RL-based Transmission Completion Time Minimization with Energy Harvesting for Time-varying Channels

Heasung Kim*, Wonjae Shin†, Heecheol Yang‡, and Jungwoo Lee*
*Department of Electrical and Computer Engineering, Seoul National University, Seoul, Korea
†Department of Electrical Engineering, Pusan National University, Busan, Korea
‡School of Electronic Engineering, Kumoh National Institute of Technology, Gumi, Korea
E-mails: *{heasung1130, junglee}@snu.ac.kr, †wjshin@pusan.ac.kr, ‡hc.yang@kumoh.ac.kr

***Abstract*—In this paper, we consider the problem of minimizing the transmission completion time in energy harvesting devices on time-varying channels with a reinforcement learning approach. Because of the randomness of energy arrival and fading channel in wireless communications, a reinforcement learning algorithm often converges to suboptimal points with a degraded performance. To solve this problem, we first prove that the expected discounted reward sum in the environment is an increasing function of negative time, amount of data sent, channel gain, harvested energy, and remaining battery. We leverage this proof to construct a partially monotonic network that efficiently approximates the optimal action-value function for learning. Experimental results show that our approach with the exploitation of the partial monotonicity of the desired function achieves better performance than existing power allocation policies. Further experiments show that the performance of our learning-based approach is close to the theoretical upper bound over rapidly time-varying channels.**

***Index Terms*—energy harvesting communications, transmission completion time minimization, reinforcement learning.**

I. Introduction

Transmit power allocation to achieve energy efficiency and low latency is essential for green communication networks. In particular, transmission completion time minimization (TCTM) has become one of the essential requirements of the emerging Internet of Things (IoT) and sensor networks. However, the devices for the networks with an energy-harvesting technology cannot employ the conventional power allocation schemes because there is no guarantee that the same amount of energy will be supplied continuously. Moreover, due to the randomness of the channel state in time-varying channels, it is often difficult to find an optimal power allocation policy in an analytical form. In this paper, to address the realistic constraints of green wireless communications, we deal with the TCTM problem in time-varying channels with an energy-harvesting constraint.

Power allocation problems can be categorized according to the type of available information the transmitter can exploit for establishing its policy. The first is the offline case, a scenario in which the transmitter knows about all future energy arrivals or channel state transitions. In [1], the optimal policy for a single user of a TCTM problem in an offline scenario according to packet preparation has been studied. Similar to TCTM, delay minimization problems, which minimize the interval between the packet arrival and transmission times, were addressed with broadcast channels and infinite-sized battery in recent works [2]. The authors of paper [3] solved the optimal policy of TCTM in the fading channel through the directional water-filling approach. In [4], the fading channel was studied with the energy harvesting constraint, and the problem of minimizing the energy required for packet transmission was considered. In [5], the TCTM for the uplink transmissions with multiple nodes was considered. In [6], the problem of deciding the time of harvesting energy and delivering data with a specific transmission rate for TCTM was studied. A non-orthogonal multiple-access environment was considered with the minimization of the offloading delay in [7].

If the transmitter has only the casual information on the energy arrival or channel states, it is classified as an online scenario. An online policy for time-invariant channels was provided with a performance bound of the policy in [8]. The authors of [6] provided the online solution running on the wireless power transmission network with separate charging and transmission phases. In this paper, we assume the online scenario and consider the time-varying channels as well as the randomness of the energy arrivals.

We formulate a TCTM problem into a Markov decision process (MDP) problem. This follows from the fact that the MDP problem with a small number of states can be solved by tabular-based reinforcement learning (RL) methods [9], but the methods take longer to learn as the number of states increases due to the curse of dimensions. In light of this, RL approach through function approximators, especially the value-based method, is applied to efficiently solve this problem. In particular, as the strong capability of neural networks has been proven, many RL algorithms have shown improved performance using neural networks as function approximators [10]–[12].

However, using such a capable neural network indiscriminately will make the function approximator overly complex for approximating the desired function, thereby inhibiting the learning process [13]. To solve this problem, we prove that the

978-1-7281-7440-2/20/$31.00 ©2020 IEEE

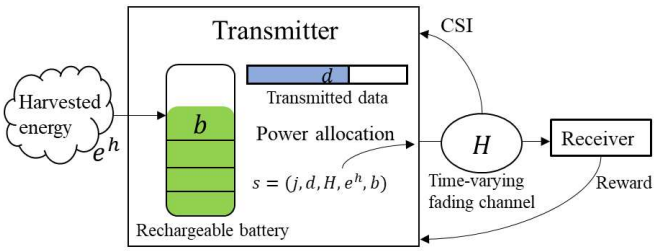

Fig. 1. An energy harvesting communication system in a time-varying channel.

optimal action-value function is a partially and monotonically increasing function to avoid a heuristic network construction, and provide a mathematical basis for function approximator design.

Our contributions to solving the problems of network construction for the function approximation and TCTM are summarized as follows:

- To reflect the realistic wireless communications, we provide an online policy by interpreting the TCTM problem as a reinforcement learning problem considering time-varying channels with an energy-harvesting constraint.
- We prove that the action-value function widely used in reinforcement learning problems is a partially monotonic function that increases over negative time, transmitted data, channel gain, harvested energy, and remaining battery in the TCTM problem. Based on the proof, we build a partially and monotonically increasing network with respect to its input variables by adopting the lattice network.
- Numerical results show that the performance of our learning approach outperforms several existing approaches and achieves low transmission delay which is close to that of the upper bound of the online approaches.

II. System Model

An energy harvesting communication system is considered with a time-varying channel as shown in Fig. 1. An AWGN fading channel is assumed where (the received signal at the receiver is) $Y = \sqrt{H}X + Z$, X is the transmitted signal, and Z is a Gaussian noise with power density N_0. It is assumed that the transmitter transmits a signal first when a time slot begins and then harvests energy. The harvested energy is stored in the finite-sized rechargeable battery of the transmitter. The harvested energy at time slot i is denoted by e_i^h which is independent and identically distributed (i.i.d.) for all $i \geq 0$. The harvested energy is used only for the data transmission. The harvested energy $e_i^h (\leq e_{\max}^h) \in \mathcal{E}$ and the amount of energy in the rechargeable battery $b_i (\leq b_{\max}) \in \mathcal{B}$ are discrete values in bounded discrete spaces $\mathcal{E}$ and $\mathcal{B}$, respectively. The finite size of the battery, $\max \mathcal{B} = b_{\max}$, is assumed. Note that $H_i \in \mathcal{H}$ represents the channel gain at time slot i, which is i.i.d. for all $i \geq 0$. $\mathcal{H}$ is discrete space and the transmitter has the channel state information (CSI) on H_i at time slot i.

The Shannon's capacity for Gaussian noise channels is given by

$$D(H, p) = WT_t \log(1 + \frac{Hp}{WN_0}) \tag{1}$$

where W is the bandwidth the transmitter uses, H is the channel gain, and p is the transmission power during the transmission time T_t. We write a negative time j $(= -i$ for $i < -r_b) \in \mathcal{J}$ and i is increased by 1 every time slot. If $i \geq -r_b$, $j = r_b$. The remaining battery determines the action space $P(s) \subset [0, b/T_t]$ every time slot and the transmitter uses power $p \in P(s)$ based on the state s consists of five components: $s = (j, d, H, e^h, b)$. p_i is the transmission power at time slot i. In addition, the amount of data sent to the receiver by the time slot i is denoted by $d \in \mathcal{D}$. $\mathcal{J}$ and $\mathcal{D}$ are discrete spaces and the state space is $S = \mathcal{J} \times \mathcal{D} \times \mathcal{H} \times \mathcal{E} \times \mathcal{B}$. The reward function r can be expressed as

$$r(s, p) \begin{cases} = 1 \text{ for } j - r_b > 0, d + D(H, p) \geq d_o \\ = 0 \text{ otherwise,} \end{cases} \tag{2}$$

where d_0 is the total size of the desired data packet that the transmitter want to send to the receiver and $(r_b < 0)$ is the baseline performance which is determined before learning. This reward function r indicates that the faster the transmitter completes the transmission than the baseline performance, the higher the reward. The state transition probability is denoted $p_s(s_{i+1}|s_i, p_i)$ which means the the probability of moving from state s_i to s_{i+1} with performing the action p_i.

Our goal is to maximize the discounted reward sum by allocating power p_i every time slot with the energy constraint as

$$\begin{aligned} \underset{\{p_i\}_{i=0}^{\infty}}{\text{maximize}} \quad & \mathbb{E}\left[\sum_{i=0}^{\infty} \gamma^i r(s_i, p_i) | s_0\right] \\ \text{subject to} \quad & b_{i+1} = m(b_i, p_i, e_i^h), \end{aligned} \tag{3}$$

where $\gamma \in (0, 1)$ is the discount factor. Function $m(b, p, e^h) = \min\left\{b - T_t p + e^h, b_{\max}\right\}$ indicates the causality of the rechargeable battery. When the transmitter uses power p_i at time slot i, the remaining battery b_{i+1} is decreased by $T_t p_i$ from b_i then increases by the harvested energy e_i^h. The maximum power in the remaining battery cannot exceed the maximum capacity of the battery, $b_{\max}$.

III. Value Function Analysis for Transmission Completion Time Minimization

In this section, we prove that the optimal value function, which takes a five-element state as an input, is an increasing function for each element in the system model. The increasing property demonstrated in this section are the basis for our network construction for RL.

As the state space S in the system model is countable and discrete, and the action space $\mathcal{A}$ is finite and countable, there exists an optimal stationary policy $\pi^*(s)$ that maximizes the discounted reward sum [14]. Let us denote the stationary optimal power allocation policy as π^* that maximize the objective function in (3). The optimal value function V^* can be

obtained through the optimal policy that satisfies the Bellman optimality equation [14] which is given as

$$V^*(s) = \max_{p \in P(s)} \Big\{ r(s,p) + \gamma \sum_{\bar{s}} p_s(\bar{s}|s,p) V^*(\bar{s}) \Big\}, \quad (4)$$

where notation $\bar{\cdot}$ means the next timeslot value, i.e., $\bar{s_i} = s_{i+1}$. Time slot notations (i) of all variables are omitted in this section. The stationary policy π^* also satisfies the Bellman optimality equation which is given as

$$\pi^*(s) = \arg\max_{p \in P(s)} \Big\{ r(s,p) + \gamma \sum_{\bar{s}} p_s(\bar{s}|s,p) V^*(\bar{s}) \Big\}. \quad (5)$$

As a consequence, solving the optimization problem (3) is equivalent to finding the optimal value function V^* which satisfies (4) in our system dynamics. The value iteration is well known algorithm that converges to the optimal value function V^* by the contraction dynamic operator, starting with any initial function with discrete action and state spaces and bounded reward function [14]. The update method of the value iteration can be obtained by repeating

$$V_{n+1}(s) = \max_{p \in P(s)} \Big\{ r(s,p) + \gamma \sum_{\bar{s}} p_s(\bar{s}|s,p) V_n(\bar{s}) \Big\}. \quad (6)$$

Note that, if $n \to \infty$, then $V_n \to V^*$. Repeating (6) can be impractical for large state and action spaces since it requires to compute (6) for all $s \in S$. Due to the impracticability, we use the gradient-based method to estimate the optimal value function. However, we exploit the value iteration to extract some features of the optimal value function, which is useful for building function approximators. By using the convergence property of the value iteration which is widely used [15]–[17], the following lemmas can be proved.

Lemma 1: The optimal value function $V^*(j, d, H, e^h, b)$ is increasing in j for any given d, H, e^h, and b.

Proof: Let us assume that V_n is increasing in j for any given d, H, e^h, and b. As the state transition probability p_s is partially deterministic, we can write (6) in detail using the value iteration of

$$V_{n+1}(j, d, H, e^h, b) = \max_{p \in P(s)} \Big\{ r(s,p) + \gamma \mathbb{E}_{\bar{e}^h, \bar{H}} \big[V_n(\bar{j} - 1, \bar{d} + D(H,p), \bar{H}, \bar{e}^h, m(b,p,e^h)) \big] \Big\}. \quad (7)$$

As j is a negative time, it is decremented by 1 at each time step and $\bar{j} = j$. If $j = r_b$, $\bar{j} - 1 = j$. The total amount of data sent in the next state is increased by $D(H,p)$ and if $d + D(H,p) \geq d_o$, $\bar{d} + D(H,p) = d_o$, otherwise, $\bar{d} = d$. The remaining battery amount is added or subtracted by the battery causality. This extension of the value iteration equality can also be applied to j' which is greater than j as

$$V_{n+1}(j', d, H, e^h, b) = \max_{p \in P(s)} \Big\{ r(j', d, H, e^h, b, p) + \gamma \mathbb{E}_{\bar{e}^h, \bar{H}} \big[V_n(\bar{j}' - 1, \bar{d} + D(H,p), \bar{H}, \bar{e}^h, m(b,p,e^h)) \big] \Big\}. \quad (8)$$

Let us denote p^* which is the optimal action for the equation (7), i.e.,

$$p^* = \arg\max_{p} \Big\{ r(s,p) + \gamma \mathbb{E}_{\bar{e}^h, \bar{H}} \big[V_n(\bar{j} - 1, \bar{d} + D(H,p), \bar{H}, \bar{e}^h, m(b,p,e^h)) \big] \Big\}. \quad (9)$$

The optimal action p'^* for the state $s' = (j', d, H, e^h, b)$ in (8) can be represented as

$$p'^* = \arg\max_{p} \Big\{ r(s',p) + \gamma \mathbb{E}_{\bar{e}^h, \bar{H}} \big[V_n(\bar{j}' - 1, \bar{d} + D(H,p), \bar{H}, \bar{e}^h, m(b,p,e^h)) \big] \Big\}. \quad (10)$$

By using (9) and (10), we can get the following

$$V_{n+1}(j', d, H, e^h, b) \quad (11)$$

$$= \max_{p \in P(s)} \Big\{ r(j', d, H, e^h, b, p) + \gamma \mathbb{E}_{\bar{e}^h, \bar{H}} \big[V_n(\bar{j}' - 1, \bar{d} + D(H,p), \bar{H}, \bar{e}^h, m(b,p,e^h)) \big] \Big\} \quad (12)$$

$$= r(j', d, H, e^h, b, p'^*) + \gamma \mathbb{E}_{\bar{e}^h, \bar{H}} \big[V_n(\bar{j}' - 1, \bar{d} + D(H,p'^*), \bar{H}, \bar{e}^h, m(b,p'^*,e^h)) \big] \quad (13)$$

$$\geq r(j', d, H, e^h, b, p^*) + \gamma \mathbb{E}_{\bar{e}^h, \bar{H}} \big[V_n(\bar{j}' - 1, \bar{d} + D(H,p^*), \bar{H}, \bar{e}^h, m(b,p^*,e^h)) \big]. \quad (14)$$

The equality between (12) and (13) holds by definition of p'^*. Note that action p'^* is the optimal action for state $s' = (j', d, H, e^h, b)$. As there is no guarantee that p^* is also the optimal action for state $s' = (j', d, H, e^h, b)$, the inequality between (13) and (14) holds. The reward function r in (2) is clearly an increasing function of j for given d, H, e^h, and b. Consequently, we have

$$r(j', d, H, e^h, b, p^*) + \gamma \mathbb{E}_{\bar{e}^h, \bar{H}} \big[V_n(\bar{j}' - 1, \bar{d} + D(H,p^*), \bar{H}, \bar{e}^h, m(b,p^*,e^h)) \big] \quad (15)$$

$$\geq r(j, d, H, e^h, b, p^*) + \gamma \mathbb{E}_{\bar{e}^h, \bar{H}} \big[V_n(\bar{j} - 1, \bar{d} + D(H,p^*), \bar{H}, \bar{e}^h, m(b,p^*,e^h)) \big] \quad (16)$$

$$= V_{n+1}(j, d, H, e^h, b). \quad (17)$$

The inequality (15) and (16) holds by the increasing property of r for j and assumption that V_n is an increasing function for j. The equality between (16) and (17) holds by definition of the $(n+1)$th value function (7). By mathematical induction, we can conclude that if V_n is an increasing function for j, then V_{n+1} is still an increasing function for j. Regardless of the initial value function V_0, the value iteration algorithm (6) make the value function converge to the optimal value function [14]. V_0 can be initialized as an increasing function for j, and $\lim_{n \to \infty} V_n = V^*$ is still an increasing function for j. ■

Lemma 2: The optimal value function $V^*(j, d, H, e^h, b)$ is increasing in d for any given j, H, e^h, and b.

Proof: Let us assume that V_n is increasing in d for any given j, H, e^h, and b. By using the definition of p^* and $d'(> d)$, we have

$$V_{n+1}(j, d', H, e^h, b) \tag{18}$$
$$= \max_{p \in P(s)} \Big\{ r(j, d', H, e^h, b, p) + \gamma \mathbb{E}_{\bar{e}^h, \bar{H}} \big[V_n(\bar{j} - 1, \bar{d}' + D(H, p), \bar{H}, \bar{e}^h, m(b, p, e^h)) \big] \Big\} \tag{19}$$
$$\geq r(j, d', H, e^h, b, p^*) + \gamma \mathbb{E}_{\bar{e}^h, \bar{H}} \big[V_n(\bar{j} - 1, \bar{d}' + D(H, p^*), \bar{H}, \bar{e}^h, m(b, p^*, e^h)) \big] \tag{20}$$
$$\geq r(s, p^*) + \gamma \mathbb{E}_{\bar{e}^h, \bar{H}} \big[V_n(\bar{j} - 1, \bar{d} + D(H, p^*), \bar{H}, \bar{e}^h, m(b, p^*, e^h)) \big]$$
$$= V_{n+1}(j, d, H, e^h, b). \tag{21}$$

The inequality between equations (18) and (20) holds because p^*, which is the optimal action for state $s = (j, d, H, e^h, b)$ does not guarantee optimality in state (j, d', H, e^h, b). Note that r is an increasing function of d for any given j, H, e^h, b, and p. By the increasing properties of r and V_n for d, the inequality between (20) and (21) is satisfied. The increasing property of the value function V_n for d does not disappear by the value iteration. As in Lemma 1, by mathematical induction, V^* is still an increasing function for d. ■

Since the following lemmas can be proved in the similar manners as in Lemmas 1 and 2, we omit the proofs.

Lemma 3: The optimal value function $V^*(j, d, H, e^h, b)$ is increasing in H for any given j, d, e^h, and b.

Lemma 4: The optimal value function $V^*(j, d, H, e^h, b)$ is increasing in b for any given j, d, H, and e^h.

Lemma 5: The optimal value function $V^*(j, d, H, e^h, b)$ is increasing in e^h for any given j, d, H, and b.

Finally, we prove that the optimal action-value function, $Q^*(s, p) = r(s, p) + \gamma \sum_{\bar{s}} p_s(\bar{s}|s, p) V^*(\bar{s})$, is a partially and monotonically increasing function for s.

Theorem 1: The optimal action-value function Q^* is an increasing function for j, d, H, e^h, and b. In other words, the optimal action-value function is an increasing function of some of its variables (j, d, H, e^h, and b) but not all (p).

Proof: Combining Lemma 1, 2, 3, 4, and 5 proves that the optimal value function V^* is an increasing function for all variables in the input elements j, d, H, e^h, and b. As we mentioned above, the transition probability is partially deterministic. The optimal action-value function can be expanded as

$$Q^*(s, p) = r(s, p) + \gamma \mathbb{E}_{\bar{e}^h, \bar{H}} \big[V^*(\bar{j} - 1, \bar{d} + D(H, p), \bar{H}, \bar{e}^h, m(b, p, e^h)) \big]. \tag{22}$$

Due to the increasing property of the optimal value function V^*, we have

$$Q^*(s, p) \leq r(\check{j}, \check{d}, \check{H}, \check{e}^h, \check{b}, p) + \gamma \mathbb{E}_{\bar{e}^h, \bar{H}} \big[V^*(\check{\bar{j}} - 1, \check{\bar{d}} + D(\check{H}, p), \bar{H}, \bar{e}^h, m(\check{b}, p, \check{e^h})) \big]. \tag{23}$$

where checked variables with $\check{}$ are equal to or greater than non-checked variables. Consequently, for given p, we have

$$Q^*(s', p) \geq Q^*(s, p) \quad \forall s', s \in S \tag{24}$$

where s_n is nth element of vector s and $s'_n \geq s_n$ for all $1 \leq n \leq 5$. ■

IV. Gradient-based Q-Learning with Partially Monotonic Network

1) Partially Monotonic Network: We use a lattice network [18], which is a interpolated multidimensional look-up table, to reflect a partially and monotonically increasing behavior of the optimal action-value function we proved in the previous section. The action value function is represented by the partially monotonic lattice, which is defined as

$$Q(s, p) = \theta^T \phi(s, p) \tag{25}$$

where θ is the vector of the lattice parameters and $\phi(s, p)$ is the interpolation function with a 5-spaced linear calibration layer [18, Sec. 9]. Let us denote the number of vertices in the lattice network along the dth feature by v_d ($1 \leq d \leq 6$). Then the total number of the vertices in the lattice network is $\Pi_d v_d$. The set of the lattice parameter θ is $\Pi_d v_d$ dimensional vector. Each lattice parameter in θ corresponds to a vertex in the lattice network and can be updated. When the lattice network takes inputs in-between the vertices, the output value is interpolated with the simplex interpolation method [18]. We build our partially monotonic lattice network as a single framework for fast learning and the network is forced monotonicity only for the j, d, H, e^h, and b with a linear transformation layer for normalization.

2) Gradient-based Q-Learning with function approximator: Although information is known by the transmitter that the channel or energy arrival is i.i.d., the exact distributions are known very rarely. Therefore, we use the Q-learning [9] that updates the action-value function $Q(s, p)$ to the optimal action-value function $Q^*(s, p)$ even without the information on the state transition probability p_s. Due to the impracticality of the traditional action-value function update process, we use a gradient-based Q-learning techinique. As we mentioned in the previous subsection IV-1, the lattice network enforce the monotonicity on its input variables except p. The partial monotonicity of the lattice network enables robust learning in the random wireless communication environments, since the action-value function (the lattice network) have the increasing property of the optimal action-value function from the beginning of the learning. We build one more lattice network $\theta'^T \phi'(s, p)$ called the target network to overcome

the instability of gradient update and adopt the double Q-learning update technique [19]. The parameters of the lattice network $\theta^T\phi(s,p)$ are updated with the learning rate α by a gradient descent technique in the direction that minimizes the loss function L as

$$L = \mathbb{E}[(y_i - \theta^T\phi(s_i,p_i))^2], \tag{26}$$

$$y_i = r_i + \gamma\theta'^T\phi'(s_{i+1}, \arg\max_p \theta^T\phi(s_{i+1},p)). \tag{27}$$

To remove the correlation between the sampled data used to approximate (26), transition information of each step is stored in the finite experience replay buffer and then randomly sampled [11]. The parameters in the lattice network are updated with the following constraint

$$\theta^T\phi(j,d,H,e^h,b,p) \leq \theta^T\phi(\check{j},\check{d},\check{H},\check{e^h},\check{b},p) \tag{28}$$

due to the partial increasing property of the optimal action-value function in Theorem 1. In order to take into consideration the balance between exploration and exploitation, the learning agent follows the greedy policy

$$p_i = \text{argmax}_p\theta^T\phi(s_i,p) \tag{29}$$

from the evaluation lattice network with probability $1-\epsilon$ and takes action randomly from $P(s_i)$ with probability ϵ. Since the transmitter knows H_i, if there is a candidate set of p_i that can complete the packet transmission at time slot i, the smallest p_i value is selected from the set. The learning rate α is decayed by a factor of α_d every step. Algorithm 1 shows the double Q-learning scheme with the partially monotonic lattice network.

Algorithm 1 Partially Monotonic Lattice with Q-Learning

```
Set hyperparameters for partially monotonic lattice network
for e = 0, N_e do
    Get initial state s_{i=0}
    while d_0 is not transmitted do
        Select action p_i with the ε-greedy method
        Apply action p_i to environment and get r_i with s_{i+1} (2)
        Store data (s_i, p_i, r_i, s_{i+1}) in replay memory
        Update the parameters in the lattice network in direction of minimizing L with the monotonicity constraint (28) by using batch sampled from replay memory
        s_i ← s_{i+1} and α ← α × α_d
        if (d_0 is transmitted) then
            θ' ← θ, φ' ← φ
            End episode
        end if
    end while
end for
```

V. Numerical Results

A. Simulation Environments

The bandwidth is set as $W = 1$ MHz with $N_0 = 10^{-20}$ W/Hz. Under the Rician fading assumption, H has a mean of 2×10^{-13} and a standard deviation of 2×10^{-14}. The energy arrival distribution is assumed to be a uniform distribution from 0 to $e^h_{\max}$, and discretized in units of 5×10^{-7}J. The maximum battery capacity is set as 5×10^{-6}J, and the packet size is $d_o = 150$ (bits). $\mathcal{B}$ and $\mathcal{A}$ are discretized in unit of 1×10^{-6}J.

To measure the general performance of each algorithm, all algorithms were trained through the 100 training datasets which are randomly generated. The random episodes were created under the same state transition probability, and we also generated the 100 episodes to measure the validation performance. During the validation phase, the learning agent does not follow the random exploration with probability ϵ. We halt the learning process if there is no validation performance improvement in 10 episodes with $\epsilon < 0.03$.

We compare the performance of our proposed method with other power allocation policies, which are described below.

1) Offline Policy: It is assumed that the battery size is *infinite* and the transmitter knows all future channel and harvested energy state transitions. This ideal offline assumption contrasts with the online scenario where only the causal channel gains and harvested energy arrivals can be observed. In this ideal situation, the transmitter optimally uses its energy to minimize the transmission completion time and the performance is an upper bound for the various online schemes.

2) Partially Monotonic Lattice with Q-Learning (LQL): This is the method we propose. We use a partially monotonic lattice network to reflect the partially and monotonically increasing tendencies of the optimal action-value function. The partially monotonic lattice network θ is constructed with $v_d = 2\ \forall d$. We use the Adam optimizer for learning with a learning rate of 1×10^{-3} and a decay α_d of 0.99. The batch of size 64 is made from the replay buffer of size 4,096. All input variables for the lattice network are normalized. For the ϵ-greedy action selection, we set the initial value of ϵ as 1.0 and multiply it by 0.995 every time slot.

3) Lattice Q-Learning without Partial Monotonicity (LQL-NC): To observe how large the performance improvement is, by forcing the network to have partial increasing behaviors, we test this algorithm that uses the same setting as LQL but does not have the partially increasing characteristics.

4) Greedy Policy: This policy uses $p_i = b_i/T_t$ for all i to achieve the maximum transmission rate at time slot i and avoid the battery overflow. The average performance of the greedy policy is considered as the baseline for the problem.

5) Random Policy: The transmitter chooses a transmit power value randomly from $\mathcal{B}$ every time slot i with the same probability. If the transmit power value selected by the transmitter is greater than b_i/T_t, the value is clipped to b_i/T_t.

B. Discussion

In Fig. 2, the average performance of the LQL exceeds that of other naive online power allocation policies and the LQL-NC during the learning process. These experimental results show that the exploitation of the increasing property extracted from the desired function of the system model enables the

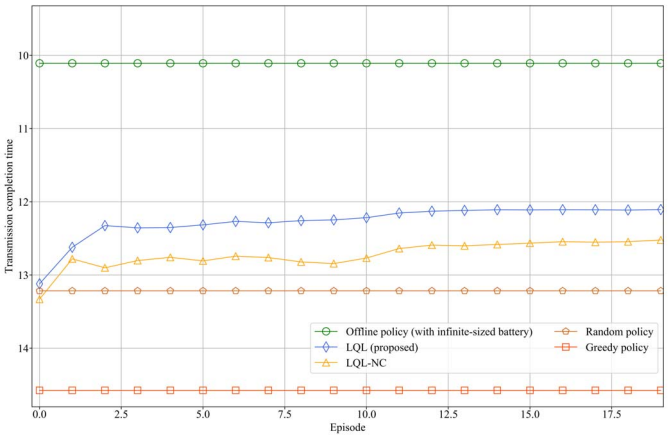

Fig. 2. Transmission completion time according to the episode. All the algorithms are tested on a randomly generated validation dataset which has never been used for the training. $e^h_{\max} = 5e-7$J is assumed.

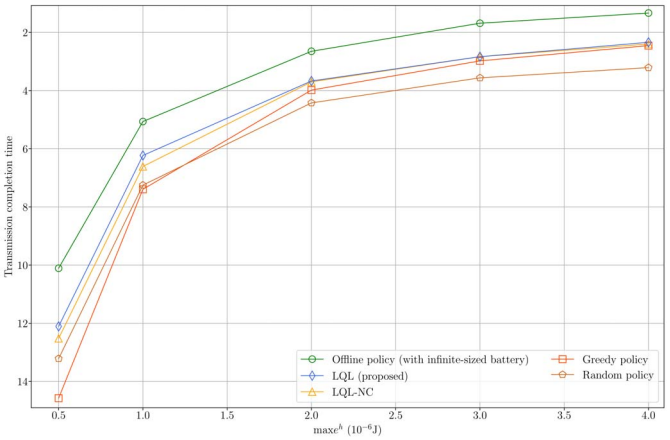

Fig. 3. Transmission completion time according to the maximum amount of energy harvested per second.

agent to avoid converging the suboptimal point. In the course of the learning process, the average performance of the LQL algorithm does not exceed about 13 time slots to complete the packet transmission. After 15 learning episodes, the average performance of the LQL algorithm achieves about 12 time slots for the packet transmission completion. On the other hand, the LQL-NC converges to the suboptimal performance and the greedy policy constitutes the lower bound in this environment. The performance difference between LQL and the offline policy is much smaller than that of other naive algorithms and the offline policy.

Fig. 3 shows the measurement of algorithms' performance in different environments for validation. The greater the amount of energy coming in per unit time, the averages of transmission completion times for all the algorithms decrease gradually. In all the tested environments, the performance of the LQL exceeds that of all the algorithms except the performance of the offline policy. The greater the amount of energy that can be collected relative to the amount of battery, the closer the performances of the learning-based algorithms are to the performance of the greedy algorithm, because they consume more power in each step to avoid the battery overflow. In such an environment, the random policy performs worse than the greedy policy because the battery overflow happens due to the randomness.

VI. Conclusion

In this paper, we proposed a novel power allocation technique based on the reinforcement learning method to minimize the transmission completion time for the time-varying channel in energy harvesting communications. We proved that the action-value function is a partially increasing function of negative time, transmitted data, channel gain, harvested energy, and remaining battery. This proof provided a basis for constructing the partially monotonic lattice network that is optimized for the desired action-value function. The gradient-based Q-learning technique with the optimized lattice network overcame the severe randomness of the energy harvesting communications systems and achieved low delay. Through the performance comparisons, we show that our approach with the shape constraints for the function approximator according to the proven system characteristics outperforms the existing approaches and it is able to achieve performance closer to that of the offline policy.

Acknowledgement

This work is in part supported by Basic Science Research Program (NRF-2017R1A2B2007102, NRF-2018R1C1B5085940, and NRF-2019R1C1C1006806) through NRF funded by MSIT, Technology Innovation Program (10051928) funded by MOTIE, Bio-Mimetic Robot Research Center funded by DAPA (UD130070ID), Samsung Electronics AI Grant, MSIT-IITP grant (No.2019-0-01367, BabyMind), INMAC, and BK21-plus.

References

[1] J. Yang and S. Ulukus, "Optimal packet scheduling in an energy harvesting communication system," *IEEE Transactions on Communications*, vol. 60, no. 1, pp. 220–230, 2011.

[2] A. Arafa, T. Tong, M. Fu, S. Ulukus, and W. Chen, "Delay minimal policies in energy harvesting communication systems," *IEEE Transactions on Communications*, vol. 66, no. 7, pp. 2918–2930, 2018.

[3] O. Ozel, K. Tutuncuoglu, J. Yang, S. Ulukus, and A. Yener, "Transmission with energy harvesting nodes in fading wireless channels: Optimal policies," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 8, pp. 1732–1743, 2011.

[4] F. M. Ozcelik, G. Uctu, and E. Uysal-Biyikoglu, "Minimization of transmission duration of data packets over an energy harvesting fading channel," *IEEE Communications Letters*, vol. 16, no. 12, pp. 1968–1971, 2012.

[5] K. Chi, Y.-H. Zhu, Y. Li, L. Huang, and M. Xia, "Minimization of transmission completion time in wireless powered communication networks," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1671–1683, 2017.

[6] F. Shan, J. Luo, W. Wu, and X. Shen, "Delay minimization for data transmission in wireless power transfer systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 2, pp. 298–312, 2018.

[7] Z. Ding, D. W. K. Ng, R. Schober, and H. V. Poor, "Delay minimization for noma-mec offloading," *IEEE Signal Processing Letters*, vol. 25, no. 12, pp. 1875–1879, 2018.

[8] X. Zheng, S. Zhou, and Z. Niu, "On the online minimization of completion time in an energy harvesting system," in *2016 14th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*. IEEE, 2016, pp. 1–8.

[9] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[10] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Advances in neural information processing systems*, 2000, pp. 1057–1063.

[11] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

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

[13] D. M. Hawkins, "The problem of overfitting," *Journal of chemical information and computer sciences*, vol. 44, no. 1, pp. 1–12, 2004.

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

[15] T. Zhang, W. Chen, Z. Han, and Z. Cao, "A cross-layer perspective on energy-harvesting-aided green communications over fading channels," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 4, pp. 1519–1534, 2014.

[16] S. Mao, M. H. Cheung, and V. W. Wong, "Joint energy allocation for sensing and transmission in rechargeable wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 6, pp. 2862–2875, 2014.

[17] H. Kim, W. Shin, H. Yang, N. Lee, and J. Lee, "Rate maximization with reinforcement learning for time-varying energy harvesting broadcast channels," in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.

[18] M. Gupta, A. Cotter, J. Pfeifer, K. Voevodski, K. Canini, A. Mangylov, W. Moczydlowski, and A. Van Esbroeck, "Monotonic calibrated interpolated look-up tables," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 3790–3836, 2016.

[19] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Thirtieth AAAI conference on artificial intelligence*, 2016.