Power Control in Internet of Drones by Deep Reinforcement Learning

Jingjing Yao, *Student Member, IEEE* and Nirwan Ansari, *Fellow, IEEE*
Advanced Networking Lab., Helen and John C. Hartmann Department of Electrical & Computer Engineering
New Jersey Institute of Technology, Newark, NJ 07102, USA.
Emails: {jy363, nirwan.ansari}@njit.edu

Abstract—**Internet of Drones (IoD) employs drones as the internet of things (IoT) devices to provision applications such as traffic surveillance and object tracking. Data collection service is a typical application where multiple drones are deployed to collect information from the ground and send them to the IoT gateway for further processing. The performance of IoD networks is constrained by drones' battery capacities, and hence we utilize both energy harvesting technologies and power control to address this limitation. Specifically, we optimize drones' wireless transmission power at each time epoch in energy harvesting aided time-varying IoD networks for the data collection service with the objective to minimize the average system energy cost. We then formulate a Markov Decision Process (MDP) model to characterize the power control process in dynamic IoD networks, which is then solved by our proposed model-free deep actor-critic reinforcement learning algorithm. The performance of our algorithm is demonstrated via extensive simulations.**

Index Terms—**Power control, internet of drones (IoD), energy harvesting, deep reinforcement learning, actor-critic, quality of service (QoS)**

I. Introduction

Internet of drones (IoD), which employs drones as the internet of things (IoT) devices, has been explored in applications including object tracking, traffic surveillance and disaster rescue [1, 2]. A widely used application of IoD is the data collection service where multiple drones are deployed to collect data (e.g., pictures and videos) from the ground. The collected data are then sent to the IoT gateway (GW) for further processing [3]. To guarantee the quality of service (QoS), a minimum wireless transmission rate is usually defined [4].

The performance of IoD networks is greatly constrained by drones' battery capacities. There are usually two approaches to address this challenge. The first approach is to utilize a wireless charging station to charge drone batteries in IoD networks [5, 6, 7]. This approach may incur additional energy cost and infrastructure expenditures. The other approach is to adjust each drone's transmission power to reduce the energy consumptions of drones' batteries [8]. Hence, power control is an important issue in IoD networks.

The conventional power control is usually posed as an one-shot static optimization problem and optimizes the system energy consumption in a greedy manner. However, the static optimization does not consider the correlations among the power control decisions across time. Owing to the limited battery capacity and wireless charging, the energy consumptions over different time epochs are related and hence the power control policies influence one another. In our work, we hence use reinforcement learning (RL), which is usually adopted for sequential decision-making problem in time-varying environment [9, 10], to address the power control problem in dynamic IoD networks. RL interacts with the environment by taking actions after observing the current environment state and then obtains a reward and transits to a new environment state. A Markov decision process (MDP) is usually modeled to characterize the interaction.

The IoD environment is usually difficult to model because the accurate and complete information of the environment is unknown in dynamic networks. Hence, we utilize a model-free RL framework [11] to address our problem. In order to determine which action to take, a state-action value function is defined to evaluate the actions in a certain state. Since it is impossible to explicitly represent each value function, we hence utilize the deep neural networks to estimate the state-action values [11].

Motivated by the above analysis, we investigate the QoS-aware power control in time-varying energy harvesting aided IoD networks for the data collection service by model-free deep reinforcement learning. Specifically, we aim to optimize the wireless transmission power for each drone at each time epoch with the objective to minimize the average system energy consumption constrained by the QoS requirement.

The rest of the paper is organized as follows. A summary of related works is presented in Section II. We describe our IoD system model with wireless charging in Section III. The power control problem in IoD networks is then formulated in Section IV. In Section V, we describe the deep actor-critic algorithm to solve the problem. Simulation results are analyzed in Section VI. Section VII concludes the paper.

II. Related Works

IoD was first proposed in [2], in which the IoD system is divided into five conceptual layers. Yao and Ansari [12] investigated the drone trajectory optimization in IoD networks for the sensing service to minimize the task completion time constrained by the drone's battery capacity. Chen and Wang [13] proposed an IoD cloud surveillance system where data collected by drones are outsourced to the cloud to be analyzed. However, none of the above works consider the power control problem in IoD networks.

MDP models are usually employed to formulate the reinforcement learning problem. Chen *et al.* [14] investigated the traffic offloading problem in cellular networks to minimize

This work was supported in part by the U.S. National Science Foundation under Grant No. CNS-1814748.

the system energy consumption and formulated the problem as an MDP model which was further solved by a Q-learning algorithm. Actor-critic reinforcement learning was designed to solve problems with continuous action space. Zhang *et al.* [15] designed an online actor-critic reinforcement learning algorithm to address the traffic offloading and resource allocation in energy harvesting aided mobile edge computing systems. Wei *et al.* [16] investigated the user scheduling and resource allocation in heterogenous networks with the objective to maximize the network energy efficiency. However, actor-critic reinforcement learning has not been deployed in IoD networks.

Yao and Ansari [17] investigated the power control in IoD networks for the data collection service to minimize the drone's power consumption while satisfying the QoS requirement. However, their work was a static optimization problem because they assumed that the network status remained the same. Moreover, their work only considered a single drone in the IoD network. To the best of our knowledge, this is the first work to consider the power control in time-varying energy harvesting aided IoD networks with multiple drones deployed to minimize the average system energy cost by deep reinforcement learning.

III. System Model

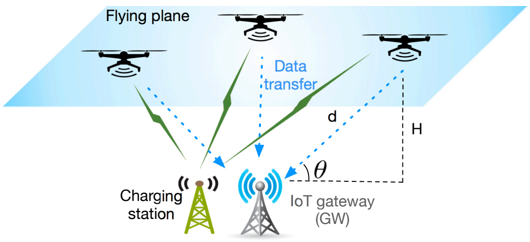


Fig. 1. Data collection in IoD.

In our system model (as shown in Fig. 1), we consider one IoT GW and N drones hovering above the service area. The set of drone indexes is denoted as $\mathcal{N} = \{1, 2, ..., N\}$. In the data collection service, drones collect data (e.g., images and videos) of different locations and send them to the IoT GW for further processing [18]. We assume the network operates at discrete time epochs. The network status is considered static within a time epoch but varies over different epochs. At each epoch, each drone $i \in \mathcal{N}$ transmits its collected data of length l_i to the IoT GW. In our system model, we characterize the QoS requirement as the minimum wireless transmission rate R_i^{th} of drone i, i.e., drone i's wireless transmission rate should be no less than R_i^{th}. A charging station is used to charge the drone batteries to help maintain drone operations including data transmissions [19].

A. Drone Data Transmission Rate

The data transmission rates of drones are the wireless transmission rates between drones and the IoT GW. We adopt the widely used probability model where the signal between drones and IoT GW can be either Line-of-Sight (LoS) or Non-Line-of-Sight (NLoS) with probabilities $Pr(LoS)$ and $Pr(NLoS)$, respectively [20]. The probabilities are functions of the height of a drone and distance between the drone and the IoT GW, which are defined as $Pr(LoS) = \frac{1}{1+\alpha \exp(-\beta[\frac{180}{\pi} \arctan(\frac{H}{d})-\alpha])}$ and $Pr(NLoS) = 1 - Pr(LoS)$, where α and β are environment-related constants (e.g., rural and urban), H is the height of the drone and d is the distance between the drone and the IoT GW. We also adopt the free space propagation loss to characterize the drone's signal path loss model [21]. For LoS and NLoS signals, the path losses are respectively defined as $PL_{LoS} = 20\log_{10}(\frac{4\pi f_c d}{c}) + \xi_{LoS}$ and $PL_{NLoS} = 20\log_{10}(\frac{4\pi f_c d}{c}) + \xi_{NLoS}$, where f_c is the carrier frequency, c is the speed of light, and ξ_{LoS} and ξ_{NLoS} are environment-related constants [21]. Hence, we utilize the average path loss to characterize the path loss between the drone and the IoT GW, which is defined as $\overline{PL} = Pr(LoS) \times PL_{LoS} + Pr(NLoS) \times PL_{NLoS}$. Therefore, the drone i's data transmission rate can be calculated by

$$r_i = W \log_2(1 + \frac{p_i G_i}{N_0 W}), \tag{1}$$

where G_i is the wireless channel gain between drone i and the IoT GW and is calculated as $G_i = 10^{-\frac{\overline{PL}}{10}}$; p_i is drone i's wireless transmission power; W is the system bandwidth and N_0 is the noise power spectrum density.

$$r_i(t+1) = W \log_2(1 + \frac{a_i(t) G_i}{N_0 W}) \tag{2}$$

B. Drone's Energy Consumption

A drone's energy is consumed for the drone's wireless data communications and hovering in the air [22]. A drone's energy consumption for hovering is not related to its wireless transmission power and is usually a fixed number over different equal-length time epochs [22], and hence does not affect the results of our power control optimization problem. Hence, we only include the energy consumption for wireless data transmissions in our objective function. Drone i's energy consumption for transmitting collected data can be expressed as [23]

$$E_i = p_i T_i = \frac{p_i l_i}{r_i} = \frac{p_i l_i}{W \log_2(1 + \frac{p_i G_i}{N_0 W})}, \tag{3}$$

where T_i is the time duration for drone i's data transmission, l_i is the data size of drone i's collected data, and r_i is the drone i's data transmission rate from Eq. (2).

C. Rechargeable Drone Battery

We assume that all drone batteries are rechargeable in our system. The charged energy can be stored in the battery and used for future data transmission [24]. The charging process can be implemented by controllable energy harvesting technologies (e.g., RF energy harvesting) in a charging station [19] as shown in Fig. 1. We denote $\boldsymbol{b}(t) = [b_1(t), b_2(t), ..., b_N(t)]$ as the states of drone batteries, where $b_i(t) \in [0, B^{max}]$ is drone i's battery level at the beginning of time epoch t and B^{max} is the battery capacity of each drone. At each epoch t, if the battery level of drone i is smaller than the energy consumption of data transmission E_i, the battery should be charged to its fullest B^{max}. Otherwise, the data transmission uses the existing energy in the drone's battery. We utilize a binary variable $x_i(t) \in \{0, 1\}$ to indicate whether drone i's battery is charged or not at time epoch t. $x_i(t)$ is then given by

$$x_i(t) = \begin{cases} 1, \text{ if } b_i(t) < E_i(t), \\ 0, \text{ otherwise.} \end{cases} \tag{4}$$

If drone i is charged at epoch t, its battery energy level becomes $B^{max} - E_i(t)$ at the beginning of epoch $t+1$. Otherwise, the battery energy level becomes $b_i(t) - E_i(t)$. Hence, drone i's battery energy level evolves based on

$$\begin{aligned} b_i(t+1) &= [B^{max} - E_i(t)]x_i(t) + [b_i(t) - E_i(t)](1 - x_i(t)) \\ &= b_i(t) - E_i(t) + [B^{max} - b_i(t)]x_i(t). \end{aligned} \tag{5}$$

If drone i is charged at epoch t, it consumes $E_i(t)$ battery energy for data transmission and $\frac{B^{max} - b_i(t)}{\rho_i}$ energy of the charging station, where ρ_i is the energy harvesting efficiency to measure how much energy consumed in the charging station can be transformed into the drone's battery [5]. The system energy cost consists of the energy cost from all drones' batteries and the charging station. Therefore, the system energy cost $E^{sys}(t)$ at time epoch t can be calculated as

$$\begin{aligned} E^{sys}(t) &= \sum_{i=1}^{N}\{[c_1 E_i(t) + c_2 \frac{B^{max} - b_i(t)}{\rho_i}]x_i(t) + \\ &\qquad c_1 E_i(t)(1 - x_i(t))\} \\ &= \sum_{i=1}^{N}\{c_1 E_i(t) + c_2 \frac{B^{max} - b_i(t)}{\rho_i} x_i(t)\} \\ &= \sum_{i=1}^{N}\{\frac{c_1 l_i p_i(t)}{W \log_2[1 + \frac{p_i(t)G_i(t)}{N_0 W}]} \\ &\qquad + c_2 \frac{B^{max} - b_i(t)}{\rho_i} x_i(t)\}, \end{aligned} \tag{6}$$

where c_1 and c_2 are the coefficients of the drone's and charging station's energy consumption respectively to measure the importances of these two parts [19]. In practice, c_1 and c_2 can be considered as the energy cost per joule of drone's battery and charging station, respectively.

IV. Problem Formulation

We formulate the power control problem in IoD networks for sensing service in this section. N drones are deployed to collect information which is then sent to the IoT GW for further processing. Our aim is to minimize the average system energy cost of all drones while satisfying the QoS requirements. The problem is then formulated as

$$\textbf{P0:} \quad \min_{p_i(t)} \frac{1}{t}\sum_{t=1}^{\infty} E^{sys}(t) \tag{7}$$

$$s.t. \quad p_i(t) \leq P^m, \ \forall i \in \mathcal{N}, t \in \{0, 1, 2...\}, \tag{8}$$

$$W \log_2(1 + \frac{p_i(t)G_i(t)}{N_0 W}) \geq R_i^{th}, \ \forall i \in \mathcal{N}, t \in \{0, 1, 2...\}. \tag{9}$$

Eq. (7) is the objective function to minimize the average system energy consumption. Eq. (8) defines the maximum wireless transmission power p^m. Eq. (9) is the QoS constraint to impose each drone's wireless transmission rate to be no less than the threshold R_i^{th}.

Note that problem **P0** is non-convex, and hence it is challenging to obtain the global optimal solution. Moreover, problem **P0** at each epoch requires complete information of different epochs (i.e., both the historical and future epochs) to achieve global optimality because they are coupled with each other through each drone's battery level status and energy consumption. However, such complete information may be not available in practice, especially in a dynamic network environment (e.g., changing IoT data, locations of drones and wireless channel conditions); obtaining optimal strategies in this case becomes intractable. We hence utilize the reinforcement learning method to make decisions by interacting with the environment. Specifically, the reinforcement learning maps the environment states to optimal actions by the learning experiences in order to minimize the generated cost.

We define a Markov Decision Process (MDP) $< \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{C} >$ to model the power control process of our work, which consists of the network state space $\mathcal{S}$, associated action space $\mathcal{A}$, state transition (from one state to another) probability density function $\mathcal{F} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto [0, \infty)$ and cost functions $\mathcal{C} : \mathcal{S} \times \mathcal{A} \mapsto [0, \infty)$. In our system, the network controller (i.e., IoT GW) observes the network state $\boldsymbol{s}(t)$ in the current epoch t and then determines the corresponding action $\boldsymbol{a}(t)$ with continuous state space and action space at the beginning of the epoch, while generating a cost $c(\boldsymbol{s}(t), \boldsymbol{a}(t))$ at the end of the epoch. We define $\boldsymbol{s}(t)$ as a set of two parts including all drone's wireless transmission rates and battery levels. Thus, the network state at epoch t can be expressed as

$$\boldsymbol{s}(t) = [r_1(t), r_2(t), ..., r_N(t), b_1(t), b_2(t), ..., b_N(t)]. \tag{10}$$

The action of the system $\boldsymbol{a}(t)$ determines the power control strategy $p_i(t)$ of each drone and can be defined as

$$\boldsymbol{a}(t) = [p_1(t+1), p_2(t+1), ..., p_N(t+1)]. \tag{11}$$

Note that the constraints in problem **P0** (i.e., Eqs. (8) and (9)) must be satisfied. We hence define the action space at epoch t as

$$\mathcal{A}(t) = \{\boldsymbol{p}(t+1) \mid \frac{N_0 W}{G_i}(2^{\frac{R_i^{th}}{W}} - 1) \leq p_i(t+1) \leq P^m, \ \forall i \in \mathcal{N}\}. \tag{12}$$

When the network is in state $\boldsymbol{s}(t)$ and action $\boldsymbol{a}(t)$ is chosen as the power control action, the generated cost $c(\boldsymbol{s}(t), \boldsymbol{a}(t))$ at epoch t can be considered as the total system energy cost, i.e.,

$$c(\boldsymbol{s}(t), \boldsymbol{a}(t)) = E^{sys}(t), \tag{13}$$

where $E^{sys}(t)$ can be obtained from Eq. (6).

The aim of the MDP model is to find an optimal power control policy $\pi(\boldsymbol{s}, \boldsymbol{a}) = Pr\{\boldsymbol{a}(t) = \boldsymbol{a} | \boldsymbol{s}(t) = \boldsymbol{s}\}$, which indicates the probabilities of actions to take for a certain state, with the objective to minimize the expected value of discounted cost $J(\pi)$ over all time epochs. We define the expected value of the future discounted cost starting from $\boldsymbol{s}(t)$ and $\boldsymbol{a}(t)$ (i.e., the state-action value function) as

$$Q(\boldsymbol{s}, \boldsymbol{a}) = \mathbb{E}\{\sum_{i=t}^{\infty} \gamma^{(i-t)} c(\boldsymbol{s}(t), \boldsymbol{a}(t))\}, \tag{14}$$

where $\gamma \in [0,1]$ is the discounted factor to measure the importance of future cost. In extreme cases, we only minimize the energy cost at current epoch t when $\gamma = 0$ and the energy costs of all epochs are equally important when $\gamma = 1$. Then, the objective of the MDP is to minimize the expected cost $J(\pi)$ from the start state, which can be expressed as [25]

$$J(\pi) = \mathbb{E}\{Q(\boldsymbol{s}(0), \boldsymbol{a}(0))\}. \tag{15}$$

In order to solve the MDP model and obtain the optimal policy π, the transition probabilities $\mathcal{F}$ are required to calculate $J(\pi)$. However, the exact value of $\mathcal{F}$ is difficult to obtain because state space may be huge and requires large computing resources to list all $(\boldsymbol{s}(t), \boldsymbol{a}(t), \boldsymbol{s}(t+1)$ samples in reality. Moreover, the predefined state transition model may deviate from the actual dynamic network conditions. Hence, a model-based MDP solutions may not be practical to solve the MDP model and we thus utilize a model-free reinforcement learning method to solve this problem [11].

Another challenge of MDP is the curse of model dimension, i.e., the computational complexity greatly increases with the size of state and action spaces. Hence, it is impossible to explicitly represent each $Q(\boldsymbol{s}, \boldsymbol{a})$. Therefore, we utilize the deep neural network to estimate the state-action values [11].

V. Deep Actor-critic Reinforcement Learning

In this section, we describe the deep actor-critic reinforcement learning method to obtain the power control policy of our problem with the objective to minimize the average system energy consumption. The actor-critic reinforcement learning method is considered as an efficient tool to solve problems with continuous action spaces and deep neural networks are used to learn policies [26]. In reinforcement learning, a controller (i.e., IoT GW) optimizes its policy by interacting with the environment and generates a cost after taking the action to minimize the total accumulated cost.

Deep actor-critic reinforcement learning combines two deep neural networks (i.e., actor and critic). The actor learns the parameterized policy while the critic approximates the state-action value function and evaluates the policy obtained from the actor. Specifically, the actor uses parameterized function $\pi_\vartheta(\boldsymbol{s})$ to produce continuous action for specific state $\boldsymbol{s}$, where ϑ is the parameter of the actor's deep neural network. The critic evaluates the actor's policy by adapting the parametrized state-action value function $Q_\theta(\boldsymbol{s}, \boldsymbol{a})$ and update its parameter θ by temporal difference method [27], where θ is the parameter of the critic's deep neural network. Then, the actor's policy parameters can then be updated according to the critic's state-action value function $Q_\theta(\boldsymbol{s}, \boldsymbol{a})$ by policy gradient method [11].

A. Policy Gradient Method (Actor)

The actor utilizes the policy gradient method which produces continuous actions by parameterized policy and updates the parameter ϑ by the gradients of the objective function $J(\pi)$ defined in Eq. (15). The gradient of the objective function $\nabla_\vartheta J(\pi_\vartheta)$ can be calculated as follows:

$$\nabla_\vartheta J(\pi_\vartheta) = \frac{\partial J(\pi_\vartheta)}{\partial \pi_\vartheta} \frac{\partial \pi_\vartheta}{\partial \vartheta} = \mathbb{E}\{\nabla_a Q_\theta(\boldsymbol{s}, \boldsymbol{a}) \nabla_\vartheta \pi_\vartheta(\boldsymbol{s})\}, \tag{16}$$

where $Q_\theta(\boldsymbol{s}, \boldsymbol{a})$ is from the critic. Then, the parameter ϑ of the actor's deep neural network is updated by $\vartheta = \vartheta + \omega_a \nabla_\vartheta J(\pi_\vartheta)$, where ω_a is the actor learning rate.

Algorithm 1: Deep Actor-Critic Reinforcement Learning

```
Input  : $P^m, R_i^{th}, W, N_0, G_i, l_i, d^\pi(\boldsymbol{s}), \omega_a, \omega_c, \gamma, \tau$
Output: policy $\pi$
Initialize actor neural network $\pi_\vartheta(\boldsymbol{s})$ and critic neural network $Q_\theta(\boldsymbol{s}, \boldsymbol{a})$;
Initialize actor and critic target networks $\pi'_{\vartheta'}(\boldsymbol{s})$ and $Q'_{\theta'}(\boldsymbol{s}, \boldsymbol{a})$;
Initialize epoch $t = 0$;
Initialize state $\boldsymbol{s}(0)$;
for each time epoch $t$ do
    Calculate action $\boldsymbol{a}(t)$ based on the actor neural network $\pi_\vartheta(\boldsymbol{s})$;
    Observe network state $\boldsymbol{s}(t+1)$;
    Generate cost $c(\boldsymbol{s}(t), \boldsymbol{a}(t))$;
    Store transition $< \boldsymbol{s}(t), \boldsymbol{a}(t), c(\boldsymbol{s}(t), \boldsymbol{a}(t)), \boldsymbol{s}(t+1) >$ in the replay buffer;
    Sample a mini-batch of transitions from the reply buffer;
    Update the critic neural network by temporal difference method;
    Update the actor neural network by policy gradient method;
    Update the target actor and critic networks by $\theta' = \tau\theta + (1-\tau)\theta'$ and $\vartheta' = \tau\vartheta + (1-\tau)\vartheta'$;
end
```

B. Temporal difference Method (Critic)

The critic evaluates the policy $\pi_\vartheta(\boldsymbol{s})$ from the actor and then utilizes the temporal difference method to update the parameters θ of the critic's deep neural network. The temporal difference error $\delta(t)$ is usually used as a measurement to predict the state-action value function and can be calculated as [11]

$$\begin{aligned} \delta(t) = {} & c(\boldsymbol{s}(t+1), \boldsymbol{a}(t+1)) \\ & + \gamma Q_\theta(\boldsymbol{s}(t+1), \boldsymbol{a}(t+1)) - Q_\theta(\boldsymbol{s}(t), \boldsymbol{a}(t)), \end{aligned} \tag{17}$$

where $c(\boldsymbol{s}(t+1), \boldsymbol{a}(t+1)) + \gamma Q_\theta(\boldsymbol{s}(t+1), \boldsymbol{a}(t+1))$ is defined as the target value. The parameter θ is then updated according to the temporal difference error in a gradient descent manner, i.e.,

$$\theta(t+1) = \theta(t) + \omega_c \delta(t) \nabla_\theta Q_\theta(\boldsymbol{s}(t), \boldsymbol{a}(t)), \tag{18}$$

where ω_c is the learning rate of the critic.

Since the updated critic's neural network $Q_\theta(\boldsymbol{s}, \boldsymbol{a})$ is also used in calculating the target value in Eq. (17), this may cause conflict in the calculation and update processes [26]. Hence, we create a copy of the critic network (i.e., target critic network) $Q'_{\theta'}(\boldsymbol{s}, \boldsymbol{a})$ to calculate the target value in Eq. (17). The target network is updated by $\theta' = \tau\theta + (1-\tau)\theta'$, where $\tau \ll 1$ is to slowly change the target network so that the stability of learning can be improved [26]. Similarly, we create a target actor network $\pi'_{\vartheta'}(\boldsymbol{s})$ and update its parameter by $\vartheta' = \tau\vartheta + (1-\tau)\vartheta'$.

C. Replay Buffer

The replay buffer is a finite sized first-in-first-out cache to store transitions $< \boldsymbol{s}(t), \boldsymbol{a}(t), c(\boldsymbol{s}(t), \boldsymbol{a}(t)), \boldsymbol{s}(t+1) >$ from past experiences. When the replay buffer is full, the oldest transitions are discarded. At each time epoch, the parameters ϑ and θ of the actor and critic are updated by sampling a minibatch of the transitions in the replay buffer to train the actor and critic's neural networks.

The detailed process of the deep actor-critic reinforcement learning algorithm is delineated in Alg. 1. Lines 1-2 initialize the actor and critic neural networks and target actor and critic neural networks. Lines 5-14 calculate the policy for each time epoch and update all actor and critic networks. Line 6 gets the action $\boldsymbol{a}(t)$ according to $\pi_{\vartheta}(\boldsymbol{s})$. Line 9 stores the transition in the replay buffer. Line 10 samples a mini-batch from the reply buffer which is used for updating the critic and actor neural networks in Lines 11-12. The target actor and critic networks are updated in Line 13.

VI. Performance Evaluation

Table I. Summary of simulation parameters.

Parameter	Value
Area	1000 m × 1000 m
Number of drones N	30
Number of BSs M	5
Drone flying height H	500 m
System bandwidth W	10 *MHz*
Noise power density N_0	-174 *dBm/Hz*
BS backhaul date rate B	500 *Mbps*
Data size of drones l	1.0 *Mb* $\sim$ 4.0 *Mb*
Maximum wireless transmission power p	3 *W*
Energy harvesting efficiency ρ	20%
Battery capacity B^{max}	100 *J*

We evaluate the performance of our deep actor-critic reinforcement learning algorithm (denoted as "Actor-critic") in this section. We compare our proposed algorithm with existing greedy algorithm (denoted as "Greedy") in [17] where power control optimization is operated only within each time epoch and considers neither the past nor the future epochs. We also utilize the existing work [28] as our comparison algorithm, where the power control is not considered and the wireless transmission power is fixed (denoted as "No-power-control"). In No-power-control, we set each drone's transmission power as the minimum power to satisfy the QoS requirement in Eq. (9).

In our simulations, we consider a 1000 m × 1000 m area, where the IoT GW is located at the center of the area. 30 drones are randomly distributed in the area to collect information from the ground. The height of the flying plane is 500 m, where all drones fly in the plane. The parameters α and β for calculating $Pr(LoS)$ are 9.6 and 0.28, respectively. The speed of light c is 3×10^8 *m/s*. The carrier frequency f_c is 2 *GHz*. The parameters ξ_{LoS} and ξ_{NLoS} for calculating the path losses PL_{LoS} and PL_{NLoS} are 1 and 20 *dB*, respectively. Note that the above drone-related parameters are inspired by [20]. The system bandwidth W is 10 *MHz* and the noise power density $N_0 = -174$ *dBm/Hz*. The data size collected by drones are randomly distributed from 1.0 to 4.0 *Mb*. The drone's maximum transmission power $P_m = 3$ *W*. The QoS requirement of each drone is 100 *Mbps*. The battery capacity of each drone $B^{max} = 100$ *Joule*. The energy harvesting efficiency is ρ is 20% [5]. In Actor-critic, both the actor and critic's deep neural networks contain 2 hidden layers and 64 nodes for each layer.

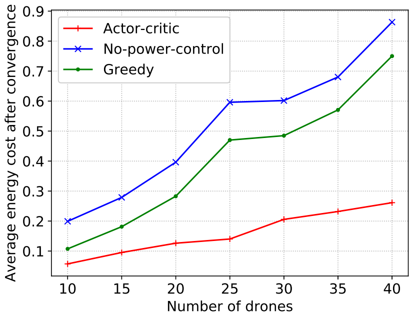


Fig. 2. Average energy cost after convergence vs number of drones.

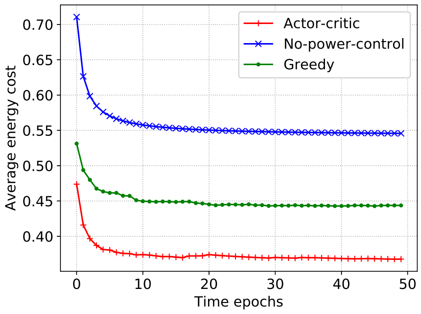


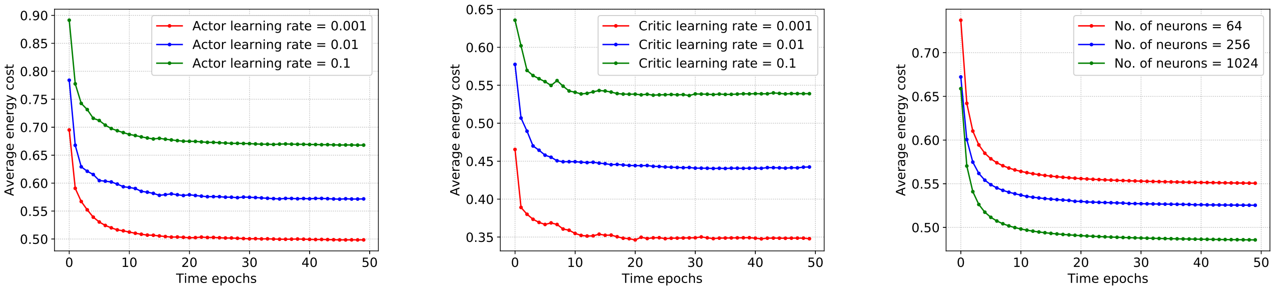
Fig. 3. Average energy cost vs time epochs.

Fig. 2 evaluates the average system energy cost after convergence with different numbers of drones ranging from 10 to 40. The average energy costs of all the three algorithms increase with the number of drones because deploying more drones implies that more energy may be consumed for wireless data transmissions and battery charging. Actor-critic generates the least average energy cost as compared with No-power-control and Greedy. Actor-critic performs better than No-power-control because it adjusts drones' wireless transmission power and hence helps reduce the energy consumption. Actor-critic generates lower energy cost than Greedy because it utilizes the past experiences to train the neural networks and hence improves its performance. Moreover, Greedy performs better than No-power-control because it optimizes the power control policy to minimize the energy cost for each time epoch.

Fig. 3 compares the average energy costs of Actor-critic, No-power-control and Greedy within the first 50 time epochs. All the three algorithms converge after several steps. Among the three algorithms, Actor-critic generates the least average energy cost, the next is Greedy, and No-power-control the most for the similar reasons in Fig. 2.

We then explore impacts of different parameters on the Actor-critic's performance in Fig. 4. Fig. 4(a) investigates the impact of different actor learning rates on average cost.

We compare the three values of actor learning rates including 0.001, 0.01 and 0.1. We can observe in Fig. 4(a) that a smaller actor learning rate achieves better performance because a larger actor learning rate may result in local optimum. Fig. 4(b) illustrates the Actor-critic's average energy costs with different critic learning rates including 0.001, 0.01 and 0.1. Similar to Fig. 4(a), a smaller critic learning rate generates less average energy cost. Fig. 4(c) compares the Actor-critic's average energy costs with three different numbers of neurons including 64, 256 and 1024. Increasing the number of neurons improves the complexity and hence the accuracy of neural networks. Hence, Actor-critic with 1024 neurons achieves the least average energy cost.



(a) different actor learning rates

(b) different critic learning rates

(c) different numbers of neurons

Fig. 4. Average energy cost vs time epochs with different parameters.

VII. Conclusion

In this paper, we have investigated the power control in time-varying IoD networks with wireless charging for the data collection service. We have tried to optimize the wireless transmission power of each drone at each time epoch to minimize the system energy consumption. An MDP model has been formulated to characterize the time-varying IoD network status. Then, a deep actor-critic reinforcement learning algorithm has been designed to obtain the power control policy. We have demonstrated by simulations that our designed algorithm performs better than the existing algorithms, and the performances of our algorithm are affected by actor and critic learning rates as well as the number of neurons.

References

[1] J. Yao and N. Ansari, "Fog resource provisioning in reliability-aware IoT networks," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8262–8269, Oct. 2019.

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

[3] J. Yao and N. Ansari, "Joint content placement and storage allocation in C-RANs for IoT sensing service," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 1060–1067, Feb. 2019.

[4] ——, "QoS-aware joint BBU-RRH mapping and user association in Cloud-RANs," *IEEE Transactions on Green Communications and Networking*, vol. 2, no. 4, pp. 881–889, Dec. 2018.

[5] N. Kawashima and K. Takeda, "Laser energy transmission for a wireless energy supply to robots," in *Robotics and Automation in Construction*. IntechOpen, 2008.

[6] L. Zhang and N. Ansari, "A framework for 5G networks with in-band full-duplex enabled drone-mounted base-stations," *IEEE Wireless Communications*, vol. 26, no. 5, pp. 121–127, Oct. 2019.

[7] N. Ansari, Q. Fan, X. Sun, and L. Zhang, "Soarnet," *IEEE Wireless Communications*, vol. 26, no. 6, pp. 37–43, Dec. 2019.

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

[9] B. Kiumarsi, K. G. Vamvoudakis, H. Modares, and F. L. Lewis, "Optimal and autonomous control using reinforcement learning: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 6, pp. 2042–2062, June 2018.

[10] J. Yao and N. Ansari, "Task allocation in fog-aided mobile IoT by lyapunov online reinforcement learning," *IEEE Transactions on Green Communications and Networking*, 2019, doi: 10.1109/TGCN.2019.2956626, early access.

[11] I. Grondman, L. Busoniu, G. A. D. Lopes, and R. Babuska, "A survey of actor-critic reinforcement learning: Standard and natural policy gradients," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 1291–1307, Nov. 2012.

[12] J. Yao and N. Ansari, "QoS-aware rechargeable UAV trajectory optimization for sensing service," in *IEEE International Conference on Communications (ICC)*, Shanghai, May 20-24, 2019.

[13] Y. Chen and L. Wang, "Privacy protection for internet of drones: A network coding approach," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1719–1730, Apr. 2019.

[14] X. Chen, J. Wu, Y. Cai, H. Zhang, and T. Chen, "Energy-efficiency oriented traffic offloading in wireless networks: A brief survey and a learning approach for heterogeneous cellular networks," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 4, pp. 627–640, Apr. 2015.

[15] Z. Zhang, F. R. Yu, F. Fu, Q. Yan, and Z. Wang, "Joint offloading and resource allocation in mobile edge computing systems: An actor-critic approach," in *2018 IEEE Global Communications Conference (GLOBECOM)*, Dec. 2018, pp. 1–6.

[16] Y. Wei, F. R. Yu, M. Song, and Z. Han, "User scheduling and resource allocation in hetnets with hybrid energy supply: An actor-critic reinforcement learning approach," *IEEE Transactions on Wireless Communications*, vol. 17, no. 1, pp. 680–692, Jan. 2018.

[17] J. Yao and N. Ansari, "QoS-aware power control in internet of drones for data collection service," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 7, pp. 6649–6656, Jul. 2019.

[18] N. Ansari and X. Sun, "Mobile edge computing empowers internet of things," *IEICE Transactions on Communications, (Invited Paper)*, vol. E101-B, no. 3, pp. 604–619, Mar. 2018.

[19] M. Lu, M. Bagheri, A. P. James, and T. Phung, "Wireless charging techniques for UAVs: A review, reconceptualization, and extension," *IEEE Access*, vol. 6, pp. 29 865–29 884, 2018.

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

[21] A. Al-Hourani, S. Kandeepan, and A. Jamalipour, "Modeling air-to-ground path loss for low altitude platforms in urban environments," in *Proc. IEEE Global Communications Conference (GLOBECOM) 2014*, Austin, Texas, USA, Dec. 2014, pp. 2898–2904.

[22] 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.

[23] 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.

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

[25] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Proceedings of the 12th International Conference on Neural Information Processing Systems*, ser. NIPS'99. Cambridge, MA, USA: MIT Press, 1999, pp. 1057–1063. [Online]. Available: http://dl.acm.org/citation.cfm?id=3009657.3009806

[26] 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.

[27] J. N. Tsitsiklis and B. Van Roy, "An analysis of temporal-difference learning with function approximation," *IEEE Transactions on Automatic Control*, vol. 42, no. 5, pp. 674–690, May 1997.

[28] Z. Zhou, J. Feng, B. Gu, B. Ai, S. Mumtaz, J. Rodriguez, and M. Guizani, "When mobile crowd sensing meets UAV: Energy-efficient task assignment and route planning," *IEEE Transactions on Communications*, vol. 66, no. 11, pp. 5526–5538, Nov. 2018.