ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-7605-5/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICASSP39728.2021.9415110

AN ACTOR-CRITIC REINFORCEMENT LEARNING APPROACH TO MINIMUM AGE OF INFORMATION SCHEDULING IN ENERGY HARVESTING NETWORKS

Shiyang Leng

The Pennsylvania State University
Electrical Engineering Department

Aylin Yener

The Ohio State University
Electrical and Computer Engineering Department

ABSTRACT

We study age of information (AoI) minimization in a network consisting of energy harvesting transmitters that are scheduled to send status updates to their intended receivers. We consider the user scheduling problem over a communication session. To solve online user scheduling with causal knowledge of the system state, we formulate an infinite-state Markov decision problem and adopt model-free on-policy deep reinforcement learning (DRL), where the actor-critic algorithm with deep neural network function approximation is implemented. Comparable AoI to the offline optimal is demonstrated, verifying the efficacy of learning for AoI-focused scheduling and resource allocation problems in wireless networks.

***Index Terms*—** Age of information, energy harvesting, user scheduling, actor-critic deep reinforcement learning.

1. INTRODUCTION

Timely information exchange is crucial for many of forthcoming wireless networking applications, including vehicular networks, unmanned aerial vehicle networks, and IoT networks [1]. Maintaining information freshness in such networks brings about the need for a new network design metric. *Age of information* (AoI) [2, 3] quantifies the time elapsed since the generation of the latest successfully received update. Distinct from metrics of delay or latency, AoI captures the timeliness of information from a receiver's perspective. Reference [3] has analyzed AoI from a queueing theoretic perspective, and characterized AoI for single-source M/M/1, M/D/1, and D/M/1 queues with first-come-first-served (FCFS) service, revealing that AoI minimization offers different insights than delay minimization.

User scheduling in wireless networks is a classical resource allocation problem whose history spans decades, often with throughput as the metric, see for example [4, 5]. Recent references have considered user scheduling for minimum AoI. In [6], the multi-source scheduling problem is identified as NP-hard as an integer linear program, and a suboptimal algorithm is proposed to reduce complexity. In energy harvesting communication networks, where communication is powered by intermittently acquired energy, energy availability has to be explicitly taken into account to ensure the information freshness. Consequently, AoI-optimal update policies under energy harvesting constraints have generated significant recent interest. In the class of renewal policies, the optimal policy is proved to have an energy-dependent multi-threshold structure [7–10].

Most works on AoI-focused energy harvesting communications study optimal update policies and their properties for simple network structures, e.g., point-to-point transmission, which are amenable to model-based analytical approaches. In this paper, we consider a more elaborate model consisting of multiple users capable of energy harvesting to send status updates, which is unlikely to admit a simple solution, but could benefit from learning-based approaches. We develop the online user scheduling policy based on the current and the past observations, that is, only causal information of the system state is available. For our model-free system with continuous-valued states, we address the online user scheduling leveraging DRL. An actor-critic algorithm is utilized, which is an on-policy algorithm and does not require large memory for experience replay in contrast to DQN adopted in our earlier work [11]. We observe experimentally that DRL achieves near-optimal AoI performance with a significant reduction in runtime as compared to optimization solvers.

2. SYSTEM MODEL

We consider a system consisting of K users and their intended receivers. Each user wants to send status updates, e.g., of a physical process, and would like to keep the information fresh at its intended receiver, as shown in Fig. 1. The transmitter-receiver pairs are fixed throughout the session. Multiple transmitters may have the same intended receiver. The user index is denoted by $k \in \{1, 2, \ldots, K\}$. The users harvest energy from ambient energy sources and transmit update packets consuming the harvested energy. The battery capacity at each user is assumed to be sufficiently large. Time is slotted and the duration of each slot is normalized 1 second for simplicity. The jth slot indicates the time interval $[t_{j-1}, t_j)$, where $j = 1, 2, \ldots$ and $t_0 = 0$. The channel between each

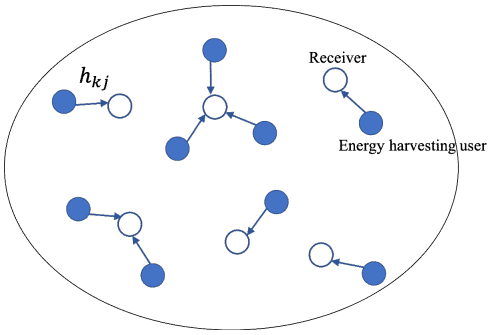

Fig. 1. System model. h_{kj} denotes the channel gain of user k in time slot j.

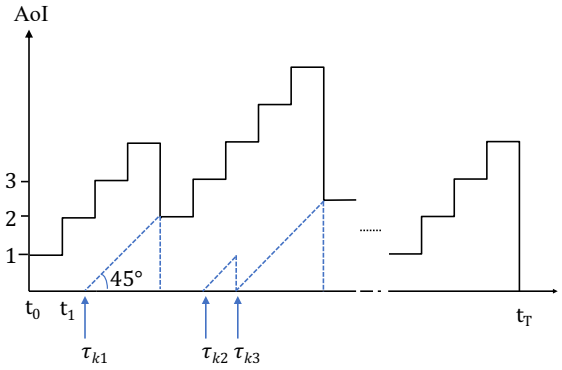

Fig. 2. A sample path of AoI. Dashed lines indicate the packet waiting time at the user and the solid stair-shaped lines indicate the AoI counted discretely at the end of each slot by the receiver. The first and the third packet are delivered. The second update is replaced by the third one.

user and its receiver is assumed to be flat-fading. The path loss and Rayleigh multipath fading are taken into account for the channel gain, which is denoted by h_{kj} for user k in slot j. At most one user is scheduled to transmit in each slot. Each user either transmits to its receiver *or* harvests energy in each slot so that idle users harvest and accumulate energy. Let p_{kj} denote the transmission power and e_{kj} denote the energy that user k can harvest in slot j.

Each user sends update packets that are generated by itself or received from an external source. Either scenario is referred to as packet arrivals in the paper. The timestamp for the arrival of the uth update at user k is denoted by τ_{ku}, for $u = 1, 2, \ldots, U_k$, where U_k is the total number of updates in T slots. The new update packet replaces the old one that has not been sent out. Thus, only the newest packet is buffered at each user for the sake of information freshness. We assume that the size of the update packet is uniform and small, for which the transmission takes one slot. An update is delivered successfully by user k if the received signal-to-noise ratio (SNR) is larger than a target SNR γ_k^*, that is,

$$\frac{p_{kj}h_{kj}}{\sigma^2} \geq \gamma_k^*, \tag{1}$$

where σ^2 is the noise power.

We adopt a linear AoI model [2, 3]. AoI is defined as the time elapsed since the most recently received update is generated. Let a_{kj} denote the AoI for the user k at t_j, which indicates the age of the received packets at the end of slot j. At each slot, the scheduled user is enabled to transmit an update packet. If the delivery is successful, i.e., the received SNR is above the target, the age drops to $t_j - \tau_{ku_j}$ for the delivery of packet u_j, where u_j is the newest packet by t_{j-1}. Otherwise, the age grows by 1, as shown in Fig. 2. AoI evolves as follows for all k, j.

$$a_{kj} = \begin{cases} t_j - \tau_{ku_j}, & \text{if user } k \text{ delivers } u_j \text{ at } t_j \text{ successfully,} \\ a_{k,j-1} + 1, & \text{otherwise,} \end{cases} \tag{2}$$

where a_{k0} is the AoI at t_0 for user k. Let $y_{kj} \in \{0, 1\}$ denote the update scheduling variable, where $y_{kj} = 1$ indicates user k is scheduled to send an update in slot j and $y_{kj} = 0$ indicates it is idle and harvesting energy.

3. ONLINE AOI MINIMIZATION FORMULATION

In the online setting, we consider user scheduling for AoI minimization based on the causal knowledge of the system state information in a centralized manner. The user scheduling decision is made at each slot with the past and the current states available. We aim to derive an online policy that sequentially schedules status updates over time to minimize the long-term average AoI of the system. The Markov decision process (MDP) is defined by the following components.

State: The system state at the beginning of each time slot, denoted by $S_j \in \mathcal{S}$, consists of the AoI, the packet waiting time, the required energy, and the available energy for all users, i.e., $S_j = (\mathbf{a}_{j-1}, \mathbf{w}_j, \mathbf{q}_j, \mathbf{E}_{j-1})$, where $\mathbf{a}_{j-1}$ and $\mathbf{E}_{j-1}$ are the vectors with entries $a_{k,j-1}$ and $E_{k,j-1}$ for $k = 1, 2, \ldots, K$ given in (2) and (3), respectively.

$$E_{kj} = E_{k,j-1} + e_{kj}(1 - y_{kj}) - p_{kj}. \tag{3}$$

where $E_{k,0}$ is the initial energy of user k. $\mathbf{w}_j$ defines the vector of packet waiting times at the beginning of slot j, whose entries are

$$w_{kj} = \begin{cases} t_{j-1} - \tau_{ku_j}, & \text{if a packet is at user } k \text{ at } t_{j-1}, \\ -1, & \text{if no packet is at user } k \text{ at } t_{j-1}, \end{cases} \tag{4}$$

for $k = 1, 2, \ldots, K$. Vector $\mathbf{q}_j$ denotes the required energy by all users for successful updates at slot j, i.e., $q_{kj} = \frac{\gamma_k^* \sigma^2}{h_{kj}}$. Note that the state space $\mathcal{S}$ is infinite (the states are continuous-valued).

Action: A_j in each slot is the index of the scheduled user, i.e., $A_j \in \mathcal{A} = \{0, 1, 2, \ldots, K\}$; $A_j = 0$ implies no one is scheduled as we consider at most one user is scheduled per slot.

Reward: The immediate reward is the negative of the user-averaged AoI of the system, since we aim to minimize AoI. More specifically, given state $S = (\mathbf{a}, \mathbf{w}, \mathbf{q}, \mathbf{E}) \in \mathcal{S}$, if action $A \in \mathcal{A}$ is taken and the next state is $S' = (\mathbf{a}', \mathbf{w}', \mathbf{q}', \mathbf{E}') \in \mathcal{S}$, the immediate reward is defined as

$$r(S, A) = -\frac{1}{K} \sum_{k=1}^{K} a'_k. \tag{5}$$

Transition Probability: The transition probability of reaching state S' from state S by taking action A, denoted by $\mathbb{P}(S'|S, A)$, defines the dynamics of the system, where the transition depends only on S but not the history of the earlier states. Note that when $A_j = k$, user k checks if an update packet is available at t_{j-1} and determines a packet availability variable $v_{kj} \in \{0, 1\}$, based on the constraints that the packet arrival timestamp needs to be smaller than t_{j-1} and each packet can be either sent for only once or dropped. $v_{kj} = 1$ indicates there is an update to send at slot j and $v_{kj} = 0$ otherwise. Then, the actual scheduling variable $y_{kj} \in \{0, 1\}$ is obtained by further checking the energy causality constraint and the required energy constraint. Therefore, we have

$$y_{kj} = \begin{cases} v_{kj} \mathbf{1}_{\hat{E}_{kj} \geq 0}, & \text{if } A_j = k \text{ and } q_{kj} \leq p_{\max}, \\ 0, & \text{otherwise}, \end{cases} \tag{6}$$

$$p_{kj} = y_{kj} q_{kj}, \tag{7}$$

where $p_{\max}$ is the maximum transmit power, $\hat{E}_{kj} = E_{k,j-1} - q_{kj}$ and $\mathbf{1}_\alpha$ denotes the indicator function of α, that $\mathbf{1}_\alpha = 1$ if α is true and $\mathbf{1}_\alpha = 0$ otherwise. Hence, the AoI can be obtained by

$$a_{kj} = y_{kj}(t_j - \tau_{ku_j}) + (1 - y_{kj})(a_{k,j-1} + 1). \tag{8}$$

Taking action A_j on the system results in going through the above steps so that the next state is determined.

The goal is to maximize the cumulative reward in the long run. Here, we consider the sum of the discounted rewards from a starting slot onward, i.e., $G_j = \sum_{k=j}^{\infty} \beta^{k-j} r(S_k, A_k)$, where $\beta \in (0, 1)$ is the discount rate. For any given state, the policy specifies the action, i.e., the mapping from the state space to the action space, denoted by $\pi : \mathcal{S} \to \mathcal{A}$. The value function is the measure of "how good" to be in a state or to perform an action in a state under a given policy. Mathematically, the state-value function is the expected return given an initial state, which is defined as $V_\pi(S) = \mathbb{E}_\pi[G_j | S_j = S]$. The objective is to find the optimal policy π^* that enables the system to act in the way that maximizes the expected discounted return, i.e.,

$$\pi^* = \underset{\pi}{\operatorname{argmax}}\, V_\pi(S), \ \forall S \in \mathcal{S}. \tag{9}$$

4. ACTOR-CRITIC DEEP REINFORCEMENT LEARNING ALGORITHM

We consider the general case that *does not assume any statistics* of the random processes of the energy harvesting nor the packet arrivals. Without an explicit model of the system dynamics, a reinforcement learning problem with the need for model-free methods naturally arises.

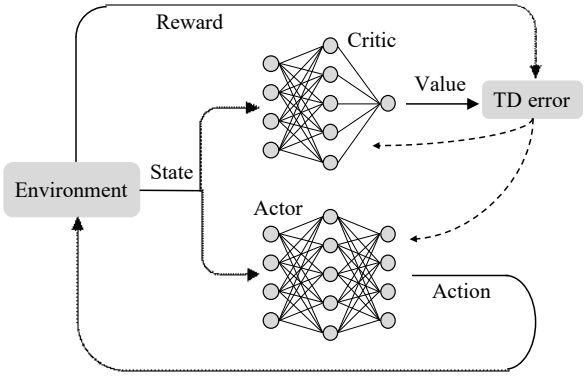

Fig. 3. The schematics of the actor-critic deep reinforcement learning algorithm. The dashed lines indicate the parameter update of the actor and the critic network.

Here, we focus on the advantage actor-critic (A2C) algorithm [12, 13]. As shown in Fig. 3, the actor is a network parameterized by $\boldsymbol{\theta}_a$, which consists of an input layer of $4K$ neurons and a softmax output layer with hidden layers in between. The actor network maps the system state observation S_j to the action probability distribution $\pi(A_j|S_j; \boldsymbol{\theta}_a)$, and schedules an user based on the estimated probabilities at each slot. The critic is a second network that approximates the state-value function with parameter $\boldsymbol{\theta}_c$. Its input layer has the same number of neurons for the $4K$-dimensional state input, and the estimated state value is given by the output layer with one neuron. In the training stage, at each step, the agent first interacts with the environment for $j_{\max}$ slots following the current policy given by the actor, which generates $j_{\max}$ states, actions, and rewards from the current slot looking ahead. Then, the critic evaluates the policy by the $j_{\max}$-step TD error of the value function:

$$\delta_j = G_{j:j+j_{\max}} - V(S_j; \boldsymbol{\theta}_c). \tag{10}$$

Specifically, $G_{j:j+j_{\max}}$ is the sum of the discounted rewards for $j_{\max}$ steps and the estimated value for the future steps, which is given by

$$G_{j:j+j_{\max}} = \sum_{k=j}^{j+j_{\max}-1} \beta^{k-j} r(S_k, A_k) + \beta^{j_{\max}} V(S_{j+j_{\max}}; \boldsymbol{\theta}_c). \tag{11}$$

The parameters of the actor and the critic networks are updated in the direction of maximizing the expected return and

Algorithm 1 The Training Procedure of the Actor-Critic Deep Reinforcement Learning Agent for User scheduling

Set episode number N, episode length T, $j_{\max}$, and learning rate η_a, η_c. Initialize $\boldsymbol{\theta}_a$ and $\boldsymbol{\theta}_c$.
for episode $n = 1, \ldots, N$ **do**
  Reset the environment and initialize state S_1
  for step $j = 1, \ldots, T$ **do**
    for step $i = j, \ldots, j + j_{\max} - 1$ **do**
      Generate action A_i by policy $\pi(\cdot|S_i; \boldsymbol{\theta}_a)$;
      Take action A_i, observe next state S_{i+1} according to (6)-(8), and obtain reward $r(A_i, S_i)$ by (5);
      Compute value $V(S_i; \boldsymbol{\theta}_c)$ of the critic network.
    end for
    Calculate return $G_{j:j_{\max}}$ in (11) based on the experience: $\{S_i, A_i, r(A_i, S_i), S_{i+1}\}_{i=j}^{j+j_{\max}-1}$.
    Compute TD error: $\delta_j = G_{j:j_{\max}} - V(S_j; \boldsymbol{\theta}_c)$.
    Calculate $d\boldsymbol{\theta}_a$ and $d\boldsymbol{\theta}_c$ by (12) and (13).
    $\boldsymbol{\theta}_a \leftarrow \boldsymbol{\theta}_a + \eta_a \, d\boldsymbol{\theta}_a$.
    $\boldsymbol{\theta}_c \leftarrow \boldsymbol{\theta}_c + \eta_c \, d\boldsymbol{\theta}_c$.
  end for
end for

minimizing the TD error, respectively, where the gradients are given by

$$d\boldsymbol{\theta}_a = \sum_{j=1}^{j_{\max}} \delta_j \nabla_{\boldsymbol{\theta}_a} \ln \pi(A_j|S_j; \boldsymbol{\theta}_a), \tag{12}$$

$$d\boldsymbol{\theta}_c = \sum_{j=1}^{j_{\max}} \delta_j \nabla_{\boldsymbol{\theta}_c} V(S_j; \boldsymbol{\theta}_c), \tag{13}$$

The pseudocode for the training procedure is summarized in Algorithm 1. With a well-trained actor-critic agent, the user scheduling action A_j is determined by the actor alone based on the input state S_j, and performing the action on the system gives the next state.

5. RESULTS

In the simulations, each user and its associated receiver are located randomly and separated by a uniformly distributed distance in $[10, 200]$ meters. The channel gain takes into account the path loss and the Rayleigh fading. We set $\sigma^2 = -71$ dBm, $p_{\max} = 0.1$ watt, $a_{k0} = 1$, and $e_{k0} = 0.01$ joule for all k. e_{kj} is an exponential random variable with mean 1 mJ and is i.i.d. for all users and slots. Each user switches between exponential energy-harvesting period and non-energy-harvesting period, with mean 5 and 2 slots, respectively. The inter-arrival time between successive packets is exponential with mean μ. The actor-critic network is trained by the simulated data. We set $T = 50$, $j_{\max} = 50$, $\eta_a = 0.001$, $\eta_c = 0.005$ and $\beta = 0.99$. The second layer of the actor/critic network consists of 64 neurons and the ReLU activation function is used.

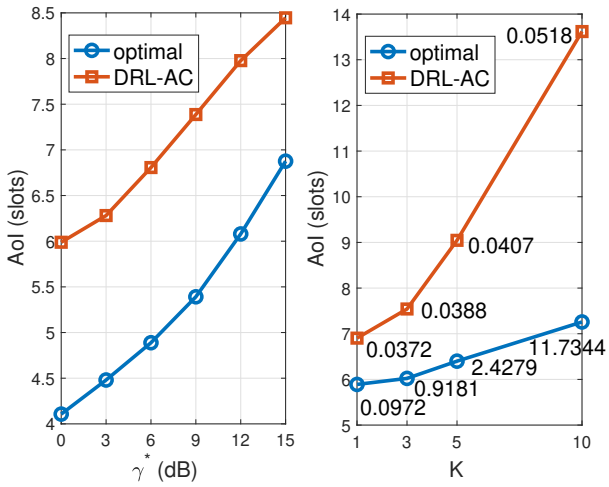

Fig. 4. (a) AoI vs. γ^* for $K = 3$ and $\mu = 2$ slots. (b) AoI vs. K for $\mu = 4$ slots and $\gamma^* = 3$ dB

In Fig. 4 (a), varying the update SNR threshold, we show the average AoI obtained by DRL actor-critic algorithm, which is comparable to the results of the offline mixed integer linear optimization problem that is solved by CPLEX solver. Fig. 4 (b) illustrates the AoI for different number of users, K. As expected, the learning algorithm is more likely to achieve the near-optimal performance for the system with a small number of users, due to the system state space is larger as the number of users grows, which challenges learning approaches. In particular, for $K = 1$ and $K = 3$, we omit the hidden layer of the actor network to simplify the structure of the network and shorten the training process. However, we also note that for larger networks, the learning-based approaches offer feasibility of near-optimal policies. As evidence, we list the average runtime (seconds) for an episode of 50 slots in Fig. 4 (b). The average computation time for the CPLEX solver increases significantly with K since the complexity increases exponentially with the size of the optimization problem. On the other hand, the average testing time by the actor-critic DRL agent does not vary much as the computation task is dominated by the training process, which takes time of the order of 10000 seconds.

6. CONCLUSION

In this paper, we have considered user scheduling, i.e., transmission times and powers, for AoI minimization in energy harvesting networks. For the online setting, we have proposed an actor-critic DRL algorithm for sequential user scheduling based on the MDP formulation. We have shown that the learning algorithms can be a viable alternative to optimization relaxation or approximation methods to find near-optimal solutions to computationally hard problems.

7. REFERENCES

[1] Mohamed A Abd-Elmagid, Nikolaos Pappas, and Harpreet S Dhillon, "On the role of age of information in the internet of things," *IEEE Communications Magazine*, vol. 57, no. 12, pp. 72–77, 2019.

[2] Sanjit Kaul, Marco Gruteser, Vinuth Rai, and John Kenney, "Minimizing age of information in vehicular networks," in *Proceedings of 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, 2011, pp. 350–358.

[3] Sanjit Kaul, Roy Yates, and Marco Gruteser, "Real-time status: How often should one update?," in *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)*, 2012, pp. 2731–2735.

[4] Barry M Leiner, Donald L Nielson, and Fouad A Tobagi, "Issues in packet radio network design," *Proceedings of the IEEE*, vol. 75, no. 1, pp. 6–20, 1987.

[5] Leonard Kleinrock and John Silvester, "Spatial reuse in multihop packet radio networks," *Proceedings of the IEEE*, vol. 75, no. 1, pp. 156–167, 1987.

[6] Qing He, Di Yuan, and Anthony Ephremides, "Optimal link scheduling for age minimization in wireless systems," *IEEE Transactions on Information Theory*, vol. 64, no. 7, pp. 5381–5394, 2018.

[7] Xianwen Wu, Jing Yang, and Jingxian Wu, "Optimal status update for age of information minimization with an energy harvesting source," *IEEE Transactions on Green Communications and Networking*, vol. 2, no. 1, pp. 193–204, 2018.

[8] Ahmed Arafa, Jing Yang, Sennur Ulukus, and H Vincent Poor, "Age-minimal transmission for energy harvesting sensors with finite batteries: Online policies," *IEEE Transactions on Information Theory*, vol. 66, no. 1, pp. 534–556, 2020.

[9] Shiyang Leng and Aylin Yener, "Impact of imperfect spectrum sensing on age of information in energy harvesting cognitive radios," in *Proceedings of IEEE International Conference on Communications (ICC)*, 2019.

[10] Shiyang Leng and Aylin Yener, "Age of information minimization for an energy harvesting cognitive radio," *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 2, pp. 427–439, 2019.

[11] Shiyang Leng and Aylin Yener, "Age of information minimization for wireless ad hoc networks: A deep reinforcement learning approach," in *IEEE Global Communications Conference (GLOBECOM)*, 2019.

[12] Richard S Sutton and Andrew G Barto, *Reinforcement learning: An introduction*, MIT press, 2018.

[13] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *Proceedings of the International conference on machine learning*, 2016, pp. 1928–1937.