BEAR: Reinforcement Learning for Throughput Aware Borrowing in Energy Harvesting Systems

Anubhav Sachan[1], Deepak Mishra[2], and Ganesh Prasad[1]
[1]Department of Electronics and Communication Engineering, National Institute of Technology Silchar, India
[2]School of Electrical Engineering and Telecommunications, University of New South Wales, Australia
Emails: anubhav4sachan@gmail.com, d.mishra@unsw.edu.au, gpkeshri@ece.nits.ac.in

***Abstract*—Energy Borrowing (EB) aided Energy harvesting (EH) systems provide a greener alternative to self-sustaining electronic devices in a complex, unprecedented environment by borrowing energy from a supplementary source to regulate the data transmission flow. We propose a reinforcement learning-based algorithm for energy scheduling policy which jointly optimizes the EB and utilizes harvested energy for efficient data transfer at every time instant. As the exact pattern of harvested energy and channel conditions at any time slot is unknown, the proposed algorithm, BEAR (Borrowing Energy with Adaptive Rewards), based on actor-critic architecture, learns the optimal power allocation policy for the transmission node. Our designed reward function accommodates the concept of adaptive penalty to punish the transmission node for selecting unfavourable actions. Our simulations show that the BEAR algorithm providing efficient energy management with a focus on throughput maximization yields a 35.45% enhancement in sum throughput over a typical non-borrowing system. Lastly, nontrivial design insights are outlined via numerical results to quantify the practical efficacy of BEAR for EH systems.**

I. Introduction

Energy harvesting (EH) allows the nodes in wireless communication system to utilize ambient energy for their electrical energy needs [1]. However, as the energy arrival is sporadic, energy borrowing (EB) allows a the nodes to borrow energy from a secondary source, and utilize it for efficient data transfer[2]. Additionally, a dynamic environment, comprising random (harvested) energy arrival and channel conditions, elevates the complexity in optimally managing the harvested energy. Since the patterns of energy arrival (from harvested energy) and channel conditions cannot be determined statistically, the research community is resorting to data-driven approaches, such as machine learning, to build intelligent nodes. The integration of concepts of utilizing harvested energy and temporary borrowing will play significant role in sustainability of low lower wireless devices in Internet of things (IoT).

A. Related Works

Recently, reinforcement learning (RL) based algorithms are proving quite valuable for deploying intelligent nodes in unknown dynamic environments. These nodes smartly manage the available energy and, with experience, improve their performance with time [3], [4]. RL methods are efficiently used in a point-to-point EH communication system to learn (1) the transmission power allocation policy to maximize the received data using SARSA algorithm [5], and (2) a transmission policy to maximize the expected transmitted data with Q-Learning methods [6]. The Actor-Critic algorithm [7] combines the best of value-based methods and policy gradient methods. Generally, the actor selects an action according to a parameterized policy. The critic indicates the policy's quality by giving a feedback value to the actor. This scalar output from the critic helps optimize parameters for the policy to select an optimal action. The actor-critic algorithm is widely used in user scheduling and resource allocation [3], and energy management in wireless EH nodes [4].

The discussed EH wireless communication systems can transmit data if the energy source attached with the transmitter possesses a required amount of energy for transmission, irrespective of the channel conditions. In a typical scenario, where the channel conditions are suitable for data transmission, but the requisite amount of energy is not available, these communication systems will not transmit data. However, since the channel conditions are favourable, the transmitting node should transmit the data by borrowing the requisite amount of energy (for transmission) from a nearby power source [2] with a caveat that the borrowed energy has to be returned along with a packet of additional energy serving as an interest [8]. Assuming prefect channel and EH states information availability, the borrowing and returning schedules as developed by Reddy et al. based on Water-Filling outperformed the cross entropy based statistical method [2] in complexity.

B. Motivation and Contributions

EB scheduling protocols in [2], [8] can't be deployed in real-life environments where an underlying pattern of harvested energy and channel conditions is random and unknown. In complex realistic scenarios, we need to predict or estimate the channel and EH conditions by building novel machine learning-based framework for borrowing-aided EH system.

The key contributions of this paper are three-fold.

1) An actor-critic based novel RL algorithm called BEAR (Borrowing Energy with Adaptive Rewards), is proposed to maximize the throughput over finite time slots establishing a RL-based benchmark for borrowing-aided EH point-to-point wireless communication system.
2) BEAR algorithm is equipped with an adaptive reward and penalty functions where the underlying transmission power allocation policy is modeled by a parameterized Gaussian Distribution and ensures to return the borrowed

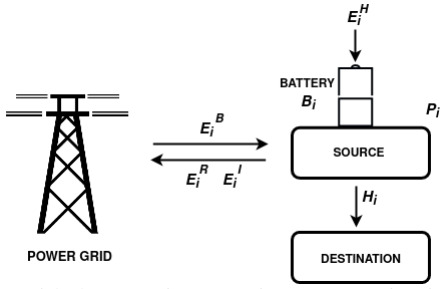

Fig. 1: EB aided EH point-to-point communication system.

energy to maximize the sum throughput over a given time slots while the learning EH and channel conditions.

3) Computer simulations showed that with the introduction of borrowing with adaptive rewards, BEAR algorithm can achieve significantly higher throughput, providing 35.45% gain over an EH system without borrowing.

II. System Model

A. Communication and Energy Harvesting Models

We have considered a point-to-point wireless communication system comprising a power grid, an information source, and a destination node, as depicted in Fig. 1. The information source (S) is capable of harvesting energy from the available ambient sources (solar or wind) and subsequently stores the harvested energy in a battery (B) of finite capacity. The source S has to transmit data to destination D and has the liberty of either utilizing the harvested energy or borrowing the required amount of energy from a nearby power grid (PG) for the transmission, where PG is assumed to have infinite energy.

The communication time is divided equally into time slots of length t_s. At time slot i, the source S has the battery energy level B_i and has harvested energy E_i^H from solar/wind sources with E_i^H following Gaussian distribution with mean μ_{E} and variance σ_d^2. This random variable $\mathcal{E}_{\mathcal{N}}$ is governed by the normal probability density function $f_{\mathcal{E}_{\mathcal{N}}}(E_{i+1}^H|E_i^H)$ defining the probability of the transition from harvested energy level E_i^H to next energy level E_{i+1}^H. The channel channel coefficient H_i at time slot i can be estimated with the aid of pilot signals [4] known to both S and D. The Gaussian distributed transition probability density function $f_{\mathcal{H}_{\mathcal{N}}}(H_{i+1}|H_i)$ defines the probability of transitioning to channel state H_{i+1} from H_i.

The transmission of data from S to D at a time slot i requires power given by P_i and we assume that power consumption other than required for transmission, such as by internal circuitry, are neither satisfied by the battery B nor by the harvested energy at any time slot. The energy borrowed from PG is E_i^B and the energy returned to PG is E_i^R with an interest E_i^I at a time slot i. The system model uses the harvest-store-use strategy to utilize harvested energy [4].

B. Energy Scheduling Protocol

The EB and energy returning (ER) schedules [8] provide a definitive battery energy level at the end of a time slot i as

$$B_i = \min\left\{(B_{i-1} + E_i^H), B_{\max}\right\} + E_i^B - E_i^R - t_s P_i \quad (1)$$

where $B_i \in [0, B_{\max}]$, E_i^B is EB schedule, $B_{\max}$ is maximum battery capacity and E_i^R is the ER schedule.

1) Energy Borrowing Schedule: During the transmission of data at power P_i, if S is short of energy from the available energy pool at the slot i i.e. $(B_{i-1}+E_i^H)$, the requisite amount is borrowed (E_i^B) from PG to satisfy the crisis at S following borrowing schedule

$$E_i^B = \begin{cases} t_s P_i - (B_{i-1} + E_i^H), & \text{if } t_s P_i > (B_{i-1} + E_i^H) \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

2) Energy Returning Schedule: Since the energy during transmission is sometimes borrowed from PG as per (2), it has to be returned to PG in future utilizing the harvested energy. The amount of energy returned at a time slot i is given by the energy-returning schedule as

$$E_i^R = \begin{cases} \zeta E_i^E, & \text{if } E_{i-1}^U > E_i^E \\ \zeta E_{i-1}^U, & \text{otherwise} \end{cases} \quad (3)$$

where E_i^E $(= B_{i-1} + E_i^H + E_i^B - t_s P_i)$ is defined as the excess energy at the end of $i-$th time slot, $\zeta \in (0, 1]$ denotes the efficiency of energy transfer from S to PG and E_{i-1}^U is the unreturned energy in the previous time slot $i-1$.

The unreturned energy E_i^U at a time slot i is given by

$$E_i^U = \begin{cases} E_{i-1}^U + E_i^B, & \text{if } t_s P_i > (B_{i-1} + E_i^H) \\ E_{i-1}^U + E_i^I - E_i^R, & \text{if } t_s P_i \le (B_{i-1} + E_i^H) \\ & \text{and } E_i^R \le (E_{i-1}^U + E_i^I) \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

where E_i^I is the cost incurred in form of an interest (caused due to delay in the return of the borrowed energy) defined by

$$E_i^I = \eta E_{i-1}^U \quad (5)$$

where $\eta \in [0, 1]$ is the rate of interest per time slot. The threshold $E_{\max}^U$ is introduced to limit the excessive borrowing, and it serves as the upper bound for the unreturned energy during the entire transmission process given by $E_i^U \le E_{\max}^U$. The transfer efficiency ζ in (3), along with rate of interest η in (5) are assumed to be constant throughout the process.

III. Reinforcement Learning Framework for EB

A. Problem Definition

We focus on determining the optimal values for decision variable P_i across slots such that the sum throughput for the complete data transmission is maximized. During this transmission, at any particular time slot, if the requisite amount of energy is not available, the source can borrow energy from PG to best utilize channel condition. However, entire "borrowed" energy has to be returned by the end of all time slots along with the levied interest.

B. State-Action Space

We have split the state space $\mathcal{S}$ into source state space comprising the set of values defining the battery energy level and environmental state space comprising the values from the continuous random variables $\mathcal{E}_{\mathcal{N}}$ and $\mathcal{H}_{\mathcal{N}}$. The actual

values for the harvested energy E_i^H and channel gain H_i are sampled from the transition probability density functions $f_{\mathcal{E}_N}(E_{i+1}^H|E_i^H)$ and $f_{\mathcal{H}_\mathcal{N}}(H_{i+1}|H_i)$ respectively. The action space $\mathcal{A}$ contains the possible values of transmission power. The state S_i at a time slot i is described as a tuple of energy harvested, channel coefficient and battery energy level at the same slot i i.e., $S_i = (E_i^H, H_i, B_i)$. The action at a time slot i is defined with power P_i consumed due to data transmission.

C. Objective: Sum Throughput Maximization

Since the values of harvested energy and channel conditions are independent of their past values given the present, the decision process can be formulated as a Markov Decision Process with continuous state and action spaces. The Markov Decision Process is formally depicted as a tuple $(\mathcal{S}, \mathcal{A}, \mathbf{P}, r, \gamma)$ [9] representing a continuous set of states $\mathcal{S}$, a continuous set of actions $\mathcal{A}$, the transition probability $\mathbf{P}$ modeled by the probability density function for transitioning to the next state S_{i+1} when an action A_i is taken at the current state S_i, the immediate reward $r(S_i, A_i)$ attained by taking action A_i in the current state S_i, and the discount factor γ.

The spectral efficiency C_i for a time slot i on transmission from S to D by taking the action P_i at state S_i is given by

$$C_i(S_i, P_i) = \log_2\left(1 + \frac{P_i|H_i|^2}{N_P}\right). \tag{6}$$

Here N_P is noise power with reward function at slot i being

$$r_i(S_i, P_i) = \begin{cases} C_i(S_i, P_i), & \text{if } i = 0 \\ C_i(S_i, P_i) - \beta E_{i-1}^U, & \text{if } i \in (0, N] \end{cases} \tag{7}$$

where β is a penalty constant having units bps per Hz-J. We have further refined our reward function (7) by defining an adaptive β for a time slot $i \; \forall \; i \in (0, N]$ as

$$\beta = \varepsilon^{\lfloor i/l \rfloor}. \tag{8}$$

Here ε is a sensitivity parameter for the penalty constant β and l is the length of sliding window over which β will remain constant. The adaptive β brings dynamicity in the reward function and indicates the source (with higher β in later time slots) about the consequences of accumulating borrowed energy. The adaptive β ensures that borrowing is encouraged at all the time slots, but hoarding of borrowed energy is punishable and the penalty is awarded to the transmitting information source. Mathematically, objective is defined as

$$\max_{P_i} \sum_{i=0}^{N} \log_2\left(1 + \frac{P_i|H_i|^2}{N_P}\right). \tag{9}$$

Additionally, all the borrowed energy should be returned with due interests at the end of all N time slots.

IV. Proposed Solution Methodology

We consider the Model-free policy gradient methods of RL that focus on learning the parameterized policy directly using the gradient of a performance measure $J(\boldsymbol{\theta})$ with respect to policy parameter $\boldsymbol{\theta}$ [7]. Unlike value-based methods, these methods do not have complicacies due to continuous state-action space and have guaranteed convergence (at least, to local optimum) [7]. The effect of the policy parameter $\boldsymbol{\theta}$ on actions $\mathcal{A}$ and subsequently on rewards r, When a state s is given, can be determined by comprehending the parameterization, but the effect of policy on state distribution (being a function of the environment) is generally unknown[7], [10]. Hence, the policy gradient theorem was introduced to provide an analytical expression for the gradient of the performance $J(\boldsymbol{\theta})$ with respect to the policy parameter $\boldsymbol{\theta}$ and independent of the derivative of the state distribution [7]. The general form of policy gradient for the episodic task (with finite horizon) is

$$\nabla J(\boldsymbol{\theta}) \propto \sum_s \mu(s) \sum_a \; q_\pi(s, a) \; \nabla\pi(a|s, \boldsymbol{\theta}) \tag{10}$$

where the gradients are column vectors of partial derivatives with respect to the components of $\boldsymbol{\theta}$, and π denotes the policy with parameter vector $\boldsymbol{\theta}$. $\mu(s)$ describes the stationary distribution of the Markov Chain under the policy π. The proportionality constant for an episodic task is given by the average length of an episode and for a continuous task, it is equal to 1, denoting the equality[7]. We adopt hybrid actor critic approach to combine value-based and policy gradient methods. The actor takes the action according to policy, and learns from the feedback values given by the critic which act as a baseline for the actor to further improve the policy [11].

A. Role of Actor

The actor focuses on optimizing the parameterized stochastic policy $\boldsymbol{\pi}(P_i|S_i, \boldsymbol{\theta})$ using policy gradient methods [7]. The policy gradient theorem maximizes the average value of the states ($J(\boldsymbol{\theta})$). The performance measure for a continuous process is $J(\boldsymbol{\theta}) = \int_\mathcal{S} \mathrm{d}^\pi(S_i) \int_\mathcal{P} \boldsymbol{\pi}(P_i|S_i, \boldsymbol{\theta}) \; q_{\boldsymbol{\pi}}(S_i, P_i) \; \mathrm{d}P_i \; \mathrm{d}S_i$, where $d^\pi(S_i)$ is the stationary distribution of the MDP in accordance to the policy $\boldsymbol{\pi}(P_i^T|S_i, \boldsymbol{\theta})$ and $q_{\boldsymbol{\pi}}(S_i, P_i^T)$ is the action-value for the state-action pair (S_i, P_i^T) under the parameterized policy $\boldsymbol{\pi}(P_i^T|S_i, \boldsymbol{\theta})$ [7]. The policy parameter $\boldsymbol{\theta}$ follows gradient ascent [12] as $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \; \nabla J(\boldsymbol{\theta})$, where α is the learning rate, and ∇ is the gradient of the performance measure $J(\boldsymbol{\theta})$ with respect to the parameter $\boldsymbol{\theta}$.

For a differentiable policy, $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$ can be approximated to

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \approx E_{\boldsymbol{\pi}}[\nabla_{\boldsymbol{\theta}} \; \ln(\boldsymbol{\pi}(P_i|S_i, \boldsymbol{\theta})) \; Q(S_i, P_i, \boldsymbol{w})] \tag{11}$$

where $Q(S_i, P_i, \boldsymbol{w})$ is action-value function approximated by the critic network [7]. The weight vector $\boldsymbol{w}$ is used by the critic network to approximate $Q(S_i, P_i, \boldsymbol{w})$ as described in section IV-B. The update equation for policy parameter θ is

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \; \nabla_{\boldsymbol{\theta}} \; \ln(\boldsymbol{\pi}(P_i|S_i, \boldsymbol{\theta})) \; Q(S_i, P_i, \boldsymbol{w}) \tag{12}$$

where $\nabla_{\boldsymbol{\theta}} \; \ln(\boldsymbol{\pi}(P_i|S_i, \boldsymbol{\theta})) \; Q(S_i, P_i, \boldsymbol{w})$ is the stochastic estimate for the approximation of $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$ [7].

Our policy $\boldsymbol{\pi}(P_i|S_i, \boldsymbol{\theta})$ as modeled by a Gaussian distribution function with a parameterized mean $\boldsymbol{\mu}(S_i, \boldsymbol{\theta}_\mu)$ and standard deviation $\boldsymbol{\sigma}(S_i, \boldsymbol{\theta}_\sigma)$ and can be written as

$$\boldsymbol{\pi}(P_i|S_i, \boldsymbol{\theta}) = \frac{1}{\sqrt{2\pi(\boldsymbol{\sigma}(S_i, \boldsymbol{\theta}_\sigma))^2}} \; e^{-\frac{(t_s P_i - \boldsymbol{\mu}(S_i, \boldsymbol{\theta}_\mu))^2}{2(\boldsymbol{\sigma}(S_i, \boldsymbol{\theta}_\sigma))^2}} \tag{13}$$

where the parameter $\boldsymbol{\theta} = [\boldsymbol{\theta}_\mu \ \boldsymbol{\theta}_\sigma]^\top$ and proposed mean $\boldsymbol{\mu}(S_i, \boldsymbol{\theta}_\mu)$ is a bounded function and we use hyperbolic tangent function to satisfy the constraint [4] and is calculated using

$$\boldsymbol{\mu}(S_i, \boldsymbol{\theta}_\mu) = \max\left\{0, E^A\left(\frac{1+\tanh(\boldsymbol{\theta}_\mu^\top \boldsymbol{\phi}(S_i))}{2}\right)\right\} \quad (14)$$

where $E^A = \left(B_{i-1} + E_i^H - E_{i-1}^U\right)$ (the effective energy level), and $\boldsymbol{\phi}(S_i)$ is the feature vector providing additional information about the given state S_i. The feature vector $\boldsymbol{\phi}(S_i) = [\phi_1(S_i), \phi_2(S_i)]$ comprises two binary functions $\phi_1(S_i)$ and $\phi_2(S_i)$ such that, $\phi_1(S_i) = 1$ if the battery energy level exceeds its minimum value and $\phi_2(S_i) = 1$ if the battery energy level has achieved its maximum value. The features $\phi_1(S_i)$ and $\phi_2(S_i)$ are set to zero, in all other cases.

The standard deviation should always be positive, hence modeled through an exponential function [4] given as

$$\sigma(S_i, \boldsymbol{\theta}_\sigma) = \exp\left(\ \boldsymbol{\theta}_\sigma^\top \boldsymbol{\phi}(S_i)\right) \quad (15)$$

Our policy parameter vector $\boldsymbol{\theta}$ updates as follows

$$\boldsymbol{\theta}_\mu \leftarrow \boldsymbol{\theta}_\mu + \alpha\left[\nabla_{\boldsymbol{\theta}_\mu} \ln(\boldsymbol{\pi}(P_i|S_i, \boldsymbol{\theta}))\ Q(S_i, P_i, \boldsymbol{w})\right], \quad (16)$$

$$\boldsymbol{\theta}_\sigma \leftarrow \boldsymbol{\theta}_\sigma + \alpha\left[\nabla_{\boldsymbol{\theta}_\sigma} \ln(\boldsymbol{\pi}(P_i|S_i, \boldsymbol{\theta}))\ Q(S_i, P_i, \boldsymbol{w})\right]. \quad (17)$$

The selection of transmission power at each time slot i is as per the proposed Gaussian policy (13). However, in last k time slots except the N-th slot, the transmission power follows

$$P_{N-k} = \begin{cases} \max\{0, E_{N-k}^H - E_{N-k}^U\}, & \text{if } E_{N-k}^U > \frac{k}{\kappa}\frac{\sum_{i=0}^{N-k} E_i^H}{N-k} \\ \text{Sampled from } \boldsymbol{\pi} \text{ (13)}, & \text{otherwise} \end{cases} \quad (18)$$

To ensure that all the borrowed energy is returned to the power grid, the policy is updated to follow (18). It should be noted that $k \propto (E_{\max}^U / \text{Average}\{E^H\})$ and secures the sufficient amount of energy such that in last k time slots, the pool of unreturned energy can be returned efficiently. For the last or Nth time slot, we define $P_N = \max\{0, E_N^H - E_N^U\}$. Numerical constant $\kappa \in (0, 1]$ provides an essential role in determining the margin due to the variance in harvested energy.

B. Role of Critic

The critic network of the actor-critic algorithm is responsible for the approximation of the action-value function and *"criticize"* the policy evaluated by the actor. Our work uses a three layered neural network to approximate the action-value function $Q(S_i, P_i, \boldsymbol{w})$, where $\boldsymbol{w}$ is the weight vector for the network using the temporal difference error δ [7].

To tackle instability issues in deep RL, we use a separate neural network with weights $\boldsymbol{w_t}$ to approximate the target action-value function [14]. Another reason to employ a target network is the correlation dynamics between the action-values and target values. The small updates to Q during the learning process may considerably modify the policy. Consequently, it would bring changes in data distribution and the correlation between the action-value and target values[14]. Hence, we update parameters of target action-value function $\hat{Q}(S_i, P_i, \boldsymbol{w}_t)$ every C training iterations to preserve the correlation[14]. This allows us to define the temporal difference error δ as

$$\delta = \{r_i + \gamma \hat{Q}(S_{i+1}, P_{i+1}, \boldsymbol{w}_t)\} - Q(S_i, P_i, \boldsymbol{w}). \quad (19)$$

Our critic neural network learns by back-propagating the absolute value of temporal difference error using stochastic gradient descent[15] with learning rate α_c.

Algorithm 1: BEAR Algorithm

Input : Lower and upper bounds: $B_{\min}$, $B_{\max}$, $E_{\min}^H$, $E_{\max}^H$, $H_{\min}$, $H_{\max}$, $P_{\max}$, $E_{\max}^U$, and known parameters t_s, ζ, η, ε, γ, σ_d, N_P, k, C, M, N
Output: Optimal policy parameter $\boldsymbol{\theta} = [\boldsymbol{\theta}_\mu \ \boldsymbol{\theta}_\sigma]^\top$

Initialize $\boldsymbol{\theta}_\mu$, $\boldsymbol{\theta}_\sigma$ with values sampled from standard uniform distribution in range $[0, 1]$
Initialize weights $\boldsymbol{w}$ of the action-value function with Xavier weight initialization process[13]
Initialize target network with weights $\boldsymbol{w}_t = \boldsymbol{w}$
for *trial* $= 0$ **to** $M - 1$ **do**
  Initialize B_0 with uniform distribution in $[B_{\min}, B_{\max}]$
  Set $\mu_E = (E_{\max} + E_{\min})/2$ and $\mu_H = (H_{\max} + H_{\min})/2$
  Sample E_i^H from a truncated normal probability distribution function $f_{\mathcal{E}_\mathcal{N}}$ in $[E_{\min}^H, E_{\max}^H]$ with mean μ_E, variance σ_d
  Sample H_i from a truncated normal probability distribution function $f_{\mathcal{H}_\mathcal{N}}$ in $[H_{\min}, H_{\max}]$ with mean μ_H, variance σ_d
  for *timeslot* $i = 0$ **to** $N - 1$ **do**
    Set $S_i = (B_i, E_i^H, H_i)$
    Get features $\boldsymbol{\phi}_1(S_i)$, $\boldsymbol{\phi}_2(S_i)$
    Determine $\boldsymbol{\mu}$, $\boldsymbol{\sigma}$ for actor using (14), (15) respectively
    if $i = N - 1$ **then**
      $P_i = \max\{0, E_i^H - E_i^U\}$
    else if $i > N - k$ *and* $i \neq N - 1$ **then**
      Select P_i as per (18)
    else
      Select P_i using (13)
    Determine E_i^B, E_i^R, E_i^I using (2), (3), (5) respectively
    Get E_i^U using (4)
    Observe the reward r_i using (7)
    Sample E_{i+1}^H from a truncated normal probability distribution function $f_{\mathcal{E}_\mathcal{N}}$ in $[E_{\min}^H, E_{\max}^H]$ with mean μ_E, variance σ_d
    Sample H_{i+1} from a truncated normal probability distribution function $f_{\mathcal{H}_\mathcal{N}}$ in $[H_{\min}, H_{\max}]$ with mean μ_H, variance σ_d
    Calculate B_{i+1} using (1)
    Set $S_{i+1} = (B_{i+1}, E_{i+1}^H, H_{i+1})$
    Determine $\boldsymbol{\mu}$, $\boldsymbol{\sigma}$ using (14), (15) respectively
    if $i + 1 = N - 1$ **then**
      Select $P_{i+1} = \max\{0, E_{i+1}^H - E_{i+1}^U\}$
    else if $i + 1 > N - k$ *and* $i + 1 \neq N - 1$ **then**
      Select P_{i+1} as per (18)
    else
      Select P_{i+1} using (13)
    Calculate temporal difference error δ (19)
    Update weights $\boldsymbol{w}$ using back propagation of the temporal difference error δ
    Update $\boldsymbol{\theta}_\mu$, $\boldsymbol{\theta}_\sigma$ using (16), (17) respectively
  Every C trials, update target weights $\boldsymbol{w}_t \leftarrow \boldsymbol{w}$

C. Implementation Details

The proposed BEAR Algorithm is procedurally shown in Algorithm 1 and it takes input of lower and upper bounds, along with the known parameters. At the end of M trials, our algorithm converges (explained in Section V) to provide the optimal policy parameter $\boldsymbol{\theta}$. The algorithm is implemented in Python 3.6 with PyTorch deep learning framework in an Anaconda Environment. The training and testing was done on an Intel® Core™ i9-7900X CPU with 62.6 GiB RAM on Ubuntu 18.04.4 LTS release along with NVIDIA® Quadro® GV100 32.5 GiB Graphics Processor.

V. Numerical Performance Evaluation

A. Simulation Environment and Default Parameter Values

In this section, we present the numerical results. Our experimentation has assumed the length of each time slot $t_s = 1$s, hence, we have used the term transmission power and transmission energy interchangeably. The noise power N_P is set to 4×10^{-15} W, and PG has infinite source of energy. The battery has maximum capacity of storing energy $(B_{\max})$ equivalent to 3 J. We have considered a solar panel is attached with S for harvesting energy, and it has an area of 100 cm^2 producing the output of 100 mW/cm^2 with a maximum efficiency of 15%. This allows the value of maximum harvested energy at a particular time slot $(E^H_{\max})$ to be 1.5 J. We have also assumed the maximum channel coefficient $H_{\max} = 1.0$ along with maximum value of $P_{\max} = 10$ W. The maximum (available) pool of unreturned energy $(E^U_{\max})$ is 5 J. The limitation in transmission power and unreturned energy at each time slot provide the learning transmission node a sense of available constraints for achieving its goals. The values of $B_{\min}$, $E^H_{\min}$ and $H_{\min}$ are set to 0 in their respective units. The mean of the normal probability distribution functions $f_{\mathcal{E}_\mathcal{N}}$ and $f_{\mathcal{H}_\mathcal{N}}$ is given by $\mu_{\text{E}} = 0.75$ and $\mu_{\text{H}} = 0.5$ respectively. The variance of both normal distribution functions is given by $\sigma_d = 0.5$. The rate of interest for borrowing energy (η) is 10%, and the length of sliding window l for which the sensitivity parameter ε remains constant in adaptive penalty β (8) is set to 10 time slots. Furthermore, we have set $k = 20$, and $\kappa = 1$ in the transmission power for last k time slots (18).

We trained our algorithm for $M = 60000$ training epochs and for $N = 100$ time slots, for different sensitivity parameters $\varepsilon \in [1.00, 1.21]$. The learning rates of actor (α) and for critic (α_C) are set to 2×10^{-4} and 1×10^{-4} respectively. The Stochastic Gradient Descent Optimizer [15] was used to optimize the critic network. The discount factor $\gamma = 0.9$ for calculating the temporal difference error (19). The parameters of the target network $(\boldsymbol{w}_t)$ are updated every $C = 1$ trials.

B. Performance Comparison and Validation

We introduced the concept of smartly borrowing energy to facilitate the data transmission to exploit good environmental conditions. In Fig. 2, we compare performance of the proposed BEAR Algorithm with different numerical values of the sensitivity parameter ε of the adaptive penalty constant β in (8). The

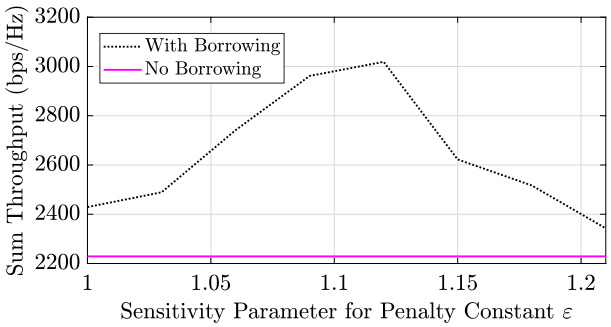

Fig. 2: Sum Throughput for $N = 100$ time slots.

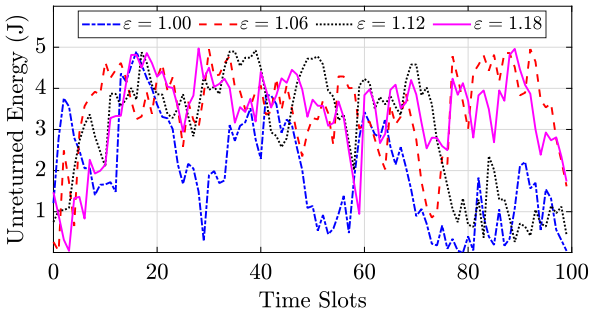

Fig. 3: Unreturned Energy at i-th time slot for different ε

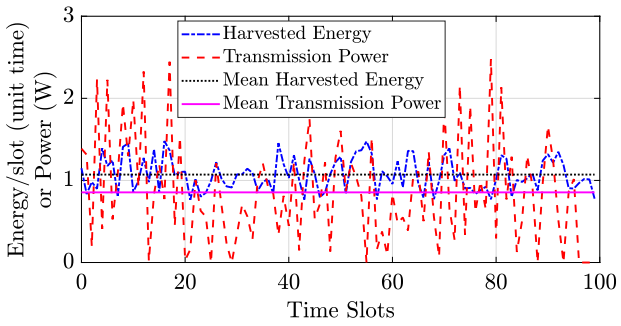

Fig. 4: Illustration of Energy for every time slot for $N = 100$.

graph lucidly illustrates that there exists an optimal value of sensitivity parameter ε for sum throughput. As the sensitivity parameter ε increases beyond the optimal point, the penalty increases on the transmission agent implicitly discouraging the borrowing from secondary power sources such as the power grid PG in our case. Also, it can be clearly noted that the propose BEAR algorithm clearly outperforms the non-borrowing benchmark, thus corroborating its practical utility.

Next, via Fig. 3, we demonstrate the utility of the introduced novel adaptive penalty (8) as a function of ith time slot that allows the agent to make an informed decision on the cumulative amount of unreturned energy. Higher is the unreturned energy, more difficult it is to return. Hence, the adaptive penalty, an exponential function described by (8), adapts itself with time to increase the penalty at later time slots. It allows our transmitting node to borrow energy (if the channel conditions allow) in initial time slots and return the

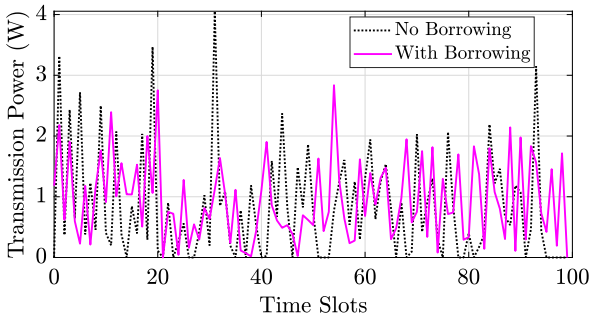

Fig. 5: Transmit power per slot, with and without borrowing.

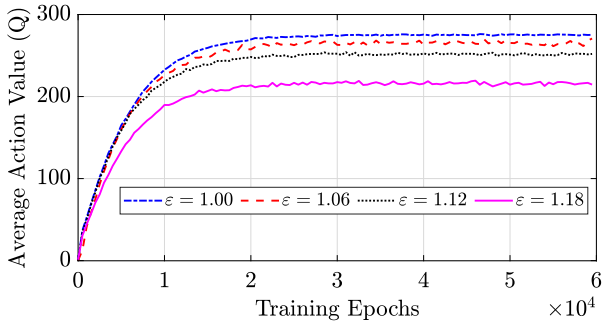

Fig. 6: Average predicted action value (Q).

pool of unreturned energy in later time slots. The numerical value of ε affects the pool of unreturned energy at each time slot as depicted by Fig. 3. Additionally, as evident from Fig. 3, our proposed BEAR Algorithm ensures that all the unreturned energy is returned by the transmission agent in the later slots, shown by the dip in the values of pool of unreturned energy.

For time slots $N = 100$, Fig. 4 plots the distribution of harvested energy and energy consumed during the transmission against N time slots. The harvested energy is completely utilized in transmission of data and in energy costs incurred as a result of interest during returning of energy. The difference between the mean harvested energy and mean transmission power in Fig. 4 denotes energy given to the secondary power source (such as power grid PG) in the form of interest as described by (5). Similarly, Fig. 5 shows a descriptive transmission power in each of the time slot for an EH with and without borrowing. These results in Figs. 4 and 5 shed key nontrivial design insights on the optimal energy schedules as obtained via the proposed machine learning framework.

Lastly via Fig. 6, we discourse the fast convergence of proposed RL algorithm BEAR in the form of how the state-action value function $Q(S_i, P_i)$ estimates the Q-value for all possible actions in a particular state S_i. If an optimal action P_i^* is taken at a state S_i, P_i^* will yield higher Q-value than a non-optimal action P_i at S_i. The average of Q-values over N time slots for each training epoch is shown in Fig. 6, where each point is average of all state-action values for a particular training iteration over all N time slots. This saturation after 30000 training epochs indicates the proposed policy can be used readily and efficiently for EH IoT devices.

VI. Concluding Remarks

We proposed an enhanced EH wireless communication system with borrowing, which focuses on formulating a power allocation policy to optimize the harvested energy and sum throughput jointly. Since the (harvested) energy arrival and channel conditions are random and their pattern is unknown in a complex real-life environment, we resort to deep RL methods to obtain an optimal policy. The proposed BEAR algorithm is equipped with the concept of adaptive rewards (in the form of a penalty) to ensure that the transmission agent can extract its goal of joint optimization without facing any hurdles. The sensitivity parameter ε in the adaptive penalty constant β shows that there exists an optimal value of ε for throughput maximization. The simulation results prove that with the introduction BEAR, there is a significant gain of 35.45% in sum throughput over a non-borrowing EH system. The future work will include the EB in multi-user and multi-antenna EH systems, using multi-agent deep RL.

References

[1] K. Tutuncuoglu and A. Yener, "Energy harvesting networks with energy cooperation: Procrastinating policies," *IEEE Transactions on Communications*, vol. 63, no. 11, pp. 4525–4538, 2015.

[2] Z. Sun, L. Dan, Y. Xiao, P. Yang, and S. Li, "Energy borrowing for energy harvesting wireless communications," *IEEE Communications Letters*, vol. 20, no. 12, pp. 2546–2549, 2016.

[3] Y. Wei, F. R. Yu, M. Song, and Z. Han, "User scheduling and resource allocation in hetnets with hybrid energy supply: An actor-critic reinforcement learning approach," *IEEE Transactions on Wireless Communications*, vol. 17, no. 1, pp. 680–692, 2017.

[4] A. Masadeh, Z. Wang, and A. E. Kamal, "An actor-critic reinforcement learning approach for energy harvesting communications systems," in *Proc. ICCCN*, July 2019, pp. 1–6.

[5] A. Masadeh, Z. Wang, and A. E. Kamal, "Reinforcement learning exploration algorithms for energy harvesting communications systems," in *Proc. IEEE ICC*, 2018, pp. 1–6.

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

[7] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: A Bradford Book, 2018.

[8] G. K. Reddy, D. Mishra, and L. N. Devi, "Scheduling protocol for throughput maximization in borrowing-aided energy harvesting system," *IEEE Networking Letters*, pp. 1–1, 2020.

[9] E. Brunskill and R. Sarkar, "Lecture 2: Making good decisions given a model of the world," in *CS234: Reinforcement Learning – Stanford University*, Stanford CA, 2019. [Online]. Available: https://web.stanford.edu/class/cs234/CS234Win2019/slides/lnotes2.pdf

[10] J. Peters, "Policy gradient methods," *Scholarpedia*, vol. 5, no. 11, p. 3698, 2010, revision #137199.

[11] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "A brief survey of deep reinforcement learning," *CoRR*, vol. abs/1708.05866, 2017. [Online]. Available: http://arxiv.org/abs/1708.05866

[12] J. Sorg, R. L. Lewis, and S. Singh, "Reward design via online gradient ascent," in *Advances in Neural Information Processing Systems*, J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, Eds., vol. 23. Curran Associates, Inc., 2010, pp. 2190–2198.

[13] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. International Conference on Artificial Intelligence and Statistics*, 2010, pp. 249–256.

[14] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[15] L. Bottou, "Online learning and stochastic approximations," *On-line learning in neural networks*, vol. 17, no. 9, p. 142.