AN ABSTRACT OF THE THESIS OF

Pingan Zhu for the degree of Master of Science inComputer Science and Mathematics presented on September 23, 2013.

Title: Revenue-Based Spectrum Management Via Markov Decision Process

Abstract approved: _____

Thinh P. Nguyen Mina E. Ossiander

We consider the problem of wireless spectrum management in cognitive wireless networks that maximizes the revenue for a spectrum operator. Specifically, we study the problem on how a wireless spectrum operator can optimally allocate its limited spectrum to various classes users/devices who pay differently for their spectrum per unit time. We show that the problem of maximizing the revenue for the spectrum operator can be cast in the Markov Decision Process (MDP) framework. To that end, we investigate the performance of two MDP formulations: the finite-horizon and the discounted infinite-horizon models. We show that for small scenarios with known system paramters, it is feasible to obtain the optimal solution for the finite horizon MDP using the backward induction algorithm. For larger scenarios and unknown system parameters, Q-learning is used to approximate the optimal solution/policy via simulations. For real-world scenarios with many system states, it is memory inefficient to represent the optimal policy using large tables. Instead, we also show how to compactly represent the optimal policy using support vector machine (SVM). The SVM representation also allows for the prediction of the optimal actions based on the states that might not be explored during training. The existence of the compact structure of the optimal policies (SVM) for this problem motivates us to explore more efficient solutions. Specifically, under some mild assumptions, we are able to give a threshold policy, which is not only optimal but also very efficient to implement. Simulation results are used to verify our approach.

Revenue-Based Spectrum Management Via Markov Decision Process

by

Pingan Zhu

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented September 23, 2013
Commencement June 2014

Master of Science thesis of Pingan Zhu presented on September 23, 2013.

APPROVED:

Co-Major Professor, representing Computer Science

Co-Major Professor, representing Mathematics

Director of the School of Electrical Engineering and Computer Science

---

Chair of the Department of Mathematics

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Pingan Zhu, Author

ACKNOWLEDGEMENTS

I would like to express my deep gratitude to my advisor Dr. Thinh Nguyen for guidance and ideas which helped me complete my thesis. Also I would like to thank Dr. Alan Fern and Mr. Jervis Pinto, who helped me a lot when I just started my graduate school.

I would like to thank Dr. Yevgeniy Kovchegov , Dr. Mina E. Ossiande and Dr. Glencora Borradaile whose courses and ideas in probability and computer theory influenced me a lot, and hence for this project.

Finally, I would thank to my family back in China for their support and encouragement without which this would not have been possible.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF TABLES

Chapter 1: Introduction

1.1 The Problem

Arguably, modern societies have been driven by forces of technological advances and economic models during the past century. Specifically, the digital revolution and market economies have been two of the hallmarks of the later part of the twentieth century and are likely to remain the dominant driving forces for years to come. At the center of the digital revolution is the proliferation of wireless technologies that promises connecting people to people, from anywhere, at anytime, encompassing different modalities from data and voice to streaming audio and video. The future of wireless communications, however, comes with a set of unique challenges. As the number of wireless devices grows, they will ultimately compete for the same finite resource: *wireless spectrum.* Unlike wired communications in which, theoretically, more fibers can be used to accommodate the increasing bandwidth demand, wireless spectrum cannot be arbitrarily increased due to the fundamental limitations imposed by the physical laws. In addition, regulatory policies may further limit the ability to allocate the spectrum efficiently. Thus, the proliferation of wireless communication depends critically not only on the technological advances, but also on its marriage to sound regulatory policies and economic incentives.

That said, the current static spectrum allocation for licensed access, e.g., TV and cellular phones, which account for most of the usable wireless spectrum is far from efficient. According to the FCC's Spectrum Policy Task Force, much of the usable spectrum is pre-allocated, but unused at a given time and location [5]. This observation is the basis for much recent research on the concept of *Dynamic Spectrum Access* (DSA) [20, 1][7, 8][10, 17][11, 3]. DSA allows a device's operating band to be allocated dynamically in both spatial and temporal dimensions in order to utilize spectrum more efficiently. From the technological perspective, future wireless devices, commonly known as cognitive radio devices, must be able to detect the presence of others and cooperate with each other in such a way to enhance the overall spectrum efficiency. However, wireless spectrum is not free. In addition, to build and maintain such a large-scale, distributed, and complex

DSA infrastructure requires multiple players from industries and goverments. Therefore, profit structure must be first established in order to provide incentives for companies to to build and operate such networks [6, 12, 19] . From the market economty perspective, a popular approach is to associate a price for spectrum. The spectrum owner can set up protocol that enable devices/users compete for spectrum access [9, 13]. The assumption is that with market competition is easy to implement in a distributed manner, and that it will eventually leads to efficienty operating point.

1.2 Thesis Overview

In the spirit of market economy, this thesis aims to enhance the spectrum efficiency through pricing for well-defined and limited settings. It does not address the overall goals of the DSA that involves technological advances, market forces, and governmental regulatory policies at a broader scope. In contrast, the thesis studies the feasibility and efficiency of incorporating priced-based wireless access protocols for small wireless network such as Wi-Fi hotspots or femto cell networks as shown in Fig. 1.1. In such networks, such a scenario, a spectrum owner is assumed to own the wireless access point. For a device/user to be granted access to the networks by the spectrum owner, it must pay the owner a certain price per spectrum unit per time unit. Based on the price the user is willing to pay, and the associated cost to operate the network, the owner can decide whether to grant or to deny the user access to the wireless spectrum. The goal of the owner is to maximize his/her profit. The goal of the user is to get the wireless service at a reasonable price according to his or her own need. The assumption is that market force will automatically drive the system to an efficient operating point.

Our approach is based on the classic *Markov Decision Process*(MDP)[14], a well-known framework for making decisions optimally under uncertainty. In particular, we investigate the performance of two MDP formulations: the finite-horizon and the discounted infinite-horizon models. We show that for small scenarios with known system parameters, it is feasible to obtain the optimal solution for the finite horizon MDP using the backward induction algorithm. For larger scenarios and unknown system parameters, Q-learning is used to approximate the optimal solution/policy via simulations. For real-world scenarios with many system states, it is memory inefficient to represent the optimal policy using large tables. Instead, we also show how to compactly represent

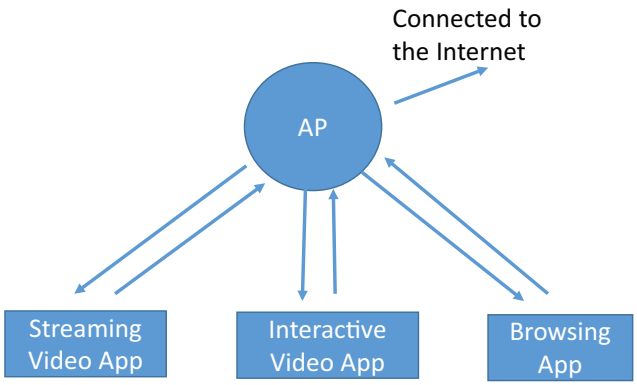


Figure 1.1: Network Access via a Wi-Fi hotspot

the optimal policy using support vector machine (SVM). The SVM representation also allows for the prediction of the optimal actions based on the states that might not be explored during training. The existence of the compact structure of the optimal policies (SVM) for this problem motivates us to explore more efficient solutions. Specifically, under some mild assumptions, we are able to give a threshold policy, which is not only optimal but also very efficient to implement.

1.3 Thesis Organization

In Chapter 2, we will give an overview of the Markov Decision Process. In Chapter 3, we present the finite-horizon MDP model used to find the optimal policy for settings that involve a small number of wireless users/devices. In this model, the owner considers short-term profit or revenue and has accurate estimation of the MDP parameters (which are usually not known in real-world scenarios). In Chapter 4, we present the infinite-horizon MDP model used to find the optimal policy for larger wireless networks. The setup aims to maximize the long-term revenue profit for the spectrum owner. We will describe how Q-learning is employed to obtain the approximately optimal policy. We will also show empirically that SVM can be used to represent, and in some cases, predict the optimal policies. Finally, in 5, we derive the optimality conditions for a policy for the proposed problem under some mild conditions.

Chapter 2: Markov Decision Process

2.1 Overview of MDP

The MDP[14] framework is used to study the optimal decision making processes under uncertainty. Typically, MDP is used to model a dynamical process that involves: 1) a controller who acts based on his or her current observations of the environment; 2) a reward associated with the controller's action; 3) the environment which changes probabilistically under the controller's actions. The goal of MDP is to derive the optimal policy, i.e., which action to take given the current observations in order to maximize the expected cumulative reward. There are a number of definitions for expected cumulative rewards. The two most are are to be defined shortly for the finite and infinite horizon models. In addition, In this thesis, our focus will be on discrete-time MDP. In a discrete-time MDP, the all observations and all actions are taken in discrete time steps.

Formally, an MDP system is characterized by following:

- a finite set of states $\mathcal{S}$, which represents the possible states of the system that can be observed by the controller,
- a set of control actions $\mathcal{A}$ by the controller,
- a transition probability P which models the changes of the environment induced by the controller's actions,
- and a reward function r given to the controller for taking a particular action in a particular state.

A bit more precisely, the transition probability characterize the dynamics of the system. The transition probability denotes as $P(s_{n+1}|s_n, a)$ denotes the probability of the system moving to state s_{n+1} at time $n+1$ after taking action a in the current state s_n at time n. The dynamics are Markovian in the sense that the probability of the next state s_{n+1} depends only on the current state s_n and action a, and not on any previous history. The reward function $r(s_n, a_n)$ is typically a real-valued function of the current state s_n and

the action a_n. Now, a policy π is a sequence of decisions (actions) a_1, a_2, $\ldots a_n$ taken by the controller with t denotes the time index. Formally, a policy specifies a mapping from states to actions at each time step $\pi_n : \mathcal{S} \rightarrow \mathcal{A}$. The policy π that is called stationary if its actions depends only on the state s, independent of time index. For infinite-horizontal MDP model, there exists an optimal stationary policy. Furthermore, a stationary policy induces a time-invariant transition probability matrix.

Every policy π is associated with a value function V^{π} such that $V^{\pi}(s)$ gives the expected cumulative reward achieved by Π when starting in state s_0. The solution to an MDP problem is the optimal policy π^* that maximizes the expected cumulative reward over some finite or infinite number of time steps.

We give the detail discussion of finite-horizon and infinite-horizon MDP in the next section. Here, we briefly show the value functions for both settings: the former and later are termed finite-horizon MDP and discounted infinite-horizon MDP respectively.

$$V_N^{\pi}(s) = E_s^{\pi}\left\{\sum_{n=1}^{N-1} r_n(s_n, a_n) + r_N(s_N)\right\}, \tag{2.1}$$

$$V^{\pi}(s) = E_s^{\pi}\left\{\sum_{n=1}^{\infty} \beta^n r_n(s_n, a_n)\right\}, \tag{2.2}$$

where $0 \leq \beta < 1$ denotes a given discount factor which provides convergence of $V^{\pi}(s)$, but also carries the notion of discounting the future reward, i.e., putting less emphasis on the rewards in the far future than those in the near future. In the following sections, we give a brief introduction to both finite-horizon MDP and infinite-horizon MDP. The detail solution approaches are given in Chapter 3, Chapter 4 and Chap 5.

2.2 Finite-Horizon MDP

The finite-Horizon MDP model [14] aims to maximize the reward in a given finite number of time steps, which can be viewed as an average short-term based reward. Therefore, for the finite-horizon MDP, the primarily concern is to determine a policy $\pi^* \in \Pi$ with the largest expected total reward. As in (2.1), in the finite-horizon MDP, we use the

expected total reward criterion. That is to say we need a policy π^*, such that

$$V_N^{\pi^*}(s) \geq V_N^{\pi}(s), \quad s \in \mathcal{S}$$

for all $\pi \in \Pi$, such policy is called an *optimal policy*. However, in some finite-horizon MDP, there might not exist an optimal policy. As a result, in stead of seeking an *optimal policy*, we look for an ϵ-*optimal policy*, which means for an $\epsilon > 0$, a policy π_ϵ^* with the property that

$$V_N^{\pi_\epsilon^*}(s) + \epsilon > V_N^{\pi}(s), \quad s \in \mathcal{S}$$

for all $\pi \in \Pi$. Therefore, we define the characterization of value of the MDP, v_N^* as

$$V_N^*(s) \equiv \sup_{\pi \in \Pi} V_N^{\pi}(s) \quad s \in \mathcal{S} \tag{2.3}$$

and when the supremum in (2.3) is attained, by

$$V_N^*(s) = \max_{\pi \in \Pi} V_N^{\pi}(s), \quad s \in \mathcal{S}.$$

Now it is easy to see that, the expected total reward of an optimal policy π^* satisfies

$$V_N^{\pi^*}(s) = V_N^*(s), \quad s \in \mathcal{S}$$

and the value of an ϵ-optimal policy π_ϵ^* satisfies

$$V_N^{\pi_\epsilon^*}(s) + \epsilon > V_N^*(s), \quad s \in \mathcal{S}.$$

By the definition of the supremum, such a policy exists for any $\epsilon > 0$.

2.3 Infinite-Horizon MDP

Unlike the finite-horizon model, the infinite-horizon MDP[14] needs to evaluate an infinite sequence of rewards at all states in $\mathcal{S}$. Consequently, we need a pointwise convergent function on $\mathcal{S}$, whose limits are defined separately for each s in $\mathcal{S}$. There are a few methods that can return an convergent function in $\mathcal{S}$, such the discounted reward criterion.

For average reward model, the value function can be defined as:

$$V^{\pi}(s) = E_s^{\pi} \sum_{n=1}^{\infty} \beta^n r_n(s_n, a_n) \quad .$$

Under certain conditions $V^{\pi}(s)$ can be bounded, but often it is not. Therefore, to guarantee convergence for $V^{\pi}(s)$, a discounted factor, $0 \leq \beta < 1$, is introduced into the infinite horizon MDP. Such a model is termed discounted infinite horizon MDP.

For a discounted model, (2.2) shows the value function of infinite-horizon discounted MDP. Similarly to finite-horizon MDP, we say that a policy π^* is total reward optimal whenever

$$V^{\pi^*}(s) \geq V^{\pi}(s) \quad \text{for each } s \in \mathcal{S} \text{ and all } \pi \in \Pi.$$

Define the value of the MDP by

$$V^*(s) \equiv \sup_{\pi \in \Pi} V^{\pi}(s)$$

An optimal policy $\pi^* \in \Pi$ exists whenever

$$V^{\pi^*}(s) = V^*(s) \quad \forall s \in \mathcal{S}$$

We use $0 \leq \beta < 0$ to denote the discounted factor and for a policy π^* is discount optimal, if for fixed β, whenever

$$V_{\beta}^{\pi^*}(s) \geq V_{\beta}^{\pi}(s) \quad \text{for each} s \in \mathcal{S} \text{ and all } \pi \in \Pi$$

As a result, the value of the MDP, $V_{\beta}^{\pi}(s)$ is defined as below:

$$V_{\beta}^*(s) \equiv \sup_{\pi \in \Pi} V_{\beta}^{\pi}(s)$$

Consequently, we can say that a discount optimal policy $\pi \in \Pi$ exists whenever

$$V_{\beta}^{\pi^*}(s) = V_{\beta}^*(s) \quad \forall s \in \mathcal{S}$$

Note that the discount model, there exist an optimal and stationary policy.

There are many algorithmic solutions for finding the optimal policy for the finite-horizon MDP such as Backward Induction and infinite-horizon MDP such as policy and value iterations.

Chapter 3: Revenue Based Pricing Protocol using finite-horizon MDP

In this chapter, we describe the optimal pricing scheme based on the finite-horizon MDP model.

3.1 System Model

In this section, we consider an approach for designing spectrum access mechanisms that maximize the revenue for a spectrum operator. Specifically, we consider the model in which radio bands are allocated dynamically for exclusive use. Furthermore, the spectrum allocation is done at the time scale of the application duration that can lead to higher spectrum efficiency than those resulting from the larger time scale allocation. We assume all devices to be cognitive radio devices, i.e., they can operate on different radio bands. We assume the following protocol for regulating the wireless access of a device:

1. A device sends a message to the spectrum operator requesting for access and the price it is willing to pay for the transmission.

2. The spectrum operator decides to accept the request or not based on the current system state and the cost of maintaining the transmission.

3. If the user/device is accepted, the device will be granted exclusive use of a radio band, and the spectrum operator will receive an amount of revenue proportional to time the device using the band.

4. If the offer is not accepted, the spectrum operator will not get any additional revenue. The device can try to access the spectrum again.

The goal of the spectrum operator is to maximize its expected revenue over some finite number or infinite number of time steps. To do so, whenever there is an joining request

from a device, the spectrum operator will make a decision on whether or not to accept the offer from a joining device, based on various factors including the current spectrum demand, the price a device is willing to pay, its the application classes, its environmental factors such as Signal-to-Noise Ratio (SNR).

The spectrum operator is intended to be a device that manages the spectrum access using the protocol above. This protocol aims to model a number of scenarios including Wi-Fi hotspots, or micro cells, or femto cells. For example, let us consider a scenario of a wireless service provider providing Wi-Fi access for passengers at airports as shown in Fig. 1.1. The spectrum operator would be the AP that manages the channel access based on the current demand of passengers, their types of traffic/applications, their willingness to pay for the wireless service. Importantly, the above model allows for the payment on the scale of application duration, i.e., on the order of seconds, minutes, or hours. The goal is to implement an automatic scheme on whether or not to accept a price from a device/users requesting to join the network in such a way to maximize the expected revenue for the wireless service provider.

By using the model above, we assume that the spectrum operator has M available bands which it can sell to a requesting user/device. A request includes an amount of bandwidth, an application class, the price the user is willing to pay, the user's environmental factors SNR which can directly affect the cost of delivering the bits to the users. For example, for application class, a user may request 2 Mbps for video streaming (streaming application class) or 96 Kbps for IP-telephony (interactive application class). Regarding the price, a user is willing to pay with 10 cents per second per Mbps for streaming class, but only 1 cent per second per Mbps for browsing the web. Furthermore, a user might be located far away from the AP, or has other interference such that its receiving SNR is 15 dB. In this case, the AP might have to increase its transmission power in order to achieve the requested bandwidth. Thus, the cost of operating the AP for this user is higher than than other users that are located closer to the AP or have less interference that result in a larger SNR, e.g., 25 dB. Intuitively, the spectrum operator, or the AP in this case, should take this these information into account in order to make decisions that maximize its overall revenue.

In addition to the user requests, to make the optimal decision, the AP should also rely on the past statistics of requesting users. Specifically, at every time step, there is a probability q_i that an existing user of application class i will be leaving the system and

a probability β_j that the new user of application class j will request an access. The AP should also keep track of the current available spectrum. These information will help the AP devising a scheme of whether or not to accepting a current request in order to maximize its expected revenue over time.

Given the assumption above, it is not clear what the optimal strategy for the spectrum operator should be based on the information it has. For example, let us consider a strategy for the spectrum operator that always accepts every requests regardless of the offered prices, in a first come first server manner. This strategy will ensure that the AP will have more users in the network with the intuition that more users implies higher revenue. However, this strategy does not necessarily maximize the revenue. In fact, by accepting every requesting users even ones with lower prices, until the spectrum owner has no more spectrum left, on the average, the spectrum owner will have lower revenue per second. Intuitively, this strategy takes away the opportunities to generate higher revenue by waiting for users who are willing to pay at higher prices at a later time. The situations are further made more complex by considering other factors including application classes, SNR, and statistics of how frequent the users requests to join and leave the networks.

In summary, an optimal strategy should consider the current access demand for different application classes, the available spectrum, the characteristic of the current request, e.g., prices per Hz per second, user's environmental characteristics, e.g., SNR to make a decision on whether or not to accept the current request. In what follows, we present a precise MDP model that takes into account all of these points.

3.2 MDP Formulation

The first step in applying the MDP framework to a specific scenario is to model the appropriate components of the abstract MDP: namely, the action space $\mathcal{A}$, the state space $\mathcal{S}$, the reward r, and system dynamics via the transition probability matrix P. In a real-world scenario, the action space is limited by what can be done without too much overhead. The state space is also restricted to what can be observed by the MDP controller. The reward is used to model the objective of the system. Finally, the transition probabilities are the properties of the environments. Also, in many scenarios, state and action spaces are intentionally restricted to smaller sizes in order to make

the solutions tractable. To that end, we model the MDP components for the spectrum management problem as follows:

State space. A system state s_n at time n includes the current request and the system status. Specifically, the current request is a vector that includes the application class $c_{i,n}$, the associated requested bandwidth b_n, a_n the price user/device is willing to pay, whose unit is \$ per Mhz per second, and its SNR value. The system status is an $M \times 2$ matrix. The M rows of the matrix represent the M radio band. The first column of the matrix represents the class of users/applications i currently in the system $c_{i,n}$. The second column shows the corresponding revenue $v_{i,n}$ obtained by from user i at time step n per Mhz per second. $v_{i,n}$ for are calculated by multiplying the price per bandwidth per second with the number of bands requested by the user i, then subtracting it from the power cost of transmit the data to user i. This cost is based on the user i's SNR. Finally, a row $(0, 0)$ in the status matrix implies the corresponding band is idle, i.e., not yet assigned.

$$s_n = \begin{bmatrix} (c_{i,n}, b_n, a_n, snr_n) & , & \begin{pmatrix} c_{1,n} & v_{1,n} \\ c_{2,n} & v_{2,n} \\ ... & ... \\ c_{M,n} & v_{M,n} \end{pmatrix} \end{bmatrix}$$

Control action space. After receiving a request of a device, the spectrum operator (MDP controller) will decide whether or not it accepts the request. As discussed previously, the request includes the information on the price the requested user is willing to pay, the amount of bandwidth, the SNR, and the application class. Based on these information, the spectrum operator will have only two options: to accept or to reject the request. The decision by the spectrum operator will induce a probability of the system state to transition to a some other state s_{n+1} in the next time step $n+1$. This transition probability is characterized next.

Transition probability. We assume that a user only quits at the end of a time slot, and a new user only joins at the beginning of the time slot. Given this, the transition probability from state $s_n = s$ to $s_n = s'$ at time step n can be computed as the product of three different probabilities: $q_{i,n}$ the probability of a certain user of class i leaving the system, $1 - q_{i,n}$ the probability of a user of class i remaining in the system, the probability of a new user of class i requesting access $\beta_{i,n}$ at time step n. Given the two

states $s_n = s$ to $s_n = s'$, and let $c^s_{m,n}$ and $c^{s'}_{m,n}$ be specified as in the state space above for the states s and s', then the four probabilities can be computed as follows:

$$
\begin{aligned}
P_{quit} &= \prod_{c^s_{i,n}>0, c^{s'}_{i,n}=0} q_{c^s_{i,n}} \\
P_{remain} &= \prod_{c^s_{i,n}=c^{s'}_{i,n}>0} [1 - q_{c^s_{i,n}}] \\
P_{assign} &= \frac{1}{\left[\sum_{i=1}^{M} \mathbb{1}_{c^{s'}_{i,n}=0}\right] + 1} \\
P_{request} &= \beta_{j,n}
\end{aligned}
$$

The transition probability from state s to state s' is then

$$
p_{s'|s,a} = \begin{cases} P_{quit} P_{remain} P_{request} & \text{if } a = 0 \\ P_{quit} P_{remain} P_{request} P_{assign} & \text{if } a \neq 0 \end{cases}
$$

Reward. Since the goal of the spectrum operator is to maximize the expected reward, a natural way to achieve this goal is to set the immediate reward to the revenue that the operator earns per time step. This amount depends on to the number of users and the type of users/application currently in the system. A higher number of use rs and high priority applications leads to higher revenue per time step. Formally, if $v_{i,n}$ denotes the price the operator sets for the user i (and is accepted) minus the transmission cost (related to SNR) then the immediate reward at time step n can be computed as:

$$
r_n = \sum_{i=1}^{M} v_{i,n} \tag{3.1}
$$

3.3 Solution Approach: Backward Induction

In this section, we give a basic algorithm for evaluating the policy of finite-horizon MDP and we give the proof of its optimality in Appendix A. As stated in Chapter

3, the finite-horizon MDP aims to maximize the reward in a given finite number of time steps. For a problem with limited state and action spaces, the classic backward induction (BI) algorithm is an efficient method for finding the optimal policy, i.e., pricing schemes. The BI algorithm starts from the last time step, moves backward, explores all the possible policies at the previous time step for each state and stores the "best" one in the optimal policy table. It always returns the optimal policy. Let the optimal policy $\pi^* = (d^*(s_1), d^*(s_2), ..., d^*(s_N))$, where $d(s_i)$denotes the optimal action in state s at time i, then the pseudo-code for the BI algorithm is shown below:

1. Set $n = N$ and $U_N^*(s_N) = r_N(s_N)$ for all $s_N \in S$,
2. Substitute $n-1$ for n and compute $U_n^*(s_n)$ for each $s_n \in S$ by

$$U_n^*(s_n) = \max_{a \in A} \left\{ r_n(s_n, a) + \sum_{j \in S} p(j|s_n, a) U_{n+1}^*(j) \right\}.$$

   Set

$$d^*(s_n) = \arg\max_{a \in A} \left\{ r_n(s_n, a) + \sum_{j \in S} p(j|s_n, a) U_{n+1}^*(j) \right\}.$$

3. If $n = 1$, stop. Otherwise return to step 2.

In Section 3.2, we already showed how to compute r_n and $p(j|s_n, a_n)$ based on the assumption that we know the statistics of spectrum demand (probabilities of users with different application classes requesting access) . Thus, the optimal policy can be easily computed.

3.4 Simulation Results

We consider a Wi-Fi scenario in which the AP is assumed to be the spectrum operator who manages a total of M radio bands of equal bandwidth. Before sending data, each wireless user must send a message to the AP, requesting for transmission of a certain of application class ,the associated bandwidth requirements and the price it is willing to pay. Application class specifies the priority of their transmissions which can translate to interactive, streaming, or web-browsing traffic. For simplicity, we assume that the requested amount of bandwidth is always an integer number of bands. Based on the

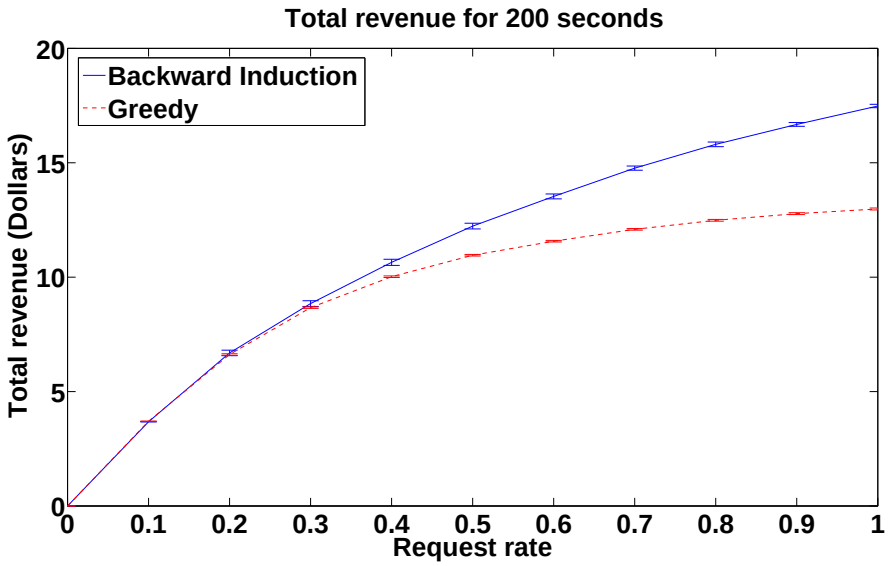
Table 3.1: Parameters for the finite-horizon model

Number of application classes	3
Capacity (number of radio bands M)	3
Price user pays (cents/sec· Mhz)	$\{1, 3, 5\}$
Cost to maintain transmission	$\{1, 1, 2\}$
Quit rate	$\{0.1, 0.15, 0.2\}$

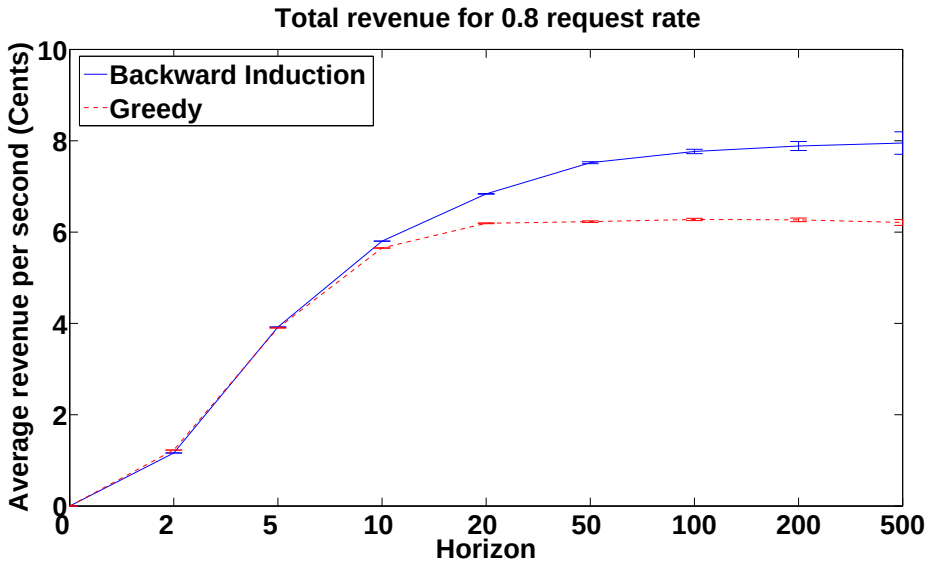
request and current system state, the AP decides to accept the request or not. If the offer is accepted, the spectrum operator will gain a revenue equal to the product of the price and the time the user staying in the system. Otherwise, the spectrum operator receives no additional revenue. The average time of a user staying in the system depends on the user's application. For example, the average time for a video application tend to be longer than a web transaction. Thus, we model the average time of an application by assuming that there is a probability q_i that a user having application class i will quit and leave the system at any time step. Specific values for these parameters in the simulations are shown in Table 3.1. For simplicity, we assume the cost of transmissions take on three possible discrete values. These are computed and normalized based on typical SNR of Wi-Fi networks.

In this simulation, we assume there are five different classes, denoted by number 1 to 3. There is a total of three radio bands, and thus that there are at most three simultaneously active users. For simplicity, every user pays a fixed price throughout time, which unit is cent per second per Mhz. For simplicity, we also assume that same classes of user cost same amount of profit for AP to maintain the transmission. This assignment makes sense since as the user has a better price offer stays in system shorter, the AP should pick up this property by accepting more that kind of class of users to maximize the revenue.

Figure 3.1(a) shows the revenue of the spectrum operator over 200 time steps as a function of rates of request arrivals for the BI and greedy algorithms. Using the greedy algorithm, the AP greedily accepts the request if the system is not fully occupied. As seen, when the requesting rates are low, the greedy algorithm aiming to utilize as many bands as possible, performs similar to the optimal algorithm due to the fact that there are always many available bands anyway. In other words, optimization does not matter



(a) Total revenue for different request rate



(b) Average revenue for different time step

Figure 3.1: Finite Horizon with Backward Induction

in this case. However, when the requesting rate increases, a more careful optimization is needed to avoid accepting low revenue users and filling up the spectrum. Thus in this

case, the optimal policy outperforms the greedy policy significantly. We also note that the larger requesting rates are, the larger the revenue gaps between the optimal and greedy algorithms.

Figure 3.1(b) shows the revenue per time step as a function of the size of the optimization horizon, assuming a fixed requesting rate at 0.8. For the optimization horizon of size one, the greedy and optimal algorithms are identical, and thus they produce the same revenue. However, as the optimization horizon increases, the optimal policy performs much better due to having more opportunities for optimization.

Chapter 4: Revenue Based Pricing Protocol via Q-Learning

In previous chapter, we study the pricing scheme problem assuming that the system parameters such as the transition probability matrix are known. In the real-world, this is often not the case. Therefore, in this chapter, we propose a Q-learning[18] approach to learn the optimal policy directly from the simulations. Q-learning approach is useful where it is complex to analytically derive the system parameters. In addition, we will assume a discount infinite-horizon MDP model for our study. The infinite-horizon model aims to model the long-term revenue with future discount.

To do so, we also simplify the state of the MDP. Such simplification no longer allows us to analytically derive the transition probability. On the other hand, we no longer need the system parameters since the optimal policy can be found directly through simulations using Q-learning.

Also, as a preview of our result, we show that the policy will have a special structure that can be compactly represented via Support Vector Machine (SVM)[16] which will be discussed subsequently. This result is also a precursor to our theoretical analysis on the conditions for optimality of a *thresholding* policy for our pricing scheme model in Chapter 5.

4.1 System Model

The setting used in this section is identical to the model that we state in Chapter 3. The difference only lies in the state modeling, the cumulative reward function, and the solution approach.

4.2 MDP Formulation

We note that the immediate reward r_n is observed at every time step and is given to the agent by environment. The action a_n at every time step is typically selected in a greedy manner. Importantly, for our spectrum management the observations (states)

are intentionally reduced significantly. Specifically, instead of using the detail states as described in Section 3.2, we simplify the states to be:

$$s_n = [(c_n, snr_n), u_n, v_n],$$

where s_n denotes the state, (c_n, snr_n) denotes the application/user request type and its SNR value, u_n denotes the amount of spectrum being used, and v_n denotes the total revenue paid by all users in system, at time step n. This simplification is necessary in order to handle a system having large capacity. To see why, assuming a finite-horizon model with M radio bands for access, then the number of possible states is $O(N^M)$, on the contrary, if we keep track of just the used radio bands, then the number of states is only $O(M)$.

We can see that the reward and control action space remain the same as in previous chapter. However, by using such model, we are not able to track the transition probability any more since we have lost the information from the original states. Therefore, we need an algorithm such that it can learn the approximate optimal policy through the simulation without knowing the transition probability. Reinforcement learning(RL)[4, 15] is one of such learning algorithms.

4.3 Solution Approach: Q-Learning

Reinforcement learning (RL)[15] approach allows one to learn the optimal policy directly via simulations. In addition to learning the optimal policy without knowing the transition probability, RL algorithms also employ simulations which enable them to consider only the probable states, reducing the need of exploring all the possible states as required by the BI algorithm. Q-Learning[18] is one of the most well-known reinforcement learning algorithm, and is the algorithm under investigation in this paper.

Q-learning model consists of an agent who acts on the environment based on its observations. For each agent's action, a reward is given to the agent, and the environment is changed based on its action. The goal of the agent is to maximize the total reward by taking an appropriate action based on its current observation. Often, the transition probability that characterizes a complex environment is difficult to obtain analytically. Thus, the environment is typically simulated, and the reward from the simulator is given to the agent directly. This setup allows the Q-learning algorithm to find an optimal

policy in an off-line manner via simulations. Once the optimal policy is found via simulations, it can be used online for the real world scenarios, providing that the real-world is similar to the simulations. In certain situations, the Q-learning algorithm can learn the approximately optimal policy quickly and directly from the real world without the need of running simulations offline. That said, the Q-learning algorithm finds the optimal policy by iterative updating the Q function until it converges. Specifically, the Q function is updated according to:

$$Q(s_n, a_n) \leftarrow (1-\alpha)Q(s_n, a_n) + \alpha[r_{n+1} + \gamma(\max_{a_{n+1}} Q(s_{n+1,a_{n+1}})], \tag{4.1}$$

where α denotes the learning rate and γ denotes the discount factor. A larger α value allows the algorithm to learn the policy quickly, but might not be optimal. A larger discount factor γ weighs the future reward more importantly.

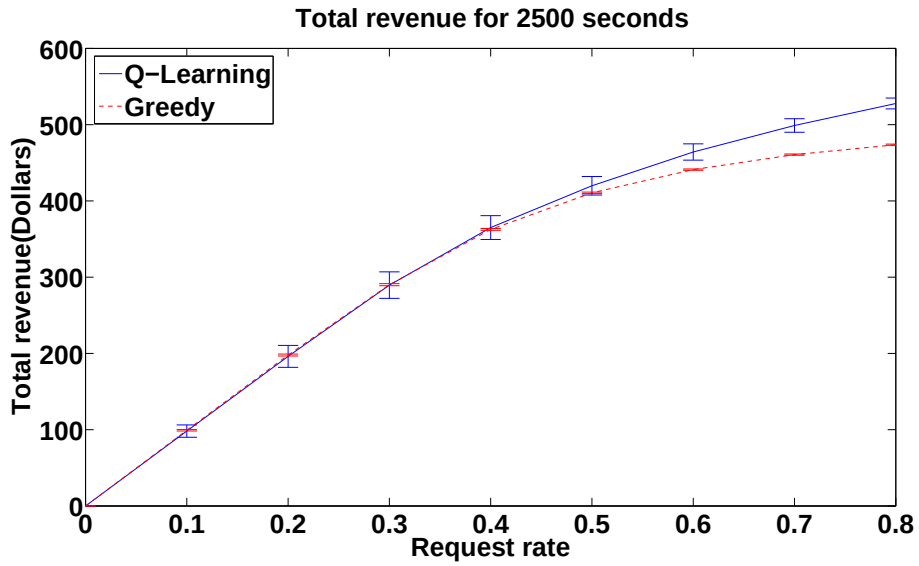
We note that since the Q-learning algorithm is used to maximize the reward for the infinite-horizon model, after running the simulations sufficiently long, the update Equation (4.1) will produce the approximately maximum value of $Q(s_n, a_n)$ corresponding to the approximately optimal pair (s_n, a_n), i.e., the approximately optimal price schemes.

4.4 Simulation Results

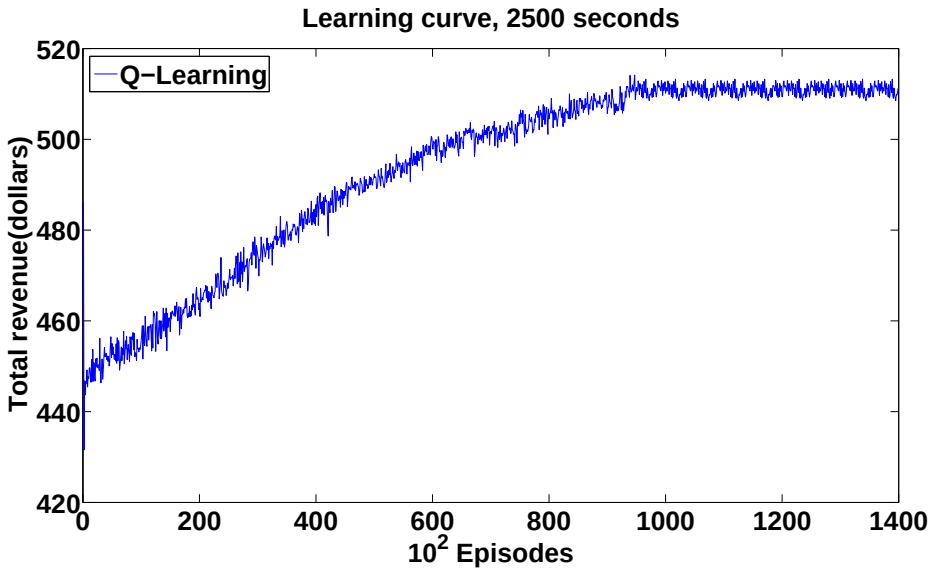
Table 4.1: Parameters for Infinite-Horizon MDP

Number of Class	5
Capacity	10
Price user pays (cents/sec· Mhz)	$\{1, 3, 5, 7, 9\}$
Quit rate	$\{0.1, 0.15, 0.2, 0.2, 0.3\}$
Cost to maintain transmission	$\{1, 1, 2, 3, 4\}$
Learning Rate	1/# of states
Discount Factor	0.99

We now show the results for Q-learning algorithm as applied to the infinite-horizon MDP. The parameters for the simulations shown in Table 4.1 consists of all the parameters in the finite-horizon model, and two more parameters: discount rate and learning rate that are specific to the infinite-horizon MDP and Q-learning algorithm. Figure 4.1(a)



(a) Total revenue for different request rate



(b) Learning curve for Q-Learning

Figure 4.1: Infinite Horizon with Q-Learning and Greedy Algorithm

shows the revenue per time step generated by the Q-learning and greed algorithms as functions of the arrival rate of requests. As seen, the Q-learning algorithm outperforms

the greedy algorithm significantly when there is a high access demand which provides more opportunities for optimization. Figure 4.1(b) shows the revenue per time step as a function of the size of the optimization horizon, assuming a fixed requesting rate at 0.9.

From the 4.1(a) we show that same as finite horizon case, when the request rate is low, the greedy algorithm is just the optimal policy, so that Q-learning algorithm can not out perform that. However when the request rate increases there is more "space" for Q-learning to learn the optimized policy, as a result it out performs greedy algorithm.

We note that as Q-learning is based on the simulations. The longer simulation time produces more accurate results. In Fig. 4.1(b), we show that the convergence of Q-learning algorithm to the optimal solutions. We switch off the learning magnesium after every 100 iterations and evaluate the policy it learned. As seen, it is the result after 100,000 iterations, Q-learning achieves reasonably good policies.

Though this MDP has far fewer states than the one in the finite-horizon model, it still takes a long time to run due to simulations. In addition, as the number of user class and price range increases, the state table also increases. As a result, it takes a large amount of time to go through the table to obtain the action for each state. Therefore, one of challenges of using MDP policy is how represent the them compactly for real-world scenarios. A naive approach would be a table-based representation which lists out all the possible states. Associated with each state is a correspond optimal action. As shown previously, our states are discretized and thus smaller quantization of can lead a large number of possible states. This approach requires much memory. Thus, we investigate a different approach in which the policies are represented implicitly via a function. Specifically, since our action space consists of only two possible actions: accept or reject, we will use support vector machine (SVM) [16] to compactly represent our policies.

SVM is typically employed in machine learning to classify whether an instance is belonged to one class or the other based on its feature vector. It does so by determining a plane that separates between the instances that belong to different classes. The coefficients in the plane is determined via training using a set of labeled instances.

To apply SVM to our problem, we obtained the optimal policies using Q-learning using a certain level quantization on the states, e.g., SNR, application classes. These quantization levels of the states, together with the resulted optimal actions are the labeled instances as inputs to the SVM. Using these inputs, SVM produces a plane that

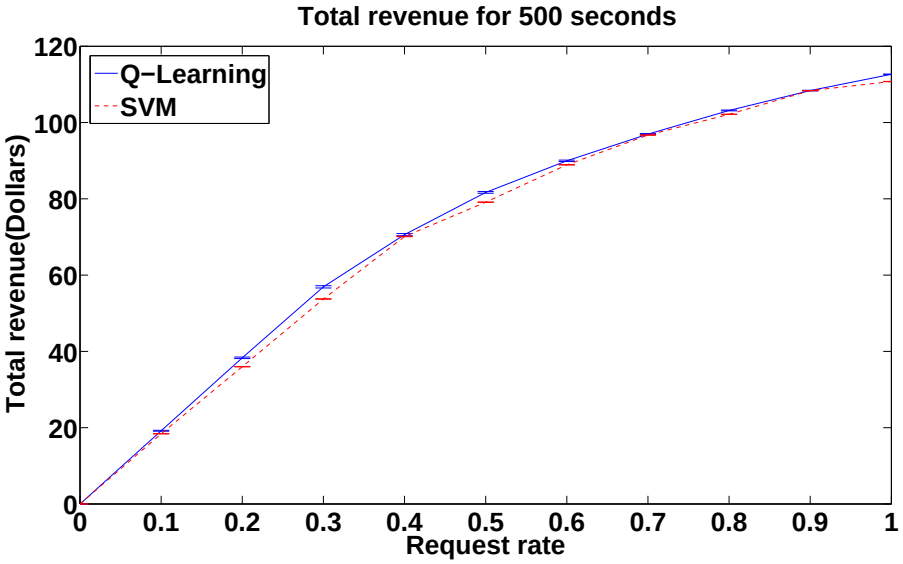
separate the states that the spectrum owner would accept from those that would be rejected. Since the SVM plane can be compactly represented by its coefficient, there is no need to use a large table for storing the policy. Instead, the states can be directly input into the SVM plane. If the result is above the plane, the action is accept. Otherwise, the action is rejected.

Importantly, the SVM approach can now be used to determine the actions for the cases where the states are not discretized. For example, during training phase, SVM uses discrete data for training, but during the real-time operational phase, the observed states can be directly input into the SVM. The result is to accept or to reject based on the classification of the observed state. In a way, SVM predicts the optimal actions for continuous states based on the discrete states based on the training phase.

Fig. 4.2(a) shows that based on part of information from state, we are able to draw a line that separates states with policy accept and policy reject. Fig. 4.2(b) shows that the predicted performance of the SVM is not much different from that of Q-learning for different requesting rates. This implies that not only SVM allows us to represent the policies compactly, but it also allows us to predict the optimal policies based on the states that have not been used for training.

By the observation of SVM, we consider a more efficient MDP model for real-life scenarios with monotone policy.

(a) Categorization line from SVM



(b) Policy comparison with Q-Learning

Figure 4.2: Support vector machine result

Chapter 5: Optimal Monotone Policy for Pricing Protocol

Based on the existence of the structured optimal policy via SVM as empirically supported by SVM, we investigate the conditions for which one obtain the optimal policy. Specifically, we show that under some mild conditions, the optimal pricing protocol will have thresholding structure. Thresholding policy is a special case of the class of monotone policies which are well explored in existing literature. Our contribution is the recognition of the thresholding policy for the proposed pricing protocol under a certain assumption. It is noted the thresholding policy lends itself to much more efficient implementation of algorithmic solutions, e.g., as compare to the standard solutions such as policy or value iteration algorithms.

5.1 Monotone Policy

A monotone policy is defined as follows. Let $d_t(s)$ be the decision rule at time t with state s, then the decision rules has the form of

$$d_t(s) = \begin{cases} a_1 & s < s^* \\ a_2 & s \geq s^*, \end{cases}$$

where a_1 and a_2 are distinct actions and s^* is a control limit. Monotone policy[14, 2] allows immediate feed back in real life scenarios with just calculation of a function instead of searching through the whole policy table. Therefore, it gives a very efficient way for implementing the MDP with large state space. A thresholding policy, as will be described later, is a type of monotone policy that when all the state variables except one are fixed, the optimal policy is monotone with respect to that remaining state variable.

5.2 System Model

This system model is similar to the ones defined in Chapter 3 and 4, but we make a few assumptions in order to meet the requirement for optimal monotonic policy.

- There are N class of users shares Mhz bandwidth and we assume that N and M are large enough, if necessary.
- The class j user pays price $R(j)$/hz·second for transmitting and $R(j)$ is non-decreasing in j. Also we use class 0 denote there is no user, hence $R(0) = 0$.
- There is a cost function $C(i)$ related to occupancy i, where $i \in [0, M]$, which is **Strictly** increase in i. We assume that $C(0) = 0$ and $C(\infty) = \infty$, if necessary.
- For arrival, it follows a Bernoulli process, with parameter Γ and once there is a arrival, the probability for it is a class j request is γ_j and $\gamma_j = 0$. Then we have

$$\sum_{j=1}^{N} \gamma_j = \Gamma \text{ and } \gamma_0 = 1 - \Gamma$$

- For departure(quit), it also follows a Bernoulli process with parameter q, which is same for all the user in the system. In order to avoid trivial solution, we can assume that q is a function of occupancy, denote as $q(j)$, if necessary.

5.3 MDP Formulation

- State: we only two elements in our state, that is

$$s = \begin{bmatrix} j \\ i \end{bmatrix}$$

  where j is the requesting user class and i is the occupancy. We assume that system knows the function $R(j)$, which allows system to have the knowledge that the price is paid by requesting user j.
- Action Set: $A = \{0, 1\}$, where 0 is reject and 1 is accept. We assume that, if there is no request or system is full, we can only take action 0.
- Decision Epoch: Since we run on a discrete time scale, a decision needs to be made on every time slot.

- Transition Probability: We denote $[j_1, i_1]$ as the current state and $[j_2, i_2]$ as the new state. Then we have the transition probability from current state to new state by taking action a we have

$$p([j_2, i_2]|[j_1, i_1], a) = \begin{cases} \binom{i_1+a}{i_1-i_2+a} \cdot q^{|i_1-i_2+a|} \cdot (1-q)^{i_2} \cdot \gamma_{j_2} & i_2 \leq i_1 + a \\ 0 & \text{otherwise} \end{cases}$$

- Since every user in the current has same departure property, we can use expectation criterion to calculate the reward. At time t, state s with occupancy i and arrival user class j, by taking action a. Also if we also can assume that q is a function of price j, then we can have below:

$$r_t([j, i], a) = \begin{cases} [R(j) - (C(i+1) - C(i))] \cdot \frac{1}{q(j)} & \text{if } a = 1 \\ 0 & \text{if } a = 0 \end{cases}$$

  where $q(j)$ is non-increasing in j. Such model focuses on average reward during one user's transmission, in which a user either pays higher price to stay longer or offers a lower price for transmitting a small amount of time.

- Total reward through out the time is

$$E[\mathcal{R}] = \sum_t r_t(s, a)$$

5.4 Solution Approach

In [14], gives conditions for a MDP has an optimal monotone policy. In this section, we list all the conditions of optimal monotone policy for finite-horizon problems and give proof that our model meet all such requirements, hence it has an optimal monotone policy. In Appendix B, we show the proof that such conditions guarantee monotone policy is optimal for certain MDPs.

First we give the notations and definitions for MDP used in this section.

i. S is the state space and it is partially ordered,

ii. $\mathcal{A}$ is the action space and A_{s_t} be the actions is valid for state s at time t,

iii. $p(s_2|s_1, a)$ is the one-step transition probability from s_1 to s_2 by taking action a,

iv. $r_t(s, a)$ is the one-step reward in state s if action a is taken at time t.

Condition for Optimal Monotone Policy

Now we give the condition for a MDP has an optimal monotone policy, proof of its correctness is given in Appendix B

Theorem 1. *Suppose for $t = 1, ..., N-1$ that*

1. *$r_t(s, a)$ is non-decreasing in s for all $a \in A_{s_t}$,*
2. *$q_t(k|s, a)$ is non-decreasing in s for all $k \in \mathcal{S}$ and $a \in A_{s_t}$,*
3. *$r_t(s, a)$ is superadditive(subadditive) function on $\mathcal{S} \times A_{s_t}$.*
4. *$q_t(k|s, a)$ is superadditive(subadditive) function on $\mathcal{S} \times A_{s_t}$ for all $k \in \mathcal{S}$, and*
5. *$r_N(s)$ is non-decreasing in s.*

Then there exist optimal decision rules $d_t(s)$ which are non-decreasing(non-increasing) in s for $t = 1, ..., N-1$.

Proof of Existing Optimal Monotone Policy

Here we give the proof that proposed MDP in Section 5.3 has an optimal monotone policy. We first prove the condition related with $r_t(x, a)$, which is condition $1, 3, 5$; then we show the condition related with transition probability, which is condition $2, 4$. Here we denote our fixed i as i^*.

Condition 1 $r_t(s, a)$ needs to be non-decreasing in s. As stated before

$$r_t([j, i^*], a) = a \cdot \{R(j) - [C(i^* + 1) - C(i^*)]\} \cdot \frac{1}{q(j)}, \text{ where } a \in A$$

- If $a = 0$, $r_t([j, i^*], a) = 0, \forall [j, i] \in \mathcal{S}$, so it is non-decreasing.
- If $a = 1$, since we fix i, the state space is immediately ordered in terms of j. Also $R(j)$ is non-decreasing and and $q(j)$ is non-increasing in j. Therefore $\{R(j) - [C(i^* + 1) - C(i^*)]\} \cdot \frac{1}{q(j)}$ is non-decreasing in j.

Condition 3 $r_t(x, a)$ needs to be super-additive. Since super-additive means that for $x^+ \geq x^-$ and $y^+ \geq y^-$, we need to have that

$$g(x^+, y^+) - g(x^+, y^-) \geq g(x^-, y^+) - g(x^-, y^-)$$

Since for us we only have two actions in A, therefore we just need to prove that

$$r_t(s, 1) - r_t(s, 0)$$

is non-decreasing in s. This should be straight forward, because we have that $r_t([j, i^*], 0) = 0$. Suppose that $[j_1, i^*] \preccurlyeq [j_2, i^*]$, we have

$$r_t([j_1, i^*], 1) - r_t([j_1, i^*], 0) - [r_t([j_2, i^*], 1) - r_t([j_2, i^*], 0)] = \frac{R(j_1)}{q(j_1)} - \frac{R(j_2)}{q(j_2)} \leq 0$$

Therefore we have that $r_t([j, i^*], a)$ is super-additive.

Condition 5 $r_T(s)$ needs to be non-decreasing in s, where T is the total horizon(the last time slot of MDP). Since this is the last second of MDP, we just let

$$r_T([j, i^*]) = 0, \forall j$$

Hence it is immediately non-decreasing.

Condition 2 $q_t(k|s, a)$ needs to be non-decreasing in x for all $k \in \mathcal{S}$ and $a \in A$, where $q_t(k|s, a)$ is the probability that in state s by taking action a, it exceeds k.

$$q_t(k|s, a) = \sum_{m=k}^{\infty} p_t(m|s, a)$$

For a fixed i^*, the order only depends on j. That is, it orders j from lowest to highest, since $R(j)$ is strictly increasing in j and $q(j)$ is non-increasing in j. Suppose that we have $[j_1, i^*] \quad [j_2, i^*]$ and $k = [j_k, i_k] \in \mathcal{S}$. Then we can have that

$$\begin{aligned}
q_t([j_k, i_k]|[j_1, i^*], 0) &= \sum_{m=k}^{\infty} p_t(m|[j_1, i^*], 0) = \sum_{l=j_k}^{N} \gamma_l \cdot \binom{i^*}{i^* - i_k} \cdot q^{|i^* - i_k|} \cdot (1-q)^{i_k} \\
&= q([j_k, i_k]|[j_2, i^*], 0) \\
q_t([j_k, i_k]|[j_1, i^*], 1) &= \sum_{m=k}^{\infty} p_t(m|[j_1, i^*], 1) = \sum_{l=j_k}^{N} \gamma_l \cdot \binom{i^*+1}{i^*+1 - i_k} \cdot q^{|i^*+1 - i_k|} \cdot (1-q)^{i_k} \\
&= q([j_k, i_k]|[j_2, i^*], 1)
\end{aligned}$$

we can see that $q_t(k|x, a)$ is non-decreasing in x for all $k \in X$ and $a \in A$.

Condition 4 $q_t(k|s,a)$ needs to be super-additive function on $\mathcal{S} \times A$ for all $k \in \mathcal{S}$, which means

$$q_t(k|s,1) - q_t(k|s,0)$$

needs to be non-decreasing in x. By following **condition 2**, we can easily reached conclusion, since

$$\begin{aligned} &q_t([j_k,i_k]||[j_1,i^*],1) - q_t([j_k,i_k]||[j_1,i^*],0) \\ &- [q_t([j_k,i_k]||[j_2,i^*],1) - q_t([j_k,i_k]||[j_2,i^*],0)] = 0 \quad \forall [j_1,i^*],[j_2,i^*] \in \mathcal{S} \end{aligned}$$

Therefore we proved condition 4 for a fixed i^*.

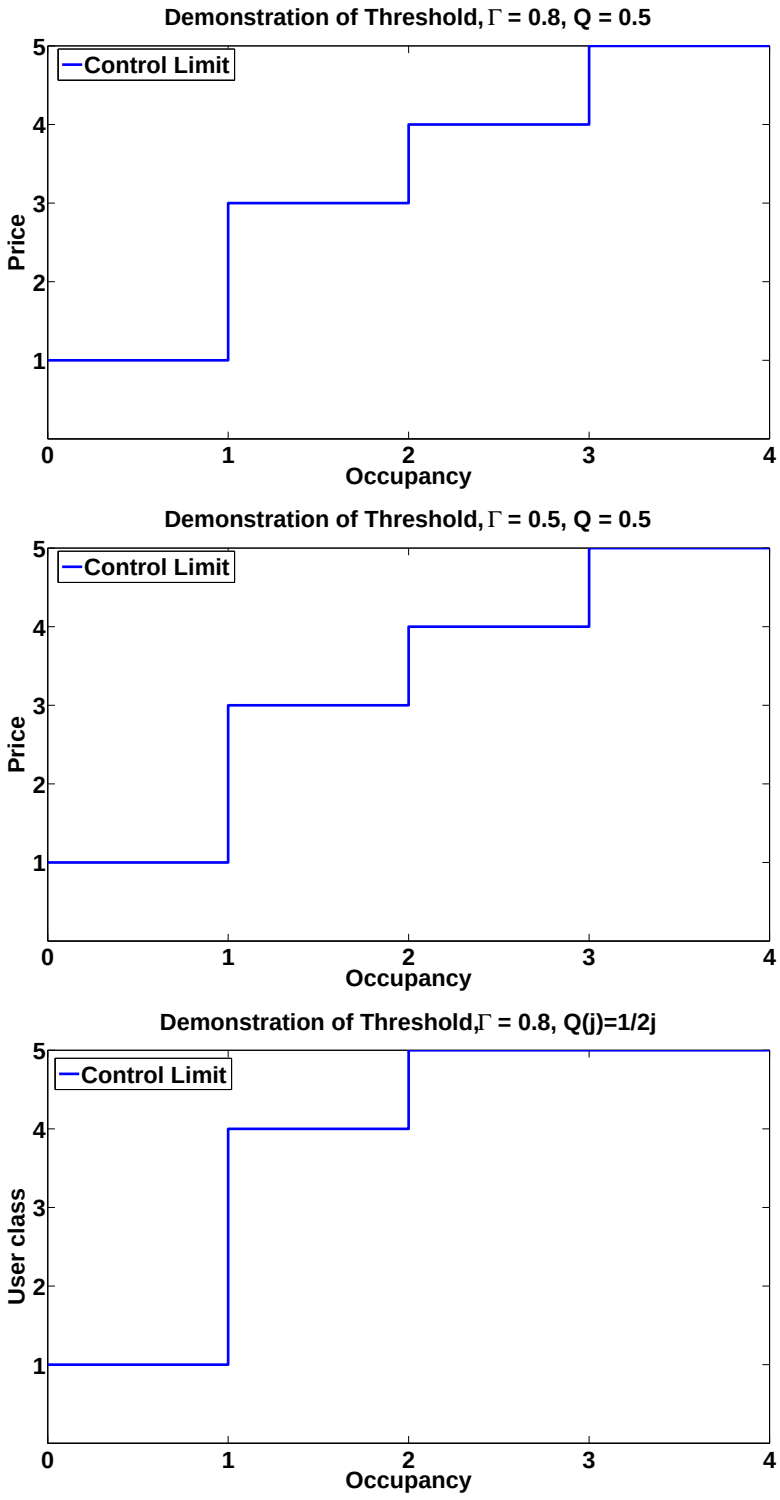
Conclusion: From above we can see that for a fixed i^* we can have an optimal monotone policy in j. Therefore for the space $[0,M] \times [0,N]$ we are able to curve the space into two parts to obtain the threshold curve.

5.5 Simulation Results

We first use backward induction to find the optimal policy for the MDP we defined in Section 5.3. In simulation we have 5 class of users and 4hz bandwidth for share, which means $M = 4$ and $N = 5$. We set $R(j) = j$ and $C(i) = 1.2i$, therefore they both strictly increase and $C(j)$ increase faster than $R(j)$. The arrival rate is 0.8 and it is uniformly distributed among all the users. Also departure rate for each user in the system is 0.1.

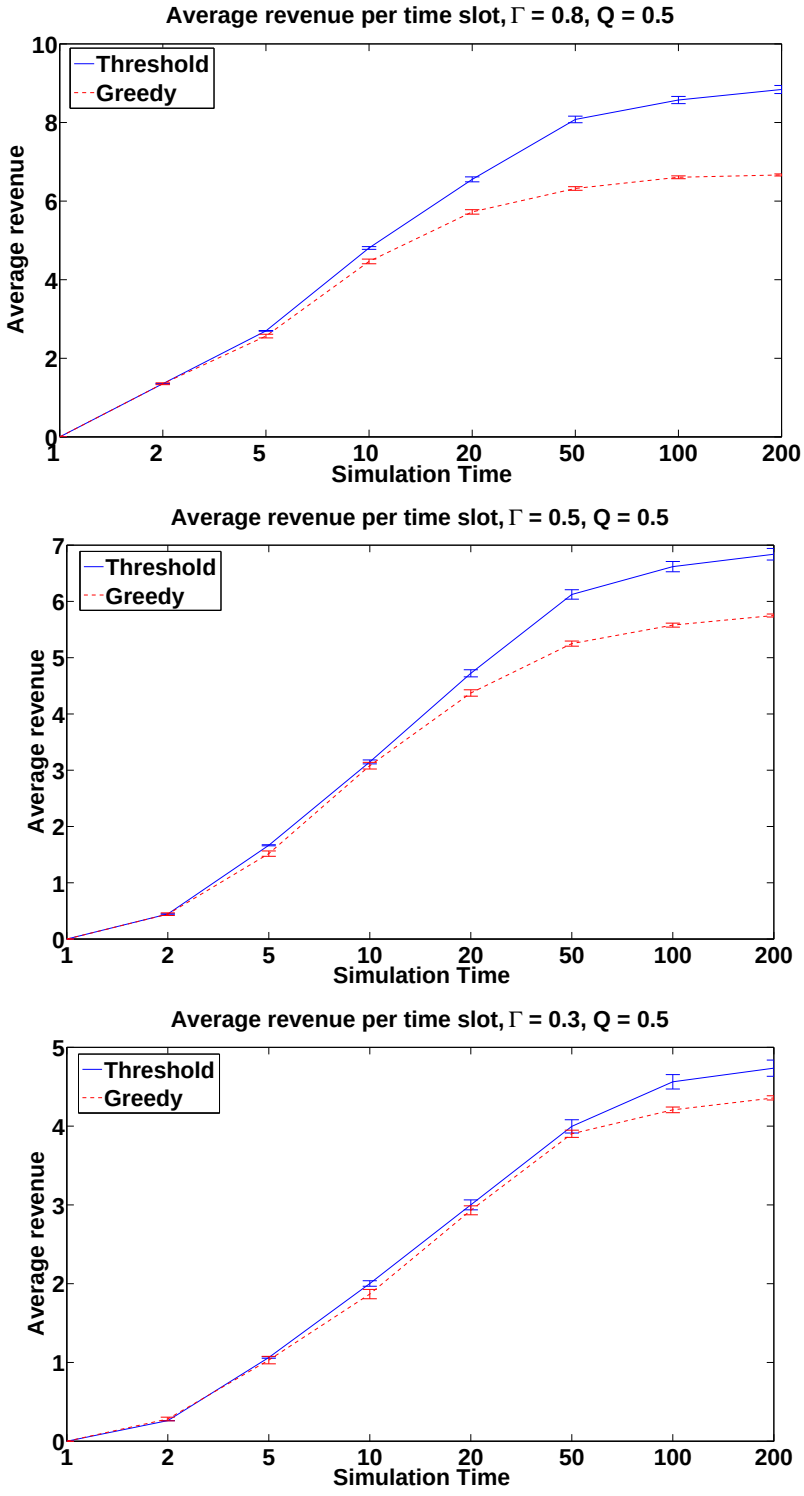
In Fig. 5.1, we can see there is a control limit for every occupancy. The users greater than such limit is accepted while users less than the limit is rejected. By linking, all those threshold together, we can the whole state space is divided into two parts. The upper part takes action accept and the lower part takes action reject. Since this policy is returned by backward induction, it is must be optimal(see Appendix A). Therefore, it also proves that there exists an optimal policy that is monotone. We also show that for different arrival and departure rate setup, the threshold holds. Additionally, we show a graph that, if the service rate is a function of price, the control limit changes accordingly. Such change meets our assumption in the first place, which is the system favors higher average reward during one user's transmission.

In Fig. 5.2, we compare the monotone policy obtained by backward induction to the greedy policy. It shows that our policy is much better than the greedy policy and also as fast as greedy policy. Because we only need to compare the user with the control limit, which gives almost instant feedback. We also show that the threshold policy exceeds greedy policy in different scenarios. In Fig. 5.2, we can see that as the arrival rate decreases, the improvement from greedy policy to threshold policy also decreases. Because, there are more available transmitting resource than the request from users.



Threshold of occupancy

Figure 5.1: Monotone policy



Policy comparison with greedy

Figure 5.2: Monotone policy and greed policy

Chapter 6: Conclusion

In this work, we study the problem on how a spectrum operator can optimally allocate its limited spectrum resources by managing spectrum among different classes of users. We show that the problem of maximizing the revenue for the spectrum operator can be cast as an MDP problem.

We investigate two formulations of MDP: the finite-horizon and discounted infinite horizon models. We show that for small scenarios, the backward induction algorithm is feasible and produces the optimal solution for the finite horizon MDP. For larger scenarios, Q-learning is used to approximate the optimal solution for the discounted infinite horizon. We also show how to use SVM to compactly represent and predict good policies. For real-life condition, which needs efficient policy determination, we establish a model that has an optimal monotone(threshold) policy. Such model returns policy just by comparing the request and control limit, as a result, it gives almost immediate feed back on policy.

Our simulation results show that the obtained MDP-based schemes are able to generate more revenue than that of the greedy algorithm, especially when there is a high demand for wireless access for all three scenarios that we discuss in this work.

Bibliography

[1] Ian F Akyildiz, Won-Yeol Lee, Mehmet C Vuran, and Shantidev Mohanty. Next generation/dynamic spectrum access/cognitive radio wireless networks: a survey. *Computer Networks*, 50(13):2127–2159, 2006.

[2] Eitan Altman and Shaler Stidham Jr. Optimality of monotonic policies for two-action markovian decision processes, with applications to control of queues with delayed information. *Queueing Systems*, 21(3-4):267–291, 1995.

[3] Yochai Benkler. Overcoming agoraphobia: building the commons of the digitally networked environment. *Harv. JL & Tech.*, 11:287, 1997.

[4] G.W. Evans and S. Honkapohja. *Learning and expectations in macroeconomics*. Princeton Univ Pr, 2001.

[5] FCC Spectrum Policy Task Force. Report of the spectrum efficieny working group. In *Online. Available at http://www.fcc.gov/sptf/reports.html*, 2002.

[6] Lin Gao, Xinbing Wang, Youyun Xu, and Qian Zhang. Spectrum trading in cognitive radio networks: A contract-theoretic modeling approach. *Selected Areas in Communications, IEEE Journal on*, 29(4):843 –855, april 2011.

[7] Dale N Hatfield and Philip J Weiser. Property rights in spectrum: Taking the next step. In *New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on*, pages 43–55. IEEE, 2005.

[8] Thomas W Hazlett. Assigning property rights to radio spectrum users: Why did fcc license auctions take 67 years?*. *The Journal of Law and Economics*, 41(S2):529–576, 1998.

[9] Zhu Ji and K.J.R. Liu. Cognitive radios for dynamic spectrum access - dynamic spectrum sharing: A game theoretical overview. *Communications Magazine, IEEE*, 45(5):88 –94, may 2007.

[10] Long Bao Le and Ekram Hossain. Resource allocation for spectrum underlay in cognitive radio networks. *Wireless Communications, IEEE Transactions on*, 7(12):5306–5315, 2008.

[11] William Lehr and Jon Crowcroft. Managing shared access to a spectrum commons. In *New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on*, pages 420–444. IEEE, 2005.

[12] D. Niyato and E. Hossain. Competitive pricing for spectrum sharing in cognitive radio networks: Dynamic game, inefficiency of nash equilibrium, and collusion. *Selected Areas in Communications, IEEE Journal on*, 26(1):192 –202, jan. 2008.

[13] D. Niyato and E. Hossain. Competitive pricing for spectrum sharing in cognitive radio networks: Dynamic game, inefficiency of nash equilibrium, and collusion. *Selected Areas in Communications, IEEE Journal on*, 26(1):192 –202, jan. 2008.

[14] Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1994.

[15] R.S. Sutton and A.G. Barto. *Reinforcement learning: An introduction*, volume 1. Cambridge Univ Press, 1998.

[16] Johan AK Suykens and Joos Vandewalle. Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300, 1999.

[17] Bin Wang and Dongmei Zhao. Performance analysis in cdma-based cognitive wireless networks with spectrum underlay. In *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, pages 1–6. IEEE, 2008.

[18] ChristopherJ.C.H. Watkins and Peter Dayan. Q-learning. *Machine Learning*, 8:279–292, 1992.

[19] Hyenyoung Yoon, Junseok Hwang, and Martin B.H. Weiss. An analytic research on secondary-spectrum trading mechanisms based on technical and market changes. *Computer Networks*, 56(1):3 – 19, 2012.

[20] Qing Zhao and Brian M Sadler. A survey of dynamic spectrum access. *Signal Processing Magazine, IEEE*, 24(3):79–89, 2007.

APPENDICES

Appendix A: Proof for Optimality of Backward Induction

In the appendix, we show that the policy returned by backward induction in optimal. There are two main kinds of policy in MDP, deterministic and random. Since in this work, the system proposed uses a deterministic policy, the proof given here assumes policy is deterministic. Readers who are interested in random policies can refer to [14] for details.

Definition 1. *Let H_t be the history at time t and*

$$u_t(h_t) = \sup_{a \in A_{s_t}} \left\{ r_t(s_t, a) + \sum_{j \in \mathcal{S}} p_t(j|s_t, a) u_{t+1}(h_t, a, j) \right\} \tag{A.1}$$

for $t = 1, ..., N-1$ and $h_t = (h_{t-1}, a_{t-1}, s_t) \in H_t$ be the optimal function. For $t = N$, we add the boundary condition

$$u_N(h_N) = r_N(s_N)$$

for $h_N = (h_{N-1}, a_{N-1}, s_N) \in H_N$. If the supremum in (A.1) is attained, it can be replaced by "max", that is

$$u_t(h_t) = \max_{a \in A_{s_t}} \left\{ r_t(s_t, a) + \sum_{j \in \mathcal{S}} p_t(j|s_t, a) u_{t+1}(h_t, a, j) \right\}$$

Lemma 1. *Let w be a real-valued function on an arbitrary discrete set W and let $q(\cdot)$ be a probability distribution on W. Then*

$$\sup_{u \in W} w(u) \geq \sum_{u \in W} q(u)w(u)$$

Proof. Let $w^* = \sup_{u \in W} w(u)$. Then

$$w^* = \sum_{u \in W} q(u)w^* \geq \sum_{u \in W} q(u)w(u)$$

□

The theorem below shows the optimality properties of solutions of the optimal equation.

Theorem 2. *Suppose u_t is a solution of (A.1) for $t = 1, ..., N-1$, and u_N satisfies the boundary condition. Then*

i. $u_t(h_t) = u_t^*(h_t)$ *for all* $h_t \in H$, $t = 1, ..., N$, *and*

ii. $u_1(s_1) = v_N^*(s_1)$ *for all* $s_1 \in \mathcal{S}$

Proof. The author in [14] divides the proof into two parts. First, an induction for $u_n(h_n) \geq u_n^*(h_n)$ for all $h_n \in H_n$ and $n = 1, 2, ..., N$ is constructed. By the boundary condition, there is no decision is made, so that $u_N(h_N) = r_N(s_N) = u_N^\pi(h_N)$ for all $h_N \in H_n$ and $\pi \in \Pi$. As a result, we have $u_N(h_N) = u_N^*(h_N)$ for all $h_N \in H_N$. Now assume that $u_t(h_t) \geq u_t^*(h_t)$ for all $h_t \in H_t$ for $t = n+1, ..., N$. Let $\pi' = (d_1', d_2', ..., d_{N-1}')$ be an arbitrary policy in Π. For $t = n$, the optimal equation is

$$u_n(h_n) = \sup_{a \in A_{s_n}} \left\{ r_n(s_n, a) + \sum_{j \in \mathcal{S}} p_n(j|s_n, a) u_{n+1}(h_n, a, j) \right\}$$

By the induction hypothesis, we have

$$\begin{aligned} u_n(h_n) &\geq \sup_{a \in A_{s_n}} \left\{ r_n(s_n, a) + \sum_{j \in \mathcal{S}} p_n(j|s_n, a) u^*_{n+1}(h_n, a, j) \right\} \\ &\geq \sup_{a \in A_{s_n}} \left\{ r_n(s_n, a) + \sum_{j \in \mathcal{S}} p_n(j|s_n, a) u^{\pi'}_{n+1}(h_n, a, j) \right\} \\ &\geq \quad q_{d'_n(h_n)}(a) \left\{ r_n(s_n, a) + \sum_{j \in \mathcal{S}} p_n(j|s_n, a) u^{\pi'}_{n+1}(h_n, a, j) \right\} \\ &= u^{\pi'}_n(h_n) \end{aligned}$$

The first inequality holds because of the induction hypothesis and the non-negativity of p_n. The second inequality is based on **Lemma 1**, in which we let $W = A_{s_n}$ and w equal to the expression in brackets. Since π' is arbitrary, we have

$$u_n(h_n) \geq u^{\pi}_n(h_n) \quad \forall \pi \in \Pi$$

Thus $u_n(h_n) \geq u^*_n(h_n)$ and the induction hypothesis holds. Now we establish that for any $\epsilon > 0$, there exists a $\pi' in \Pi$for which

$$u^{\pi'}_n(h_n) + (N - n)\epsilon \geq u_n(h_n) \tag{A.2}$$

for all $h_n \in H_n$ and $n = 1, 2, ..., N$. In order to prove that we construct a policy $\pi' = (d_1, d_2, ..., d_{N-1})$ by choosing $d_n(h_n)$ to satisfy

$$r_n(s_n, d_n(h_n)) + \sum_{j \in \mathcal{S}} p_n(j|s_n, d_n(h_n)) u^{\pi'}_{n+1}(h_n, d_n(h_n), j) + \epsilon \geq u_n(h_n)$$

We show this by induction. Since $u^{\pi'}_N = u_N(h_N)$, the induction hypothesis holds for

$t = N$. Assume that $u_t^{\pi'}(h_t) + (N-t)\epsilon \geq u_t(h_t)$ for $t = n+1, ..., N$. Then we have

$$\begin{aligned} u_n^{\pi'}(h_n) &= r_n(s_n, d_n(h_n)) + \sum_{j \in \mathcal{S}} p_n(j|s_n, d_n(h_n)) u_{n+1}^{\pi'}(s_n, d_n(h_n), j) \\ &\geq r_n(s_n, d_n(h_n)) + \sum_{j \in \mathcal{S}} p_n(j|s_n, d_n(h_n)) u_{n+1}(s_n, d_n(h_n), j) - (N-n-1)\epsilon \\ &\geq u_n(h_n) - (N-n)\epsilon \end{aligned}$$

Thus the induction hypothesis is satisfied and (A.2) holds for $n = 1, 2, ..., N$. Therefore for any $\epsilon > 0$, there exists a $\pi' \in \Pi$ for which

$$u_n^*(h_n) + (N-n)\epsilon \geq u_n^{\pi'}(h_n) + (N-n)\epsilon \geq u_n(h_n) \geq u_n^*(h_n)$$

so that (i) follows and (ii) follows by the definition. □

Now we give the proof that the policy returned by backward induction is optimal

Theorem 3. *Suppose $u_t^*, t = 1, ..., N$ are solutions of the optimality equations, which supremum can be attained, subject to boundary condition and policy $\pi^* = (d_1^*, d_2^*, ..., d_{N-1}^*) \in \Pi$ satisfies*

$$\begin{aligned} & r_t(s_t, d_t^*(h_t)) + \sum_{j \in \mathcal{S}} p_t(j|s_t, d_t^*(h_t)) u_{t+1}^*(h_t, d_t^*(h_t), j) \\ = & \max_{a \in A_{s_t}} \left\{ r_t(s_t, a) + \sum_{j \in \mathcal{S}} p_t(j|s_t, a) u_{t+1}^*(h_t, a, j) \right\} \end{aligned}$$

for $t = 1, ..., N-1$. Then

1. *For each $t = 1, 2, ..., N$*

$$u_t^{\pi^*}(h_t) = u_t^*(h_t), \quad h_t \in H_t.$$

2. *π^* is an optimal policy, and*

$$v_N^{\pi^*}(s) = v_N^*(s), \quad s \in \mathcal{S}.$$

Proof. Here we prove (i), since (ii) follows from **Theorem 2**. We prove this theorem by induction. By definition, we have

$$u_N^{\pi^*}(h_n) = u_N^*(h_n), \quad h_n \in H_n.$$

Now assume the result holds for $t = n+1, ..., N$. Then, for $h_n = (h_{n-1}, d_{n-1}^*(h_{n-1}), s_n)$, we have

$$\begin{aligned} u_n^*(h_n) &= \max_{a \in A_{s_n}} \left\{ r_n(s_n, a) + \sum_{j \in \mathcal{S}} p_n(j|s_n, a) u_{n+1}^*(h_n, a, j) \right\} \\ &= r_n(s_n, d_n^*(h_n)) + \sum_{j \in \mathcal{S}} p_n(j|s_n, d_n^*(h_n)) u_{n+1}^{\pi^*}(h_n, d_n^*(h_n), j) \\ &= u_n^{\pi^*}(h_n) \end{aligned}$$

The second equality follows from the induction hypothesis and the condition assumed by theorem. Thus the induction hypothesis is satisfied and the result follows. □

Appendix B: Proof for Optimality of Monotone Policy

In this appendix, we gives proof of optimality of monotone policy for both finite-horizon and infinite-horizon. We use the notations that according to the original article, [14, 2].

B.1 Finite-Horizon

From [14], we know that only MDP with certain properties has an optimal decision rules with monotone form. In this section we give the detailed proof that, if a MDP meets certain conditions it has an optimal monotone policy.

Definition and Notation

First we give the notations and definitions for the MDP.

i. S is the state space and it is partially ordered,

ii. $\mathcal{A}$ is the action space and A_{s_t} be the actions is valid for state s at time t,

iii. $p(s_2|s_1, a)$ is the one-step transition probability from s_1 to s_2 by taking action a,

iv. $r_t(s, a)$ is the one-step reward in state s if action a is taken at time t.

Proof for Optimality

Definition 2. *Let*

$$u_t^*(s_t) = \max_{a \in A_{s_t}} \left\{ r_t(s, a) + \sum_{j \in S} p_t(j|s_t, a) u_{t+1}^*(j) \right\}$$

be the optimal value function for finite-horizon MDP.

We demonstrate optimality of monotone policies, by inductively showing that the optimal value functions from t onward, $u_t^*(s)$, are non-decreasing or non-increasing in S.

Then we show that

$$w_t(s,a) = r_t(s,a) + \sum_{j=0}^{\infty} p_t(j|s,a)u_t^*(j) \tag{B.1}$$

is superadditive or subadditive.

Remark 1. *For (B.1), superadditive(subadditive) means that*

$$w_t(s,a^+) - w_t(s,a^-)$$

is non-decreasing(non-increasing) in s, where $a^+.a^- \in A_{s_t}$ and $a^- < a^+$.

We can see that by proving (B.1) is superadditive(subadditive), we are able to conclude that there is an optimal policy for the finite-horizon problem, which is monotonically increasing(decreasing) in s.

In order to prove (B.1) is superadditive(subadditive), we first show some lemmas.

Lemma 2. *Let $\{x_j\}, \{x'_j\}$ be real-valued non-negative sequences satisfying*

$$\sum_{j=k}^{\infty} x_j \geq \sum_{j=k}^{\infty} x'_j$$

for all k, with equality holding for $k = 0$.

Suppose $v_{j+1} \geq v_j$ for $j = 0, 1, ...,$ then we can say

$$\sum_{j=0}^{\infty} v_j x_j \geq \sum_{j=0}^{\infty} v_j x'_j$$

where limits exists but maybe infinite.

Proof. Let k be arbitrary and $v_{-1} = 0$. Then

$$\begin{aligned}\sum_{j=0}^{\infty} v_j x_j &= \sum_{j=0}^{\infty} x_j \sum_{i=0}^{j} (v_i - v_{i-1}) = \sum_{j=0}^{\infty} (v_j - v_{j-1}) \sum_{i=j}^{\infty} x_i \\ &= \sum_{j=1}^{\infty} (v_j - v_{j-1}) \sum_{i=j}^{\infty} x_i + v_0 \sum_{i=0}^{\infty} x_i \geq \sum_{j=1}^{\infty} (v_j - v_{j-1}) \sum_{i=j}^{\infty} x'_i + v_0 \sum_{i=0}^{\infty} x'_i \\ &= \sum_{j=0}^{\infty} v_j x'_j.\end{aligned}$$

□

Lemma 2 gives us the ability to prove the optimal value function u_t^* is monotone. Now we provide condition for proving u_t^* is monotone.

Lemma 3. *By using u_t^* defined in **Definition 2.** and if we have*

1). $r_t(s, a)$ is non-decreasing(non-increasing) in s for all $k \in S$, $a \in A$, and $t = 1, ..., N-1$.

2). $q_t(k|s, a)$ is non-decreasing in s for all $k \in S$, $a \in A$, and $t = 1, ..., N-1$.

3). $r_N(s)$ is non-decreasing(non-increasing) in s.

Then $u_t^*(s)$ is non-decreasing(non-increasing) in s for $t = 1, ..., N$.

Proof. We use backward induction method to prove this lemma. We only show the non-decreasing case, since the proof of non-increasing case is identical. We know $u_N^*(s) = r_N^*(s)$ and by assumption $r_N^*(s)$ is non-decreasing, so base case hold.

Now, we assume that $u_n^*(s)$ is non-decreasing for $n = t+1, ..., N$. By assumption, there exists an $a_s^* \in A$ which attains the maximum in

$$u_t^*(s) = \max_{a \in A} \left\{ r_t(s, a) + \sum_{j=0}^{\infty} p_t(j|s, a) u_{t+1}^*(j) \right\}$$

so that

$$u_t^*(s) = r_t(s, a_s^*) + \sum_{j=0}^{\infty} p_t(j|s, a_s^*) u_{t+1}^*(j)$$

Now we let $s' \geq s$. By 1) and 2), the induction hypothesis, and Lemma 2 applied with $x'_j = p_t(j|s, a^*_s)$, $x_j = p_t(j|s', a^*_s)$ and $v_j = u^*_{t+1}(j)$, we have

$$u^*_t(s) \leq r_t(s', a^*_s) + \sum_{j=0}^{\infty} p_t(j|s', a^*_s) u^*_{t+1}(j)$$
$$\leq \max_{a \in A} \left\{ r_t(s', a) + \sum_{j=0}^{\infty} p_t(j|s', a) u^*_{t+1}(j) \right\} = u^*_t(s')$$

Thus we can say that $u^*_t(s)$ is non-decreasing. □

Now we give the proof of **Theorem 1** stated in **Section 5.4**, under which there exists monotone optimal policies.

Proof. Here we only show the superadditive case, which means $w_t(s, a)$, defined in (B.1), is superadditive when $q_t(j|s, a)$ and $r_t(s, a)$ are also superadditive. Let $s^- \leq s^+$ and $k \in S$ we have

$$\sum_{j=k}^{\infty} [p_t(j|s^-, a^-) + p_t(j|s^+, a^+)] \geq \sum_{j=k}^{\infty} [p_t(j|s^-, a^+) + p_t(j|s^+, a^-)]$$

From **Lemma 3**, we know that $u_t * (s)$ is non-decreasing in s for all t, so by applying **Lemma 2** gives us

$$\sum_{j=k}^{\infty} [p_t(j|s^-, a^-) + p_t(j|s^+, a^+)] u_t(j) \geq \sum_{j=k}^{\infty} [p_t(j|s^-, a^+) + p_t(j|s^+, a^-)] u_t(j).$$

Thus for each t, $\sum_{j=0}^{\infty} p_t(j|s, a) u_t(j)$ is superadditive.
From condition (3), $r_t(s, a)$ is superadditive and the sum of superadditive functions is superadditive. Thus we have $w_t(s, a)$ is superadditive. □

Lemma 4. *Suppose g is a superadditive function on $X \times Y$ and for each $x \in X$, $\max_{y \in Y} g(x, y)$ exists. Then*

$$f(x) = \max \left\{ y' \in \arg\max_{y \in Y} g(x, y) \right\}$$

is monotone non-decreasing in x.

Proof. Let $x^+ \geq x^-$ and choose $y \leq f(x^-)$. Then, by the definition of f,

$$g(x^-, f(x^-)) - g(x^-, y) \geq 0,$$

and by the definition of superadditive, we have

$$g(x^-, y) + g(x^+, f(x^-)) \geq g(x^-, f(x^-)) + g(x^+, y).$$

Then we rearrange the equation, we have

$$g(x^+, f(x^-)) \geq g(x^+, y) + \big[g(x^-, f(x^-)) - g(x^-, y)\big]$$

now we combine this with first inequality above, we have

$$g(x^+, f(x^-)) \geq g(x^+, y)$$

for all $y \leq f(x^-)$, Thus, $f(x^+) \geq f(x^-)$ □

Hence, we combine Lemma 4 and $w_t(s, a)$ is superadditive, we have monotone policy.

Remark 2. *There might exist other optimal policies which are not monotone.*

B.2 Infinite-Horizon

Similar to the finite horizon-problem, certain infinite-horizon problem also has a monotone optimal policy. [2] focuses on giving proof that under certain circumstances that infinite-horizon Markov decision problem with two actions has an optimal threshold policy. In this section, we give the proof for optimality of threshold policy in infinite-horizon MDP.

Notation and Definition

First we give the notations and definitions for our Markov decision process.

i. S is the state space and it is partially ordered,

ii. $A = \{0, 1\}$ is the action space,

iii. $p(s_2|s_1, a)$ is the one-step transition probability from s_1 to s_2 by taking action a,

iv. $r(s, a)$ is the one-step reward in state s if action a is taken,

v. β is the one-step discount factor, $0 \leq \beta < 1$.

Also we denote $\pi = \{\pi_0, \pi_1, ...\}$ is the policy, where π_t is the policy at time t.

Definition 3. *Let*

$$V_n^\pi(s) = E^\pi \left[\sum_{t=0}^{n-1} \beta^t r(S_t, A_t) | S_0 = s \right], s \in S, n \geq 1,$$

be the n-stage value function and when $n \to \infty$ we let

$$V^\pi(s) = E^\pi \left[\sum_{t=0}^{\infty} \beta^t r(S_t, A_t) | S_0 = s \right], s \in S,$$

be the value function for the infinite-horizon MDP.

Definition 4. *Let*

$$J_n(s, a) = r(s, a) + \beta E[V_n(S_1)|S_0 = s, A_0 = a]$$

and when $n \to \infty$ we let

$$J(s,a) = r(s,a) + \beta E[V(S_1)|S_0 = s, A_0 = a]$$

Remark 3. *In order to prove there is an optimal threshold policy for such MDP problem in infinite-horizon, we need to show that $J(s,a)$ is submodular(subadditive) in $S \times A$. It means that $J(s,0) - J(s,1)$ is non-decreasing in s. If such condition meets, we can conclude that the optimal policy for certain MDP is monotonically decreasing in s.*

Conditions

In [2], the author gives five conditions for proving $J(s,a)$ is an submodular function in $S \times A$. However, there is a shortcut given in the paper that one only needs three conditions to establish the proof. In this section, we list such three conditions, because our problem is qualified for taking the shortcut. This is the same as **Theorem ??**.

i. The transition probability is stochastically monotone, that is for $s_1 \geq s_2$ and $a \in A$,

$$E[f(S_1)|S_0 = s_1, A_0 = a] \geq E[f(S_1)|S_0 = s_2, A_0 = a],$$

where f is an arbitrary non-decreasing function $f : S \to \mathbb{R}$ for which the expectation is well-defined.

ii. $r(s,a)$ is submodular, which is $r(s,0) - r(s,1)$ is non-decreasing in s.

iii. If action 0 and 1 are permutable, that is

$$E[f(S_2)|S_0 = s, A_0 = 0, A_1 = 1] = E[f(S_2)|S_0 = s, A_0 = 1, A_1 = 0]$$

for all $s \in S$ and all functions $f : S \to \mathbb{R}$ such that the expectation is well-defined. Then we only need

$$E[r(s,0)+\beta r(S_1,1)|S_0 = s, A_0 = 0, A_1 = 1]-E[r_(s,1)+\beta r(S_1,0)|S_0 = s, A_0 = 1, A_1 = 0]$$

is non-decreasing in s.

Remark 4. *Since the goal for us is to prove $J_n(s,a)$ is submodular, which is $J_n(s,0) - J_n(s,1)$ is non-decreasing in s. If we denote*

$$\xi(s) = r(s,0) + \beta E[J_n(S_1,1)|S_0 = s, A_0 = 0] - r(x,1) - \beta E[J_n(S_1,0)|S_0 = s, A_0 = 1]$$

Later we show that if $J_n(s,a)$ is submodular and $\xi(x)$ is also submodular, then $J_{n+1}(s,a)$ is submodular. However, it is often hard to show that $\xi(s)$ is submodular. The assumption of condition iii gives us an easier to way to show that $\xi(s)$ is submodular. If we expand the expression of $\xi(x)$, we have

$$\begin{aligned}\xi(s) =& E[r(s,0) + \beta r(S_1,1) + \beta^2 V_n(S_2)|S_0 = s, A_0 = 0, A_1 = 1] \\ & - E[r(s,1) + \beta r(S_1,0) + \beta^2 V_n(S_2)|S_0 = s, A_0 = 1, A_1 = 0],\end{aligned} \tag{B.2}$$

then if we have action 0 and 1 are permutable, which is

$$E[f(S_2)|S_0 = s, A_0 = 0, A_1 = 1] = E[f(S_2)|S_0 = s, A_0 = 1, A_1 = 0]$$

Hence if we put the above equation back to (B.2), we only need that

$$E[r(s,0)+\beta r(S_1,1)|S_0 = s, A_0 = 0, A_1 = 1] - E[r(s,1)+\beta r(S_1,0)|S_0 = s, A_0 = 1, A_1 = 0]$$

is non-decreasing in s, which is condition iii.

Proof of Optimality

We show optimality of threshold holding policy by inductively showing $J_n(s,a)$ is submodular in $S \times A$.

Proof. Since

$$J_n(s,a) = r(s,a) + \beta E[V_n(S_1)|S_0 = s, A_0 = a]$$

i. For base case, we have condition ii, which is $r(s,a)$ is submodular. Hence $J_0(s,a)$ is submodular by assumption of $r(s,a)$.

ii. Let

$$\xi(s) = r(s,0) + \beta E[J_n(S_1,1)|S_0 = s, A_0 = 0] - r(x,1) - \beta E[J_n(S_1,0)|S_0 = s, A_0 = 1]$$

then we expand this expression

$$\begin{aligned}\xi(s) =& E[r(s,0) + \beta r(S_1,1) + \beta^2 V_n(S_2)|S_0 = s, A_0 = 0, A_1 = 1] \\ & - E[r(s,1) + \beta r(S_1,0) + \beta^2 V_n(S_2)|S_0 = s, A_0 = 1, A_1 = 0]\end{aligned}$$

By condition iii, we can say that $\xi(s)$ is non-decreasing. Now we assume that $J_n(s,a)$ is submodular, then we have

$$\begin{aligned}& J_{n+1}(s,0) - J_{n+1}(s,1) \\ =& r(s,0) + \beta E[V_{n+1}(S_1)|S_0 = s, A_0 = 0] - r(s,1) - \beta E[V_{n+1}(S_1)|S_0 = s, A_0 = 1] \\ =& r(s,0) + \beta E[\max\{J_n(S_1,0), J_n(S_1,1)\}|S_0 = s, A_0 = 0] \\ & - r(s,1) - \beta E[\max\{J_n(S_1,0), J_n(S_1,1)\}|S_0 = s, A_0 = 1] \\ =& r(s,0) + \beta E[J_n(S_1,0) + \max\{-J_n(S_1,0) + J_n(S_1,1), 0\}|S_0 = s, A_0 = 0] \\ & - r(s,1) - \beta E[J_n(S_1,0) + \max\{-J_n(S_1,0) + J_n(S_1,1), 0\}|S_0 = s, A_0 = 1] \\ =& r(s,0) + \beta E[J_n(S_1,0)|S_0 = s, A_0 = 0] - r(s,1) - \beta E[J_n(S_1,0)\}|S_0 = s, A_0 = 1] \\ & + \beta E[\max\{-J_n(S_1,0) + J_n(S_1,1), 0\}|S_0 = s, A_0 = 0] \\ & + \beta E[\min\{J_n(S_1,0) - J_n(S_1,1), 0\}|S_0 = s, A_0 = 1] \\ =& r(s,0) + \beta E[J_n(S_1,1)|S_0 = s, A_0 = 0] - r(s,1) - \beta E[J_n(S_1,0)\}|S_0 = s, A_0 = 1] \\ & + \beta E[J_n(S_1,0) - J_n(S_1,1)|S_0 = s, A_0 = a] \\ & + \beta E[\max\{-J_n(S_1,0) + J_n(S_1,1), 0\}|S_0 = s, A_0 = 0] \\ & + \beta E[\min\{J_n(S_1,0) - J_n(S_1,1), 0\}|S_0 = s, A_0 = 1] \\ =& r(s,0) + \beta E[J_n(S_1,1)|S_0 = s, A_0 = 0] - r(s,1) - \beta E[J_n(S_1,0)\}|S_0 = s, A_0 = 1] \\ & + \beta E[\max\{J_n(S_1,0) - J_n(S_1,1), 0\}|S_0 = s, A_0 = 0] \\ & + \beta E[\min\{J_n(S_1,0) - J_n(S_1,1), 0\}|S_0 = s, A_0 = 1]\end{aligned}$$

$$
\begin{aligned}
=&\xi(s) \\
&+ \beta E[\max\{J_n(S_1,0) - J_n(S_1,1), 0\}|S_0 = s, A_0 = 0] \\
&+ \beta E[\min\{J_n(S_1,0) - J_n(S_1,1), 0\}|S_0 = s, A_0 = 1]
\end{aligned}
$$

By assumption, the three lines of last equality are all non-decreasing in s. Therefore $J_{n+1}(s,0) - J_{n+1}(s,1)$ is non-decreasing in s. Thus $J_{n+1}(s,a)$ is submodular.

iii. Hence by i. and ii. we can conclude that $J_n(s,a)$ is submodular for all $n \in N$.

Remark 5. *Let $n \to \infty$, we can say that this proof still holds. Therefore such conditions work for infinite-horizon MDP problem on discrete time scale.*

□