\pi f_c d}{c}) + \xi^{LoS}, \quad (3)$$

and

$$PL^{NLoS} = 20 \log_{10}(\frac{4\pi f_c d}{c}) + \xi^{NLoS}, \quad (4)$$

where f_c is the carrier frequency, c is the speed of light, and ξ^{LoS} and ξ^{NLoS} are environment-related constants. The path loss between the drone and IoD gateway is then modeled as the average path loss between the LoS and NLoS signal, and is calculated by

$$PL = Pr^{LoS} PL^{LoS} + Pr^{NLoS} PL^{NLoS}. \quad (5)$$

The wireless channel gain between drone i and IoD gateway G_i^{BS} is a function of the path loss, i.e.,

$$G_i^{BS} = 10^{-\frac{PL_i}{10}}, \quad (6)$$

where PL_i is the path loss between drone i and the IoD gateway. Therefore, drone i's wireless transmission rate r_i can be calculated by the Shannon's formula

$$r_i = W_i \log_2(1 + \frac{p_i G_i^{BS}}{N_0 W_i}), \quad (7)$$

where G_i^{BS} is the wireless channel gain between drone i and the IoD gateway, p_i is drone i's wireless transmission power, W_i is the system bandwidth allocated to drone i and N_0 is the noise power spectrum density. Hence, drone i's wireless transmission time of sending the sensed data to the IoD gateway is

$$\tau_i = \frac{l_i}{r_i} = \frac{l_i}{W_i \log_2(1 + \frac{p_i G_i^{BS}}{N_0 W_i})}, \quad (8)$$

where l_i is the data size of drone i's sensed data.

B. Drone's Energy Consumption

Drone i's energy consumption for transmitting the sensing data can be expressed as [34]

$$E_i^{trs} = p_i \tau_i = \frac{p_i l_i}{W_i \log_2(1 + \frac{p_i G_i^{BS}}{N_0 W_i})}, \quad (9)$$

where τ_i is drone i's wireless data transmission time which is defined in Eq. (8).

We assume RF energy harvesting technology is used to charge the drone batteries, and the amount of the harvested energy depends on the transmitted energy from the charging station and the wireless channel gain between the charging station and the drone. Hence, we utilize the widely used linear energy harvesting model [9] to calculate the harvested energy E_i^{hrv}, i.e.,

$$E_i^{hrv} = \eta_i G_i^{EH} e_i, \quad (10)$$

where η_i is drone i's energy harvesting efficiency, G_i^{EH} is the wireless channel gain between drone i and the charging station and can be similarly calculated by Eq. (6), and e_i is the transmitted energy from the charging station to drone i.

In our work, all drone batteries are rechargeable, and the charged energy can be stored in the battery for future use [35]. We denote the system battery level vector at time epoch t as

$$\boldsymbol{b}(t) = [b_1(t), b_2(t), ..., b_N(t)], \quad (11)$$

where $b_i(t) \in [0, B_i^{max}], i \in \mathcal{N}$ is drone i's battery level at time epoch t and is bounded between 0 and the battery capacity B_i^{max}. Hence, drone i's battery level evolves from time epoch t to time epoch $t+1$ by

$$b_i(t+1) = \min\{b_i(t) + E_i^{hrv}(t) - E_i^{trs}(t), B_i^{max}\}, \quad (12)$$

where $b_i(t+1) \geq 0$, i.e.,

$$b_i(t) + E_i^{hrv}(t) - E_i^{trs}(t) \geq 0, \quad (13)$$

which is equivalent to

$$b_i(t) + \eta_i G_i^{EH} e_i(t) - \frac{p_i(t) l_i(t)}{W_i \log_2(1 + \frac{p_i(t) G_i^{BS}}{N_0 W_i})} \geq 0. \quad (14)$$

We assume the system energy cost comes from both drone energy consumption and charging station energy consumption, and can be calculated by

$$E_{sys}(t) = c_1 \sum_{i=1}^{N} E_i^{trs}(t) + c_2 \sum_{i=1}^{N} e_i(t), \tag{15}$$

where c_1 and c_2 is the energy cost per joule of drone's battery and charging station, respectively [7]. $e_i(t)$ is the transmitted energy from the charging station to drone i in time epoch t.

IV. Problem Formulation

In this section, we formulate the wireless power control and the harvested energy control problem for sensing service in IoD networks, where drones are deployed to sense the environmental information. In order to build an energy efficient system, our objective is to minimize the long-term average system energy cost. Then, the problem can be formulated as

$$\textbf{P0:} \min_{p_i(t), e_i(t)} \frac{1}{M} \sum_{t=1}^{M} [c_1 \sum_{i=1}^{N} E_i^{trs}(t) + c_2 \sum_{i=1}^{N} e_i(t)] \tag{16}$$

$$s.t. \quad p_i(t) \le P_i^m, \ \forall i \in \mathcal{N}, t \in \mathcal{M}, \tag{17}$$

$$\frac{l_i}{W_i \log_2(1 + \frac{p_i G_i^{BS}}{N_0 W_i})} \le T_i^{th}, \ \forall i \in \mathcal{N}, t \in \mathcal{M}, \tag{18}$$

$$b_i(1) = B_i^{max}, \forall i \in \mathcal{N}, \tag{19}$$

$$b_i(t+1) = \min\{b_i(t) + \eta_i G_i^{EH} e_i(t) - \frac{p_i(t) l_i(t)}{W_i \log_2(1 + \frac{p_i(t) G_i^{BS}}{N_0 W_i})}, B_i^{max}\}, \ \forall i \in \mathcal{N}, t \in \mathcal{M}, \tag{20}$$

$$b_i(t) + \eta_i G_i^{EH} e_i(t) - \frac{p_i(t) l_i(t)}{W_i \log_2(1 + \frac{p_i(t) G_i^{BS}}{N_0 W_i})} \ge 0, \quad \forall i \in \mathcal{N}, t \in \mathcal{M}. \tag{21}$$

In Eq. (16), $M \in \{1, 2, ..., \infty\}$ denotes the total number of time epochs and the objective is to minimize the average system energy cost from time epoch 1 to time epoch M. For simplicity, we define $\mathcal{M}$ as the set $\{1, 2, ..., M\}$ to denote time epochs from 1 to M. Eq. (17) imposes drone i's wireless transmission power to be less than the maximum transmission power P_i^m. Eq. (18) is the QoS requirement which imposes drone i's wireless data transmission time to be less than the threshold T_i^{th}. Eq. (19) imposes drone i's initial battery level to be B_i^{max}. Eq. (20) denotes the drone battery level evolution. Eq. (21) indicates the feasibility of each drone's battery level. Although the energy consumed for drone's air hovering also accounts for the drone energy consumption [36], it is related to the drone's physical properties (e.g., weight and propellers) and hence is fixed at each equal-length time epoch [37]. The hovering energy consumption is affected by neither the wireless power control nor the energy harvesting control strategies. The energy cost generated by the drone hovering is hence a constant and can be ignored in the objective function which minimizes the average system energy cost. Therefore, we do not include the hovering energy consumption and only focus on the energy consumption for wireless transmission.

Lemma 1. *Constraint (20) is equivalent to*

$$b_i(t+1) = b_i(t) + \eta_i G_i^{EH} e_i(t) - \frac{p_i(t) l_i(t)}{W_i \log_2(1 + \frac{p_i(t) G_i^{BS}}{N_0 W_i})}, \tag{22}$$

and

$$b_i(t) + \eta_i G_i^{EH} e_i(t) - \frac{p_i(t) l_i(t)}{W_i \log_2(1 + \frac{p_i(t) G_i^{BS}}{N_0 W_i})} \le B_i^{max}, \quad \forall i \in \mathcal{N}, t \in \mathcal{M}. \tag{23}$$

Proof: We use the proof of contradiction to demonstrate this lemma.

Assume that solution $\langle p_i^*(t), e_i^*(t) \rangle$ achieves the minimum system energy cost

$$\phi^* = \frac{1}{M} \sum_{t=1}^{M} [c_1 \sum_{i=1}^{N} E_i^{trs}(t) + c_2 \sum_{i=1}^{N} e_i^*(t)] \tag{24}$$

while satisfying that $b_i(t+1) = b_i(t) + \eta_i G_i^{EH} e_i^*(t) - \frac{p_i^*(t) l_i(t)}{W_i \log_2(1 + \frac{p_i^*(t) G_i^{BS}}{N_0 W_i})} > B_i^{max}$. We can always find another $\langle p_i^*(t), \tilde{e}_i(t) \rangle$, where $\tilde{e}_i(t) < e_i^*(t)$, that satisfies $b_i(t+1) = b_i(t) + \eta_i G_i^{EH} \tilde{e}_i(t) - \frac{p_i^*(t) l_i(t)}{W_i \log_2(1 + \frac{p_i^*(t) G_i^{BS}}{N_0 W_i})} \le B_i^{max}$ and achieves the system energy cost

$$\tilde{\phi} = \frac{1}{M} \sum_{t=1}^{M} [c_1 \sum_{i=1}^{N} E_i^{trs}(t) + c_2 \sum_{i=1}^{N} \tilde{e}_i(t)]. \tag{25}$$

Since $\tilde{e}_i(t) < e_i^*(t)$, it can be observed that $\tilde{\phi} < \phi^*$ which violates the assumption that ϕ^* is the optimum solution to minimize the system energy cost. Hence, the lemma is proved. ■

It is challenging to obtain the global optimum solution of Problem **P0** because of its non-convexity [38]. Additionally, drones' battery levels are coupled with each other over different time epochs and the complete battery level information of all time epochs are required in order to achieve the optimum; this may not be practical in reality. Note that problem **P0** can be considered as a sequential decision-making problem (i.e., wireless transmission power and harvested energy) in a time-varying IoD environment. To solve the time-varying decision-making problem, we first utilize a Markov decision process (MDP) to model the time-varying decision-making problem [10], and then solve the MDP model by a deep reinforcement learning algorithm [39] in the following section.

V. Deep Reinforcement Learning

To obtain the solution of problem **P0**, which is a sequential decision-making problem in a time-varying IoD environment, an MDP is utilized to model problem **P0**. We then describe our proposed Power and Energy hArvesting control deep Reinforcement Learning (PEARL) algorithm, which is a modified actor-critic deep reinforcement learning algorithm to solve the MDP model.

A. MDP model

We use a MDP $\langle \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{C} \rangle$ to model the power and energy harvesting control in a time-varying IoD network, which consists of the network state space $\mathcal{S}$, action space $\mathcal{A}$, state transition probability density functions $\mathcal{F} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto [0, \infty)$, and cost functions $\mathcal{C} : \mathcal{S} \times \mathcal{A} \mapsto [0, \infty)$. Specifically, at each time epoch t, the IoD gateway (acting as the network controller) observes the network state $s(t)$ and takes an action $a(t)$. The network system then generates a system cost $c(t)$ according to the action, and transits to the next network state $s(t+1)$.

1) Network state: We define the network state $s(t)$ at time epoch t as a set of drones' sensed data sizes and battery levels

$$s(t) = [l_1(t), l_2(t), ..., l_N(t), b_1(t), b_2(t), ..., b_N(t)], \quad (26)$$

where $l_i(t)$ and $b_i(t)$ are drone i's sensed data size and battery level, respectively. Hence, the network state space $\mathcal{S}$ can be defined as

$$\mathcal{S}(t) = \{s(t) \mid l_i(t) \geq 0, 0 \leq b_i(t) \leq B_i^{max}, i \in \mathcal{N}\}. \quad (27)$$

2) Action: The action of the network system $a(t)$ at time epoch t determines $p_i(t)$ (drone i's power control strategy) and $e_i(t)$ (the transmitted energy from the charging station to drone i). Hence, $a(t)$ can be defined as

$$a(t) = [p_1(t), p_2(t), ..., p_N(t), e_1(t), e_2(t), ..., e_N(t)]. \quad (28)$$

Note that constraints (17) and (18) must be satisfied, which are equivalent to

$$\frac{N_0 W_i}{G_i^{BS}}(2^{\frac{l_i(t)}{T_i^{th} W_i}} - 1) \leq p_i(t) \leq P_i^m. \quad (29)$$

Also, transmitting more energy than the drone battery capacity is a waste of energy in practice, and hence

$$0 \leq e_i \leq B_i^{max}. \quad (30)$$

We hence define the action space $\mathcal{A}(t)$ at epoch t as

$$\mathcal{A}(t) = \{\boldsymbol{a}(t) \mid \boldsymbol{a}^{min}(t) \leq \boldsymbol{a}(t) \leq \boldsymbol{a}^{max}(t)\}, \quad (31)$$

where

$$\boldsymbol{a}^{min}(t) = [\frac{N_0 W_1}{G_1^{BS}}(2^{\frac{l_1}{T_1^{th} W_1}} - 1), \frac{N_0 W_2}{G_2^{BS}}(2^{\frac{l_2}{T_2^{th} W_2}} - 1), ..., \frac{N_0 W_N}{G_N^{BS}}(2^{\frac{l_N}{T_N^{th} W_N}} - 1), 0, 0, ..., 0] \quad (32)$$

and

$$\boldsymbol{a}^{max}(t) = [P_1^m, P_2^m, ..., P_N^m, B^{max}, B^{max}, ..., B^{max}], \quad (33)$$

3) System cost: The generated cost $c(t)$ is related to the network state $s(t)$ and the taken action $a(t)$, and defined as the energy cost at time epoch t. Note that constraints (21) and (23) should be satisfied, i.e.,

$$0 \leq b_i(t) + \eta_i G_i^{EH} e_i(t) - \frac{p_i(t) l_i(t)}{W_i \log_2(1 + \frac{p_i(t) G_i^{BS}}{N_0 W_i})} \leq B_i^{max}. \quad (34)$$

If the taken action $a(t) = [\boldsymbol{p}(t), \boldsymbol{e}(t)]$ violates Eq. (34), a penalty should be given to the energy cost. Hence, we define the energy cost at time epoch t as

$$c(t) = \begin{cases} \mathbb{M}, \text{ if Eq. (34) is violated,} \\ \sum_{i=1}^{N} c_1 E_i^{trs}(t) + c_2 e_i(t), \text{ otherwise,} \end{cases} \quad (35)$$

where $\mathbb{M}$ is a very large number to penalize the actions that violate Eq. (34).

$$c(t) = \begin{cases} \sum_{i=1}^{N} c_1 E_i^{trs}(t) + c_2 e_i(t), \text{ if } 0 \leq b_i(t) + \eta_i G_i^{EH} e_i(t) - \frac{p_i(t) l_i(t)}{W_i \log_2(1 + \frac{p_i(t) G_i^{BS}}{N_0 W_i})} \leq B_i^{max}, \\ \mathbb{M}, \text{ otherwise.} \end{cases} \quad (36)$$

4) Network state evolution: The network state $s(t)$ at time epoch t transits to $s(t+1)$ at time epoch $t+1$ according to the taken action $a(t)$. A drone's sensed data size is only related to the dynamic environment and hence drone i's sensed data size $l_i(t)$ at time epoch t and $l_i(t+1)$ at time epoch $t+1$ are independent with each other. On the other hand, the battery levels of different time epochs are coupled with each other and evolves as $b_i(t+1) = b_i(t) + \eta_i G_i^{EH} e_i(t) - \frac{p_i(t) l_i(t)}{W_i \log_2(1 + \frac{p_i(t) G_i^{BS}}{N_0 W_i})}$ (i.e., Eq. (22)).

5) Aim of MDP: The aim of a general MDP model is to find an action at each time epoch to minimize the accumulated generated cost in the long run [10]. In our system model, the MDP model tries to find the optimal wireless transmission power and the charging station transmitted energy policy $\tau(s, a) = Pr\{a(t) = a \mid s(t) = s\}$, which denotes the probability that action a is taken for a certain state s at time epoch t, in order to minimize the accumulated generated cost in the long term. Note that problem **P0** minimizes the long-term average energy cost which is equivalent to minimizing the long-term accumulated energy cost by dividing the total number of time epochs M.

To evaluate the long-term generated energy cost, we define the state-action value function [39]

$$Q(s(t), a(t)) = \mathbb{E}\{\sum_{i=t}^{M} \gamma^{(i-t)} c(t)\}, \quad (37)$$

which denotes the expected value of all future discounted cost starting from time epoch t in network state $s(t)$ with action $a(t)$ taken. $\gamma \in [0, 1]$ is the discounted factor to measure the importance of future cost. A larger γ puts more importance on the future time epochs. For example, the energy costs of future time epochs are of equal importance when $\gamma = 1$. However, we only focus on minimizing the energy cost of time epoch t when $\gamma = 0$. Therefore, the objective of the MDP is to minimize the state-action value function $Q(s(t), a(t))$ starting from the first time epoch 1, i.e.,

$$J(\pi) = \mathbb{E}\{Q(s(0), a(0))\}, \quad (38)$$

where $J(\pi)$ is the long-term discounted energy cost.

The basic idea of solving the MDP model is to choose the action with the smallest $Q(\boldsymbol{s}(t), \boldsymbol{a}(t))$ value for network state $s(t)$ at time epoch t [10]. However, it is challenging to obtain the solution of our MDP model because its actions are continuous. It is impossible to represent all state-action values $Q(s(t), a(t))$. Also, there are infinite action possibilities to be searched and compared in the lookup table where the state-action values are stored [39]. Therefore, to solve the MDP model, we utilize the actor-critic deep reinforcement learning algorithm [11], which is specifically applicable to the

time-varying decision making problem with continuous action space.

B. Actor-critic deep reinforcement learning

The actor-critic deep reinforcement learning learns the optimum action for each time epoch by interacting with the network environment to minimize the generated cost [40]. The basic idea of actor-critic deep reinforcement learning is to combine two deep neural networks (DNNs), i.e., an actor and a critic, to learn optimum power control and energy harvesting control policies. The actor generates continuous actions according to the current network state while the critic evaluates the generated actions and helps the actor update its parameters to generate the actions with better performance, as shown in Fig. 2.

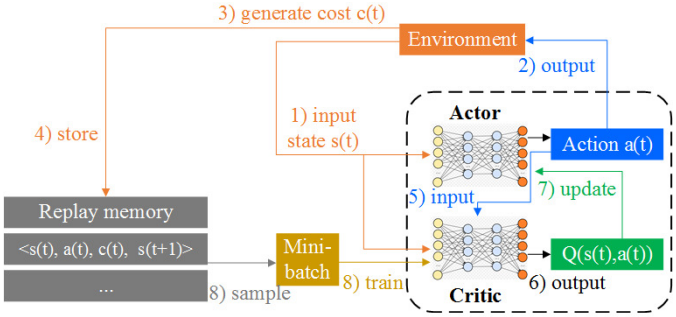

Fig. 2. Actor-critic deep reinforcement learning.

1) Actor DNN: The actor uses parameterized function $\pi_\vartheta(s)$, where ϑ is the parameter of actor DNN, to generate continuous actions for network state s. The actor takes the network state $s(t)$ as the input and outputs the action $a(t)$. Specifically, the number of nodes of the actor's input layer is $2N$ which represents the dimension of the network state vector $s(t)$, and the number of nodes of the actor's output layer is $2N$ which represents the dimension of the action a. The parameters of the actor DNN is updated by the policy gradient method [11] with the objective to minimize the long-term energy cost $J(\pi_\vartheta)$ (defined in Eq. (38)). The gradient of the objective function is

$$\nabla_\vartheta J(\pi_\vartheta) = \frac{\partial J(\pi_\vartheta)}{\partial \pi_\vartheta}\frac{\partial \pi_\vartheta}{\partial \vartheta} = \mathbb{E}\{\nabla_a Q_\theta(s,a)\nabla_\vartheta \pi_\vartheta(s)\}, \quad (39)$$

where $Q_\theta(s,a)$ is the parametrized station-action value function of the critic and θ is the critic DNN's parameter. Then, the actor's parameter ϑ is updated by the gradient descend

$$\vartheta = \vartheta - \omega_a \nabla_\vartheta J(\pi_\vartheta), \quad (40)$$

where ω_a is the actor's learning rate.

Note that the generated action may not be optimal. We hence consider the tradeoff between the exploitation and exploration [39]. Specifically, we prefer to exploit the actions with predicted smallest energy cost (i.e., the generated actions by the actor). Moreover, we still need to explore the unknown actions. Therefore, the chosen action can be calculated as

$$a(t) = \begin{cases} \text{random feasible action, with probability } \epsilon, \\ \text{actor generated } a(t), \text{ with probability } 1-\epsilon, \end{cases} \quad (41)$$

where ϵ is the probability of exploring random actions.

2) Critic DNN: The critic evaluates the actor's generated action by adapting the parametrized state-action value function $Q_\theta(s,a)$, where θ is the critic DNN's parameter. The critic takes both the network state and the actor's action $[s(t), a(t)]$ as the input, and hence the node number of the input layer becomes $4N$. The critic outputs the state-action value $Q(s(t), a(t))$ and its node number of the output layer is 1. To improve the accuracy of the critic, its parameter is updated at each time epoch by analyzing the actual generated cost from the environment with the temporal difference method [11]. The temporal difference error $\delta(t)$ is defined to measure the accuracy of the critic, and can be calculated as [11]

$$Q(s(t),a(t)) = c(t) + \gamma Q(s(t+1), a(t+1)), \quad (42)$$

$$a(t) = \underset{a}{\operatorname{argmin}}\, Q(s(t), a) \quad (43)$$

$$\delta(t) = c(t) + \gamma Q_\theta(s(t+1), a(t+1)) - Q_\theta(s(t), a(t)), \quad (44)$$

where $s(t)$, $c(t)$, $s(t+1)$, and $a(t+1)$ can be found in the replay memory. The critic's parameter θ is then updated by the gradient descend to minimize the temporal difference error $\delta(t)$, i.e.,

$$\theta = \theta - \omega_c \nabla_\theta Q_\theta(\boldsymbol{s}(t), \boldsymbol{a}(t)), \quad (45)$$

where ω_c is the critic's learning rate.

3) Replay memory: To train the critic DNN (i.e., update the critic's parameter), the network state, action, and generated cost should be stored in a replay memory, which is a finite sized first-in-first-out cache. The training sample $\langle s(t), a(t), c(t), s(t+1)\rangle$ is collected after the action is made by the actor at each time epoch. When the replay memory is full, the old samples will be discarded. At each time epoch, a mini-batch is sampled from the replay memory to train the critic DNN and update its parameter θ.

4) Operation process: The detailed process of actor-critic deep reinforcement learning is shown in Fig. 2. At each time epoch, the following steps (i.e., steps 1-8 in Fig. 2) are processed:

1) The network state $s(t)$ is inputted to the actor DNN.
2) The actor generates the action $a(t)$.
3) The action $a(t)$ acts on the environment and generates the cost $c(t)$.
4) One training sample $\langle s(t), a(t), c(t), s(t+1)\rangle$ is collected and stored in the replay memory.
5) The network state $s(t)$ and the actor's action $a(t)$ are combined and inputted to the critic DNN.
6) The critic generates the state-action value $Q(s(t), a(t))$.
7) The state-action value $Q(s(t), a(t))$ is then utilized to update the actor's parameter ϑ according to Eqs. (39) and (40).
8) A mini-batch is sampled from the replay memory to update the critic's parameter θ according to Eqs. (44) and (45).

C. Modified actor-critic deep reinforcement learning

The actor-critic deep reinforcement learning generates the power control and energy harvesting control actions from the action space $\mathcal{A}(t)$ defined in Eq. (31). The generated action may not be feasible and violate the constraint Eq. (34). In this case, the actor-critic deep reinforcement learning algorithm adds a penalty (which is usually a large number) to the generated energy cost. Hence, it may take a very long time to converge because many infeasible solutions are considered and compared in the action space. In order to address this problem, we propose PEARL which is a modified actor-critic deep reinforcement learning algorithm.

Algorithm 1: PEARL

Input : $N, M, G_i^{BS}, G_i^{EH}, N_0, W_i, l_i, T_i^{th}, P_i^m, B_i^{max}, c_1, c_2, \gamma, \omega_a, \omega_c, \epsilon$

Output: policy π

1 Initialize the actor and critic DNNs with weight parameters ϑ and θ, respectively ;
2 Initialize the time epoch $t = 1$;
3 Initialize network state $s(1)$;
4 **for** *each time epoch* t **do**
5 Calculate the action $a(t) = [\boldsymbol{p}(t), \boldsymbol{e}(t)]$ based on the actor DNN according to Eq. (41);
6 Choose the wireless transmission power $\boldsymbol{p}^*(t) = \boldsymbol{p}(t)$;
7 Calculate the feasible transmitted energy $\boldsymbol{e}^*(t)$ according to Eq. (48) ;
8 Choose the action $a^*(t) = [\boldsymbol{p}^*(t), \boldsymbol{e}^*(t)]$;
9 Generate the cost $c(t)$ according to Eq. (51) ;
10 Observe the network state $\boldsymbol{s}(t+1)$;
11 Store the tuple $\langle s(t), a^*(t), c(t), s(t+1)\rangle$ in the replay memory ;
12 Update the actor DNN parameter ϑ according to Eqs. (39) and (40) ;
13 Sample a mini-batch of tuples from the replay memory ;
14 Update the critic DNN parameter θ according to Eqs. (44) and (45);
15 $t \leftarrow t + 1$;
16 **end**

The basic of PREAL is to only utilize the power control policy $p_i^*(t), i \in \mathcal{N}$ from the actor-critic deep reinforcement learning algorithm, and substitute $p_i^*(t)$ to the constraint Eq. (34). Hence, we have

$$\begin{aligned}&\frac{1}{\eta_i G_i^{EH}}[\frac{p_i^*(t) l_i(t)}{W_i \log_2(1+\frac{p_i^*(t)G_i^{BS}}{N_0 W_i})} - b_i(t)] \le e_i(t) \\ &\le \frac{1}{\eta_i G_i^{EH}}[B_i^{max} + \frac{p_i^*(t) l_i(t)}{W_i \log_2(1+\frac{p_i^*(t)G_i^{BS}}{N_0 W_i})} - b_i(t)]\end{aligned} \tag{46}$$

$$\frac{1}{\eta_i G_i^{EH}}[\frac{p_i^*(t) l_i(t)}{W_i \log_2(1+\frac{p_i^*(t)G_i^{BS}}{N_0 W_i})} - b_i(t)] \le e_i(t) \le \frac{1}{\eta_i G_i^{EH}}[B_i^{max} + \frac{p_i^*(t) l_i(t)}{W_i \log_2(1+\frac{p_i^*(t)G_i^{BS}}{N_0 W_i})} - b_i(t)] \tag{47}$$

which is then utilized to constrain the transmitted energy $e_i(t)$. Then, the feasible transmitted energy $e_i^*(t)$ can be calculated by

$$e_i^*(t) = \begin{cases} e_i^{min}(t), \text{ if } e_i(t) < e_i^{min}(t), \\ e_i^{max}(t), \text{ if } e_i(t) > e_i^{max}(t), \\ e_i(t), \text{ otherwise}, \end{cases} \quad \forall i \in \mathcal{N}, \tag{48}$$

where we denote

$$e_i^{min}(t) = \frac{1}{\eta_i G_i^{EH}}[\frac{p_i^*(t) l_i(t)}{W_i \log_2(1+\frac{p_i^*(t)G_i^{BS}}{N_0 W_i})} - b_i(t)] \tag{49}$$

and

$$e_i^{max}(t) = \frac{1}{\eta_i G_i^{EH}}[B_i^{max} + \frac{p_i^*(t) l_i(t)}{W_i \log_2(1+\frac{p_i^*(t)G_i^{BS}}{N_0 W_i})} - b_i(t)] \tag{50}$$

for simplicity. Since the action $[\boldsymbol{p}^*(t), \boldsymbol{e}^*(t)]$ guarantees the feasibility, the generated energy cost from the IoD networks can be calculated by

$$c(t) = \sum_{i=1}^{N}[c_1 \frac{p_i^*(t) l_i}{W_i \log_2(1+\frac{p_i^*(t)G_i^{BS}}{N_0 W_i})} + c_2 e_i^*(t)]. \tag{51}$$

The detailed process of our proposed PEARL is delineated in Alg. 1. Lines 1-3 initialize the actor, critic, and network state. Lines 4-16 are operated at each time epoch. Line 5 calculates the generated action $a(t)$ based on the actor DNN. Lines 6-8 fix the power control policy $\boldsymbol{p}(t)$, try to find a feasible energy harvesting policy $\boldsymbol{e}^*(t)$, and choose the modified action $a^*(t)$. Lines 9-11 generate the energy cost $c(t)$ and observe the next network state $s(t+1)$. Line 11 stores a training sample $\langle s(t), a^*(t), c(t), s(t+1)\rangle$ in the replay memory. Line 12 update the actor DNN parameter ϑ. Lines 13-14 sample a mini-batch from the replay memory to update the critic DNN parameter θ.

VI. Performance Evaluation

We setup simulations to evaluate the performances of our proposed algorithm, PEARL, in this section. The simulations are implemented in Python by TensorFlow which is a machine learning platform [41]. The implementation of a real drone testbed will be left as our future work. We compare PEARL with two benchmark algorithms No-energy-control and Greedy. No-energy-control is a deep reinforcement learning algorithm proposed in our ICC2020 paper [28], where only the drones' wireless transmission powers are optimized. Greedy minimizes the transmitted energy at each time epoch to minimize the system energy cost while fixing the wireless transmission power as the maximum power to minimize the wireless transmission delay.

In our simulations, we consider a 1000 m × 1000 m, where the IoD gateway is located at the center of the area. The charging station is located near the IoD gateway. There are $N = 12$ drones deployed in the flying plane at the height of $H = 50$ m. The drones are randomly distributed in the flying plane to collect information from the ground. The environment related parameters in Eq. (1) are $\alpha = 9.6$ and $\beta = 0.28$. The carrier frequency $f_c = 2$ GHz. The speed of

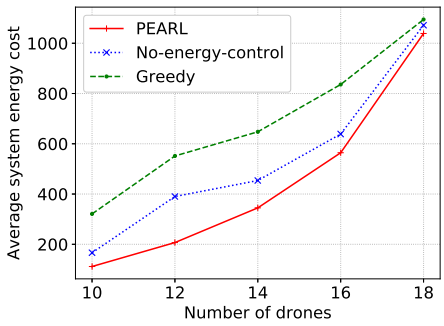

Fig. 3. Average system energy cost vs number of drones

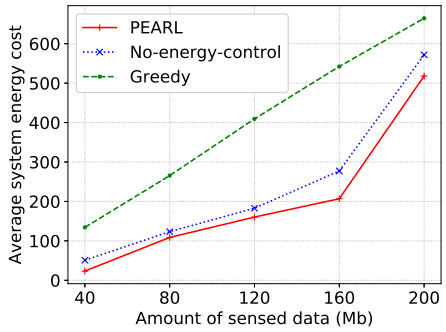

Fig. 4. Average system energy cost vs amount of sensed data

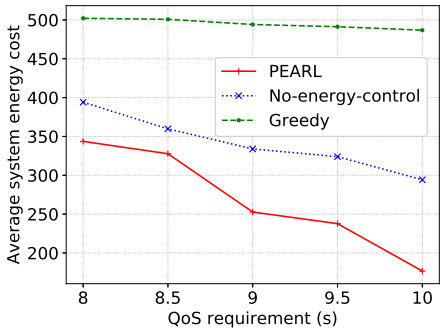

Fig. 5. Average system energy cost vs QoS requirement

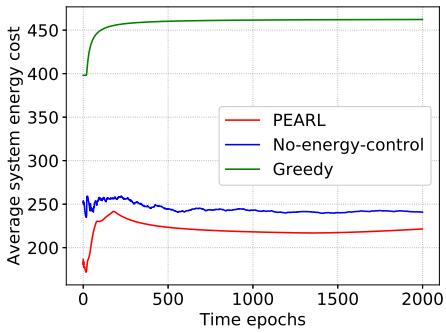

Fig. 6. Average system energy cost vs time epochs with different algorithms

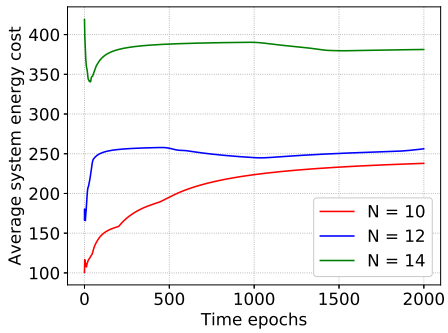

Fig. 7. Average system energy cost vs time epochs with different numbers of drones

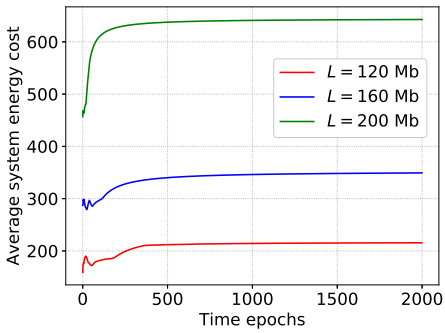

Fig. 8. Average system energy cost vs time epochs with different amount of sensed data

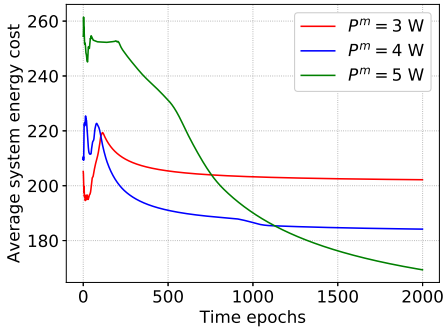

Fig. 9. Average system energy cost vs time epochs with different maximum wireless transmission powers

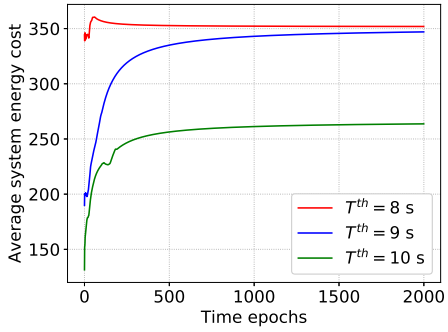

Fig. 10. Average system energy cost vs time epochs with different QoS requirements

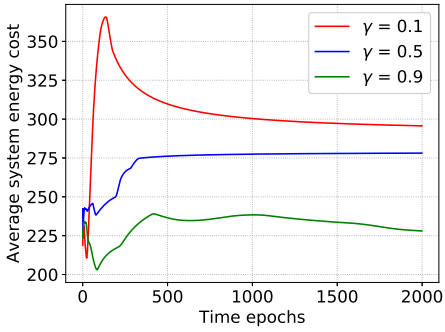

Fig. 11. Average system energy cost vs time epochs with different discounted factors

light $c = 3 \times 10^8$ *m/s*. The environment-related parameters for calculating the path losses in Eqs. (3) and (4) are $\xi^{LoS} = 1$ and $\xi^{NLoS} = 20$ *dB*. The system bandwidth $W = 20$ *MHz* and is evenly allocated to all drones. The noise power density $N_0 = -174$ *dBm/Hz*. The amount of sensed data of each drone is randomly chosen from 100 to 200 *Mb*. Each drone's maximum wireless transmission power is $P^m = 5$ *W*. The battery capacity of each drone is $B^{max} = 800$ *J*. The energy harvesting efficiency $\eta = 0.5$. The unit energy cost c_1 and c_2 are normalized as 1 and 10^{-12}, respectively. In PEARL, the discounted factor $\gamma = 0.9$, both actor and critic DNN are fully connected and have 1 hidden layer, and each hidden layer has 64 nodes. Note that the above parameters are default values and they may change if we specify them.

Fig. 3 illustrate the average system energy cost of three different algorithms after convergence with different numbers of drones ranging from 10 to 18. The average system energy costs of all three algorithms increase with the number of drones because more drones incur more battery charging and a larger amount of transmitted data and hence more energy consumption. PEARL generates the less energy cost than No-energy-control because it jointly optimizes the wireless transmission power and the transmitted energy from the charging station, while No-energy-control only optimizes the transmission power and assumes that all drones' batteries are charged to its fullest. PEARL performs better than Greedy because PEARL considers the policies over different time epochs and utilizes the past experiences to improve its performance, while

Greedy only optimizes its solution within one time epoch.

Fig. 4 compares the average system energy cost of PEARL with that of No-energy-control and Greedy with different amounts of sensed data ranging from 40 to 200 *Mb*. The average system energy costs of all three algorithms become larger when the amount of sensed data increases because more sensed data means more energy is required to transmit these data, thus increasing the system energy cost. PEARL generates the least average system energy cost among the three algorithms for the same reason as in Fig. 3.

Fig. 5 evaluates the PEARL's average system energy cost with different QoS requirements (i.e., minimum data transmission delay) ranging from 8 to 10 *s*. The average system energy cost of all three algorithms decreases when the QoS requirement becomes less strict (i.e., larger minimum data transmission delay), because a less strict QoS requirement implies that less energy is required to meet the requirement. Similarly, PEARL performs better than No-energy-control and Greedy.

Fig. 6 illustrates how PEARL, No-energy-control and Greedy converge at different time epochs. Greedy independently optimizes its solutions within each time epoch and so is more likely to obtain similar results at different time epochs when the network status is stable. Hence, Greedy achieves a fast convergence rate. However, both PEARL and No-energy-control are deep reinforcement learning algorithms which are trial-and-error processes, and hence require more time to converge. Additionally, we can observe that PEARL performs the best among the three algorithms for the similar reason in Fig. 3.

We then investigate the impacts of different parameters on the performance of PEARL in Figs. 7 to 11. Fig. 7 illustrates PEARL's average system energy cost with three different numbers of drones including 10, 12, and 14. A larger number of drones incur more energy cost because more drones imply more data to be transmitted and more energy to transmit these data. Fig. 8 compares the PEARL's average system energy costs with different amounts of sensed data including 120, 160, and 200 *Mb*. A larger amount of sensed data incurs more energy cost because more sensed data requires more energy to transmit them. Fig. 9 evaluates PEARL's average system energy cost with different maximum wireless transmission powers including 3, 4, and 5 *W*. A larger maximum wireless transmission power incurs less energy cost because a larger maximum wireless transmission power provides a larger action space and more possible solutions, hence improving the probability of finding a solution with better performance. However, a larger action space requires more time to converge. Therefore, a larger maximum wireless transmission power incurs a slower convergence rate. Fig. 10 evaluates PEARL's performance with difference QoS requirements including 8, 9, and 10 *s*. A stricter QoS requirement (i.e., smaller minimum data transmission delay) incurs less system energy cost because less energy is required to meet the QoS requirement. Fig. 11 illustrates PEARL's average energy cost with different discounted factors including 0.1, 0.5, and 0.9. The discounted factor measures the importance of future time epochs. Since we try to minimize the long-term average system energy cost, a larger gamma, which puts more importance to future time epochs, achieves a better performance, i.e., less average system energy cost.

VII. Conclusion

In this paper, we have investigated the joint optimization of power control and energy harvesting control in time-varying IoD networks. We have formulated our joint optimization problem to determine each drone's wireless transmission power and the transmitted energy from the charging station to each drone at each time epoch with the objective to minimize the long-term average system energy cost constrained by the drones' battery capacities and QoS requirements. An MDP has been formulated to characterize our problem in time-varying IoD networks to show how the network status evolves with different power and energy harvesting control policies. We have designed a modified actor-critic deep reinforcement learning algorithm to solve our problem. We have demonstrated via extensive simulations that our proposed algorithm performs better than the existing algorithms and the impacts of different parameters on the performance of our proposed algorithm.

References

[1] J. Yao and N. Ansari, "Task allocation in fog-aided mobile IoT by Lyapunov online reinforcement learning," *IEEE Trans. Green Commun. Netw.*, vol. 4, no. 2, pp. 556–565, Jun. 2020.

[2] N. Ansari *et al.*, "SoarNet," *IEEE Wireless Commun.*, vol. 26, no. 6, pp. 37–43, Dec. 2019.

[3] H. Ghazzai *et al.*, "Energy-efficient management of unmanned aerial vehicles for underlay cognitive radio systems," *IEEE Trans. Green Commun. Netw.*, vol. 1, no. 4, pp. 434–443, Dec. 2017.

[4] M. Gharibi, R. Boutaba, and S. L. Waslander, "Internet of drones," *IEEE Access*, vol. 4, pp. 1148–1162, Mar. 2016.

[5] J. Yao and N. Ansari, "QoS-aware power control in internet of drones for data collection service," *IEEE Trans. Veh. Technol.*, vol. 68, no. 7, pp. 6649–6656, Jul. 2019.

[6] L. Gupta, R. Jain, and G. Vaszkun, "Survey of important issues in UAV communication networks," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1123–1152, 2016.

[7] M. Lu *et al.*, "Wireless charging techniques for UAVs: A review, reconceptualization, and extension," *IEEE Access*, vol. 6, pp. 29 865–29 884, 2018.

[8] X. Lu *et al.*, "Wireless networks with RF energy harvesting: A contemporary survey," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 2, pp. 757–789, 2015.

[9] I. Krikidis, S. Timotheou, S. Nikolaou, G. Zheng, D. W. K. Ng, and R. Schober, "Simultaneous wireless information and power transfer in modern communication systems," *IEEE Commun. Mag.*, vol. 52, no. 11, pp. 104–110, Nov. 2014.

[10] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2014.

[11] I. Grondman *et al.*, "A survey of actor-critic reinforcement learning: Standard and natural policy gradients," *IEEE Trans. Cybern.*, vol. 42, no. 6, pp. 1291–1307, Nov. 2012.

[12] M. Wazid *et al.*, "Design and analysis of secure lightweight remote user authentication and key agreement scheme in internet of drones deployment," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3572–3584, Apr. 2019.

[13] B. Bera *et al.*, "Blockchain-envisioned secure data delivery and collection scheme for 5G-based IoT-enabled internet of drones environment," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 9097–9111, Aug. 2020.

[14] J. Yao and N. Ansari, "Online task allocation and flying control in fog-aided internet of drones," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5562–5569, May 2020.

[15] D. Altinel and G. K. Kurt, "Modeling of hybrid energy harvesting communication systems," *IEEE Trans. Green Commun. Netw.*, vol. 3, no. 2, pp. 523–534, Jun. 2019.

[16] T. D. Nguyen, J. Y. Khan, and D. T. Ngo, "A distributed energy-harvesting-aware routing algorithm for heterogeneous IoT networks," *IEEE Trans. Green Commun. Netw.*, vol. 2, no. 4, pp. 1115–1127, Dec. 2018.

[17] J. Yao and N. Ansari, "Caching in energy harvesting aided internet of things: A game-theoretic approach," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3194–3201, Apr. 2019.

[18] A. M. Jawad *et al.*, "Wireless power transfer with magnetic resonator coupling and sleep/active strategy for a drone charging station in smart agriculture," *IEEE Access*, vol. 7, pp. 139 839–139 851, 2019.

[19] J. Yao and N. Ansari, "QoS-aware fog resource provisioning and mobile device power control in IoT networks," *IEEE Trans. Netw. Service Manag.*, vol. 16, no. 1, pp. 167–175, Mar. 2019.

[20] K. Lee and J. Hong, "Power control for energy efficient D2D communication in heterogeneous networks with eavesdropper," *IEEE Commun. Lett.*, vol. 21, no. 11, pp. 2536–2539, Nov. 2017.

[21] P. Mach and Z. Becvar, "Cloud-aware power control for real-time application offloading in mobile edge computing," *Trans. Emerg. Telecommun. Technol.*, vol. 27, no. 5, p. 648661, May 2016. [Online]. Available: https://doi.org/10.1002/ett.3009

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

[23] P. Pace *et al.*, "Intelligence at the edge of complex networks: The case of cognitive transmission power control," *IEEE Wireless Commun.*, vol. 26, no. 3, pp. 97–103, 2019.

[24] N. C. Luong *et al.*, "Applications of deep reinforcement learning in communications and networking: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3133–3174, 2019.

[25] L. Lei *et al.*, "Multiuser resource control with deep reinforcement learning in IoT edge computing," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10 119–10 133, Dec. 2019.

[26] J. Yao and N. Ansari, "Caching in dynamic IoT networks by deep reinforcement learning," *IEEE Internet Things J.*, 2020, doi: 10.1109/JIOT.2020.3004394, early access.

[27] C. H. Liu, Q. Lin, and S. Wen, "Blockchain-enabled data collection and sharing for industrial IoT with deep reinforcement learning," *IEEE Trans. Ind. Informat.*, vol. 15, no. 6, pp. 3516–3526, Jun. 2019.

[28] J. Yao and N. Ansari, "Power control in internet of drones by deep reinforcement learning," in *Proc. IEEE ICC 2020*, Dublin, Jun. 7-11, 2020, pp. 1–6.

[29] M. Ku *et al.*, "Advances in energy harvesting communications: Past, present, and future challenges," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1384–1412, 2016.

[30] R. G. Bernacki and N. Salamon, "Experimental study of energy harvesting in UHF band," in *J. Phys. Conf. Ser.*, vol. 709, 2016, p. 012009.

[31] J. Ho and M. Jo, "Offloading wireless energy harvesting for IoT devices on unlicensed bands," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3663–3675, 2019.

[32] B. Kiumarsi and F. L. Lewis, "Actorcritic-based optimal tracking for partially unknown nonlinear discrete-time systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 1, pp. 140–151, Jan. 2015.

[33] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Drone small cells in the clouds: Design, deployment and performance analysis," in *Proc. IEEE GLOBECOM 2015*, San Diego, CA, USA, Dec. 2015, pp. 1–6.

[34] G. Auer *et al.*, "How much energy is needed to run a wireless network?" *IEEE Wireless Commun.*, vol. 18, no. 5, pp. 40–49, Oct. 2011.

[35] S. Sudevalayam and P. Kulkarni, "Energy harvesting sensor nodes: Survey and implications," *IEEE Commun. Surveys Tuts.*, vol. 13, no. 3, pp. 443–461, Third quater 2011.

[36] 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.

[37] J. Gundlach, *Designing unmanned aircraft systems: a comprehensive approach*. American Institute of Aeronautics and Astronautics, 2012.

[38] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex Optimization*. Cambridge university press, 2004.

[39] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[40] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[41] M. Abadi *et al.*, "Tensorflow: A system for large-scale machine learning," in *Proc. USENIX OSDI 2016*, Savannah, GA, USA, 2016, pp. 265–283.

Jingjing Yao (S17) received the B.E. degree in information and communication engineering from Dalian University of Technology (DUT) and the M.E. degree in information and communication engineering from University of Science and Technology of China (USTC). She is currently working towards the Ph.D. degree in Computer Engineering at the New Jersey Institute of Technology (NJIT), Newark, New Jersey. Her research interests include Internet of Things, Machine Learning, Drone-assisted Network, Mobile Edge Computing/Caching.

Nirwan Ansari (S78-M83-SM94-F09), Distinguished Professor of Electrical and Computer Engineering at the New Jersey Institute of Technology (NJIT), received his Ph.D. from Purdue University, MSEE from the University of Michigan, and BSEE (summa cum laude with a perfect GPA) from NJIT. He is also a Fellow of National Academy of Inventors (NAI).

He authored Green Mobile Networks: A Networking Perspective (Wiley-IEEE, 2017) with T. Han, and co-authored two other books. He has also (co-)authored more than 600 technical publications. He has guest-edited a number of special issues covering various emerging topics in communications and networking. He has served on the editorial/advisory board of over ten journals including as Associate Editor-in-Chief of IEEE Wireless Communications Magazine. His current research focuses on green communications and networking, cloud computing, drone-assisted networking, and various aspects of broadband networks.

He was elected to serve in the IEEE Communications Society (ComSoc) Board of Governors as a member-at-large, has chaired some ComSoc technical and steering committees, is current Director of ComSoc Educational Services Board, has been serving in many committees such as the IEEE Fellow Committee, and has been actively organizing numerous IEEE International Conferences/Symposia/Workshops. He is frequently invited to deliver keynote addresses, distinguished lectures, tutorials, and invited talks. Some of his recognitions include several excellence in teaching awards, a few best paper awards, the NCE Excellence in Research Award, several ComSoc TC technical recognition awards, the NJ Inventors Hall of Fame Inventor of the Year Award, the Thomas Alva Edison Patent Award, Purdue University Outstanding Electrical and Computer Engineering Award, the NCE 100 Medal, and designation as a COMSOC Distinguished Lecturer. He has also been granted more than 40 U.S. patents.