Optimal Transmission Policy in Energy Harvesting Wireless Communications: A Learning Approach

Keyu Wu, Chintha Tellambura, and Hai Jiang
Department of Electrical and Computer Engineering
University of Alberta, Edmonton, Alberta T6G 1H9, Canada

***Abstract*—We consider an energy harvesting wireless communication link, where arriving data packets have different importance values. The wireless transmitter needs to decide whether each arriving data packet should be transmitted or not, based on the packet's importance value, channel condition, and energy status. Under certain conditions, we show this high dimensional control problem can be transformed to a one dimensional continuous value function estimation problem using the notion of after-state. Then, by analyzing the structure of the value function, we propose a polynomial approximation to effectively compress the continuous function space into a finite weight space. Furthermore, we develop a reinforcement learning algorithm for our after-state setting. Finally, the proposed function approximation and learning algorithm are investigated under various system parameter settings via simulation.**

***Index Terms*—Energy harvesting, data importance, after-state, reinforcement learning, function approximation.**

I. Introduction

Energy harvesting (EH) wireless devices are designed to collect energy from ambient environments, such as solar energy, indoor illumination, vibration, and others [1]. This ability promises the green evolution of future wireless communication networks, where due to the self-sufficiency of energy, the life time of the system is constrained by the limits of hardware, rather than by the battery life time. In the literature, power allocation in wireless communication networks with constant power supply has been widely investigated [2]–[5], where one major focus is to deal with time-varying wireless channels. In EH wireless systems, the amount of harvested energy randomly changes over time [1]. Thus, system performance optimization is challenging to achieve when considering both the time-varying wireless channels and harvested energy.

In this research, we develop an adaptive learning algorithm to address the above challenge in an EH wireless system where different data packets have different *importance* values[1]. Power allocation and energy management in EH wireless systems for the transmission of data packets with different importance values have been investigated in [6]–[9]. In [6], the amount of energy replenishment from harvesting sources and energy consumption for each packet transmission are both assumed to be one quantum, and thus, the battery dynamic is modeled as an N-state Markov chain. The optimal policy is proven to be threshold based for different importance value given certain battery level. With the same assumption, works in [7] and [8] show that a heuristic policy with single threshold for data importance value regardless of the battery's level can achieve performance comparable to that of the optimal policy, but with less complexity. However, works in [6]–[8] do not consider the wireless fading in decision making. Wireless fading is considered in a recent work [9], which assumes that amounts of energy replenishment from harvesting and energy consumption for each packet transmission are general random variables (r.v.s). The optimal policy and low-complexity heuristic policy are developed. However, the general r.v.s are quantized, which results in the curse of dimensionality [10, p. 201] and may lead to slow learning speed.

In this paper, optimal transmission decisions are made based on the amount of harvested energy, data importance, wireless channel state and the residual batter energy. We model the problem as a continuous state Markov decision process (MDP). This is a high dimensional problem with multiple continuous r.v.s. By exploiting the notion of the after-state, we reduce the dimension of the problem to be one (i.e., only the battery dimension). Furthermore, by analyzing the structure of optimal value function and policy, we propose a linear function approximation with polynomial basis functions to effectively deal with continuous state space and to accelerate the learning speed. Finally, based on the approximation, we develop a policy iteration based algorithm to achieve near optimal control.

The rest of paper is organized as follows. Section II describes the system model. The optimal control problem is formulated as an MDP in Section III, and it is transformed into after-state setting in Section IV. The after-state value function and policy are analyzed in Section V. An after-state reinforcement learning (RL) algorithm is developed in Section VI, and its performance is investigated in Section VII. Section VIII concludes the whole paper.

II. System Model

We consider a communication link with harvested energy supply (Fig. 1). The transmitter works in a time-slotted manner, and each arriving data packet, which is with fixed length, triggers a time slot. Note that, for the case where the interval of two successive data packets is random, the duration of each time slot is random. Different arriving data packets

This work is jointly supported by China Scholarship Council/University of Alberta Scholarship, Alberta Innovates Graduate Student Scholarship and Natural Sciences and Engineering Research Council of Canada.

[1]For example, in wireless sensor networks, packets containing measurement values that far deviate from the mean value are considered to be more important.

may have different importance values, denoted as D, which is assumed to be an independent and identically distributed (i.i.d.) r.v. with probability density function (pdf) $\mathrm{f}_D(d)$. We assume that $\mathbb{E}[D]$ and $\mathbb{E}[D^2]$ exist and are bounded, where $\mathbb{E}[\cdot]$ means expectation. Harvested energy during last time slot is assumed to arrive as an energy package at the beginning of current time slot. The amount of energy in each energy package is denoted as E, and is modeled as an i.i.d. r.v. with pdf $\mathrm{f}_E(e)$. Via the receiver's feedback, the transmitter knows the current channel state. Depending on the modulation and coding scheme used, the transmitter can calculate the amount of energy needed to guarantee one successful transmission, which is an i.i.d. r.v., denoted as H, with pdf $\mathrm{f}_H(h)$. And the amount of energy remaining in battery is denoted as B. The maximum value of B is limited by battery capacity $B_{\max}$. Note that the transmitter does not know the distributions $\mathrm{f}_E(e)$, $\mathrm{f}_D(d)$, and $\mathrm{f}_H(h)$.

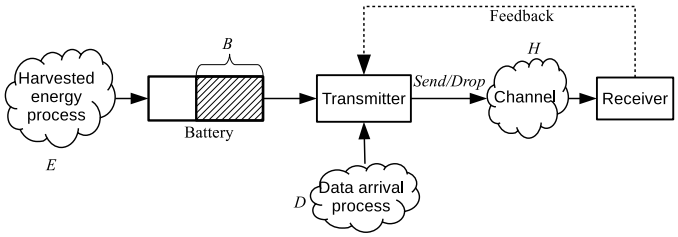

Fig. 1. Components of the investigated system

At time slot t,[2] based on b_t (the remaining energy in battery), e_t (harvested energy), d_t (the importance value of data packet), and h_t (the amount of energy needed for transmission), the transmitter needs to decide to transmit or not. When the transmitter decides to send, i.e., action at slot t is $a_t = 1$, it makes one successful transmission, if current available energy is sufficient, i.e., $b_t + e_t \geq h_t$; however, the attempt of sending will result in transmission failure and waste of energy, if $b_t + e_t < h_t$. When the transmitter decides to drop current packet, i.e., $a_t = 0$, the harvested energy will be stored, which increases the level of battery, and therefore, one successful transmission in next time slot is more likely. At each time slot, the transmitter must intelligently decide the transmission action, in order to have the most efficient use of energy and therefore maximize the sum of importance values of successfully transmitted data packets.

III. Problem formulation

A. MDP model

We use an infinite time horizon discounted continuous state Markov decision process (MDP) to formulate the above problem by defining 4-tuple $< \mathbb{S}, \mathbb{A}, p, r >$, namely state space, action space, state transition kernel and reward associated with each state-action pair:

1) Each state $s \in \mathbb{S}$ is defined as the composition of four variables $[b, e, h, d]$, in which b, e, h, d are battery level at the beginning of a slot, harvested energy, energy needed for a successful transmission, and importance value of the current packet, respectively;

2) The action $a \in \mathbb{A}$ available at each state is binary, with '1' means to send, and '0' means to drop;

3) The state transition kernel $p(\cdot|s, a)$ is a pdf over state space $\mathbb{S}$ given state action pair $(s, a) \in \mathbb{S} \times \mathbb{A}$. For $s' = [b', e', h', d'] \in \mathbb{S}$,[3] $p(s'|s, a)$ is defined as the pdf to state s' given initial state s and action a, expressed as

$$p(s'|s, a) \triangleq \mathrm{f}_E(e') \cdot \mathrm{f}_H(h') \cdot \mathrm{f}_D(d') \cdot \delta(b' - \varrho(s, a)), \quad (1)$$

where $\delta(\cdot)$ is Dirac Delta function and $\varrho(s, a)$ is the remaining battery energy level after action a for initial state s, expressed as:

$$\varrho(s, a) \triangleq \min\{\max\{b + e - h \cdot a, 0\}, B_{\max}\}. \quad (2)$$

4) The reward $r(s, a)$ is the immediate reward associated with each state action pair, defined

$$r(s, a) \triangleq \mathbb{1}(a = 1) \cdot \mathbb{1}(b + e \geq h) \cdot d, \quad (3)$$

where $\mathbb{1}(\cdot)$ is the indicator function.

B. Classical formulation for controlling

To achieve the optimal control of MDP, it is sufficient to consider stationary deterministic policies Π [11], which are stationary mappings from state space to action space. The optimal Bellman equation, which works at the central role in classical MDP theory, is defined as follows,

$$V(s) = \max_a \{r(s, a) + \gamma \cdot \mathbb{E}[V(S')|s, a]\}, \quad (4)$$

where the constant $\gamma \in [0, 1)$ is the discounting factor and r.v. S' means the next state after state s with an action. The solution to (4), denoted as V^*, can be used to define an optimal policy π^* as follows [11, p. 154],

$$\pi^*(s) = \arg\max_a \{r(s, a) + \gamma \cdot \mathbb{E}[V^*(S')|s, a]\}. \quad (5)$$

The optimality of π^* is in the sense that starting from any state $s \in \mathbb{S}$, no policy can get larger discounted accumulated reward than π^* does. Formally speaking, for any $\pi \in \Pi$, we can define the value function $V^\pi(s)$ for any state s as

$$V^\pi(s) \triangleq \mathbb{E}\left[\sum_{\tau=t}^{\infty} \gamma^{\tau-t} r(s_\tau, a_\tau)|s_t = s, a_\tau \in \pi(s_\tau)\right], \quad (6)$$

and $V^{\pi^*}(s) = \sup_{\pi \in \Pi}\{V^\pi(s)\}$. Furthermore, it can be shown that $V^*(s) = V^{\pi^*}(s)$ for $\forall\, s$ [11, p. 152]. Therefore, we use $V^*(s)$ and $V^{\pi^*}(s)$ interchangeably. It is easy to check that (6) can be recursively written as

$$V^\pi(s) = r(s, \pi(s)) + \gamma \cdot \mathbb{E}[V^\pi(S')|s, \pi(s)], \quad (7)$$

which will be used in the transforming to after-state formulation in Section IV.

[2]Throughout this paper, the subscript t denotes time slot index.

[3]Throughout the paper, x' means notation x for the state in next time slot.

In our setting where the transmitter does not know the distributions $\mathrm{f}_E(e)$, $\mathrm{f}_D(d)$, and $\mathrm{f}_H(h)$, the main challenges for using (4) and (5) to develop optimal policy are as follows. First, the $\max\{\cdot\}$ operation outside of $\mathbb{E}[\cdot]$ operation in (4) will impede us to use samples to estimate V^*. Second, $\mathbb{E}[\cdot]$ operation for the action selection in (5) will impede us to get the optimal action, even if V^* is known.

In the next section, we will address these challenges by transforming the standard MDP formulation into after-state formulation.

IV. Transform to After-state MDP

A. The notion of after-state

The notion of after-state is a trick in RL for the learning tasks of playing games [10, p. 145]. For example, when playing chess, the learner can deterministically control its move, and what is random is the opponent's action. Before a move needed to be decided, the learner is facing certain positions of pieces on chessboard, which is in the same sense of "state" of classical MDP formulation. The after-state for a move of the learner is defined as the resulted positions on chessboard after this move but before the opponent's move. If we can learn the chance of winning for all different after-states, we can use these values to behave optimally: we simply choose the action whose after-state has the highest chance of winning.

This notion can be applied in our problem, where, at each time slot, the control of battery level at the next time slot is deterministic for both $a = 1$ and $a = 0$, and what is random at the next time slot is the exogenous r.v.s E, H and D. Because of the i.i.d. assumptions of the exogenous r.v.s, at time slot t, knowing the battery level resulted from action a_t applied on state s_t, (s_t, a_t) has nothing to do with the future decision making. Therefore, we define the battery level $\varrho(s, a)$ after action a applied on state s as the after-state of (s, a). For presentation clarity, we use β to denote the battery level whenever referring it as the after-state from certain state action pair (s, a). If we can know the maximum expected total rewards that one can get starting from any battery level β, denoted as $J^*(\beta)$, the optimal decision at state s is the action a that maximizes $r(s, a) + J^*(\varrho(s, a))$.

B. After-state formulation for controlling

Before delving into the solving of $J^*(\beta)$, we formalize the sense of maximization of $J^*(\beta)$. We define the value function $J^\pi(\beta)$ of after-state β as the expected accumulated discounted reward one can obtain, if equipping battery level β at the end of time t and following policy π starting from time $t + 1$. Comparing with the definition of V^π in (6), the relationship of J^π and V^π can be easily shown as follows,

$$J^\pi(\beta) = \gamma\mathbb{E}\left[V^\pi(S')|\beta\right] = \gamma\mathbb{E}\left[V^\pi(\beta, D', H', E')\right], \quad (8)$$

where γ is introduced for discounting, since the expectation is starting from next time slot. And plug (7) into (8), we can get the recursive equation respected to J^π as follows:

$$J^\pi(\beta) = \gamma\mathbb{E}\left[r(S', \pi(S')) + J^\pi(\varrho(S', \pi(S')))\right], \quad (9)$$

where $S' = [\beta, E', H', D']$.

In order for solving $J^*(\beta)$, we define the optimal Bellman equation as

$$J(\beta) = \gamma\mathbb{E}\left[\max_{a'}\{r(S', a') + J(\varrho(S', a'))\}\right], \quad (10)$$

with $S' = [\beta, E', H', D']$.

Theorem 1. *There is a unique J^* that satisfies (10), and we have $J^*(\beta) = \sup_\pi\{J^\pi(\beta)\}$ for all β, and furthermore from J^* we can define an optimal policy:*

$$\pi^*(s) = \arg\max_a\{r(s, a) + J^*(\varrho(s, a))\}, \quad (11)$$

such that $J^{\pi^*}(\beta) = J^*(\beta)$ for all β. Finally J^* can be calculated via value iteration algorithm, i.e., with J_0 being arbitrary bounded function, the sequence of functions $\{J_l\}_{l=0}^N$ defined by the following iteration equation

$$J_{l+1}(\beta) \leftarrow \gamma\mathbb{E}\left[\max_{a'}\{r(S', a') + J_l(\varrho(S', a'))\}\right], \quad (12)$$

converges to J^* when $N \to \infty$.

Proof is omitted due to the space limitation.

C. Optimality of V^* from J^*

We will now establish the relationship between V^* and J^*. Assume V^* exists, and define a function G over after-state space as: $G(\beta) \triangleq \gamma\mathbb{E}\left[V^*(\beta, D', H', E')\right]$. Expanding V^* using (4), we have $G(\beta) = \gamma\mathbb{E}\left[\max_{a'}\{r(S', a') + G(\varrho(S', a'))\}\right]$, with $S' = [\beta, E', H', D']$. And this is nothing, but exactly the optimal Bellman equation defined in (10). And according to Theorem 1, we must have $G = J^*$. Then we have $V^*(s) = \max_a\{r(s, a) + J^*(\varrho(s, a))\}$. Therefore, the existence of J^* implies the existence of V^*.

Note that in (10), the $\mathbb{E}[\cdot]$ operation is outside a $\max\{\cdot\}$ operation, which enables the possibility of sample estimation, and in (11), the action selection does not need the knowledge of $\mathrm{f}_E(e)$, $\mathrm{f}_D(d)$, and $\mathrm{f}_H(h)$, if J^* is known. Furthermore, compared with V^*, working with J^* considerably reduces the amount of space for representing a value function, which can save computation resources of solving the problem.

Therefore, transforming the standard MDP formulation into after-state MDP setting is not only useful in establishing the theoretical result, but also beneficial in achieving much lower computation complexity for practical purposes.

V. Analysis of Optimal Value and Policy

In this section, we will present several analytical results regarding the optimal value function and optimal policy. They help us to understand the problem and also serve as the guidance for us to develop an efficient function approximation. The proof of the rest analytical results is omitted due to space limitation.

Theorem 2. *The optimal after-state value J^* is monotone non-decreasing respected to battery level β. The optimal state*

value V^* is monotone non-decreasing respected to b, e, d and $-h$.

Theorem 2 confirms our intuitive understanding that higher battery level, more harvested energy, higher data importance and better channel condition correspond to a "good" state.

Theorem 3. *The optimal policy π^* is threshold based non-decreasing respected to d, i.e., given any b, h and e, if $\pi^*(b, \underline{d}, h, e) = 1$, then $\pi^*(b, \overline{d}, h, e) = 1$, for any $\overline{d} \geq \underline{d}$.*

Theorem 3 shows that the optimal policy is non-decreasing threshold based respected to d. However, the threshold-based property does not necessarily hold for b or e or $-h$.

Theorem 4. *The $J^*(\beta)$ is continuous function respected to β, if D, H, E are all continuous r.v.s.*

VI. After-state reinforcement learning

A. Polynomial basis linear function approximation

Choosing an appropriate function approximation is crucial for the successful application of RL algorithms when facing a continuous after-state space. Among different approximations, linear function approximation, where the value function is approximated as a sum of weighted basis functions, is of our interest because of the simplicity of the structure and the powerful representation ability. Well designed basis functions can not only reduce continuous state space into much smaller weight space, but also provide experience sharing among different states that have weights in common to efficiently accelerate the learning speed.

When facing continuous variables, quantization or state aggregation is often used in the EH literature [9], [12], [13], where the continuous variable space is discretized or aggregated into several non-overlapped intervals or clusters, and variables inside the same interval/cluster are considered as the same and have identical value. Although this treatment is simple, the resulted learning algorithm can be very slow, because learning experience only shares inside the interval/cluster, which is known to have poor generalization ability. Therefore, when certain range of variables are seldom reached during the learning process, the learning of value function of the corresponding interval/cluster can be problematic.

For better generalization ability, a more sophisticated approximation is thus needed. In this paper, we propose a polynomial approximation:

$$\hat{J}(\beta) =< \mathbf{w}, \beta^{\mathbf{p}} > \triangleq w_0 + \sum_{i=1}^{N} w_i \beta^{p_i}, \tag{13}$$

where N is the number of polynomials, $\mathbf{w} = [w_0, ..., w_N]$ are weights with $w_i \in \mathbb{R}$ for $i \in \{0, 1, ..., N\}$ and $\mathbf{p} = [p_0, ..., p_N]$ are polynomial power with $p_i \in \mathbb{R}$ for $i \in \{1, 2, ..., N\}$ and $p_0 = 0$. In Section V, we have showed that J^* is a non-decreasing function, and when the involving r.v.s are continuous, J^* is, in addition, a continuous function. With carefully chosen hyper-parameter $\mathbf{p}$, both monotonicity and continuity can be well presented by our proposed polynomial approximation via optimizing $\mathbf{w}$.

B. Approximate policy improvement

Given J^{π_k} for any policy π_k, we can generate another policy π_{k+1} from J^{π_k} in a greedy manner,

$$\pi_{k+1}(s) = \arg\max_a \{r(s, a) + J^{\pi_k}(\varrho(s, a))\}, \ \forall s. \tag{14}$$

It was showed in [10, p. 82] that π_{k+1} is better than π_k in the sense: $J^{\pi_{k+1}}(\beta) \geq J^{\pi_k}(\beta)$ for all β, which is known as policy improvement property, and therefore (14) is called policy improvement operation. Note that if policy improvement results in an unchanged policy, i.e. $\pi_{k+1} = \pi_k$, then comparing it with (11), it implies $\pi_k = \pi^*$, and $J^{\pi_k} = J^*$.

However, we need to consider how to represent a policy, which is a mapping from four dimensional continuous state space into binary action space. Therefore, explicitly representing a policy is a challenge. Fortunately, there is an elegant way to use weights to implicitly represent a policy as follows:

$$\pi_{\mathbf{w}}(s) \triangleq \arg\max_a \{r(s, a) + < \mathbf{w}, \varrho(s, a)^{\mathbf{p}} >\}. \tag{15}$$

Noticing the similarity between (14) and (15), if the function $\hat{J}(\beta) =< \mathbf{w}, \beta^{\mathbf{p}} >$ is an approximated value function of some policy π_k, $\pi_{\mathbf{w}}(s)$ is "approximately" better than π_k [14, theorem 3.1].

C. Temporal difference learning

Temporal difference learning (TD) is an incremental weight adjusting algorithm for policy evaluation [15]. Given any policy π, our goal is to find a way to adjust weights $\mathbf{w}$ in order to approximate J^π.

At time t, when facing some state s_t, we take action $a_t = \pi(s_t)$ and get reward r_t and observe the next state s_{t+1}, where the action will be $a_{t+1} = \pi(s_{t+1})$, although it has not happened yet. With this 5-tuple information $(s_t, a_t, r_t, s_{t+1}, a_{t+1})$,[4] the classical TD algorithm can get one adjustment respect to the weights. But because we are dealing with after-states, the 5-tuple needs to be converted into after-state setting. Using (2), we can infer β_t and β_{t+1} from (s_t, a_t) and (s_{t+1}, a_{t+1}), respectively, and using (3) we can infer r_{t+1} from (s_{t+1}, a_{t+1}), although it has not happened yet. Then, we get a 3-tuple $(\beta_t, r_{t+1}, \beta_{t+1})$. At time t, given any $\mathbf{w}_t$, we can evaluate and get the approximated value at β_t and β_{t+1} as $\hat{J}(\beta_t)$ and $\hat{J}(\beta_{t+1})$, respectively. Based on these, we define a scale η_t, called temporal different error, as follows,

$$\eta_t = \gamma \left(r_{t+1} + \hat{J}(\beta_{t+1}) \right) - \hat{J}(\beta_t). \tag{16}$$

Using η_t, we can get one updating respected to $\mathbf{w}$ as follows,

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + \alpha_t \cdot \eta_t \cdot \beta_t^{\mathbf{p}}, \tag{17}$$

where $\alpha_t \in (0, 1)$ is the step size. After this update, setting time to be $t+1$ and starting with s_{t+1}, the weights adjustment

[4] This 5-tuple information (s, a, r, s, a) gives an abbreviation SARSA, and this is how the name of resulted control algorithm comes in the RL literature.

process can go on incrementally. With properly decreasing step size, the limiting point of approximated value function $\hat{J}(\beta)$ approaches the best least square approximation of J^{π} as $t \to \infty$ [15].

D. After-state SARSA algorithm

In the RL field, SARSA is a light-weight incremental algorithm for (near) optimal control via combining approximate policy improvement and TD policy evaluation [10]. With the same spirit, we realize the after-state version SARSA, named as A-SARSA, summarized in Algorithm 1. Note that at time t using TD we have one incremental update respect to weights $\mathbf{w}$ to evaluate current policy (in line 15), and at time $t+1$, the updated weights are immediately used to generate action, and therefore, the policy is iterated (in line 12). Since the weights used to generate policy have not yet accomplished the policy evaluation, the policy improvement property does not necessarily hold for every policy iteration. However, compared with the case where the policy evaluation is fully accomplished, this optimistic policy iteration method is more computationally efficient in practice. However, the theoretic analysis of the algorithm is difficult. The strong convergence result of SARSA is now still an open question in the RL field [10, p. 224]. But it has been proved that SARSA will never diverge [16], and in the worst case it chatters among a group of good policies, and "this kind of solution can be completely satisfactory in practice" [17].

Algorithm 1 A-SARSA algorithm with polynomial approximation

1: **procedure** A-SARSA
2: Randomly initialize $\mathbf{w}_0$
3: Initialize $\mathbf{p}$ to some proper value
4: Starting from some battery level β_0
5: Sample environment and get $[e_0, h_0, d_0]$
6: Get $s_0 = [\beta_0, e_0, h_0, d_0]$
7: Infer $a_0 = \pi_{\mathbf{w}_0}(s_0)$ based on (15)
8: **for** t from 0 to ∞ **do**
9: Apply a_t, get reward r_t and $\beta_t = \varrho(s_t, a_t)$
10: Sample environment and get $[e_{t+1}, h_{t+1}, d_{t+1}]$
11: Get $s_{t+1} = [\beta_t, e_{t+1}, h_{t+1}, d_{t+1}]$
12: Infer $a_{t+1} = \pi_{\mathbf{w}_t}(s_{t+1})$ based on (15)
13: Infer $\beta_{t+1} = \varrho(s_{t+1}, a_{t+1})$ and $r_{t+1} = r(s_{t+1}, a_{t+1})$
14: $\eta_t = \gamma\left(r_{t+1} + \mathbf{w}_t^T \beta_{t+1}^{\mathbf{p}}\right) - \mathbf{w}_t^T \beta_t^{\mathbf{p}}$
15: $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + \alpha_t \cdot \eta_t \cdot \beta_t^{\mathbf{p}}$
16: **end for**
17: **end procedure**

VII. Numerical Simulation

In this section, we investigate the A-SARSA algorithm for different settings, and compare its performance with those of several different function approximations, a greedy policy and a heuristic single threshold policy.

A. Performance under different energy supply rate

We use Shannon's capacity to bridge the relationship between channel state and the energy needed for one transmission, i.e.,

$$H = T\frac{(2^{\frac{R}{B}} - 1)N_0 B}{h_s h_f},$$

where the data packet length T is assumed to be 10 ms; R is the bit rate, and B is the bandwidth, and R/B is assumed to be 2 bit/s/Hz; N_0 is the thermal noise spectral density, and $N_0 B$ is assumed to be -107 dBm; h_s representing path loss is assumed to be -102.45 dB; and h_f describing channel fading is assumed to follow exponential distributed with mean equal to $\mu_h = 2$. And the data importance D is modeled as gamma distributed with shape parameter $D_k = 0.1$ and scale parameter $D_\theta = 10$. And we assume the battery capacity $B_{\max}$ is equal to 10^{-4} Joule. We also model the amount of energy supply E in each time slot as gamma distributed with shape parameter $E_k = 1$, and scale parameter E_θ varying during simulation to evaluate the performance of algorithms under different energy supply rates.

The performance of the proposed polynomial approximation (Poly) is compared with that of three other commonly used function approximations, namely state aggregation (State Aggr), tile coding, and Gaussian radial basis function (GRBF) [10]. Tile coding can be understood as modified state aggregation, where the aggregated groups have certain pattern of overlapping, and the learning experience at an after-state is shared among the corresponding overlapped clusters. For GRBF, value function is approximated by weighted Gaussian shape basis functions, and learning experience at an after-state is shared among all weights in proportion to the value of corresponding basis functions. Therefore, compared with State Aggr, tile coding and GRBF have better generalization ability. We also compare the performance with a greedy policy (Greedy), which always chooses to transmit whenever the energy is sufficient.

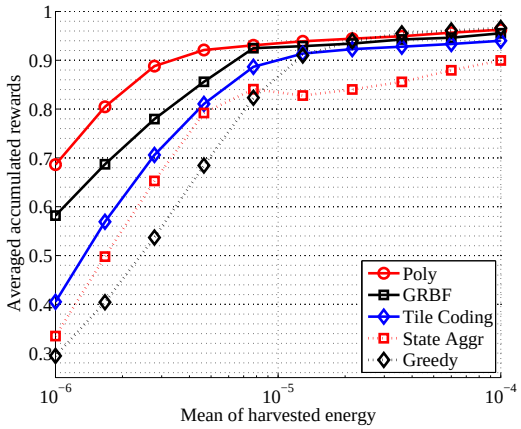

Fig. 2. Average accumulated rewards under different energy supply rate

We vary E_θ, which is equal to the mean of E, from 10^{-6} to 10^{-4} Joule. Fig. 2 shows the accumulated rewards obtained by Greedy and different function approximations during the first 10^5 steps. It is as expected that when the harvested energy supply is low, intelligently saving energy for future use can achieve better performance than Greedy; while when the harvested energy supply is high, simply greedy use of energy is optimal. Because the uniform generalization ability

of polynomial structure, Poly achieves the best performance under all different energy supply rates. GRBF and tile coding have moderate generalization ability, and therefore, show modest performance gain in low energy rate region, and are slightly less powerful than Greedy in high energy region. As the State Aggr has no generalization across the different clusters, it has difficulty to learn the value function in high battery region when energy supply is low, which results in the slowest performance gain. When the energy supply is high, its learned value function in low battery level region is again poor, which results in lots of unnecessarily conservative actions, and therefore, makes it perform even much worse than Greedy.

B. Compared with heuristic single threshold policy

Single threshold policy is often used as a compromised solution when facing the curse of dimensionality [8] [9]. We construct a heuristic single threshold policy (Heu), which chooses to transmit if energy is sufficient, and the importance of data packet is above certain fixed threshold, whose optimal value is found via brute force search. The performance of Heu is compared with Poly and Greedy under different data variances.

We set $E_\theta = 10^{-5}$, and $\mu_h = 1$. And the data importance D is considered to be a mixture of two data classes with equal probability. We model both data classes as gamma distribution, with mean of the first class equal to 10 and variance equal to 1, and mean of the second class equal to 5, and varying its variance for comparison.

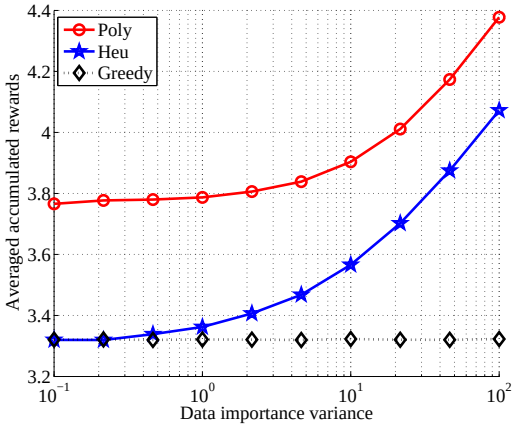

Fig. 3. Average accumulated rewards under different data importance variance of the second data class

The performance for Poly, Greedy, and Heu is shown in Fig. 3, where Greedy is considered as a baseline. It can be seen that the performance gain of Heu heavily depends on the data importance variance. When the variance of the second class data importance is high, both Poly and Heu show high performance gain, and the achieved performance of Heu is approaching Poly. However, when the variance is small, the performance gain of Heu is marginal. When the variance is about 10^{-1}, Heu shows no improvement over Greedy.

VIII. CONCLUSION

In this paper, the optimization of transmission strategy has been studied for energy harvesting communication systems. By exploiting the stochastic structure and after-state notion, we have shown that the optimal control problem whittles down to a one dimensional value function estimation problem, which greatly simplifies problem and increases application flexibility. Based on the analysis of optimal policy and value function, we have proposed a polynomial approximation, which accelerates the learning process compared to commonly used function approximations. Furthermore, with the polynomial approximation, we have developed the A-SARSA learning algorithm via tailoring a classical learning algorithm into after-state setting. Finally, the performance of proposed algorithm has been evaluated comprehensively via simulations.

REFERENCES

[1] S. Ulukus, A. Yener, E. Erkip, O. Simeone, M. Zorzi, P. Grover, and K. Huang, "Energy Harvesting Wireless Communications: A Review of Recent Advances," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 3, pp. 360–381, 2015.

[2] J. Zander, "Performance of optimum transmitter power control in cellular radio systems," *IEEE Trans. Veh. Technol.*, vol. 41, no. 1, pp. 57–62, 1992.

[3] M. Chiang, P. Hande, T. Lan, and C. W. Tan, "Power control in wireless cellular networks," *Found. Trends Netw.*, vol. 2, no. 4, pp. 381–533, 2007.

[4] X. Gong, S. A. Vorobyov, and C. Tellambura, "Optimal Bandwidth and Power Allocation for Sum Ergodic Capacity Under Fading Channels in Cognitive Radio Networks," *IEEE Trans. Signal Process.*, vol. 59, no. 4, pp. 1814–1826, 2011.

[5] Z. Liu, J. Wang, Y. Xia, R. Fan, H. Jiang, and H. Yang, "Power Allocation Robust to Time-Varying Wireless Channels in Femtocell Networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 4, pp. 2806–2815, 2016.

[6] J. Lei, R. Yates, and L. Greenstein, "A generic model for optimizing single-hop transmission policy of replenishable sensors," *IEEE Trans. Wirel. Commun.*, vol. 8, no. 2, pp. 547–551, 2009.

[7] N. Michelusi, K. Stamatiou, and M. Zorzi, "On Optimal Transmission Policies for Energy Harvesting Devices," in *Proc. Infor. Theory Appl. Work.*, 2012, pp. 249 – 254.

[8] M. Nicolo, S. Kostas, and Z. Michele, "Transmission policies for energy harvesting sensors with time-correlated energy supply," *IEEE Trans. Commun.*, vol. 61, no. 7, pp. 2988–3001, 2013.

[9] J. Fernandez-Bes, J. Cid-Sueiro, and A. G. Marques, "An MDP Model for Censoring in Harvesting Sensors: Optimal and Approximated Solutions," *Sel. Areas Commun. IEEE J.*, vol. 33, no. 8, pp. 1717–1729, 2015.

[10] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. Cambridge: Cambridge Univ. Press, 2011.

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

[12] P. Blasco, D. Gunduz, and M. Dohler, "A learning theoretic approach to energy harvesting communication system optimization," *IEEE Trans. Wirel. Commun.*, vol. 12, no. 4, pp. 1872–1882, 2013.

[13] T. Zhang, W. Chen, Z. Han, and Z. Cao, "A Cross-Layer Perspective on Energy-Harvesting-Aided Green Communications Over Fading Channels," *IEEE Trans. Veh. Technol.*, vol. 64, no. 4, pp. 1519–1534, 2015.

[14] M. G. Lagoudakis and R. Parr, "Least-squares policy iteration," *J. Mach. Learn. Res.*, vol. 4, pp. 1107–1149, 2003.

[15] J. N. Tsitsiklis and B. V. Roy, "An Analysis of Temporal-Difference Learning with Function Approximation," *IEEE Trans. Autom. Control*, vol. 42, no. 5, pp. 674–690, 1997.

[16] G. J. Gordon, "Reinforcement learning with function approximation converges to a region," *Adv. Neural Infor. Process. Syst.*, pp. 1040–1046, 2001.

[17] R. S. Sutton, "Open theoretical questions in reinforcement learning," in *Proc. Comput. Learn. Theory*, 1999, pp. 11–17.