Deep Reinforcement Learning for Dynamic Multichannel Access

Conference Paper · January 2017

CITATIONS
3

READS
55

4 authors, including:

Pedro Henrique Gomes
University of Southern California
28 PUBLICATIONS **105** CITATIONS
SEE PROFILE

Bhaskar Krishnamachari
University of Southern California
380 PUBLICATIONS **16,949** CITATIONS
SEE PROFILE

Deep Reinforcement Learning for Dynamic Multichannel Access

Shangxing Wang*, Hanpeng Liu†, Pedro Henrique Gomes* and Bhaskar Krishnamachari*
Ming Hsieh Department of Electrical Engineering, University of Southern California, Los Angeles, USA
Email: *{shangxiw, pdasilva, bkrishna}@usc.edu
Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China
Email: †lhp13@mails.tsinghua.edu.cn

***Abstract*—We consider the problem of dynamic multichannel access in a Wireless Sensor Network (WSN) containing N correlated channels, where the states of these channels follow a joint Markov model. A user at each time slot selects a channel to transmit a packet and receives a reward based on the success or failure of the transmission, which is dictated by the state of the selected channel. The objective is to find a policy that maximizes the expected long-term reward. The problem can be formulated as a partially observable Markov decision process (POMDP), which is PSPACE-hard and intractable. As a solution, we apply the concept of online learning and implement a Deep Q-Network (DQN) that can deal with large state space without any prior knowledge of the system dynamics. We compare the performance of DQN with a myopic policy and a Whittle Index-based heuristic through simulations and show that DQN can achieve near-optimal performance. We also evaluate the performance of DQN on traces obtained from a real indoor WSN deployment. We show that DQN has the capability to learn a good policy in complex real scenarios, which do not necessarily show Markovian dynamics.**

I. Introduction

Inspired by the seminal work [1], dynamic spectrum access in a Wireless Sensor Network (WSN) is believed to be one of the keys to improving the spectrum utilization and meet the increasing need for larger deployments. The arrival of cognitive radio has allowed second users to search and use idle channels that are not being used by their primary users (PU). Although there are many existing works that focus on the algorithm design and implementation in this field, nearly all of them assume a simple independent-channel (or PU activity) model. The *de facto* physical layer employed in most of the WSNs (namely, IEEE 802.15.4) uses Industrial, Scientific, and Medical (ISM) bands, such as the globally available 2.4 GHz or 868/900 MHz. ISM bands are shared by various wireless technologies (e.g. Wi-Fi, Bluetooth, RFID), as well as industrial/scientific equipment and appliances (e.g. micro-wave ovens). Thus, external interference can cause the channels in WSNs to be highly correlated, and the design of new algorithms and schemes in dynamic multichannel access is required to resolve this challenge.

In this paper, we consider a wireless network with N correlated channels, and each channel has two possible states: *good* or *bad*. There is a single user (wireless node) that selects one channel at each time slot to transmit a packet. If the selected channel is in the *good* state, the transmission is successful; otherwise, there is a transmission failure. The goal is to obtain as many successful transmissions as possible over time. We use a Markov chain with 2^N states to describe the joint states of N channels. Since the user is only able to sense the selected channel and no full observation of the system is available, the problem can be formulated as a partially observable Markov decision process (POMDP), which is PSPACE-hard and has an exponential computation complexity [2].

We investigate the use of Deep Reinforcement Learning from the field of machine learning as a way to overcome the prohibitive computational requirements due to the large state space. We implement a Deep Q-Network (DQN) [3] that can find a channel access policy through online learning. This DQN approach is able to deal with large systems, as well as find a good policy directly from historical observations without any requirement to know the system dynamics *a-priori*. We show through simulations that DQN can achieve a near-optimal performance. In addition, we also evaluate DQN with real data traces collected from an indoor WSN deployment, and DQN is able to find a good policy even though the Markovian property may not hold in real scenarios.

The rest of the paper is organized as follows. In Sec. II, we provide the problem formulation of the dynamic multichannel access problem when channels are potentially correlated. In Sec. III, a Myopic and a Whittle Index-based heuristic policies are presented to solve the problem. In Sec. IV, we present the DQN framework to find the policy through online learning. We present the experiment and evaluation results in Sec. V and conclude our work in Sec. VI.

II. Problem Formulation

Consider a dynamic multichannel access problem where there is a single user that dynamically chooses one out of N channels to transmit packets. Each channel can be in one of two different states: *good* (1) or *bad* (0). Since channels may be correlated, the whole system can be described as a 2^N-state Markov chain. At the beginning of each time slot, a user selects one channel to sense and transmit one packet. If the channel quality is good, the transmission succeeds

This work was funded in part by NSF through awards number 1248017 and 1423624. Hanpeng Liu was supported by the Institute for Interdisciplinary Information Sciences, Tsinghua University.

and the user receives a positive reward $(+1)$. Otherwise, the transmission fails and the user receives a negative reward (-1). The objective is to design a sensing policy that maximizes the expected long-term reward.

Let the state space of the Markov chain be $\mathcal{S} = \{\mathbf{s}_1, ..., \mathbf{s}_{2^N}\}$. Each state $\mathbf{s}_i$ ($i \in \{1, ..., 2^N\}$) is a length-N vector $[s_{i1}, ..., s_{iN}]$, where s_{ik} is the binary representation of the state of channel k: good (1) or bad (0). The transition matrix of the Markov chain is denoted as $\mathbf{P}$. Since the user can only sense one channel and observe its state at the beginning of each time slot, the full state of the system, i.e., the states of all channels, is not observable. However, the user can infer the system state according to his sensing decisions and observations. Thus, the dynamic multichannel access problem falls into the general model of POMDP. Let $\Omega(t) = [\omega_{\mathbf{s}_1}(t), ..., \omega_{\mathbf{s}_{2^N}}(t)]$ represent the belief vector maintained by the user, where $\omega_{\mathbf{s}_i}(t)$ is the conditional probability that the system is in state $\mathbf{s}_i$ given all previous decisions and observations. Given the sensing action $a(t) \in \{1, ..., N\}$ representing which channel to sense at the beginning of time slot t, the user can observe the state of channel $a(t)$, denoted as $o(t) \in \{0, 1\}$. Then, based on this observation, he can update the belief vector at time slot t, denoted as $\hat{\Omega}(t) = [\hat{\omega}_{\mathbf{s}_1}(t), ..., \hat{\omega}_{\mathbf{s}_{2^N}}(t)]$. The belief of each possible state $\hat{\omega}_{\mathbf{s}_i}(t)$ is updated as follows:

$$\hat{\omega}_{\mathbf{s}_i}(t) = \begin{cases} \frac{\omega_{\mathbf{s_i}}(t)\mathbb{1}(\mathbf{s}_{ik}(t)=1)}{\sum_{i=1}^{2^N} \omega_{\mathbf{s}_i}(t)\mathbb{1}(\mathbf{s}_{ik}(t)=1)} & a(t) = k, o(t) = 1 \\ \frac{\omega_{\mathbf{s}_i}(t)\mathbb{1}(\mathbf{s}_{ik}(t)=0)}{\sum_{i=1}^{2^N} \omega_{\mathbf{s}_i}(t)\mathbb{1}(\mathbf{s}_{ik}(t)=0)} & a(t) = k, o(t) = 0 \end{cases} \quad (1)$$

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

Combining the newly updated belief vector $\hat{\Omega}(t)$ for time slot t with the system transition matrix $\mathbf{P}$, the belief vector for time slot $t+1$ can be updated as:

$$\Omega(t+1) = \hat{\Omega}(t)\mathbf{P} \quad (2)$$

A sensing policy $\pi : \Omega(t) \to a(t)$ is a function that maps the belief vector $\Omega(t)$ to a sensing action $a(t)$ at each time slot t. Given a policy π, the long-term reward considered in this paper is the expected accumulated discounted reward over infinite time horizon, defined as below:

$$\mathbb{E}_\pi[\sum_{t=1}^{\infty} \gamma^{t-1} R_{\pi(\Omega(t))}(t)|\Omega(1)] \quad (3)$$

where $0 \leq \gamma < 1$ is a discounted factor, $\pi(\Omega(t))$ is the action (i.e., which channel to sense) at time t when the current belief vector is $\Omega(t)$, and $R_{\pi(\Omega(t))}(t)$ is the corresponding reward.

If no information about the initial distribution of the system state is available, one can assume the initial belief vector $\Omega(1)$ to be the stationary distribution of the system. Our objective is to find a sensing policy π^* that maximizes the expected accumulated discounted reward over infinite time

$$\pi^* = \arg\max_{\pi} \mathbb{E}_\pi[\sum_{t=1}^{\infty} \gamma^{t-1} R_{\pi(\Omega(t))}(t)|\Omega(1)] \quad (4)$$

As the dynamic multichannel access problem is a POMDP, the optimal sensing policy π^* can be found by converting the POMDP into a MDP with the belief as the state and solving the corresponding MDP instead [2]. In theory, the optimal policy π^* can be obtained by solving via dynamic programming. However, this approach is computationally prohibitive due to the large size of the continuous belief space and the impact of the current action on the future reward.

III. Myopic Policy and Whittle Index

In the domain of dynamic multichannel access, there are many existing works on finding the optimal/near-optimal policy with low computation when the channels independent and system statistics ($\mathbf{P}$) is known. The Myopic Policy and the Whittle Index Policy are two easy-to-implement approaches for this settings.

A. Myopic Policy

A myopic policy only focuses on the immediate reward obtained from an action and ignores its effects in the future. Thus the user always tries to select a channel which gives the maximized expected immediate reward.

The myopic policy is not optimal in general cases. Researchers in [4], [5] have studied its optimality when N channels are independent and statistically identical Gilbert-Elliot channels that follow the same 2-state Markov chain with the transition matrix as $\begin{bmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{bmatrix}$. It is shown the myopic policy is optimal when the channel quality is positively correlated, i.e., $p_{11} \geq p_{01}$. In addition, the myopic policy has a simple robust structure that reduces the channel selection to a simple round-robin procedure.

B. Whittle Index Based Heuristic Policy

When channels are independent, the dynamic multichannel access problem can also be considered as a restless multi-armed bandit problem (RMAB) if each channel is treated as an arm. An index policy assigns a value to each arm based on its current state and chooses the arm with the highest index at each time slot. Similarly, the index policy does not have optimality guarantee in general.

In [6], the Whittle Index is introduced in the case when $\mathbf{P}$ is known and all channels are independent but may follow different 2-state Markov chain models. In this case, the Whittle Index policy can be represented in closed-form, and it has been proved to be optimal when all channels are i.i.d. and positively correlated. In addition, under these conditions, the Whittle Index policy has the same round-robin structure as the myopic policy.

When channels are correlated, the Whittle Index cannot be defined and thus the Whittle Index policy cannot be directly applied to our problem. To leverage its simplicity, we propose a heuristic that ignores the correlations among channels and uses the joint transition matrix $\mathbf{P}$ and Bayes' Rule to compute the 2-state Markov chain for each individual channel. After each channel model is found, we apply the Whittle Index policy accordingly.

In the case of independent channels, the Myopic and the Whittle Index policies are easy to implement and they can

achieve optimality under certain conditions. However, so far to the best of our knowledge there are no easy-to-implement policies applicable to the general case where channels are correlated. Moreover, both policies require the prior knowledge of the system's transition matrix, which is hard to be obtained in practice. Thus, we need to come up with a new approach that copes with these challenges.

IV. Deep Reinforcement Learning Framework

When channels are correlated and system dynamics is unknown, there are two ways to approach the dynamic multichannel access problem: (i) estimate the transition matrix $\mathbf{P}$ from observations and then apply either the Myopic policy or the Whittle index-based policy; (ii) learn the policy directly through interactions with the system. The first approach may not scale well when the system becomes large, as the size of the transition matrix grows exponentially with the number of channels. The second approach, by incorporating the idea of Reinforcement Learning, does not need to deal with the large transition matrix and can be easily extended to very large and complicated systems.

A. Q-Learning

We focus on Reinforcement Learning paradigm, Q-learning specifically, to incorporate learning in the solution for the dynamic multichannel access problem. The goal of Q-learning is to find an optimal policy, i.e., a sequence of actions that maximizes the long-term expected accumulated discounted reward. Q-learning is a value iteration approach and the essence is to find the Q-value of each state and action pair, where the state $\mathbf{x}$ is a function of observations (and rewards) and the action a is some action that user can take given the state $\mathbf{x}$. The Q-value of a state-action pair $(\mathbf{x}, a)$ from policy π, denoted as $Q^{\pi}(\mathbf{x}, a)$, is defined as the sum of the discounted reward received when taking action a and then following the policy π thereafter. Then the optimal policy π^* is $\pi^*(\mathbf{x}) = \arg\max_a Q^{\pi^*}(\mathbf{x}, a), \forall \mathbf{x}$.

One can use online learning method to find $Q^{\pi^*}(\mathbf{x}, a)$ without any knowledge of the system dynamics. Assume at the beginning of time slot $t+1$, the agent takes an action $a_t \in \{1, ..., N\}$ that maximizes its Q-value of state-action pair $(\mathbf{x}_t, a_t)$ given the state is $\mathbf{x}_t$, and gains a reward r_{t+1}. Then the online update rule of Q-values with learning rate $0 < \alpha < 1$ is given as follows:

$$\begin{aligned} Q(\mathbf{x}_t, a_t) \leftarrow & Q(\mathbf{x}_t, a_t) \\ & + \alpha[r_{t+1} + \gamma \max_{a_{t+1}} Q(\mathbf{x}_{t+1}, a_{t+1}) - Q(\mathbf{x}_t, a_t)] \end{aligned} \quad (5)$$

It has been shown that in the MDP case, if each action is executed in each state an infinite number of times on an infinite run and the learning rate α decays appropriately, the Q values will converge with probability 1 to the optimal Q^{π^*} [7].

In the context of the dynamic multichannel access, the problem can be converted to an MDP when considering the belief space, and Q-learning can be applied consequently. However, this approach is impractical since the belief update is maintained by knowing the system transition matrix $\mathbf{P}$ *a-priori*, which is hardly available in practice. Instead, we apply the Q-learning framework to the original POMDP by directly using the history of actions and observations. We define the state for the Q-learning at time slot t as the combination of the channels that are decided to sense and the corresponding observed conditions of such sensed channels over previous M time slots, i.e., $\mathbf{x}_t = [a_{t-1}, o_{t-1}, ..., a_{t-M}, o_{t-M}]$. Then we can execute the online learning through Eq. (5) to find the sensing policy. Intuitively, the more historical information we consider (i.e., the larger M is), the better Q-learning can learn.

B. Deep Q-Network

Q-learning works well when the problem state and action spaces are small, as a look-up table can be used to execute the update rule in Eq. (5). But this is impossible when the state-action space becomes very large. Even worse, many states are rarely visited, so the corresponding Q-values are seldom updated, which may require a very long time to converge.

Researchers have proposed both linear and non-linear Q-value approximations to overcome these issues. In 2015, DeepMind developed a Deep Q-Network (DQN), which makes use of a deep neural network to approximate the Q-value, and it achieves human-level control in the challenging domain of classic Atari 2600 games [3]. A neural network is a biologically-inspired programming paradigm organized in layers. Each layer is made up of a number of nodes known as neurons. Each neuron takes the weighted linear combination of the outputs from neurons in the previous layer as input and outputs the result from its nonlinear activation function to the next layer. A deep neural network is a neural network that can be considered as a deep graph with many processing layers. A deep neural network is able to learn from low-level observed multi-dimensional data and find its success in areas such as computer vision and natural language processing [8], [9].

DQN combines Q-learning with deep learning, and the Q-function is approximated by a deep neural network that takes the state-action as input and outputs the corresponding Q-value. The basic idea behind DQN is the use of a neural network function approximator with weights θ as a Q-network. The Q-network updates its weights at each iteration i to minimize the loss function $L_i(\theta_i) = \mathbb{E}[(y_i - Q(\mathbf{x}, a; \theta_i))^2]$, where y_i is also derived from the same Q-network with old weights, i.e., $y_i = \mathbb{E}[r + \max_{a'} Q(\mathbf{x}', a'; \theta_{i-1})]$, and $\mathbf{x}'$ is the new state after taking action a given the state $\mathbf{x}$.

In a typical multichannel WSN based on the widely used IEEE 802.15.4-2015 standard [10], nodes have to choose one out of 16 available channels to sense at each time slot. The state space tends to be large especially when considering the potential correlations among channels. Since we directly use the previous actions and observations as the state for the Q-learning, the state space becomes very large. Therefore, a DQN implementation is needed to help to find a tractable policy implementation in the dynamic multichannel access problem.

V. Experiment and Evaluation

In this section, we present the details of our DQN implementation together with its evaluations based on both simulations and real traces.

A. DQN Architecture details

In the experiment, we follow the *Deep Q-learning with Experience Replay Algorithm* [3]. The structure of our DQN is finalized as a five-layer fully connected neural network with each hidden layer containing 50 neurons. The activation functions used for neurons are either all *ReLU* functions or all *tanh* functions. The state of the DQN is defined as the combination of previous actions and observations over previous M steps, and the considered number of historical time slots is the same as the number of channels in the system. We apply the ϵ-greedy policy with ϵ fixed as 0.1 to balance the exploration and exploitation, i.e., with probability 0.1 the agent uniformly selects an action, and with probability 0.9 the agent chooses the action that maximizes the Q value of a given state. When updating the weights θ of the DQN, a minibatch of 32 samples are randomly selected from the replay memory to compute the loss function, then a recently proposed Adam algorithm [11] is used to conduct the stochastic gradient descent to update the weights (details on the hyperparameters are listed in Table I).

B. Performance Evaluation

We compare the DQN primarily with two other policies: the Random Policy and the Whittle Index Based Heuristic Policy. In the Random Policy, at the beginning of each time slot, the user randomly selects one channel with equal probability. In the Whittle Index Policy, the user assumes all channels are independent. Since there is no information about the probability distribution of each channel, the user first observes each channel by sensing it for some time, and then uses Maximum Likelihood Estimation (MLE) to estimate the transition matrix of the 2-state Markov chain. We do not consider the Myopic policy in general, as the transition matrix $\mathbf{P}$ is too large to access. However, in some simulation cases when $\mathbf{P}$ is sparse and easy to access, we implement the myopic policy as a genie and evaluate its performance.

1) Simulations with perfectly correlated scenario: We consider a highly correlated scenario. In a 16-channel system, we assume only two or three channels are independent, and other channels are exactly identical or opposite to one of these independent channels. This is the case when some channels are perfectly correlated, i.e., the correlation coefficient ρ is either 1 or -1.

During the simulation, we arbitrarily set the independent channels to follow the same 2-state Markov chain with $p_{11} \geq p_{01}$. When the correlation coefficient $\rho = 1$, the myopic policy with known $\mathbf{P}$ (sparse and easy to access) is optimal and has a simple round robin structure alternating among independent channels [4], [5]. In the case when $\rho = -1$, though the myopic policy with known $\mathbf{P}$ is not proved optimal, our conjecture is that its performance is near-optimal.

TABLE I: List of DQN Hyperparameters

Hyperparameters	Values
ϵ	0.1
Minibatch size	32
Optimizer	Adam
Activation Function	ReLU or tanh
Learning rate	10^{-5}
Experience replay size	$1,000,000$
γ	0.9

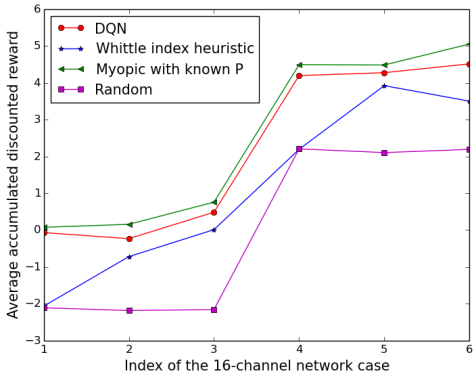

Fig. 1: Average discounted reward for 6 different cases. Each case considers a different set of correlated channels.

In Fig. 1 we present the performance of all four policies: (i) DQN, (ii) Random, (iii) Whittle Index Based Heuristic Policy, and (iv) Myopic policy with known $\mathbf{P}$. In the first three cases (x-axis 1, 2 and 3), the correlation coefficient ρ is fixed as 1 and in the last three cases (x-axis 4, 5 and 6), ρ is fixed as -1. We also vary the set of correlated channels to make cases different. The myopic policy in the first three cases is optimal, and in the last three cases is conjectured to be near-optimal. As it is shown in Fig. 1, the myopic policy, which is implemented based on the full-knowledge of the system, is the best among all six cases and serves as an upper bound. DQN provides a performance very close to the myopic policy without any knowledge of the system dynamics. The Whittle Index policy performs worse than DQN in all cases as it ignores correlations among channels.

In addition, we collect the Q-values predicted from the DQN to show that DQN, indeed, tries to learn and improve its performance. Given a state $\mathbf{x}$, the maximum Q-value over all actions, i.e., $\max_a Q(\mathbf{x}, a)$, represents the estimate of the maximum expected accumulated discounted reward starting from $\mathbf{x}$ over an infinite time horizon. For each simulation case, we fix a set of states that are randomly selected, and then plot the average maximum Q value of all these states as the training is executed. As it is shown in Fig. 2, in all cases, the average maximum Q-value first increases and then becomes stable, which indicates the DQN learns from experience to improve its performance and converges to a good policy.

2) Simulations with real data traces: We use real data traces collected from our indoor testbed Tutornet [1] to train and

[1]More information about the testbed on http://anrg.usc.edu/www/tutornet/

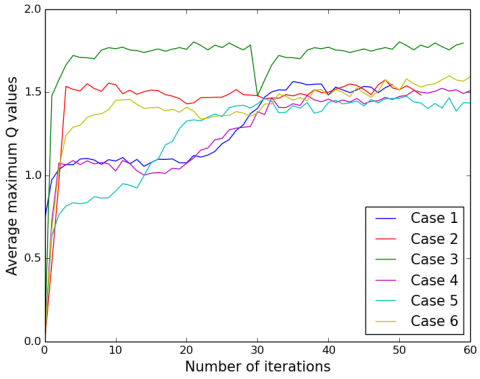

Fig. 2: Average maximum Q-value of a set of states in 6 different simulation cases

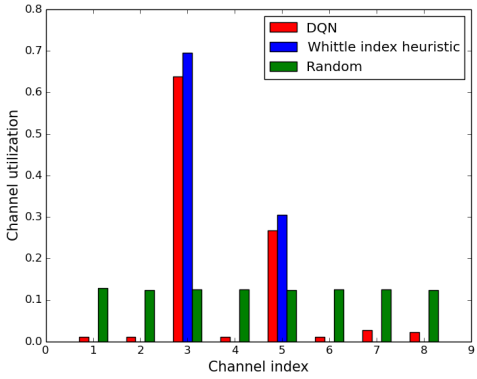

Fig. 3: Channel utilization of 8 channels in the testbed

evaluate the performance of DQN on real systems. The testbed is composed of TelosB nodes with IEEE 802.15.4 radio. We programmed a pair of motes distanced approximately 20 meters to be transmitter/receiver. The transmitter continually transmits one packet on each one of the 16 available channels periodically and the receiver records the successful and failed attempts. Both nodes are synchronized to avoid packet loss due to frequency mismatch and the other motes on the testbed are not in use. The only interference suffered is from surrounding Wi-Fi networks and multi-path fading. There are 8 Wi-Fi access points on the same floor and dozens of people working in the environment, which creates a very dynamic scenario for multichannel access.

The data are collected for around 17 hours. Due to the configuration of Wi-Fi central channels, there are 8 channels whose conditions are significantly better than others. Randomly selecting one channel from these good channels and keeping using it can lead to a good performance. Thus, in order to create a more adverse scenario and test the learning capability of the DQN, we ignore all these good channels and only use the data trace from the rest 8 channels.

We use the same data trace to train the DQN and to compute the MLE of the transition matrices of each channel for the Whittle index based heuristic policy. We compare the performance of the DQN policy, the Whittle index based heuristic policy and the Random policy. The average accumulated discounted reward from each policy is listed in descending order: 0.947 (DQN), 0.767 (Whittle Index) and -2.170 (Random Policy) It can be seen that DQN performs best in the complicated real scenario. We also present the channel utilization of each policy in Fig. 3 to illustrate the difference among them. It shows DQN benefits from using other channels when the two best channels (used by the Whittle Index Heuristic all the time) may not be in good states.

VI. Conclusion

In this paper, we considered the dynamic multichannel access problem in a more general and practical scenario when all channels are correlated. As the problem is POMDP without a tractable solution, we have applied DQN that directly utilizes historical actions and observations to find the access policy via online learning. It has been shown from simulations that DQN can achieve a near-optimal performance without *a-priori* knowing any system dynamics. In addition, when applying real data traces, which may not even have the Markovian property, DQN still perform better than other existing algorithms. There are a couple of open directions suggested by the present work. First, we plan to apply the DQN framework to consider more realistic and complicated scenarios such as multi-hop and simultaneous transmissions in WSN. Second, we intend to study the structure and property of the policy learned from DQN which might enable us to design heuristics that can perform better without the burden of the long learning period.

References

[1] R. Knopp and P. Humblet, "Information capacity and power control in single-cell multiuser communications," in *Proceedings IEEE International Conference on Commun. ICC*. IEEE, 1995.

[2] Q. Zhao, L. Tong, A. Swami, and Y. Chen, "Decentralized cognitive MAC for opportunistic spectrum access in ad hoc networks: A POMDP framework," *IEEE J. Sel. Areas in Commun.*, vol. 25, no. 3, pp. 589–600, apr 2007.

[3] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[4] Q. Zhao, B. Krishnamachari, and K. Liu, "On myopic sensing for multi-channel opportunistic access: structure, optimality, and performance," *IEEE Trans. Wireless Commun.*, vol. 7, no. 12, pp. 5431–5440, dec 2008.

[5] S. H. A. Ahmad, M. Liu, T. Javidi, Q. Zhao, and B. Krishnamachari, "Optimality of myopic sensing in multichannel opportunistic access," *IEEE Trans. Inf. Theory*, vol. 55, no. 9, pp. 4040–4050, sep 2009.

[6] K. Liu and Q. Zhao, "Indexability of restless bandit problems and optimality of whittle index for dynamic multichannel access," *IEEE Trans. Inf. Theory*, vol. 56, no. 11, pp. 5547–5567, nov 2010.

[7] C. Watkins and P. Dayan, "Technical note: Q-learning," *Machine Learning*, vol. 8, no. 3-4, 1992.

[8] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[9] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems 27*. Curran Associates, Inc., 2014, pp. 3104–3112.

[10] "802.15.4-2015 - IEEE Standard for Low-Rate Wireless Personal Area Networks (WPANs)," 2015.

[11] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.