2018

Reinforcement Learning Exploration Algorithms for Energy Harvesting Communications Systems

Ala'eddin Masadeh
Iowa State University, amasadeh@iastate.edu

Zhengdao Wang
Iowa State University, zhengdao@iastate.edu

Ahmed Kamal
Iowa State University, kamal@iastate.edu

Reinforcement Learning Exploration Algorithms for Energy Harvesting Communications Systems

Abstract
Prolonging the lifetime, and maximizing the throughput are important factors in designing an efficient communications system, especially for energy harvesting-based systems. In this work, the problem of maximizing the throughput of point-to-point energy harvesting communications system, while prolonging its lifetime is investigated. This work considers more real communications system, where this system does not have a priori knowledge about the environment. This system consists of a transmitter and receiver. The transmitter is equipped with an infinite buffer to store data, and energy harvesting capability to harvest renewable energy and store it in a finite battery. The problem of finding an efficient power allocation policy is formulated as a reinforcement learning problem. Two different exploration algorithms are used, which are the convergence-based and the epsilon-greedy algorithms. The first algorithm uses the action-value function convergence error and the exploration time threshold to balance between exploration and exploitation. On the other hand, the second algorithm tries to achieve balancing through the exploration probability (i.e. epsilon). Simulation results show that the convergence-based algorithm outperforms the epsilon-greedy algorithm. Then, the effects of the parameters of each algorithm are investigated.

Keywords
Learning (artificial intelligence), Energy harvesting, Communication systems, Batteries, Markov processes, Throughput, Transmitters

Comments
This is a manuscript of a proceeding published as Masadeh, Alaeddin, Zhengdao Wang, and Ahmed E. Kamal. "Reinforcement Learning Exploration Algorithms for Energy Harvesting Communications Systems." In *2018 IEEE International Conference on Communications (ICC)*. 2018. DOI: 10.1109/ICC.2018.8422710. Posted with permission.

Reinforcement Learning Exploration Algorithms for Energy Harvesting Communications Systems

Ala'eddin Masadeh, Zhengdao Wang, Ahmed E. Kamal
Iowa State University (ISU), Ames, IA 50011, USA,
emails: {amasadeh,zhengdao,kamal}@iastate.edu

***Abstract*—Prolonging the activation period, and maximizing the throughput are important factors in designing an efficient communications system, especially for energy harvesting-based systems. In this work, the problem of maximizing the throughput of point-to-point energy harvesting communications system, while prolonging its lifetime is investigated. This work considers more real communications system, where this system does not have a priori knowledge about the environment. This system consists of a transmitter and receiver. The transmitter is equipped with an infinite buffer to store data, and energy harvesting capability to harvest renewable energy and store it in a finite battery. The problem of finding an efficient power allocation policy is formulated as a reinforcement learning problem. Two different exploration algorithms are used, which are the convergence-based and the epsilon-greedy algorithms. The first algorithm uses the action-value function convergence error and the exploration time threshold to balance between exploration and exploitation. On the other hand, the second algorithm tries to achieve balancing through the exploration probability (i.e. epsilon). Simulation results show that the convergence-based algorithm outperforms the epsilon-greedy algorithm. Then, the effects of the parameters of each algorithm are investigated.**

***Index Terms*—Energy harvesting communications, Markov decision process, Reinforcement learning, Exploration, Exploitation.**

I. Introduction

Energy harvesting has been considered as an efficient solution that provides more sustainable wireless communication systems. Energy harvesting communications have been introduced to find communication nodes that are able to recharge their batteries using natural sources, and then use this energy for data transmission [1]. To find efficient energy harvesting communications systems, it is needed to optimize their parameters such as transmission power.

Based on the available knowledge about the environment, there are two main frameworks used to optimize energy harvesting systems [2]. The first one is the offline framework, where communications systems have non-causal information about the environment. The second framework is the online approach. This framework is more realistic, where the system performance is optimized based on the available statistical information about the environment [3]. The Markov decision process (MDP) is one of the techniques that are able to formulate such decision-making problems [2].

In the previous two frameworks, a priori knowledge, either deterministic or statistical, of the energy harvesting process is required. However, in more practical scenarios, this knowledge might be unavailable, or the energy harvesting process is non stationary that makes it challenging to be tracked [2], [3].

To solve such challenges, the well-known learning approach that is called reinforcement learning is used to optimize the performance of such systems [3]. Reinforcement learning is considered as an efficient technique, which enables an autonomous agent to select optimal actions at different states in an unknown environment [4].

In [3], [5], the problem of optimizing energy harvesting communications systems are investigated. In this context, at each time, the energy harvesting nodes have only current local knowledge of the energy harvesting process. The authors aim to find a power allocation policy that maximizes the throughput. In these two works, the reinforcement learning algorithm, which is known as the state action reward state action (SARSA), is used to evaluate the taken actions. On the other hand, the ϵ-greedy exploration algorithm is used to balance between exploring and exploiting available actions.

In [2], a point-to-point communications system is investigated. The transmitter is capable to harvest energy and store it in rechargeable battery. The energy and data arrivals are formulated as Markov processes. In this work, the authors use Q-learning to find the optimal transmission policy when the system does not have a priori information on the Markov processes governing the system. They use ϵ-greedy exploration algorithm to balance between exploration and exploitation.

Balancing between exploiting the current greedy policy, and exploring new policies that may have better performance than the current greedy policy is known as the exploration-exploitation dilemma [6]. Balancing problem is one of the main challenges facing reinforcement learning. Finding a balancing criteria between exploration and exploitation contributes in maximizing the cumulative rewards, which is the goal of the reinforcement learning.

This balancing dilemma has been discussed intensively in the literature [7]–[13]. Boltzmann and ϵ-greedy exploration algorithms are considered as the most popular exploration algorithms [14], where these two methods are intensively used in the literature [7]–[12]. These two algorithms use random action selection to evaluate new actions [14]. In ϵ-greedy, the agent takes a new action from uniformly distributed action set with probability ϵ, while selects the greedy action with probability $1-\epsilon$ [9]. Boltzmann or softmax exploration algorithm is characterized by using Boltzmann distribution for assigning selection probability to different actions [10].

In this work, a real point-to-point communications system is studied. This communications system does not have a priori knowledge about the environment. The goal is to optimize the transmission power to prolong its battery life and maximize its throughput. Reinforce learning is used to solve this problem. SARSA learning algorithm is used to evaluate different actions. The performance of proposed model is investigated using two different exploration algorithms, which are the convergence-based algorithm, and the ϵ-greedy algorithm. The convergence-based algorithm tries to balance between exploration and exploitation using two parameters, which are the exploration time threshold τ, and the action-value function convergence error ζ. In the first session of this algorithm, the agent tries to evaluate available actions, and then it exploits the best resulted policy in the remaining time. On the other hand, the ϵ-greedy tries to find a balance point between exploration and exploitation through the exploration probability ϵ. Then, We show the performance of proposed model using different methods. It is noticed that the convergence-based algorithm outperforms the ϵ-greedy algorithm in our numerical experiments. Finally, the effects of the parameters of each exploration algorithm are studied.

II. REINFORCEMENT LEARNING

In this section, the reinforcement learning framework is explained, which will be used in later sections. Firstly, Markov Decision Processes is presented. Secondly, State-action-reward-state-action (SARSA) learning algorithm is described. Finally, the term of exploration algorithms is illustrated.

A. Markov Decision Processes

In general, Markov decision process (MDP) is used to describe an environment for reinforcement learning [15]. An MDP can be described by the following elements:

1. A set of states $\mathcal{S}$, which consists of discrete states $\mathcal{S} \triangleq \{s^1, s^2, ..., s^{N_s}\}$, where N_s is the number of possible states. The state at time slot i is denoted by s_i, where $s_i \in \mathcal{S}$.

2. A set of discrete actions $\mathcal{A}$, where $\mathcal{A} = \{a^1, a^2, ..., a^{N_a}\}$, and N_a is the number of available actions. Each state s has a subset of actions $\mathcal{A}^s$ such that $\mathcal{A}^s \in \mathcal{A}$. At time slot i, the executed action is denoted by a_i, where $a_i \in \mathcal{A}$.

3. Transition probabilities between states, where $p(s, a, s')$ is the transition probability from current state s to next state s', given that the action a is selected at the state s.

4. The immediate reward $R(s, a, s)$, which is the obtained reward when transiting from state s to state s' such that the action a is selected at state s.

5. A discount factor $\gamma \in [0, 1]$. It is used to weight the immediate reward relative to future rewards. In general, this factor has a value less than one to guarantee that the cumulative rewards is finite given that the immediate reward is bounded [16].

With the MDP defined, there is an important definition that should be visited, which is the policy. The deterministic policy $\pi(s)$ can be defined as mapping the visited states to actions to be taken at these states. In reinforcement learning, the goal is to find the optimal policy π^*, which is mapping the visited states to the optimal actions that maximize the expected cumulative reward over an infinite horizon [15]. The expected sum reward is given by:

$$\mathbb{E}\left[\sum_{i=1}^{\infty} \gamma^i R(s_i, a_i, s_{i+1}) | a_i = \pi(s_i)\right] \tag{1}$$

For a state s, let us define two important functions, which are the state-value function v_π, and the action-value function q_π. The state-value function is the expected reward given that the agent follows the policy π starting from state s [6]

$$v_\pi(s) = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{j+k+1} \,\middle|\, s\right] \tag{2}$$

The action-value function is the expected reward starting from state s, selecting action a and following policy π thereafter [6]:

$$q_\pi(s, a) = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{j+k+1} \,\middle|\, s, a\right] \tag{3}$$

The optimal state-value function for state s, and the optimal action-value function for the state-action pair (s, a) are given, respectively, by:

$$v_{\pi^*}(s) = \max_\pi v_\pi(s) \qquad \forall s \in \mathcal{S} \tag{4}$$

$$q_{\pi^*}(s, a) = \max_\pi q_\pi(s, a) \qquad \forall a \in \mathcal{A}^s, \forall s \in \mathcal{S} \tag{5}$$

From (4), (5)

$$v_{\pi^*}(s) = \max_a q_{\pi^*}(s, a) \qquad \forall s \in \mathcal{S} \tag{6}$$

A main property for action-value function is that it can be written recursively in the form that is known as the Bellman equation [6]. The Bellman equation for the action-value function is given by:

$$q_\pi(s, a) = \sum_{s' \in \mathcal{S}} P(s, a, s')\big[R(s, a, s') + \gamma v_\pi(s')\big] \tag{7}$$

B. State-action-reward-state-action (SARSA)

In this work, SARSA learning is used to estimate the action-value function for different state-action pairs. SARSA is an on-policy updating strategy, which attempts to evaluate the policy that is used to make decisions. On the other hand, in off-policy methods, the value function is estimated for the policy that may be unrelated to the policy used for evaluation [6].

Updating in SARSA works as follows. Starting from the time slot i, let the agent be at state s (i.e. $s_i = s$), and the selected action according to the current policy π be $a_i = a$. Based on the selected action, it moves to the next state s_{i+1} (i.e. $s_{i+1} = s'$) and receives a reward $R(s, a, s)$. According to the current policy π, the action $a_{i+1} = a'$ is selected for the state s_{i+1}. At this point, the action-value function $q^i_\pi(s, a)$ is

updated using the gained experience. The updating equation in SARSA is given by [3]

$$q_\pi^{i+1}(s,a) \leftarrow q_\pi^i(s,a) + \alpha[R(s,a,s) + q_\pi^{i+1}(s',a') - q_\pi^i(s,a)] \quad (8)$$

where $0 < \alpha < 1$ refers to the learning rate. This factor determines the amount of contribution of the newly acquired information for updating the the action-value function. If $\alpha = 0$, then the agent will not learn any thing from the acquired information. On the other hand, if $\alpha = 1$, the agent will only consider the newly acquired information [17].

III. The Exploration Algorithms

The exploration algorithms play an essential role in reinforcement learning. Their role appears in finding a balance between exploration and exploitation to maximize the cumulative rewards. The exploitation mode can be defined as using the current available knowledge to select the best policy to be used. On the other hand, exploration is known as investigating new policies in the hope of getting policy that is better than the current best one [6].

A. The ϵ-greedy algorithm

This algorithm [9] uses the exploration probability ϵ to find a balancing point between exploration and exploitation modes. This parameter changes the mode based on its value at each time slot.

In this algorithm, the current best action is selected with probability $1 - \epsilon$. On the other hand, a random non-greedy action is selected with probability ϵ. The ϵ can be either fixed [6], or with adaptive value during the learning time [11]. In the case of adaptive ϵ-greedy, ϵ takes values that changes with time. For example, in [11], ϵ is set to $1/i$, where i the time slot number. In this case, at the beginning of the session, the exploration probability ϵ has large values to increase the probability of exploration. As the time increases, the probability of exploration decreases and the exploitation probability increases. This is to increase the opportunity of exploitation at the end of the session, where most of the policies have been explored and it is preferred to exploit the best current policy.

B. The convergence-based algorithm

This algorithm [13] uses two parameters to balance between exploration and exploitation. The first parameter is the action-value function convergence error ζ. This parameter measures the error in action-value function when the same action at a state is exploited for a number of trials. The second parameter is the exploration time threshold τ. This parameter controls the exploration process, where the agent can explore different actions for a maximum time of τ, after that, the agent is forced to exploit the best policy during the remaining time.

In this algorithm, random actions are assigned to all available states. At each state, the taken action is exploited for a time till its action-value function converges to a value determined by ζ. Once the action-value function for a state-action pair converges with an error ζ, a new random action is assigned from uniformly distributed unexplored actions to that state. This mechanism continues for all states, and it stops in two cases.

The first one occurs if all available actions for a states are evaluated before reaching τ. At this time, the action with the best value is exploited in the future. The second case occurs when the available time reaches τ. Then, the agent suspends exploration, and starts exploiting the best available policy regardless of exploring all available actions or not.

One of the main advantages of the convergence-based algorithm is that once an action for a state is explored, and the action-value function has converged to unfavorable value, this action will not be used in the future. This is an important property that contributes in discarding actions that may reduce the cumulative rewards in the future. One more characteristic is that it assigns dynamic learning time for different state-action pairs.

IV. System Model

In this section, a point-to-point communications system consisting of a source (SR) and a destination (DE) is considered. As shown in Fig. 2, Both SR and DE are equipped with infinite data buffers to store data. SR is able to harvest renewable energy and store it in a finite battery. A time slotted system with time slots of equal length is considered. Each time slot consists of two equal sub-slots. The first sub-slot is used to transmit data from SR to DE. On the other hand, SR harvests energy in the second one. Fig. 1 illustrates the slotted time system model.

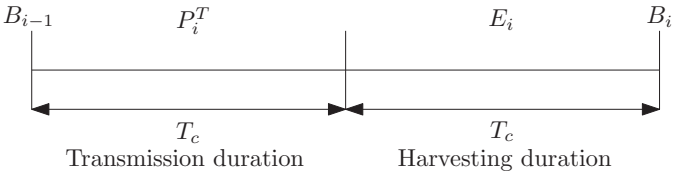

Fig. 1: Slotted system model.

In this context, the harvested, stored, and transmitted energy occurs in amounts that are an integer multiple of a fundamental unit. The battery has a limited storage capacity of B_{max}. Let B_i denote the battery charge level of SR at the beginning of time slot i, where $B_i \in \mathcal{B} \triangleq \{b_1, b_2, ..., b_{N_B}\}$, $b_{N_B} = B_{max}$, and N_B is the number of elements in $\mathcal{B}$.

During time slot i, the amount of harvested energy is denoted by E_i, where $E_i \in \mathcal{E} \triangleq \{e_1, e_2, ..., e_{N_E}\}$, and N_E represents the number of elements in $\mathcal{E}$. The transition probability of harvested energy from state e_j to state e_k during one time slot is given by $p_{\mathcal{E}}(e_j, e_k)$. The channel state during time slot i is given by H_i, where $H_i \in \mathcal{H} \triangleq \{h_1, h_2, ..., h_{N_H}\}$, and N_H denotes the number of elements in $\mathcal{H}$. The channel transition probability from state h_j to state h_k during one time slot is given by $p_{\mathcal{H}}(h_j, h_k)$.

Let the transmitted power during the time slot i be denoted by P_i^T, where $P_i^T \in \mathcal{P} \triangleq \{p_1^T, p_2^T, ..., p_{N_p}^T\}$, and N_p is the

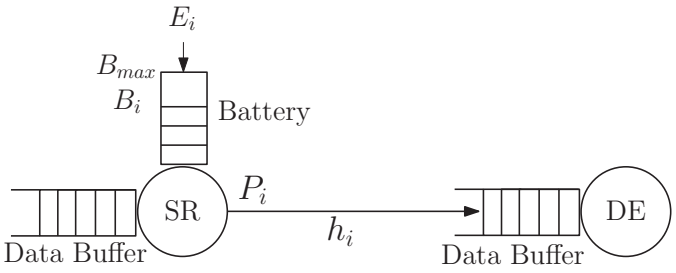

Fig. 2: Point-to-point communication system with an energy harvesting source.

number of elements in $\mathcal{P}$. Let T_c be the transmission duration, which has a fixed value of 1 second during all time slots.

For this model, each state of the system s_j consists of three elements, which are the battery level of SR, amount of harvested energy, and channel gain (i.e. $s_j = (b_j, e_j, h_j)$). In this context, the states satisfies Markov property, where the future state depends only on the current state, and independent of the system states in previous time slots [6].

Based on the current battery level, SR selects the action (i.e. transmission power level) that maximizes the cumulative sum rates. The immediate reward for this model is the achievable rate during time slot i, which is given by

$$R_i = \log_2\left(1 + \frac{|H_i|^2 P_i^T}{\sigma_i^2}\right) \tag{9}$$

where σ_i^2 is the noise variance.

In this model, energy consumption is considered only due to data transmission, and it does not take into account any other energy consumption, such as processing, circuitry, etc.

V. Simulation Results

In this section, the performance for different methods is evaluated. Then, the effects of the parameters are investigated for the convergence-based and the ϵ-greedy exploration algorithms.

In the numerical experiments, it is assumed that each time slot consists of two equal sub-slots, each of them is with 1 second duration. during the first sub-slot, the transmitter transmits its signal to the receiver, while during the second sub-slot, the transmitter harvests energy. The available bandwidth BW is 1 MHz, and the noise spectral density is $N_0 = 10^{-16}$ W/Hz. The discount factor γ is set to 0.99. The learning rate α is set to 0.1. All results are averaged over 1000 runs.

In these experiments, SR is equipped with solar panels with area of 100 cm^2 and 10% harvesting efficiency, where an outdoor solar panel can get the benefit of 10 mW/cm^2 solar irradiance under standard environments with harvesting efficiency between 5% and 30%, which depends on the used material in the panel [18]. It is also assumed that the fundamental energy unit for the net harvested, stored, and transmitted is 20 mJ.

In all simulations, it is assumed that the set of harvested energy is $\mathcal{E} = \{0, 1\}$ corresponding to fundamental energy unit with transition probabilities $p_{\mathcal{E}}(e_j, e_j) = 0.8$. Let the set of channel gains be $\mathcal{H} = \{0.022361, 6.7082\} \times 10^{-7}$ with transition probabilities $p_{\mathcal{H}}(h_j, h_j) = 0.9$. The equipped battery has a maximum capacity 2 units.

A. Comparison

In this part, we evaluate the performance of the ϵ-greedy and the convergence based algorithms, where they are compared to the optimal performance scenario. In the optimal performance scenario the optimal policy is used from the first time slot. This presents the upper-bound performance. This scenario needs a priori knowledge of the environment, which is not available to the other two algorithms. For the optimal scenario, the value iteration algorithm (VI) [19] is used to find the optimal policy before the simulation.

In this experiment, adaptive ϵ-greedy exploration algorithm is used [3]. In this algorithm, the exploration probability $\epsilon = 1/i$, where i is the time slot number. For the convergence-based algorithm, ε is set to 0.1, and the T_{thr} is set to 0.05.

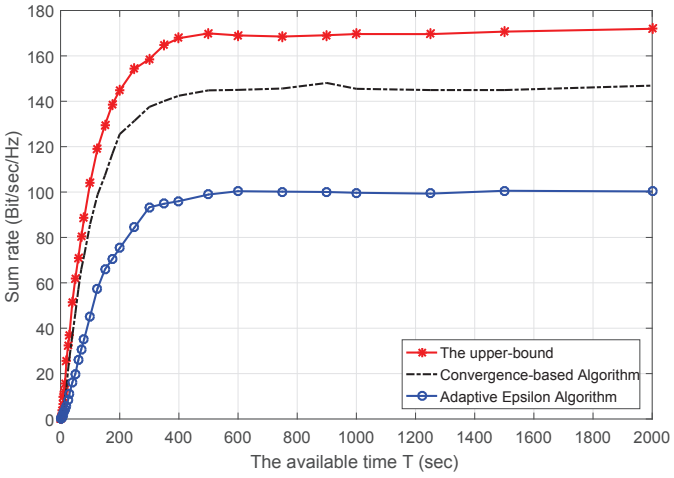

Fig. 3: Sum rates versus the total available time for different approaches

Fig. 3 shows the sum rates versus the available time. It can be noticed that the sum rates of all approaches increase with increasing the available time in the beginning of the session. After that, all of them take near-constant patterns. This can be explained by the effect of the discount factor, which has a value decreases with time. This factor diminishes the effect of the future rewards on the cumulative rewards after a time, which causes the near constant constant performance as the time increases.

As shown, the upper-bound of the sum rates can be achieved by exploiting the optimal policy from the first time slot. It can also be noticed that the convergence-based algorithm outperforms the adaptive ϵ-greedy. This returns to the reason that the convergence-based algorithm starts by evaluating most of the actions in an early stage of the session. This enables the SR to exploit the best resulted policy based on the convergent values in an early time, where the time effect on the discount factor is small, and the exploited policy at these times affect

on the sum rates. On the other hand, the ϵ-greedy explores the actions randomly, and exploits the greedy actions without any criterion that ensures exploiting one of the best available policies in early time. This is reflected on degrading the system performance.

B. Effect of the τ - Convergence-based algorithm

This experiment investigates the effect of the exploration time threshold τ on the performance of the convergence-based algorithm. In this experiment, ζ is set to 0.1.

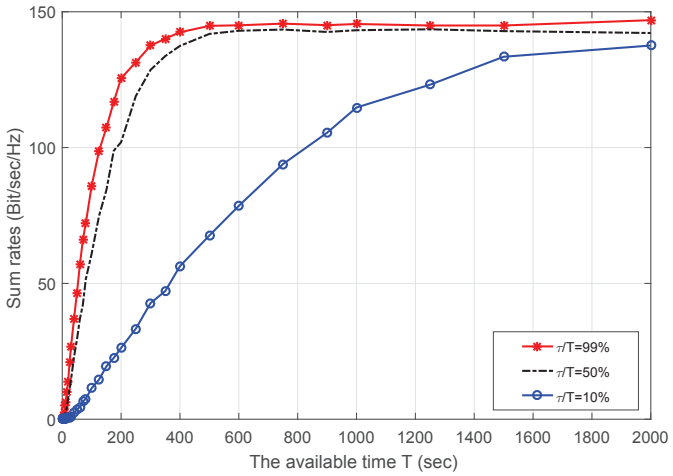

Fig. 4: Sum rates versus the total available time for different values of the τ

Fig. 4 shows the sum rates versus the total available time. The sum rates for different values of τ increase as the available time increases in the beginning. Then, they take a near-constant shape due to the discount factor effect. As shown in this figure, the sum rates increases as τ increases. As mentioned previously, the SR is forced to exploit the current greedy policy once reaching this threshold regardless of evaluating all available policies. So, as the value of this threshold increases, the opportunity of getting the optimal or near optimal policy increases, which contributes in increasing the sum rates that can be achieved.

C. Effect of the ζ - Convergence-based algorithm

In this experiment, the effect of ζ on the performance of the convergence-based algorithm is studied. The value of τ is set to 0.3.

Fig. 5 shows the influence of the available time on the sum rates at different values of ζ. As shown, for all value of ζ, increasing the available time increases the sum rates up to a point, and then it takes a near-constant pattern due the time effect on the discount factor. This figure also shows that the best performance is achieved when ζ has a value of 0.5, and the performance decreases as the convergence error decreases. This returns to the reason that decreasing the convergence error increases the required time to achieve that error. This slows down the exploration process without achieving significant difference after a certain value of ζ, and then, delays the exploitation of the best evaluated policy, which is reflected on the sum rates by reduction.

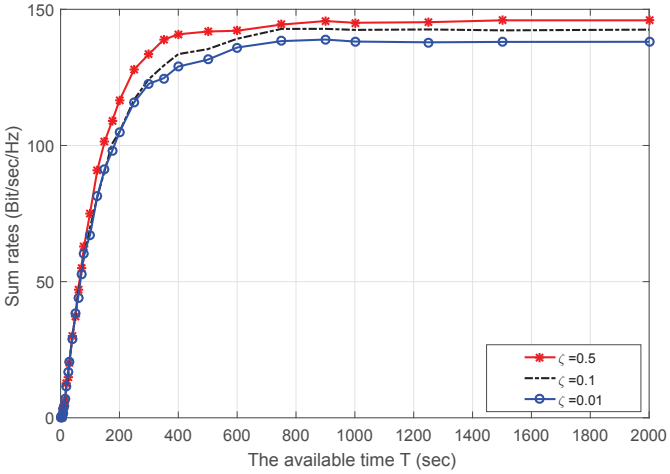

Fig. 5: Sum rates versus the total available time for different values of the ζ

D. Effect of the ϵ - ϵ-greedy algorithm

This part discusses the effect of the ϵ on the system performance for the ϵ-greedy algorithm.

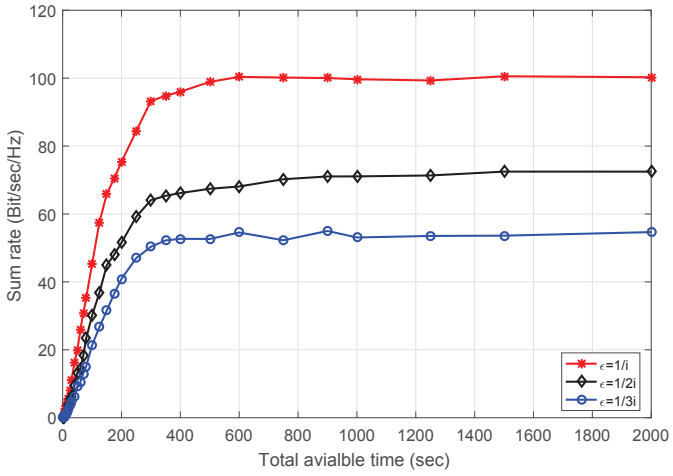

Fig. 6: Sum rates versus the total available time for different values of the ϵ

Fig. 6 compares the adaptive ϵ-greedy algorithm with three scenarios ($\epsilon = 1/i$, $\epsilon = 1/2i$, and $\epsilon = 1/3i$), where i is the time slot number. This figure shows the increase in sum rates with experience (i.e. with increasing the total available time) in the beginning for all scenarios. And then, they take a near-constant shape due to the effect of the time on the discount factor. It can be noticed that the $\epsilon = 1/i$ scenario outperforms the other two scenarios, since it explores more in the beginning, which gives this scenario more opportunity to find the optimal policy earlier. This is also correct when comparing the other two scenarios. In general, it can be concluded that increasing the ϵ increases the sum rates, since

it increases the probability of finding the optimal policy, and exploiting it earlier.

VI. Conclusions

In this paper, a more realistic energy harvesting communication system was investigated. This system does not have a prior knowledge about the environment. The source is equipped with an infinite data buffer to carry data packets and finite battery to store the harvested energy. We formulated the problem of optimizing transmission power as a reinforcement learning problem. Two different exploration algorithms were used, which are the convergence-based and ϵ-greedy algorithms. It was noticed that the convergence-based algorithm outperforms the other one. Finally, we discussed the effects of the parameters of each algorithm on the system performance. As a future work, these two algorithms can be compared with other algorithms and this work can be extended to consider the case of having infinite number of states.

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 Journal on Selected Areas in Communications*, vol. 33, no. 3, pp. 360–381, Mar. 2015.

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

[3] A. Ortiz, H. Al-Shatri, X. Li, T. Weber, and A. Klein, "Reinforcement learning for energy harvesting point-to-point communications," in *Proc. of the IEEE International Conference on Communications (ICC 2016), Kuala Lumpur, Malaysia*, May 2016, pp. 1–6.

[4] T. Mannucci, E.-J. van Kampen, C. de Visser, and Q. Chu, "Safe exploration algorithms for reinforcement learning controllers," *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, no.99, pp. 1–13, 2017.

[5] A. Ortiz, H. Al-Shatri, X. Li, T. Weber, and A. Klein, "Reinforcement learning for energy harvesting decode-and-forward two-hop communications," *IEEE Transactions on Green Communications and Networking*, vol. 1, no.3, pp. 309–319, Sep. 2017.

[6] R. Sutton and A. Barto, *Reinforcement learning: An introduction*. Cambridge, MA, MIT Press, 1998.

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

[8] M. Emre, G. Gür, S. Bayhan, and F. Alagöz, "Cooperativeq: Energy-efficient channel access based on cooperative reinforcement learning," in *Proc. of the IEEE International Conference on Communication Workshop (ICCW), London, UK*, June 2015, pp. 2799–2805.

[9] Z. Xia and D. Zhao, "Online reinforcement learning by bayesian inference," in *Proc. of the International Joint Conference on Neural Networks (IJCNN), Killarney, Ireland*, July 2015, pp. 1–6.

[10] A. D. Tijsma, M. M. Drugan, and M. A. Wiering, "Comparing exploration strategies for q-learning in random stochastic mazes," in *Proc. of the IEEE Symposium Series on Computational Intelligence (SSCI), Athens, Greece*, Dec. 2016, pp. 1–8.

[11] A. Ortiz, H. Al-Shatri, X. Li, T. Weber, and A. Klein, "Reinforcement learning for energy harvesting point-to-point communications," in *Proc. of the IEEE International Conference on Communications (ICC), Kuala Lumpur, Malaysia*, May 2016, pp. 1–6.

[12] C. Szepesvári, *Algorithms for reinforcement learning*. Morgan & Claypool Publishers, 2010.

[13] A. Masadeh, Z. Wang, and A. E. Kamal, "Convergence-based exploration algorithm for reinforcement learning," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing, Calgary, Alberta, Canada*, submitted October 2017.

[14] J. van Ast and R. Babuska, "Dynamic exploration in q (λ)-learning," in *Proc. of the IEEE International Joint Conference on Neural Network Proceedings, Vancouver, BC, Canada*, July 2006, pp. 41–46.

[15] M. Pieters and M. A. Wiering, "Q-learning with experience replay in a dynamic environment," in *Proc. of the IEEE Symposium Series on Computational Intelligence (SSCI), Athens, Greece*, Dec. 2016, pp. 1–8.

[16] V. Heidrich-Meisner, M. Lauer, C. Igel, and M. A. Riedmiller, "Reinforcement learning in a nutshell." in *ESANN*, 2007, pp. 277–288.

[17] Y. Xu, W. Zhang, W. Liu, and F. Ferrese, "Multiagent-based reinforcement learning for optimal reactive power dispatch," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 1742–1751, Dec. 2012.

[18] R. Vullers, R. Schaijk, H. Visser, J. Penders, and C. Hoof, "Energy harvesting for autonomous wireless sensor networks," *IEEE Solid-State Circuits Magazine*, vol. 2, no. 2, pp. 29–38, Spring 2010.

[19] T. Wang, C. Jiang, and Y. Ren, "Access points selection in super wifi network powered by solar energy harvesting," in *Proc. of the IEEE Wireless Communications and Networking Conference (WCNC 2016), Doha, Qatar*, Apr. 2016, pp. 1–5.