Non-Thesis Project:
Spectrum Management with Replicated Q-Learning

Luke Prince

May 2019

Contents

Chapter 1

Introduction

1.1 Learning with Markov Decision Processes

Markov processes are stochastic processes that are able to model systems which exhibit Markovian evolution, where the probability of progressing to a new state is solely dependent on the current state and independent of any previous state history. Formally, the Markov property may be written as:

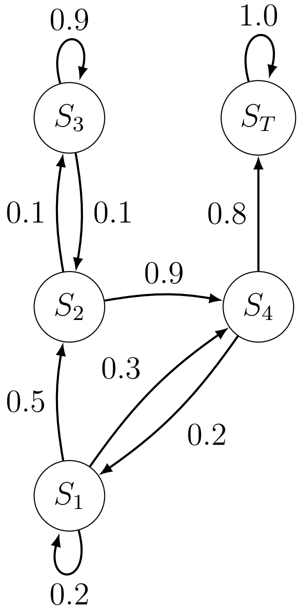
$$\Pr(S_{n+1}|S_n) = \Pr(S_{n+1}|S_1, \dots, S_n)$$

A Markov process may exhibit terminating states, in which case the process is of finite duration. They may also only have non-terminating states. Then the process continues indefinitely. In the context of temporal states, a Markov process of such infinite time-horizon may be used to model a process with a well-defined probabilistic state structure that has no definite beginning or end while it is being observed.

An example of a finite Markov process with finite and discrete states is depicted in Figure 1.1.

The state transition probabilities for the example process may be summarized in a state transition probability matrix:

Figure 1.1: Example of a finite Markov process with discrete state



$$\mathcal{P}_{ss'} = \begin{array}{cc} & \begin{matrix} S_1 & S_2 & S_3 & S_4 & S_T \end{matrix} \\ \begin{matrix} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_T \end{matrix} & \begin{pmatrix} 0.2 & 0.5 & 0.0 & 0.3 & 0.0 \\ 0.0 & 0.0 & 0.1 & 0.9 & 0.0 \\ 0.0 & 0.1 & 0.9 & 0.0 & 0.0 \\ 0.2 & 0.0 & 0.0 & 0.0 & 0.8 \\ 0.0 & 0.0 & 0.0 & 0.0 & 1.0 \end{pmatrix} \end{array}$$

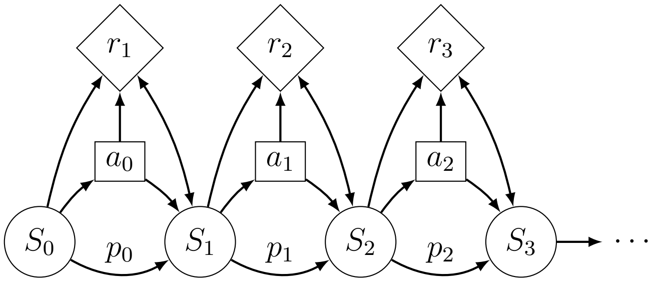
The same may be done for similar infinite time-horizon process.

A learning agent may be defined as any entity which learns to reason through observations of an environment. Many real world environments may be successfully modelled as a Markov process.

Extending on this, Markov decision processes (MDPs) incorporate actions and rewards into the Markov process model allowing for learning agents to decide upon actions and collect rewards from their actions. The MDP extension of Markov processes is graphically depicted in Figure 1.2. MDPs are equivocally a Markov model which follow an alternative Markov property:

$$\begin{aligned} p(s', r|s, a) &\triangleq \Pr(S_{n+1} = s', R_{n+1} = r|S_n = s; A_n = a) \\ &= \Pr(S_{n+1}, R_{n+1}|S_n, \ldots, S_1; A_n, \ldots, A_1) \end{aligned}$$

Figure 1.2: Markov Decision Process (MDP) extension to Markov processes. The decision aspect is reflected by the actions a_n taken and the rewards r_{n+1} received. Here the subscript indices reflect a sequence in time



Thus the MDP is fully characterized by a joint state transition and reward distribution. With this, the complete dynamics of the underlying MDP are captured[1]. In addition, the following marginal distribution may be derived:

$$\begin{aligned} p(s'|s,a) &\triangleq \Pr(S_{n+1} = s'|S_n = s; A_n = a) \\ &= \sum_{r \in \mathcal{R}} p(s', r|s,a) \end{aligned}$$

$$\begin{aligned} p(r|s,a) &\triangleq \Pr(R_{n+1} = r|S_n = s; A_n = a) \\ &= \sum_{s' \in \mathcal{S}} p(s', r|s,a) \end{aligned}$$

We may also construct functions for the expected reward for state-action pairs:

$$\begin{aligned} r(s,a) &\triangleq E\left\{R_{n+1}|S_n = s, A_n = a\right\} \\ &= \sum_{r \in \mathcal{R}} r \sum_{s' \in \mathcal{S}} p(s', r|s,a) \end{aligned}$$

The expected reward for state-action-state triples follows from application of the chain rule:

$$
\begin{aligned}
p(s', r|s, a) &= \frac{p(s', r', s, a)}{p(s, a)} \\
&= \frac{p(r|s, a, s')p(s, a, s')}{p(s, a)} \\
&= p(r|s, a, s')p(s'|s, a) \\
r(s, a, s') &\triangleq E\left\{R_{n+1}|S_n = s, A_n = a, S_{n+1} = s'\right\} \\
&= \sum_{r \in \mathcal{R}} r \frac{p(s', r|s, a)}{p(s'|s, a)}
\end{aligned}
$$

In a learning environment modelled by an MDP, the environment states are influenced by the decided actions an interacting learning agent follows and every possible state transition is characterized by a worth to the agent through an immediate reward. It is in an agent's best interest to select actions which maximize the reward it expects to gain, either in the short term or long term, through a sequences of action decisions. Learning comes into play when an agent seeks to find the optimal action decision policy to follow for the environment is interacting with.

1.2 Learning a Decision Policy through Dynamic Programming

The computation of an optimal decision policy, denoted as π, tabulating the optimal action choices of an agent for every environment state falls into the subject of dynamic programming (DP) [1, 2, 3]. The concept of dynamic programming was developed and introduced by Richard Bellman as a solution to the problem of optimal control of dynamic systems and finds application in the design of controllers which maximize or minimize a measure of such a dynamic system over time [1, 4]. The methods of dynamic programming rely on the understanding of a dynamic system's state and of a value function defining the measure which to maximize or minimize. In the context of dynamic programming this is often referred to as the "Bellman equation". In general, dynamic programming encompasses methods of solving complex problems that consist of both optimal substructure (an optimal solution does indeed exist) and overlapping sub-problems (recurring sub-problems exist in which solutions may be remembered and reused). A discrete stochastic variant of the dynamic systems in the optimal control problem which possess the Markov property are the previously introduced Markov decision processes and are shown to satisfy both conditions of optimal substructure and overlapping sub-problems, allowing for the successful application of dynamic programming to their solution [5]. While dynamic programming algorithms are able to solve for the optimal decision policies for MDPs, they are limited by computational expense and the dependency of having full knowledge of an underlying MDPs parameters, in particular the transition probabilities between states which characterize the stochastic behaviour of it's discrete states.

1.3 Learning Decision Policies through Reinforcement

Reinforcement Learning (RL) [1, 6, 7] has evolved in the past few decades as an approach to solving MDPs when the transition probabilities are unknown. Reinforcement learning is modelled after how living creatures naturally learn through positive and negative feedback from their choices and actions. Q-learning [6] is a popular classic RL approaches to finding optimal decision policies involving the updating of a table $Q(s,a)$ with each element representing the value of taking different actions a from a particular state s through a learning update step:

$$Q(s,a) \mapsfrom Q(s,a) + \Delta Q(s,a)$$

For a reinforcement learning agent, it is important that for each update step, the agent possesses a perfect observation of the current state of the environment. With this complete state information along with the results from past iterations, the learning agent is then able to make an optimal decision of next action according to its current knowledge all while updating its tabulated values.

1.4 Learning in Primary User Spectrum Environment

In information theory, communications, and intelligent radio research, Markov processes and MDPs are a topic both historical significance and modern interest [8, 9, 10]. Recent experimental measurements from [11] have shown that the RF activities of a primary user can be modelled appropriately or approximately characterized by Markov processes. Partially observable MDPs (POMDPs), in which observations of state are imperfect, are of particular interest and have seen use as a spectrum model for the problem of spectrum sensing and decision making in the presence of primary licensed users [12, 13, 14]. POMDPs capture uncertainty in the underlying Markov model and extends the MDP to processes with hidden states. These states may not be observable or may be incompletely characterized, Partial observability may also capture the effect of perception/sensing noise at the agent/environment interface. In a POMDP, it is impossible to know the current state with complete certainty. Solving POMDP problems are inherently more complex then the equivalent MDP.

A prediction, or belief, of the set of current truth states in a POMDP is a function of the action and state histories leading up to the current partial observation. If a distribution of state beliefs $b(s)$ can be established, then it forms a sufficient statistic from which a POMDP may be treated as a completely observable MDP. Following in this direction, the task of determining the belief state for each agent/environment interaction becomes a critical aspect of success for learning agents assuming an underlying Markov environment model.

Extending upon solution methods for POMDPs [15], reinforcement learning methods have also been applied to both channel sensing and subband selection in the presence of primary licensed user activity [16, 17, 18].

Continued research in spectrum sensing and management is motivated by the ever increasing demand of available spectrum for use in wireless devices and radio communications. The topic of spectrum availability and under-utilization is addressed in Appendix B.

1.5 Overview

In this project, reinforcement learning (RL) techniques are examined for their potential in application towards distributed autonomous spectrum management by intelligent radio agents.

Experiments in spectrum access are performed by spectrum agents to observe the empirical limits of distributed and decentralizing learning with multiple channels in a primary user and secondary (ab)user (PU/SU) paradigm.

In Chapter 2 the concept of a spectrum agent is elaborated upon and connected to the idea of a “cognitive radio”. Challenges such agents encounter such as spectrum management and minimal primary user interference are explored and linked to the task of learning through RF environment observation and interaction. Spectrum access paradigms are defined which set up the motivation for the application of Markov modelling and RL techniques to the spectrum management problem.

In Chapter 3 a brief background in the relevant MDP, DP, and RL topics supporting the experimental work is provided. Of primary interest is learning in a decentralized manner; that is, without a common control channel, data fusion center, or relay nodes facilitating in the sharing of observations, actions, and learning between spectrum agents. In this context, each spectrum agent learns independently of other agents which may be present in the spectrum environment. Thus solutions to the single agent learning problem are analogous to the “optimal control” problem which DP and RL address.

Of primary interest in this work is an extension of Q-learning called replicated Q-learning [15, 16, 19, 17] which has shown promise in use to learn and update a SU decision policy for spectrum access. In the application of replicated Q-learning, the assumption of spectrum channel independence is made. Due to this, the overall spectrum subband state transition model with L channels may be factored into L channel state transition models. Providing an expected reward function for each channel then constructs an equivalent factored MDP problem with each factor characterizing an indivdual and independent channel of the overall subband.

For single agent learning, an alternative formulation of spectrum learning with independent channels is the restless multi-armed bandit (MAB) problem [20], where MABs are a specialized case of MDPs consisting of a single state and $r(s, a, s') = r(1, a, 1) = r(a)$. Optimal or near optimal policies can be found for a restless MAB [20, 21, 22]. A subband spectrum POMDP wih i.i.d expected rewards for each independent channel may be considered equivalent to the respective restless MAB.

In Chapter 4, a spectrum channel MDP environment is developed along with a corresponding composite spectrum subband MDP.

By exploiting the factored MDP representation, replicated Q-learning may be applied to each channel MDP. In this project, the factored replicated Q-learning method is constructed for a spectrum access problem and demonstrated to be well performing in a toy spectrum access regime. The experimental simulations of Chapter 5 contained within verify results of a similar approach and application of replicated Q-learning to the spectrum access problem by Bkassiny [19], with the main difference of approach being the agents learn from the factored MDP representation of channel MDPs rather than the composite subband MDP representation. This takes full advantage of the sub-problem structure forming the foundation of dynamic programming and likewise reinforcement learning.

Chapter 2

Spectrum Agents and Cognitive Radio

2.1 The Concept of a Cognitive Radio

Wireless spectrum is currently an underutilized resource. The availability of spectrum varies in the temporal, spectral, and geological domains. A primary limiting factor for the use of available spectrum are the current practices of spectrum licensing. To meet the overwhelming demand for wireless services, new solutions that overhaul the current legacy licensing model are necessary. A widely adapted ideology and term is that of *cognitive radio* (CR) originally proposed by Mitola in 1999 [23]. Expanding upon the concept of software defined radios (SDRs), Mitola envisioned cognitive radios as multiband multimode reconfigurable radio systems which incorporated reasoning, or "cognitive", capabilities. In other words, it is a radio that could adapt to its RF environment dynamically.

The term *cognitive*, though, implies more than just a learning capability for the radios. It suggests self-awareness of an RF environment and an ability to apply knowledge to better fulfill the needs of the communication channel. Therefore, a CR is an intelligent agent capable of learning. This learning is accomplished during the radios "cognition cycle" in which the CR interacts with the RF environment [24]. Such a cognition cycle involves *observation* of the environment, or a sensing problem. These observations may also include internal states of the CR, which leads the cycle towards *learning* via transforming the observed information into knowledge collectively kept in a knowledge base. We then also have a learning problem. At this stage in the cycle, the CR makes a reasoned *decision* on how to allocate it's resources to achieve its communication goals. This may be considered a policy or decision making problem. These decisions may change the internal state of the CR. The transition to new states both internally and in the RF environment lead to new observations thus repeating the cognition cycle.

2.2 Signal Processing Challenges for Cognitive Radios

Following the concept of a cognition cycle, a more adequate definition of cognitive radio may be developed:

> "...A cognitive radio is a multiband multimode, wideband SDR supporting autonomous decision-making and the capability to learning such that its operating mode and internal state may be optimally reconfigured in response to both the observations of the RF environment and the communication needs of the user..." [25].

In order to conform to this definition, the cognitive radio architecture must possess a sort of *cognitive engine* performing the necessary signal processing functionality to support the learning operations and the decision-making. Adhering to the same categorical decomposition as in [25, Ch 2 Sec 5], the signal processing challenges of cognitive radio are grouped as follows:

Wideband operation As evident from the measurement campaigns referenced in Appendix B, spectrum opportunities present themselves across a wide, non-contiguous region of frequency bands. This presents the CR with potential opportunity in communication channels which may be separated by orders of magnitude in frequency. The wide spectrum range leads to the challenge of how to effectively scan different bands and channels for openings and opportunities. Physical challenges present themselves in the form of noise floors varying by location, unknown noise distribution, and complex wireless channel fading models. Signal processing algorithms must be robust against the inherent stochastic nature of varying RF environments.

From a hardware standpoint, there is a trade-off between sensing efficiency and hardware complexity. Consider an expensive wideband antenna. For non-contiguous spectrum opportunities in different bands, the gain and performance of the antenna may not be enough for adequate sensing of both regions. However, a lower-cost reconfigurable antenna would allow for modes of operation in which multiple bands could be sensed. This presents a trade-off in the form of a longer sensing time and energy cost overhead from the reconfiguration of the radio to sense different bands.

Thus, signal processing algorithms for knowledge acquisition in wideband spectrum must strike a balance between robustness to the environmental state, and efficiency in scanning the entire spectrum.

Partially observable environments Factors such as channel noise, hardware noise, power restrictions, sensing errors, and limited channel sensing capabilities only contribute towards partial and imperfect observation of the true state of the RF environment. With incomplete state knowledge, the CR is left with the task optimal decision-making and learning with only partial observations. While solution methods for partially observable environments do exist, they are significantly more complex than methods for environments with complete state information available.

Distributed Multi-agent environments We have so far talked about cognitive radio in the singular, but real radio networks involve multiple users coexisting across the wireless medium. The users may be a mixture of CRs and licensed users, all a part of different legacy (licensed) and non-legacy systems, all fighting for the same limited spectrum. There may or may not exist a common control channel (CCC) or data fusion center linking user's knowledge bases together. Thus, the multi-agent communication problem may also be a centralized or decentralized communication problem with cooperative or non-cooperative operation by users in each respective system. Distributed agents are yet another variable contributing towards partial information of the RF environment as well as an obstacle towards learning and decision-making for the CR, and must be considered during signal processing.

Autonomous operation A distributed agent, or CR, must support independent and autonomous operation. Learning and decision-making on a CR platform ideally adapts to the RF environment it is present in. Therefore, prior training with an algorithm on one environment may lead to sub-optimal decision-making in another. The autonomous operation of a CR may also be affected by other agents operating in the same spectrum bands.

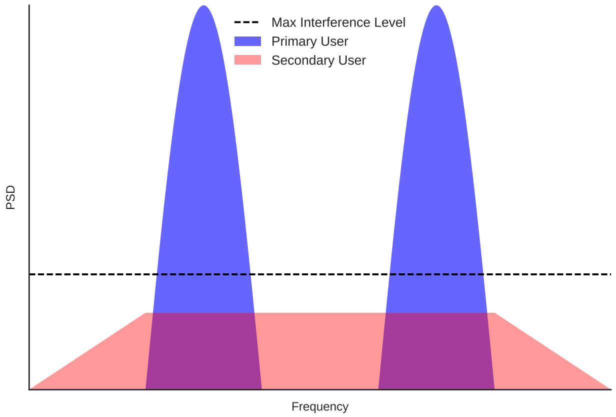
2.3 Dynamic Spectrum Access for Secondary Users

A cognitive radio operating as an unlicensed secondary user (SU) may coexist on the frequency bands of a licensed primary user (PU) provided they have permission to operate on those bands in a particular geographic region. Other agreements may also be in place dictating guidelines and quality of service (QoS) requirements for the PU.

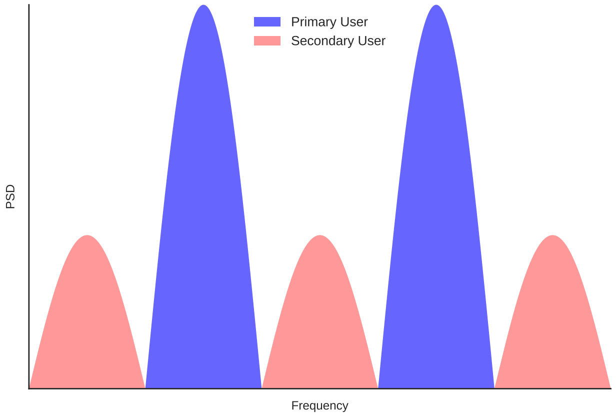
The unlicensed SU must meet the QoS requirements without violating the dictated terms of access.

This type of spectral coexistence is enabled by different radio network paradigms for spectrum allocation and may collectively be called dynamic spectrum sharing (DSS). There are two well studied radio network paradigms used to achieve DSS-based spectrum coexistence: spectrum underlay and spectrum interweave/overlay [25, Ch 3 Sec 1] and [26]. In *spectrum underlay* there is simultaneous use of PU channel by SU with interference below a threshold (that is, not significantly affect the PU's QoS). In *spectrum interweave/overlay* there is opportunistic use of PU channel by SU when PU are not present. The concept of both approaches are graphically depicted in Figure 2.1a and 2.1b.

Following the *spectrum interweave/overlay* approach, the SU searches for *spectrum opportunities* in the form of "holes", or openings occurring when a spectrum channel is unused by a PU. This idea fuels the concept of dynamic spectrum access (DSA), which is a DSS ideology where the burden of managing spectral coexistence between PUs and SUs is placed solely on the SUs [25, Ch 3 Sec 3].



(a) Spectrum Underlay Model. SUs transmit along with PUs but while conforming to power constraints. Typically a max interference threshold is defined which should not be exceeded by SUs such that the PU may maintain a required quality of service.



(b) Spectrum Interweave/Overlay Model. SUs occupy ”temporal holes” or vacant channels in the operating band in-between usage by licensed PUs.

Chapter 3

Dynamic Programming and Reinforcement Learning Concepts

3.1 Introduction to MDPs and Spectrum Prediction

For dynamic spectrum access and autonomous spectrum management, the spectrum prediction problem requires the SU (e.g. a cognitive radio) to decide which spectrum subband or channel to sense at any given time. In order to sense spectrum efficiently and effectively, the radio follows a sensing policy governing the decisions it makes on what to sense. The radio may be governed by a fixed protocol or random guessing, but these policies are not in any way learning from accumulated knowledge. Thus, these policies cannot improve and approach an optimal policy. It arises that the cognitive radio must have some form of reasoning or learning from its spectrum observations to determine the best such policy.

Traditional detection methods simply allow a radio to make reasoned decisions and pick a particular hypothesis based on an observation. Consider the case where a prediction must be made of the true hypothesis for the next time instant $n + 1$ before having an observation at time $n + 1$. For example: radios may need to predict idle channels during the next time period in order to find a suitable channel for transmission. Cognitive radios may also need to make predictions, or reasoned decisions, about future states of the RF environment in order to select the best action. This becomes important if the value of taking an action at the present significantly affects the reward gained by the radio in the future. As an example of this, one may consider a learnable trend of a PU's channel access scheme. If by looking ahead multiple time steps a radio determines a high probability of PU interference, the user may optimally search for a new channel or subband to transmit in thus securing the reward of continued transmission with minimal interference. The selection scheme of a particular action based on reasoned decisions is called a decision rule.

A decision rule may be likened to an action taken upon an observation, or the perceived state of an environment. The prediction of the CR for the next RF environmental state may be determined by the last sensing observation made. Of course, there is nothing limiting the radio from making predictions beyond the next time step. How myopic (short-sighted) or far-sighted the predictions are is a controllable parameter. Developing decision rules such that there are specific responses (actions) taken for each of the possible observed states results in the creation of a policy, denoted as π.

Policies may act as the set of rules a SU may use to make decisions in an RF environment depending upon the observed state of the environment. The states of the environment evolve over time, and linking this evolution of state with sets of actions and rewards results in a decision process. Of primary interest are decision processes dependent solely on the previous state. In other words they exhibit the Markov property. They are respectively called Markov Decision Processes (MDPs).

3.2 Markov Decision Processes

A Markov decision process (MDP) may be defined as a tuple $< \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma >$ [3]. MDP models have found application in several engineering fields, including signal processing, AI, telecommunications, and economics [27]. It may be used to model the RF environment in which spectrum agents learn decision policies for efficient operation. The MDP tuple may be elaborated in the context of discrete spectrum agents:

$\mathcal{S}$ is a finite set of RF environment states (i.e. PU channel occupancy).

$\mathcal{A}$ is a finite set of actions which may be taken by the spectrum agents (i.e SU's making channel access decisions) interacting with the MDP modelled environment. The actions an agent takes may or may not influence the future states of the environment[1].

$\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto [0, 1]$ is a state transition probability matrix, or just the transition matrix. Given the current state s and action choice a, then $\mathcal{P}^a_{ss'} \triangleq Pr(s'|s, a)$ specifies the probability that the next state s' is reached. Note that $\sum_{s' \in \mathcal{S}} P^a_{ss'} = 1$.

$\mathcal{R} : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ is a reward function denoting the expected reward received if an action is taken while in a certain state. Given the current state s and action choice a, then $\mathcal{R}^a_s \triangleq r(s, a)$ is a function which gives the associated expected reward. Alternatively, as drawn in Figure 1.2, the reward may not be independent of the next environment state observed after an action is taken. $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto \mathbb{R}$. The expected reward function is then $\mathcal{R}^a_{ss'} \triangleq r(s, a, s)$.

$\gamma \in [0, 1)$ represents a discount variable which facilitates the tradeoff between short-term (myopic) and long-term rewards. A reward obtained n steps into the future, due to its inherent uncertainty, would thus be attenuated by the weight γ^n. A large γ values immediate rewards more than rewards far into the future. This becomes relevant later when the concept of total discounted reward is introduced.

In an MDP with state space $\mathcal{S}$, the state is in essence a sufficient statistic of the history, or past states of the process, and the next state of the MDP is dependent only on current state as per the Markov property. A Markov decision process possesses the Markov property:

$$\begin{aligned} p(s', r|s, a) &\triangleq \Pr(S_{n+1} = s', R_{n+1} = r|S_n = s; A_n = a) \\ &= \Pr(S_{n+1}, R_{n+1}|S_n, \ldots, S_1; A_n, \ldots, A_1) \end{aligned} \tag{3.1}$$

Thus the MDP is fully characterized by a joint state transition and reward distribution. With this, the complete dynamics of the underlying MDP are captured [1]. In addition, the following marginal distributions for state and reward transitions follow directly:

$$\begin{aligned} \mathcal{P}^a_{ss'} \triangleq p(s'|s, a) &\triangleq \Pr(S_{n+1} = s'|S_n = s; A_n = a) \\ &= \sum_{r \in \mathcal{R}} p(s', r|s, a) \end{aligned} \tag{3.2}$$

[1] A spectrum prediction does not interfere with the environment. Such an action by an agent would not alter or otherwise corrupt the underlying MDP controlling the possible future states. However, spectrum access influences the future state in an MDP modelling PU and SU occupancy. For a stationary MDP modelling PU's only, a SU occupancy would introduce noise and environment observations would deviate from baseline statistics. For multiple SUs observing the environment, it would appear as nonstationary due to the competitive SU interference. This is a significant challenge to multi-agent spectrum learning and management.

$$
\begin{aligned}
p(r|s,a) &\triangleq \Pr(R_{n+1} = r|S_n = s; A_n = a) \\
&= \sum_{s' \in \mathcal{S}} p(s', r|s,a)
\end{aligned} \tag{3.3}
$$

We may also construct functions for the expected reward for state-action pairs[2]:

$$
\begin{aligned}
\mathcal{R}_s^a \triangleq r(s,a) &\triangleq E\left\{R_{n+1}|S_n = s, A_n = a\right\} \\
&= \sum_{r \in \mathcal{R}} r \sum_{s' \in \mathcal{S}} p(s', r|s,a)
\end{aligned} \tag{3.4}
$$

The expected reward for state-action state triples follows from application of the chain rule:

$$
\begin{aligned}
p(s', r|s,a) &= \frac{p(s', r, s, a)}{p(s,a)} \\
&= \frac{p(r|s,a,s')p(s,a,s')}{p(s,a)} \\
&= p(r|s,a,s')p(s'|s,a)
\end{aligned}
$$

Then the expected reward function may be constructed:

$$
\begin{aligned}
\mathcal{R}_{ss'}^a \triangleq r(s,a,s') &\triangleq E\left\{R_{n+1}|S_n = s, A_n = a, S_{n+1} = s'\right\} \\
&= \sum_{r \in \mathcal{R}} r \frac{p(s', r|s,a)}{p(s'|s,a)} \\
&= \sum_{r \in \mathcal{R}} r \frac{p(s', r|s,a)}{\mathcal{P}_{ss'}^a} \qquad (3.5)
\end{aligned}
$$

We can relate the two reward function by multiplying both sides by $\mathcal{P}_{ss'}^a$ and summing over $s' \in \mathcal{S}$:

$$\begin{aligned}
\mathcal{P}^a_{ss'}\mathcal{R}^a_{ss'} &= \sum_{r\in\mathcal{R}} rp(s',r|s,a) \\
\sum_{s'\in\mathcal{S}} \mathcal{P}^a_{ss'}\mathcal{R}^a_{ss'} &= \sum_{s'\in\mathcal{S}}\sum_{r\in\mathcal{R}} rp(s',r|s,a) \\
&= \mathcal{R}^a_s \\
\Rightarrow \mathcal{R}^a_s &= \sum_{s'\in\mathcal{S}} \mathcal{P}^a_{ss'}\mathcal{R}^a_{ss'} \qquad (3.6)
\end{aligned}$$

In the case of a stationary Markov process, the transition probabilities are time invariant. That is, after each time step (state progression) the underlying state transition probabilities remain the same as in previous time step. In the following, stationary MDPs shall be considered.

3.2.1 Finite and Infinite Horizon MDP's and Decision Policies

Agents interacting with a MDP through actions may select their actions based upon a decision policy. A policy $\pi : \mathcal{S} \times \mathcal{A}$ is a mapping which specifies an action $a \in \mathcal{A}$ to be selected for each current state $s \in \mathcal{S}$. The policy is denoted by π or $\pi(a|s) = Pr(a_n = a|s_n = s)$ at time n. For a particular state s, this policy provides a PDF over the actions such that $\sum_{a\in\mathcal{A}} \pi(a|s) = 1$.

This policy may be deterministic or updated via some criteria or learning algorithm. In an agent, the goal is to find an optimal policy that maximizes some reward criteria such as the total reward obtained over a finite time horizon or a total discounted reward obtained over an infinite time horizon. For a stationary MDP, the unchanging transition probabilities of the MDP allow for a stationary policy to govern action choices given state. That is, an optimal decision policy does exists to maximize some reward criteria and may be found.

MDPs may or may not have a terminating state. For MDPs with terminating states (a final state from which it will never exit), one may consider finite-horizon policies. A finite-horizon policy with finite time horizon T specifies decision rules π_n specifying actions to be taken depending on possible states for time steps $n =$

[2] If we know the full characterization of the MDP in terms of $p(s', r|s, a)$ then then the expected reward functions may be constructed from that. It is often typical in constructing an MDP model that expected rewards are assigned to specific pairs (s, a) or triples (s, a, s') in which case $p(s', r|s, a)$ may be constructed should it be desired.

$0, \dots, T-1$. Optimizing a finite-horizon policy $\pi = (\pi_0, \dots, \pi_{T-1})$ means finding decision rules resulting in the greatest value over a finite period of time T.

In contrast, a infinite-horizon policy $\pi = (\pi_0, \pi_1, \dots)$ has time horizon $T \to \infty$, where decisions have to be made for time steps $n = 0, 1, \dots$ with an unclear or nonexistent terminating time. MDPs with unclear or nonexistent terminating time are called infinite-horizon MDPs.

3.2.2 Total Discounted Reward

The act of a SU selecting an action to perform in the current time slot n is determined by the current policy being followed. This action results in a reward at the next time slot R_{n+1} for the SU determinable at the time of next environmental observation. An example of a reward may be positive reinforcement for starting transmission on an unoccupied channel or negative reinforcement for collision with another transmitting user. The *total discounted reward* G_n associated with a stationary infinite-horizon policy π is the total discounted accumulated reward for that policy:

$$G_n^\pi = \sum_{k=0}^{\infty} \gamma^k R_{n+k+1} \tag{3.7}$$

Due to the infinite summation, the discount factor $\gamma \in [0, 1)$ is included as a reward weight and is necessary for convergence of the summation. The discount may also be interpreted as an exponential weight used to depreciate the value of future rewards due to their uncertainty.

G_n^π is a random quantity, so one may consider the *expected total discounted return* (ETDR):

$$\overline{G_n^\pi} = \mathbb{E}_\pi\{G_n^\pi\} = \mathbb{E}_\pi \left\{ \sum_{k=0}^{\infty} \gamma^k R_{n+k+1} \right\} \tag{3.8}$$

3.2.3 Value Function, Q Function, and Solving for Optimal Policies

An important factor in choosing a decision policy for an MDP is understanding the value of being in a particular state of the environment. The current state $S_n = s \in \mathcal{S}$

from which a decision policy begins is also a random variable. Therefore it is useful to define a *state-value function.*

The value of state for a particular policy π can be defined as the total discounted reward given current state. Then, the state-value function of a policy π, is the ETDR given current state. For stationary infinite-horizon policies, this is defined as:

$$V^{\pi}(s) = \mathbb{E}_{\pi}\left\{G_n^{\pi}|S_n = s\right\} = \mathbb{E}_{\pi}\left\{\sum_{k=0}^{\infty}\gamma^k R_{n+k+1}\middle|S_n = s\right\} \tag{3.9}$$

Using the newly defined state-value function (henceforth just called the value function), and applying the law of iterated expectations [28], the ETDR from (3.8) may be rewritten as the average value over all initial states, or:

$$\begin{aligned}\overline{G_n^{\pi}} &= \mathbb{E}_{S_n}\Big\{\mathbb{E}_{\pi}\{G_n^{\pi}|S_n = s\}\Big\}\\ &= \sum_{s\in\mathcal{S}} V^{\pi}(s)Pr(S_n = s)\end{aligned} \tag{3.10}$$

From (3.10), one can infer that for a stationary MDP choosing a policy to maximize ETDR $\overline{R_n^{\pi}}$ is the same as choosing a policy to maximize the value function $V^{\pi}(s)$. For the sake of optimality, it is desirable to select a policy which maximizes the value function $V^{\pi}(s)$.

An important observation to make is that the value function $V^{\pi}(s)$ can be decomposed into two parts: an immediate reward, and a discounted value of next state, leading to a recursive definition. This is constructed as follows:

$$\begin{aligned}V^{\pi}(s) &= \mathbb{E}_{\pi}\left\{G_n^{\pi}\middle|S_n = s\right\}\\ &= \mathbb{E}_{\pi}\left\{R_{n+1} + \gamma G_{n+1}^{\pi}\middle|S_n = s\right\}\\ &= \mathbb{E}_{\pi}\{R_{n+1}|S_n = s\} + \gamma\mathbb{E}_{\pi}\left\{G_{n+1}^{\pi}\middle|S_n = s\right\}\end{aligned} \tag{3.11}$$

We can expand the first term of (3.11) into

$$
\begin{aligned}
\mathbb{E}_\pi\{R_{n+1}|S_n = s\} &= \sum_a \pi(a|s)\mathcal{R}_s^a \\
&= \sum_a \pi(a|s) \sum_{s',r} rp(s',r|s,a) \qquad (3.12)
\end{aligned}
$$

We can expand the second term of (3.11) into

$$
\begin{aligned}
\gamma\mathbb{E}_\pi\left\{G_{n+1}^\pi \middle| S_n = s\right\} &= \gamma \sum_{s',r,a} \pi(a|s)p(s',r|s,a)\mathbb{E}_\pi\left\{G_{n+1}^\pi \middle| S_n = s'\right\} \\
&= \gamma \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)\mathbb{E}_\pi\left\{G_{n+1}^\pi \middle| S_n = s'\right\} \\
&= \gamma \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)V^\pi(s') \qquad (3.13)
\end{aligned}
$$

Using (3.12) and (3.13) we get the recursive definition of the value function for all $s \in \mathcal{S}$:

$$
\begin{aligned}
V^\pi(s) &= \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)\big[r + \gamma V^\pi(s')\big] \\
&= \sum_a \pi(a|s) \left[\mathcal{R}_s^a + \gamma \sum_{s',r} p(s',r|s,a)V^\pi(s')\right] \\
&= \sum_a \pi(a|s) \left[\mathcal{R}_s^a + \gamma \sum_{s'} \mathcal{P}_{ss'}^a V^\pi(s')\right] \qquad (3.14) \\
&= \sum_a \pi(a|s) \left[\sum_{s'} \mathcal{P}_{ss'}^a \big(\mathcal{R}_{ss'}^a + \gamma V^\pi(s')\big)\right] \qquad (3.15)
\end{aligned}
$$

The equation in (3.14) is known as the Bellman equation, and (3.15) results from (3.6).

At this time, it is convenient to also introduce another function called the *action-value function* $Q^\pi(s,a)$ (henceforth just called the Q-function), which may be defined as the total discounted reward starting from initial state s, taking action a and then following policy π.

$$Q^\pi(s,a) = \mathbb{E}_\pi\left\{G_n^\pi \middle| S_n = s, a_n = a\right\} \tag{3.16}$$

Similarly derived as with the value function, the Q-function also has a Bellman equation representation:

$$\begin{aligned} Q^\pi(s,a) &= \mathbb{E}_\pi\left\{R_{n+1} + \gamma G_{n+1}^\pi \middle| S_n = s, A_n = a\right\} \\ &= \mathcal{R}_s^a + \gamma \sum_{s'} \mathcal{P}_{ss'}^a V^\pi(s') && (3.17) \\ &= \sum_{s'} \mathcal{P}_{ss'}^a \left[\mathcal{R}_{ss'}^a + \gamma V^\pi(s')\right] && (3.18) \end{aligned}$$

The Bellman equations may be defined in terms of each other. Using (3.17) in (3.14) we get the relationship:

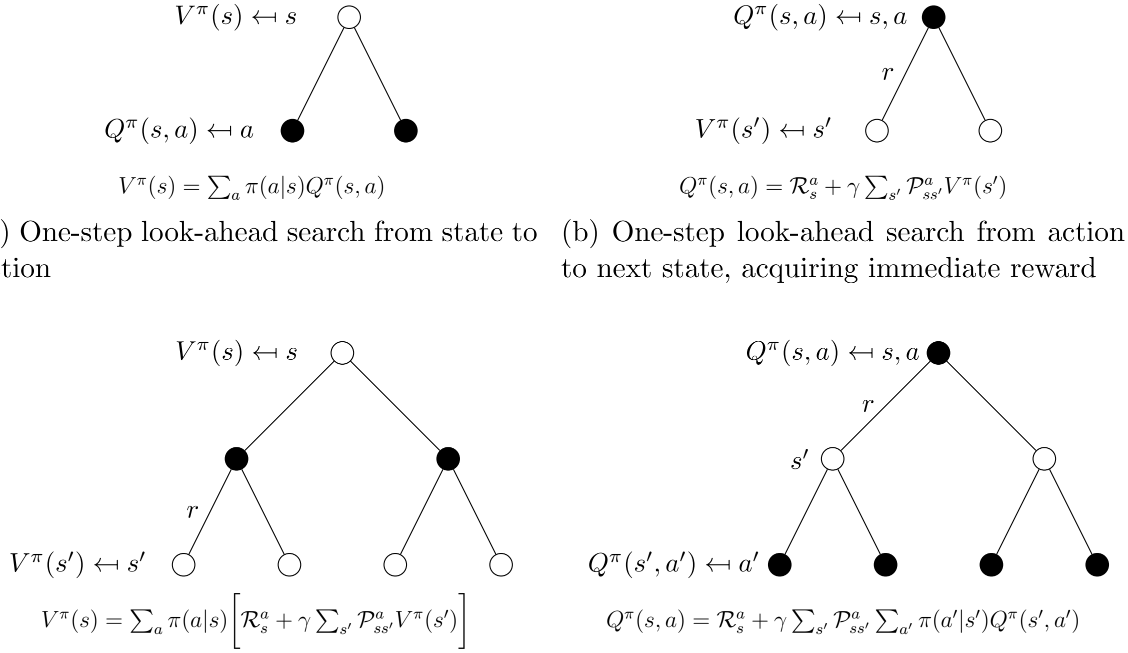
$$\begin{aligned} V^\pi(s) &= \sum_a \pi(a|s)\left[\mathcal{R}_s^a + \gamma \sum_{s'} \mathcal{P}_{ss'}^a V^\pi(s')\right] \\ &= \sum_a \pi(a|s) Q^\pi(s,a) && (3.19) \end{aligned}$$

and likewise:

$$\begin{aligned} Q^\pi(s,a) &= \mathcal{R}_s^a + \gamma \sum_{s'} \mathcal{P}_{ss'}^a V^\pi(s') \\ &= \mathcal{R}_s^a + \gamma \sum_{s'} \mathcal{P}_{ss'}^a \sum_{a'} \pi(a'|s') Q^\pi(s',a') && (3.20) \\ &= \sum_{s'} \mathcal{P}_{ss'}^a \left[\mathcal{R}_{ss'}^a + \gamma \sum_{a'} \pi(a'|s') Q^\pi(s',a')\right] && (3.21) \end{aligned}$$

The relationships shown in (3.14), (3.17), (3.19) and (3.20) between V^π and Q^π may be visualized using look ahead search diagrams. This is done in Figure 3.1,

The Q-function requires the selection of an action from the initial state and finds use in application of solving for optimal policies. Finding the action that maximizes


$V^\pi(s) \hookleftarrow s$
$Q^\pi(s,a) \hookleftarrow a$
$V^\pi(s) = \sum_a \pi(a|s) Q^\pi(s,a)$
$Q^\pi(s,a) \hookleftarrow s, a$
r
$V^\pi(s') \hookleftarrow s'$
$Q^\pi(s,a) = \mathcal{R}_s^a + \gamma \sum_{s'} \mathcal{P}_{ss'}^a V^\pi(s')$
$V^\pi(s) \hookleftarrow s$
r
$V^\pi(s') \hookleftarrow s'$
$V^\pi(s) = \sum_a \pi(a|s) \left[\mathcal{R}_s^a + \gamma \sum_{s'} \mathcal{P}_{ss'}^a V^\pi(s') \right]$
$Q^\pi(s,a) \hookleftarrow s, a$
r
s'
$Q^\pi(s',a') \hookleftarrow a'$
$Q^\pi(s,a) = \mathcal{R}_s^a + \gamma \sum_{s'} \mathcal{P}_{ss'}^a \sum_{a'} \pi(a'|s') Q^\pi(s',a')$

(a) One-step look-ahead search from state to action

(b) One-step look-ahead search from action to next state, acquiring immediate reward

(c) Two-step look-ahead search from state to next state

(d) Two-step look-ahead search from action to next action

Figure 3.1: Look-ahead search graphs for visualizing the relationship between the value function V^π and the action-value function Q^π

the expected reward returned from the Q-function results in the optimal action to be taken. This leads to the optimal action that can be taken from a particular state in a decision policy. Therefore the *policy update* rule is:

$$\pi'(s) = \arg\max_{a \in \mathcal{A}} Q^\pi(s,a) = \arg\max_{a \in \mathcal{A}} \left[\mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a V^\pi(s') \right] \tag{3.22}$$

The value of the new policy π' must be better or as good as the old policy π for all states $s \in \mathcal{S}$.

With a stationary decision policy π it is apparent that $Q^\pi(s, \pi(s))$ and $V^\pi(s)$ are equivalent. Using this relation, a *value update* rule may be defined in terms of the Q-function,

$$V^{\pi'}(s) = Q^\pi(s, \pi'(s)) = \max_{a \in \mathcal{A}} Q^\pi(s,a) = \max_{a \in \mathcal{A}} \left[\mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a V^\pi(s') \right] \tag{3.23}$$

The policy update rule in (3.22) along with the value update rule in (3.23) find there usage in several iterative algorithms used for the computation of optimal stationary policies π^* and likewise the optimal value function $V^*(s)$. It has been shown that an optimal stationary policy π exists for a stationary MDP with bounded rewards and finite states and actions [3]. Two common algorithms are *policy iteration* and *value iteration*. These algorithms require full knowledge of the MDP model parameters such as the transition probabilities $\mathcal{P}^a_{ss'}$. Further information regarding both may be found in other sources such as [1, 2] and other references in dynamic programming. The policy update rule plays a fundamental role in reinforcement learning methods such as *Q-learning*.

3.3 Extension to Partially Observable MDPs

When working with MDPs, complete knowledge of the current state is completely observable; that is there is no ambiguity in state and all elements pertaining to the state are learned upon observation. In the case where perfect or complete observations cannot be made of the MDP, classic MDP solution methods are no longer optimal and are not guaranteed to be well performing. An MDP in which state is not directly observable is called a *partially observable Markov decision process* (POMDP). POMDP models are suited to situations where decisions must be made while there is uncertainty regarding the true state.

A POMDP is a tuple $< \mathcal{S}, \mathcal{A}, \mathcal{Y}, \mathcal{P}, \mathcal{R}, \mathcal{O}, \gamma, b_0 >$ where:

- $\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma$ are as defined for a regular MDP
- $\mathcal{Y}$ is a finite set of observations
- $\mathcal{O}$ is an observation function, denoted as $\mathcal{O}^a_{s'z} = Pr(z|s', a)$, specifying the probability of observing observation z given action a was taken resulting in state s'.
- b_0 is an initial belief state of the environment, specifying a probability distribution over the initial state of the environment.

3.3.1 Belief State

In a POMDP, the observations made have incomplete information regarding the environment state. POMDPs may be converted to an equivalent completely observable MDP through the use of *belief states*. Belief state is the aposteriori probability of the current state at time n of the environment given the complete history h_n of observations, actions, and associated rewards. Formally this may be written out as

$$b_n(s) = Pr(S_n = s|h_n). \tag{3.24}$$

It has been shown [29] that the belief state is a sufficient statistic for optimal decision making in a POMDP. That is, an optimal estimate of the current environment state may be formed using only the current belief state. Therefore, instead of taking the whole history into account when determining the belief state at the next time step $n+1$, all that is necessary is knowledge of the current belief state at time step n. This implies that an update in belief state is completely determinable provided only the previous action a taken and the most recent observation z. Hence, (3.24) may be redefined as,

$$b_n(s) = Pr(S_n = s|b_{n-1}, a_{n-1}, z_n) \tag{3.25}$$

A belief state vector $\mathbf{b}$ may be defined as containing all beliefs $b(s)$ for $s \in \mathcal{S}$. Applying Bayes' Theorem, it follows that the updated belief of state s' is:

$$\begin{aligned} b'(s') &= Pr(s'|b, a, z) \\ &= \frac{Pr(z|s', b, a)Pr(s'|a, b)}{Pr(z|a, b)} \\ &= \frac{Pr(z|s', a)\sum_{s\in\mathcal{S}} Pr(s'|b, a, s)Pr(s|b, a)}{Pr(z|a, b)} \\ &= \frac{\mathcal{O}^a_{s'z}\sum_{s\in\mathcal{S}} \mathcal{P}^a_{ss'}b(s)}{Pr(z|a, b)} \\ &= \eta\mathcal{O}^a_{s'z}\sum_{s\in\mathcal{S}} \mathcal{P}^a_{ss'}b(s) \end{aligned} \tag{3.26}$$

where $b'(s')$ is the updated belief state for s' and b is the current belief state vector. There is also the normalization constant $\eta = \frac{1}{Pr(z|a,b)}$ ensuring $\mathbf{b}'$ is a probability distribution. where $Pr(z|a, b) = \sum_{s'\in\mathcal{S}} .\mathcal{O}^a_{s'z}\sum_{s\in\mathcal{S}} \mathcal{P}^a_{ss'}b(s)$ is the probability of making observation z given the current belief b and action taken a. It may model sensor noise and it may influence the probability of detection and false alarm. Assuming perfect sensing, the probability of what we observe representing truth state (or otherwise partial state) is 1, and so $\mathcal{O}^a_{s'z}$ may be eliminated. Then the simplified belief state update may be rewritten as:

$$b'(s') = \eta\sum_{s\in\mathcal{S}} \mathcal{P}^a_{ss'}b(s) \tag{3.27}$$

Applying the belief state, the POMDP may be treated as a completely observable MDP.

3.4 Reinforcement Learning

Reinforcement learning (RL) is a learning method which incorporates rewards from an environment into an update rule that modifies an agents behavior as it interacts with the environment [1]. This method of learning may be used by agents to learn autonomously without supervision, learning only from it's observations of the environment state. RL methods apply to MDPs and are able to learn optimal decision policies π that maximize the total expected reward. Optimal solutions to MDPs may be found using methods based on dynamic programming [3] such as value iteration, which assumes stationary rewards $\mathcal{R}_s^a$ and transition probabilities $\mathcal{P}_{ss'}^a$. It also requires complete knowledge of the transition probabilities to be used. This is not always the case practically, and in the situation of sensing real environments, the transition probabilities may be considered unknown apriori. However, RL algorithms find solutions to the MDP problem without knowledge of the transition probabilities

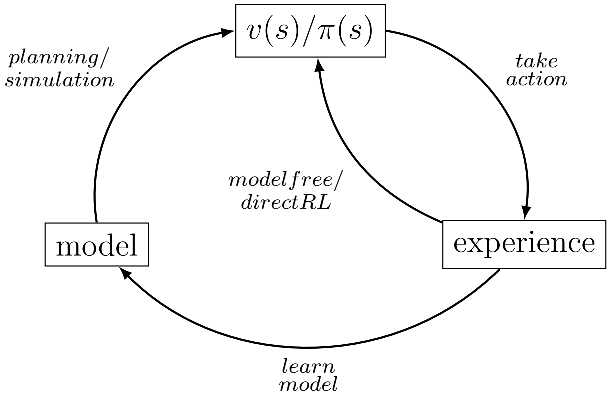
Assuming observations are perfect and complete (all state information is relayed), and the underlying process of the environment is Markov, it has been shown that RL methods such as Q-learning [6] converge over time to the optimal decision policy for that environment. The optimality of the learned policy is limited by the stationarity and complete observability of the underlying MDP model describing the environment. However, it has also been shown that satisfactory, near-optimal decision policies may be obtained for distributed partially observable MDPs [30].

3.4.1 Model-based Vs. Model-free Learning

In dynamic programming, full knowledge of an underlying MDP parameters must be known apriori. That is a reliable model of the MDP has been constructed. This model may be truth, as in a fully characterized MDP with known transition probabilities. It may also be a black box approximation incorporating all that is known about the environment with as little or as much bias as desired.

With model-based learning, an initial model is used for planning and simulation of an environment from which an estimate of state/action values or decision policies may be determined. Actions are selected and from an agent's experience, the model is updated, and the cycle continues until the model converges to truth. Such methods of indirect learning through modeling and simulating an environment are called planning

Figure 3.2: Model Vs. Model-free Learning



methods, of which policy and value iteration are both examples. Hence they both also fall into the category of model-based learning.

In contrast, model-free learning updates the state/action values or decision policies directly from experience. Otherwise it follows a similar cycle as model-based learning as depicted in Figure 3.2. Q-learning falls into the category of model-free learning, as the model (e.g. state transition probabilities) are generally unknown and the action values (Q-values) are updated via a Q-learning update rule.

3.4.2 Q-Learning

In the Q-learning algorithm, a table of Q-values is maintained. Each Q-value $Q(s, a)$ represents a value assigned to taking a particular action a when in a given state s. It is analogous to the state-value function used in policy iteration for the dynamic programming solution to MDPs. However, unlike policy and value iteration, Q-learning does not require knowledge of the state transition probabilities. The Q-learning algorithm updates the Q-values based on actions and rewards resulting from the actions without any knowledge of the underlying Markov process parameters. The strength of Q-learning is it may be used by an agent to find an optimal decision policy for an environment without complete knowledge of the transition probabilities.

The Q-learning algorithm is as follows [25]: If as a result of taking action $a_{n-1} = a$ when in state $S_{n-1} = s$ at time $n - 1$, the learner observed the delayed reward r and the state transitioned to a new state $S_n = s'$, then the Q value corresponding to the state-action pair (s, a) is updated with the following update rule:

$$Q(s, a) \mapsfrom Q(s, a) + \Delta Q(s, a) \tag{3.28}$$

where

$$\Delta Q(s,a) = \alpha \left[r + \gamma \max_{a' \in \mathcal{A}} Q(s',a') - Q(s,a) \right] \tag{3.29}$$

where $\alpha \in [0,1]$ denotes the learning rate and $\gamma \in [0,1]$ denotes a discount factor.

3.4.3 The Exploitation/Exploration Problem

When learning an optimal decision policy π, an agent must avoid getting stuck in local optima. This can occur in Q-learning if there is inadequate state/action exploration. Q-learning is known to converge to an optimal policy for an MDP when all pairs (s,a) continue to be visited and updated[1]. If a greedy decision policy is chosen, then only the action of most value from a given state will be taken. While this might fully exploit the current information contained in the Q-table, it does not account for the possibility of better actions being determined once more information is gathered through experience. .

In order to converge to an optimal decision policy, it is in an agent's best interest to gather enough information to make the overall best decisions. Balancing exploration vs. exploitation to maximize cumulative reward and thus learn globally optimal decision policies is in essence the exploitation/exploration (EE) problem.

In a greedy, full-exploitation decision policy, an agent always takes the action of most value in a current state. An agent may introduce a form of naive exploration by adding "noise" to the greedy policy in the form of a probability ϵ of taking a random exploratory action. Such an approach, called an ϵ-greedy policy, may be taken where the actions can be selected according to the decision rule:

$$a^* = \begin{cases} \arg\max_{a \in \mathcal{A}} Q(s,a) & \text{with probability } 1-\epsilon \\ \sim P(\mathcal{A}) & \text{with probability } \epsilon \end{cases} \tag{3.30}$$

where $P(\mathcal{A})$ denotes a suitable probability distribution over the set of actions $\mathcal{A}$ and $\epsilon \in [0,1]$ is the probability of taking the exploratory action. The choice for distribution $P(A)$ is unlimited, though the uniform distribution $U(A)$ is typical.

In the initial states of learning, it is typically desirable to have a high level of exploration, a commonly adopted scheme is to vary ϵ over time, progressively decreasing exploration as more is learned about the environment and optimal Q-values are discovered.

With Q-learning in particular, optimistic initialization of the action-values to high value is another technique for promoting early exploration of (s, a) pairs. It assumes that actions are best initially until proven otherwise. It is important to note however that this does not guarantee that the optimal decision policy will be found or converged to.

The EE problem itself is tied intimately to the solutions of classical multi-armed bandit (MAB) problems as discussed later in the chapter.

3.4.4 Modifying Q-Learning for POMDPs

In a POMDP, environment state is not completely observable. Q-learning is aimed at learning optimal policies for stationary, completely observable MDPs. However, most real environments can only be modelled as a POMDP. Without knowledge of current state, the Q-learning algorithm must be modified in order to solve for near-optimal solutions in POMDPs.

If the environment state transition probabilities are known, then it is possible to define a completely observable MDP from a POMDP in terms of a belief state vector $\mathbf{b}$. Replicated Q-learning [15, 18, 31, 32] attempts to learn a POMDP in a reinforcement learning setting and generalized Q-learning using the belief vector $\mathbf{b}$. In replicated Q-learning, the update rule is defined as:

$$\Delta_{\mathbf{b}} Q(s, a) = \alpha b(s) \left[r + \gamma \max_{a' \in \mathcal{A}} Q(\mathbf{b}', a') - Q(s, a) \right] \tag{3.31}$$

where

$$Q(\mathbf{b}, a) = \sum_{s \in \mathcal{S}} b(s) Q(s, a) \tag{3.32}$$

is the expected action value for belief state $b(s), \forall s \in \mathcal{S}$. The belief of all states are summarized in the belief state vector $\mathbf{b}$. Recall that $\mathbf{b}$ is a posterior distribution over the beliefs of being in state s, thus $\sum_{s \in \mathcal{S}} b(s) = 1$ In the update rule, $\mathbf{b}'$ is the next belief state vector.

Consider the case of completely observable states (thus reducing the POMDP to an MDP). Then it is evident that $b(s) \in \{0, 1\}$ with 1 for a single belief state and 0 for the rest. Then 3.31 reduces to the standard Q update rule:

$$\Delta Q(s,a) = \alpha \left[r + \gamma \max_{a' \in \mathcal{A}} Q(s', a') - Q(s,a) \right] \tag{3.33}$$

The Replicated Q-learning update may be written as:

$$Q(s,a) \leftarrow Q(s,a) + \Delta_{\mathbf{b}} Q(s,a) \tag{3.34}$$

If the belief state at time n is $\mathbf{b}_n$, then the equivalent decision rule to (3.30) is:

$$a^* = \begin{cases} \arg\max_{a \in \mathcal{A}} Q(\mathbf{b}_n, a) & \text{with probability } 1 - \epsilon \\ \sim P(\mathcal{A}) & \text{with probability } \epsilon \end{cases} \tag{3.35}$$

3.5 Single State MDPs: Multi-armed Bandit Problems

The multi-armed bandit (MAB) problem was originally formulated around 1940 [33]. The classical MAB problem consists of a "bandit" (so-named as an analogy to slot-machines, or the "one-armed bandit") with K arms. The multi-armed bandit itself represents an environment in which an agent may choose one (or in some cases several) of the arms as an action. Associated with each arm is a statistic relating to the maximum reward receivable through pulling of the arm, the analogy to be made being that of a slot machine at a casino. The objective of solving a MAB problem is to discover the decision policy which maximizes the expected reward (i.e. discovering which arm provides the greatest expected reward). Compared to the full reinforcement learning problem, this is a simplified case with a single, preferred action, but it has found its place in the study of spectrum sharing with unknown channel statistics [11, 34, 35, 36, 37].

3.5.1 The MAB Model

The multi-armed bandit can be viewed as a environment allowing a finite set of actions $\mathcal{A} \in \{1, \ldots, L\}$. In the context of the bandit, these actions correspond to bandit arm pulls. Associated with each bandit arm is a reward distribution $(D_1, \ldots, D_L)$ along with associated statistical averages $(\mu_1, \dots, \mu_L)$ and variances $(\sigma_1, \dots, \sigma_2)$. To the agent interacting with the MAB, the statistics μ_i and σ_i for $i \in \mathcal{A}$ are initially unknown. At each time step, the agent interacts with an arm and receives a reward. The goal of the agent is to discover the arm leading to greatest expected reward which is the arm with the largest average and to gain the greatest total reward over the entire period of interaction. Bandit algorithms allow an agent to learn the best decision policy for action selection while interacting with the environment.

3.5.2 MAB Algorithms

Multiple algorithms have been studied for the solution of MAB problems. These algorithms are directly tied to the problem of exploration in the bandit environment. The balance of exploitation and exploration is an important factor in reinforcement learning methods, hence the importance of their treatment and overview. The following subsections review a few select algorithms: ϵ-greedy, Boltzmann exploration, and UCB1. In them empirical averages $\hat{\mu}_i(t)$ are used, representing the mean for the ith arm after t periods. The probability of picking arm i at time t is denoted as $p_i(t)$.

ϵ-greedy Algorithm

In the ϵ-greedy algorithm, the currently optimal arm is selected with probability $1-\epsilon$ with probability of random arm selection ϵ. In other words:

$$p_i(t+1) = \begin{cases} 1-\epsilon & \text{if } \arg\max_{l=1,\dots,L} \hat{\mu}_j(t) \\ \epsilon & \text{otherwise} \end{cases} \tag{3.36}$$

Boltzmann (Softmax) Exploration

Softmax methods pick each each action with a probability that is propotional to its average reward. Boltzmann exploration is based on the principle that the frequency of bandit arm pulls for each arm should be proportional to it's average reward [38]. Boltzmann exploration as proposed in [1] for reinforcement learning problem is a softmax method which chooses actions using the Boltzmann distribution, thus:

$$p_i(t+1) = \frac{e^{\frac{\hat{\mu}_i(t)}{\tau}}}{\sum_{l=1}^{L} e^{\frac{\hat{\mu}_l(t)}{\tau}}}, i = 1, \dots, n \tag{3.37}$$

where τ is a temperature parameter which controls the randomness of the selection.

The Upper Confidence Bound

In [39], upper confidence bound (UCB) algorithm was proposed, in particular the popular UCB1 algorithm is described. The UCB1 algorithm solves the MAB problem optimally up to a constant factor [38]. The UCB1 algorithm maintians the number of times that each action has been taken, denoted by $n_i(t)$ along with the empirical means $\hat{\mu}_i(t)$. Initially each action is made once, then on period t the algorithm picks the arm $l(t)$ following:

$$l(t) = \underset{i=1,\ldots,L}{\arg\max} \left(\hat{\mu}_i(t) + \sqrt{\frac{2\ln(t)}{n_i}} \right) \tag{3.38}$$

3.5.3 Restless MABs and Belief Updates

Restless multi-armed bandits (MABs) generalizes the classical bandit problem, which is limited to interaction with a single arm at a time [22]. In the restless problem, multiple arms may be played at a time with the user receiving reward. In addition, the remaining arms are “restless” in that they may also change state and give reward [20].

Given that the arms are independent and the arm state transition probabilities are known, an optimal solution involving an indexing policy was proposed by Whittle in [20]. Of more relevant interest, a method involving belief updates was applied to a single agent spectrum management problem modelled as a restless MAB in [21]. The PU occupancies on the channels were modeled using independent two-state Markov processes with known channel state transition probabilities. Belief states are used as a sufficient statistic [29, 40] of the state of unobserved arms and the belief is updated following the simplified rule given in (3.27).

An alternative view of this problem is as a POMDP with independent channels [16, 19, 22]. In the following chapter this view will be applied.

Chapter 4

Modeling the Spectrum Environment

4.1 MDP Channel and Subband Models

4.1.1 Single-chanel MDP

A signal channel may be modelled by a Markov process using the state diagram shown in Figure 4.1. The channel has two possible states: $busy(1)$, representing occupancy by a PU, and $idle(0)$, representing vacancy. The channel model has transition probabilities independent of actions taken, thus $p_n(s'|s,a) = p_n(s'|s)$. This is valid for both spectrum prediction and access if we consider the PU access independent of SU actions as would be the case following the spectrum overlay paradigm of Figure 2.1b. With state transition probabilities α and β, the channel state transition matrix may be written out as:

$$\mathcal{P} = \mathcal{P}^a_{ss'} = \mathcal{P}_{ss'} = \begin{bmatrix} p_n(0|0) & p_n(0|1) \\ p_n(1|0) & p_n(1|1) \end{bmatrix} = \begin{bmatrix} 1-\alpha & \alpha \\ \beta & 1-\beta \end{bmatrix} \tag{4.1}$$

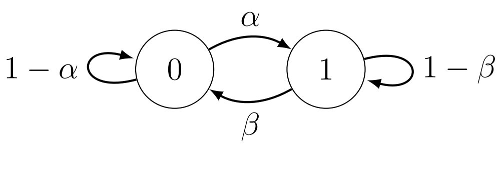


Figure 4.1: Channel MDP model with idle/vacant state 0, busy/occupied state 1, and transition probabilities α and β.

Note that $\mathcal{P}$ in the above case is a 2×2 matrix. In general for a finite state Markov process, $\mathcal{P}$ will be a $N \times N$ matrix with N being the number of states.

On the other hand, the expected rewards depend on the actions an agent interacting with the channel takes and the risk associated with each action.

For the problem of spectrum prediction, actions map to agent guessing the next state of the environment, either vacant or occupied by a PU. An example set of rewards $r(s, a, s')$ is given by the following:

$$\mathcal{R}^a_{s0} = r(s, a, 0) = \begin{bmatrix} \text{reward} & \text{penalty} \\ \text{reward} & \text{penalty} \end{bmatrix} \tag{4.2}$$

$$\mathcal{R}^a_{s1} = r(s, a, 1) = \begin{bmatrix} \text{penalty} & \text{reward} \\ \text{penalty} & \text{reward} \end{bmatrix} \tag{4.3}$$

Here *reward* is a placeholder for a positively reinforcing reward for correct prediction and likewise *penalty* is a negatively reinforcing reward for incorrect prediction. In both (4.2) and (4.3), the rewards are independent of s, so we may define:

$$\mathcal{R}^a_{s'} = r(a, s') = \begin{bmatrix} \text{reward} & \text{penalty} \\ \text{penalty} & \text{reward} \end{bmatrix} \tag{4.4}$$

For the problem of spectrum access, actions map to agent transmission or waiting in the environment with the potential for channel utility with no PU collision or channel impedance with PU collision. An example set of rewards $r(s, a, s')$ is given by the following:

$$\mathcal{R}^a_{s0} = r(s, a, 0) = \begin{bmatrix} \text{null} & \text{reward} \\ \text{null} & \text{reward} \end{bmatrix} \tag{4.5}$$

$$\mathcal{R}^a_{s1} = r(s, a, 1) = \begin{bmatrix} \text{null} & \text{penalty} \\ \text{null} & \text{penalty} \end{bmatrix} \tag{4.6}$$

Similar as before, *reward* is a placeholder for a positively reinforcing reward for access without interference and likewise *penalty* is a negatively reinforcing reward for interference with an occupying PU. Additionally *null* is a placeholder for a desired (possibly 0 valued) neutral reinforcing reward for the decision to not access spectrum in the next time frame. Once again, using these definitions the expected are independent of s. So we may define:

$$\mathcal{R}_{s'}^{a} = r(a, s') = \begin{bmatrix} \text{null} & \text{null} \\ \text{reward} & \text{penalty} \end{bmatrix} \tag{4.7}$$

Using the preceding definitions we may model a single simple spectrum channel as an MDP with it's entire dynamics explicitely defined.

4.1.2 Multi-channel States Space

Consider a subband consisting of several channels. The set of primary channels may be designated as $\mathcal{C} = (1, \dots, L)$. The resultant state space is $\mathcal{S} = \mathcal{S}_1 \times \cdots \times \mathcal{S}_L$. A state $s \in \mathcal{S}$ may be conveniently represented as a L-tuple $s = (s_1, \dots, s_L)$. For L two-state channels, there are $N = 2^L$ possible PU occupancy states the subband may be in. That is $\mathcal{S} = \{0, 1, \dots, 2^L - 1\}$

If we let $\mathcal{P}_i$ be the transition matrix for the channel $i \in \mathcal{C}$ and we consider the transition of state in every channel to be independent of one another, the transition matrix of the subband model may be determined from the individual transition matrices as follows:

$$\mathcal{P} = \mathcal{P}_1 \otimes \cdots \otimes \mathcal{P}_i \otimes \cdots \otimes \mathcal{P}_L \tag{4.8}$$

where $\mathbb{P}_i$ is the transition matrix for channel i and $\otimes$ denotes the Kronecker product. The resulting transition matrix is of dimension $2^L \times 2^L$

The Markov processes describing both channel and subband are non-terminating, that is they do not have a final state. Therefore a MDP constructed using these models will be an infinite-horizon MDP. For an infinite-horizon MDP, stationary reward values $\mathcal{R}_{ss'}^{a} = r(s, a, s')$ are assigned for taking actions $a \in \mathcal{A}$ from states $s \in \mathcal{S}$, resulting in next state $s' \in \mathcal{S}$. For the channel model and subband model, the state space is $\mathcal{S} = \{0, 1\}$ and $\mathcal{S} = \{0, \dots, 2^L - 1\}$ respectively.

4.1.3 Multi-channel Action Spaces

For the application of accessing idle channels within a subband an action space must be defined. There are $L + 1$ possible actions $\mathcal{A} = \{0, 1, \dots, i, \dots, L\}$ for the case of accessing a single channel at a time within a subband, where action $a = 0$ represents accessing no channels (the null action) and $a = i$ represents accessing channel i for

$i \in \mathcal{C}$. For up to k-channels, $\mathcal{A} = \left\{0, 1, \ldots, \sum_{i=0}^{k} \binom{L}{i}\right\}$ and for L-channels $\mathcal{A} = \left\{0, 1, \ldots, 2^L = \sum_{i=0}^{L} \binom{L}{i}\right\}$, where $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ is the binomial coefficient.

4.1.4 Multi-channel Reward Spaces

A subband infinite-horizon MDP may be constructed from an appropriate stationary reward function $r(s, a, s')$ with finite states $s, s \in \mathcal{S}$ and finite actions $a \in \mathcal{A}$. Depending on whether channel prediction, channel access, or some combination thereof is desired, each channel within the subband may be modelled with expected reward $\mathcal{R}_i = r_i(a, s')$ for $i \in \mathcal{C}$ and a a resultant composite reward formed:

$$\mathcal{R} = \sum_{i \in \mathcal{C}} \mathcal{R}_i \tag{4.9}$$

Chapter 5

Spectrum Simulation and Autonomous Distributed Learning with SU Spectrum Agents

5.1 Factored Replicated Q-Learning

An overall spectrum subband model with L channels, represented by the composite state transition matrix given in (4.8) forms a single MDP with 2^L states for an agent to monitor and learn policies from. The convergence of Q-learning requires the visitation and continued exploration of all environment states. Classic tabular reinforcement learning methods begin to exhibit poor performance when operating in large state and/or action spaces.

As evident from the construction of (4.8), the single subband PU spectrum MDP model may be factored into its constituent parts; that is L PU channel MDPs of 2 states (vacant (0) or occupied (1)) each represented by the channel state transition matrices:

$$\mathcal{P}_i = \begin{bmatrix} 1-\alpha_i & \alpha_i \\ \beta_i & 1-\beta_i \end{bmatrix} \tag{5.1}$$

for $i \in L$ and $\alpha_i,\ \beta_i \in [0,1]$ the respective state transition probabilities. Q-learning may now be applied to the factored MDP problem. For the case of partial observability, the channel transition probabilities of (5.1) are presumed to be known. In the case of perfect sensor occupancy decisions for the observed channels, the channel belief vectors $\mathbf{b}_i$ are then updated following the simplified rule of (3.27) such that:

$$\text{(vacant belief) } b_i(0) = \begin{cases} 1 & \text{if } i \text{ observed as vacant} \\ 0 & \text{if } i \text{ observed as occupied} \\ \mathcal{P}_{00} b_i(0) + \mathcal{P}_{10} b_i(1) & \text{if not sensed} \end{cases} \tag{5.2}$$

$$\text{(occupied belief) } b_i(1) = \begin{cases} 0 & \text{if } i \text{ observed as vacant} \\ 1 & \text{if } i \text{ observed as occupied} \\ \mathcal{P}_{01} b_i(0) + \mathcal{P}_{11} b_i(1) & \text{if not sensed} \end{cases} \tag{5.3}$$

This is equivalent to the restless MAB belief update in [29] and is possible due to the independent channel assumption.

Rather than update decisions purely through belief vectors, replicated Q-learning updates are applied to each factored channel i following (3.31), (3.34), and best actions chosen by (3.35). By using Q-tables rather than direct belief decisions, some resiliency towards the non-stationary nature of multi-agent learning in a shared PU spectrum environment is introduced [19]. Also by handling each channel individually as a factored MDP, multiple, simultaneous applications of replicated Q-learning may be applied to each reduced problem facilitating more optimal performance.

5.2 Experimental Setup

The setup emulates that of [19] closely for the MDP model values and reward distribution. However it differs both in how spectrum utilization is computed, the composition of the learning agents, and the scenarios which are simulated. In addition, the time frame in which the spectrum agents exist and operate within the environment differs in that agents sense channel occupancy in the current time frame in order to predict their access decision for the next time frame rather than having a occupancy sensing period at the beginning of each time frame. This is elaborated upon below.

For each channel, the transition probabilities are chosen to be $\alpha = 0.1$ and $\beta = 0.2$. For $i \in \mathcal{C}$, the channel transition matrix is then:

$$\mathcal{P}_i = \begin{bmatrix} 1-\alpha & \alpha \\ \beta & 1-\beta \end{bmatrix} = \begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix} \tag{5.4}$$

A spectrum reward of +1 is given provided a spectrum agent takes a successful action. For spectrum access, a successful action amounts to choosing channel(s) which are not busy at the next state. Similarly, a spectrum penalty is given when failure of the previous events occur.

Following the template given in (4.7), for the task of spectrum access the following rewards are assigned to each channel.

$$\mathcal{R}^{a}_{s'} = r(a, s') = \begin{bmatrix} \text{null} & \text{null} \\ \text{reward} & \text{penalty} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1 & -0.5 \end{bmatrix} \tag{5.5}$$

Agents follow a framed access schedule. At the beginning of a time frame, the agent senses and observes a set of subband channels. It makes a learning update from the reward received by the decision made last time step and resulting from the current observation. The agent then decides on the action for the next time step. PUs are presumed to begin/end transmission on their assigned channels at the beginning of a frame, hence updating the channel states. At the beginning of the next frame, the agent executes it's access decision for the maximum number of channels it sense/access. If this is not equal to the number of channels in the subband, then only partial state of the subband is observable during the sensing/observation period of each frame.

To simulate a spectrum environment for simple spectrum agents, several agent types are considered. The term 'sensor' will be used broadly to mean the agent mechanism to sense/access spectrum. For each agent an appropriate decision pruning policy is chosen such that the learned optimal decisions are selected to maximize hole utilization given the number of sensors available per agent.

- Random - random channel prediction and access. Channel decisions are uniformly random as well as sensor access of available channels.
- Genie - Spectrum environment is fully characterized and known and optimal greedy actions are taken depending on current environment state. If the optimal decision involves less channel access than available sensors, then additional available sensors are used for sensing remaining channels arbitrarily.
- Belief State Genie - Spectrum environment is fully characterized and known but observations reveal partial information. That is, only k out of n channels are able to be sensed or accessed at any instance. Given this partial information, a belief state is determined and optimal greedy actions are taken depending on the current belief state. An available sensor is not used for channel access if the likelihood of a channel being idle during the next time frame is lower than the likelihood of it being occupied based upon the current belief of the channel state. Additional available sensors are used for sensing remaining channels arbitrarily.

- Replicated Q-Learning - A belief state is maintained and a modified Q-Learning update rule is used to update the Q-table. An ϵ-greedy decision policy is followed using the action values in the current Q-table depending on the current belief state. The number of active accessing sensors chosen is determined by the idle likelihood as done in other belief state agents but Q-value determines which channels will be accessed. As before, additional available sensors are used for sensing remaining channels arbitrarily to maximize state information in the next time frame.

The agent's performance is assessed by the spectrum hole utility at each time step averaged over multiple Monte Carlo simulations of the spectrum environment.

5.3 Single Agent Multiple Sensor Simulations

A Replicated Q-learning agent is realized with the following parameterization:

- learning rate α: 0.2
- discount factor γ: 0.2
- minimum ϵ: 0.05
- ϵ schedule: A decaying ϵ schedule is followed to promote early exploration of the agent. At the beginning of simulation the agent has $\epsilon = 1.0$ which decays by a factor of 0.025 each time step until the minimum ϵ is reached.

For each spectrum simulation, a total of 2000 randomly seeded Monte Carlo runs are average over to determine an empirical average utility metric at each time frame. Each MC simulation steps through 40 time frames before terminating.

The following arrangement of spectrum PU subband environments and available agent sensors are considered, each with i.i.d channel matrices as given in (5.4):

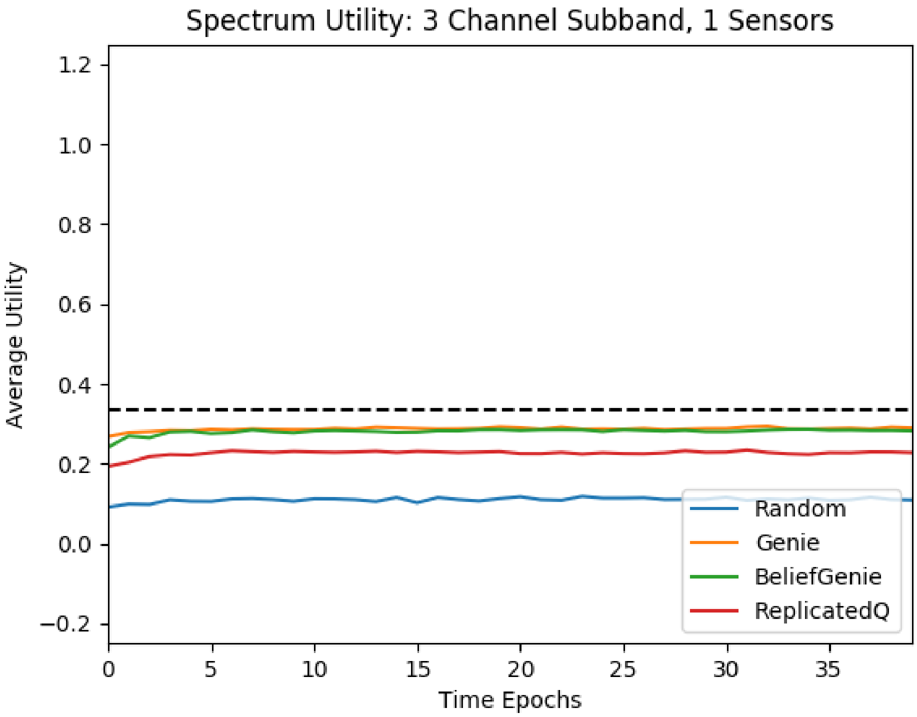
- 3 Channel PU Subband with
  - 1 and 3 available sensors
- 5 Channel PU Subband with
  - 1, 3, and 5 available sensors
- 8 Channel PU Subband with
  - 1, 3, 5, and 8 available sensors
- 10 Channel PU Subband with
  - 1, 3, 5, 8, and 10 available sensors

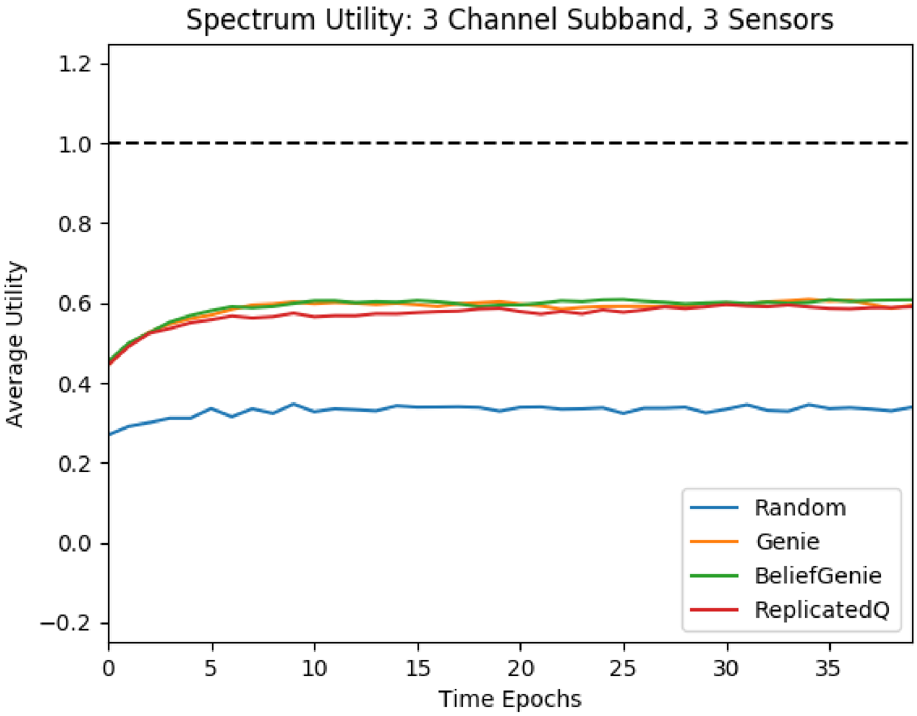
Simulation results are collected in Appendix A. Replicated Q-Learning is found to perform well in comparison to that of the 'Genie' agents, with arbitrarily close convergence (depending upon how small the exploration rate ϵ is) as the number of available agent sensors approaches the number of channels available to sense.

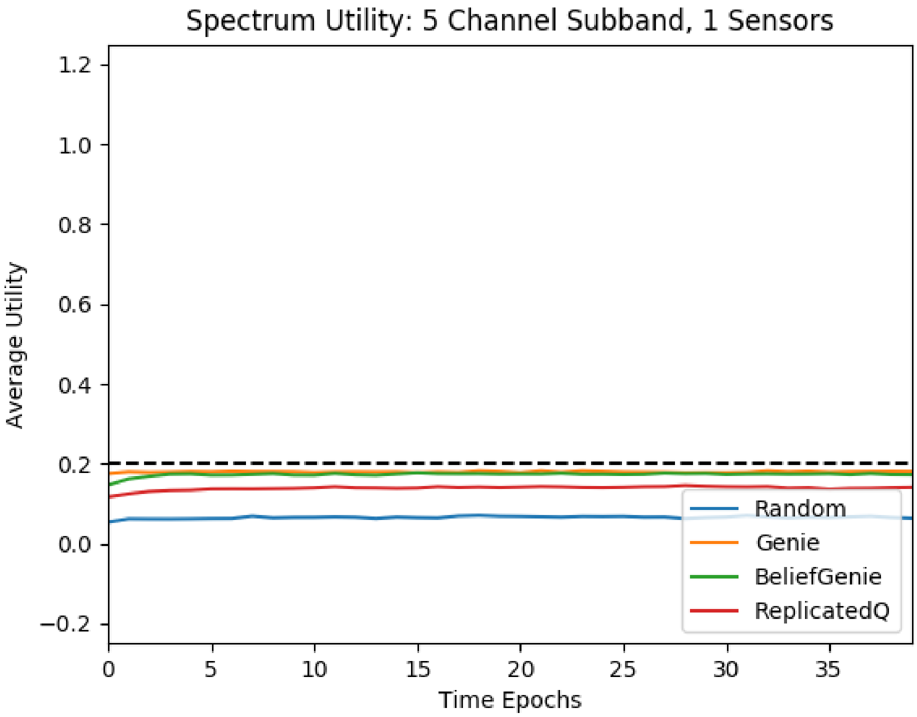
Appendix A

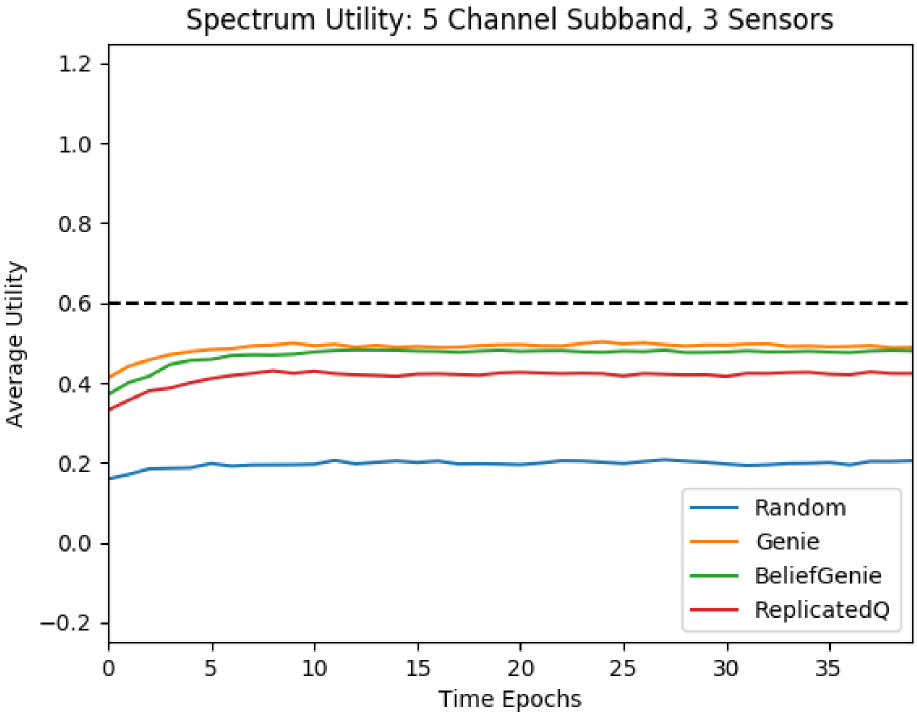
Average Spectrum Hole Utilization Results

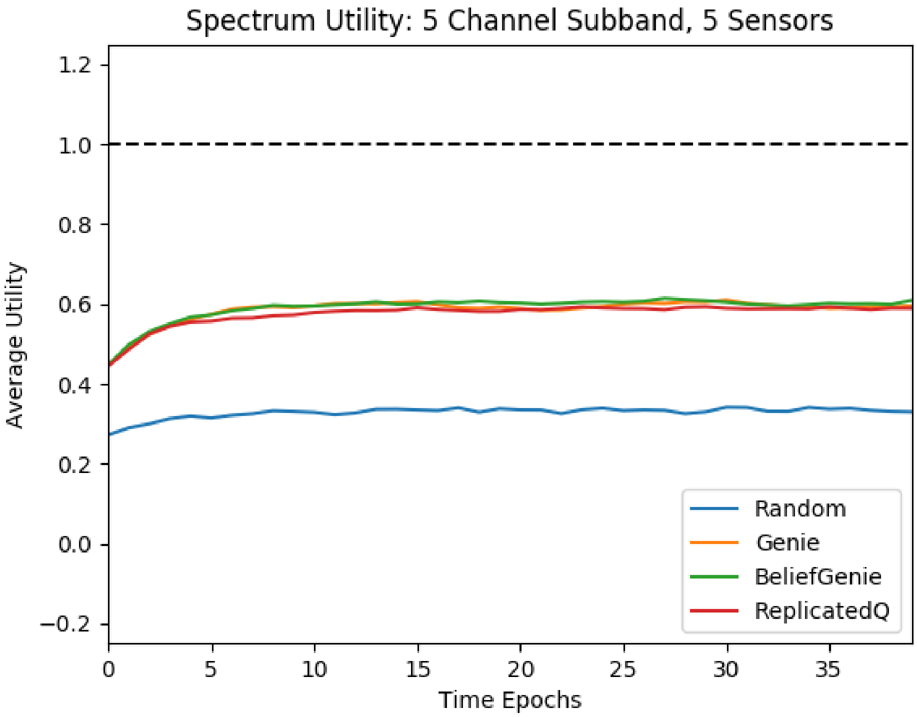
In all figures below, the maximum possible spectrum utilization of available channels within a subband by a secondary user in the case where the presence of primary users is disregarded is denoted by a horizontal dashed line. This limit is included to aid in perspective in the average spectrum hole utilization by an agent as it varies the number of channels it can sense and access. Spectrum hole utilization of the agent during each time frame is determined by the number of successful channel accesses in a time frame. Success is considered as no collision or interference with a PU accessing the same channel during the same frame.

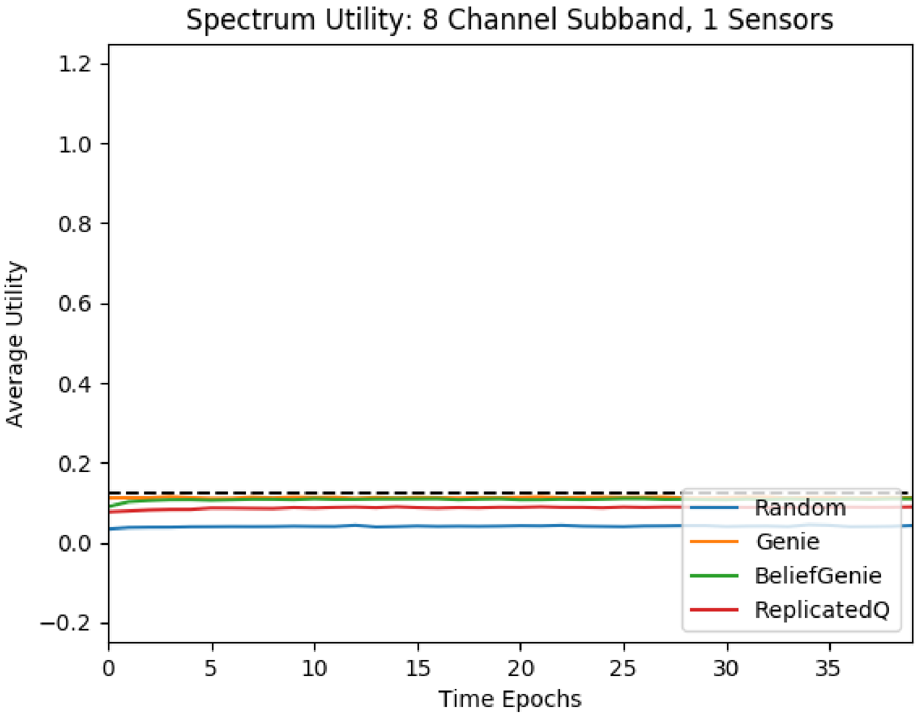

Spectrum Utility: 3 Channel Subband, 1 Sensors
Average Utility
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0
5
10
15
20
25
30
35
Time Epochs
Random
Genie
BeliefGenie
ReplicatedQ

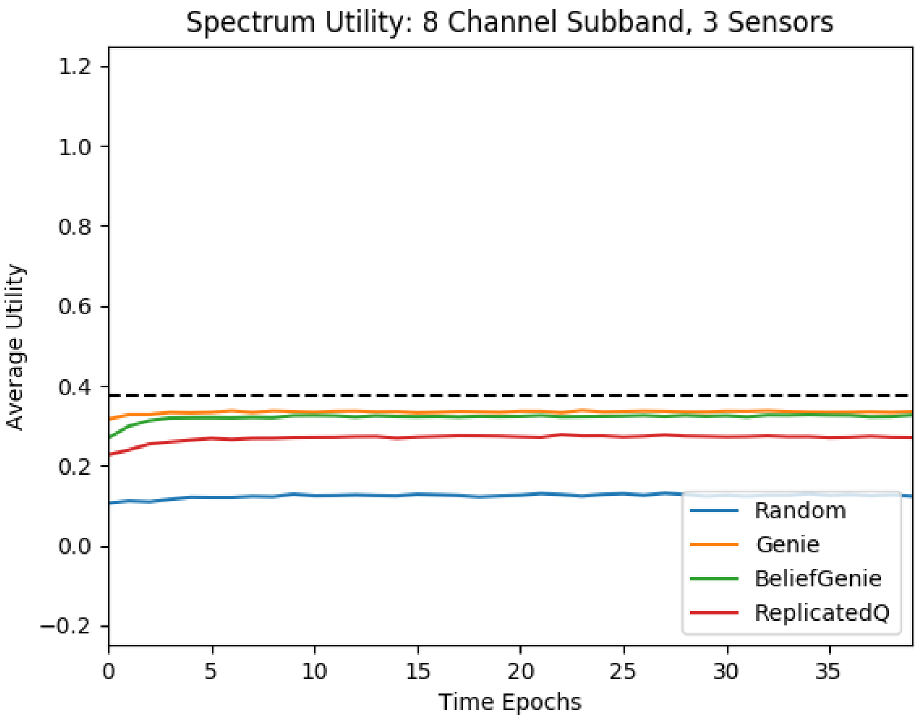

Spectrum Utility: 3 Channel Subband, 3 Sensors
Average Utility
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0
5
10
15
20
25
30
35
Time Epochs
Random
Genie
BeliefGenie
ReplicatedQ

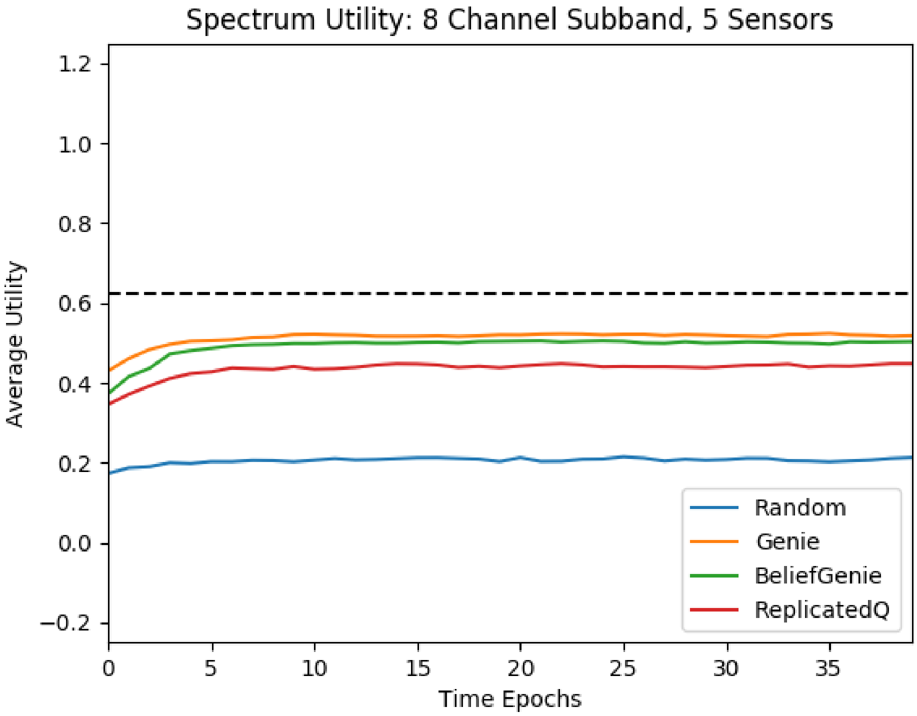

Spectrum Utility: 5 Channel Subband, 1 Sensors
Average Utility
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0
5
10
15
20
25
30
35
Time Epochs
Random
Genie
BeliefGenie
ReplicatedQ

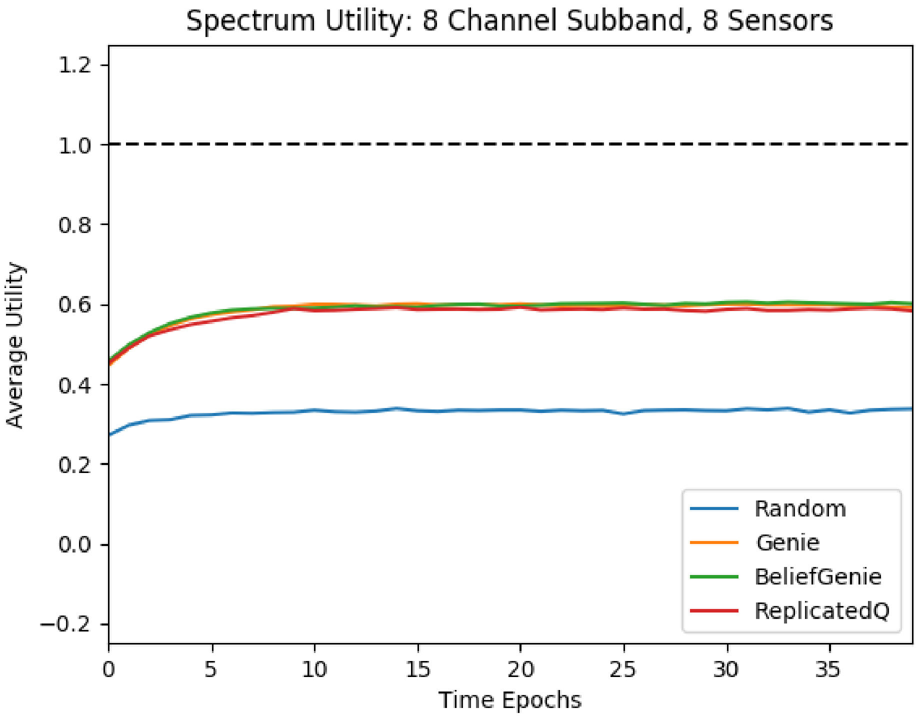

Spectrum Utility: 5 Channel Subband, 3 Sensors
Average Utility
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0
5
10
15
20
25
30
35
Time Epochs
Random
Genie
BeliefGenie
ReplicatedQ

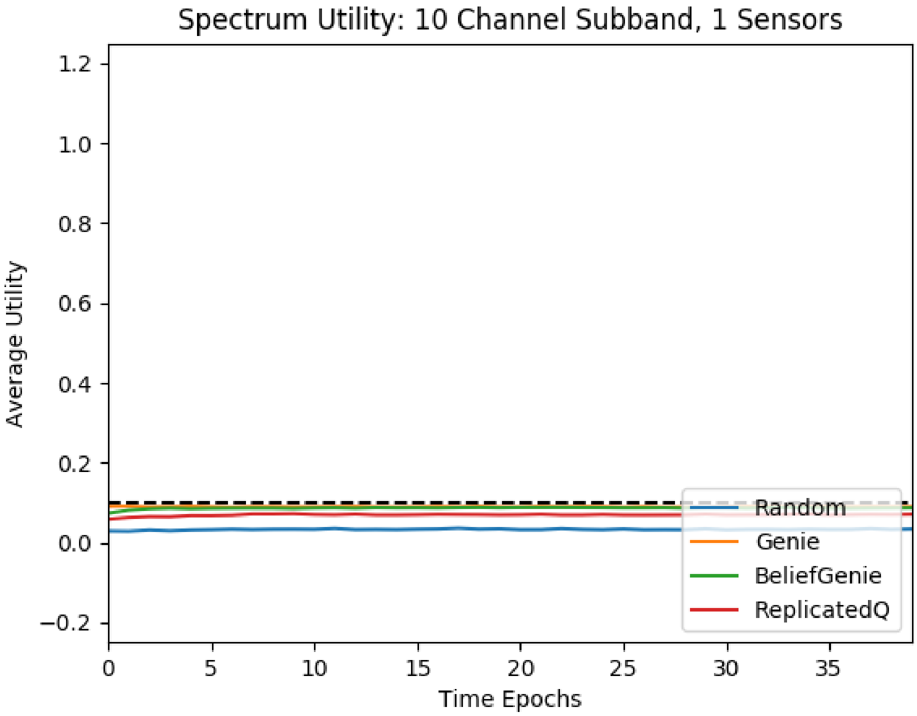

Spectrum Utility: 5 Channel Subband, 5 Sensors
Average Utility
Time Epochs
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0
5
10
15
20
25
30
35
Random
Genie
BeliefGenie
ReplicatedQ

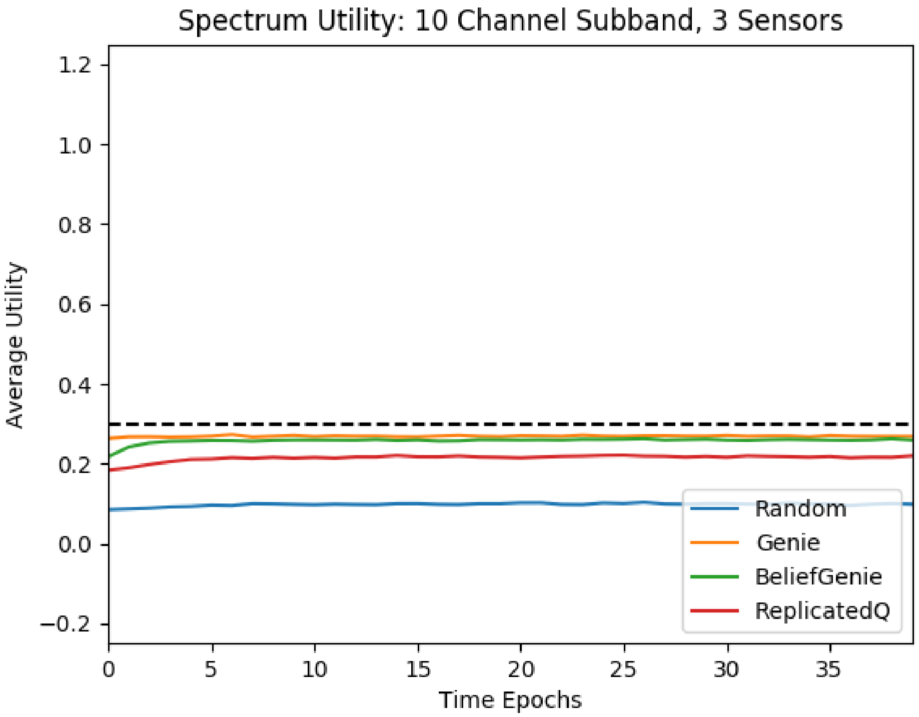

Spectrum Utility: 8 Channel Subband, 1 Sensors
Average Utility
Time Epochs
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0
5
10
15
20
25
30
35
Random
Genie
BeliefGenie
ReplicatedQ

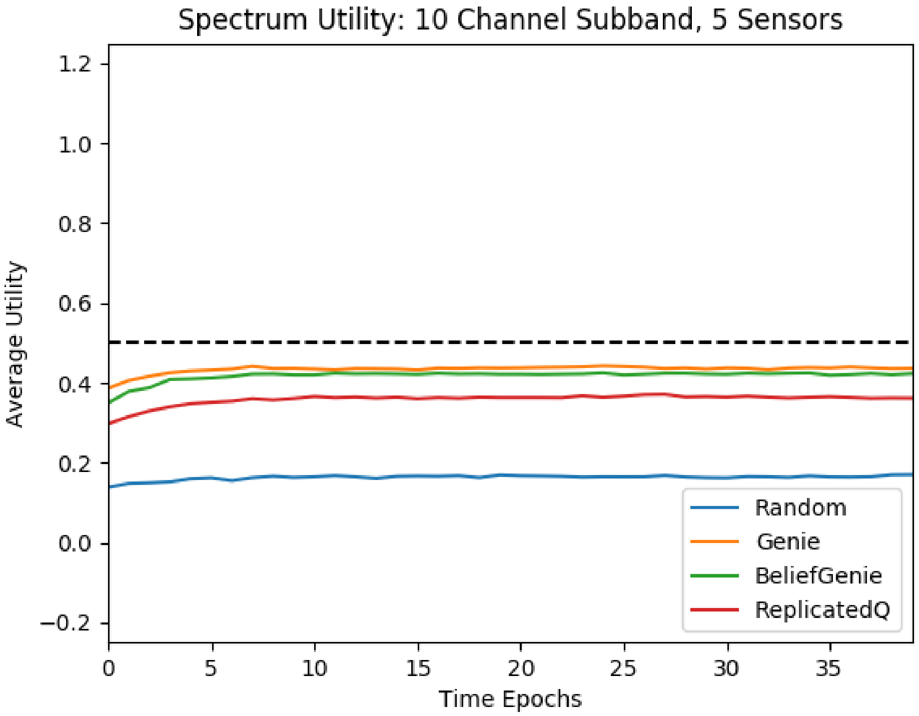

Spectrum Utility: 8 Channel Subband, 3 Sensors
Average Utility
Time Epochs
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0
5
10
15
20
25
30
35
Random
Genie
BeliefGenie
ReplicatedQ

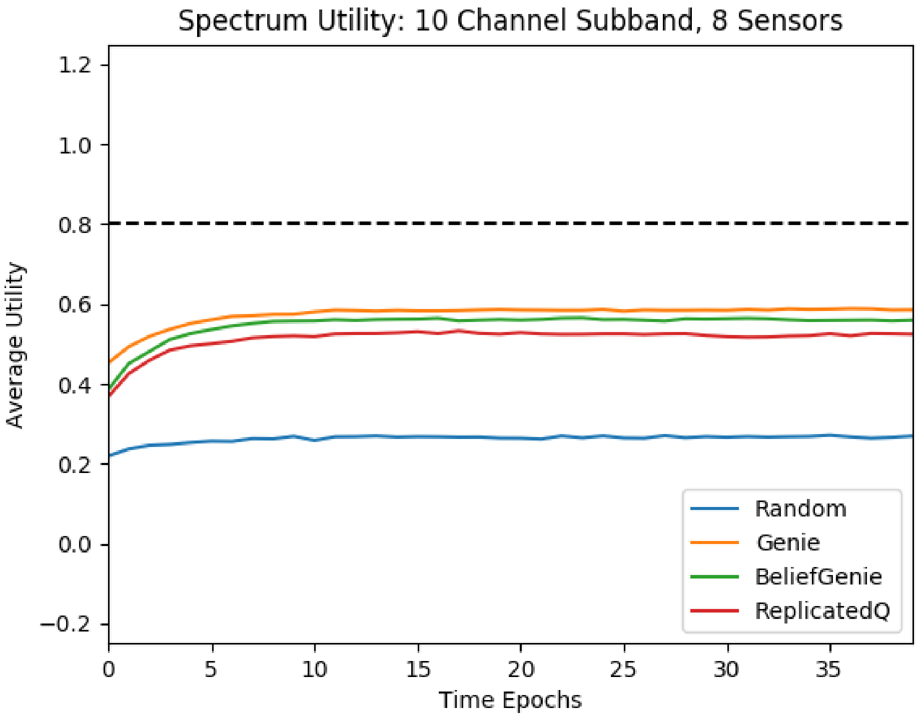

Spectrum Utility: 8 Channel Subband, 5 Sensors
Average Utility
Time Epochs
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0
5
10
15
20
25
30
35
Random
Genie
BeliefGenie
ReplicatedQ

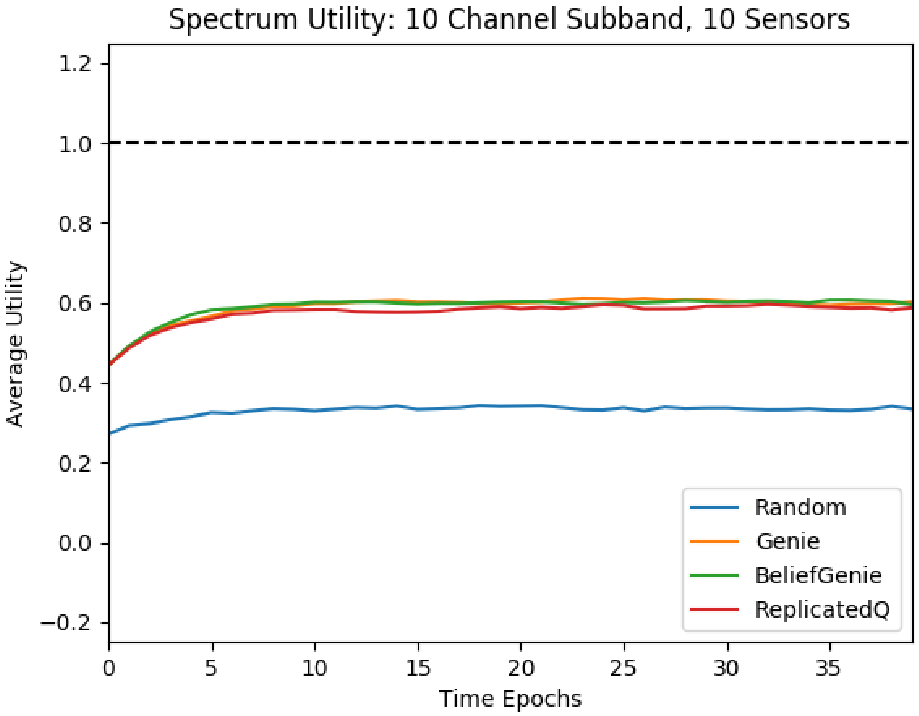

Spectrum Utility: 8 Channel Subband, 8 Sensors
Average Utility
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0
5
10
15
20
25
30
35
Time Epochs
Random
Genie
BeliefGenie
ReplicatedQ


Spectrum Utility: 10 Channel Subband, 1 Sensors
Average Utility
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0
5
10
15
20
25
30
35
Time Epochs
Random
Genie
BeliefGenie
ReplicatedQ


Spectrum Utility: 10 Channel Subband, 3 Sensors
Average Utility
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0
5
10
15
20
25
30
35
Time Epochs
Random
Genie
BeliefGenie
ReplicatedQ


Spectrum Utility: 10 Channel Subband, 5 Sensors
Average Utility
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0
5
10
15
20
25
30
35
Time Epochs
Random
Genie
BeliefGenie
ReplicatedQ


Spectrum Utility: 10 Channel Subband, 8 Sensors
Average Utility
Time Epochs
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0
5
10
15
20
25
30
35
Random
Genie
BeliefGenie
ReplicatedQ


Spectrum Utility: 10 Channel Subband, 10 Sensors
Average Utility
Time Epochs
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0
5
10
15
20
25
30
35
Random
Genie
BeliefGenie
ReplicatedQ

Appendix B

Brief Overview of Spectrum Availability and Utilization

Wireless, radio frequency (RF) spectrum is a precious commodity. In the United States, the Federal Communications Commission (FCC) regulates RF communications in the United States. Historically, RF spectrum has been regulated through licensing bands of spectrum for Federal and non-Federal use. Non-Federal use of of RF spectrum is regulated under Title III of the Communications Act of 1934, and includes TV broadcasting, public radio, cellular phone services, citizen's band radio, etc. Since 1994, competitive RF spectrum auctions have been held by the FCC, both raising billions of for the U.S. Treasury and assigning thousands of licenses.

The rapid growth of mobile wireless communications in the late 20th century strained the availability of RF spectrum for other services and technologies. The FCC studied the utilization of spectrum and found that most licensed spectrum was heavily underutilized, in particular spectrum whose regional usage varied such as broadcast television services. Other findings revealed a significant increase in spectrum efficiency in the so called industrial, scientific, and medical (ISM) "garbage" band at 2.4 - 2.4835 GHz, which is occupied by various wireless signals.

From their findings, the FCC began rethinking their policies for spectrum allocation, and in 2002 the Spectrum Policy Task Force (SPTF) of the FCC published a report detailing their recommendations for the future of spectrum allocation policies [41]. In the report, several specific recommendations were made, with a major theme being both the optimization of licensed spectrum through further band partitioning, increase in spectrum efficiency, and the expansion of access to secondary and unlicensed users provided they do not significantly interfere with licensed use and quality of service (QoS).

Numerous experiments measuring spectral occupancy have been conducted in the past several years verifying the report by the task force regarding spectrum underutilization. In [42], a short-term (2 day) measurement campaign was conducted in Chicago, Illinois in November 2005. Band-by-band spectral occupancy measurements were made in RF regions from 30 - 3000 MHz. The campaign concluded an abundance of unused spectral capacity, with an overall average spectrum use of 17.4%. A summary of their occupancy results is shown in Figure B.1.

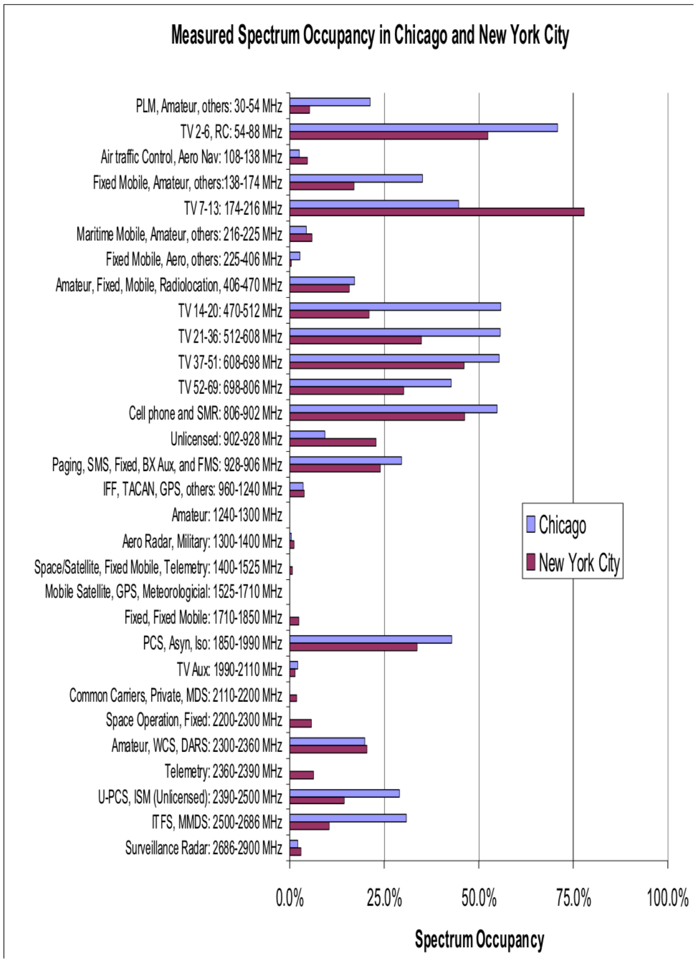
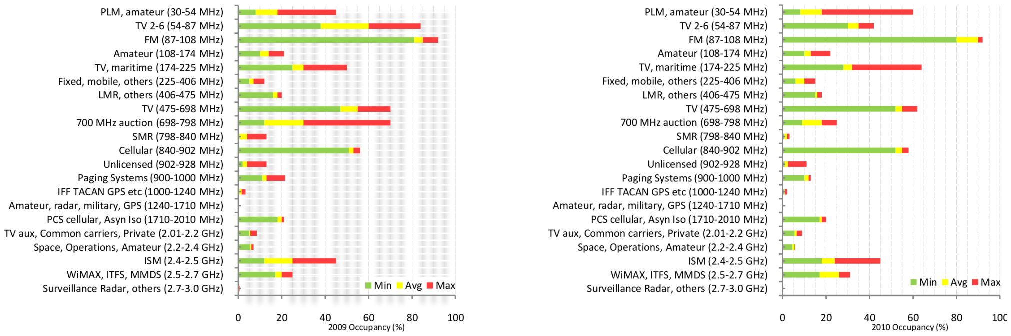


Figure B.1: Average spectrum occupancy by band: Chicago vs. New York. Source: [42]

A later long-term, multiple year measurement campaign in [43] also in Chicago verified continued low occupancy of the 30 - 3000 MHz region into the current decade. It also provided daily, weekly, and seasonal occupancy trends observable in many bands, such as the land mobile radio (LMR) and cellular bands. The long-term campaign also captured the FCC spectral auction campaign for the 700 MHz band (formally licensed for broadcast television) and observable in the spectrum occupancy shifts shown in Figures B.2.



(a) Estimated occupancy by band for 2009. Average overall occupancy is 15% for 30-3000 MHz.

(b) Estimated occupancy by band for 2010 (up to October). Average overall occupancy is 14% for 30-3000 MHz.

Figure B.2: Minimum, average, and maximum occupancies in 21 different spectrum bands in Chicago, IL from 2009-2010. Source: [43]

Man-made and other ambient noise sources were determined to affect spectral occupancy measurements in [44]. These measurements were collected in both an indoor and outdoor setting using energy detection (ED) methods which make decisions based on a predetermined threshold level, making ED methods sensitive to signal-to-noise ratio. From the results, it was concluded that spectrum occupancy is highly dependent on sensing location and decision threshold.

Acknowledging the ever-increasing demand for RF spectrum services, another general survey of RF bands from 30 - 3000 Mhz was conducted in [45] by the Shared Spectrum Company in Vienna, Virginia. Their short-term measurement results found that a number of non-contiguous bands have a low measured spectrum occupancy. Along with occupancy measurements, they also collected data revealing trends in spectral holes throughout several frequency bands. These trends support the aforementioned occupancy measurements in that there are many available channels in most of the spectrum bands that are currently underutilized. These results suggest that a number of spectrum bands are excellent candidates for either FCC reallocation through auctioning or spectrum sharing methods with unlicensed users.

For further information, a few other popular measurement campaigns may be found in [44, 46] as well as a survey of several short and long-term studies of spectrum occupancy in [47]. Overall, the spectral occupancy statistic worldwide is low, and more efficient means of spectrum licensing and spectrum sharing must be put in place to fulfill the ever growing need of available spectrum.

Appendix C

Software Tools for Simulation

The simulation software routines were written in Python 3. They are currently maintained using Git in two separate web-hosted repositories: one for the spectrum environment and the other for the interacting spectrum agents.

- `https://github.com/lukeprince20/gym-spectrum`
- `https://github.com/lukeprince20/spectrum-agents`

Bibliography

[1] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. 2nd ed. MIT press, 2018.

[2] Ronald A Howard. “DYNAMIC PROGRAMMING AND MARKOV PROCESSES..” In: (1960).

[3] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 1994.

[4] Richard Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[5] Richard Bellman. “A Markovian decision process”. In: *Journal of Mathematics and Mechanics* (1957), pp. 679–684.

[6] Christopher John Cornish Hellaby Watkins. “Learning from delayed rewards”. PhD thesis. University of Cambridge England, 1989.

[7] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. “Reinforcement learning: A survey”. In: *Journal of artificial intelligence research* 4 (1996), pp. 237–285.

[8] C. E. Shannon. “A mathematical theory of communication”. In: *The Bell System Technical Journal* 27.3 (July 1948), pp. 379–423. ISSN: 0005-8580. DOI: 10.1002/j.1538-7305.1948.tb01338.x.

[9] Robert G Gallager. *Information theory and reliable communication*. Vol. 2. Springer, 1968.

[10] Hong Shen Wang and N. Moayeri. “Finite-state Markov channel-a useful model for radio communication channels”. In: *IEEE Transactions on Vehicular Technology* 44.1 (Feb. 1995), pp. 163–171. ISSN: 0018-9545. DOI: 10.1109/25.350282.

[11] Stefan Geirhofer, Lang Tong, and Brian M Sadler. “Dynamic spectrum access in WLAN channels: Empirical model and its stochastic analysis”. In: *Proceedings of the first international workshop on Technology and policy for accessing spectrum*. ACM. 2006, p. 14.

[12] Qing Zhao et al. "Decentralized cognitive MAC for opportunistic spectrum access in ad hoc networks: A POMDP framework". In: *IEEE Journal on selected areas in communications* 25.3 (2007).

[13] Y. Chen, Q. Zhao, and A. Swami. "Joint Design and Separation Principle for Opportunistic Spectrum Access in the Presence of Sensing Errors". In: *IEEE Transactions on Information Theory* 54.5 (May 2008), pp. 2053–2071. ISSN: 0018-9448. DOI: 10.1109/TIT.2008.920248.

[14] J. Unnikrishnan and V. V. Veeravalli. "Algorithms for Dynamic Spectrum Access With Learning for Cognitive Radio". In: *IEEE Transactions on Signal Processing* 58.2 (Feb. 2010), pp. 750–760. ISSN: 1053-587X. DOI: 10.1109/TSP.2009.2028970.

[15] Anthony R Cassandra and Leslie Pack Kaelbling. "Learning policies for partially observable environments: Scaling up". In: *Machine Learning Proceedings 1995: Proceedings of the Twelfth International Conference on Machine Learning, Tahoe City, California, July 9-12 1995.* Morgan Kaufmann. 1995, p. 362.

[16] Mario Bkassiny, Sudharman K Jayaweera, and Keith A Avery. "Distributed Reinforcement Learning based MAC protocols for autonomous cognitive secondary users". In: *Wireless and Optical Communications Conference (WOCC), 2011 20th Annual.* IEEE. 2011, pp. 1–6.

[17] Yang Li et al. "Learning-aided sub-band selection algorithms for spectrum sensing in wide-band cognitive radios". In: *IEEE Transactions on wireless communications* 13.4 (2014), pp. 2012–2024.

[18] M. A. Aref et al. "Replicated Q-learning based sub-band selection for wideband spectrum sensing in cognitive radios". In: *2016 IEEE/CIC International Conference on Communications in China (ICCC).* July 2016, pp. 1–6. DOI: 10.1109/ICCChina.2016.7636732.

[19] Mario Bkassiny. "Wideband Spectrum Sensing and Signal Classification for Autonomous Self-Learning Cognitive Radios". PhD thesis. University of New Mexico, 2013, p. 200. URL: http://search.proquest.com/docview/1508569171/abstract/44E23DB69155445FPQ/1?accountid=28833.

[20] Peter Whittle. "Restless bandits: Activity allocation in a changing world". In: *Journal of applied probability* 25.A (1988), pp. 287–298.

[21] Keqin Liu and Qing Zhao. "Indexability of restless bandit problems and optimality of whittle index for dynamic multichannel access". In: *IEEE Transactions on Information Theory* 56.11 (2010), pp. 5547–5567.

[22] Jarmo Lunden, Visa Koivunen, and H Vincent Poor. "Spectrum exploration and exploitation for cognitive radio: Recent advances". In: *IEEE signal processing magazine* 32.3 (2015), pp. 123–140.

[23] "Cognitive radio: making software radios more personal". In: *IEEE Personal Communications* 6.4 (1999), pp. 13–18. DOI: 10.1109/98.788210.

[24] Jospeph Mitola III. "Cognitive Radio: An Integrated Agent Architecture for Software Defined Radio". PhD thesis. KTH, Sweden, 2000, p. 313.

[25] Sudharman K Jayaweera. *Signal processing for cognitive radios.* John Wiley & Sons, 2014.

[26] Andrea J Goldsmith et al. "Breaking spectrum gridlock with cognitive radios: An information theoretic perspective." In: *Proceedings of the IEEE* 97.5 (2009), pp. 894–914.

[27] Qiying Hu and Wuyi Yue. *Markov decision processes with their applications.* Vol. 14. Springer Science & Business Media, 2007.

[28] Neil A Weiss. *A course in probability.* Addison-Wesley, 2006.

[29] Charlotte Striebel. "Sufficient statistics in the optimum control of stochastic systems". In: *Journal of Mathematical Analysis and Applications* 12.3 (1965), pp. 576–592.

[30] Ana Galindo-Serrano and Lorenza Giupponi. "Distributed Q-learning for aggregated interference control in cognitive radio networks". In: *IEEE Transactions on Vehicular Technology* 59.4 (2010), pp. 1823–1834.

[31] Lonnie Chrisman. "Reinforcement learning with perceptual aliasing: The perceptual distinctions approach". In: *AAAI*. 1992, pp. 183–188.

[32] R Andrew McCallum. "First results with utile distinction memory for reinforcement learning". In: (1992).

[33] Aditya Mahajan and Demosthenis Teneketzis. "Multi-armed bandit problems". In: *Foundations and Applications of Sensor Management.* Springer, 2008, pp. 121–151.

[34] Lifeng Lai et al. "Cognitive medium access: Exploration, exploitation, and competition". In: *IEEE Transactions on Mobile Computing* 10.2 (2011), pp. 239–253.

[35] Sarah Filippi, Olivier Cappé, and Aurélien Garivier. "Optimally sensing a single channel without prior information: The tiling algorithm and regret bounds". In: *IEEE Journal of Selected Topics in Signal Processing* 5.1 (2011), pp. 68–76.

[36] Cem Tekin and Mingyan Liu. "Online learning in opportunistic spectrum access: A restless bandit approach". In: *INFOCOM, 2011 Proceedings IEEE.* IEEE. 2011, pp. 2462–2470.

[37] Kehao Wang and Lin Chen. "On optimality of myopic policy for restless multi-armed bandit problem: An axiomatic approach". In: *IEEE Transactions on Signal Processing* 60.1 (2012), pp. 300–309.

[38] Volodymyr Kuleshov and Doina Precup. “Algorithms for multi-armed bandit problems”. In: *arXiv preprint arXiv:1402.6028* (2014).

[39] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. “Finite-time analysis of the multiarmed bandit problem”. In: *Machine learning* 47.2-3 (2002), pp. 235–256.

[40] Richard D Smallwood and Edward J Sondik. “The optimal control of partially observable Markov processes over a finite horizon”. In: *Operations research* 21.5 (1973), pp. 1071–1088.

[41] Federal Communications Commission Spectrum Policy Task Force. *Report of the Spectrum Efficiency Working Group.* Tech. rep. 2002, p. 37.

[42] Mark A. McHenry et al. “Chicago spectrum occupancy measurements & analysis and a long-term studies proposal”. In: *Proceedings of the first international workshop on Technology and policy for accessing spectrum - TAPAS ’06.* New York, New York, USA: ACM Press, 2006, 1–es. ISBN: 159593510X. DOI: `10.1145/1234388.1234389`.

[43] Tanim M. Taher et al. “Long-term spectral occupancy findings in Chicago”. In: *2011 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN).* IEEE, May 2011, pp. 100–107. ISBN: 978-1-4577-0177-1. DOI: `10.1109/DYSPAN.2011.5936195`.

[44] Matthias Wellens, Jin Wu, and Petri Mahonen. “Evaluation of Spectrum Occupancy in Indoor and Outdoor Scenario in the Context of Cognitive Radio”. In: *2007 2nd International Conference on Cognitive Radio Oriented Wireless Networks and Communications.* IEEE, Aug. 2007, pp. 420–427. ISBN: 978-1-4244-0814-6. DOI: `10.1109/CROWNCOM.2007.4549835`.

[45] Shared Spectrum Company. “General Survey of Radio Frequency Bands – 30 MHz to 3 GHz”. In: *Shared Spectrum Company Reports* (2010), p. 47.

[46] Robin I. C. Chiang, Gerard B. Rowe, and Kevin W. Sowerby. “A Quantitative Analysis of Spectral Occupancy Measurements for Cognitive Radio”. In: *2007 IEEE 65th Vehicular Technology Conference - VTC2007-Spring.* IEEE, Apr. 2007, pp. 3016–3020. ISBN: 1-4244-0266-2. DOI: `10.1109/VETECS.2007.618`.

[47] Deepa Das and Susmita Das. “A Survey on Spectrum Occupancy Measurement for Cognitive Radio”. In: *Wireless Personal Communications* 85.4 (Dec. 2015), pp. 2581–2598. DOI: `10.1007/s11277-015-2921-1`.