Learning to Minimize Age of Information over an Unreliable Channel with Energy Harvesting

Elif Tuğçe Ceran, Deniz Gündüz and András György

Abstract

The time average expected age of information (AoI) is studied for status updates sent over an error-prone channel from an energy-harvesting transmitter with a finite-capacity battery. Energy cost of sensing new status updates is taken into account as well as the transmission energy cost better capturing practical systems. The optimal scheduling policy is first studied under the hybrid automatic repeat request (HARQ) protocol when the channel and energy harvesting statistics are known, and the existence of a threshold-based optimal policy is shown. For the case of unknown environments, average-cost reinforcement-learning algorithms are proposed that learn the system parameters and the status update policy in real-time. Value-based, policy-based, and deep reinforcement learning methods are exploited and the effectiveness of the proposed methods is demonstrated through numerical results.

Index Terms

Age of information, energy harvesting, hybrid automatic repeat request (HARQ), Markov decision process, reinforcement learning, policy gradient, deep Q-network (DQN).

I. INTRODUCTION

Many status update systems, including wireless sensor networks in Internet of things (IoT) applications, are powered by scavenging energy from renewable sources (e.g., solar cells [2], wind turbines [3], piezoelectric generators [4], etc.). Harvesting energy from ambient sources provides

environmentally-friendly and ubiquitous operation for remote sensing systems. Therefore, there has been a growing interest in maximizing the timeliness of information in energy harvesting (EH) communication systems [5]–[17]. In these systems, the staleness of the information at the receiver is measured by the age of information (AoI), defined as the time elapsed since the generation time of the most recent status update successfully received at the receiver.

Prior works have investigated online [5], [7], [9], [12], [13] as well as offline [5], [14] methods for different scenarios in order to optimize the timeliness of information under energy causality constraints in EH systems. The structure of an optimal policy is derived and heuristic algorithms are proposed in [7], [9], [11], [12] for a finite-size battery considering only the cost of transmissions. Earlier literature on AoI assumed that the cost of sensing (monitoring) the status of a process is negligible compared to the cost of transmitting the status updates. However, in many practical sensing systems acquiring a new sample of the underlying process of interest also has a considerable energy cost [18], [19]. The sampling/sensing cost has been taken into account in [19], [20] and [21], where a status update system with an unlimited energy source is considered. In [19] and [20], closed form expressions are presented for the energy consumption and average AoI with known transmission error probability and automatic repeat request (ARQ), assuming that a packet is re-transmitted until either it is received, or a prescribed maximum number of transmissions is reached. In our previous work, we studied status-update systems under a transmission-rate constraint, or equivalently, an average power constraint [22]–[25].

In this paper, we study a status update system considering both the sensing and transmission energy costs, better capturing practical systems. Moreover, we consider an EH transmitter, which uses the energy harvested from the environment to power the sensing and communication operations. Unlike prior work [19]–[21], we consider the intermittent availability of energy with unknown energy arrivals and a hybrid automatic repeat request (HARQ) protocol, where the partial information obtained from previous unsuccessful transmission attempts is combined to increase the probability of successful decoding. When employing HARQ in status update systems, there is an inherent trade-off between sending a new update after a failed transmission attempt, which may result in a lower AoI, and retransmitting the failed update, which has a lower probability of error. Introducing sensing cost to the system model makes this trade-off even more challenging and interesting, as retransmissions incur no sensing cost.

In most practical scenarios statistical information about either the energy arrival process or the channel conditions are not available a-priori, or may change over time [26]. Previous works

on EH communication systems without a-priori information on the random processes governing the system exploited reinforcement learning (RL) methods in order to maximize information freshness [12], [13], [16], [17]. [12], [16] and [17] consider deep RL (DRL) algorithms that can learn the status update policies for multiple source nodes powered by energy harvesting while [13] uses a tabular Q-learning method to find an optimal policy without information on the true battery levels of the source nodes. To the best of our knowledge, this is the first work that considers the HARQ protocol with also distinct sensing and transmission energy costs and proposes assorted RL algorithms, including both deep RL and tabular value or policy based RL methods, that can adapt the status-update scheme to the unknown energy arrival process as well as the channel statistics.

Our goal will be to identify the optimal policy that can judiciously balance the AoI benefits of transmitting a new status update with its additional sensing cost and lower success probability. The optimal decisions will depend not only on the current AoI and retransmission count, but also on the battery and energy harvesting states. We consider a value-based RL algorithm, *GR-learning* [27], a policy-based RL algorithm, *finite-difference policy gradient (FDPG)* [28], and a deep RL algorithm *deep Q-network (DQN)*, and compare their performances with that of the relative value iteration (RVI) algorithm which assumes a-priori knowledge on the system characteristics. We also propose a threshold policy with low computational complexity and demonstrate that a policy gradient algorithm exploiting the structural characteristics of a threshold policy outperforms the GR-learning algorithm. This paper is an extended version of work presented at [1]. We extend our previous work in several directions by showing that the threshold policy is optimal, by investigating the effect of preemption in different scenarios followed by proposing a novel RL method, i.e., FDPG with preemption, and also by exploiting a deep reinforcement learning algorithm.

The main contributions of this paper are outlined as follows:

- The average AoI is studied under energy replenishment constraints, i.e., energy causality as well as battery capacity constraints, imposed on the transmitter, which limit the energy consumption in a stochastic manner.
- The optimal policy is shown to be stationary, deterministic, and monotone with respect to the AoI, while both retransmissions and preemption following a failed transmission are considered.
- Scheduling algorithms are designed using multiple average-cost RL algorithms; in particular,
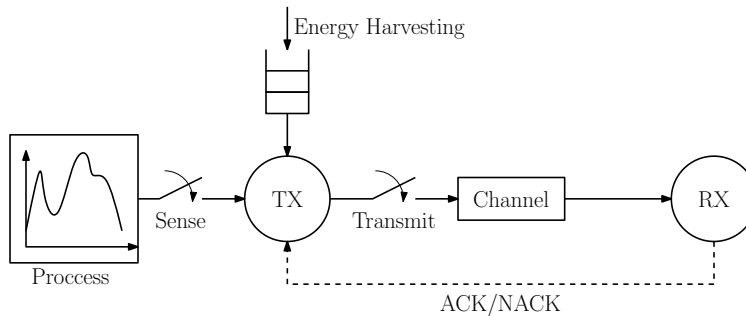
Figure 1. An EH status update system over an error-prone link in the presence of ACK/NACK feedback.

a value-based RL algorithm, *GR-learning* [27], a policy-based RL algorithm, *FDPG* [28], and a deep RL algorithm, *(DQN)*, are used to learn the optimal scheduling decisions when the transmission success probabilities and energy arrival statistics are unknown.

- Numerical simulations are conducted in order to investigate the effects of the EH process on the average AoI. In particular, we have found that temporal correlations in EH increase the average AoI significantly.

- The average AoI with EH is compared with the average AoI under an average transmission constraint, and it is shown that the average AoI obtained by the EH transmitter is similar to the one obtained under an average transmission constraint for a battery with unlimited capacity and zero sampling/sensing cost.

The rest of the paper is organized as follows. The system model is presented in Section II. The problem of minimizing the average AoI with HARQ under energy replenishment constraints is formulated as a Markov decision process (MDP) and the structure of the optimal policy is investigated in Section III. Section IV shows the application of RL algorithms to minimize the AoI in an unknown environment. Simulation results are presented in Section V, and the paper is concluded in Section VI.

II. System Model

We consider a time-slotted status update-system over an error-prone wireless communication link (see Fig. 1). The transmitter (TX) can sense the underlying time-varying process and generate a status update in each time slot at a certain energy cost. Status updates are communicated to the receiver (RX) over a time-varying wireless channel. Each transmission attempt takes constant time, which is assumed to be equal to the duration of one time slot.

The AoI measures the timeliness of the status information at the receiver, and is defined at any time slot t as the number of time slots elapsed since the generation time $U(t)$ of the most up-to-date packet successfully decoded at the receiver. Although the AoI can be arbitrarily large, in practice the utility of status updates typically becomes zero beyond a certain AoI, so the AoI can be assumed bounded by setting an upper bound on the age (which will be denoted by Δ_{max}.). Formally, the AoI at the receiver at time t is defined as $\Delta_t^{rx} \triangleq \min(t - U(t), \Delta_{max})$, where a maximum value Δ_{max} on the AoI is imposed to limit the impact of the AoI on the performance after some level of staleness is reached.

We assume that the channel changes randomly from one time slot to the next in an independent and identically distributed (i.i.d.) fashion, and the instantaneous channel state information is available only at the receiver. We further assume the availability of an error- and delay-free single-bit feedback from the receiver to the transmitter for each transmission attempt. Successful reception of the status update at the end of time slot t is acknowledged by an ACK signal (denoted by $K_t = 1$), while a NACK signal is sent in case of a failure (denoted by $K_t = 0$).

There are three possible actions the transmitter can take in each time slot t (the transmitter's action is denoted by A_t). It can either sample and transmit a new status update ($A_t = \text{n}$), remain idle ($A_t = \text{i}$), or retransmit the last failed status update ($A_t = \text{x}$). If an ACK is received at the transmitter, we can restrict the action space to $\{\text{i}, \text{n}\}$ as retransmitting an already decoded status update is strictly suboptimal. Also note that, even though the transmitter can just sense and generate a new update but not transmit it, this would be suboptimal, so we do not consider such an action separately.

We consider the HARQ protocol: that is, the received signals from previous transmission attempts for the same packet are combined for decoding. The probability of error using r retransmissions, denoted by $g(r)$, depends on r and the particular HARQ scheme used for combining multiple transmission attempts (an empirical method to estimate $g(r)$ is presented in [29]). As in any reasonable HARQ strategy, we assume that $g(r)$ is non-zero for any r and is non-increasing in the number of retransmissions r; that is, $g(r_1) \geq g(r_2) > 0$ for all $r_1 \leq r_2$. We also assume that the transmissions are successful with a positive probability $g(r) < 1$ for all r. Standard HARQ methods only combine information from a finite maximum number of retransmissions [30]. Accordingly, we consider a truncated retransmission count of a status update, denoted by R_t for the status update transmitted at time t, where $R_t \in \{0, \ldots, R_{max}\}$; that is, the receiver can combine information from the last R_{max} retransmissions at most. We

set $R_0 = 0$ so that there is no previously transmitted packet at the transmitter at time $t = 0$.

At the end of each time slot t, a random amount of energy is harvested and stored in a rechargeable battery at the transmitter, denoted by $E_t \in \mathcal{E} \triangleq \{0, 1, \ldots, E_{max}\}$, following a first-order discrete-time Markov model, characterized by the stationary transition probabilities $p_E(e_1|e_2)$, defined as $p_E(e_1|e_2) \triangleq Pr(E_{t+1} = e_2|E_t = e_1)$, $\forall t$ and $\forall e_1, e_2 \in \mathcal{E}$. It is also assumed that $p_E(0|e) > 0$, $\forall e \in \mathcal{E}$. Harvested energy is first stored in a rechargeable battery with a limited capacity of $B_{\max}$ energy units. The energy consumption for status sensing is denoted by $E^{\mathrm{s}} \in \mathbb{Z}^+$, while the energy consumption for a transmission attempt is denoted by $E^{\mathrm{tx}} \in \mathbb{Z}^+$.

The battery state at the beginning of time slot t, denoted by B_t, can be written as follows:

$$B_{t+1} = \min(B_t + E_t - (E^{\mathrm{s}} + E^{\mathrm{tx}})\mathbb{1}[A_t = \mathrm{n}] - E^{\mathrm{tx}}\mathbb{1}[A_t = \mathrm{x}], B_{\max}), \tag{1}$$

and the energy causality constraints are given as:

$$(E^{\mathrm{s}} + E^{\mathrm{tx}})\mathbb{1}[A_t = \mathrm{n}] + E^{\mathrm{tx}}\mathbb{1}[A_t = \mathrm{x}] \leq B_t, \tag{2}$$

where the indicator function $\mathbb{1}[C]$ is equal to 1 if event C holds, and zero otherwise. (1) implies that the battery overflows if energy is harvested when the battery is full, while (2) imposes that the energy consumed by sensing or transmission operations at time slot t is limited by the energy B_t available in the battery at the beginning of that time slot. We set $B_0 = 0$ so that the battery is empty at time $t = 0$.

Let Δ_t^{tx} denote the number of time slots elapsed since the generation of the most recently sensed status update at the transmitter side, while Δ_t^{rx} denote the AoI of the most recently received status update at the receiver side. Δ_t^{tx} resets to 1 if a new status update is generated at time slot $t - 1$, and increases by one (up to Δ_{max}) otherwise, i.e.,

$$\Delta_{t+1}^{tx} = \begin{cases} 1 & \text{if } A_t = \mathrm{n}; \\ \min(\Delta_t^{tx} + 1, \Delta_{max}) & \text{otherwise.} \end{cases}$$

On the other hand, the AoI at the receiver side evolves as follows:

$$\Delta_{t+1}^{rx} = \begin{cases} 1 & \text{if } A_t = \mathrm{n} \text{ and } K_t = 1; \\ \min(\Delta_t^{tx} + 1, \Delta_{max}) & \text{if } A_t = \mathrm{x} \text{ and } K_t = 1; \\ \min(\Delta_t^{rx} + 1, \Delta_{max}) & \text{otherwise .} \end{cases}$$

Note that once the AoI at the receiver is at least as large as at the transmitter, this relationship holds forever; thus it is enough to consider cases when $\Delta_t^{rx} \geq \Delta_t^{tx}$.

To determine the success probability of a transmission, we need to keep track of the number of current retransmissions. The number of retransmissions is zero for a newly sensed and generated status update and increases up to R_{max} as we keep retransmitting the same packet. In practical systems R_{max} is limited to a few retransmissions (e.g., $R_{max} = 3$ for IEEE 802.16 standards [30] and LTE systems with a maximum allowed retransmissions of 27 while the default is $R_{max} = 3$ [31]). It is easy to see that retransmitting when $\Delta_{t+1}^{tx} = \Delta_{max}$ is suboptimal, therefore we explicitly exclude this action by setting the retransmission count to 0 in this case. Also, it is suboptimal to generate a new update and retransmit the old one, thus whenever a new status update is generated, the previous update at the transmitter is dropped and cannot be retransmitted. Thus, the evolution of the retransmission count is given as follows:

$$
R_{t+1} = \begin{cases}
0 & \text{if } K_t = 1 \\
& \text{or } \Delta_{t+1}^{tx} = \Delta_{max}; \\
1 & \text{if } A_t = \text{n and } K_t = 0; \\
R_t & \text{if } A_t = \text{i} \\
& \text{and } \Delta_{t+1}^{tx} \neq \Delta_{max}; \\
\min(R_t + 1, R_{max}) & \text{if } A_t = \text{x}, K_t = 0 \\
& \text{and } \Delta_{t+1}^{tx} \neq \Delta_{max}.
\end{cases}
$$

The state of the system is formed by five components $S_t = (E_t, B_t, \Delta_t^{rx}, \Delta_t^{tx}, R_t)$. In each time slot, the transmitter knows the current state, and takes action from the set $\mathcal{A} = \{\text{i}, \text{n}, \text{x}\}$. The goal is to find a policy π which minimizes the expected average AoI at the receiver over an infinite time horizon, which is given by:

Problem 1.

$$
J^* \triangleq \min_{\pi} \lim_{T \to \infty} \frac{1}{T+1} \mathbb{E} \left[\sum_{t=0}^{T} \Delta_t^{rx} \right] \tag{3}
$$

subject to (1) *and* (2).

In [24], we have considered status updates with HARQ under an average power constraint. In that case, it is possible to show that it is suboptimal to retransmit a failed update after an idle period. Restricting the actions of the transmitter accordingly, the AoI at the receiver after

a successful transmission event is equal to the number of retransmissions of the corresponding update. Therefore in addition to the AoI at the receiver, we only need to track the retransmission count. However, in the current scenario, retransmissions of a status update can be interrupted due to energy outages, which means that we also need to keep track of the AoI at the transmitter side (hence we need to have both Δ_t^{rx} and Δ_t^{tx} in the state of the system).

III. MDP Formulation

It is easy to see that Problem 1 can be formulated as an average-cost finite-state MDP: An MDP is defined by the quadruple $(\mathcal{S}, \mathcal{A}, P, c)$ [32]: The finite set of states $\mathcal{S}$ is defined as $\mathcal{S} = \{s = (e, b, \delta^{rx}, \delta^{tx}, r) : e \in \mathcal{E}, b \in \{0, \ldots, B_{\max}\}, \delta^{rx}, \delta^{tx} \in \{1, \ldots, \Delta_{max}\}, r \in \{0, \ldots, R_{max}\}, \Delta^{rx} \geq \Delta^{tx}\}$, while the finite set of actions $\mathcal{A} = \{i, n, x\}$ is already defined. Note that action x cannot be taken in states with retransmission count $r = 0$. P refers to the transition probabilities, where $P(s'|s, a) = \Pr(S_{t+1} = s' \mid S_t = s, A_t = a)$ is the probability that action a in state s at time t will lead to state s' at time $t + 1$, which is characterized by the EH statistics and channel error probabilities. The cost function $c : \mathcal{S} \times \mathcal{A} \to \mathbb{Z}$, is the AoI at the receiver, and is defined as $c(s, a) = \delta^{rx}$ for any $s \in \mathcal{S}$, $a \in \mathcal{A}$, independent of the action taken, where δ^{rx} is the component of s describing the AoI at the receiver.

To solve Problem 1, we need to find a policy for the transmitter, determining its actions for every state $s \in \mathcal{S}$, which can minimize the average AoI at the receiver.

It is easy to see that MDP formulated for Problem 1 is a communicating MDP by Proposition 8.3.1 of [32][1] , i.e., for every pair of $(s, s') \in \mathcal{S}$, there exists a deterministic policy under which s' is accessible from s. By Theorem 8.3.2 of [32], an optimal stationary policy exists with constant gain. In particular, there exists a function $h : \mathcal{S} \to \mathbb{R}$, called the *differential cost function* for all $s = (e, b, \delta^{rx}, \delta^{tx}, r) \in \mathcal{S}$, satisfying the following *Bellman optimality equations* for the average-cost finite-state finite-action MDP [32]:

$$h(s) + J^* = \min_{a \in \{i, n, x\}} \left(\delta^{rx} + \mathbb{E}\left[h(s')|a \right] \right), \tag{4}$$

---

[1]By Proposition 8.3.1 of [32], MDP is communicating since there exists a stationary policy which induces a recurrent Markov chain, e.g., a state $(0, B_0, \Delta_{max}, \Delta_{max}, R_0)$ is reachable from all other states considering a policy which never transmits and in a scenario where no energy is harvested. Another example can be given as: from a state with $B_t \neq B_0$ the state $(0, B_0, \Delta_{max}, \Delta_{max}, R_0)$ is reachable considering a policy which stays idle until energy sufficient for $B_0 + R_0$ retransmissions and then begin transmitting the same update R_0 times and then idles in a scenario where every transmission fails and no energy is harvested.

where $s' \triangleq (e', b', \delta^{rx'}, \delta^{tx'}, r')$ is the next state obtained from $(e, b, \delta^{rx}, \delta^{tx}, r)$ after taking action a, and J^* represents the optimal achievable average AoI under policy π^*. Note that the function h satisfying (4) is unique up to an additive factor, and with selecting this additive factor properly, it also satisfies

$$h(s) = \mathbb{E}\left[\sum_{t=0}^{\infty}(\Delta_t^{rx} - J^*)\big|S_0 = s\right].$$

We also introduce the *state-action cost function*:

$$Q((e, b, \delta^{rx}, \delta^{tx}, r), a) \triangleq \delta^{rx} + \mathbb{E}\left[h(e', b', \delta^{rx'}, \delta^{tx'}, r')|a\right]. \tag{5}$$

Then an optimal policy, for any $(e, b, \delta^{rx}, \delta^{tx}, r) \in \mathcal{S}$, takes the action achieving the minimum in (5):

$$\pi^*(e, b, \delta^{rx}, \delta^{tx}, r) \in \underset{a \in \{i,n,x\}}{\arg\min}\, Q((e, b, \delta^{rx}, \delta^{tx}, r), a). \tag{6}$$

An optimal policy solving (4), (5) and (6) defined above can be found by relative value iteration (RVI) for finite-state finite-action average-cost MDPs from Sections 8.5.5 and 9.5.3 of [32]: Starting with an arbitrary initialization of $h_0(s)$, $\forall s \in \mathcal{S}$, and setting an arbitrary but fixed reference state $s^{ref} \triangleq (e^{ref}, b^{ref}, \delta^{rxref}, \delta^{txref}, r^{ref})$, a single iteration of the RVI algorithm $\forall (s, a) \in \mathcal{S} \times \mathcal{A}$ is given as follows:

$$Q_{n+1}(s, a) \leftarrow \Delta_n^{rx} + \mathbb{E}\left[h_n(s')\right], \tag{7}$$

$$V_{n+1}(s) \leftarrow \min_a (Q_{n+1}(s, a)), \tag{8}$$

$$h_{n+1}(s) \leftarrow V_{n+1}(s) - V_{n+1}(s^{ref}), \tag{9}$$

where $Q_n(s, a)$, $V_n(s)$ and $h_n(s)$ denote the state-action value function, value function and differential value function at iteration n, respectively. By Theorem 8.5.7 and Section 8.5.5 of [32], h_n converges to h, and $\pi_n^*(s) \triangleq \arg\min_a Q_n(s, a)$ converges to $\pi^*(s)$.

A. Structure of the Optimal Policy

Next, we present the structure of the optimal policy and show that the solution to Problem 1 is of threshold-type.

Theorem 1. *There exits an optimal stationary policy $\pi^*(s)$ that is monotone with respect to Δ_t^{rx}, that is, $\pi^*(s)$ is of threshold-type.*

Proof. The proof is given in Appendix A. □

Following Theorem 1, we present a threshold-based policy which will be termed as a *double-threshold policy* in the rest of this paper.

$$
A_t = \begin{cases} i & \text{if } \Delta_t^{rx} < \mathcal{T}_{\mathrm{n}}(e, b, \delta^{tx}, r), \\ n & \text{if } \mathcal{T}_{\mathrm{n}}(e, b, \delta^{tx}, r) \leq \Delta_t^{rx} < \mathcal{T}_{\mathrm{x}}(e, b, \delta^{tx}, r), \\ x & \text{if } \Delta_t^{rx} \geq \mathcal{T}_{\mathrm{x}}(e, b, \delta^{tx}, r), \end{cases} \tag{10}
$$

for some threshold values denoted by $\mathcal{T}_{\mathrm{n}}(e, b, \delta^{tx}, r)$ and $\mathcal{T}_{\mathrm{x}}(e, b, \delta^{tx}, r)$, where $\mathcal{T}_{\mathrm{n}}(e, b, \delta^{tx}, r) \leq \mathcal{T}_{\mathrm{x}}(e, b, \delta^{tx}, r)$.

We can simplify the problem by making an assumption on the policy space in order to obtain a simpler *single-threshold policy*, which will result in a more efficient learning algorithm: We assume that a packet is retransmitted until it is successfully decoded, provided that there is enough energy in the battery, that is, the transmitter is not allowed to preempt an undecoded packet and transmit a new one.

The solution to the simplified problem is also of threshold-type, that is,

$$
A_t = \begin{cases} i & \text{if } \Delta_t^{rx} < \mathcal{T}(e, b, \delta^{tx}, r), \\ n & \text{if } \Delta_t^{rx} \geq \mathcal{T}(e, b, \delta^{tx}, r), \text{ and } r = 0 \\ x & \text{if } \Delta_t^{rx} \geq \mathcal{T}(e, b, \delta^{tx}, r) \text{ and } r \neq 0, \end{cases} \tag{11}
$$

for some $\mathcal{T}(e, b, \delta^{tx}, r)$.

In Section IV-B, we present a RL algorithm to find the threshold values defined in this section.

IV. RL Approach

In some scenarios, it can be assumed that the channel and energy arrival statistics remain the same or change very slowly and the same environment is experienced for a sufficiently long time before the time of deployment. Accordingly, we can assume that the statistics regarding the error probabilities and energy arrivals are available before the time of transmission. In such scenarios, RVI algorithm presented in Section III can be used. However, in most practical scenarios, channel error probabilities for retransmissions and the EH characteristics are not known at the time of deployment, or may change over time. In this section, we assume that the transmitter does not know the system characteristics *a-priori*, and has to learn them. In our previous works [22]–[24], we have employed learning algorithms for constrained problems

with countably infinite state spaces such as average-cost SARSA. While these algorithms can be adopted to the current framework by considering an average transmission constraint of 1, they do not have convergence guarantees. However, Problem 1 constitutes an unconstrained MDP with finite state and action spaces, and there exist RL algorithms for unconstrained MDPs which enjoy convergence guarantees. Moreover, we have shown the optimality of a threshold type policy for Problem 1, and RL algorithms which exploit this structure can be considered. Thus, we employ three different RL algorithms, and compare their performances in terms of the average AoI as well as the convergence speed. First, we employ a value-based RL algorithm, namely GR-learning, which converges to an optimal policy. Next, we consider a structured policy search algorithm, namely FDPG, which does not necessarily find the optimal policy but performs very well in practice, as demonstrated through simulations in Section V. We also note that GR-learning learns from a single trajectory generated during learning steps while FDPG uses Monte-Carlo roll-outs for each policy update. Thus, GR-learning is more amenable for applications in real-time systems. Finally, we employ the DQN algorithm, which implements a non-linear neural network estimator in order to learn the optimal status update policy.

A. GR-Learning with Softmax

The literature for average-cost RL is quite limited compared to discounted cost problems [33], [34]. For the average AoI minimization problem in (3), we employ a modified version of the *GR-learning* algorithm proposed in [27], as outlined in Algorithm 1, with *Boltzmann* (*softmax*) exploration. The resulting algorithm is called *GR-learning with softmax*.

Notice that, by only knowing $Q(s, a)$, one can find the optimal policy π^* using (6) without knowing the transition probabilities P, characterized by $g(r)$ and p_E. Thus, *GR-learning with softmax* starts with an initial estimate of $Q_0(s, a)$ and finds the optimal policy by estimating state-action values in a recursive manner. $Q_n(s, a)$ and J_n denote the estimates of state-action value function and the average cost (average AoI) at iteration n, respectively while S_n, and A_n denote the state and action at iteration n. Note that since GR-learning is an online algorithm each iteration of the algorithm corresponds to a single time slot. In the nth iteration, after taking action A_n, the transmitter observes the next state S_{n+1} and the instantaneous cost value Δ_n^{rx}. Based on these and the estimate of the average cost J_n, the estimate of $Q_{n+1}(s, a)$ is updated by a weighted average of the previous estimate $Q_n(s, a)$ and the estimated expected value of

the current policy in the next state S_{n+1}. Moreover, we update the estimate of the average cost J_n at every time slot based on the empirical average of AoI.

In each time slot, the learning algorithm

- observes the current state $S_n \in \mathcal{S}$,
- selects and performs an action $A_n \in \mathcal{A}$,
- observes the next state $S_{n+1} \in \mathcal{S}$ and the instantaneous cost Δ_n^{rx},
- updates its estimate of $Q(S_n, A_n)$ using the current estimate of J_n by

$$Q_{n+1}(S_n, A_n) \leftarrow Q_n(S_n, A_n) + \alpha(m(S_n, A_n, n))[\Delta_n^{rx} - J_n + Q_n(S_{n+1}, A_{n+1}) - Q_n(S_n, A_n)],$$
(12)

where $\alpha(m(S_n, A_n, n))$ is the update parameter (learning rate) in the nth iteration, and depends on the function $m(S_n, A_n, n)$, which is the number of times the state–action pair (S_n, A_n) has been visited till the nth iteration.

- updates its estimate of J_n based on the empirical average as follows:

$$J_{n+1} \leftarrow J_n + \beta(n) \left[\frac{nJ_n + \Delta_n^{rx}}{n+1} - J_n \right],$$
(13)

where $\beta(n)$ is the update parameter in the nth iteration.

The transmitter action selection method should balance the *exploration* of new actions with the *exploitation* of actions known to perform well. We use the *Boltzmann* (*softmax*) action selection method, which chooses each action randomly relative to its expected cost as follows:

$$\pi(a|S_n) = \frac{\exp(-Q(S_n, a)/\tau_n)}{\sum\limits_{a' \in \mathcal{A}} \exp(-Q(S_n, a')/\tau_n)}.$$
(14)

Parameter τ_n in (14) is called the temperature parameter and decays exponentially with decay parameter $\gamma_\tau \leq 1$ at each iteration. High τ_n corresponds to more uniform action selection (exploration) whereas low τ_n is biased toward the best action (exploitation). According to Theorem 2 of [27], if α, β satisfy $\sum_{m=1}^{\infty} \alpha(m), \sum_{m=1}^{\infty} \beta(m) \to \infty$, $\sum_{m=1}^{\infty} \alpha^2(m), \sum_{m=1}^{\infty} \beta^2(m) < \infty$, $\lim_{x \to \infty} \frac{\beta(m)}{\alpha(m)} \to 0$, *GR*-Learning converges to an optimal policy.

B. Finite-Difference Policy Gradient (FDPG)

GR-learning in Section IV-A is a value-based RL method, which learns the state-action value function for each state-action pair. In practice, Δ_{max} can be large, which might slow down the convergence of GR-learning due to a prohibitively large state space. Instead, we are going

---

Algorithm 1 GR-learning with softmax

1: $\tau_0 \leftarrow 1,$ /* temperature parameter */

2: $\gamma \leftarrow 0.95,$ /* softmax decay coefficient */ .

3: $Q_0 \leftarrow 0, \forall(s,a)$ /* initialization of Q */

4: $J_0 \leftarrow 0,$ /* initialization of the gain */

5: **for** $n = 1, 2, \ldots$ **do**

6:      OBSERVE current state S_n

7:      **for** $a \in \mathcal{A}$ **do**

$$\pi(a|S_n) = \frac{\exp(-Q(S_n, a)/\tau_n)}{\sum_{a' \in \mathcal{A}} \exp(-Q(S_n, a')/\tau_n)}$$

8:      **end for**

9:      SAMPLE and PERFORM A_n from $\pi(a|S_n)$

10:      OBSERVE the next state S_{n+1} and cost Δ_t

11:      UPDATE $Q(S_n, A_n)$ and J_n by

$$Q(S_n, A_n) \leftarrow Q(S_n, A_n) + \alpha(m(S_n, A_n, n))[\Delta_t - J_n + \min_{A_{n+1}} Q(S_{n+1}, A_{n+1}) - Q(S_n, A_n)]$$

$$J_{n+1} \leftarrow J_n + \beta(n)\left[\frac{nJ_n + \Delta_t}{n+1} - J_n\right]$$

12:      UPDATE step size parameters

$$\tau_{n+1} \leftarrow \gamma\tau_n$$

$$\alpha(n) \leftarrow 1/\sqrt{m(S_n, A_n, n)}$$

$$\beta(n) \leftarrow 1/n$$

$$m(S_n, A_n) \leftarrow m(S_n, A_n) + 1$$

$$m(s, a) \leftarrow m(s, a), \ \forall(s, a) \neq (S_n, A_n)$$

13: **end for**

---

to propose a learning algorithm that exploits the structure of the optimal policy exposed in Theorem 1. A monotone policy is shown to be average optimal in the previous section; however, it is not possible to compute the threshold values directly for Problem 1.

Note that, $A_t = \text{i}$ if $B_t < E^{\text{tx}}$ ($B_t < E^{\text{tx}} + E^{\text{s}}$) for $r \geq 1$ ($r = 0$); that is, $\mathcal{T}(e, b, \delta^{tx}, r) = \Delta_{max} + 1$ if $b < E^{\text{tx}}$ for $r \geq 1$ and $b < E^{\text{tx}} + E^{\text{s}}$ for $r = 0$. This ensures that energy causality constraints in (2) hold. Other thresholds will be determined using policy gradient.

In order to employ the policy gradient method, we approximate the policy by a parameterized smooth function with parameters $\theta(e, b, \delta^{tx}, r)$, and convert the discrete policy search problem

into estimating the optimal values of these continuous parameters, which can be numerically solved by stochastic approximation algorithms [35]. Continuous stochastic approximation is much more efficient than discrete search algorithms in general.

In particular, with a slight abuse of notation, we let $\pi_\theta(e, b, \delta^{rx}, \delta^{tx}, r)$ denote the probability of taking action $A_t = \text{n}$ ($A_t = \text{x}$) if $r = 0$ ($r \neq 0$), and consider the parameterized sigmoid function:

$$\pi_\theta(e, b, \delta^{rx}, \delta^{tx}, r) \triangleq \frac{1}{1 + e^{-\frac{\delta^{rx} - \theta(e, b, \delta^{tx}, r)}{\tau}}}. \tag{15}$$

We note that $\pi_\theta(e, b, \delta^{rx}, \delta^{tx}, r) \to \{0, 1\}$ and $\theta(e, b, \delta^{tx}, r) \to \mathcal{T}(e, b, \delta^{tx}, r)$ as $\tau \to 0$. Therefore, in order to converge to a deterministic policy π, $\tau > 0$ can be taken as a sufficiently small constant, or can be decreased gradually to zero. The total number of parameters to be estimated is $|\mathcal{E}| \times B_{\max} \times \Delta_{max} \times R_{max} + 1$ minus the parameters corresponding to $b < E^{\text{tx}}$ ($b < E^{\text{tx}} + E^{\text{s}}$) for $r > 0$ ($r = 0$) due to energy causality constraints as stated previously.

With a slight abuse of notation, we map the parameters $\theta(e, b, \delta^{tx}, r)$ to a vector $\overline{\theta}$ of size $d \triangleq |\mathcal{E}| \times B_{\max} \times \Delta_{max} \times R_{max} + 1$. Starting with some initial estimates of $\overline{\theta}_0$, the parameters can be updated at each iteration n using the gradients as follows:

$$\overline{\theta}_{n+1} = \overline{\theta}_n - \gamma(n) \ \partial J / \partial \overline{\theta}_n, \tag{16}$$

where the step size parameter $\gamma(n)$ is a positive decreasing sequence and satisfies the first two convergence properties given at the end of Section IV-A from the theory of stochastic approximation [36].

Computing the gradient of the average AoI directly is not possible; however, several methods exist in the literature to estimate the gradient [35]. In particular, we employ the FDPG [28] method, where the gradient is estimated by estimating J at slightly perturbed parameter values. First, a random perturbation vector D_n of size d is generated according to a predefined probability distribution, e.g., each component of D_n is an independent Bernoulli random variable with parameter $q \in (0, 1)$. The thresholds are perturbed with a small amount $\sigma > 0$ in the directions defined by D_n to obtain $\overline{\theta}_n^{\pm}(e, b, \delta^{tx}, r) \triangleq \overline{\theta}_n(e, b, \delta^{tx}, r) \pm \sigma D_n$. Then, empirical estimates $\hat{J}^{\pm}$ of the average AoI corresponding to the perturbed parameters $\overline{\theta}_n^{\pm}$, obtained from Monte-Carlo rollouts, are used to estimate the gradient:

$$\partial J / \partial \overline{\theta}_n \approx (D_n^{\intercal} D_n)^{-1} D_n^{\intercal} \frac{(\hat{J}^+ - \hat{J}^-)}{2\sigma}, \tag{17}$$

Algorithm 2 FDPG

Input: error probabilities $g(r)$ and harvesting probabilities p_E are unknown

1: $\tau_0 \leftarrow 0.3$,   */ temperature parameter */

2: $\zeta \leftarrow 0.99$,   */ decaying coefficient for τ */.

3: $\overline{\theta}_0 \leftarrow 0$ */ initialization of $\overline{\theta}$ */

4: **for** $n = \{1, 2, \ldots\}$ **do**

5:     GENERATE random perturbation vector

        $D_n = binomial(\{0, 1\}, q = 0.5, d)$

6:     PERTURB parameters $\overline{\theta}_n$

        $\overline{\theta}_n^+ = \overline{\theta}_n + \beta D_n, \overline{\theta}_n^- = \overline{\theta}_n - \beta D_n$

7:     ESTIMATE $\hat{J}_n^{\pm}$ from Monte-Carlo rollouts using policies $\pi_{\theta_n^{\pm}}$:

8:     **for** $t \in \{1, \ldots, T\}$ **do**

9:         OBSERVE current state S_t and USE policy $\pi_{\theta_n^{\pm}}$

10:     **end for**

11:     ESTIMATE $\hat{J}_n^{\pm}$ from Monte-Carlo rollouts using policy $\pi_{\theta_n^{\pm}}$

        $\hat{J}_n^{\pm} = \frac{1}{T} \sum_{t=1}^{T} \Delta_t^{rx}$

12:     COMPUTE the estimate of the gradient $\partial J / \partial \overline{\theta}_n$

13:     UPDATE

        $\overline{\theta}_{n+1} = \overline{\theta}_n - \gamma(n) \; \partial J / \partial \overline{\theta}_n$

        $\tau_{n+1} \leftarrow \zeta \tau_n$ */ decrease τ */

14: **end for**

---

where $D_n^{\intercal}$ denotes the transpose of vector D_n. The pseudo code of the finite difference policy gradient algorithm is given in Algorithm 2.

Similar steps can be followed to find the thresholds for the double-threshold policy, where $\mathcal{T}(e, b, \delta^{tx}, r)$ and $\theta(e, b, \delta^{tx}, r)$ are replaced by $\mathcal{T}_{\mathrm{n}}(e, b, \delta^{tx}, r)$, $\mathcal{T}_{\mathrm{x}}(e, b, \delta^{tx}, r)$ and $\theta_{\mathrm{n}}(e, b, \delta^{tx}, r)$, $\theta_{\mathrm{x}}(e, b, \delta^{tx}, r)$, respectively.

C. Deep Q-Network (DQN)

A DQN uses a multi-layered neural network in order to estimate the values $Q(s, a)$ of the underlying MDP; that is, for a given state s, DQN outputs a vector of state-action values, $Q_\theta(s, a)$, where θ denotes the parameters of the network. For the 5-dimensional state space ($|S| = 5$) and an action space containing 3 actions ($|\mathcal{A}| = 3$) as in our problem formulation, the neural network is a function from $|S|$ inputs to $|\mathcal{A}|$ outputs, which are the estimates of the Q-function $Q_\theta(s, a)$.

We apply the DQN algorithm of [37] to learn a scheduling policy. We create a simple feed-forward neural network of 3 layers, one of which is the hidden layer with 24 neurons. We use *Huber loss* [38] and the *Adam* algorithm [39] to conduct stochastic gradient descent to update the weights of the neural network.

We exploit two important features of DQNs as proposed in [37]: *experience replay* and a *fixed target network*, both of which provide algorithm stability. For *experience replay*, instead of training the neural network with a single observation $< s, a, s', c(s,a) >$ at the end of each step, many experiences (i.e., (state, action, next state, cost) quadruplets) can be stored in the replay memory for batch training, and a minibatch of observations randomly sampled at each step can be used. The DQN uses two neural networks: a target network and an online network. The *target network*, with parameters θ^-, is the same as the online network except that its parameters are updated with the parameters θ of the online network after every T steps, and θ^- is kept fixed in other time slots. For a minibatch of observations for training, temporal difference estimation error ε for a single observation can be calculated as

$$\varepsilon(s, a, s', c) = Q_\theta(s,a) - (-c(s, a) + \gamma Q_{\theta^-}(s', \arg\max Q_\theta(s',a))). \tag{18}$$

Huber loss is defined by the squared error term for small estimation errors, and a linear error term for high estimation errors, allowing less dramatic changes in the value functions, further improving the stability. For a given estimation error ε and loss parameter d, the Huber loss function, denoted by $\mathrm{L}^d(\varepsilon)$ is defined as:

$$\mathrm{L}^d(\varepsilon(s, a, s', c)) = \begin{cases} \varepsilon(s, a, s', c)^2 & \text{if } \varepsilon(s, a, s', c) \leq d, \\ d(|\varepsilon(s, a, s', c)| - \frac{1}{2}d)) & \text{if } \varepsilon(s, a, s', c) > d, \end{cases}$$

and loss over minibatch $\mathcal{B}$ is computed as:

$$\mathrm{L}_\mathcal{B} \triangleq \frac{1}{|\mathcal{B}|} \sum_{<s,a,s',c(s,a)> \in \mathcal{B}} \mathrm{L}^d(\varepsilon(s, a, s', c)).$$

We apply the ϵ-greedy policy to balance exploration and exploitation, i.e., with probability ϵ the source randomly selects an action, and with probability $1 - \epsilon$ it chooses the action with the minimum Q value. We let ϵ decay gradually from ϵ_0 to ϵ_{min}; in other words, the source explores more at the beginning of training and exploits more at the end. The hyperparameters of the DQN algorithm are tuned for our problem experimentally, and are presented in Table I.

Table I

HYPERPARAMETERS OF THE DQN ALGORITHM

Parameter	Value	Parameter	Value
discount factor γ	0.99	optimizer	Adam
minibatch size	32	loss function	Huber loss
replay memory length	2000	exploration coeff. ϵ_0	1
learning rate α	10^{-4}	ϵ decay rate β	0.9
episode length T	1000	ϵ_{min}	0.01
activation function	ReLU	hidden size	24

V. SIMULATION RESULTS

In this section, we provide numerical results for all the proposed algorithms, and compare the achieved average AoI. Motivated by previous research on HARQ [40], [29], [30], we assume that the decoding error reduces exponentially with the number of retransmissions, that is, $g(r) \triangleq p_0 \lambda^r$ for some $\lambda \in (0, 1)$, where p_0 denotes the error probability of the first transmission and r is the retransmission count (set to 0 for the first transmission). The exact value of the rate λ depends on the particular HARQ protocol and the channel model. Following the *IEEE 802.16* standard [30], the maximum number of retransmissions used for decoding is set to $R_{max} = 3$. In the following experiments, λ and p_0 are set to 0.5. E^{tx} and E^s are both assumed to be constant and equal to 1 unit of energy unless otherwise stated. Δ_{max} is set to 40.

We choose the exact step sizes for the learning algorithms by fine-tuning in order to balance the algorithm stability in the early time steps with nonnegligible step sizes in the later time steps. In particular, we use step size parameters of $\alpha(m), \beta(m), \gamma(m) = y/(m+1)^z$, where $0.5 < z \le 1$ and $y > 0$ (which satisfy the convergence conditions), and choose y and z such that the oscillations are low and the convergence rate is high. We have observed that a particular choice of parameters gives similar performance results for scenarios addressed in simulations results. DQN algorithm in this section is configured as in Table I and trained for 500 episodes. The average AoI for DQN is obtained after 10^5 time steps and averaged over 100 runs.

As a baseline, we have also included the performance of a greedy policy, which senses and transmits a new status update whenever there is sufficient energy. It retransmits the last transmitted status update when the energy in the battery is sufficient only for transmission, and
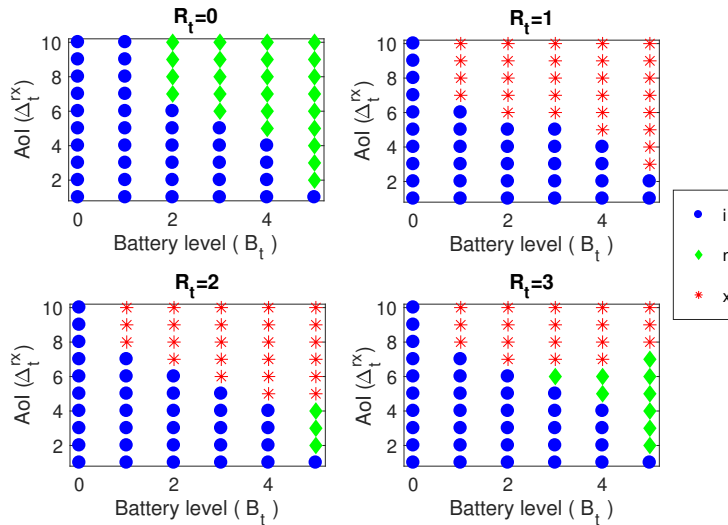
Figure 2. Optimal policy for memoryless EH when $B_{\max} = 5$, $R_{max} = 3$, $p_e = 0.5$ and $E^{\mathrm{s}} = E^{\mathrm{tx}} = 1$. The decoding error probabilities are given by $g(r) = 2^{-(r+1)}$.

it remains idle otherwise; that is,

$$
A_t^{greedy} = \begin{cases} \mathrm{i} & \text{if } B_t < E^{\mathrm{tx}}, \\ \mathrm{n} & \text{if } B_t \geq E^{\mathrm{tx}} + E^s, \\ \mathrm{x} & \text{if } E^{\mathrm{tx}} \leq B_t < E^{\mathrm{tx}} + E^s. \end{cases} \tag{19}
$$

A. Memoryless EH Process

We first investigate the average AoI with HARQ when the EH process, $E_t \in \mathcal{E} = \{0, 1\}$, is i.i.d. over time with probability distribution $p_e \triangleq Pr(E_t = 1)$, and $1 - p_e \triangleq Pr(E_t = 0) \; \forall t$. Fig. 2 illustrates the policy obtained by the RVI algorithm in Section III. The resulting policy is more likely to transmit if the battery level or the AoI is high as expected. Moreover, the policy tends to retransmit the previous update rather than sensing a new update when the battery level is low and the AoI is high. We can also observe from the figure that the optimal policy exhibits a threshold structure as shown in Theorem 1.

The effects of the battery capacity $B_{\max}$, energy consumption of sensing E^{s}, and the energy harvesting probability p_e on the average AoI are shown in Fig. 3. As expected, the average AoI increases with decreasing $B_{\max}$, decreasing p_e and increasing E^{s}. We note that, when $E^{\mathrm{s}} = 0$ and $B_{\max} = \infty$, the problem defined in (3) corresponds to minimizing the average AoI under
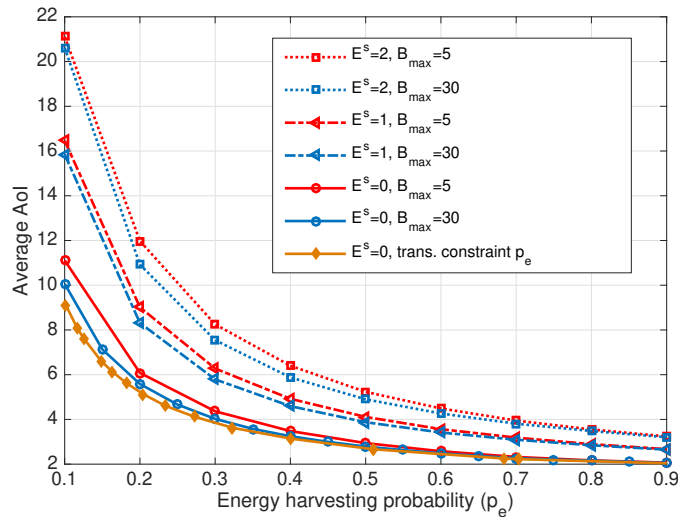
Figure 3. Average AoI for different $B_{\max}$, E^{s} and p_e values for memoryless EH and $E^{\mathrm{tx}} = 1$.

an average transmission rate constraint p_e, studied in [22], [24]. The average AoI under average transmission rate constraint ($B_{\max} = \infty$) is also shown in Fig. 3, and we observe that its performance can be approximated with a finite battery size of $B_{\max} = 30$ at low p_e values, while a battery size of $B_{\max} = 5$ is sufficient when p_e increases.

Fig. 4 shows the evolution of the average AoI over time when the average-cost RL algorithms are employed. It can be observed that the average AoI achieved by the proposed RL algorithms converge to values close to the one obtained from the RVI algorithm, which has *a priori* knowledge of $g(r)$ and p_e, while the AoI of the greedy algorithm is significantly higher. Although GR-learning enjoys theoretical guaranties to converge to the optimal policy, the FDPG which benefits from the structural guarantees of a threshold policy (including a single-threshold policy not allowing preemption of an undecoded status update), performs better than GR-learning since it tries to learn significantly smaller number of threshold values (i.e., $\Delta_{max} \times B_{\max} \times R_{max} + 1$) compared to GR-learning, which learns one value for each state-action pair (i.e., $\Delta_{max}^2 \times B_{\max} \times (R_{max} + 1) \times |\mathcal{A}|$). We also observe that, among the FDPG methods, the one with a single threshold converges faster but the double-threshold policy finally attains a slightly lower AoI. Therefore, the choice between the two may depend on the stochasticity of the environment. DQN algorithm performs better than GR-learning but it requires a training time before running the simulation and does not have convergence guarantees. Moreover, its
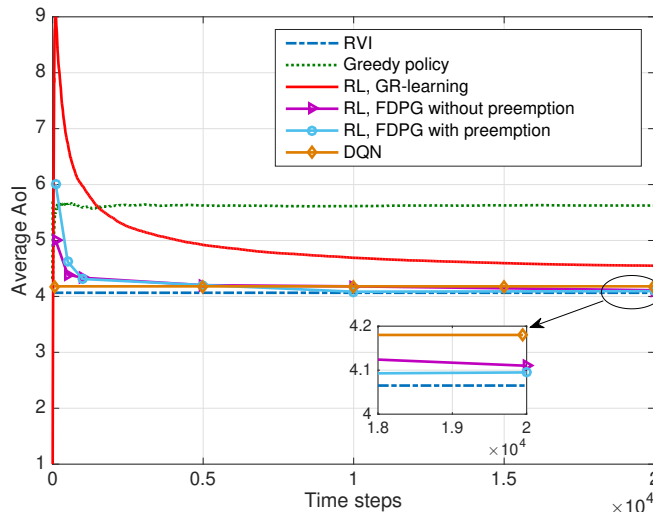
Figure 4. Performance of RL algorithms when $B_{\max} = 5$, $E^{\mathrm{s}} = E^{\mathrm{tx}} = 1$, and $p_e = 0.5$. FDPG with and without preemption represent the double-threshold and the single-threshold policies, respectively.

final performance is slightly worse than both of the FDPG algorithms.

B. Temporally Correlated EH

Next, we investigate the performance when the EH process has temporal correlations. A symmetric two-state Markovian EH process is assumed, such that $\mathcal{E} = \{0, 1\}$ and $Pr(E_{t+1} = 1 | E_t = 0) = Pr(E_{t+1} = 0 | E_t = 1) = 0.3$. That is, if the transmitter is in harvesting state, it is more likely to continue harvesting energy, and vice versa for the non-harvesting state.

Fig. 5 illustrates the policy obtained by RVI. As it can be seen from the figure, the resulting policy is less likely to transmit if the battery level or the AoI is low. As shown in Theorem 1, the optimal policy exhibits a threshold structure on Δ^{rx}. Moreover, the policy tends to retransmit the previous update rather than sensing a new update when the battery level is low and the AoI is high. When the system is in the non-harvesting state (i.e., $E_t = 0$), the transmitter is more conservative in transmitting the status updates compared to the case $E_t = 1$, e.g., it might not transmit even if the battery is full depending on the AoI level.

Fig. 7 shows the evolution of the average AoI over time when the average-cost RL algorithms are employed in this scenario. It can be observed again that the average AoI achieved by the FDPG method in Section IV-B performs very close to the one obtained by the RVI algorithm, which has *a priori* knowledge of $g(r)$ and p_e. GR-learning, on the other hand, outperforms the
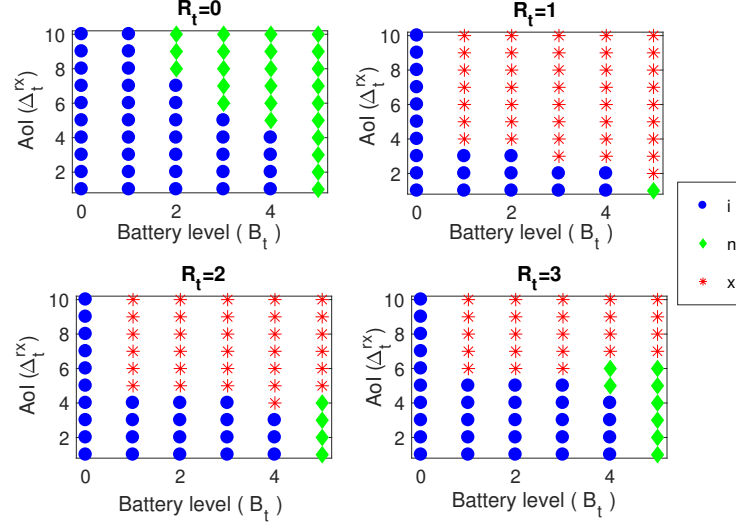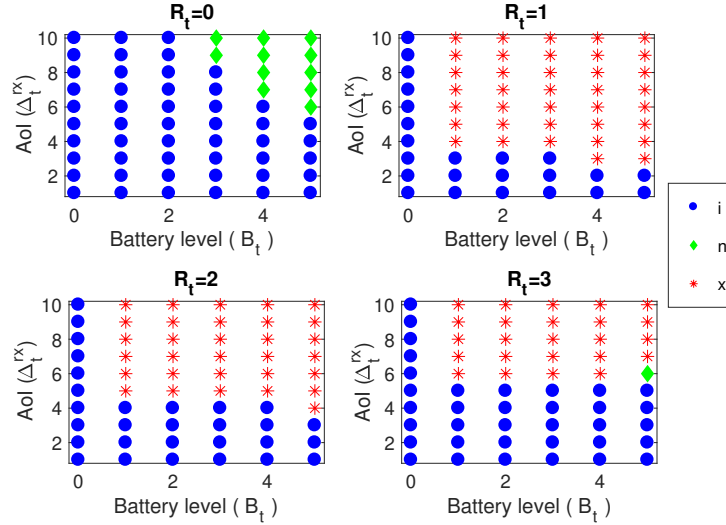
(a) $E_t = 1$

(a) $E_t = 0$

Figure 5. Optimal policy for $B_{\max} = 5$, $R_{max} = 3$, $p_E(1,1)$, $p_E(0,0) = 0.7$, $E^{\text{s}} = E^{\text{tx}} = 1$ and $\Delta_t^{tx} = R_t + 1$. The decoding error probabilities are given by $g(r) = 2^{-(r+1)}$.

greedy policy but converges to the optimal policy much more slowly, and the gap between the two RL algorithms is larger compared to the i.i.d. case. Tabular methods in RL, like GR-learning, need to visit each state-action pair infinitely often for RL to converge [33]. GR-learning in the case of temporally correlated EH does not perform as well as in the i.i.d. case since the state space becomes larger with the addition of the EH state. We also observe that the gap between

the final performances of single- and double-threshold FDPG solutions is larger compared to the memoryless EH scenario, while the single threshold solution still converges faster. DQN algorithm performs better than GR-learning but it requires a training time before running the simulation and does not have convergence guarantees. Moreover, it still falls short of the final performance of double-threshold FDPG.

Fig. 6 illustrates the effect of preemption and the performance improvement of double-threshold FDPG over single-threshold FDPG for a scenario where preemption is inherently needed, e.g., $g(r)$ is same for all retransmissions r representing a standard ARQ protocol and dropping a failed update improves the performance. As it can be seen from Fig. 6, although single-threshold FDPG converges very close to the RVI without preemption, the average AoI obtained by single-threshold FDPG is still considerable higher than that of double-threshold FDPG for standard ARQ protocol.
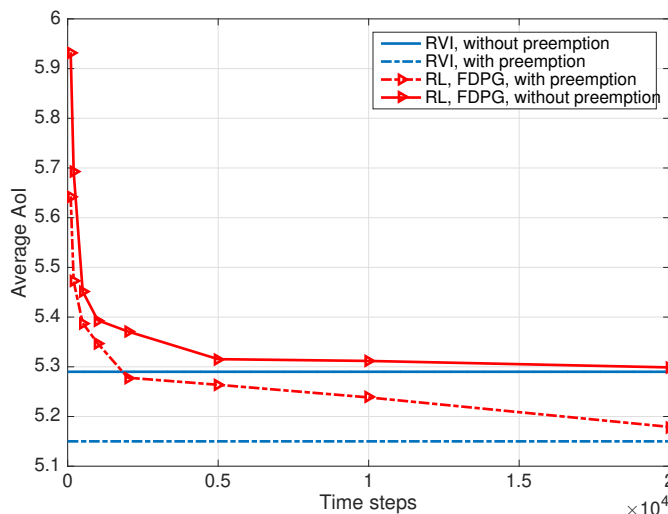

Figure 6. The performance of FDPG algorithms when $B_{\max} = 5$, $R_{max} = 3$, $p_E(1,1)$, $g(r) = 0.5$, $\forall r$, $p_E(0,0) = 0.7$ and $E^{\mathrm{s}} = E^{\mathrm{tx}} = 1$. FDPG with and without preemption represent the double-threshold and the single-threshold policies, respectively.

Next, we investigate the impact of the burstiness of the EH process, measured by the correlation coefficient between E_t and E_{t+1}. Fig. 8 illustrates the performance of the proposed RL algorithms for different correlation coefficients, which can be computed easily for the 2-state symmetric Markov chain; that is, $\rho \triangleq (2p_E(1,1) - 1)$. Note that $\rho = 0$ corresponds to memoryless EH with $p_e = 1/2$. We note that the average AoI is minimized by transmitting new packets successfully at regular intervals, which has been well investigated in previous works [5], [6], [22]. Intuitively, for
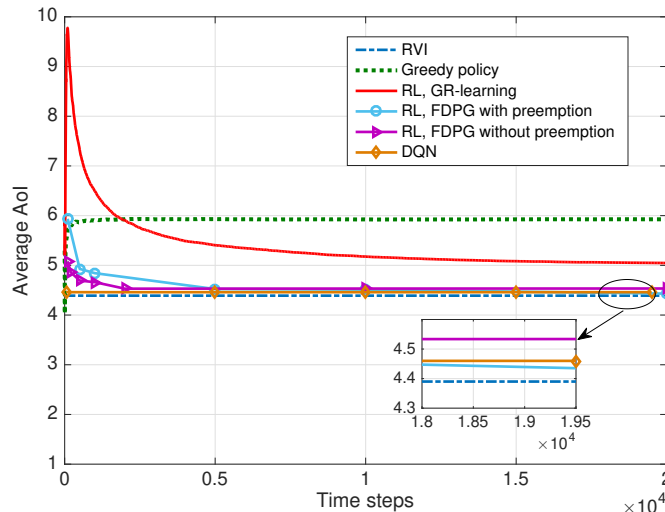
Figure 7. The performance of RL algorithms when $B_{\max} = 5$, $p_E(1,1)$, $p_E(0,0) = 0.7$ and $E^s, E^{tx} = 1$. FDPG with and without preemption represent the double-threshold and the single-threshold policies, respectively.

highly correlated EH, there are either successive transmissions or successive idle time slots, which increases the average AoI. Hence, the AoI is higher for higher values of ρ. Fig. 8 also shows that both RL algorithms result in much lower average AoI than the greedy policy and FDPG outperforms GR-learning since it benefits from the structural characteristics of a threshold policy. We can also conclude that the single threshold policy can be preferable in practice especially in highly dynamic environments, as its performance is very close to that of the double threshold FDPG, but with faster convergence.

VI. CONCLUSIONS

We have considered an EH system with a finite size battery and investigated scheduling policies transmitting time-sensitive data over a noisy channel with the average AoI as the performance measure. We have assumed the presence of an ACK/NACK feedback from the receiver, and allowed retransmissions with an HARQ protocol to increase the probability of correct reception of status updates. This results in a trade-off between sending new status updates and retransmitting failed status updates as the former results in a lower AoI at the receiver while the latter is more likely to succeed. This trade-off is exacerbated in the model considered in this paper by the introduction of a sensing cost, which increases the cost of new status updates, and requires judicious decisions at the transmitter due to limited and stochastic availability of energy.
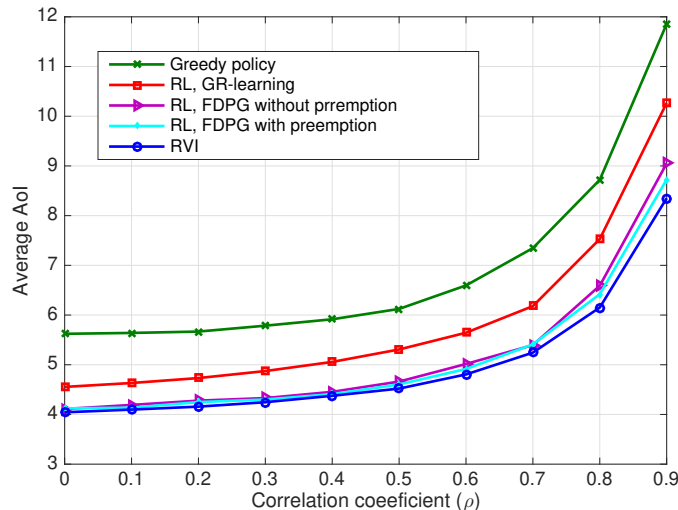
Figure 8. The performance of RL algorithms obtained after $2 \cdot 10^4$ time steps and averaged over 1000 runs for different temporal correlation coefficients.

In addition to identifying a RVI solution for the optimal policy and showing that the optimal policy is stationary, deterministic, and monotone with respect to the AoI when the system characteristics are known, efficient RL algorithms are presented for practical applications when the system characteristics may not be known in advance. The effects of the battery size, EH characteristics, and the HARQ structure on the average AoI are investigated through numerical simulations.

We have presented three types of RL algorithms with different levels of complexity and training requirements and compared their performances for the current problem under a variety of system setting. We have observed that FDPG policies that exploit the threshold structure of the optimal policy provide both better performance and convergence behaviour. Moreover, a simplified single-threshold FDPG alternative is shown to increase the convergence speed with a negligible increase in the average AoI.

APPENDIX

A. *Proof of Theorem 1:*

By (5) and (6), Theorem 1 holds if $Q((e, b, \delta^{rx}, \delta^{tx}, r), a)$ has a *sub-modular* structure in (δ^{rx}, a) [41]: that is, when the difference between the Q function is monotone with respect to

the state-action pair (δ^{rx}, a) for any $E_t = e$, $B_t = b$, $\Delta_t^{tx} = \delta^{tx}$, and $R_t = r$. We show the submodularity by verifying the following inequality for 3 action pairs $(a_1, a_2) \in \{(i, n), (i, x), (n, x)\}$:

$$Q((e, b, \delta^{rx} + 1, \delta^{tx}, r), a_2) - Q((e, b, \delta^{rx} + 1, \delta^{tx}, r), a_1)$$

$$\leq Q((e, b, \delta^{rx}, \delta^{tx}, r), a_2) - Q((e, b, \delta^{rx}, \delta^{tx}, r), a_1) \tag{20}$$

Inequality (20) can be rewritten for $(a_1, a_2) = (i, n)$ using (5),

$$Q((e, b, \delta^{rx}, \delta^{tx}, r), n) = \delta^{rx} + \sum_{e' \in \mathcal{E}} p_E(e, e')[g(r)h(e', b + e - E^s - E^{tx}, \delta^{rx} + 1, 1, 1)$$

$$+ (1 - g(r))h(e', b + e - E^s - E^{tx}, 1, 1, 0)] \tag{21}$$

and

$$Q((e, b, \delta^{rx}, \delta^{tx}, r), i) = \delta^{rx} + \sum_{e' \in \mathcal{E}} p_E(e, e')h(e', b + e, \delta^{rx} + 1, \delta^{tx} + 1, r). \tag{22}$$

(21) and (22) are inserted into (20), and since the next state $E_{t+1} = e'$ is independent of action (A_t) and AoIs $(\Delta_t^{rx}, \Delta_t^{tx})$, the following is equivalent to (20):

$$g(0)\big[h(e', b + e - E^s - E^{tx}, \delta^{rx} + 2, 1, 1) - h(e', b + e - E^s - E^{tx}, \delta^{rx} + 1, 1, 1)\big]$$

$$- \big[h(e', b + e, \delta^{rx} + 2, \delta^{tx} + 1, r) - h(e', b + e, \delta^{rx} + 1, \delta^{tx} + 1, r)\big] \leq 0. \tag{23}$$

Also we note that $\Delta_{t+1}^{rx} = \delta^{rx} + 2$, $\Delta_{t+1}^{tx} = \delta^{tx} + 1$ and $R_t = r$ are truncated to Δ_{max}, Δ_{max} and R_{max} respectively.

The same steps can be repeated for $(a_1, a_2) \in (i, x)$ and $(a_1, a_2) \in (n, x)$, and we obtain the following:

$$g(r)\big[h(e', b + e - E^{tx}, \delta^{rx} + 2, \delta^{tx} + 1, r + 1) - h(e', b + e - E^{tx}, \delta^{rx} + 1, \delta^{tx} + 1, r + 1)\big]$$

$$- \big[h(e', b + e, \delta^{rx} + 2, \delta^{tx} + 1, r) - h(e', b + e, \delta^{rx} + 1, \delta^{tx} + 1, r)\big] \leq 0 \tag{24}$$

$$g(r)\big[h(e', b + e - E^{tx}, \delta^{rx} + 2, \delta^{tx} + 1, r + 1) - h(e', b + e - E^{tx}, \delta^{rx} + 1, \delta^{tx} + 1, r + 1)\big]$$

$$- g(0)\big[h(e', b + e - E^s - E^{tx}, \delta^{rx} + 2, 1, 1) - h(e', b + e - E^s - E^{tx}, \delta^{rx} + 1, 1, 1)\big] \leq 0 \tag{25}$$

Therefore, (23), (24) and (25) are the sufficient conditions for the submodularity of Q function.

First, we note that Eqns. (23), (24) and (25) hold with equality for $(\delta^{rx} + 1, \delta^{rx}) = (\Delta_{max}, \Delta_{max} - 1)$. Then, we show by induction that if (23), (24) and (25) hold for $(\delta^{rx} + 2, \delta^{rx} + 1)$ then they hold for $(\delta^{rx} + 1, \delta^{rx})$.

First, we check for $(a_1, a_2) = (\text{i}, \text{x})$, and assume that h satisfies (23), (24) and (25). We define the related Q functions with optimal actions denoted by a_1^*, a_2^*, a_3^* and a_4^* such that:

$$Q((e, b - E^{tx}, \delta^{rx} + 1, \delta^{tx} + 1, r + 1), a_1^*) \triangleq h(e, b - E^{tx}, \delta^{rx} + 1, \delta^{tx} + 1, r + 1) + J^*, \quad (26)$$

$$Q((e, b - E^{tx}, \delta^{rx}, \delta^{tx} + 1, r + 1), a_2^*) \triangleq h(e, b - E^{tx}, \delta^{rx}, \delta^{tx} + 1, r + 1) + J^*, \quad (27)$$

$$Q((e, b, \delta^{rx} + 1, \delta^{tx} + 1, r), a_3^*) \triangleq h(e, b, \delta^{rx} + 1, \delta^{tx} + 1, r) + J^*, \quad (28)$$

$$Q((e, b, \delta^{rx}, \delta^{tx}, r), a_4^*) \triangleq h(e, b, \delta^{rx}, \delta^{tx}, r) + J^*. \quad (29)$$

We need to show that (24) holds for $(\delta^{rx} + 1, \delta^{rx})$, which can be rewritten using (26), (27), (28) and (29):

$$g(r)\big[Q((e, b - E^{tx}, \delta^{rx} + 1, \delta^{tx} + 1, r + 1), a_1^*) - Q((e, b - E^{tx}, \delta^{rx}, \delta^{tx} + 1, r + 1), a_2^*)\big]$$
$$- \big[Q((e, b, \delta^{rx} + 1, \delta^{tx} + 1, r), a_3^*) - Q((e, b, \delta^{rx}, \delta^{tx}, r), a_4^*)\big] \leq 0 \quad (30)$$

We add terms $g(r)[-Q((e, b - E^{tx}, \delta^{rx} + 1, \delta^{tx} + 1, r + 1), a_2^*) + Q((e, b - E^{tx}, \delta^{rx} + 1, \delta^{tx} + 1, r + 1), a_2^*)]$ and $[Q((e, b, \delta^{rx}, \delta^{tx}, r), a_3^*) - Q((e, b, \delta^{rx}, \delta^{tx}, r), a_3^*)]$ to the LHS of (30) and obtain:

$$g(r)\big[Q((e, b - E^{tx}, \delta^{rx} + 1, \delta^{tx} + 1, r + 1), a_1^*) - Q((e, b - E^{tx}, \delta^{rx} + 1, \delta^{tx} + 1, r + 1), a_2^*)$$
$$+ Q((e, b - E^{tx}, \delta^{rx} + 1, \delta^{tx} + 1, r + 1), a_2^*) - Q((e, b - E^{tx}, \delta^{rx}, \delta^{tx} + 1, r + 1), a_2^*)\big]$$
$$- \big[Q((e, b, \delta^{rx} + 1, \delta^{tx} + 1, r), a_3^*) - Q((e, b, \delta^{rx}, \delta^{tx}, r), a_3^*)$$
$$+ Q((e, b, \delta^{rx}, \delta^{tx}, r), a_3^*) - Q((e, b, \delta^{rx}, \delta^{tx}, r), a_4^*)\big] \leq 0 \quad (31)$$

$Q((e, b_1, \delta + 1, r + 1), a_1^*) - Q((e, b_1, \delta + 1, r + 1), a_2^*)$ is less than or equal to 0 from the optimality of a_1^*. Similarly, $Q((e, b, \delta^{rx}, \delta^{tx}, r), a_4^*) - Q((e, b, \delta^{rx}, \delta^{tx}, r), a_3^*)$ is less than or equal to 0 from the optimality of a_4^*. Then, we only need to show:

$$g(r)[Q((e, b - E^{tx}, \delta^{rx} + 1, \delta^{tx} + 1, r + 1), a_2^*) - Q((e, b - E^{tx}, \delta^{rx}, \delta^{tx} + 1, r + 1), a_2^*)]$$
$$- Q((e, b, \delta^{rx} + 1, \delta^{tx} + 1, r), a_3^*) + Q((e, b, \delta^{rx}, \delta^{tx}, r), a_3^*) \leq 0. \quad (32)$$

The condition in (32) can be checked for all possible values of (a_2^*, a_3^*) pair: First we investigate for the pair $(a_2^*, a_3^*) = (\text{i}, \text{i})$; that is,

$$g(r)\big[1 + h(e', b - E^{tx} + e, \delta^{rx} + 2, \delta^{tx} + 1, r + 1) - h(e', b - E^{tx} + e, \delta^{rx} + 1, \delta^{tx} + 1, r + 1)\big]$$
$$- 1 + h(e', b + e, \delta^{rx} + 1, \delta^{tx} + 1, r) - h(e', b + e, \delta^{rx} + 2, \delta^{tx} + 1, r) \leq 0. \quad (33)$$

LHS of (33) is equivalent to LHS of (24) plus the term $g(r) - 1$, which is smaller than and equal to 0 since (24) holds and $g(r) \leq 1$. For pair $(a_2^*, a_3^*) = (\mathrm{x}, \mathrm{x})$; that is,

$$g(r)\big[1 + g(r+1)h(e', b - 2E^{tx} + e, \delta + 2, r + 2) - g(r+1)h(e', b - 2E^{tx} + e, \delta^{rx} + 1, \delta^{tx} + 1, r + 1)\big]$$
$$-1 - g(r)h(e', b - E^{tx} + e, \delta^{rx} + 2, \delta^{tx} + 1, r + 1) + g(r)h(e', b - E^{tx} + e, \delta^{rx} + 1, \delta^{tx} + 1, r + 1) \leq 0 \tag{34}$$

Similarly, $g(r) - 1$ is less than 0 and (34) holds.

For pair $(a_2^*, a_3^*) = (\mathrm{i}, \mathrm{x})$:

$$g(r)\big[1 + h(e', b - E^{tx} + e, \delta^{rx} + 2, \delta^{tx} + 1, r + 1) - h(e', b - E^{tx} + e, \delta^{rx} + 1, \delta^{tx} + 1, r + 1)\big]$$
$$-1 - g(r)h(e, e', b - E^{tx} + e, \delta^{rx} + 2, \delta^{tx} + 1, r + 1) + g(r)h(e', b - E^{tx} + e, \delta^{rx} + 1, \delta^{tx} + 1, r + 1) \leq 0, \tag{35}$$

which is equal to:

$$g(r) - 1 \leq 0. \tag{36}$$

For pair $(a_2^*, a_3^*) = (\mathrm{x}, \mathrm{i})$:

$$g(r)\big[1 + g(r+1)h(e', b - 2E^{tx} + e, \delta + 2, r + 2) - g(r+1)h(e', b - 2E^{tx} + e, \delta^{rx} + 1, \delta^{tx} + 1, r + 1)\big]$$
$$-1 + h(e', b + e, \delta^{rx} + 1, \delta^{tx} + 1, r) - h(e', b + e, \delta^{rx} + 2, \delta^{tx} + 1, r) \leq 0. \tag{37}$$

which is equal to:

$$g(r)g(r+1)\big[h(e', b - 2E^{tx}, \delta + 2, r + 2) - h(e', b - 2E^{tx} + e, \delta^{rx} + 1, \delta^{tx} + 1, r + 1)\big]$$
$$- h(e', b + e, \delta^{rx} + 2, \delta^{tx} + 1, r) + h(e', b + e, \delta^{rx} + 1, \delta^{tx} + 1, r) + g(r) - 1 \leq 0. \tag{38}$$

From (24), $-h(e', b + e, \delta^{rx} + 2, \delta^{tx} + 1, r) + h(e', b + e, \delta^{rx} + 1, \delta^{tx} + 1, r) \leq g(r)(-h(e', b + e - E^{tx}, \delta^{rx} + 2, \delta^{tx} + 1, r + 1) + h(e', b + e - E^{tx}, \delta + 1, r + 1))$, and $1 - g(r) \leq 0$; thus (38) is smaller than

$$g(r)\Big\{g(r+1)\big[h(e', b - 2E^{tx}, \delta + 2, r + 2) - h(e', b - 2E^{tx} + e, \delta^{rx} + 1, \delta^{tx} + 1, r + 1)\big]$$
$$+ (-h(e', b + e - E^{tx}, \delta^{rx} + 2, \delta^{tx} + 1, r + 1) + h(e', b + e - E^{tx}, \delta + 1, r + 1))\Big\}, \tag{39}$$

which is smaller than 0, since the expression inside the braces is equivalent to (24) with $r \to r+1$ and $g(r) \geq 0$.

The same holds for $(a_2^*, a_3^*) = (\text{x}, \text{n})$ and $(a_2^*, a_3^*) = (\text{n}, \text{x})$. Similar steps could be followed for other $(a_1, a_2) = (\text{i}, \text{n})$ and $(a_1, a_2) = (\text{n}, \text{x})$} pairs and are not included in this paper due to space limitations.

Thus, the condition is satisfied, i.e., Q function is submodular in (δ^{rx}, a). From [41], it can be concluded that the status update policy is of threshold-type.

REFERENCES

[1] E. T. Ceran, D. Gündüz, and A. György, "Reinforcement learning to minimize age of information with an energy harvesting sensor with HARQ and sensing cost," in *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, April 2019.

[2] C. Alippi and C. Galperti, "An adaptive system for optimal solar energy harvesting in wireless sensor network nodes," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 55, no. 6, pp. 1742–1750, July 2008.

[3] M. A. Weimer, T. S. Paing, and R. A. Zane, "Remote area wind energy harvesting for low-power autonomous sensors," in *IEEE Power Electronics Specialists Conference*, June 2006, pp. 1–5.

[4] A. Nechibvute, A. Chawanda, and P. Luhanga, "Piezoelectric energy harvesting devices: An alternative energy source for wireless sensors," *Smart Materials Research*, p. 13, 2012.

[5] B. T. Bacinoglu, E. T. Ceran, and E. Uysal-Biyikoglu, "Age of information under energy replenishment constraints," in *2015 Information Theory and Applications Workshop (ITA)*, Feb 2015, pp. 25–31.

[6] R. D. Yates, "Lazy is timely: Status updates by an energy harvesting source," in *IEEE International Symposium on Information Theory (ISIT)*, June 2015, pp. 3008–3012.

[7] B. T. Bacinoglu and E. Uysal-Biyikoglu, "Scheduling status updates to minimize age of information with an energy harvesting sensor," in *2017 IEEE International Symposium on Information Theory (ISIT)*, 2017, pp. 1122–1126.

[8] X. Wu, J. Yang, and J. Wu, "Optimal status update for age of information minimization with an energy harvesting source," *IEEE Transactions on Green Communications and Networking*, vol. 2, no. 1, pp. 193–204, March 2018.

[9] B. T. Bacinoglu, Y. Sun, E. Uysal, and V. Mutlu, "Optimal status updating with a finite-battery energy harvesting source," *Journal of Communications and Networks*, vol. 21, no. 3, pp. 280–294, 2019.

[10] S. Feng and J. Yang, "Age of information minimization for an energy harvesting source with updating erasures: Without and with feedback," *IEEE Transactions on Communications*, pp. 1–1, 2021.

[11] A. Arafa, J. Yang, S. Ulukus, and H. V. Poor, "Age-minimal online policies for energy harvesting sensors with incremental battery recharges," *Information Theory and Applications Workshop (ITA)*, Feb 2018.

[12] M. A. Abd-Elmagid, H. S. Dhillon, and N. Pappas, "A reinforcement learning framework for optimizing age of information in rf-powered communication systems," *IEEE Transactions on Communications*, vol. 68, no. 8, pp. 4747–4760, 2020.

[13] M. Hatami, M. Jahandideh, M. Leinonen, and M. Codreanu, "Age-aware status update control for energy harvesting IoT sensors via reinforcement learning," in *2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*, 2020, pp. 1–6.

[14] E. Gindullina, L. Badia, and D. Gündüz, "Age-of-information with information source diversity in an energy harvesting system," *IEEE Transactions on Green Communications and Networking*, pp. 1–1, 2021.

[15] N. Pappas, Z. Chen, and M. Hatami, "Average AoI of cached status updates for a process monitored by an energy harvesting sensor," in *2020 54th Annual Conference on Information Sciences and Systems (CISS)*, 2020, pp. 1–5.

[16] N. Zhao, C. Xu, S. Zhang, Y. Xie, X. Wang, and H. Sun, "Status update for correlated energy harvesting sensors: A deep reinforcement learning approach," in *2020 International Conference on Wireless Communications and Signal Processing (WCSP)*, 2020, pp. 170–175.

[17] S. Fu, Y. Tang, Y. Wu, N. Zhang, H. Gu, C. Chen, and M. Liu, "Energy-efficient uav-enabled data collection via wireless charging: A reinforcement learning approach," *IEEE Internet of Things Journal*, vol. PP, pp. 1–1, 01 2021.

[18] J. Brusey, J. Kemp, E. Gaura, R. Wilkins, and M. Allen, "Energy profiling in practical sensor networks: Identifying hidden consumers," *IEEE Sensors Journal*, vol. 16, no. 15, pp. 6072–6080, Aug 2016.

[19] J. Gong, X. Chen, and X. Ma, "Energy-age tradeoff in status update communication systems with retransmission," in *2018 IEEE Global Communications Conference (GLOBECOM)*, 2018, pp. 1–6.

[20] M. Xie, J. Gong, S. Cai, and X. Ma, "Age-energy tradeoff for two-hop status update systems with heterogeneous truncated ARQ," *IEEE Wireless Communications Letters*, pp. 1–1, 2021.

[21] B. Zhou and W. Saad, "Joint status sampling and updating for minimizing age of information in the internet of things," *IEEE Transactions on Communications*, vol. 67, no. 11, pp. 7468–7482, 2019.

[22] E. T. Ceran, D. Gündüz, and A. György, "Average age of information with hybrid ARQ under a resource constraint," in *IEEE Wireless Communications and Networking Conference (WCNC)*, 2018.

[23] ——, "Reinforcement learning approach to age of information in multi-user networks," in *IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2018.

[24] E. T. Ceran, D. Gündüz, and A. György, "Average age of information with hybrid ARQ under a resource constraint," *IEEE Transactions on Wireless Communications*, vol. 18, pp. 1900–1913, March 2019.

[25] E. T. Ceran, D. Gündüz, and A. György, "A reinforcement learning approach to age of information in multi-user networks with HARQ," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 5, pp. 1412–1426, 2021.

[26] D. Gunduz, K. Stamatiou, N. Michelusi, and M. Zorzi, "Designing intelligent energy harvesting communication systems," *IEEE Communications Magazine*, vol. 52, pp. 210–216, 2014.

[27] A. Gosavi, "Reinforcement learning for long-run average cost," *European Journal of Operational Research*, vol. 155, pp. 654 – 674, 2004.

[28] J. Peters and S. Schaal, "Policy gradient methods for robotics," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2006, pp. 2219–2225.

[29] V. Tripathi, E. Visotsky, R. Peterson, and M. Honig, "Reliability-based type ii hybrid ARQ schemes," in *IEEE International Conference on Communications,*, vol. 4, May 2003, pp. 2899–2903 vol.4.

[30] "IEEE standard for local and metropolitan area networks part 16: Air interface for fixed and mobile broadband wireless access systems amendment 2: Physical and medium access control layers for combined fixed and mobile operation in licensed bands and corrigendum 1 (incorporated into IEEE standard 802.16e-2005 and IEEE std 802.16-2004/cor1-2005)," *IEEE Std P802.16/Cor1/D5*, 2006.

[31] [Online]. Available: "http://rfmw.em.keysight.com/wireless/helpfiles/n76 25bpxb/Content/RT/UL-SCHSettings.htmMaximumNumberofRetrans missions"

[32] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York, NY, USA: John Wiley & Sons, 1994.

[33] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*, 1st ed. Cambridge, MA, USA: MIT Press, 1998.

[34] S. Mahadevan, "Average reward reinforcement learning: Foundations, algorithms, and empirical results," *Machine Learning*, vol. 22, no. 1, pp. 159–195, 1996.

[35] J. C. Spall, *Introduction to Stochastic Search and Optimization*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2003.

[36] H. J. Kushner and G. G. Yin, *Stochastic Approximation Algorithms and Applications*. Orlando, FL, USA: New York: Springer-Verlag, 1997.

[37] 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.

[38] P. J. Huber, "Robust estimation of a location parameter," *The Annals of Mathematical Statistics*, vol. 35, no. 1, pp. 73–101, 03 1964. [Online]. Available: https://doi.org/10.1214/aoms/1177703732

[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[40] P. Frenger, S. Parkvall, and E. Dahlman, "Performance comparison of HARQ with chase combining and incremental redundancy for HSDPA," in *Proc. IEEE Vehicular Technology Conference*, vol. 3, 2001, pp. 1829–1833.

[41] D. M. Topkis, "Minimizing a submodular function on a lattice," *Operational Research*, vol. 26, no. 2, pp. 305–321, Apr. 1978.