Energy Management in Energy Harvesting Wireless Networks: A Reinforcement Learning Framework

Chengrun Qiu, Yang Hu, and Yan Chen
School of Information and Communication Engineering,
University of Electronic Science and Technology of China, Chengdu, Sichuan, China
E-mail: cr_qiu@std.uestc.edu.cn, yanghu@uestc.edu.cn eecyan@uestc.edu.cn

***Abstract*—In this paper, we propose a novel energy management algorithm based on the reinforcement learning to optimize the net bit rate in energy harvesting (EH) networks. By utilizing deep deterministic policy gradient (DDPG), the proposed algorithm is applicable for the continuous states and realizes the continuous energy management. With only one day's real solar data and the simulative channel data for training, the proposed algorithm shows excellent performance in the validation with about 800 days length of real solar data. Compared with the state-of-the-art algorithms, the proposed algorithm achieves better performance in terms of long-term average net bit rate.**

I. Introduction

In the literature, different energy management strategies have been proposed for the energy harvesting wireless communications, and the most typical and effective ones are the water-filling algorithm, Markov decision process (MDP) and Lyapunov algorithm. Water-filling algorithm is based on the convex optimization with the Karush-Kuhn-Tucher (KKT) condition, and is only capable of optimizing the convex objective within a finite horizon. In [1], Ulukus et al. reviewed the water-filling algorithms for different reward optimizations, models and constraints. Ozel et al. in [2] optimized the average throughput with finite epochs of an energy harvesting communication system with the KKT condition.

In order to derive effective online energy management algorithms, MDP has been widely utilized in energy harvesting communications [3], [4], [5], [6]. In [3], Ku et al. optimized the long-term average net bit rate in energy harvesting communications with an energy-modulation management algorithm using MDP. In energy harvesting cooperative communications, Ku et al. [4] applied MDP to minimize the expected symbol error rate in one-way relay energy harvesting network with the decode-and-forward (DF) protocol, while Li et al. [5] used MDP to minimize the long-term average outage probability in two-way relay energy harvesting networks with both DF and amplify-and-forward (AF) protocols. In [7], the authors presented a partially observable MDP to manage the energy in energy harvesting sensors. While MDP is an effective tool for designing an effective online energy management algorithm, it faces the curse of dimensionality when the number of states is large.

To overcome the shortcomings of MDP, many researchers tried to use Lyapunov optimization to optimize the long-term objectives. Unlike the MDP method, Lyapunov optimization works with the continuous state and action, i.e., discretization is not necessary. In [8], [9], [10], Qiu et al. applied Lyapunov optimization theory to optimize different energy harvesting wireless networks. The results showed that Lyapunov optimization can achieve better performance compared with MDP [3], especially at the high signal-to-noise ratio (SNR) regimes. In [11], Amirnavaei et al. maximized the long-term average throughput in energy harvesting wireless networks with Lyapunov optimization. Their algorithm showed better performance than the results in [2]. In [12], Cui et al. utilized Lyapunov optimization to study the delay-aware resource control problem, where the system throughput, the sum delay and the power consumption were jointly optimized.

With the development of the reinforcement learning, more and more works have focused on solving the energy harvesting problems with the aid of reinforcement learning. By utilizing the Q-learning, Blasco et al. [13] proposed algorithms to maximize long-term expected throughput of the point-to-point energy harvesting wireless communications. Based on SARSA, Ortiz et al. proposed a power allocation policy to maximize the throughput at the receiver of a two-hop energy harvesting communications in [14]. However, the existing works that applied the model-free reinforcement learning framework all have to discretize the continuous variables. Therefore, researchers began to use the actor-critic algorithm in reinforcement learning to handle with the continuous data. Aoudia et al. in [15] used actor-critic algorithm to manage the energy to maximize the quality of service while avoiding power failures for energy harvesting wireless sensor networks.

In this paper, we investigate continuous energy management to maximize the net bit rate in energy harvesting wireless communications using DDPG [16]. Instead of optimizing the long-term throughput based on Shannon capacity, we try to deploy DDPG for maximizing the average net bit rate which is a non-convex problem. To resolve the optimization problem with DDPG, we propose to re-write the problem into a state, action and reward form. Moreover, in order to achieve better results, we propose a state normalization to preprocess the input. We also theoretically derive the time and space complexity of the training and the validation processes. Simulations are conducted to evaluate the performance of the proposed policy. The results show that, compared with the state-of-the-art algorithms, our trained policy has better performance especially at the low SNR region.

The rest of the paper is organized as follows. In section II, we briefly review DDPG. Then, the system models of energy harvesting wireless networks are introduced in section III. The energy management policy using DDPG is proposed in detail in section IV. Section V shows the simulation results and section VI concludes this paper.

II. Brief Review of DDPG

DDPG has two main networks: the critic net and the actor net. Both the critic net and the actor net contains two sub-nets: the online net and the target net, whose architectures are the same. These four neural networks are composed of various layers, and all layers contain their corresponding parameters. All parameters in a specific network are denoted as θ. The critic net is trained to simulate the real Q-table using neural networks without the curse of dimensionality. The actor net is trained for generating a deterministic policy instead of the policy gradient which chooses a random action from a determined distribution.

In policy gradient, the agent's behavior a is determined by π, which maps states to a probability distribution over the actions. Given the instantaneous state s_t and the action a_t, if the action's policy is deterministic, denoted as μ, we can avoid the inner expectation and write the Q-table as

$$Q^{\mu}(s_t,a_t)=\mathbb{E}_{r_t,s_{t+1}\sim\Psi}\Big[r(s_t,a_t)+\gamma[Q^{\mu}(s_{t+1},\mu(s_{t+1}))]\Big]. \quad (1)$$

where $r(s_t,a_t)$ represents the reward of the state s_t and the action a_t, γ stands for the discount factor in Bellman equation and Ψ is the corresponding expectation distribution for s_{t+1} and r_t.

When the deterministic policy μ is generated from a randomly initialized stochastic policy ψ, with the approximative Q-table parameterized by θ^Q, the loss of the critic net is defined to measure the distance between the two side of the Bellman equation, which can be expressed as

$$L(\theta^Q)=\mathbb{E}_{s_t\sim\rho^{\psi},a_t\sim\psi,r_t\sim\Psi}\Big[(Q(s_t,a_t|\theta^Q)-y_t)^2\Big], \quad (2)$$

where ρ^{ψ} represents the distribution of the state s_t under the current deterministic policy ψ, θ^Q can be considered as the variables in deep Q network and y_t is defined as follows

$$y_t=r(s_t,a_t)+\gamma Q(s_{t+1},\mu(s_{t+1})|\theta^Q). \quad (3)$$

The actor net updates the policy with the aid of the critic net, where the policy's updating gradient can be written as follows

$$\nabla_{\theta^{\mu}}J\approx\mathbb{E}_{s_t\sim\rho^{\psi}}\Big[\nabla_a Q(s,a|\theta^Q)|_{s=s_t,a=\mu(s_t)}\nabla_{\theta^{\mu}}\mu(s|\theta^{\mu})|_{s=s_t}\Big] \quad (4)$$

where θ^{μ} can be considered as the variables of the online actor net.

The procedure of an entire training process can be described as follows. Firstly, with the action $\mu(s_t)$ given by the actor net after the previous training, DDPG adds some noise n_t and generates the action $a_t=\mu(s_t)+n_t$. Then with the action a_t working on the environment, DDPG can get a reward r_t and a next state s_{t+1}. DDPG will store the set (s_t,a_t,r_t,s_{t+1}) in the experience replay buffer. After that, DDPG randomly chooses N sets in the buffer to make up a mini-batch and inputs it to both the actor net and the critic net. With the mini-batch, the target net of the actor net outputs the action $\mu'(s_{i+1})$ with regard to $\theta^{\mu'}$ to the critic net. With the mini-batch and $\mu'(s_{i+1})$, the target net of the critic net can calculate y_i based on (3) and input it to the online net.

With a given optimizer, e.g., Adamoptimizer, the critic net will update its own online net. Afterwards, the actor online net gives the mini-batch action $a=\mu(s_i)$ to the critic online net to achieve the action a's gradient $\nabla_a Q(s,a|\theta^Q)|_{s=s_i,a=\mu(s_i)}$. With its own optimizer, the parameter θ^{μ}'s gradient $\nabla_{\theta^{\mu}}\mu(s|\theta^{\mu})|_{s=s_i}$ can be derived. With the above two gradients, the actor net can update the actor online net with the following approximation

$$\nabla_{\theta^{\mu}}J\approx\frac{1}{N}\sum_i\Big[\nabla_a Q(s,a|\theta^Q)|_{s=s_i,a=\mu(s_i)}\nabla_{\theta_{\mu}}\mu(s|\theta^{\mu})|_{s=s_i}\Big]. \quad (5)$$

Finally, DDPG softly updates the target nets in both critic net and actor net with a small constant τ, i.e.,

$$\begin{aligned}\theta^{Q'} &\leftarrow \tau\theta^Q+(1-\tau)\theta^{Q'},\\ \theta^{\mu'} &\leftarrow \tau\theta^{\mu}+(1-\tau)\theta^{\mu'}.\end{aligned} \quad (6)$$

III. System Models of Energy Harvesting Wireless Networks

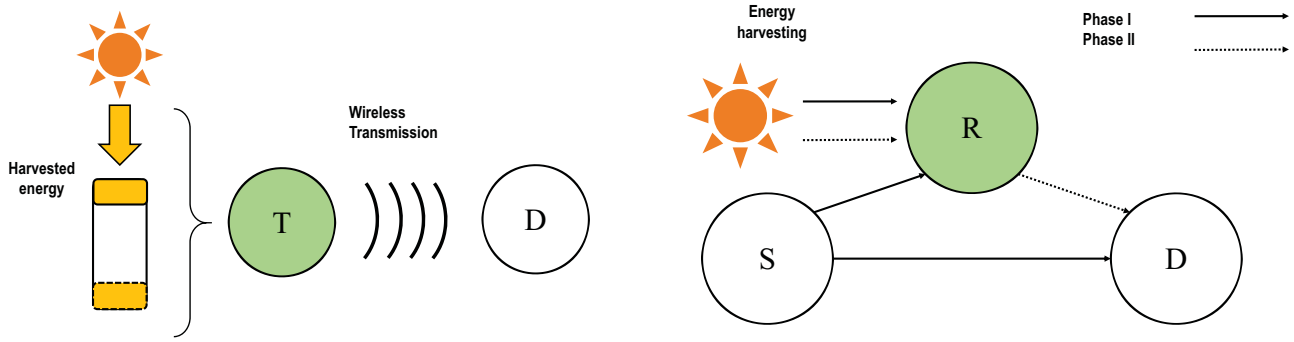
As shown in Fig. 1, in this paper, we consider the energy management for two energy harvesting networks: the point-to-point network [3] and one-way relay network [4]. In the point-to-point network, the transmitter with energy harvesting capability sends the packets to the destination with the energy in the battery while the energy harvester keeps collecting the renewable generations and stores them in the battery. In particular, the energy harvested at present is only available in the subsequent periods, i.e., the harvest-store-use (HSU) model is used in this paper [17]. Thus, the energy in the battery can be written as follows

$$b_{t+1}=\min\{b_t-\omega_t+E_{H,t},b_{max}\}, \quad (7)$$

where t is the time index, $E_{H,t}$ is the collected energy, $\omega_t\in[0,b_t]$ means the consumed energy, and b_{max} represents the battery capacity.

In the one-way relay network, there exist two phases in an entire transmission period. In the first phase, the source broadcasts the packets to the relay and the destination, and the relay decodes the packets. In the second phase, if the decoding in the first phase succeeds, the relay transmits the re-encoded packets to the destination. Different from the point-to-point network, the energy harvester collects the energy in both two phases but only consumes the energy in the second phase. In both networks, the channels are assumed to be i.i.d. with Rayleigh fading.

In this paper, we use the net bit rate, which represents the expected good bits per packet transmission [3], as the system objective. The net bit rate can be influenced by many factors, including the number of bits per packet, the packet



(a) (b)

Fig. 1. Energy harvesting wireless networks: (a) point-to-point network; (b) one-way relay network.

error rate and the rate of sending packages. According to [3], the instantaneous net bit rate can be written as follows

$$R(\zeta_t,\omega_t,b_t) = \begin{cases} \frac{\chi_m L_S}{T_p}(1-P_e)^{\chi_m L_S} & ,\omega \neq 0, \\ 0 & ,\omega = 0, \end{cases}$$
$$s.t. \quad 0 \leq \omega_t \leq b_t \tag{8}$$

where χ_m, L_S, T_p, P_e and ζ_t represent the bit number per symbol, the symbol number per packet, the packet transmission duration, the bit error rate, and the instantaneous channel power, respectively.

Equation (8) means the number of correct bits received per unit time. It is mainly determined by the bit error rate P_e and the number of bits transmitted per unit time, i.e., $\frac{\chi_m L_S}{T_p}$. With the bit error rate, the probability of one packet transmission without any bit error is $(1-P_e)^{\chi_m L_S}$. Then, we can calculate the net bit rate by multiplying the decoding success probability with the total number of bits per unit time. The condition with $\omega = 0$ is written separately because when a transmitter turned off, the transmission will stop.

The bit error rate P_e is determined by the SNR, which is different for different networks. In the point-to-point network, we choose an approximation of the bit error rate as in [3], which can be expressed as

$$P_e(\zeta_t,\omega_t,b_t) = \sum_r \alpha(m,r)\cdot\frac{1}{2}erfc\left(\sqrt{\frac{\beta(m,r)\omega_t\zeta_t}{2N_0T_L}}\right),$$
$$s.t. \quad 0 \leq \omega_t \leq b_t \tag{9}$$

where $erfc(\cdot)$ represents complementary error function, N_0 is the noise power, $\frac{\zeta_t}{N_0}$ stands for the instantaneous channel-to-noise ratio, $\alpha(m,r)$ and $\beta(m,r)$ are two parameters related to the modulation m, which are shown in Table I [3], and r represents the specific constants determined by modulation m. The instantaneous SNR of the point-to-point network is written as $\frac{\omega_t\zeta_t}{N_0T_L}$ in (9), which significantly affects the bit error rate P_e.

Similarly, based on the SNR in the one-way relay network with the decode-and-forward (DF) protocol, the corresponding

TABLE I
MODULATION RELATED PARAMETERS

Modulation schemes	Parameters$(\alpha(m,r),\beta(m,r))$
QPSK	$(\alpha(m,0),\beta(m,0)) = (1,1)$
8PSK	$(\alpha(m,0),\beta(m,0)) = (\frac{2}{3}, 2sin^2(\frac{\pi}{8}))$ $(\alpha(m,1),\beta(m,1)) = (\frac{2}{3}, 2sin^2(\frac{3\pi}{8}))$
16QAM	$(\alpha(m,0),\beta(m,0)) = (\frac{3}{4}, \frac{1}{5})$ $(\alpha(m,1),\beta(m,1)) = (\frac{1}{2}, \frac{9}{5})$

bit error rate can be written as

$$P_e(\zeta_t,\omega_t,b_t) = \begin{cases} \sum_r \alpha(m,r)\frac{1}{2}erfc\left(\sqrt{\frac{\beta(m,r)[(\omega_t/T_L)\zeta_{rd,t}+\Psi_s\zeta_{sd,t}]}{2N_0}}\right), \\ \qquad\qquad d=1; \\ \sum_r \alpha(m,r)\frac{1}{2}erfc\left(\sqrt{\frac{\beta(m,r)\Psi_s\zeta_{sd,t}}{2N_0}}\right), \\ \qquad\qquad d=0, \end{cases}$$
$$s.t. \quad 0 \leq \omega_t \leq b_t \tag{10}$$

where $d = 1$ corresponds to the case where the decoding in the relay is successful, Ψ_s represents the transmission power of the source node, and ζ_{sd} and ζ_{rd} are the channel power of the source-to-destination (SD) link and the relay-to-destination (RD) link, respectively. The decoding condition d is determined by the SNR of the source-to-relay link, and d is equal to 1 only if $\frac{\Psi\zeta_{sr}}{N_0} \geq \mathfrak{T}$, where $\mathfrak{T}$ is a constant standing for the decoding capacity threshold. The SNR of the one-way relay network is determined by both of the relay route and the source route, which is written as $\frac{[(\omega/T_L)\zeta_{rd}+\Psi_s\zeta_{sd}]}{N_0}$ in (10).

IV. ENERGY MANAGEMENT WITH DDPG

A. Problem Formulation

The system objective in this paper is to maximize the long-term average net bit rate in the point-to-point network and the one-way relay network, i.e.,

$$\max_{\omega_t} \quad \lim_{T\to+\infty}\frac{1}{T}\sum_{t=0}^{T-1} R(\zeta_t,\omega_t,b_t),$$
$$s.t. \quad 0 \leq \omega_t \leq b_t, \tag{11}$$

where the bit error rate in the long-term objective function is defined in (9) and (10) for the point-to-point network and the one-way relay network, respectively.

In DDPG, the neural networks are trained from the set (s_t, a_t, r_t, s_{t+1}). Therefore, we have to define the state and action sets of our problem for DDPG. Moreover, with the constraint of the energy for transmission, we have to define a different action for DDPG instead of directly deploying the ω_t. In the following, we describe in detail the state set, the action set, and the reward.

1) The set of states s_t: In the point-to-point communications, the states needed to be considered in a management period include b_t, ζ_t and $E_{H,t}$, where b_t determines the maximum energy the transmitter node can consume, ζ_t influences the bit error rate directly, and $E_{H,t}$ affects b_{t+1}. Therefore, for the point-to-point network, the sets are $s_t = (b_t, \zeta_t, E_{H,t})$.

On the other hand, there exist three independent fading channels in the one-way relay network. Nevertheless, the states contain only the channel power of the SD link $\zeta_{sd,t}$ and the channel power of the RD link $\zeta_{rd,t}$, i.e., $s_t = (b_t, \zeta_{sd,t}, \zeta_{rd,t}, E_{H,t})$. The reason of ignoring the channel power of the SR link $\zeta_{sr,t}$ is that $\zeta_{sr,t}$ only influences the decoding condition. In the training process, if the decoding fails in the first phase, the training will be directly skipped. In the validation process, if the decoding fails in the first phase, the relay will not manage the energy for transmission. The SR link only influences whether the relay is on or off, but has no impact on the energy management.

2) The set of actions a_t: In this paper, we set the continuous possible action $a_t \in [0, 1]$ since b_t varies with different t and the actor function in DDPG has to be bounded by some constants. With such an action set, the final energy consumed for transmission is $a_t \times b_t$, which can guarantee that the consumed energy will not exceed the remaining energy in the battery. Thus, the action in this paper can be written as

$$a_t = \mu((b_t, \zeta_t, E_{H,t})|\theta^\mu), \tag{12}$$

or

$$a_t = \mu((b_t, \zeta_{sd,t}, \zeta_{rd,t}, E_{H,t})|\theta^\mu), \tag{13}$$

with the range of $[0, 1]$.

3) Reward: With the Q-table in Q-learning, the long-term average net bit rate can be written using the Bellman equation as follows

$$Q^\mu(s_t, a_t) = \mathbb{E}_{r_t, s_{t+1} \sim E}\Big[R(s_t, a_t) + \gamma[Q^\mu(s_{t+1}, \mu(s_{t+1}))]\Big], \tag{14}$$

where the reward $R(s, a)$ is chosen as the corresponding net bit rate in (8). Now the problem is to find an action to maximize the Q value in (14) with DDPG.

B. State Normalization

In deep learning, the distribution of each layer's inputs keeps changing, which slows down the training by requiring lower learning rates and careful parameter initialization. Ioffe et al. proposed the batch normalization in [18] to allow the training to use much higher learning rates and relaxed initialization. Similar to the batch normalization, we propose a state normalization to preprocess the training sample states for a much easier and faster training.

Algorithm 1 State Normalization

Require:
All instantaneous variables needed to be normalized: b_t ,ζ_t ($\zeta_{sd,t}$, $\zeta_{rd,t}$) and $E_{H,t}$;
Scale factors: λ_1, λ_2;
Means and standard deviations of the variables: η_{E_H}, η_ζ, σ_{E_H} and σ_ζ;

Ensure:
Normalized variables $\hat{b}_t$, $\hat{\zeta}_t$ ($\hat{\zeta}_{sd,t}$, $\hat{\zeta}_{rd,t}$) and $\hat{E}_{H,t}$

1: $\hat{b}_t = \frac{b_t}{\lambda_1}$
$\hat{\zeta}_t = \frac{\zeta_t - \eta_\zeta}{\sigma_\zeta}$
$\hat{E}_{H,t} = \frac{E_{H,t} - \eta_{E_H}}{\lambda_2 \sigma_{E_H}}$
2: **return** $\hat{b}_t$, $\hat{\zeta}_t$ ($\hat{\zeta}_{sd,t}$, $\hat{\zeta}_{rd,t}$) and $\hat{E}_{H,t}$

The three variables b_t, ζ_t, and $E_{H,t}$ in the state set may lie in different ranges, which may cause problem in the training process. To prevent such a problem, we normalize the variables b_t, ζ_t, and $E_{H,t}$ separately. The state normalization is shown in the Algorithm 1, where we use two extra scale factors in the normalization. The reason can be explained as follows. According to (7), the energy in the battery is in the form of a queue and it will be influenced by the action. In such a case, it is difficult to use a constant value to approximate the average energy in the battery. Thus, we scale down the remaining energy in the battery of all epoches. We also scale the value of the normalized $\hat{E}_{H,t}$ to adjust the balance between the influence of $E_{H,t}$ and b_t.

C. Complexity Analysis

From the above discussions, we can see that the training algorithm includes the normalization, the replay buffer and four neural networks, while the validation algorithm is only made up of the normalization and the online actor net. In the following, we will derive the time complexity (computations) with regard to FLOPS (floating point operations per second) and space complexity (memory) of the training and validating algorithms, respectively.

1) Training: The state normalization is conducted at every epoch of the training because without action we cannot know the value of b_t. Thus, the time complexity of state normalization is $\mathcal{N}(s)$, where $\mathcal{N}(s)$ is the number of the variables in the state set. The space complexity is related to the number of the variables in the state, i.e., $2\mathcal{N}(s)$ because the algorithm has to record the means and standard deviations to avoid repeated calculation. The experience replay buffer in DDPG occupies some space to store the state sets, hence the space complexity is N.

Since the input state of the energy harvesting communications is different from that of the image/video, there is no convolution layer in both the actor net and the critic net. For dot products of a P vector and an $P \times Q$ matrix, the FLOPS

computations is $(2P-1)Q$ because for every column in matrix we need to multiply P times and add $P-1$ times.

We also have to derive the computations of activation layers. When calculating FLOPS, we usually count addition, subtraction, multiplication, division, exponentiation, square root, etc as a single FLOP. The computations is Q with Q inputs for Relu layers, $4 \times Q$ for sigmoid layers, and $6 \times Q$ for tanh layers.

Assuming that the actor net contains J fully connected layers and the critic net contains K fully connected layers, considering the bias adding in fully connected layers the time complexity can be calculated as

$$\begin{aligned} & v_{activation} u_i + 2 \times \sum_{j=0}^{J-1} u_{actor,j} u_{actor,j+1} \\ & +2 \times \sum_{k=0}^{K-1} u_{critic,k} u_{critic,k+1} \\ =& O\left(\sum_{j=0}^{J-1} u_{actor,j} u_{actor,j+1} + \sum_{k=0}^{K-1} u_{critic,k} u_{critic,k+1} \right) \end{aligned} \tag{15}$$

where u_i means the unit number in the i^{th} layer, u_0 equals the input size and $v_{activation}$ means the corresponding parameters determined by the type of the activation layer.

For a fully connected layer, there is a $P \times Q$ matrix and a Q bias vector. Hence, the memory of one fully connected layer is $(P+1)Q$. Because the activation do not need saved weights, the space complexity of the neural networks is formulated as:

$$\begin{aligned} & \sum_{j=0}^{J-1} (u_{actor,j}+1) u_{actor,j+1} + \sum_{k=0}^{K-1} (u_{critic,k}+1) u_{critic,k+1} \\ =& O\left(\sum_{j=0}^{J-1} u_{actor,j} u_{actor,j+1} + \sum_{k=0}^{K-1} u_{critic,k} u_{critic,k+1} \right) \end{aligned} \tag{16}$$

Therefore, the overall time complexity of our training algorithm is

$$\begin{aligned} & 2 \times \sum_{j=0}^{J-1} u_{actor,j} u_{actor,j+1} + 2 \times \sum_{k=0}^{K-1} u_{critic,k} u_{critic,k+1} \\ & +v_{activation} u_i + \mathcal{N}(s) \\ =& O\left(\sum_{j=0}^{J-1} u_{actor,j} u_{actor,j+1} + \sum_{k=0}^{K-1} u_{critic,k} u_{critic,k+1} \right) \\ & +O(\mathcal{N}(s)), \end{aligned} \tag{17}$$

and the overall space complexity of our training algorithm is

$$\begin{aligned} & \sum_{j=0}^{J-1} (u_{actor,j}+1) u_{actor,j+1} + \sum_{k=0}^{K-1} (u_{critic,k}+1) u_{critic,k+1} \\ & +2 \times \mathcal{N}(s) + N \\ =& O\left(\sum_{j=0}^{J-1} u_{actor,j} u_{actor,j+1} + \sum_{k=0}^{K-1} u_{critic,k} u_{critic,k+1} \right) \\ & +O(\mathcal{N}(s)) + O(N)). \end{aligned} \tag{18}$$

TABLE II
DDPG ARCHITECTURE (THE COMPLEXITY IS EVALUATED WITH FLOPS, I.E., THE NUMBER OF FLOATING-POINT MULTIPLICATION-ADDS).

Net	Layer	Units	Activation	FLOPS	Params
Actor	Fully connected	60			
	Fully connected	30			
	Fully connected	1	Sigmoid	4.02K	2.10K
Critic	Fully connected	60	RELU		
	Fully connected	60			
	Fully connected	60	Tanh		
	Fully connected	60			
	Fully connected	60	RELU		
	Fully connected	1		29.88K	15.00K

2) Validating: Since the critic net is generated to help the actor net have a faster and easier training, there is no critic net and replay buffer in the validation process. Only the state normalization and the online net in the actor net is needed. Therefore, the time complexity of the validation algorithm is

$$O\left(\sum_{j=0}^{J-1} u_{actor,j} u_{actor,j+1} \right) + O(\mathcal{N}(s)), \tag{19}$$

and the space complexity is the same

$$O\left(\sum_{j=0}^{J-1} u_{actor,j} u_{actor,j+1} \right) + O(\mathcal{N}(s)). \tag{20}$$

V. NUMERICAL SIMULATIONS

A. EH Communications and DDPG Setup

In our simulations, we use the real solar power data collected in every five minutes from 7am to 5pm in June from 2010 to 2012 [19]. The solar panel size is 4cm^2 and the energy conversion efficiency is assumed to be 20%. The Rayleigh fading channel is generated with the aid of Jakes model, which can guarantee that the channel varies in a smooth way. We assume that each packet contains 1000 symbols ($L_S = 1000$) and the packet duration $T_P = 0.01s$. One management period is set as 5×60 seconds, while for the one-way relay energy harvesting communication the length of one phase is 150 seconds. Additionally, for the one-way relay energy harvesting network, the extra settings are shown as follows. The source node's transmission power is set as 40 mW and the decoding capacity threshold $\mathfrak{T}$ is set as 15 dB. Finally, the SNR of the SR channel is limited to 40dB, independent from the other two channels.

The architecture of the actor net and the critic net is shown in Table II. In DDPG, we use three fully connected layers to build the online subnet and the target subnet for the actor net, while double fully connected layers are used to generate the critic net. In the actor net, we only use the sigmoid activation to ensure that our final output action is bounded by 0 and 1. For the critic net, aside from more layers and units, we also add two Relu activation layers and one tanh activation layer. This is because that the net bit rate is a complex non-linear function which includes the complementary error function (erfc). Hence

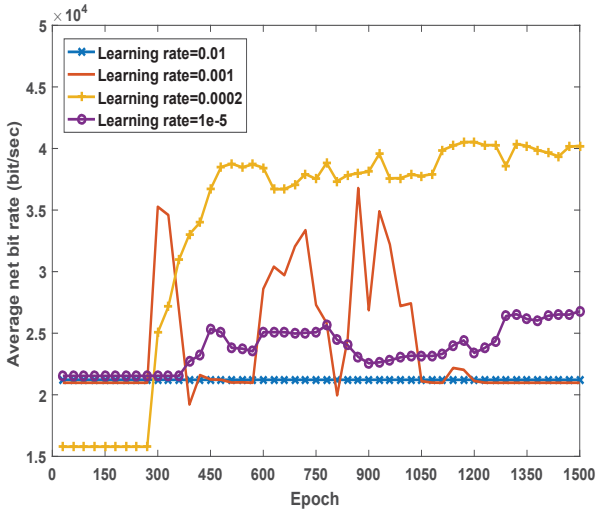


Fig. 2. Convergence of training P2P policy with different learning rate.

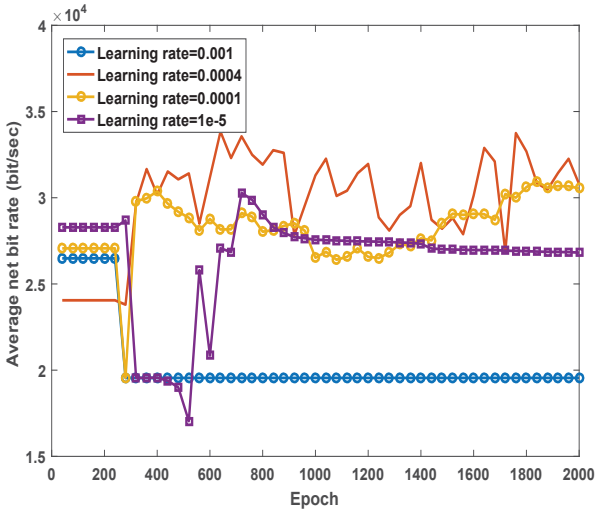


Fig. 3. Convergence of training one-way relay policy with different learning rate.

more layers and activations are helpful to approach the non-linear Q-table. We only use three layers in the actor net mainly for reducing the complexity of the energy management policy, especially in real applications.

The parameter settings of DDPG are shown as below. For the point-to-point network, the training sample number is 120, where we only use the solar data of June 1st for training. The length of the training epochs is 2000, the replay buffer's capacity is 40000 and the size of the mini-batch is 80. The learning rate of the actor net and the critic net are both set to be 2×10^{-4}. The discount factor γ of the Q-table is 0.999 and the target subnet soft update factor τ is set to be 0.01. With the behavior noise, the initialized noise's average value is set as 10 and the noise's decay factor κ is set as 0.9995. Finally, the scale factors for the state normalization λ_1 and λ_2 are set as 100 and 2, respectively.

For the one-way relay network, the length of the training sample is 240, where we apply the solar data of June 1st and June 2nd. We use more training samples since there exists a probability of the failed decoding. If the decoding fails, the policy in energy harvesting relay cannot be trained. The length of the training epochs is 1500, the learning rates of the actor and critic net are both 4×10^{-4}, and the discount factor γ in Q-table is set as 0.9. The other parameters are the same as those of DDPG in the point-to-point network.

B. Convergence with different learning rates

The influence of the learning rate of DDPG can be inferred from Fig.2 and Fig.3, where the modulations are both QPSK, the channel-to-noise ratio in the peer-to-peer network is -10dB and the SNR in the one-way relay network is 4dB. The learning rates of the critic net and the actor net are assumed the same.

From Fig.2, we can see that when the learning rate is 0.01, there is no advantage in the training since the policy turns to a greedy algorithm in this case. With a learning rate of 0.001, the result fluctuates with the development of the epochs. If the learning rate goes smaller, e.g., 0.0002, the result generally maintains improving with slight fluctuations. When the learning rate becomes 10^{-5}, we can see that the result grows slowly but still with slight fluctuations.

In Fig.3, we observe similar phenomenon as that in Fig.2. If the learning rate is too large, the average net bit rate will quickly saturate at a bad value. In a reasonable range, with a higher learning rate, the result can grow faster but with larger fluctuations. On the other hand, if the learning rate is smaller, fluctuations will be reduced at the sacrifice of the speed of the performance growth. Therefore, the learning rate should be selected properly, neither too large nor too small. Compared with Fig.2, we find that it is harder for DDPG to converge at a good result in the one-way relay network. This is because we artificially make the relay off if the decoding d in (11) is 0. This may make the state input in DDPG not continuous in time and cause more difficulty in training a good action.

C. Performance Comparison

We first compare the average net bit rate performance among the proposed DDPG method, the Lyapunov optimization [8], and the Greedy algorithm, under different modulations. The results are shown in Fig.4, Fig.5 and Fig.6. For the one-way relay network, the SNR refers to the signal-to-noise ratio of the SD link, i.e., $\frac{\Psi_s \zeta_{sd}}{N_0}$ and the channel noise power of the SD link and RD link is the same. For the point-to-point network, since there is no relay, the average channel-to-noise ratio is set to be SNR-10.

Fig. 4 shows the average net bit rate with the QPSK modulation under different methods. From the figure, we can see that the proposed method performs the best in both the point-to-point network and the one-way relay network, while the Lyapunov optimization performs better than the greedy algorithm. At the low SNR region, the performance gap among

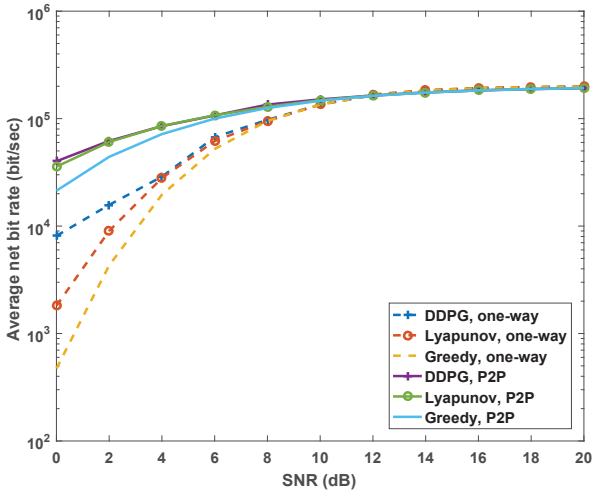


Fig. 4. Average net bit rate under the QPSK modulation.

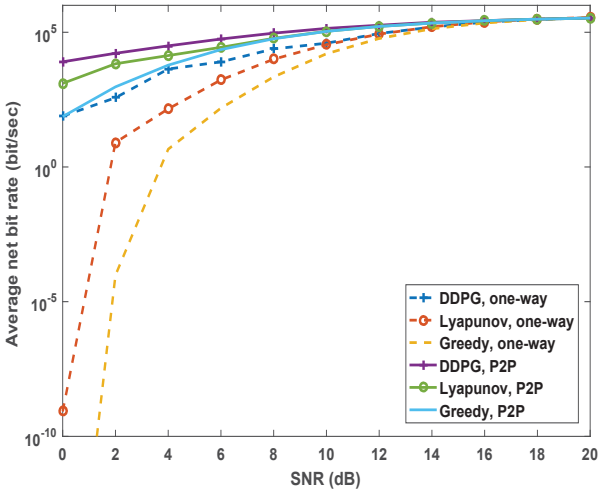


Fig. 6. Average net bit rate under the 16QAM modulation.

different methods is significant. When the SNR is 0dB, the achieved average net bit rate with the proposed method is around 8×10^3, which raises approximately 4 times compared with the Lyapunov optimization, and 10 times compared with the greedy strategy. For the point-to-point network, the gap is much smaller. This is mainly because there is no link with constant power supply in the point-to-point network, due to which the received SNR will be affected significantly. With the growth of SNR, the average net bit rate of different methods all saturates at the value of 2×10^5, which corresponds to the case with zero BER in (8).

Fig. 5 and Fig. 6 show the performance comparison of different policies with the 8PSK and 16QAM modulations. Similar to the results in Fig. 4, the performance of the proposed method is the best, and the Lyapunov optimization performs better than the greedy algorithm, especially at the low SNR region. With the increase of the SNR, the net bit rate of all schemes saturates at the value of $\frac{\chi_m L_S}{T_P}$, i.e., 3×10^5 and 4×10^5. However, compared with the results in Fig.4, we can see that the gaps among different strategies in Fig. 5 and Fig. 6 are much larger. The reason is that with the same energy consumed for transmission, the probability of successfully transmitting an entire packet under 16QAM and 8PSK will be much smaller than that under QPSK. With the proposed method, the energy management is much more careful, which thus leads to much better performance.

Fig.7 illustrates the performance comparison of the proposed method and the MDP algorithm in the point-to-point network. We do not evaluate the MDP in the one-way relay network since the equation (14) in [3] cannot be derived with

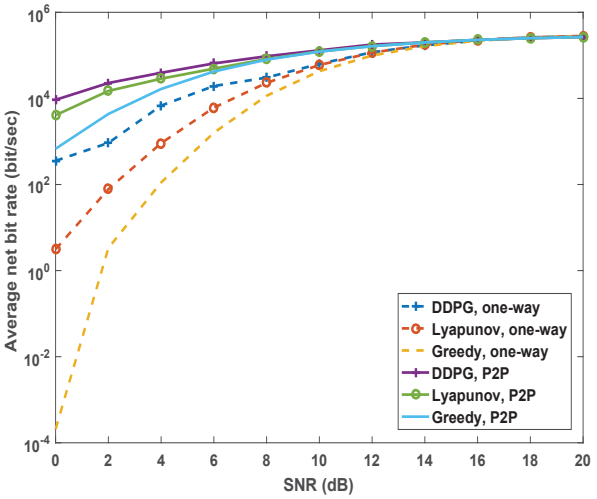


Fig. 5. Average net bit rate under the 8PSK modulation.

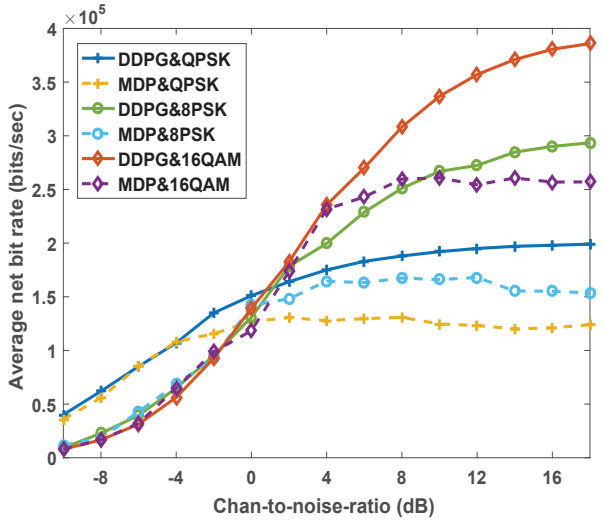


Fig. 7. Comparison between our method and MDP[3].

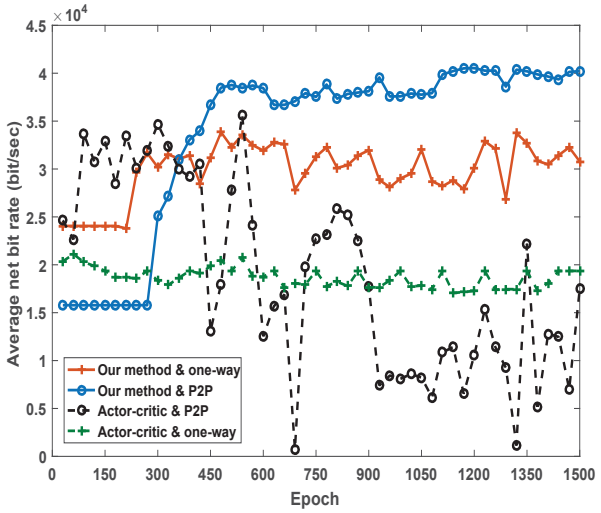


Fig. 8. Comparison between our method and Actor-critic RL.

two channel links. The interval number of the solar, battery, channel and action in MDP are set as 4,8,4,8, respectively. From the figure, we can see that at the low SNR region, the proposed method has a similar performance with the MDP algorithm, regardless the modulation. However, with the growth of the SNR, the performance of MDP finally converges at a much smaller value than that of the proposed method. This phenomenon may be due to the following reason. In MDP, if the average harvesting rate of the solar power is smaller than the basic action level, the transmitter has to wait several epoches for a basic energy quantum for transmission. As a result, in some management periods the transmitter are not capable of transmitting any bits because the energy amount in battery cannot reach the basic action power [3].

Finally, we compare the proposed method with the basic actor-critic method [20]. The structures of the actor net and the critic net in the basic actor-critic method are the same as the proposed method. Moreover, the learning rates and other setup parameters are the same as those in our methods, and the normalization has also been applied. The main difference is that in the basic actor-critic method there is no replay buffer and the action is estimated by a distribution instead of a deterministic function. From the result in Fig. 8, we can see that the basic actor-critic method is not able to learn an efficient state-action strategy in the stochastic energy harvesting problems. The energy management strategy cannot converge at a good value in the point-to-point network, and in one-way relay network the strategy cannot make any improvement with the actor-critic learning. Therefore, the basic actor-critic method cannot learn the complex state-action pattern in the stochastic energy harvesting wireless communications.

VI. Conclusion

In this paper, we studied the energy management problem in energy harvesting wireless networks. Our objective was to maximize the long-term average net bit rate in the point-to-point network and one-way relay network. We employed DDPG to train an optimal energy management policy to optimize the net bit rate. We proposed a state normalization algorithm to make the training much faster and easier. We also theoretically analyzed the time and space complexity of the proposed training and validating algorithms. Compared with the state-of-the-art algorithms, the proposed algorithm achieves better performance in terms of long-term average net bit rate.

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, March 2015.

[2] O. Ozel, K. Tutuncuoglu, J. Yang, S. Ulukus and A. Yener, "Transmission with energy harvesting nodes in fading wireless channels: optimal policies," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 8, pp. 1732-1743, September 2011.

[3] M.L. Ku, Y. Chen and K.J.R. Liu, "Data-driven stochastic models and policies for energy harvesting sensor communications," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 8, pp. 1505-1520, Augsut 2015.

[4] M. L. Ku, W. Li, Y. Chen and K. J. R. Liu, "On energy harvesting gain and diversity analysis in cooperative communications," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 12, pp. 2641-2657, December 2015.

[5] W. Li, M. L. Ku, Y. Chen and K. J. R. Liu, "On outage probability for two-way relay networks with stochastic energy harvesting," *IEEE Trans. Commun.*, vol. 64, no. 5, pp. 1901-1915, May 2016.

[6] W. Li, M.L. Ku, Yan Chen, K.J.R. Liu, and S.H. Zhu, "Performance analysis for two-way network-coded dual-relay networks with stochastic energy harvesting," *IEEE Trans. Wireless Commun.*, vol. 16, no. 9, pp. 5747-5761, September 2017.

[7] A. Yadav, M. Goonewardena, W. Ajib, O. A. Dobre and H. Elbiaze, "Energy management for energy harvesting wireless sensors with adaptive retransmission," *IEEE Trans. Commun.*, vol. 65, no. 12, pp.5487-5498, December 2017.

[8] C. Qiu, Y. Hu and Y. Chen, "Lyapunov Optimization for Energy Harvesting Wireless Sensor Communications," *IEEE Internet Things J.*, Vol. 5, No. 3, 1947-1956, June 2018.

[9] C. Qiu, Y. Hu and Y. Chen, "Lyapunov Optimized Cooperative Communications with Stochastic Energy Harvesting Relay," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1323-1333, April 2018.

[10] Y. Hu, C. Qiu and Y. Chen, "Lyapunov Optimized Two-Way Relay Networks with Stochastic Energy Harvesting," *IEEE Trans. Wireless Communications*, vol. 17, no. 9, pp. 6280-6292, July 2018.

[11] F. Amirnavaei and M. Dong, "Online power control optimization for wireless transmission with energy harvesting and storage," *IEEE Trans. Wireless Commun.*, vol. 15, no. 7, pp. 4888-4901, July 2016.

[12] Y. Cui, V.K.N. Lau, R. Wang, H. Huang and S.Q. Zhang, "A survey on delay-aware resource control for wireless systems - large deviation theory, stochastic Lyapunov drift, and distributed stochastic learning," *IEEE Trans. Inf. Theory*, vol. 58, no. 3, pp. 1677-1701, March 2012.

[13] P. Blasco, D. Gündüz and M. Dohler, "A learning theoretic approach to energy harvesting communication system optimization," *IEEE Trans. Wireless Commun.*, vol. 12, no. 4, pp. 1872-1972, November 2013.

[14] A. Ortiz, H. Al-Shatri, X. Li, T. Weber and A. Klein , "Reinforcement learning for energy harvesting decode-and-forward two-hop communications," *IEEE Trans. Green Commun. Netw.*, Vol. 1, No. 3, 309-319, September 2017.

[15] F. A. Aoudia, M. Gautier and O. Berder , "RLMan: an energy manager based on reinforcement learning for energy harvesting wireless sensor networks," *IEEE Trans. Green Commun. Netw.*, Vol. 2, No. 2, 408-417, June 2018.

[16] T. P. Lillicrap, J. J. Hunt, A. Pritzel et al., "Continuous control with deep reinforcement learning," International Conference on Learning Representations (ICLR) 2016. arXiv preprint arXiv:1509.02971v5 [cs.LG] 29 February 2016

[17] S. Sudevalayam and P. Kulkarniand, "Energy harvesting sensor nodes: survey and implications," *IEEE Commun. Surv. Tuts.*, vol. 13, no. 3, pp. 443-461, Third Quart. 2011.

[18] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” arXiv preprint arXiv:1502.03167, 2015.
[19] NREL.(2010, September, 2011, September, 2012, September). Cooperative networks for renewable resource measurements (CONFRRM) solar energy resource data, Elizabeth City, North Carolina, USA. [Online]. Available: http://www.nrel.gov/rredc
[20] R.S. Suttion and A.G. Barto, “Reinforcement learning: an introduction”, MIT Press, Cambridge, 2014.