Enabling Opportunistic and Dynamic Spectrum Access Through Learning Techniques

Omar Alsaleh, Pavithra Venkatraman, Bechir Hamdaoui, Alan Fern
School of EECS, Oregon State University
{alsaleh,venkatrp,hamdaoui,afern}@eecs.oregonstate.edu

***Abstract*— The expected shortage in spectrum supply is well understood to be primarily due to the inefficient, static nature of current spectrum allocation policies. In order to address this problem, FCC promotes the so-called Opportunistic Spectrum Access (OSA) to be applied on Cognitive Radio Networks (CRNs). In short, the idea behind OSA is allowing unlicensed users to use unused licensed spectra as long as they do not cause interference to licensed users. In this paper, we present and evaluate learning schemes that allow unlicensed users to locate and use spectrum opportunities effectively, thus improving efficiency of CRNs. We separately consider two models: single and multiple unlicensed user(s). For the latter model, we present two schemes: non-cooperative and cooperative Q-learning. All proposed schemes do not require prior knowledge or prediction models of the environment's dynamics and behaviors, yet can still achieve high performance by learning from interaction with the environment. Using simulations, we show that the proposed schemes achieve good performances in terms of throughput and fairness.**

Keywords: opportunistic/dynamic spectrum access, Q-learning; reinforcement learning, cognitive radio networks.

I. Introduction

Spectrum has been traditionally partitioned by Federal Communications Commission (FCC) into frequency bands and assigned to licensees, also referred to as *primary users* (PUs). PUs have exclusive and flexible access rights as well as are protected against interference when using their assigned bands. The traditional spectrum assignment methods are no longer suitable, as there has been an expected shortage in the spectrum supply due mainly to the inefficient, static nature of these traditional methods, thus calling for new ways that can exploit the available spectrum more effectively. This fact is well supported by measurement-based studies [1, 2], which show that the average occupancy of spectrum over most frequencies is very low. This measurement data confirms the availability of many spectrum opportunities along time, frequency, and space that wireless devices and networks can potentially utilize. Therefore, it is imperative to develop mechanisms that enable effective exploitation of these opportunities.

In order to meet the growing demand of spectrum resources, FCC's long-term vision is to evolve towards more liberal, flexible spectrum allotment policies and usage rights, where spectrum will be managed and controlled dynamically by network entities and end-user devices themselves with little to no involvement of any centralized regulatory bodies. As an initial step towards this direction, FCC promotes the so-called *opportunistic spectrum access* (OSA), which improves spectrum utilization efficiency.

The basic idea of OSA is to allow unlicensed users, also referred to as *secondary users* (SUs), to exploit the unused licensed spectrum on an instant-by-instant basis, but in a manner that limits interference to PUs so as to maintain compatibility with legacy systems. In this paper, a group of two or more SUs who wish to communicate together is referred to as *an agent*[1]. In order to communicate with each other, all SUs in the same group must be tuned to the same spectrum band. And prior to using a licensed band, SUs must first sense the band to assess whether it is vacant, and if it is, then they can switch to and use it as long as no PUs are present. Upon the detection of the return of any PUs to their band, SUs must immediately vacate the band.

Due to its great potentials, OSA has generated significant research interests and works, ranging from protocol design [3, 4] to performance optimization [5, 6], and from market-oriented access strategies [7] to new management and architecture paradigms [8, 9]. Some research efforts have also been given to the development of adaptive approaches that can promote OSA. Most of these proposed approaches require and rely on models that can capture and predict the environment's dynamics and behaviors. However, due mainly to its very unique characteristics, it is too difficult, if not impossible, to construct models that can predict the dynamics of the OSA environment accurately, thus calling for innovative techniques that can achieve good performances by learning directly from interaction with the environment, and without needing models of such environments. Indeed, reinforcement learning (RL) is a foundational idea built on the basis of learning from interaction without requiring models of the environment's dynamics [10]. In this paper, we investigate three learning schemes: 1) Q-learning for single SU, 2) non-cooperative Q-learning for multiple SUs, and 3) cooperative Q-learning for multiple SUs. These schemes are RL-based schemes that are well suited for OSA environments, allowing SUs to

[1]Throughout this work, agents will also be referred to as secondary user groups (SUGs); the terms agent and SUG will then be used interchangeably.

learn by themselves from interaction, and use their acquired knowledge to locate and find best spectrum opportunities, thus achieving efficient utilization of spectral resources.

We evaluate the performance of these proposed schemes and compare them with the random access scheme. Simulation results show that partial and fully cooperative schemes perform better than the non-cooperative and the random schemes in terms of achieved throughput and balanced traffic loads. Depending on the communication overhead due to the extra traffic incurred when exchanging information between the cooperating users, different levels of partial cooperation can be used. Overall, the proposed learning techniques achieve high throughput performance by learning from interaction with the environment and intelligently locating and exploiting spectrum opportunities.

This paper is organized as follows. In Section II, we present some related works. In Section III, we state the problem. In Section IV, we formulate the RL framework, and present the proposed learning schemes. Section V evaluates the schemes, and Section VI concludes this work.

II. Related Work

Brik et al. [11] proposed a centralized protocol for OSA, called dynamic spectrum access protocol (DSAP). DSAP relies on a central unit to coordinate and dynamically allocate spectrum resources. DSAP architecture consists of clients, a server, and transmitters. Each client senses the network to collect information about spectrum usage, and sends this information to the server via a predefined common control channel. The server uses this information to allocate spectrum. Although simple, this model presents scalability, single-point failure, and vulnerability issues due to its centralized nature. Raychaudhuri et al. [12] proposed a spectrum etiquette protocol, called CCSC, for coordinating network nodes in the unlicensed spectra. Unlike DSAP, CCSC is distributed. In CCSC, nodes periodically broadcast spectrum usage information on a dedicated channel so as other nodes, monitoring this channel, can hear about and learn which channels are available. One issue with CCSC is that it does not guarantee, nor does it always result in, the selection of optimal channels.

There have also been several works that proposed learning-based approaches for OSA [13–15]. For example, in [13], Fangwen et al. proposed an RL-based approach that allows cognitive radios to select frequency bands (FBs) with the most available resources. The detection of spectral resources is formulated as a Markov decision process, and a solution strategy based on an actor-critic method is proposed. The objective is to minimize the mutual interference between PUs and SUs while maximizing the utilization of available resources. This scheme assumes prior knowledge of the environment's dynamics. Unlike these works, our work does not require prior knowledge of such dynamics, giving more practical ways of promoting effective OSA.

III. Problem Statement

Reinforcement learning (RL) is the concept of learning from past and present to decide what to do best in the future. That is, the learner, also referred to as *agent*, learns from experience by interacting with the environment, and uses its acquired knowledge to select the *action* that maximizes a cumulative *reward*. RL is well suited for systems whose behaviors are, by nature, too complex to predict, but the reward, or reinforcement, resulting from taking an action can easily be assessed or observed. For example, in OSA, although it may be difficult to predict which spectrum band will be available in the near future, the reward resulting from using a band can easily be determined. The reward can be assessed, for example, through amount of obtained throughput, experienced interference, packet success rate, etc. Thus, RL techniques are a natural choice for OSA because it is difficult to precisely specify an explicit model of the environment, but it is easy to provide a reward function.

We assume that all SUs are associated with a *home* spectrum band (HSB) to which they have usage rights at all time. In order to communicate with each other, all SUs in the same group must be tuned to the same band, being either their HSB or any unused licensed band. While using the HSB, each secondary-user group can opportunistically look for spectrum opportunities in another band. This typically happens when, for example, any of the SUs judge that the quality of their current band is no longer acceptable. This technique can be done by simultaneously assessing and monitoring the quality of the band using quality metrics, such as signal-to-noise ratio (SNR), packet success rate, achievable data rate, etc. The secondary-user group is triggered to start seeking for spectrum opportunities whenever the monitored quality no longer meets for e.g. a minimum threshold that can be defined a priori.

Upon the return of any PUs and/or when the quality of current band drops below the threshold, the agent must either return to its HSB or seek for an available band. Hereafter, we say that an *exploration event* is triggered when either (i) PUs return back to their licensed band, and/or (ii) the band's quality is dropped below the threshold. In the RL terminology, we therefore consider that the agent and the environment interact at each of a sequence of discrete time steps, each of which takes place at the occurrence of an exploration event.

RL is typically formalized in the context of MDPs. In general, an MDP represents a dynamic system, and is specified by giving a finite set of *states* $\mathcal{S}$, representing all possible system states, a set of control *actions* $\mathcal{A}$, a *transition function* δ, and a *reward function* r. The dynamics are Markovian in the sense that the probability of being in the next state s_j depends only on the current state s_i and action a_k, but not on any previous history. A policy for an MDP is a mapping from states to actions.

The objective is then to find a policy that maximizes the expected cumulative reward during its execution. In the next section, we first formulate OSA as a finite MDP, and then describe the Q-learning schemes for OSA.

IV. Q-Learning for OSA

In this section, we formulate OSA as a finite MDP. An MDP is defined by a state sets $\mathcal{S}$, an action set $\mathcal{A}$, a transition function δ and a reward function r. We first consider OSA systems with single SU, and then consider systems with multiple SUs.

A. Systems with Single SU

State set. $\mathcal{S}$ consists of $m+1$ states where m is the number of bands, $\{s_0, s_1, \ldots, s_m\}$, where the system is said to be in state s_i if the agent is either *using* or *sensing* band b_i at the current time step. The agent cannot use any band unless it is free, and the agent cannot know whether the band is free unless it senses it. If band b_i is not available due to the presence of PUs, the system is still considered to be in state s_i. Note that s_0 corresponds to the state when the agent is using its HSB b_0 which is always available.

Action set. $\mathcal{A}$ has $m+1$ actions, $\{a_0, a_1, \ldots, a_m\}$, where taking action a_i always leads to state s_i. At every time step (i.e., an exploration event) while in state s_i, the agent can either choose to *exploit* by switching back to its HSB b_0, or choose to *explore* by searching for new spectrum opportunities. The number of bands that are *sensed* before either finding the first available band or switching back to HSB is referred to as the *dynamic exploration index*, n. The value of this index, which is learned and set via the Q-learning, varies over time and depends on current PUs' load and condition. The Q-learning has the ability to adaptively find the optimal index n that balances between the desire to keep switching/sensing overhead low and the need to maximize the chances of finding spectrum opportunities.

Transition function. $\delta : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ specifies the next state the system enters given the current state s_j and the action a_k to be taken. For any pair (s_j, a_k), $\delta(s_j, a_k) = s_k$.

Reward function. $r : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ specifies the reward $r(s_i, a_k, s_k)$ the agent earns when transitioning to next state s_k as a result of taking action a_k while in state s_i. Specifically, the reward perceived by the agent when entering state s_k is a function of the *quality level* the agent receives when *using* the band it ends up selecting. We therefore assume that each band b_k is associated with a quality level q_k, which can be determined via metrics, such as SNR, packet success rate, data rates, etc. Hereafter, q_k will be used to represent the *positive* reward (without including the cost of exploration yet) that band b_k offers.

Exploration also comes with a cost. Recall that SUs are allowed to use any licensed band only if the band is vacant, and that discovery of opportunities is done through spectrum sensing. That is, SUs periodically, or proactively, switch to and sense certain bands to find out whether any of them are vacant. However, sensing incurs some cost, which is often referred to as *sensing overhead*. This overhead can be of multiple types: energy consumed to perform sensing, delays resulting from switching across bands, throughput wasted as a result of ceasing communication, etc. By letting c_{ik} denote the cost incurred as a result of exploring band b_k while in state s_i, the reward function can be written as

$$r(s_i, a_k, s_k) = \begin{cases} q_k - c_{ik} & \text{if } b_k \text{ is available} \\ -c_{ik} & \text{else} \end{cases}$$

Q-learning. The goal of the agent is to learn a policy, $\pi : \mathcal{S} \rightarrow \mathcal{A}$, for choosing the next action a_k based on its current state s_i that produces the greatest possible expected cumulative reward. A cumulative reward R is typically defined through a discount factor γ, $0 \leq \gamma < 1$, as $\sum_{t=0}^{\infty} \gamma^t r(s_{i+t}, a_{k+t})$. Because it is naturally desirable to receive rewards sooner than later, the reward is expressed in a way that future rewards are discounted with respect to immediate rewards.

The optimal policy is calculated using Q-learning [10]. A function, $Q : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$, is defined for each state-action (s_i, a_k) pair as *the maximum discounted cumulative reward* that can be achieved when starting from state s_i and taking action a_k according to the optimal policy. Thus, given the Q function, it is possible to act optimally by selecting actions that maximize $Q(s_i, a_k)$ at each state. Q can be constructed recursively as follows. The Q-learning algorithm learns an estimate $\hat{Q}$ of the optimal Q-function by selecting actions and observing their effects. In particular, each step in the environment involves taking an action a_k in state s_i and then observing the following state s_k and the resulting reward. Given this information, Q is updated via the following equation:

$$\begin{aligned} \hat{Q}(s_i, a_k) \leftarrow \quad & (1 - \alpha_l)\hat{Q}(s_i, a_k) + \alpha_l\{r(s_i, a_k) \\ & + \gamma \max_{k'} \hat{Q}(\delta(s_i, a_k), a_{k'})\} \end{aligned}$$

where $\alpha_l = 1/(1 + visits_l(s_i, a_k))$ and $visits_l(s_i, a_k)$ is the total number of times this state-action pair has been visited up to and including the l^{th} iteration. This stochastic approximation algorithm is guaranteed to converge to the optimal Q-function in any MDP given the appropriate exploration during learning [10].

B. Systems with Multiple SUs

We now consider the case of multiple SUs. We assume there is no HSB.

State set. $\mathcal{S}$ consists of one state s only ($\mathcal{S} = \{s\}$).

Action set. At each time step, the agent chooses an action from the action set $\mathcal{A} = \{a_1, a_2, \ldots, a_m\}$, where m is the number of bands. The number of actions is equal to the

number of spectrum bands in the system. Taking action a_i while using spectrum band b_j makes an agent enter and use spectrum band b_i.

Reward function. We assume that each band b_i has its own bandwidth capacity V_i, and when more than one SUG use a spectrum band, the bandwidth is equally divided among all the SUGs using the band. For example, if there is a total number of 3 SUGs, A, B, and C, each taking action i, j, and k respectively, then the reward of SUG A, denoted by ra_{ijk}, can be calculated as

$$ra_{ijk} = \begin{cases} V_i/3 & \text{when } i = j = k \\ V_i/2 & \text{when } i = j \neq k \text{ or } i = k \neq j \\ V_i & \text{when } i \neq j \neq k \end{cases}$$

Non-cooperative Q-learning. The function Q, as defined in the previous section, can be constructed recursively [14] as follows.

$$\begin{aligned} Q(s, a_i)(t+1) = \quad & Q(s, a_i)(t) + \alpha \\ & \times (E[r(s, a_i)] - Q(s, a_i)(t)) \end{aligned}$$

where $0 < \alpha < 1$ is the learning rate. When using the non-cooperative Q-learning scheme, each SUG calculates its Q table independently from other SUGs.

Action selection. The action selection mechanism plays a very important role in Q-learning. During the learning process, this selection mechanism is what enables the agent to choose its actions. We consider the ϵ-greedy exploration as the action selection mechanism, where the action corresponding to the highest Q value in that time step is chosen with a probability of $(1 - \epsilon) + \epsilon/m$, and any other action from the action set $\mathcal{A}$ is chosen with a probability of ϵ/m. The ϵ-greedy mechanism balances between exploration and exploitation.

Probability vector. Based on the ϵ-greedy exploration, we define the probability vector over the action set as follows. $X = (x_1, x_2, \ldots, x_m)$, where x_i is the probability of taking action i

$$x_i = \begin{cases} (1 - \epsilon) + \epsilon/m & \text{if } Q_i \text{ is the highest value} \\ \epsilon/m & \text{otherwise} \end{cases}$$

where again m is the number of actions.

Cooperative Q-learning. Our multi-agent cooperative scheme is based on the multi-agent Q-learning approach derived in [16]. To illustrate, suppose that SUG A with probability vector X is going to cooperate with two other SUGs, B and C, with probability vectors Y and Z, respectively. The Q table entry for SUG A choosing action i can be calculated as [16]:

$$\begin{aligned} Q(s, a_i)(t+1) &= Q(s, a_i)(t) + x_i(t)\alpha \\ &\times [(\textstyle\sum_{j=1}^{j=m} y_j(t) \sum_{k=1}^{k=m} (ra_{ijk})(z_k(t))) - Q(s, a_i)(t)] \end{aligned}$$

Similarly, each SUG can compute its Q table values based on the probability vectors of the other SUGs.

V. Evaluation

We now evaluate the performance of the proposed schemes. We first show the results for the single SU model, and then for the multiple SUs model.

A. Single SU

We study the single-user Q-learning by evaluating and comparing its performance to a *random* access model. This model will be used here as a baseline for comparison, and is defined as follows. Whenever an exploration event is triggered, the secondary-user group, using the random access model, selects a spectrum band among all bands randomly. If the selected band is idle, then the group uses it until the return of any PUs associated with this band. Otherwise, i.e., if the selected band happens to be busy, then the group goes back to its home band. This process repeats until an idle band is found.

1) **Environment Setup:** We assume that the spectrum is divided into m non-overlapping bands, and that each band is associated with a set of PUs. PUs' traffic in the spectrum band is mimicked by considering ON and OFF alternating periods. ON periods denote that PUs are present while OFF periods denote their absence. ON and OFF periods on the i^{th} band are taken from the exponential distribution with rates λ_i and μ_i. The PU traffic load η_i of the i^{th} band is then expressed as $\mu_i/(\mu_i + \lambda_i)$.

The exponential distributions will be used in this work to generate samples to be able to evaluate our learning scheme. Recall that the power of RL lies in its capability to converge to approximately optimal behavior without needing prior knowledge of PUs traffic behavior. Throughout, we characterize PUs traffic activities by the average, $\bar{\eta} = \frac{1}{m}\sum_{i=1}^{m} \eta_i$, and the coefficient of variation, $CoV = \sigma/\bar{\eta}$, of PUs traffic loads across all bands, where σ denotes the standard deviation of these traffic loads.

2) **Effect of the Average of PUs Traffic Load:** We begin by studying the effect of the average of PUs traffic load $\bar{\eta}$ on the achievable throughput. Fig. 1 plots the total throughput, normalized w.r.t. the maximal achievable throughput, that the SUG achieves as a result of using single-user Q-learning (Q-OSA in the figures) and the random access for two different PUs traffic loads: $\bar{\eta}$ = 0.5 and $\bar{\eta} = 0.8$. The measured throughput is based on what the SUG receives from the m licensed bands only; i.e., not counting for the HSB. In this simulation scenario, CoV is set to 0.1443 and the total number of bands m is set to 10.

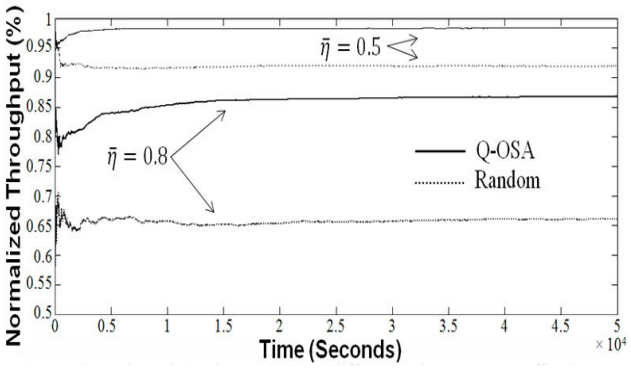


Fig. 1. Throughput behavior under two different PUs traffic loads, $\bar{\eta} = 0.5$ and 0.8, for $m = 10$ and $CoV = 0.1$

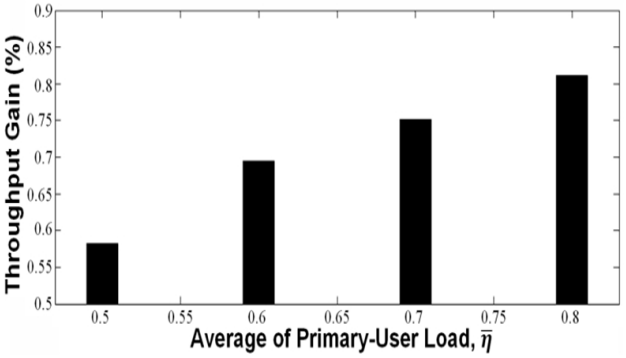


Fig. 2. Throughput gain as a function of PUs average loads $\bar{\eta}$ for $m = 10$ and $CoV = 0.1443$

First, note that as expected, the higher the $\bar{\eta}$, the lesser the achievable throughput under both schemes. However, regardless of the PUs traffic load, Q-learning always outperforms the random scheme. Also, note that the more loaded the system is, the higher the difference between the throughput achievable under Q-learning and that achievable under random access (e.g., gain is higher when $\bar{\eta} = 0.8$).

To further illustrate the effect of $\bar{\eta}$ on the performance of single-user Q-learning, we plot in Fig. 2 the throughput gain (single-user Q-learning w.r.t. to random access) as a function of $\bar{\eta}$. Note that the gain increases as the PUs traffic load increases. In other words, the Q-learning performs even better under heavy loaded systems. Note that the gain can be as high as 80% when $\bar{\eta} = 0.8$. When $\bar{\eta}$ is high; i.e., when spectrum opportunities are scarce, the learning capability of Q-learning allows the agent to efficiently locate where the opportunities are, whereas random access leads to less throughput since it is accessing bands randomly. When $\bar{\eta}$ is small, on the other hand, the random access scheme is able to achieve high throughput since spectrum opportunities are too many to miss even when bands are selected unintelligently. Observe that not only Q-learning outperforms the random access (which is expected), but also achieves close-optimal throughput; Fig. 1 shows that the normalized throughput can be as high as 90%, meaning that Q-learning can achieve up to 90% of that achievable under an ideal scheme.

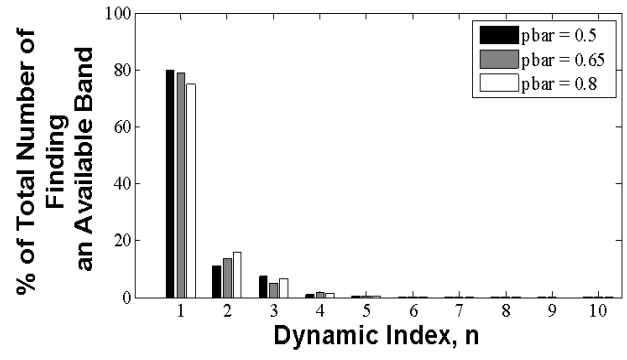


Fig. 3. Dynamic index behavior for different value of PUs average loads: $CoV = 0.1443$, $m = 10$

In Fig. 3, we show the dynamic index n behavior for different values of $\bar{\eta}$ (pbar in the figure) under Q-learning. Recall that this index denotes the number of bands that are *sensed* before either finding the first available band or switching back to HSB. Clearly, Fig. 3 shows that the agent is almost always able to find an available band from the first ($n = 1$) attempt; regardless of PUs loads, more than 75% of the time, the agent finds an available band in its first attempt. Thus, the learning capability of Q-learning allows the agent to quickly locate available bands.

To summarize, these obtained results show that the proposed Q-learning is capable of achieving between 80% and 95% of the maximal achievable throughput. Also, more than 75% of the time, Q-learning hits the available band from the first attempt. Results also show that Q-learning achieves high throughput performance even under heavy PUs traffic loads.

3) **Effect of the Variation of PUs Traffic Load:** Fig. 4 plots the total throughput that the secondary-user group achieves under Q-learning and the random access for two different PUs load variations: $CoV = 0$ and $CoV = 0.144$. Note that when $CoV = 0.144$, Q-learning outperforms the random scheme by simply locating and exploiting unused opportunities through learning. As expected, the throughput gain increases with the variation. As shown in Fig. 4, the gain is higher when $CoV = 0.144$ than when $CoV = 0$. To further illustrate the effect of PUs load variability on throughput, we show in Fig. 5 the throughput gain for different values of CoV.

The average PUs traffic load, $\bar{\eta}$, is set to 0.6 (i.e., only 40% of the spectrum is available). Observe that the higher the variation of PUs loads across different bands, the higher the throughput gain; i.e., the higher the throughput the agent/group can achieve when compared with that achievable under the random access scheme. This can be explained as follows. When the average of PUs traffic loads is kept the same, a high variation in the loads across

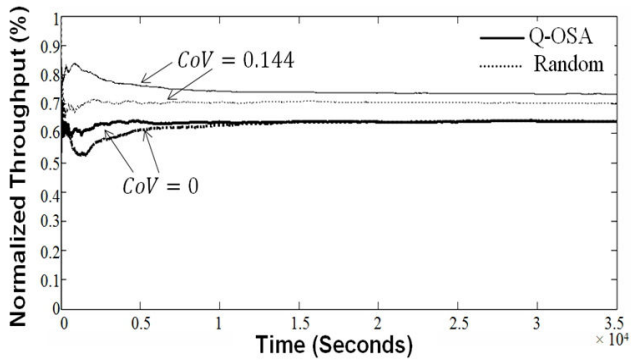


Fig. 4. Achievable throughput under single-user Q-learning and random access schemes $\bar{\eta} = 0.8$, $m = 10$

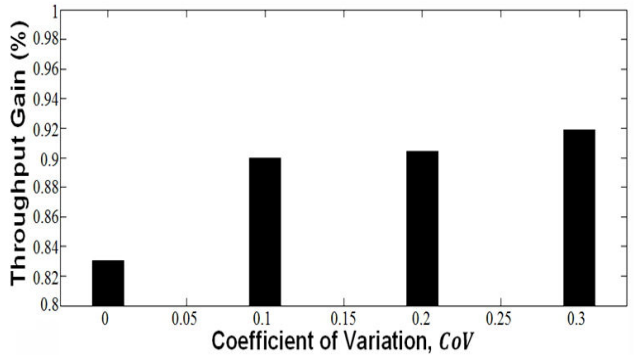


Fig. 5. Throughput gain as a function of PUs load variability: $\bar{\eta} = 0.6$,$m = 10$

different bands increases the likelihood of finding highly available spectrum bands. This, on the other hand, also increases the likelihood of finding spectrum bands with fewer opportunities. With experience, Q-learning learns about, and starts exploiting, these more available bands, yielding then more throughputs. When the load variation is low, on the other hand, Q-learning achieves lesser throughput because all bands are almost equally-loaded, and hence, there is no special (i.e., more available) bands that the agent can learn about.

To further illustrate this effect, we show the dynamic index n behavior in Fig. 6 under different values of CoV. As expected, the lower the CoV, the greater the number of bands to be sensed before finding an available band. When $CoV = 0$, Q-learning does not find an available band as fast as when $CoV = 0.35$.

B. Multiple SUs

We consider multiple SUs, and show the importance of multi-agent cooperation by comparing the per SUG average received throughput of the cooperative scheme with that of a non-cooperative one. Specifically, we study the effect that cooperation has on network load balancing by allowing SUGs to make better action decision, leading to more

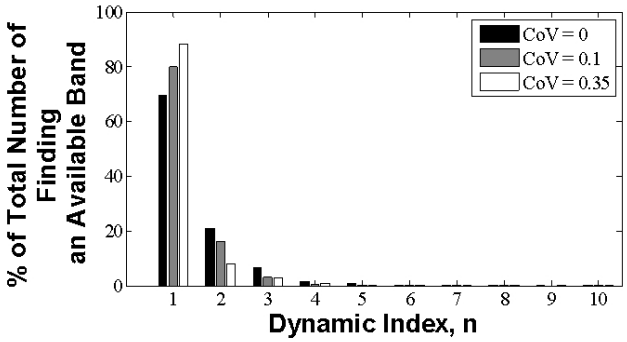


Fig. 6. Dynamic index behavior for different values of PUs load variability: $\bar{\eta} = 0.6$,$m = 10$

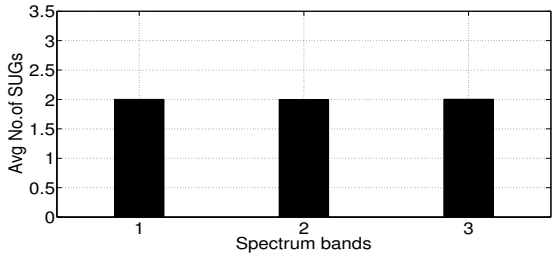
effective exploitation of bandwidth opportunities. This also ensures fairness among SUGs by making sure that all SUGs receive (approximately) equal throughput shares.

1) **Simulated Access Schemes:** We consider that the spectrum is divided into m non-overlapping spectrum bands with n SUGs (unlike the previous sections, hereafter, n represents the number of SUGs). We mimic the presence of PUs by considering different spectrum bands with different bandwidth capacities. Let V_j denote the bandwidth capacity of band j. A spectrum band with a higher bandwidth capacity is meant to have a lower PU activity, and vice versa. We consider a time-slotted system, and assume that SUGs interact with the environment in accordance with these time slots. That is, SUGs can only enter or leave a band at the beginning and at the end of these time steps. We now summarize the three access schemes that are evaluated in this subsection.

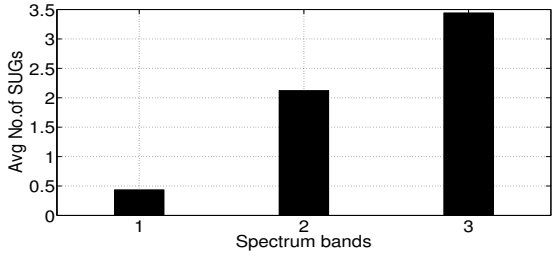
Random access scheme. At the end of each time slot/step, an SUG using the random access scheme selects a spectrum band among the m available bands randomly, and uses it during the next time slot. If more than one SUG happen to select the same spectrum band, they share the bandwidth of the band equally.

Non-cooperative Access Scheme. In the non-cooperative access scheme, each SUG uses the non-cooperative Q-learning policy discussed in Section IV-B to create and update its own Q table. Each SUG enters the environment and takes actions based on its own Q table without cooperating with any of the other SUGs. When two or more SUGs choose the same band during the same time step, they share its bandwidth equally. Although the SUGs are typically unaware of the other agent's actions and act independently, the effect of the other SUG's actions are reflected in the reward that the SUGs receive from the spectrum band.

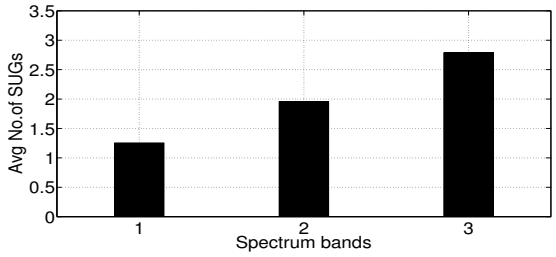
Cooperative Access Scheme. In the cooperative access scheme, each SUG maintains its own Q table using the cooperative Q-learning, discussed in Section IV-B. Here, an agent's Q table is formulated by taking into account



(a) Random



(b) Non-cooperation



(c) Full cooperation

Fig. 7. SUG distribution: $m = 3$, $n = 6$, $V_j = [5\ \ 10\ \ 15]$.

the probabilities associated with the actions of the other SUGs with which it cooperates. In this case, at each time step, the SUG is provided with the probability vector of every other SUG with which it cooperates. The tradeoff here is between the communication overhead caused by extra traffic needed for exchanging the probability vectors among the cooperating SUGs and the performance gains due to improved action selections because of cooperation.

2) **Cooperation Vs. Non-cooperation:** First, we consider an OSA system with $m = 3$ spectrum bands and $n = 6$ SUGs. Bandwidth capacities are set to $V_j = [5\ \ 10\ \ 15]$. In this scenario, an ideal balanced spectrum load is reached when each of the SUGs gets a reward of 5 units, which implies that the 1^{st} band has 1 SUG, the 2^{nd} has 2 SUGs, and the 3^{rd} band has 3 SUGs. We simulate the three different access schemes for this scenario, and plot the average number of SUGs (averaged over 10000 episodes) in each of the three spectrum bands (i.e., the distribution of SUGs) in Fig. 7.

The figure shows the average number of SUGs that end up choosing each of the three spectrum bands for each of the three studied schemes. It can be observed that the

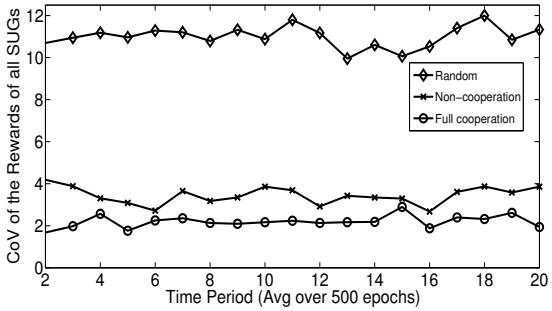


Fig. 8. Coefficient of variation of the rewards of all the SUGs at each time period: $m = 3$, $n = 6$, $V_j = [5\ \ 10\ \ 15]$.

fully cooperative access scheme leads to the ideal balanced system load. As explained earlier, this is because in the fully cooperative method, each SUG accounts for all the possible actions that could be taken by its counterparts when making a decision. On the other hand, when SUGs do not cooperate, they may not select the best available band, as they have no clue what other SUGs will select, leading to a lesser balanced load distribution when compared with that of the cooperative scheme. Clearly and as expected, the Random access scheme results in an equally distributed SUGs among all bands, leading to the worst load balance when compared with the other two schemes[2].

Fairness is another important metric that we also evaluate in this work. To do this, we plot in Fig. 8 the coefficient of variation (CoV) of the received rewards of all the SUGs as a function of time period (each time period corresponds to 500 epochs). Observe that the fully cooperative access scheme has the lowest CoV among the three schemes. The lower the CoV is, the closer the SUGs' received rewards are to one another, indicating a fairer access scheme. It can also be seen that the CoV of the non-cooperative access scheme is approximately twice that of the fully cooperative access scheme, and the CoV of the random access scheme is substantially higher than the other two. Therefore, cooperation improves performances not only in terms of network load balancing, but also in terms of ensuring fairness among all SUGs.

3) **Impact of Degree of Cooperation:** Recall that cooperation increases the performance because it allows the SUGs to make a better decision when selecting their next actions. This is because the SUGs take into account what other SUGs will select when making their action decisions. However, acquiring such information would necessitate the exchange of messages among cooperative SUGs, which clearly incurs extra overhead. Therefore, the challenge is to strike a good balance between the desire for a higher level of cooperation that enables a better action selection

[2]We want to mention that these above results do not account for the communication overhead caused by message exchange needed to share the probability vectors among cooperative SUGs.

and the need for a lower level of cooperation so as to keep the cooperation overhead to a minimum. Cooperation overhead comes from the extra traffic needed to exchange the probability vectors and also from the computing delay/time resulting from solving the complex equations involved in updating the Q table entries of the cooperative SUGs.

We now study the impact of degree of cooperation on the achievable performances of a OSA system with $m = 3$ spectrum bands and $n = 12$ SUGs. The bandwidth capacities of the spectrum bands are set to $V_j = [10\ \ 20\ \ 30]$. In this scenario, an ideal balanced load is reached when each of the SUGs earns a reward of 5 units, corresponding to when the 1^{st} band houses 2 SUGs, the 2^{nd} band houses 4 SUGs, and the 3^{rd} band houses 6 SUGs. For this simulation scenario, we evaluate and compare the performances of the cooperative access scheme by considering three degrees of cooperation: 2 (i.e, each SUG cooperates with 2 other SUGs), 4 (i.e, each SUG cooperates with 4 other SUGs), and 6 (i.e, each SUG cooperates with 6 other SUGs).

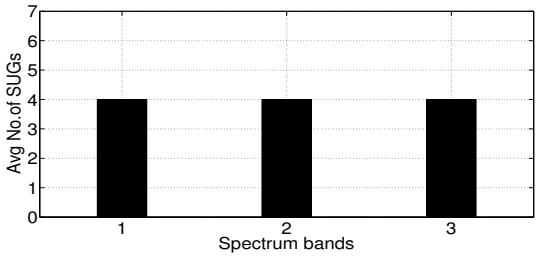
Fig. 9 shows the average number of SUGs that end up choosing each of the three spectrum bands for the random, non-cooperative, and cooperative access schemes with 2, 4 and 6 degree of cooperation. Note that as the degree of cooperation increases, the system load becomes more balanced. That is, the cooperative access scheme with degree of cooperation equal to 6 leads to a better balanced system load when compared with the other two degrees.

We also study fairness achieved under each of the three cooperation degrees, and plot the CoV of the received rewards of the SUGs in Fig. 10. Observe that cooperation with a degree of 6 has the lowest CoV, followed by a degree of 4, and then followed by a degree of 2. This indicates that a higher degree of cooperation leads to a lower CoV, meaning that SUGs receive closer amounts of rewards, thus ensuring fairness among SUGs. Therefore, cooperation improves performances not only in terms of network load balancing, but also in terms of ensuring fairness among all SUGs. Note that each of the three degrees of cooperation has a lower CoV when compared with the non-cooperative and random access schemes.

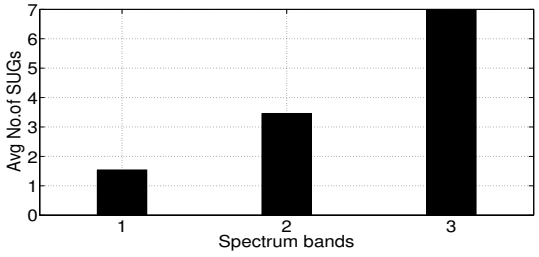
It is important to mention again that although higher degree of cooperation results in improved action selection decisions, it also incurs more communication overhead and execution times. Therefore, one must choose the degree of cooperation that balances between good selection decision and minimum overhead so as to lead to an increased overall system performance.

VI. Conclusion

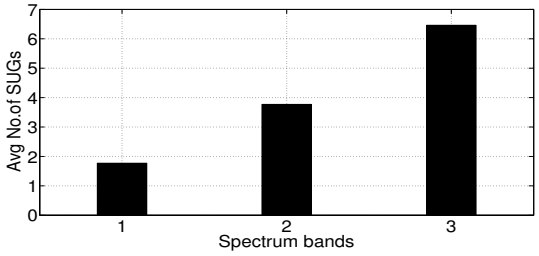
In this paper, we developed a reinforcement learning based framework for DSA system with multiple secondary users. We evaluated and compared two multi-agent Q-learning algorithms, namely the non-cooperative and the cooperative Q-learning schemes along with the random scheme. Simulation results showed that partial and fully



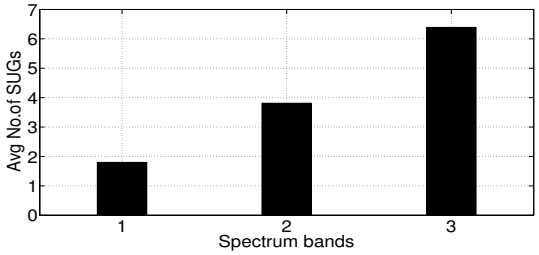
(a) Random



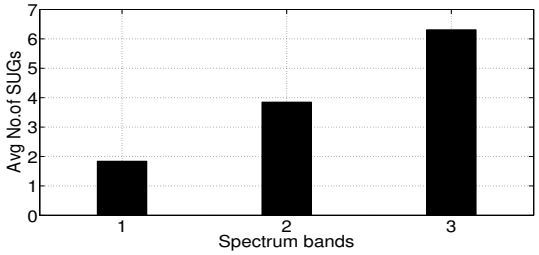
(b) Non-cooperation



(c) Cooperation with 2 SUGs



(d) Cooperation with 4 SUGs



(e) Cooperation with 6 SUGs

Fig. 9. SUG distribution: $m = 3$, $n = 12$, $V_j = [10\ \ 20\ \ 30]$.

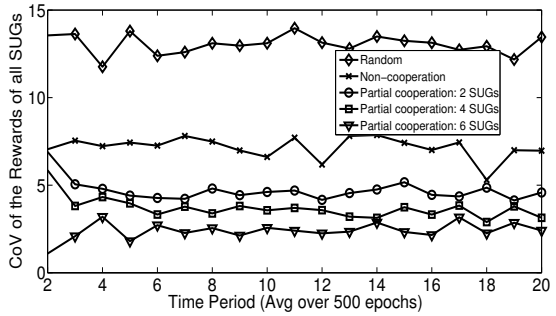


Fig. 10. Coefficient of variation of the rewards of all the SUGs at each time period: $m = 3$, $n = 12$, $V_j = [10 \;\; 20 \;\; 30]$.

cooperative access schemes perform better than the non-cooperative and the random access schemes in terms of achieving a higher throughput and a better balanced traffic loads. We also showed that cooperation improves performances not only in terms of network load balancing, but also in terms of ensuring fairness among all users.

REFERENCES

[1] M. McHenry, "Reports on spectrum occupancy measurements, shared spectrum company," in *www.sharedspectrum.com/?section=nsf_summary*.

[2] M. McHenry and D. McCloskey, "New york city spectrum occupancy measurements," *Shared Spectrum Conference*, September 2004.

[3] M. Gandetto and C. Regazzoni, "Spectrum sensing: a distributed approach for cognitive terminals," *IEEE Journal of Selected Areas in Communications*, April 2007.

[4] B. Hamdaoui and K. G. Shin, "OS-MAC: An efficient MAC protocol for spectrum-agile wireless networks," *IEEE Transactions on Mobile Computing*, August 2008.

[5] C.-T. Chou, S. Shankar, H. Kim, and K. G. Shin, "What and how much to gain by spectrum agility," *IEEE Journal on Selected Areas in Communications*, April 2007.

[6] K. Lee and A. Yener, "Outage performance of cognitive wireless relay networks," in *Proceedings of IEEE GLOBECOM*, 2006.

[7] Z. Ji and K. J. R. Liu, "Multi-stage pricing game for collusion-resistant dynamic spectrum allocation," *IEEE Journal on Selected Areas in Communications*, January 2008.

[8] R. W. Thomas, L. A. DaSilva, M. V. Marathe, and K. N. Wood, "Critical design decisions for cognitive networks," in *Proceedings of IEEE ICC*, 2007.

[9] X. Jing and D. Raychaudhuri, "Global control plane architecture for cognitive radio networks," in *Proceedings of IEEE ICC*, 2007.

[10] R. S. Sutton and A. G. Barto, *Reinforcement Learning*, The MIT Press, 1998.

[11] V. Brik, E. Rozner, S. Banerjee, and P. Bahl, "Dsap: A protocol for coordinated spectrum access," in *IEEE DySPAN*, 2005.

[12] D. Raychaudhuri and X. Jing, "A spectrum etiquette protocol for efficient coordination of radio devices in unlicensed bands," in *IEEE Personal Indoor and Mobile Radio Conference*, 2003.

[13] M. Van Der Schaar F. Fu, "Stochastic game formulation for cognitive radio networks," in *IEEE DySPAN*, 2008.

[14] V. V. Veeravalli J. Unnikrishnan, "Dynamic spectrum access with learning for cognitive radio," in *Asilomar Conference on Signals Systems and Computers*, 2009.

[15] Q. Zhao H. Liu, B. Krishnamachari, "Cooperation and learning in multiuser opportunistic spectrum access," in *IEEE ICC*, 2008.

[16] E.R. Gomes and R. Kowalczyk, "Dynamic analysis of multiagent Q-learning with ϵ-greedy exploration," in *Proceedings of the 26th International Conference on Machine Learning*, 2009, pp. 369–376.