Deep Reinforcement Learning for Fresh Data Collection in UAV-assisted IoT Networks

Mengjie Yi*†, Xijun Wang‡§, Juan Liu¶, Yan Zhang*†, and Bo Bai‖
*State Key Lab of Integrated Service Networks,
Information Science Institute, Xidian University, Xi'an, Shaanxi, 710071, China
†Science and Technology on Communication Network Laboratory, Shijiazhuang, Hebei, 050081, China
‡School of Electronics and Communication Engineering, Sun Yat-sen University, Guangzhou, 510006, China
§Key Laboratory of Wireless Sensor Network & Communication,
Shanghai Institute of Microsystem and Information Technology,
Chinese Academy of Sciences, 865 Changning Road, Shanghai 200050 China
¶School of Electrical Engineering and Computer Science, Ningbo University, Zhejiang 315211, China
‖Theory Lab (FKA Future Network Theory Lab), 2012 Labs, Huawei Technologies Co., Ltd., Hong Kong
Email: mjyi@stu.xidian.edu.cn, wangxijun@mail.sysu.edu.cn, eeliujuan@gmail.com,
yanzhang@xidian.edu.cn, ee.bobbai@gmail.com

***Abstract*—Due to the flexibility and low operational cost, dispatching unmanned aerial vehicles (UAVs) to collect information from distributed sensors is expected to be a promising solution in Internet of Things (IoT), especially for time-critical applications. How to maintain the information freshness is a challenging issue. In this paper, we investigate the fresh data collection problem in UAV-assisted IoT networks. Particularly, the UAV flies towards the sensors to collect status update packets within a given duration while maintaining a non-negative residual energy. We formulate a Markov Decision Process (MDP) to find the optimal flight trajectory of the UAV and transmission scheduling of the sensors that minimizes the weighted sum of the age of information (AoI). A UAV-assisted data collection algorithm based on deep reinforcement learning (DRL) is further proposed to overcome the curse of dimensionality. Extensive simulation results demonstrate that the proposed DRL-based algorithm can significantly reduce the weighted sum of the AoI compared to other baseline algorithms.**

I. Introduction

Owing to the fully controllable mobility and low operational cost, unmanned aerial vehicles (UAVs) emerge as promising technologies to provide wireless services [1]. One of the most important applications is to collect information from distributed sensors with the help of UAV in the Internet of Things (IoT). Since the UAV can fly close to each sensor and exploit the line-of-sight (LoS) dominant air-to-ground channel, the transmission energy of the sensors can be greatly reduced and the throughput of sensors can be significantly improved. Such advantages make UAV-assisted IoT networks attract extensive attention in recent years and arouse many research interests, ranging from the designs of UAV's flight trajectory to resource allocation, and sensors' wakeup schedule [2]–[6]. However, most of the existing works aimed at either maximizing system throughput or minimizing delay. Recently, the age of information (AoI) has been introduced to measure data freshness in IoT networks [7]–[9]. Particularly, AoI tracks the time elapsed since the latest received packet at the destination was generated at the source. In contrast to throughput and delay, the AoI metric is defined from the receiver's perspective. Therefore, previous results in the literature can not be directly used to minimize the AoI in UAV-assisted IoT networks.

There have been some recent efforts on guaranteeing data freshness in UAV-aided data collection for IoT networks. In [10], the UAV was used as a mobile relay for a source-destination pair and the trajectory is designed to minimize the average Peak AoI. In an IoT network with multiple sensors, two age-optimal trajectory planning algorithms were proposed in [11], where the UAV flies to and hovers above each sensor to collect data. This work was then extended in [12], where the UAV collects data from a set of sensors when hovering at each collection point (CP). The sensor-CP association and the UAV's flight trajectory were jointly designed to minimize the maximum AoI of the sensors. In a similar setup, an AoI deadline was imposed on each sensor and the UAV's flight trajectory was designed to minimize the number of expired packets in [13]. In these works, however, the UAV collects the data of each sensor only once and then flies back to the depot. To continuously collect data packets during a period of time, the authors of [14] optimized both the UAV's flight trajectory and the transmission scheduling of sensors to achieve the minimum weighted sum of AoI. Nonetheless, the energy consumption of the UAV has not been considered in the design of UAV's age-optimal trajectory.

In this paper, by taking the energy constraint of UAV into consideration, we study the age-optimal data collection problem in UAV-assisted IoT networks based on deep rein-

This work was supported in part by the National Natural Science Foundation of China (61971249), by Fundamental Research Funds for the Central Universities under grant 19lgpy79, by the Research Fund of the Key Laboratory of Wireless Sensor Network & Communication (Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences) under grant 20190912, and by Young Stars in Science and Technology of Shaanxi Province (2019KJXX-030).

forcement learning (DRL). In particular, a UAV is dispatched from a depot, flies towards the sensors to collect status update packets, and arrives at the destination within a given duration. The UAV has to maintain a non-negative residual energy while minimizing the weighted sum of the AoI of sensors during the flight. To find the optimal flight trajectory of the UAV and transmission scheduling of the sensors, we formulate this problem into a finite-horizon Markov decision process (MDP). Due to the high-dimensional state space, it is computationally prohibitive to solve the MDP problem using dynamic programming algorithms. To address this issue, we propose a DRL-based UAV-assisted data collection algorithm, where the UAV decides which direction to fly and which sensor to connect at each step. Extensive simulation results demonstrate that the proposed algorithm can significantly reduce the weighted sum of AoI compared to other baseline policies.

The rest of this paper is organized as follows: The system model and problem formulation are described in Section II. Section III provides the MDP formulation of the problem and presents the proposed DRL-based algorithm. The simulation results and discussions are given in Section IV. Finally, we conclude this paper in Section V.

II. System Model and Problem Formulation

A. Network Description

As shown in Fig. 1, we consider a UAV-assisted IoT network, where N sensor nodes (SNs) are randomly distributed in a certain geographical region. The set of all the SNs is denoted by $\mathcal{N} = \{1, 2, \ldots, N\}$ and the location of each SN is represented by $w_n = (x_n, y_n)$ for $n \in \mathcal{N}$. The region of interest is equally partitioned into a number of small-size grids such that the UAV's location is approximately constant within each grid. Moreover, the center of the l-th grid is represented by $c_l = (x_l, y_l)$. We denote by $\mathcal{C}$ the set containing the locations of centers for all the grids. Moreover, the spacing distance between the centers of any two adjacent grids is denoted by L'.

We assume a discrete-time system where time is divided into equal-length time slots. The length of each slot is τ seconds. Given a time duration of T slots, the rotary-wing UAV takes off from an initial location c_{start} and flies over N SNs to collect data packets. At the end of the T-th slot, the UAV lands on a final destination c_{stop}. We assume that the UAV flies along the center of the grids at a fixed altitude h. In each time slot, the UAV could hover over a certain grid or fly across one grid at a constant speed V. Let o_t denote the projection of the UAV's location on the ground at time slot t. Then, the projection of the UAV's flight trajectory is defined as a sequence of center of grids $\boldsymbol{p} = (o_1, o_2, \ldots, o_T)$, where $o_1 = c_{\text{start}}$ and $o_T = c_{\text{stop}}$.

Let $E_{\max}$ denote the initial amount of energy the UAV carries. The energy consumption of the UAV consists of the communication energy and the propulsion energy. Since the communication energy consumption is relatively small, we consider only the propulsion energy consumption in this paper. The propulsion energy of the rotary-wing UAV is mainly composed of the blade profile energy, the induced power, and the parasite energy [3]. The propulsion power consumption can be expressed as follows,

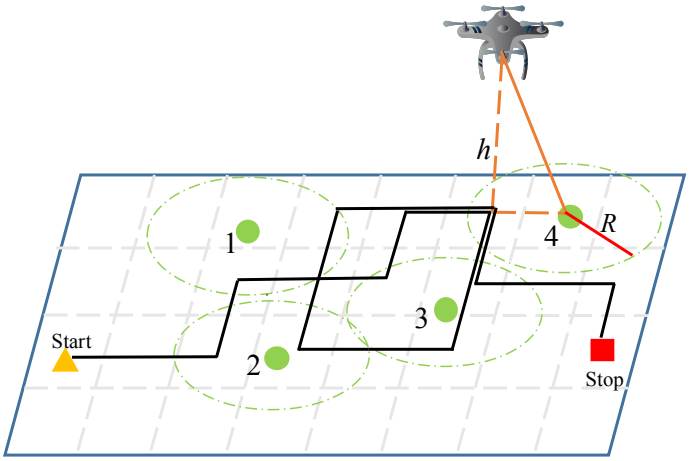

Figure 1. An illustration of the UAV-assisted data collection.

$$\tilde{P}(V_t) = P_0\left(1 + \frac{3V_t^2}{U_{tip}^2}\right) + P_1\left(\sqrt{1 + \frac{V_t^4}{4v_0^4}} - \frac{V_t^2}{2v_0^2}\right)^{\frac{1}{2}} + \frac{1}{2}d_0\rho s_0 A V_t^3, \tag{1}$$

where P_0 and P_1 represent the blade profile power and derived power of the UAV in the hovering state, respectively, V_t is the velocity of the UAV at slot t, U_{tip} represents the tip speed of the rotor blade of the UAV, v_0 represents the mean rotor induced velocity in the hovering state, d_0 is the fuselage drag ratio, ρ represents the density of air, s_0 indicates the rotor solidity, and A represents the area of the rotor disk. In particular, the power consumption when hovering (i.e., $V_t = 0$) is $\tilde{P}(0) = P_0 + P_1$.

We assume that the UAV could establish the LoS links with the SNs due to its high attitude. Then, the channel power gain from the SN to the UAV at time slot t can be given by

$$g_{n,u}(t) = \beta_0 d_{n,u}^{-2}(t) = \frac{\beta_0}{\| o_t - w_n \|^2 + h^2}, \tag{2}$$

where β_0 is the channel gain at a reference distance of 1 meter, $d_{n,u}(t)$ denotes the Euclidean distance between the SN n and the UAV at time slot t. Let P denote the transmission power of each SN. When the UAV is within the coverage of one SN, i.e., $\| o_t - w_n \| \leq R$, the SN generates a status update of size M and sends it to the UAV successfully in a time slot. Specifically, the coverage radius can be calculated as

$$R = \left(\frac{\beta_0 P}{(2^{\frac{M}{B\tau}} - 1)\sigma^2} - h^2\right)^{\frac{1}{2}}, \tag{3}$$

where B is the channel bandwidth, and σ^2 is the noise power at the UAV.

We employ AoI to measure the freshness of information. In particular, the AoI is defined as the time elapsed since the

generation of the latest status update received by the UAV. Let $U_n(t)$ denote the time at which the latest status update of SN n successfully received by the UAV was generated. The AoI of SN n at the beginning of slot t is then given by

$$\delta_{n,t} = t - U_n(t). \tag{4}$$

Let $\boldsymbol{b} = (b_1, b_2, \ldots, b_T)$ be the vector of the SNs' scheduling variables, where $b_t \in \mathcal{B} \triangleq \{0, 1, \ldots, N\}$ denotes which SN is scheduled to update its status at time slot t. In particular, $b_t = n$ indicates that SN n transmits to the UAV at slot t and $b_t = 0$ means that no transmission occurs at slot t. According to (4), if SN n is scheduled to transmit at slot t and the UAV is located in the coverage of SN n, then its AoI decreases to one; otherwise, the AoI increases by one. Then, the dynamics of the AoI can be given by

$$\delta_{n,t+1} = \begin{cases} 1, & \text{if } b_t = n \text{ and } \parallel o_t - w_n \parallel \leq R; \\ \delta_{n,t} + 1, & \text{otherwise.} \end{cases} \tag{5}$$

B. Problem Formulation

Our objective is to find the optimal trajectory of the UAV and the optimal scheduling of the SNs that minimize the weighted average AoI of all the SNs. The optimization problem can be expressed as follows:

$$\text{P1: } \min_{\boldsymbol{p}, \boldsymbol{b}} \frac{1}{T} \sum_{t=1}^{T} \sum_{n=1}^{N} \theta_n \delta_{n,t}, \tag{6}$$

$$s.t. \sum_{t=1}^{T} \tilde{P}(V_t)\tau \leq E_{\max}, \tag{7}$$

$$o_1 = c_{\text{start}}, \tag{8}$$

$$o_T = c_{\text{stop}}. \tag{9}$$

where θ_n denotes the importance of SN n. (7) ensures that the UAV will not run out of the energy before time slot T. (8) and (9) guarantee that the UAV starts from the initial location and arrives at the final location at time slot T. It is easily observed that the above optimization problem is a nonlinear integer programming one, which is computationally complex to solve for large-scale networks. In the following section, we propose a learning based algorithm for the UAV to learn its trajectory and the SNs' transmission schedule at each location along the trajectory.

III. DRL-based Approach

In this section, we first cast the UAV-assisted data collection problem into a Markov decision process (MDP) and then propose a DRL-based algorithm to minimize the weighted average AoI of all the SNs.

A. MDP Formulation

We reformulate the problem P1 via an MDP, which is usually represented by a tuple (s, a, r, p). Here, s presents the state, a denotes the action, r is the reward function, and p is the state transition probability. MDP is commonly used to model a sequential decision-making process. In particular, at time slot t, the agent observes some state s_t and performs an action a_t. After taking this action, the state of the environment transits to s_{t+1} with probability $p_{s_t, s_{t+1}}$, and the agent receives a reward r_t. We consider the UAV as the agent for performing the data collection algorithm and define the state, action, and reward function in the following.

1) State: The state at time slot t is defined as $s_t = (o_t, \boldsymbol{\delta}_t, \phi_t, \Delta_t)$, which is composed of four parts:

- $o_t \in \mathcal{C}$ is the projection of the UAV on the ground at time slot t.
- $\boldsymbol{\delta}_t = (\delta_{1,t}, \delta_{2,t}, \ldots, \delta_{N,t})$ is the AoI of all the SNs at the UAV at time slot t. For the AoI of each SN, we have $\delta_{n,t} \in \mathcal{D} \triangleq \{1, 2, \ldots, \delta_{\max}\}$, where $\delta_{\max}$ is the maximum value of AoI and can be chosen to be arbitrary large.
- $\phi_t \in \mathcal{T} \triangleq \{0, 1, \ldots, T\}$ is the difference between the remaining time of the UAV and the minimum time required to reach the final destination.
- $\Delta_t = E_{\max} - \sum_{m=1}^{t} \tilde{P}(V_t)\tau - \sum_{m=1}^{N-t} \tilde{P}(V)\tau$ is the difference between the remaining energy of the UAV and the energy required for the UAV to arrive at the final destination in the remaining time. $\Delta_t \in \mathcal{E}$, where $\mathcal{E}$ is the set of the energy level of the UAV.

Altogether, the state space of the system can be expressed as $\mathcal{S} = \mathcal{C} \times \mathcal{D}^N \times \mathcal{T} \times \mathcal{E}$.

2) Action: The action of the UAV at time slot t is characterized by its movement υ_t and the scheduling of SN b_t, i.e., $a_t = (\upsilon_t, b_t)$. In each time slot, the UAV either hovers at its current location or move to one of its adjacent cells. Specifically, $\upsilon_t \in \mathcal{V} \triangleq \{\text{North}, \text{South}, \text{East}, \text{West}, \text{Hovering}\}$. Then, the action space is given by $\mathcal{A} = \mathcal{V} \times \mathcal{B}$.

3) Reward: In our context of the UAV-assisted data collection, the reward should encourage the UAV to minimize the weighted average AoI of all the SNs under the constraints given by (7)-(9). When the UAV reaches the final destination at time slot T with a non-negative residual energy, we will give the UAV an additional reward. However, a punishment will be imposed when the constraints are violated. Let $J = \frac{1}{T} \sum_{t=1}^{T} \sum_{n=1}^{N} \theta_n \delta_{n,t}$. Then, the reward is defined as follows,

$$r_t = \begin{cases} -J - k_1, & \text{if } \phi_t < 0, \\ -J - k_2, & \text{if } \Delta_t < 0, \\ -J + k_3, & \text{if } o_T = c_{\text{stop}}, \Delta_t \geq 0, \\ -J, & \text{otherwise.} \end{cases} \tag{10}$$

where k_1, k_2, and k_3 are positive constants and set large enough.

4) State Transition: The AoI of each SN is updated as in (5). The dynamics of the UAV's location can be expressed as

$$o_{t+1} = \begin{cases} o_t + (0, L'), & \text{if } \upsilon_t = \text{North}, \\ o_t - (0, L'), & \text{if } \upsilon_t = \text{South}, \\ o_t + (L', 0), & \text{if } \upsilon_t = \text{East}, \\ o_t - (L', 0), & \text{if } \upsilon_t = \text{West}, \\ o_t, & \text{if } \upsilon_t = \text{Hovering}. \end{cases} \tag{11}$$

The time difference ϕ_t is updated based on the UAV's location. In particular, if the UAV flies towards the final

destination at slot t, ϕ_{t+1} remains the same as ϕ_t. If the UAV hovers at slot t, ϕ_{t+1} is decreased by one. While ϕ_{t+1} is decreased by two, if the UAV flies away from the final destination. Altogether, we can update ϕ_t as follows,

$$\phi_{t+1} = \begin{cases} \phi_t, & \text{if } \| o_t - c_{\text{stop}} \| > \| o_{t+1} - c_{\text{stop}} \|, \\ \phi_t - 1, & \text{if } \| o_t - c_{\text{stop}} \| = \| o_{t+1} - c_{\text{stop}} \|, \\ \phi_t - 2, & \text{if } \| o_t - c_{\text{stop}} \| < \| o_{t+1} - c_{\text{stop}} \|. \end{cases} \quad (12)$$

Since the power consumptions for hovering and flying are different, the update of energy difference Δ_t is different for these two cases. According to the definition of the energy difference Δ_t, the update of Δ_t can be given by

$$\Delta_{t+1} = \begin{cases} \Delta_t + \tilde{P}(V) - \tilde{P}(0), & \text{if } \upsilon_t = \text{Hovering}, \\ \Delta_t & \text{otherwise.} \end{cases} \quad (13)$$

Our goal is to find an age-optimal policy π^*, which determines the sequential actions over a finite horizon of length T. Given a policy π, the total expected reward of the system starting from an initial state s_1 is defined as

$$G_\pi = \sum_{t=1}^{T} \mathbb{E}_\pi \left[r_t \mid s_1 \right]. \quad (14)$$

Then, the optimal policy can be obtained by maximizing the total expected reward, i.e., $\pi^* = \arg\max_\pi G_\pi$. When the number of SNs become large, it is computationally infeasible to find the optimal strategy by standard dynamic programming method. Therefore, DRL is employed in the following subsection to solve this problem.

B. DRL Approach

We employ DQN, which is one of the most well adopted DRL method, to derive the optimal policy. In this approach, we define a state-action value function $Q_\pi(s, a)$, which represents the expected reward for selecting action a in state s and then following policy π. The optimal Q-value function can be estimated by the update

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha \left[r_t + \max_a Q(s_{t+1}, a) - Q(s_t, a_t) \right], \quad (15)$$

where α is the learning rate. The optimal policy is the one that takes the action which maximizes the Q-value function at each step.

By incorporating deep neural network (DNN) into the framework of Q-learning, DQN can overcome the curse of dimensionality. In particular, we use a DNN with weights θ to approximate the Q-value function $Q(s, a)$ with $Q(s, a; \theta)$. The DNN can be trained by minimizing a sequence of loss function $L(\theta_t)$ that changes at each slot t. Specifically,

$$L(\theta_t) = \left(r_t + \max_a Q(s_{t+1}, a; \theta_{t-1}) - Q(s_t, a_t; \theta_t) \right)^2, \quad (16)$$

where the weights are updated at slot t and the weight θ_{t-1} from the previous slot are held fixed. However, the use of one DNN may induce instability. In order to overcome this issue, two neural networks are employed [15], i.e., the current network with weights θ and the target network parameterized by θ^-. The current network is used as a function approximator and its weights are updated at every slot. While the target network computes the target Q-value function and its weights are fixed for a while and updated at every O steps (Lines 11~18). In particular, the weights of the DNN are updated by minimizing the loss function, which is defined as

$$L(\theta) = \left(r_m + \max_a Q(s_{m+1}, a; \theta^-) - Q(s_m, a_m; \theta) \right)^2, \quad (17)$$

where $Q(s_m, a_m; \theta)$ is evaluated by the current network and $Q(s_{m+1}, a; \theta^-)$ is evaluated by the target network. Based on this, the update formula for weights θ is given as follows:

$$\theta = \theta + \alpha [y_m - Q(s_m, a_m; \theta)] \nabla_\theta Q(s_m, a_m; \theta), \quad (18)$$

where $y_m = r_m + \max_a Q(s_{m+1}, a; \theta^-)$ and ∇_θ denotes the gradient with respect to θ.

Based on the DQN with two neural networks, the UAV-assisted data collection algorithm is proposed to find the optimal solution to problem P1, and the details are showed in Algorithm 1. At the beginning of the training process, the

Algorithm 1 DRL-based UAV-assisted data collection algorithm

1: Initialize the replay memory D, the probability ϵ, the current network parameter θ, and the target network parameter $\theta^- = \theta$;
2: Initialize the current network $Q(s, a; \theta)$ with weights θ and the target network $Q(s, a; \theta^-)$ with weights θ^-;
3: **for** $episode = 1 : E$ **do**
4: Initialize the environment and observe an initial state s_1;
5: **for** $t = 1 : T$ **do**
6: Select a random action a_t with probability ϵ;
7: Otherwise select $a_t = \arg\max_a Q(s_t, a; \theta)$;
8: Execute action a_t and observe the reward r_t and the next state s_{t+1};
9: Mark s_{n+1} if it is a terminal state and store transition (s_t, a_t, r_t, s_{t+1}) in the replay memory;
10: Sample a random mini-batch of transitions (s_m, a_m, r_m, s_{m+1}) from the replay memory;
11: Calculate the target value y_m:
12: **if** s_{m+1} is the terminal state **then**
13: $y_m = r_m$;
14: **else**
15: $y_m = r_m + \max_a Q(s_{m+1}, a; \theta^-)$;
16: **end if**
17: Update the current network by performing the gradient descent in (18);
18: Update target parameters, $\theta^- = \theta$, in every O steps;
19: Terminate the episode if s_{m+1} is the terminal state.
20: **end for**
21: **end for**

estimation of the Q-value function is far from accurate. Hence, the UAV should explore the environment more often at first. When the policy continues improving and the knowledge of the environment is more accurate, the UAV should exploit the learned knowledge more often. As such, we utilize a simple ϵ-greedy policy (Lines 6~7). In particular, the action is randomly selected to explore the environment with probability ϵ and the action that maximizes $Q(s_t, a; \theta)$ is chosen to exploit the policy with probability $1 - \epsilon$. Moreover, ϵ is set to be decreasing with the number of slots so that the UAV can choose the optimal action when the estimation of Q-value function converges.

Experience replay is used in the learning process. The agent stores the experience (s_t, a_t, r_t, s_{t+1}) in the replay memory, and then samples a mini-batch of the experiences from the replay memory uniformly at random to train the neural network (Lines 9~10). By using experience replay, not only the correlation among the continuous samples is reduced, but also the utilization rate of the experience data can be improved. We also note that the UAV-assisted data collection problem we considered is episodic, since the UAV is required to be arrive in the final destination at time slot T. In particular, there are three terminal cases: 1) when the UAV reaches the final destination at time slot T, 2) when $\phi_t < 0$, and 3) when $\Delta_t < 0$ (Line 19).

IV. Simulation Results

In this section, we perform extensive simulations to evaluate the performance of the DRL-based UAV-assisted data collection algorithm in an IoT network. We consider a square area of 500 m $\times$ 500 m that is virtually divided into 20×20 equally-sized grids of length 25 m. Let the center of the left lower grid of the square region be the origin with coordinate $[0, 0]$ and the index of every grid is the coordinate of the grid center divided by 25. For instance, the left lower grid is indexed by $(0, 0)$. We assume that UAV's initial and final locations are at grids $(10, 0)$ and $(10, 19)$, respectively. We also assume that the SNs have equal importance weights. Unless otherwise specified, the simulation parameters are presented in Table I.

The two neural networks in the proposed algorithm is implemented using Tensorflow. In particular, each DNN includes two fully-connected hidden layers with 200 and 256 neurons. The input layer size of the DNN is the same as the state space size and the output layer size of the DNN is equal to the total number of actions. The hypeparameters of DQN are summarized in Table II.

In the following figures, we compare the performance of the proposed algorithm with two baseline algorithms, namely AoI-based algorithm and distance-based algorithm. In the AoI-based algorithm, the UAV flies to the SN with the largest AoI in the current time slot. While in the distance-based algorithm, the flight trajectory of the UAV is divided into multiple rounds. In each round, the UAV traverses all the SNs one by one and the UAV flies to the nearest and unvisited SN in the current traversal round. Moreover, the UAV can collect status update from the SNs on its way in both baseline algorithms. When the UAV's residual energy or the remaining time is less than a threshold, it directly flies to the final destination.

Table I
System parameters

Parameter	Value
Channel bandwidth B	1 MHz
Update size M	5 Mbits
Noise power σ^2	-100 dbm
Channel gain at 1 m β_0	-60 dB
Flight altitude h	120 m
Time duration T	70 slots
UAV speed V	25 m/s
Initial energy $E_{\max}$	2.2e4 J
Air density in ρ	1.225 kg/m^3
Tip speed U_{tip}	120 m/s
Blade profile power P_0	99.66 W
Derived power P_1	120.16 W
Body resistance ratio d_0	0.48
Robustness of the rotor s	0.0001
The area of the rotor disk A	0.5 s^2
Mean rotor induced velocity in hover v_0	0.002 m/s

Table II
Hyperparameters of DQN

Parameter	Value
Episodes E	20000
Reply memory size D	40000
Mini-batch size	200
Initial ϵ	0.9
ϵ-greedy decrement	0.0001
Minimum ϵ	0
Learning rate δ	0.002
Learning rate decay rate	0.95
Learning rate decay step	10000
Update step O	300
Optimizer	Adam
Activation function	ReLU

Fig. 2 illustrates the average AoI with respect to the coverage radius R in a scenario with three randomly deployed SNs, where the value of R is normalized by the length of a grid. From Fig. 2, we can see that a higher R results in lower average AoI since it takes less time for the UAV to fly to collect data packets. Moreover, we can see that our proposed DQN-based algorithm outperforms the two baseline algorithms since it jointly considers the AoI, the location of the UAV, and the time and energy constraints. It is also shown that the AoI-based algorithm achieves almost the same performance as DQN-based algorithm when R is large. This is because there is an overlap of the coverage of all the SNs for a larger R and the UAV can fly above the overlapping area to collect data packets.

Fig.3 shows the average AoI with respect to the number of sensors N for $R = 4$. We can easily observe that by adopting our DQN-based algorithm, the average AoI is smaller than that of the baseline algorithms. Moreover, the reduction of the average AoI is more significant for a larger N. Fig. 3 also shows that the average AoI increases with the number of SNs. This is because, for a larger N, the UAV has to fly farther to collect update packets. In addition, the SNs have to wait for a

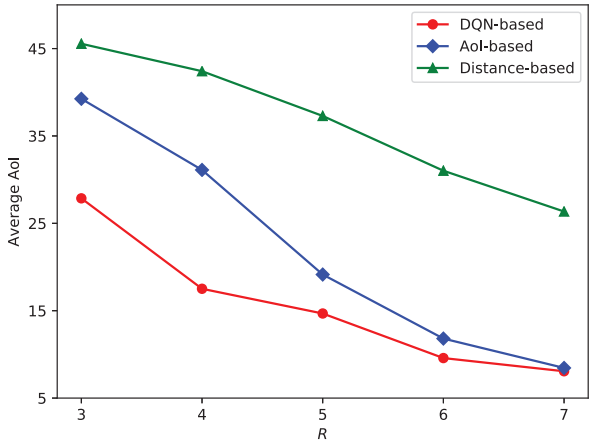

Figure 2. Effect of R on the average AoI with $N = 3$.

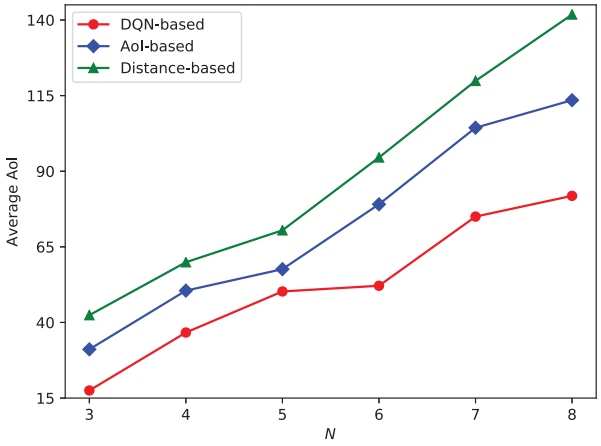

Figure 3. Effect of N on the average AoI with $R = 4$.

longer time to update their status, since the UAV can collect data packets from only one SN each time.

V. Conclusions

In this paper, we have investigated the AoI-optimal data collection problem in UAV-assisted IoT networks, where a UAV collects status update packets and arrives at the final destination under both time and energy constraints. In order to minimize the wighted sum of the AoI, we have formulated the problem as a finite-horizon MDP. We have then designed a DRL-based data collection algorithm to find the optimal flight trajectory of the UAV and the transmission scheduling of the SNs. Moreover, we have conducted extensive simulations and shown that the DRL-based algorithm is superior to two baseline approaches, i.e., the AoI-based and the distance-based algorithms. Simulation results also demonstrated that the weighted sum of the AoI is monotonically decreasing with the SN's coverage radius and monotonically increasing with the number of SNs.

References

[1] M. Mozaffari, W. Saad, M. Bennis, Y.-H. Nam, and M. Debbah, "A Tutorial on UAVs for Wireless Networks: Applications, Challenges, and Open Problems," http://arxiv.org/abs/1803.00680, Mar. 2018.

[2] Y. Zeng and R. Zhang, "Energy-Efficient UAV Communication with Trajectory Optimization," *ArXiv160801828 Cs Math*, Aug. 2016.

[3] Y. Zeng, J. Xu, and R. Zhang, "Energy Minimization for Wireless Communication with Rotary-Wing UAV," *ArXiv180402238 Cs Math*, Apr. 2018.

[4] C. H. Liu, Z. Chen, J. Tang, J. Xu, and C. Piao, "Energy-Efficient UAV Control for Effective and Fair Communication Coverage: A Deep Reinforcement Learning Approach," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 2059–2070, Sep. 2018.

[5] J. Gong, T.-H. Chang, C. Shen, and X. Chen, "Flight Time Minimization of UAV for Data Collection over Wireless Sensor Networks," *ArXiv180102799 Cs Math*, Jan. 2018.

[6] U. Challita, W. Saad, and C. Bettstetter, "Deep Reinforcement Learning for Interference-Aware Path Planning of Cellular-Connected UAVs," in *2018 IEEE International Conference on Communications (ICC)*, May 2018, pp. 1–7.

[7] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?" in *Proc. IEEE INFOCOM*, Orlando, FL, USA, Mar. 2012, pp. 2731–2735.

[8] Y. Sun, E. Uysal-Biyikoglu, R. D. Yates, C. E. Koksal, and N. B. Shroff, "Update or Wait: How to Keep Your Data Fresh," *IEEE Trans. Inf. Theory*, vol. 63, no. 11, pp. 7492–7508, Nov. 2017.

[9] Z. Jiang, B. Krishnamachari, X. Zheng, S. Zhou, and Z. Niu, "Timely Status Update in Massive IoT Systems: Decentralized Scheduling for Wireless Uplinks," *ArXiv180103975 Cs Math*, Jan. 2018.

[10] M. A. Abd-Elmagid and H. S. Dhillon, "Average Peak Age-of-Information Minimization in UAV-assisted IoT Networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 2003–2008, 2019.

[11] J. Liu, X. Wang, B. Bai, and H. Dai, "Age-optimal trajectory planning for UAV-assisted data collection," in *Proc. IEEE INFOCOM WKSHPS*, Honolulu, HI, USA, Apr. 2018, pp. 553–558.

[12] P. Tong, J. Liu, X. Wang, B. Bai, and H. Dai, "UAV-Enabled Age-Optimal Data Collection in Wireless Sensor Networks," in *Proc. IEEE ICC Workshops*, Shanghai, CN, May 2019, pp. 1–6.

[13] W. Li, L. Wang, and A. Fei, "Minimizing Packet Expiration Loss With Path Planning in UAV-Assisted Data Sensing," *IEEE Wirel. Commun. Lett.*, vol. 8, no. 6, pp. 1520–1523, Dec. 2019.

[14] M. A. Abd-Elmagid, A. Ferdowsi, H. S. Dhillon, and W. Saad, "Deep Reinforcement Learning for Minimizing Age-of-Information in UAV-assisted Networks," in *Proc. IEEE Globecom*, Puako, HI, USA, May 2019.

[15] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.