MANAGING DYNAMIC SPECTRUM ACCESS UNDER UNCERTAINTY IN COGNITIVE RADIO NETWORKS

DISTRIBUTION

Presented in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy in the Graduate School of the Ohio State University

By

Shuang Li, M.S.
Graduate Program in Computer Science and Engineering

The Ohio State University

2013

Dissertation Committee:

Ness B. Shroff, Advisor
Eylem Ekici
Dong Xuan
© Copyright by
Shuang Li
2013
ABSTRACT

Cognitive Radio Networks (CRNs) allow unlicensed users (secondary users, or SUs) to opportunistically access the licensed spectrum without causing disruptive interference to the primary users (PUs). One of the main challenges in CRNs is the ability to detect PU transmissions. Recent works have suggested the use of SU cooperation over individual sensing to improve sensing accuracy.

Our overall goal is to manage dynamic spectrum access using cooperative sensing and provide provable performance guarantees. We have included the following in the thesis:

We first consider a CRN consisting of multiple PUs and SUs to study the problem of maximizing the total expected system throughput. We propose a Bayesian-decision-rule-based-algorithm to solve the problem optimally with a constant time complexity. To prioritize PU transmissions, we re-formulate the throughput maximization problem by adding a constraint on the PU throughput. The constrained optimization problem is shown to be strongly NP-hard and solved via a greedy algorithm with time complexity $O\left(\frac{N^5}{\log^2 \frac{1}{1-\epsilon}}\right)$ where $N$ is the total number of SUs. The algorithm achieves a throughput strictly greater than $\frac{1}{2}(1 - \epsilon)$ of the optimal solution. We also investigate the case for which a constraint is put on the sensing time overhead, which limits the number of SUs that can participate in cooperative sensing. We reveal that the system throughput is monotonic over the number of SUs chosen for sensing.
Second, we study the throughput maximization problem for a multichannel CRN where each SU can only sense a limited number of channels. We show that this problem is strongly NP-hard, and propose an approximation algorithm with a factor of at least $\frac{1}{2}\mu$ where $\mu$ is a system parameter reflecting the sensing capability of SUs across channels and their sensing budgets. This performance is achieved by exploiting a nice structural property of the objective function and constructing a particular matching.

Third, we explore further into the operator based model and work on a social welfare maximization problem. To utilize spectrum resources in CRNs efficiently, an auction scheme is often applied where an operator serves as an auctioneer and accepts spectrum requests from SUs. Most existing works on spectrum auctions assume that the operator has perfect knowledge of PU activities. In practice, however, it is more likely that the operator only has statistical information of the PU traffic when it is trading a spectrum hole, and it is acquiring more accurate information in real time. We distinguish PU channels that are under the control of the operator, where accurate channel states are revealed in real-time, and channels that the operator acquires from PUs out of its control, where a sense-before-use paradigm has to be followed. Considering both spectrum uncertainty and sensing inaccuracy, we study the social welfare maximization problem for serving SUs with various levels of delay tolerance. We first model the problem as a finite horizon Markov decision process when the operator knows all spectrum requests in advance, and propose an optimal dynamic programming based algorithm. We then investigate the case when spectrum requests are submitted online, and propose a greedy algorithm that is $1/2$-competitive for homogeneous channels and is comparable to the offline algorithm for more general settings. We further extend the online algorithm to an online auction scheme, which
ensures incentive compatibility for the SUs and also provides a way for trading off social welfare and revenue.

Fourth, we develop a distributed scheduling algorithm for SUs with i.i.d. arrival processes at PUs. Developing a distributed implementation that can fully utilize the spectrum opportunities for SUs has so far remained elusive. Although throughput optimal algorithms based on the well-known Maximal Weight Scheduling (MWS) algorithm exist for cognitive radio networks, they require central processing of network-wide SU information. We introduce a new distributed algorithm that asymptotically achieves the capacity region of the cognitive radio systems. Unlike existing distributed queue-length based CSMA/CA algorithms, the proposed algorithms achieve the full SU capacity region while adapting to the channel availability dynamics caused by unknown PU activity.
I dedicate this thesis to my parents for their endless love and support along the way.
ACKNOWLEDGMENTS

First and foremost, I am sincerely grateful to my advisor, Prof. Ness B. Shroff, and co-advisor, Prof. Eylem Ekici, for their guidance and support throughout my Ph.D. They always inspire me to come up with new ideas and encourage me to work on problems that I thought was impossible to solve. Their passion for research deeply impresses me and will shed light on my future career. I appreciate them for letting me choose my own topic and explore areas I am interested in. There is no doubt that they set a good example for me and will impact me forever. I would also like to thank Zizhan for his patient and insightful discussion. Collaboration with him is fun and productive.

I also would like to thank Prof. Dong Xuan and Prof. Srinivasan for serving in my candidacy committee. Their comments and feedback made me improve the work and continue in the right direction for the dissertation.

I would like to express my gratitude to my parents and my family for their unconditional love and support. Their faith in me helped me overcome all difficulties during these years pursuing a doctoral degree overseas. Their love will keep supporting me in the future.

I would like to thank my boyfriend Kun for his love and encouragement when I am upset with research, career or life. I would like to thank all Prof. Shroff’s students (current and graduated) - Srikanth, Bo, Zhoujia, Wenzhuo, Shengbo, Swapna, Yousi, Yang, Irem, Yara, Ghada. It was my pleasure working with them and I will never
forget our fruitful discussions on coursework and research. I thank all other labmates - Ruogu, Bin, Dongyue, Fangzhou, etc. for all discussions and help in research and life.

I also would like to thank Jeri, our group secretary, for her hard work organizing and coordinating things and also her patience in answering my questions. I also thank all other professors and staff in the university who teach, help or encourage me in all my Ph.D. years.
VITA

1983 ................................. Born in Chongqing, China

2005 ................................. B.S. in Computer Science and Engineering, Wuhan University of Technology

2008 ................................. M.S. in Computer Science and Engineering, Auburn University

2008-present ........................ Ph.D. candidate/Graduate Research Associate in Computer Science and Engineering, The Ohio State University

PUBLICATIONS

Shuang Li, Eylem Ekici and Ness B. Shroff, “Power Control for AP-Based Wireless Networks under the SINR Interference Model: Complexity and Efficient Algorithm Development, Proceedings of the 20th International Conference on Computer Communications and Networks (ICCCN’11), August 2011.


Shuang Li, Zizhan Zheng, Eylem Ekici and Ness B. Shroff, “Maximizing Social Welfare in Operator-based Cognitive Radio Networks under Spectrum Uncertainty and

FIELDS OF STUDY

Major Field: Computer Science and Engineering

Specialization: Networking
# TABLE OF CONTENTS

Abstract ................................................................. ii
Dedication ................................................................. iv
Acknowledgments ......................................................... vi
Vita ................................................................... viii
List of Figures ................................................................. xiii

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1.1</td>
<td>1</td>
</tr>
<tr>
<td>1.2</td>
<td>5</td>
</tr>
<tr>
<td>1.3</td>
<td>6</td>
</tr>
<tr>
<td>1.4</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>2.1</td>
<td>12</td>
</tr>
<tr>
<td>2.2</td>
<td>15</td>
</tr>
<tr>
<td>2.3</td>
<td>19</td>
</tr>
<tr>
<td>2.3.1</td>
<td>21</td>
</tr>
<tr>
<td>2.3.2</td>
<td>24</td>
</tr>
<tr>
<td>2.3.3</td>
<td>25</td>
</tr>
<tr>
<td>3</td>
<td>28</td>
</tr>
<tr>
<td>3.1</td>
<td>29</td>
</tr>
<tr>
<td>3.2</td>
<td>30</td>
</tr>
<tr>
<td>3.3</td>
<td>34</td>
</tr>
<tr>
<td>3.3.1</td>
<td>35</td>
</tr>
<tr>
<td>3.3.2</td>
<td>37</td>
</tr>
<tr>
<td>3.4</td>
<td>37</td>
</tr>
<tr>
<td>3.4.1</td>
<td>38</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
</tr>
<tr>
<td>5.6.1</td>
<td>Performance of Greedy Online Algorithm</td>
</tr>
<tr>
<td>5.6.2</td>
<td>Tradeoff between Social Welfare and Revenue</td>
</tr>
<tr>
<td>5.7</td>
<td>Summary</td>
</tr>
<tr>
<td>6</td>
<td>Distributed Queue Length based Scheduling Algorithm</td>
</tr>
<tr>
<td>6.1</td>
<td>Introduction</td>
</tr>
<tr>
<td>6.2</td>
<td>System Model</td>
</tr>
<tr>
<td>6.3</td>
<td>The Distributed Scheduling Algorithm for CRNs</td>
</tr>
<tr>
<td>6.3.1</td>
<td>Q-CSMA Overview</td>
</tr>
<tr>
<td>6.3.2</td>
<td>Scheduling Algorithm for Single Channel CRNs</td>
</tr>
<tr>
<td>6.4</td>
<td>Simulations</td>
</tr>
<tr>
<td>6.5</td>
<td>Summary</td>
</tr>
<tr>
<td>7</td>
<td>Conclusion</td>
</tr>
<tr>
<td>7.1</td>
<td>Research Contributions</td>
</tr>
<tr>
<td>7.2</td>
<td>Future Directions</td>
</tr>
<tr>
<td>Bibliography</td>
<td></td>
</tr>
</tbody>
</table>
LIST OF FIGURES

<table>
<thead>
<tr>
<th>FIGURE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Cognitive cycle [75].</td>
</tr>
<tr>
<td>2.2</td>
<td>Network architecture for CRNs.</td>
</tr>
<tr>
<td>2.3</td>
<td>Sensing methods and decision rules for CRNs.</td>
</tr>
<tr>
<td>2.4</td>
<td>Classification of spectrum sharing in CRNs [4].</td>
</tr>
<tr>
<td>2.5</td>
<td>Scheduling policy taxonomy.</td>
</tr>
<tr>
<td>2.6</td>
<td>Dynamic spectrum sharing classification based on allocation behavior.</td>
</tr>
<tr>
<td>3.1</td>
<td>System model of an SU network coexisting with three PU networks. Small circles are SUs and rectangles are PUs. Each big circle represents the transmission range of the corresponding SU-BS/PU-BS. Each SU can sense all PUs in the system.</td>
</tr>
<tr>
<td>3.2</td>
<td>Control slot $T_c$ and data slot $T_d$. The circles are SUs.</td>
</tr>
<tr>
<td>3.3</td>
<td>Performance comparison of Bayesian decision rule, majority, AND and OR with $N = 20, T_c = 0.2$ and $\pi_0 = 0.4$.</td>
</tr>
<tr>
<td>3.4</td>
<td>Performance comparison of greedy algorithm and random selection when $T_c = 0.2, \pi_0 = 0.4$ and $r = 2$.</td>
</tr>
<tr>
<td>3.5</td>
<td>Performance comparison of SFS over different numbers of SUs in the sensing set when $N = 20, T_c = 0.2, \pi_0 = 0.4$ and $\gamma = 2$.</td>
</tr>
<tr>
<td>3.6</td>
<td>Performance comparisons with different rules/ranges of $P_m^i + P_j^f$ for Problem (C) with $\gamma = 0.5$. The restricted ranges show how diversity of SUs affect the performances of Greedy_sum.</td>
</tr>
<tr>
<td>3.7</td>
<td>Sensitivity analysis of Problem (A).</td>
</tr>
<tr>
<td>3.8</td>
<td>Sensitivity analysis of Problem (B) where $\alpha = 1.6$.</td>
</tr>
</tbody>
</table>
3.9 Sensitivity analysis of Problem (C) where $\gamma = 0.5$.

4.1 System model of an SU network overlayed with three licensed channels. SUs in each circle are capable of sensing the corresponding channel(s). SUs outside the sensing range, if selected for sensing, report random sensing results.

4.2 System throughput achieved by our algorithm, greedy algorithm and random algorithm.

5.1 System model of the CRN. In the left figure, small circles are SUs, squares represent PUs registered at the operator, and triangles are PUs out of the operator’s control. The big circle is the coverage area of the operator. The right figure shows the availability of channels in $T_1$ and $T_2$. For $T_1$ channels: 0 means idle state and 1 means busy state. For $T_2$ channels, the first element represents the actual state (0: idle; 1: busy) and the second element represents the sensed state (0: sensed idle; 1: sensed busy).

5.2 Performance of online algorithm versus offline algorithm over various request duration means with homogeneous and heterogeneous $T_2$ channels ($|T_2| = 3$), respectively.

5.3 Tradeoff between social welfare and revenue over reservation price with homogeneous and heterogeneous $T_2$ channels ($|T_2| = 3$), respectively. $|T_1| = 0$ in (a) and $|T_1| = 1$ in (b).

6.1 A SU network composed of 7 SUs overlayed with a PU network.

6.2 DTMC with the vector of transmission schedule $\vec{x}(t)$ of two links as the state. Two links interfere with each other.

6.3 DTMC with the vector of transmission schedule $\vec{x}(t)$ of three SU links as the state in CRNs. 1 and 2 interfere with each other. 2 and 3 interfere with each other. 1 and 3 can transmit simultaneously.

6.4 DTMC with both the vector of transmission schedule $\vec{x}(t)$ of two SUs and channel state as the state. Two SUs are in the conflict set of each other.
6.5 Two DTMCs evolutions for SUs inside and outside the interference range of the PU. Two SUs are in the conflict set of each other. Both DTMCs are time reversible and have product-form stationary distribution (by Propositions 6.3.2 and 6.3.4). $\alpha_2^a$ is the probability SU 2 is chosen the decision schedule $m^a$ when the channel is available. $\alpha_2^b$ is the probability SU 2 is chosen in the decision schedule $m^b$ when the channel is unavailable. Note that $|m^a| = 2$ and $|m^b| = 1$ in this example.

6.6 Conflict graph with 6 SUs.

6.7 Queue lengths of two algorithms with different loads in the 6-SU network.

6.8 Conflict graph with 16 SUs.

6.9 Channel usage percentage over load factor in the 4 by 4 grid network.

6.10 Channel usage percentage over PU traffic in the 4 by 4 grid network.

6.11 Time for each SU to saturate in the 6-SU network.

6.12 Network throughput over the probability of false notification of channel state in the 6-SU network.
CHAPTER 1
INTRODUCTION

Cognitive radio networks (CRNs) have been proposed to address the spectrum scarcity problem by allowing unlicensed users (secondary users, SUs) to access licensed spectrum on the condition of not disrupting the communication of licensed users (primary users, PUs). To this end, SUs sense licensed channels to detect primary user (PU) activities and find the underutilized “white spaces”. The Federal Communications Commission (FCC) has opened the TV bands for unlicensed access [16] - a large portion of the assigned spectrum used to be underutilized is now allowed to be used by SUs temporarily [4]. IEEE has formed a working group (IEEE 802.22 [32]) to regulate the unlicensed access without interference. Many other organizations are also making efforts on the spectrum access policy in the CRN environment, e.g., DARPA’s ‘Next Generation’ (XG) program [63] mandates that cognitive radios to sense signals and prevent interference to existing military and civilian radio systems.

1.1 Background

The technology of cognitive radios has spawned the next generation networks, known as dynamic spectrum access (DSA) networks. The cognitive capability allows SUs

1The recent FCC ruling requires the use of central TV Band usage databases to verify spectral availability. While respecting this ruling, our work explores local cooperative methods to improve sensing accuracy with the potential outcome of relieving this burdensome requirement.
to monitor and sense the surrounding radio environment. Their reconfigurability enables SUs to select the best available channels for transmission and switch to other channels when PUs reclaim the right to channel usage by reconfiguring the operating parameters. These two capabilities are designed to make SUs opportunistically access the spectrum by exploiting the licensed spectrum used by PUs.

A typical duty cycle of CRNs [75] includes detecting white space in the spectrum, selecting the best available channels, coordinating spectrum access with other SUs, and vacating the spectrum band when a PU reclaims its usage right. Then the main functions of a CRN can be summarized as: spectrum sensing and analysis, spectrum management and handoff, and spectrum allocation and sharing. Details of the network architecture, applications and these three functionalities will be covered in Chapter 2. Our main contributions fall into spectrum sensing and analysis, and spectrum allocation and sharing. In this chapter, we give a brief introduction on how our contributions make an impact on these functionalities.

**Spectrum sensing and analysis:** Spectrum Sensing can be performed via several methods, including energy detection, cyclostationary feature detection, and compressed sensing [34]. Individual sensing suffers from shadowing or heavy fading, which leads to incorrect decisions. To address these problems, cooperative sensing that jointly processes the sensing results of multiple SUs have been proposed [25][48][54]. Our contributions in Chapters 3 and 4 fall into this functionality.

Cooperative sensing overcomes the shortcomings of individual sensing results by fusing the observations of individual SUs. Three main categories of decision rules have been identified in [34]: *Soft Combining*, *Quantized Soft Combining*, and *Hard Combining*. Similar to sensor networks, linear fusion rules are widely applied to achieve a cooperative decision, such as AND, OR and majority rules [54]. In addition, there is a more advanced fusion technique that utilizes statistical knowledge [70] to
capture the correlation between SUs in cooperative sensing. However, none of the above-mentioned works identify optimal decision rules for general decision structures.

In multi-channel CRNs, cooperative sensing has new challenges. While improving sensing accuracy, it also incurs sensing and reporting overhead at the SU side, especially when a SU senses multiple channels. In particular, requiring each SU to sense all the channels in a CRN may lead to long sensing durations when the number of channels is large, which in turn reduces the average throughput of SUs. To provide efficient channel access to SUs, it is therefore reasonable to put a limit on the maximum sensing duration that a SU can afford, which translates to a budget on the number of channels that a SU can sense. Due to the hardware constraints, this budget could be different for different SUs.

Various cooperative sensing protocols have been proposed for maximizing certain system-wide performance metrics such as sensing accuracy [55] and system throughput [44, 82]. In particular, we study system throughput maximization problem in different settings [44]. An optimal Bayesian decision rule that maps a vector of local binary decisions made at SUs to a global decision on PU activity has been found for maximizing system throughput in a single channel setting [44], which achieves significantly better performance than linear rules such as AND, OR, and majority rules. However, these works either focus on a single-channel setting [55, 44] or allow each SU to sense all the channels [40, 22, 82]. For instance, an direct extension of our result in [44] to the multi-channel setting would require each SU to sense all the channels and incur high sensing duration. On the other hand, most works on multi-channel cooperative sensing put no explicit constraint on sensing duration of SUs. Furthermore, these works either use a simple linear decision rule [82] or require the transmission of the entire local sensing samples or sensing statistics at each SU. In [44], we choose to use a binary decision rule to avoid the high overhead involved
in reporting complete local sensing results. But instead of using a suboptimal linear rule as in [82], we use the optimal decision rule proposed in [44] for each channels.

**Spectrum allocation and sharing:** With the sensing decision, SUs have the knowledge of channel availabilities. However, the radio environment is time varying and channels sensed idle are of different qualities. SUs should decide on when and how to allocate the spectrum bands in order to meet the QoS requirement. Our contributions in Chapters 5 and 6 are within this functionality. Specifically, they fall into two subareas spectrum sharing game and medium access control, respectively.

1) Medium access control (MAC): In general wireless networks, lots of MAC protocols have been proposed such as Maximum Weight Scheduling (MWS), carrier sensing multiple access (CSMA), and slotted ALOHA. None of these have incorporated the cognitive capabilities in CRNs. As shown in the thesis, a direct application of these works does not guarantee good throughput. Assuming no sensing inaccuracy, there are only centralized algorithms for the optimal utilization of the so-called spectrum opportunities while avoiding interference on PUs. Maximum Weight Scheduling (MWS) algorithm [69] and its variants achieve the full capacity region of the network. However, these algorithms require the knowledge of the entire network state and centralized processing to compute conflict free schedules. Similar algorithms have also been proposed for cognitive radio networks [71], [80]. Both works require solving an NP-hard problem centrally. A distributed algorithm that achieves the full capacity region of the network is unknown.

2) Spectrum sharing game: In CRNs, users may request for spectrum usage in both cooperative and non-cooperative ways. In the non-cooperative case, selfish users compete with each other and tend to cheat on their private information such as valuation of the spectrum in order to maximize their own utilities. An auction framework is often applied where an operator serves as an auctioneer and accepts requests from
SUs. Most existing works assume that the operator has perfect knowledge of PU activities in a given period of time. In practice, the channel states are uncertain due to the frequent PU usage. So it is more likely that the operator has the statistical information of the PU activities, and acquire more accurate information over time. In our work we consider two types of PU channels: those under the control of the operator and those out of its control. For the former, accurate channel states are revealed in real-time while a sense-before-use paradigm has to be followed for the latter. The operator must first identify spectrum holes in a channel, e.g., by coordinating SUs to sense the channel, before allocating the holes to SUs. While spectrum sensing has been extensively studied in the CRN literature [44, 45, 35], the joint problem of sensing and spectrum auction remains unexplored.

1.2 Scope of the Research

In this thesis, we focus on several subareas of the two functionalities in a typical duty cycle of CRNs: spectrum sensing and analysis; spectrum allocation and sharing. Our first focus is on designing efficient cooperative sensing algorithms to maximize system throughput (Chapters 3 and 4). System throughput is a unifying objective that quantifies the effects of misdetection and false alarm probabilities on the system performance. We also take into account a variety of constraints, such as requiring the PU throughput to be above a certain threshold as well as limiting the amount of sensing overhead per SU.

Our next focus is within the functionality of spectrum allocation and sharing (spectrum sharing game and medium access control). While we apply commonly adopted framework of spectrum auction, we add the challenges of spectrum uncertainty and sensing inaccuracy into our model (Chapter 5). We distinguish PU channels that are under the control of the operator, where accurate channel information
is revealed in real-time, and channels that are out of control of the operator, where a sense-before-use paradigm has to be followed. We study the social welfare maximization problem for serving SUs with various levels of delay tolerance. We then design distributed scheduling algorithm that achieves the full capacity region of SUs subject to the PU activity (Chapter 6). Compared to existing queue-length based CSMA/CA algorithms, the channel state change over time leads to a different design.

1.3 Contributions

Generally speaking, we make the following key research contributions in this thesis:

- We design an optimal data fusion rule to (hard) combine the reported sensing results (Chapter 3). More specifically, we aim to maximize the system throughput in a CRN composed of multiple PUs operating on orthogonal channels and SUs, where SUs are allowed to sense all the channels in the network. We assume that sensing decisions are made on a per channel basis and each SU can sense all the PUs in the system. Thus, we only need to solve the system throughput maximization problem per channel.

  - In contrast to previous works that restrict the class of fusion rules, we propose a Bayesian decision rule based algorithm to solve the throughput maximization problem optimally with constant time complexity.

  - To guarantee resources for the PU, we re-formulate the problem by adding a constraint on the PU throughput. This constrained problem is shown to be strongly NP-hard by reducing the classical product partition problem [9] to it. A greedy algorithm is obtained with pseudo-polynomial time complexity. This approximation algorithm is analytically shown to achieve
strictly greater than $1/2 - \epsilon$ of the optimal solution, where $\epsilon$ is used to trade off computation time complexity.

- We investigate systems where limited sensing overhead is allowed, i.e., the number of sensing SUs is restricted. Our theoretical results show that the performance of cooperative sensing is monotonic over the number of SUs used for sensing.

- We now consider a more general setting in which practical considerations such as multi-channel capability and sensing overhead are taken into account. Specifically, we study the system throughput maximization problem in a multi-channel CRN, by deciding for each channel, a subset of SUs to sense the channel, subject to the sensing budget constraint at each SU [45] (Chapter 4).

- We show that the throughput maximization problem is NP-hard in the strong sense and hence does not have a pseudo-polynomial time algorithm unless $P = NP$.

- We prove that the system throughput function satisfies a structural property, and based on this we propose a matching-based algorithm, which achieves an approximation factor at least $\frac{1}{2} \mu$ where $\mu \in [1, 2]$ is a system parameter depending on the sensing capability of SUs across channels and their sensing budgets.

- We study the joint spectrum sensing and allocation problem to serve spectrum requests with arbitrary valuations and arbitrary levels of delay tolerance (Chapter 5). The objective of the operator is to maximize social welfare. We consider both the scenario where the operator knows all spectrum requests in advance, and the setting when spectrum requests are submitted online.
We model the joint sensing and spectrum allocation problem as a finite horizon Markov decision process and develop an optimal dynamic programming based algorithm, which serves as a baseline for the achievable social welfare.

We propose a greedy algorithm for the case when spectrum requests are submitted online. We prove that the online algorithm is 1/2-competitive for homogeneous channels, and we show that it achieves performance comparable to the offline algorithm for the more general heterogeneous channel case by numerical results.

We further extend the online algorithm by proposing an online auction scheme, which ensures incentive compatibility for SUs and also provides a way for trading off social welfare and revenue using a reservation price.

- We develop a new distributed throughput optimal scheduling algorithm for CRNs (Chapter 6). To this end, we introduce a new system state representation that includes channel state information, and design our algorithms to achieve throughput optimality without causing interference with PUs.

1.4 Outline

The main body of the thesis is organized as follows. In Chapter 2, we provide an extensive literature review of CRNs. We first identify the intrinsic challenges caused by the properties of cognitive capability and reconfigurability. We then summarize the functions in a typical duty cycle of CRNs. We then survey the existing work on the network architecture, applications and the functions that contribute to the spectrum-aware communication protocol design.

In Chapter 3, we present our first part of the work on achieving maximum system
throughput. We design optimal Bayesian decision rule based algorithm and also consider the same problem with constrains such as PU throughput above a threshold. We propose a greedy algorithm that achieves at least $\frac{1}{2} - \epsilon$ of the optimal solution where $\epsilon$ affects the time complexity of the algorithm. We also investigate systems with limited sensing overhead, i.e., the number of sensing SUs is restricted.

In Chapter 4, we put a limit on the maximum sensing duration that an SU can afford, which translates to a budget on the number of channels an SU can sense. We study the throughput optimization problem for a multi-channel CRN subject to the sensing constraint, and propose an approximation algorithm with a provable factor.

In Chapter 5, we study the social welfare maximization problem in operator-based CRNs and distinguish our work from others by considering both spectrum uncertainty and sensing inaccuracy. We propose both offline and online solutions and show their theoretical performance. We further extend the online algorithm to an online auction scheme, which ensures incentive compatibility for the SUs and also provides a way for trading off social welfare and revenue.

In Chapter 6, we aim to find a distributed solution to the complete utilization of the spectrum opportunities while avoiding interference on PUs. We propose a queue length based CSMA algorithm for CRNs that captures the dynamics of the channel state.

In Chapter 7, we conclude the work we have done. We summarize our research on efficient cooperative sensing algorithms, spectrum allocation to SU requests with various levels of delay, and distributed scheduling algorithms in CRNs. We also outline the future research directions that extend our work.
In recent years, the rapidly increasing use of wireless devices has generated a huge demand for wireless spectrum. The limited radio spectrum means that the spectrum needs to be efficiently utilized, but this is difficult to achieve under the current fixed spectrum allocation policies. Cognitive Radio Networks (CRNs), an innovative technology, has been proposed to alleviate the spectrum shortage problem by letting secondary users (SUs), who have no spectrum licenses, use the temporarily unused licensed spectrum. To this end, the Federal Communications Commission (FCC) [16] has opened the broadcast TV frequency bands for unlicensed users such as WLAN and WiFi. The technology of cognitive radio enables the next generation communication networks, also known as dynamic spectrum access (DSA) networks. The cognitive capability allows radios to monitor, sense and detect the surrounding environment, dynamically reconfigure the operational parameters, and identify the best available spectrum [4][31]. SUs, by exploiting the spectrum used by primary users (PUs), opportunistically access the spectrum. They coordinate the spectrum access with each other, select the available channels, and also vacate the channel when a PU reclaims the usage right [75].

The more flexible and intelligent use of spectrum in CRNs allows the coexistence of PUs and SUs. Conventional communication protocols, which were developed for “dumb” radios which do not dynamically change parameters or channels, are no
longer useful [4]. Allowing SUs to use the spectrum in an unregulated manner will have an adverse impact on the PU performance. New spectrum usage guidelines that capture the features of CRNs need to be developed for new challenges, specifically in spectrum sensing and dynamic spectrum allocation.

A typical duty cycle of CRNs [75] includes detecting white space in spectrum, selecting the best available channel, coordinating spectrum access with other users and vacating the frequency when a PU starts transmission (Figure 2.1). The main functions then can be summarized as:

- *Spectrum sensing and analysis*: Detecting spectrum “holes” and sharing it without disrupting the PU transmissions;

- *Spectrum management and handoff*: Selecting the best available channel to meet user communication requirements;
Spectrum allocation and sharing: Addressing how and when to use a spectrum band adapted to its dynamics.

These key CRN functionalities enable communication protocols with cognitive capability and reconfigurability. Existing works mainly fall into these three categories and many of them reflect the significant contributions in the past decade to CRNs. In this chapter, we will survey recent advances in CRN architecture, its applications and research works on the three functionalities listed. We will also discuss several important issues that have not been addressed.

2.1 Network Architecture and Applications

In a CRN architecture (Figure 2.2), secondary network may access licensed or unlicensed band. In the following, we focus on the former which involves infrastructure since it is more complicated and commonly adopted.

A CRN is composed of a primary network and a secondary network. A primary network is the existing network infrastructure which has exclusive right to the licensed spectrum band. It is usually composed of a set of primary users and at least one primary base stations. Primary users have a license to use certain spectrum bands. Their activities are controlled by the primary based station so that they do not affect the activities of each other. Also, their transmissions should not be interfered by secondary users. In general, the primary network is not equipped with CR functions.

A secondary network is not authorized to use licensed spectrum bands without regulating their transmissions. It is composed of a set of secondary users with/without a secondary base station. The secondary base station coordinates the SU transmissions opportunistically and enables the SUs to access other networks. Both secondary base stations and secondary users require additional functionalities. If several secondary networks compete for the same spectrum band, the spectrum usage may be
coordinated by a spectrum broker, which is a centralized entity that connects each network. Spectrum broker collects operation information and requests from each secondary network, and allocate the spectrum resources for efficient spectrum sharing. In Chapter 5, we investigate problems on a scale that involve operators and potential intermediaries that regulate SU transmissions, unlike what was being done before.

Cognitive radios can sense and detect the temporally unused spectrum holes in the licensed spectrum band. They reconfigure the operating parameters to access the best available channel and autonomously move to the new available spectrum if the primary user reappears. These properties of CRNs increase the spectrum usage and
channel utilization so that better quality of service can be provided. CRNs can be applied in the following cases:

1) Market potential for wireless technologies: Demand for mobile data services such as video and multimedia streaming, has been increasing exponentially. U.S. commercial networks are reported to operate at 80% of total capacity, with 26% at full capacity [2]. However, reallocating the needed spectrum in traditional ways is costly and time consuming. Cognitive radio then becomes a promising technology for wireless service providers by increasing the capacity of the networks and reduce interference. In CRNs, the secondary network can intelligently detect spectrum in use and switch communication to unused spectrum to satisfy the high bandwidth demand of the data services. Also, cognitive radio is able to autonomously select best available channels by reconfiguring its operating parameters. Hence, a large range of frequencies can be utilized including those that suffer less from heavy fading and shadowing. Thus CRN has potential market values for ubiquitous mobile wireless services in the near future.

2) Public safety and emergency network: Natural disasters can destroy the existing commercial wireless systems. Cognitive radio networks can provide resilience and continuity of operations in such wireless-impaired environments. For example, once a disastrous event happens, an emergency network is triggered and formed by recognizing the channel availabilities in the surrounding environment. Communications with public safety personnel including voice, data, and video are immediately conducted through this network with the lowest delay. Adapting to the dynamic conditions of other networks, quality of service can also be guaranteed. On the other hand, CR technology may help improve the traffic conditions, especially in metropolitan
districts. Secondary network may collect information such as car accidents, congestion, and detour, so that vehicles that have not entered the alarmed area can take corresponding actions.

3) Military applications: the military communication network intends to provide multiple types of services in hostile environment. The capacity is limited, however, due to the static frequency assignments. Idle spectrum bands are not reallocated. CR technology allows the military personnel to select the most secure channel and perform spectrum handoff. It not only alleviates spectrum congestion but also guarantees seamless and secure communications.

2.2 Spectrum Sensing and Analysis

As mentioned before, in CRNs, secondary users (SUs) are offered the opportunity of accessing the licensed channel when their activities do not cause disruptions for primary user (PU) transmissions. The Federal Communications Commission (FCC) [16] has allowed the unlicensed user access to the broadcast TV frequency bands. Most recently, congressional negotiators have reached the compromise to allow the auction of TV broadcast spectrum to wireless Internet providers [1]. IEEE has announced the IEEE 802.22 wireless network standard [32] that specifies how to utilize the unused resources between channels in the TV frequency spectrum. Since the PU activity is not known by the SUs in real time, SUs have to sense the spectrum and make sure their transmissions do not collide with the primary traffic.

Sensing can be performed via several methods, including energy detection, matched filter detection [4], feature detection, and compressed sensing [34]. We briefly overview these various approaches below.

1) Energy detection: It is a simple method easy to implement and requires no a priori knowledge of PU signals [72]. A good detector should ensure a low false
alarm probability and a high detection probability. In [61], the optimal detector for detecting a weak unknown signal from a known zero-mean constellation is shown to be the energy detector. It is also shown that a known pilot signal can help greatly. Various approaches have been proposed so far to improve the efficiency of energy detection for spectrum sensing. Noise level estimation is an important step since the detection performance is very sensitive to the estimation error [67]. An adaptive noise level estimation algorithm is proposed and its robust performance is illustrated by results performed on real RF spectrum measurements in [53]. The energy-based signal detection is based on an adaptive threshold estimation stage so that the sensor noise level is matched under various conditions. In [52], the threshold level with energy detection is optimized subject to the spectrum sensing error constraints. The sensing throughput tradeoff problem is studied in [54], where an iterative algorithm is proposed to obtain the optimal parameters. Aside from uncertainty in noise power, energy detector cannot differentiate signal types. Thus it is vulnerable to the false detection triggered by unintended signals [4].

2) Matched filter detection: If the primary signal is known a priori by the secondary users, the optimal detector is matched filter since it maximizes the received

Figure 2.3: Sensing methods and decision rules for CRNs.
signal-to-noise ratio (SNR) in the presence of additive stochastic noise [61]. The advantage of the matched filter is that it requires less time to achieve a certain detection performance due to the coherency. However, its performance degrades significantly if the information is not accurate. Also the required number of signal samples grows as the required performance level increases.

3) Feature detection: The transmission of a primary user has specific features associated. Modulate signals are generally coupled with sine wave carriers, pulse trains, repeating spreading, etc. These features are viewed as the cyclostationary features [10][23] since their mean and autocorrelation are periodic. A cyclostationary feature detector can perform better than energy detector since it can differentiate noise from primary signals and distinguish between different primary networks[68]. More generally, other features can be extracted from a primary user’s transmission, such as the transmission technologies, the amount of energy, channel bandwidth, etc [75]. Primary users can be similarly identified after matching the extracted features from the PU transmission to the a priori information.

4) Compressed sensing: The previously introduced sensing methods are all based on the observation samples in a specific band. Sensing one band at a time due to the hardware constraint causes large overhead. An alternative is to use multiple RF frontends for wideband sensing, which however leads to long sensing delay or high computational time complexity. Compressed sensing [20][11] enables the sampling of the wideband signals at sub-Nyquist rate and facilitates the detection of sparse signals by approximating and recovering the sensed spectrum with reasonable computational complexity.

The main disadvantage of individual sensor nodes making sensing decisions without cooperation with other nodes is that it could result in decreased accuracy in the presence of fading, shadowing, and unknown noise power profiles. For instance, if an
SU suffers from shadowing or heavy fading, the sensed signal tends to be weak while
the PU is transmitting, leading to incorrect decisions. Consequently, the sensing
results from other users may help improve the accuracy of the primary signal de-
tection. To address these problems while maintaining sensing simplicity, cooperative
sensing schemes that fuse the sensing results of multiple SUs have been proposed in
the literature [25][48][54]. Cooperative sensing can be conducted in a centralized or a
distributed manner. In centralized model, a secondary base station plays a role of the
fusion center and collects sensing results from individual SUs, decides on the channel
availabilities based on a predefined decision rule, and informs SUs of channel alloca-
tion. In distributed cooperative sensing, SUs exchange their sensing results with each
other without a backbone infrastructure. The decision function aims at optimizing
an objective function. Examples of such functions include maximizing sensing accu-
rcacy (generally, a function of false alarm probability and mis-detection probability)
or maximizing the system throughput. Aside from sensing accuracy related metrics,
cooperative sensing schemes are also designed to estimate the maximum transmit
power for SUs so that they do not cause disruptive interference to PUs [47]. On
the other hand, cooperative sensing incurs additional sensing delay over individual
sensing.

1) Decision rules: Three main categories of decision rules have been identified in
[34]: Soft Combining, Quantized Soft Combining, and Hard Combining. They have
different control channel bandwidth requirements. In soft combining, raw sensing
results, i.e., data sequences, are sent to the fusion center. In quantized soft combining,
quantized sensing results are sent to the fusion center for soft combining to reduce
the control channel communication overhead. While in hard combining, binary local
decisions on the sensing results are reported. Compared to the other two categories,
only a single bit is sent to the fusion center in hard combining.
Among hard combining rules, linear fusion rules [34] are widely applied to achieve a cooperative decision, such as AND, OR and majority rules [54]. AND and OR both take extreme approaches: In AND, only when all stations decide the channel to be “busy”, the decision after fusion is “busy”, which promotes the SU activity; In OR, only when all stations decide the channel to be “idle”, the decision after fusion is “idle”, which tends to protect the PU activity. The majority rule uses the majority of the local decisions as the final decision, which places it between AND and OR in terms of SU transmission eagerness. In addition, a linear-quadratic fusion rule that utilizes statistical knowledge [70] has been devised to capture the correlation between SUs in cooperative sensing. None of the aforementioned rules are shown to be optimal. In [70], the fusion rule is shown to not be optimal, and its performance compared to optimal has not been analytically characterized.

2) Information sharing: Cooperative sensing requires information exchange among SUs, including their locations, estimation of PU’s location and power, which SUs should report the sensing results, etc. If not properly designed, the information exchange algorithm may cause large overhead. In [3], a novel incremental gossiping approach is proposed to coordinate spectrum sensing. It limits the amount of information exchanged between nodes and adapts quickly to network alterations. It is shown to allow exponentially-fast information convergence. A censoring scheme is proposed in [66] where only users with reliable information are allowed to report their binary sensing decisions to the common receiver.

2.3 Dynamic Spectrum Allocation and Sharing

In the previous section, we discussed the spectrum sensing solutions for efficient detection of unused spectrum bands. With the decision, SUs will be aware of the channel availabilities in current time slot. However, the radio environment is time
varying and the spectrum band information such as the operating frequency and bandwidth also varies across PU networks. In order to meet the QoS requirement, SUs should decide on when and how to use the spectrum bands by taking into account the channel dynamics. Coexistence with PUs is also a great challenge in that interference should not be caused to disrupt their data transmissions. According to their architecture assumption, spectrum allocation behavior, and spectrum access technique, most existing solutions of spectrum sharing can be categorized in three classifications [4] (Figure 2.4). We will briefly describe these three classifications and then focus on three subareas: medium access control, spectrum handoff and spectrum sharing game.

The first classification is based on architecture assumption:

- **Centralized spectrum sharing**: A central entity controls the spectrum allocation and access [8][59][60].

- **Distributed spectrum sharing**: There is no central controller due to the high cost of infrastructure. Spectrum is shared in an ad hoc manner. Each user makes its own decision on the spectrum access strategy [12][33][46].

The second classification is based on spectrum allocation behavior:

- **Cooperative spectrum sharing**: All SUs work towards a common goal. Most
probably, they belong to the same operator. They communicate with each other in order to maximize their social welfare. Most centralized solutions are cooperative [8][12][33].

- Non-cooperative spectrum sharing: SUs do not belong to the same operator so they act in a selfish or non-cooperative manner. Each SU network pursue their own benefit [62][83][84].

The third classification is based on spectrum access technology:

- Spectrum underlay: SUs are allowed to transmit their data in the licensed spectrum when PUs are also transmitting. It exploits the spread spectrum techniques in cellular networks [33]. Its transmit power at a certain portion of the spectrum is regarded as noise by the PUs. Underlay spectrum sharing can utilize the bandwidth more efficiently. However, it requires sophisticated techniques and there are stringent constraints on transmit power.

- Spectrum overlay: Unlike spectrum underlay, SUs opportunistically access the licensed spectrum. No interference is allowed to the primary system. SUs need to sense the spectrum and detect the PU activities [8][12][46].

We next give more insights on the design of spectrum allocation and sharing schemes by focusing on several important issues related to the thesis work.

### 2.3.1 Medium Access Control in CRNs

Medium access control (MAC) regulates how SUs access a licensed spectrum in CRNs. Avoiding interference on PUs is the goal of algorithm and protocol design in the dynamic spectrum access. In general wireless networks, various MAC protocols have
been proposed such as Maximum Weight Scheduling (MWS), carrier sense multiple access (CSMA) and slotted ALOHA. In traditional wireless networks, Maximum Weight Scheduling (MWS) algorithm [69] and its variants achieve the full capacity region of the network, where a scheduling policy is said to achieve the full capacity region (or be throughput optimal) [37] if it stabilizes the system for any arrival rate vector the system can be stabilized for by some scheduling policy. However, these algorithms require the knowledge of the entire network state and centralized processing to compute conflict free schedules. Similar algorithms have also been proposed for cognitive radio networks: In [71], opportunistic scheduling policies are developed for multichannel single-hop CRNs subject to maximum collision rate constraints with PUs. In [80], scheduling algorithms are investigated in multi-channel multi-hop CRN overlayed with a PU network. The optimal throughput can be provably and asymptotically achieved in adaptive-routing scenarios. Both works require solving an NP-hard problem centrally.

These centralized throughput optimal algorithms suffer from two main shortcomings. The first is high computational complexity, and the second is the cost associated with the collection of network state information at a central location. The first problem has been countered in the literature through lower complexity suboptimal algorithms. Maximal Scheduling is such an algorithm that reduces the time complexity of MWS at the expense of achieving a fraction of the capacity region [14]. Another algorithm, Greedy Maximal Scheduling (GMS), always selects a link with the longest queue that does not cause interference to links already chosen. GMS is shown to be throughput optimal if the network topology satisfies certain topological properties known as the local pooling conditions [18]. Nevertheless, only a fraction of the capacity region can be achieved for general topologies [37]. Furthermore, the
Our contribution

distributed

MWS ...
Throughput optimal

ALOHA, CSMA, GMS, Maximal scheduling ...
All scheduling policies for CRNs

Figure 2.5: Scheduling policy taxonomy.

distributed implementation of GMS entails signaling overhead, which is not scalable [42].

Recently, a new class of distributed algorithms has been proposed to achieve throughput optimality while circumventing these problems. These new algorithms are random access algorithms based on the notion of channel sensing. These algorithms use queue lengths to determine channel access probabilities, achieving the full capacity region in ad hoc wireless networks in a distributed manner. In [36], an adaptive throughput optimal CSMA scheduling algorithm is proposed for a general interference model in continuous time. It uses transmission aggressiveness, which is a function of the queue length. Implementation considerations in 802.11 networks are discussed considering packet collisions. In [49], a discrete-time distributed randomized algorithm based on Glauber dynamics is proposed. In both [36] and [49], the queue
length based CSMA algorithms achieve the full capacity region in a single-channel ad
hoc wireless network.

The queue length-based CSMA algorithms of [36] [49] assume that the channel is
always available, a condition not satisfied in cognitive radio networks. In this paper,
we develop new distributed throughput optimal scheduling algorithms for CRNs. To
this end, we introduce a new system state representation that includes channel state
information, and design our algorithms to achieve throughput optimality without
cauising interference with PUs (Figure 2.5).

2.3.2 Spectrum Handoff

When the current channel is reclaimed by PUs or the channel conditions become
worse, spectrum handoff occurs. SUs need to stop data transmission and find other
available channels to resume the transmission. The protocols must adapt to the
new channel parameters of the operating frequency quickly in order to reduce delay.
Also, they must be transparent to end users. To make sure such transitions are
made smoothly and quickly, spectrum mobility management comes into play. In [86],
a channel reservation scheme is proposed which allows the tradeoff between forced
termination and blocking based on QoS requirement. It is shown that significantly
higher throughput, compared to the case where no reservation scheme is applied,
could be achieved with a proper number of channels reserved. Note that reserving
too much bandwidth may lead to low throughput since the primary user may not
reclaim its usage very often. A location-assisted spectrum handoff scheme is proposed
in [13]. SUs are equipped with a location estimation and/or sensing device for their
own location estimation. They can send their spectrum requests together with their
location information to the secondary base station. When spectrum handoff happens,
SUs can switch to one of the candidate channels depending on their locations.
2.3.3 Spectrum Sharing Game

The tool of game theory has been exploited in dynamic spectrum sharing. In [50], a game theoretic framework is proposed for the analysis of cognitive radios for distributed adaptive channel allocation. Both cooperative and non-cooperative users are captured in the utility definition. It is shown that the channel allocation game converges to a deterministic channel allocation Nash equilibrium point. Also, cooperation based spectrum sharing is shown to improve the overall network performance with more overhead of information exchange though. A repeated spectrum sharing game with cheat-proof strategies is proposed in [78]. Users are incentivized to share the spectrum in a cooperative way. Two mechanism design methods are proposed to suppress cheating and collusion in selfish users by introducing a transfer function in the utility function in [76].

An auction framework is also applied where an operator serves as an auctioneer and accepts requests from SUs. These frameworks are implemented via a resource allocation and a payment scheme with the objective of maximizing either social welfare or revenue [85, 19, 28, 26, 15].

Most existing works on spectrum auctions, however, assume that the operator has perfect knowledge of PU activities in a given period of time. They ignore the uncertainty of channel states caused by the uncertain and frequent PU usage. For instance, in cellular networks, the subscribers access channels at their own will and they do not make any reservation for the spectrum usage. Hence, these existing auction schemes are mainly applicable to spectrum resources that tend to be available for relatively long periods of time. For instance, the interval between two adjacent auctions is assumed to be 30 minutes or longer in [19]. However, to allow more efficient spectrum utilization and relieve spectrum congestion, spectrum holes at smaller time scales need to be explored. A straightforward extension of current approaches to this
more dynamic environment would require auctions to be conducted frequently, which would incur high communication and management overhead. A more reasonable approach is to again consider a relatively long period of time, where the operator only has statistical information of the PU traffic when trading spectrum holes. More accurate information is acquired later in real-time. Therefore, an auction scheme that takes spectrum uncertainty into account is needed. In such a dynamic environment, due to this limitation, spectrum holes at small scale cannot be efficiently utilized. The period of auction only lasts for a time slot or other units of a short period of time so that accurate channel state information can be collected. Since the collection has to be done frequently, high overhead will be incurred. Current works on dynamic
spectrum access allocates spectrum resource with complete channel state information in a long period of time. In their model, small spectrum holes are wasted since their availabilities are not known ahead of time. Only large holes in the white space are allocated for SUs. To use these small holes, allocations have to be done frequently, which results in high overhead.

To further improve spectrum utilization, besides trading spectrum holes that are fully under the control of the operator, as commonly assumed in the spectrum auction literature, the operator may choose to acquire licensed channels out of its control to further improve social welfare or revenue. To avoid interference with PUs, a *sense-before-use* paradigm must be followed in this case. Although spectrum sensing has been extensively studied in the CRN literature [44, 45, 35], the joint problem of sensing and spectrum auction remains unexplored (Figure 2.6).
CHAPTER 3
MAXIMIZING SYSTEM THROUGHPUT BY
COOPERATIVE SENSING

We focus on a CRN consisting of multiple PUs and SUs to study the problem of maximizing the total expected system throughput. We first propose a Bayesian decision rule based algorithm to solve the problem optimally with a constant time complexity. To prioritize PU transmissions, we then re-formulate the throughput maximization problem by adding a constraint on the PU throughput. The new problem is shown to be strongly NP-hard and solved via a greedy algorithm with time complexity $O\left(\frac{N^5}{\log^2 \frac{1}{1-\epsilon}}\right)$ where $N$ is the total number of SUs. The algorithm achieves a throughput strictly greater than $\frac{1}{2}(1 - \epsilon)$ of the optimal solution and results in a small constraint violation that goes to zero with $\epsilon$. We also put a constraint on the sensing time overhead, which limits the number of SUs that can participate in cooperative sensing, and investigate this constrained case. We illustrate the efficacy of the performance of our algorithms and provide sensitivity analysis via a numerical investigation.
3.1 Introduction

Sensing helps SUs be aware of the PU existence so that they can transmit without causing disruptions to PUs. Individually sensed signal is weak that leads to incorrect decisions. Cooperative sensing has been proposed to address these problems [25][48][54]. In this chapter, sensing quality is our main concern, which has two components: misdetection probability and false alarm probability. As a unifying objective, we adopt system throughput as a means to quantify the effects of misdetection and false alarm probabilities on the system performance. We design an optimal data fusion rule to (hard) combine the reported sensing results. More specifically, we aim to maximize the system throughput in a CRN composed of multiple PUs operating on orthogonal channels and SUs where SUs are allowed to sense all the channels in the network. We assume that sensing decisions are made on a per channel basis and each SU can sense all the PUs in the system. Thus, we only need to solve the system throughput maximization problem per channel. Our main contributions can be summarized as follows:

- In contrast to previous works that restrict the class of fusion rules, we propose a Bayesian decision rule based algorithm to solve the throughput maximization problem optimally with constant time complexity.

- To guarantee that the PU throughput is at least a prescribed minimum fraction $\delta$, we extend the formulation by adding a constraint on the PU throughput. This constrained problem is shown to be strongly NP-hard by a reduction from the product partition problem [9]. A greedy algorithm is developed with the time complexity $O\left(\frac{N^5}{\log^2 \frac{1}{1-\epsilon}}\right)$ where $\epsilon = 1 - 10^{-N^{10-r}}$, $N$ is the total number of SUs, and $r$ is the decimal places kept for the input. This approximation algorithm is analytically shown to achieve the system throughput (the sum of PU throughput...
and SU throughput) strictly greater than $\frac{1}{2}(1 - \epsilon)$ of the optimal solution. The PU throughput fraction achieved is shown to be at least $\frac{\delta}{1 - \epsilon} - \frac{\epsilon}{1 - \epsilon}$.

• We investigate systems where limited sensing overhead is allowed, i.e., the number of sensing SUs is restricted. Our theoretical results show that the performance of cooperative sensing is monotonic in the number of SUs used for sensing.

In the remainder of the chapter, we first introduce our system model in Section 3.2. The system throughput maximization problem is formulated in Section 3.3, and solved optimally via the Bayesian decision rule. In Section 3.4, the constrained maximization problem is formulated, which is shown to be strongly NP-hard. A greedy algorithm is proposed with an approximation factor strictly greater than $\frac{1}{2}(1 - \epsilon)$. Another direction is considered in Section 3.5 where the system throughput performance is investigated subject to a constraint on the number of sensing SUs used. In Section 3.6, numerical results are presented to illustrate the performance of our algorithms. We summarize this chapter in Section 3.7.

### 3.2 System Model

We consider a time-slotted cognitive radio network in which multiple PU networks, consisting of a PU base station (PU-BS) and PU receivers in each network, co-exist in the same area with an SU base station (SU-BS) and $M$ SUs (Figure 3.1). Since PUs that are in the interference range of each other operate on orthogonal channels, for the purpose of the analysis, one can focus on a single PU and SU parameters corresponding to that PU. We consider the uplink part for the SU system, i.e., only one SU can be active and transmit to the SU-BS at any given time over the same channel. For each PU network, we denote the set of all SUs by $\mathcal{S}$ and the set of
Figure 3.1: System model of an SU network coexisting with three PU networks. Small circles are SUs and rectangles are PUs. Each big circle represents the transmission range of the corresponding SU-BS/PU-BS. Each SU can sense all PUs in the system.

SUs whose uplink transmission causes interference to any PU receivers by $S$ and $|S| = N$ ($|S| = M \geq N$). They are indexed from 1 to $N$. SUs outside $S$ can use the corresponding PU channel to transmit at any time slot without causing interference to the PUs. For instance, PUs 1, 4, and 8 lie in the interference range of SUs in Figure 3.1, and any transmission from SUs 1 and 2 may cause interference to PU 1.

SUs in $S$ are close to the PU network and they are used to sense the channel cooperatively to reduce the sensing errors. The sensing results of individual SUs are assumed to be independent. We assume that each SU is allowed to sense any number of channels. Thus sensing decisions can be made per channel and we investigate the sensing behaviors of SUs on each channel separately. Let $B$ represent the PU activity such that $B = 1$ if PU is active, and $B = 0$ otherwise. Let $P_f^i$ denote the
probability of a false alarm for SU $i$, which is the probability that SU $i$ senses the PU to be active given that the PU is actually idle. $P^i_m$ represents the probability of mis-detection for SU $i$, which is the probability that SU $i$ senses the PU to be idle given that the PU is actually active.

**Cooperative Sensing:** Multiple SUs are chosen to sense the channel and the SU-BS predicts PU activity by collecting the sensing results from these SUs. We let $S_0$ denote the set of SUs that participate in cooperative sensing. Note that $S_0 \subseteq S \subseteq S$. In the cooperative sensing model, we assume that the SU-BS collects sensing results from SUs in $S_0$.

**Cooperative Sensing Indicator:** Let $Y_i$ denote how SU $i$ senses the PU activity, which is a random variable. More precisely, $Y_i = 1$ indicates that SU $i$ observes the PU to be active, while $Y_i = 0$ indicates that SU $i$ observes the PU to be idle. In this paper, our objective is to maximize the system throughput by characterizing $S_0$ and estimating the PU activity based on observations from $S_0$ (called the decision rule). The decision rule is denoted as a function $F : \{0, 1\}^{|S_0|} \rightarrow \{0, 1\}$. The observations form a vector $Y$, where $Y \in \{0, 1\}^{|S_0|}$ while the decision is denoted by $Z$ where $Z \in \{0, 1\}$, i.e., $Z = F(Y)$. The false alarm probability of cooperative sensing is denoted by $P^c_f = P(Z = 1|B = 0)$. The mis-detection probability of cooperative sensing is denoted by $P^c_m = P(Z = 0|B = 1)$. Each time slot is divided into a control slot $T_c$ and a data slot $T_d$ where $T_c + T_d = 1$ (Figure 3.2). In the control slot, the SU-BS collects sensing results from the set of SUs $S_0$ and notifies an SU in $S$ if the cooperative sensing result is “idle” ($Z = 0$). If the PU is active (mis-detection), the PU transmission will collide with the transmission from the SU. The length $T_c$ of the control slot is regarded as the sensing overhead and assumed to be constant.

\[1\text{Note that } S_0 = S \text{ in Sections 3.3 and 3.4 where there is no budget constraint on the number of SUs sensing the channel; } S_0 \subseteq S \text{ in Section 3.5 where the size of } S_0 \text{ is constrained.}\]
Figure 3.2: Control slot $T_c$ and data slot $T_d$. The circles are SUs.

throughout the paper, that is, a fixed time period is allocated for cooperative sensing in each slot.

The uplinks of SUs in $S$ are assumed to have the same capacity which is normalized to 1. We assume that SUs in $S$ are always backlogged. The scheduling of the transmitting SUs is beyond the scope of this paper. However, any work-conserving scheduling policy operating on idle slots can be used together with the decision rule to maximize the total system throughput. We let $\pi_0$ denote the probability that the PU is idle and we assume that the prior distribution of PU activity is accurately acquired over time. Our only assumption is that state changes occur at the beginning of a time slot. The average throughput of PUs whose transmission would be interfered by SUs in $S$ is denoted as $\gamma$. Table 3.1 summarizes the notations used in the paper.

The outline of operations for cooperative sensing is as follows. In Sections 3.3 and 3.4, we will find the optimal decision rule $F$ without and with PU throughput constraint, respectively. The problem of selecting sensing set $S_0$ will be studied in Section 3.5.

1) Each SU $i$ reports its probability of misdetection ($P^i_m$) and probability of false alarm ($P^i_f$) to the SU-BS;
2) The SU-BS determines the sensing set \( S_0 \) and the decision rule \( F \) based on \( P_{m}^{i} \), \( P_{f}^{j} \)’s and the optimization metric;

3) The SU-BS notifies SUs in \( S_0 \) with an \( ACK \) and also assigns each one of them a \( SEQ \) number for reporting sensing results;

4) SUs receiving an \( ACK \) sense the channel and report the results to SU-BS in the order of \( SEQ \);

5) SU-BS makes the decision of the PU activity based on the sensing results and \( F \) and schedules an SU for transmission if the decision is 0 (PU idle).

In Sections 3.3, 3.4, and 3.5, we formulate three system throughput maximization problems and study the solutions, respectively. Considering that the sensing decisions are made per channel, only a simplified single-channel problem is investigated in each section and the sum of maximum system throughput over each channel leads to the maximum system throughput of the entire PU-SU network.

### 3.3 System Throughput Maximization

In this section, we formulate the cooperative sensing problem with the assumption that the sensing set is fixed to be \( S \), that is, the sensing results from all SUs in \( S \) are reported to SU-BS within \( T_c \). SUs outside \( S \) can transmit without causing interference to the PUs. Thus, their performance does not depend on the choice of the sensing set or the decision rule. Our goal is to maximize the sum of the expected throughput of the SUs in \( S \) and that of the PUs whose transmission may be interfered by the SUs. It is equivalent to maximizing the expected throughput of the system with PU-SU co-existence. Instead of an abstract measure of sensing quality [80], we choose the system throughput to combine the effects of misdetection and false alarm probabilities in a meaningful manner. Misdetection and false alarm probabilities are related to PU throughput and SU throughput, respectively.
3.3.1 Problem Formulation

In this section, we formulate the system throughput maximization problem where the system throughput consists of both expected SU throughput and PU throughput defined as follows.

Let \( y_i \) denote the observation that SU \( i \) senses the PU to be idle \( (y_i = 0) \) or busy \( (y_i = 1) \). Further, let \( y \) denote the observation vector, i.e., \( y = \{y_i\}_{i \in S} \). Now, given \( B = 0 \) (the PU is idle), the probability of a particular observation vector \( y \) occurring is

\[
P(Y = y | B = 0) = \prod_{i \in S, y_i = 1} P_i \prod_{j \in S, y_j = 0} (1 - P_j).
\]

(3.3.1)

Now, the probability that the decision of cooperative sensing is idle given that the PU is idle involves summing over all values of \( y \), such that the decision \( F(y) = 0 \), i.e.,

\[
P(Z = 0 | B = 0) = \sum_{y : F(y) = 0} P(Y = y | B = 0).
\]

(3.3.2)

Hence, the false alarm probability of cooperative sensing is

\[
P_f^c = 1 - P(Z = 0 | B = 0) = 1 - \sum_{y : F(y) = 0} P(Y = y | B = 0).
\]

(3.3.3)

Likewise, given \( B = 1 \) (the PU is active), the probability of a particular observation vector \( y \) occurring is

\[
P(Y = y | B = 1) = \prod_{i \in S, y_i = 1} (1 - P_i) \prod_{j \in S, y_j = 0} P_j.
\]

(3.3.4)

And

\[
P(Z = 1 | B = 1) = \sum_{y : F(y) = 1} P(Y = y | B = 1).
\]

(3.3.5)
Then, the mis-detection probability of cooperative sensing is

\[ P_m^c = 1 - P(Z = 1 | B = 1) = 1 - \sum_{y:F(y)=1} P(Y = y | B = 1). \] (3.3.6)

Note that Equation (3.3.2) is the conditional probability that SU-BS correctly identifies the PU activity when it is idle so that one SU could transmit successfully; Equation (3.3.5) is the conditional probability that SU-BS correctly detects the PU is active so that no SU would transmit and the PU could transmit successfully. Hence, the expected throughput of the SUs can be represented by

\[ (1 - T_c) P(B = 0, Z = 0) = (1 - T_c) \pi_0 P(Z = 0 | B = 0) \]

\[ = (1 - T_c) \pi_0 \sum_{y:F(y)=0} P(Y = y | B = 0), \] (3.3.7)

since the uplinks of SUs in S have unit capacity and only one of them could be scheduled in each time slot. Now, let the PU capacity be \( \gamma \). Then, the expected throughput of the PU in the interference range of the SUs is given by (3.3.8).

\[ \gamma P(Z = 1 | B = 1) = \gamma \sum_{y:F(y)=1} P(Y = y | B = 1) \] (3.3.8)

The problem of interest to us is formulated as follows:

Problem (A):

\[ \max_F (1 - T_c) \pi_0 \sum_{y:F(y)=0} P(Y = y | B = 0) + \gamma \sum_{y:F(y)=1} P(Y = y | B = 1) \]

Note that the system throughput as the objective function in Problem (A) combines the effects of misdetection and false alarm probabilities, which is more meaningful than an arbitrary weighted sum of them. A nice property of this objective function is that the SU capacity \( (1 - T_c) \pi_0 \) and PU capacity \( \gamma \) are taken into account.
Algorithm 1 Bayesian Decision Rule Based Algorithm for maximizing the system throughput (given $\mathbf{Y} = \mathbf{y}$, decide $Z$)

1: if $(1 - T_c)\pi_0 \prod_{y_i = 1} P_i \prod_{y_j = 0} (1 - P_j) \geq \gamma \prod_{y_i = 1} (1 - P_{m,i}) \prod_{y_j = 0} P_{m,j}$ then
2: \hspace{1em} $Z \leftarrow 0$
3: \hspace{1em} else
4: \hspace{1.2em} $Z \leftarrow 1$

3.3.2 Optimal Solution with Bayesian Decision Rule

We show that Problem (A) can be converted to a Bayesian Decision problem. Algorithm 1 is then developed based on the Bayesian decision rule to minimize the posterior expected loss [6] and it is of constant time complexity.

To solve Problem (A), we formulate the equivalent problem as follows.

$$
\min_F L(B = 0, Z = 1) \left[ \pi_0 \sum_{\mathbf{y}: F(\mathbf{y}) = 1} P(\mathbf{Y} = \mathbf{y} | B = 0) \right] 
+ L(B = 1, Z = 0) \left[ (1 - \pi_0) \sum_{\mathbf{y}: F(\mathbf{y}) = 0} P(\mathbf{Y} = \mathbf{y} | B = 1) \right], \tag{3.3.9}
$$

where $L(B, Z)$ is the loss of decision $Z$ based on observation $\mathbf{x}$, which is a non-negative number. $L(B = 0, Z = 1) = (1 - T_c)$ and $L(B = 1, Z = 0) = \frac{\gamma}{1 - \pi_0}$. Thus Equation (3.3.9) is the posterior expected loss of decision $Z$ (Definition 8 of Chapter 4.4 in [6]). Using the Bayesian decision rule, Problem (3.3.9) can be solved optimally [6]: given $\mathbf{Y} = \mathbf{y}$, the decision $Z = 1$ if $L(B = 0, Z = 1)\pi_0 P(\mathbf{Y} = \mathbf{y} | B = 0) < L(B = 1, Z = 0)(1 - \pi_0)P(\mathbf{Y} = \mathbf{y} | B = 1)$ and $Z = 0$ otherwise. Algorithm 1 is designed accordingly.

3.4 Guaranteeing a Target PU Throughput

In this section, we investigate the maximum throughput problem under a PU throughput constraint. The reason why this is important is to ensure that the PU receives
at least a guaranteed amount of throughput. We first show that this constrained problem is strongly NP-hard by reducing the classical product partition problem [9] to it. Then a greedy approximation algorithm is proposed to achieve throughput that is strictly greater than $\frac{1}{2}(1-\epsilon)$ of the optimal solution. The complexity of the algorithm is shown to be $O(\frac{N^5}{\log^2 \frac{1}{1-\epsilon}})$ by solving a two-dimensional dynamic programming problem. Note that the algorithm only needs to run once until $P_m^i$ or $P_f^i$ changes.

### 3.4.1 Problem Formulation and Properties

We formulate the constrained optimization problem as follows:

Problem (B):

$$\max_F (1 - T_c)\pi_0 \sum_{y: F(y) = 0} P(Y = y | B = 0) + \gamma \sum_{y: F(y) = 1} P(Y = y | B = 1)$$

s.t. $\gamma \sum_{y: F(y) = 1} P(Y = y | B = 1) \geq \alpha. \quad (3.4.1)$

Equation (3.4.1) is the constraint we put on Problem (B) where the expected PU throughput must be no less than a preset system-dependent threshold. Note that $\alpha = \beta \cdot \gamma$ where $\beta$ is defined in Section 3.1. Problem (B) maximizes the expected system throughput given that the lowest PU throughput can be met. This is important because in cognitive radio applications, PU transmissions need to have higher importance than SU transmissions. Note that for the multichannel formulation, $\alpha$ in Equation (3.4.1) varies over difference PUs. By solving the constrained optimization problem on each channel and summing the throughput, we get the optimal system throughput across all channels subject to the throughput constraints of all PUs.

We define $\mathcal{Y}_0 = \{y | F_b(y) = 0\}$, $\mathcal{Y}_1 = \{y | F_b(y) = 1\}$ where $F_b$ is the Bayesian rule. We also define $\mathcal{Y}_0^* = \{y | F^*(y) = 0\}$ and $\mathcal{Y}_1^* = \{y | F^*(y) = 1\}$ where $F^*$ is the
optimal solution to Problem (B). For convenience, we define \( G(y) = (1 - \pi_0) P(Y = y | B = 0) \) and \( H(y) = \gamma P(Y = y | B = 1) \). Note that if \( G(y) \geq H(y) \), we have \( F_b(y) = 0 \); otherwise \( F_b(y) = 1 \). By observing the structure of Problem (B), we state Lemma 3.4.1 and show that \( \mathcal{Y}_1 \subseteq \mathcal{Y}_1^* \). In other words, observations that have decision 1 by the Bayesian rule have decision 1 as well in the optimal decision.

**Lemma 3.4.1.** In the optimal solution to Problem (B), all observations \( y \) with \( F_b(y) = 1 \) has the property that \( F^*(y) = 1 \).

*Proof.* (By contradiction) Assume that \( \mathcal{Y}_1 \not\subseteq \mathcal{Y}_1^* \), that is, \( F_b(y) = 1 \) and \( F^*(y) = 0 \) for some \( y \) (\( y \in \mathcal{Y}_1 \cap \mathcal{Y}_0^* \)). By Bayesian rule, it means \( G(y) < H(y) \). We find another rule \( \tilde{F} \) where \( \tilde{F}(y) = 1 \) if \( y \in \mathcal{Y}_1 \cap \mathcal{Y}_0^* \); otherwise \( \tilde{F}(y) = F^*(y) \). Obviously,

\[
\sum_{y:F_b(y)=1} P(Y = y | B = 1) > \sum_{y:F^*(y)=1} P(Y = y | B = 1)
\]

so that this operation still results in a feasible solution. Furthermore, the expected system throughput increases considering \( G(y) < H(y) \), which results in a better solution than the current optimal one. It causes a contradiction. Hence, we have \( F^*(y) = 1 \) for all \( y \in \mathcal{Y}_1 \). \( \square \)

With the property of Lemma 3.4.1, to solve Problem (B), we only need to find the set \( \mathcal{Y}_0 \setminus \mathcal{Y}_0^* \) that is composed of observations \( y \) with \( F_b(y) = 0 \) and \( F^*(y) = 1 \).

### 3.4.2 Proof of Strong NP-hardness

In this section, we show that Problem (B) is strongly NP-hard. By Lemma 3.4.1, it suffices to show the following problem to be strongly NP-hard: finding all \( y \) with \( F_b(y) = 0 \) and \( F^*(y) = 1 \). Recall that a problem is said to be strongly NP-complete, if it remains so even when all of its numerical parameters are bounded by a polynomial in the length of the input. A problem is strongly NP-hard if a strong NP-complete problem can be reduced to it in polynomial time [24].

**Theorem 3.4.2.** Problem (B) is strongly NP-hard.
Proof. We will reduce the product partition problem to the equivalent problem stated above and the strong NP-hardness of Problem (B) can be proved accordingly. We first state the product partition problem [9] - Given $N$ positive integers: $y_1, \cdots, y_N$, is there a way to have them partitioned into two equal-sized subsets that have the same product? For the reduction, we construct an instance of Problem (B) by setting $(1 - T_c)\pi_0 = \gamma$, $\alpha = \epsilon + \sum_{y: G(y) < H(y)} H(y)$ with $\epsilon \leq \min_{y: G(y) \geq H(y)} H(y)$. For this instance, putting any $y$ with $G(y) \geq H(y)$ to $Z = 1$ would make a feasible solution given that observations with $G(y) < H(y)$ have all been put in $Z = 1$. Choosing the observation with the minimum non-negative $G(y) - H(y)$ would be the optimal solution. Note that $G(y) - H(y) = 0$ is equivalent to $\frac{G(y)}{H(y)} = 1$. By setting $\frac{1 - P_j^i}{P_m^i} = \frac{1 - P_j^i}{P_m^i} = \eta_i$ for all $i$, we have $\frac{G(y)}{H(y)} = \prod_{y_i = 0, i = 1, \cdots, N} \eta_i \cdot \prod_{y_j = 1, j = 1, \cdots, N} \frac{1}{\eta_j}$. Now the instance becomes: given $N$ pairs of integers $(\eta_1, \frac{1}{\eta_1}), \cdots, (\eta_N, \frac{1}{\eta_N})$, exactly one number should be chosen from each pair; with this constraint, what is the minimum product that is no less than 1? Note that the operations above take polynomial time.

To verify the correctness of the reduction, we can check: if the minimum $\frac{G(y)}{H(y)}$ no less than 1 is 1, that is, the optimal solution of the instance is 1, we can answer “Yes” to the partition problem; if it is greater than 1, we can answer “No” to the partition problem. If Problem (B) can be solved in polynomial time, then the product partition problem can be solved in polynomial time as well. The product partition problem is well-known to be strongly NP-complete [9]. Assuming $P \neq NP$, Problem (B) has been proven to be strongly NP-hard.

It has been shown in Theorem 3.4.2 that finding the observation with $G(y)$ closest to $H(y)$ from above is strongly NP-hard. Hence, unless P=NP, one cannot even find a pseudo-polynomial time algorithm to solve Problem (B) [74]. Hence, we next focus on designing a good approximation algorithm.
3.4.3 Greedy Approximation Algorithm

We propose a greedy algorithm (Algorithm 2) that initially assigns all observations to $Z = 1$ and then moves observations with $G(y) \geq H(y)$ by $\frac{G(y)}{H(y)}$ from the highest to lowest to $Z = 0$ until the feasibility constraint of Problem (B) is violated. By transforming Problem (B) into the Knapsack Problem [74], we will show that the algorithm achieves strictly greater than $1/2$ of the optimal solution for Problem (B). Although the sum of $G(y)$ or $H(y)$ in the worst case has an exponential number of terms, we will design an approximation algorithm in Section 3.4.4.

In Algorithm 2, observations are chosen by $\frac{G(y)}{H(y)}$ from the highest to the lowest and assigned to $Z = 0$ after those with $G(y) < H(y)$ are assigned to $Z = 1$. Ties are broken by putting observations with smaller $H(y)$ in the front. In Lines 3-4, the algorithm checks whether a feasible solution exists for the given input by comparing the extreme case where all observations are assigned to $Z = 1$ ($\sum_{y} H(y) = \gamma$) with the threshold $\alpha$. In Line 5, observations are initialized to $Z = 1$. Lines 5-7 checks whether the feasibility constraint in Problem (B) has been satisfied under the initial assignment. If yes, observations with $G(y) \geq H(y)$ are assigned to $Z = 0$ by Bayesian decision rule. Lines 8-12 searches for observations with $G(y) \geq H(y)$ from the highest $\frac{G(y)}{H(y)}$ to lowest until $\sum_{y} H(y) \leq \gamma - \alpha$ is violated (Line 11). Note that $\sum_{y: F(y) = 0} H(y) \leq \gamma - \alpha$ and $\sum_{y: F(y) = 1} H(y) \geq \alpha$ (feasibility constraint) are equivalent since $\sum_{y} H(y) = \gamma$. $F(y)$ of these observations are set to be 0 (Line 12) in the searching process. To guarantee the $\frac{1}{2}$ approximation ratio, we have to do the comparison in Lines 13-14 (shown in the proof of Theorem 3.4.3). Next, we state Theorem 3.4.3 that gives the approximation factor of Algorithm 2.

**Theorem 3.4.3.** Algorithm 2 achieves strictly greater than $1/2$ of the optimal solution to Problem (B).
Algorithm 2 Greedy Approximation Algorithm for Problem (B)

Input: $N$, $T_c$, $\pi_0$, $\gamma$, $\alpha$, $P_m^i$, $P_f^j$ for all $i$

Output: $F$ or “infeasible”

1: $G(y) \leftarrow (1 - T_c) \pi_0 \prod_{i \in S, y_i = 1} P_f^i \prod_{j \in S, y_j = 0} (1 - P_f^j)$ for all $y$

2: $H(y) \leftarrow \gamma \prod_{i \in S, y_i = 1} (1 - P_m^i) \prod_{j \in S, y_j = 0} P_m^j$ for all $y$

3: if $\gamma < \alpha$ then

4: output “infeasible” and return

5: $F(y) \leftarrow 1$ for all $y$, $sum1 \leftarrow \sum_{y : G(y) < H(y)} H(y)$

6: if $sum1 \geq \alpha$ then

7: $F(y) = 0$ for all $y$ with $G(y) \geq H(y)$ and return

8: Sort $y$’s with $G(y) \geq H(y)$ in non-increasing order of $G(y) / H(y)$ and denote them as

9: $sum2 \leftarrow 0$

10: for $i = 1$ to $l$ do

11: if $sum2 + H(y_i) > \gamma - \alpha$ then break

12: $sum2 \leftarrow sum2 + H(y_i)$, $F(y_i) \leftarrow 0$

13: if $\sum_{n=1}^{i-1} (G(y_n) - H(y_n)) < G(y_i) - H(y_i)$ then

14: $F(y_n) \leftarrow 0$ for all $n = 1, \cdots, i - 1$

Proof. We define

$A = \sum_{y : y \in \mathcal{Y}_i} H(y)$, $B = \sum_{y : y \in \mathcal{Y}_0 \setminus \mathcal{Y}_0^*} G(y)$, $B' = \sum_{y : y \in \mathcal{Y}_0 \setminus \mathcal{Y}_0^*} H(y)$ ($B \geq B'$ by Bayesian rule), $C = \sum_{y : y \in \mathcal{Y}_0^*} G(y)$, and $C' = \sum_{y : y \in \mathcal{Y}_0^*} H(y)$ ($C \geq C'$ by Bayesian rule).

Then, $A + B + C$ is the optimal solution to Problem (B) without the PU throughput constraint since
\[ A + B + C = A + (B + C) = \sum_{y:y \in \mathcal{Y}_1} H(y) + \sum_{y:y \in \mathcal{Y}_0} G(y) \]

and \( A + B' + C \) is the optimal solution to Problem (B) since

\[ A + B' + C = (A + B') + C = \sum_{y:y \in \mathcal{Y}_1} H(y) + \sum_{y:y \in \mathcal{Y}_0} G(y) \]

which is no greater than \( A + B + C \). Note that

\[ A + B' + C' = \sum_{y} H(y) = \gamma. \]

Let \( APX \) be the solution to Problem (B) output by Algorithm 2. Let \( OPT \) be the optimal solution to Problem (B). Then, we have

\[ OPT = A + B' + C = \gamma + (C - C'). \]  \hspace{1cm} (3.4.2)

We then show that \( \mathcal{Y}_0^* \) is the optimal solution to Problem (3.4.3) and \( C - C' = \sum_{y:y \in \mathcal{Y}_0} (G(y) - H(y)) \) is the optimal objective value.

\[
\max_{W:W \subseteq \mathcal{Y}_0} \sum_{y:y \in W} (G(y) - H(y)) \\
\text{s.t. } \sum_{y:y \in W} H(y) \leq \gamma - \alpha \]  \hspace{1cm} (3.4.3)

Clearly, we only need to show that the constraint of Problem (3.4.3) and that of Problem (B) are equivalent. By definitions, we have
\[ \gamma \sum_{y:F(y)=1} P(Y=y|B=1) \geq \alpha \]
\[ \Leftrightarrow \sum_{y:F(y)=1} H(y) \geq \alpha \]
\[ \Leftrightarrow \sum_{y:y \in Y_1} H(y) + \sum_{y:y \in Y_0 \setminus W} H(y) \geq \alpha \]
\[ \Leftrightarrow \sum_{y:y \in Y_1} H(y) + \sum_{y:y \in Y_0 \setminus W} H(y) - \alpha \geq 0 \]
\[ \Leftrightarrow \sum_{y:y \in W} H(y) \leq \gamma - \alpha. \]

Note that (a) holds because \( W \) corresponds to all observations \( y \) with \( F(y) = 0 \).

Problem (3.4.3) is a Knapsack Problem and can be solved by a greedy approach [74]: choosing observations with \( G(y_i) \geq H(y_i) \) from the highest \( \frac{G(y_i) - H(y_i)}{H(y_i)} \) to the lowest until (3.4.3) is violated (the index of the observation added when the constraint is violated is labeled as \( s \)), which is exactly what we do in Algorithm 2 since \( \frac{G(y_i) - H(y_i)}{H(y_i)} \geq \frac{G(y_j) - H(y_j)}{H(y_j)} \) if and only if \( \frac{G(y_i)}{H(y_i)} \geq \frac{G(y_j)}{H(y_j)} \); A further comparison to find the maximum of \( \sum_{n=1}^{s-1} (G(y_n) - H(y_n)) \) and \( G(y_s) - H(y_s) \) guarantees \( \frac{1}{2}(C - C') \) [74]. Hence, \( APX \geq \gamma + 1/2(C - C') \) holds. Since \( \gamma > 0 \), we always have \( APX/OPT > 1/2 \) for Problem (B).

So far, we have shown that the greedy algorithm (Algorithm 2) gives an approximation factor of strictly greater than 1/2 for Problem (B). However, when \( \gamma \gg C - C' \), this factor could be arbitrarily close to 1.

### 3.4.4 Approximate Throughput Calculation

In Lines 5, 8 and 11 of Algorithm 2, we need to calculate the sum of exponential number of terms in the worst case due to its combinatorial nature. We design
an algorithm by means of dynamic programming to find the joint distribution of
\( \log \frac{G(y)}{H(y)} \), and further the sum of \( G(y) \) or \( H(y) \) for \( y \) in different sets can be calculated. Note that 1) We take logarithmic functions to make the recursive function additive (Line 15 in Algorithm 3); 2) We find the joint distribution of \( \log \frac{G(y)}{H(y)} \), \( \log H(y) \) instead of \( \log G(y), \log H(y) \) because we need to evaluate \( \frac{G(y)}{H(y)} \) in Lines 5 and 8 in Algorithm 2. The details of the algorithm will be introduced next, followed by the complexity analysis.

In Algorithm 3, we use dynamic programming to calculate the joint distribution of \( \log \frac{G(y)}{H(y)} \) and \( \log H(y) \), which counts the number of observations with the same \( \log \frac{G(y)}{H(y)} \) and the same \( \log H(y) \). Note that we only need to run this algorithm once in the time period where \( P_{m}, P_{f}, \pi_{0} \) and \( \gamma \) are fixed. Lines 3, 6, 9 and 12 of Algorithm 2 can be calculated based on these counts. \( \text{round}(b, r) \) rounds \( b \) to \( r \) decimal places by removing all digits after \( r \) decimal places. We use \( \text{round}(b, r) \times 10^{r} \) to scale and round a real \( b \) to an integer. The values of \( r \) lead to different accuracy levels for the algorithm. \( Q \) and \( q \) specify the maximum and minimum contribution, respectively, an observation \( y \) can have to \( \log \frac{G(y)}{H(y)} \), while \( Q' \) and \( q' \) specify the maximum and minimum contribution an observation \( y \) can have to \( \log H(y) \) respectively. Let \( y_{i}^{i} = \{y_{1}, \ldots, y_{i}\} \) denote the observation vector for SU 1 to \( i \). \( C(i, j, j') \) is defined as the number of observations with \( \log \frac{G(y_{i}^{i})}{H(y_{i}^{i})} \) (after rounding) equal to \( j \) and \( \log H(y_{i}^{i}) \) (after rounding) equal to \( j' \). In particular, \( C(N, j, j') \) records the number of observations with \( \log \frac{G(y_{N}^{N})}{H(y_{N}^{N})} \) (after rounding) equal to \( j \) and \( \log H(y_{N}^{N}) \) (after rounding) equal to \( j' \). Lines 9-15 use iterations to find \( C(i, j, j') \) for all \( i = 1, \ldots, N, q \leq j \leq Q \) and \( q' \leq j' \leq Q' \). The recursive function in Line 15 distinguishes two situations: if \( y_{i+1} = 0 \), \( \log \frac{G(y_{i}^{i})}{H(y_{i}^{i})} \) is increased by \( a_{i+1} \) and \( \log H(y_{i}^{i}) \) is increased by \( \lambda_{i+1} \); on the other hand, if \( y_{i+1} = 1 \), \( \log \frac{G(y_{i}^{i})}{H(y_{i}^{i})} \) is increased by \( z_{i+1} \) and \( \log H(y_{i}^{i}) \) is increased by
Note that Line 15 may encounter $C(i, j, j')$ beyond the boundaries of $j$ or $j'$, the value of which will be treated as 0. Lines 3, 6, 9 and 12 of Algorithm 2 can be calculated accordingly, the time complexity of which is dominated by that of Algorithm 3. For special cases satisfying one or more of the following conditions: $P^i_m = 0$, $P^i_f = 1$, $P^i_f = 0$ and $P^i_f = 1$, the values of $G$ or $H$ are straightforward which does not require running Algorithm 3. For instance, when $P^i_m = 0$, $H(y) = 0$ for all observations with $y_i = 0$. In the following, we consider only other more general cases. We will first show the tradeoff between the accuracy of $G(y)$ or $H(y)$ calculation and the time complexity of Algorithm 3 in Lemma 3.4.4. Next, we prove that Algorithm 2, together with Algorithm 3, can achieve strictly greater than $\frac{1}{2}(1 - \epsilon)$ of the optimal solution, where $\epsilon \in (0, 1)$ is a constant, with the time complexity $O(\frac{N^5}{\log^2 \frac{1}{1 - \epsilon}})$, and we also bound the feasibility gap in Theorem 3.4.5. As $\epsilon$ decreases, better accuracy is achieved at the cost of higher time complexity. The algorithm only needs to run once before $P^i_m$ or $P^i_f$ changes. We define $G'(y)$ and $H'(y)$ as the values of $G(y)$ and $H(y)$, respectively, calculated by Algorithm 3.

**Lemma 3.4.4.** With the complexity of $O(\frac{N^5}{\log^2 \frac{1}{1 - \epsilon}})$, Algorithm 3 calculates $G'(y) \geq (1 - \epsilon)G(y)$ and $H'(y) \geq (1 - \epsilon)H(y)$.

Proof. The rounding in Lines 1-4 makes $\log \frac{1 - P^i_f}{P^i_m}$, $\log \frac{P^i_f}{1 - P^i_m}$, $\log P^i_m$ and $\log 1 - P^i_m$ lose at most $10^{-r}$ in their values, respectively. By the definition of $G(y)$, we have $\log G(y) - \log G'(y) \leq N10^{-r}$, which is equivalent to $G'(y) \geq 10^{-N10^{-r}}G(y)$. Similarly, we have $H'(y) \geq 10^{-N10^{-r}}H(y)$. Let $\epsilon = 1 - 10^{-N10^{-r}}$, then given the input $P^i_m$ and $P^i_f$, the complexity of Algorithm 3 is $O(N^310^{2r})$, which is $O(N^5/\log^2 \frac{1}{1 - \epsilon})$. □

Based on Lemma 3.4.4, we prove the approximation factor of $\frac{1}{2}(1 - \epsilon)$ in the following theorem. We also characterize the feasibility gap which tends to 0 as $\epsilon$ goes to 0.
Theorem 3.4.5. Algorithm 2, together with Algorithm 3, achieves strictly greater than $\frac{1}{2}(1 - \epsilon)$ of the optimal solution with the time complexity of $O(\frac{N^5}{\log^2 \frac{1}{1-\epsilon}})$; it also achieves a PU throughput fraction of at least $\frac{\delta}{1 - \epsilon} - \frac{\epsilon}{1 - \epsilon}$ where $\delta \cdot \gamma = \alpha$ as defined in Section 3.1.

Proof. As in the proof of Theorem 3.4.3, we focus on the equivalent Problem (3.4.3). We denote the optimal assignment of observations without any approximations of $G$ or $H$ by $\Gamma$, the optimal assignment of observations with the approximations of $G$ or $H$ in Algorithm 3 by $\Gamma'$, and the assignment generated by Algorithm 2 with Algorithm 3 by $\Gamma'_g$. We also denote the value of the objective function in Problem (3.4.3) by $\Theta(\cdot)$ and the approximated value of the objective function in Problem (3.4.3) (by the calculation of Algorithm 3) by $\Theta'(\cdot)$, given the observation assignment. Then,

$$\Theta(\Gamma'_g) \geq \Theta'(\Gamma'_g) \geq \frac{1}{2}\Theta'(\Gamma') \geq \frac{1}{2}\Theta'(\Gamma) \geq \frac{1}{2}(1 - \epsilon)\Theta(\Gamma).$$

where (a) is by the rounding assumption, (b) is by Theorem 3.4.3, (c) is by the definition of $\Gamma'$, and (d) is by Lemma 3.4.4. We denote the optimal solution to Problem (B) by $OPT$ and the solution to Problem (B) output by Algorithm 2 together with Algorithm 3 by $APX'$, respectively. Then, $OPT = \gamma + \Theta(\Gamma)$ and $APX' \geq \gamma + 1/2(1 - \epsilon)\Theta(\Gamma)$ following a similar argument in the proof of Theorem 3.4.3. Hence, we always have $APX'/OPT > 1/2(1 - \epsilon)$.

On the other hand, the complexity is dominated by that of Algorithm 3, which is $O(\frac{N^5}{\log^2 \frac{1}{1-\epsilon}})$ as shown in Lemma 3.4.4.

To check the feasible gap, we denote the set of observations assigned to $O = 0$ in $\Gamma'_g$ by $\Delta$. By Line 11 in Algorithm 2, we have $\sum_{y \in \Delta} H'(y) \leq \gamma - \alpha$. Also by Lemma 3.4.4, $\sum_{y \in \Delta} H'(y) \geq (1 - \epsilon) \sum_{y \in \Delta} H(y)$ holds. Then we have $\sum_{y \in \Delta} H(y) \leq \frac{\gamma - \alpha}{1 - \epsilon}$. 

47
The PU throughput achieved can be represented by
\[
\sum_{y \notin \Delta} H(y) \geq \gamma - \frac{\gamma - \alpha}{1 - \epsilon} = \frac{\alpha}{1 - \epsilon} - \gamma \frac{\epsilon}{1 - \epsilon}.
\]
The PU throughput fraction is then calculated as
\[
\frac{\sum_{y \notin \Delta} H(y)}{\gamma} \geq \frac{\delta}{1 - \epsilon} - \frac{\epsilon}{1 - \epsilon}.
\]

3.5 Sensing Set Identification

In this section, we formulate a new problem where the SU-BS is free to choose any subset of \( S \) as the sensing set and maximizes the expected throughput of the system. We define \( d \) as the homogeneous reporting delay of the sensing results from an SU to the SU-BS, and \( \tilde{d} \) as the miscellaneous delay which covers all the processing required after the collection of sensing results at the SU-BS in \( T_c \). We show that the system throughput is monotonic over the number of SUs chosen in \( S_0 \) in Proposition 3.5.1. When \( N \bar{d} + \tilde{d} \leq T_c \), all SUs can be selected for sensing and it achieves the best performance. However, we are unable to characterize the hardness of the constrained problem with \( N \bar{d} + \tilde{d} \geq T_c \) (not all SUs can be selected), and hence we only provide a heuristic solution to this problem.

3.5.1 Monotonicity of System Throughput

Intuitively, sensing accuracy is increased by adding more SUs into the sensing set. In this section, we confirm this intuition and show that the system throughput is monotonic over the number of SUs in the sensing set. First, we define
\[
P(Y(S_0) = y | B = 0) = \prod_{i \in S_0, y_i = 1} P_i^j \prod_{j \in S_0, y_j = 0} (1 - P_j^j), \tag{3.5.1}
\]
which is the probability of a particular observation vector \( y \) occurring, with \( S_0 \) being the sensing set, given \( B = 0 \), and

\[
P(Y(S_0) = y|B = 1) = \prod_{i \in S_0, y_i = 1} (1 - P^i_m) \prod_{j \in S_0, y_j = 0} P^j_m, \tag{3.5.2}
\]

which is the probability of a particular observation vector \( y \) occurring with \( S_0 \) as the sensing set, given \( B = 1 \). We define

\[
J^*(S_0) = \max_F \left[ (1 - T_c) \pi_0 \sum_{y : F(y) = 0} P(Y(S_0) = y|B = 0) + \gamma \sum_{y : F(y) = 1} P(Y(S_0) = y|B = 1) \right].
\]

**Proposition 3.5.1.** The system throughput is monotonic over the SUs chosen in the sensing set: \( J^*(S'_0) \geq J^*(S_0) \), for all \( S \subseteq S'_0 \).

*Proof.* Given the sensing set \( S'_0 \), we design a decision rule as follows: we always ignore the observations made by SUs in \( S'_0 \setminus S_0 \) and make the optimal decision based on observations of SUs in \( S_0 \). Then we have \( J^*(S_0) \) as the system throughput with sensing set \( S'_0 \) under this rule. Since \( J^*(S'_0) \) is the system throughput with sensing set \( S'_0 \) under the optimal rule, we have \( J^*(S'_0) \geq J^*(S_0) \).

Using Proposition 3.5.1, we know that it is best to choose the full set as the sensing set if \( Nd + \tilde{d} \leq T_c \).

### 3.5.2 Subset Selection

We investigate the case where the number of SUs in \( S_0 \) is constrained. By Proposition 3.5.1, the problem can be formulated as follows:

Problem (C):

\[
\max_{F,S_0} (1 - T_c) \pi_0 \sum_{y : F(y) = 0} P(Y(S_0) = y|B = 0) + \gamma \sum_{y : F(y) = 1} P(Y(S_0) = y|B = 1)
\]
s.t. $|S_0| = k$, 

where $k = \lfloor \frac{T_c - \tilde{d}}{\tilde{d}} \rfloor$.

It has been shown in [56] that no non-exhaustive search method in finding a feature subset of a given size $k$ that has minimal Bayes risk always exists when observations are correlated. Due to the successful mapping between our problem and a Bayesian Decision problem (Problem (3.3.9)), the SU subset selection problem is equivalent to the feature subset problem in [56], except that the observations are assumed to be independent. The hardness of this problem has been a long standing open issue. It is not clear whether exhaustive search would be necessary as shown in [73] with independent observations. Many heuristics such as Sequential Forward Selection (SFS, [64]), Sequential Backward Selection (SBS, [64]) and their variations [41] have been proposed to solve problems of this type. We also propose a simpler greedy heuristic by choosing $k$ SUs with smallest $P_m + P_j$ from set $S$ (Greedy_sum) and compare its performance with that of SFS in Section 3.6. Although we characterize the monotonic property of system throughput over the number of SUs in the sensing set, the complexity of the problem is not clear in the case when $T_c$ is small, compared to $N (Nd + \tilde{d} > T_c)$.

### 3.6 Simulations

In this section, we study the throughput and analyze the sensitivity under Problems (A), (B), and (C) considered earlier. We first compare the performance of the Bayesian decision rule (Algorithm 1), majority, AND and OR policies [81] in Section 3.6.2. Then we present the performance of the greedy algorithm for Problem (B) (Algorithm 2), the random selection and the optimal solution in Section 3.6.3. We compare the performance of Sequential Forward Selection (SFS, [64]) is compared with the optimal solution to Problem (C) in Section 3.6.4 and other heuristics such as
Greedy sum, and the random selection are also compared with SFS in performance. Finally, we conduct sensitivity analyses with inaccurate $P_m^i$, $P_f^i$ or $\pi_0$ information.

### 3.6.1 Simulation Setting

In all of the simulation studies, if not specifically mentioned, our model is that of a cognitive radio network with $N = 20$, $T_c = 0.2$, $\pi_0 = 0.4$, and $\gamma = 2$. We generate 100 groups of practical $P_m^i$ and $P_f^i$ based on randomly generated locations of SUs. In a $50 \times 50$ square area, the locations of the PU are randomly generated and fixed over the simulation. The power level $P$ of the PU is also randomly generated between 1 and 10 and fixed then. In each of the 100 runs, we randomly generate the locations of $N$ SUs within the area and calculate the distance $d(i)$ between the PU and SU $i$. We assume free-space path loss [5] and the SNR at SU $i$ when the PU is transmitting is then calculated as $\frac{P/d(i)^2}{\theta(i)}$, where $\theta(i)$ is the normalized noise at SU $i$ randomly generated between 0.01 and 0.1. The channel gain from the PU to SU $i$ is denoted by $\frac{1}{d(i)^2}$. We let $\lambda(i)$ denote the threshold of the energy detector at SU $i$, which is randomly generated between 0 and 10. We use Equations (3.6.1) and (3.6.2) from [65] to generate 100 groups of $P_m^i$ and $P_f^i$ where the time bandwidth product $u = 3$:

\[
P_m^i = 1 - e^{-\frac{\lambda(i)^2}{2}} \sum_{n=0}^{u-2} \frac{1}{n!} \left( \frac{\lambda(i)}{2} \right)^n - \left( \frac{1 + SNR(i)}{SNR(i)} \right)^{u-1}
\]

\[
x \left[ e^{-\frac{\lambda(i)^2}{2(1 + SNR(i))}} - e^{-\frac{\lambda(i)^2}{2}} \sum_{n=0}^{u-2} \frac{1}{n!} \left( \frac{\lambda(i)SNR(i)}{2(1 + SNR(i))} \right)^n \right]
\]

\[
P_f^i = \frac{\Gamma(u, \frac{\lambda(i)^2}{2})}{\Gamma(u)}
\]

In the equations above, $\Gamma(\cdot, \cdot)$ is the incomplete gamma function, and $\Gamma(\cdot)$ is the gamma function [65]. $SNR(i)$ is the SNR at SU $i$ when the PU is transmitting.
3.6.2 Maximum System Throughput

We have shown in Section 3.3 that Algorithm 1, the Bayesian decision rule based algorithm is optimal. In Figure 3.3, we compare its performance versus majority, AND, OR rules in terms of system throughput, which is the objective function value of Problem (A). When using the majority rule, the decision is 1 only when the majority of the SUs sense an active PU; for the AND rule, the decision is 1 only when all SUs sense an active PU; for the OR rule, the decision is 1 if any of the SUs senses an active PU. We vary $\gamma$, the average PU throughput in Figure 3.3. The Bayesian decision rule strictly outperforms the other algorithms. Among them, the OR and majority rules have similar performance and are both better than AND since the PU transmission is better protected by the OR rule. We show different scenarios where the Bayesian wins over other rules by a small gap in Figure 3.3(a) and by a significant gap in Figures 3.3(b) and 3.3(d). Also, we show the scenarios where OR is always better than AND in Figure 3.3(c) and AND performs better than OR when $\gamma$ is low in Figure 3.3(b). Note that in Figure 3.3(a), the performance of majority is close to that of Bayesian while they are far apart in [44] with randomly generated $P_{m}^i$ and $P_{f}^i$. We observe that $P_{m}^i + P_{f}^i \geq 1$ occurs often there while it never does in the practically generated $P_{m}^i$ and $P_{f}^i$; majority rule over bad SUs ($P_{m}^i + P_{f}^i \geq 1$) leads to unwise decisions. In Figure 3.3(d) we reduce the time bandwidth product to 2 and regenerate 100 groups of $P_{m}^i$ and $P_{f}^i$. We observe a significant number of SUs with $P_{m}^i + P_{f}^i \geq 1$ in each group, which leads to the big gap between Bayesian and majority compared to Figure 3.3(a) when $u = 3$.

3.6.3 Maximum System Throughput under Guaranteed PU Throughput

As shown in Section 3.4, greedy algorithm (Algorithm 2) can achieve throughput strictly greater than $1/2$ of the optimal throughput in Problem (B). We compare it
Figure 3.3: Performance comparison of Bayesian decision rule, majority, AND and OR with $N = 20$, $T_c = 0.2$ and $\pi_0 = 0.4$. 

with its counterpart using the OR rule (Greedy\_OR) and random selection. Greedy\_OR initially assigns observations to $Z = 0$ or $Z = 1$ by OR rule (only observation 0 is assigned to $Z = 0$ in this step); if feasibility is not met, it moves observation 0 to $Z = 1$ as the last chance to satisfy feasibility; if feasibility after the initial step (only observation 0 in $Z = 0$) is met, observations are sorted in $Z = 1$ in decreasing order of $\frac{G(y)}{H(y)}$ and moved to $Z = 0$ until feasibility is violated. Random selection is based on Bayesian decision rule, which means Algorithm 1 is first executed; after
that, observations with $G(y) \geq H(y)$ are randomly selected to put in $Z = 1$ until the feasibility is satisfied. Thus, the main difference between greedy algorithm and random selection lies in the selection criterion of observations with $G(y) \geq H(y)$ after the initial assignment based on Bayesian decision rule.

In addition, we set $r = 2$. We vary parameters such as $\gamma$, the average PU throughput, $\alpha$, the PU throughput constraint, and $N$, the number of SUs in Figure 3.4. Normalized throughput is defined as the system throughput under the algorithm over the optimal solution. Two boundary cases are excluded in the result presentation where both the greedy algorithm and random selection will give the optimal solution: 1) Bayesian decision rule gives the optimal solution; 2) It is optimal to put all observations in $Z = 1$. Hence, we only show their performance when at least one but not all observations with $G(y) \geq H(y)$ have to be moved to $Z = 1$.

In Figure 3.4(a), the normalized throughput of the greedy algorithm, random selection and Greedy.OR are compared for different values of $\gamma$, the average PU throughput in the system. We set $\alpha$ to be $0.8\gamma$ for a fair comparison. With a higher $\gamma$, the factor decreases gradually for all three algorithms. The greedy algorithm, which has a provable lower bound, outperforms the other two algorithms. Potentially, the Bayesian decision rule assigns more SUs to $Z = 1$ compared to a lower $\gamma$ case. Thus, the initial assignment is closer to $\alpha$, the PU throughput constraint. Since we only consider cases where Bayesian decision rule is not optimal, all algorithms tend to have worse performance when the initial assignment approaches $\alpha$ because they get more sensitive to wrong observation selections. Random selection wins over Greedy.OR in that the decision rule still plays an important role in the constrained problem.

In Figure 3.4(b), we vary $\alpha$, the minimum PU throughput constraint, and compare the performance of the algorithms. Again, the greedy algorithm outperforms the other two. The normalized throughput increases with $\alpha$, although it is a minor increase in
Figure 3.4: Performance comparison of greedy algorithm and random selection when $T_c = 0.2$, $\pi_0 = 0.4$ and $r = 2$.

the two greedy algorithms. The increase can be explained similarly as in Figure 3.4(a): a higher $\alpha$ makes the initial assignment farther away from it so that the performance is less sensitive to the choice of observations.

In Figure 3.4(c), we test the performance of our greedy algorithm by varying the number of SUs from 10 to 20. The normalized throughput of all algorithms degrades with more SUs. However, it is always far above $1/2$ for the greedy algorithm, as proved in Theorem 3.4.3. Greedy.OR drops below $1/2$ when $N$ is large as shown in the figure.
Figure 3.5: Performance comparison of SFS over different numbers of SUs in the sensing set when $N = 20$, $T_c = 0.2$, $\pi_0 = 0.4$ and $\gamma = 2$.

### 3.6.4 Maximum System Throughput with Subset Selection

As stated in Section 3.5, the hardness of Problem (C) is unknown. Therefore, here, we focus on the performance of heuristics such as SFS and Greedy\_sum. In SFS, we start from an empty sensing set. At every step, only the SU that is not yet chosen and has the largest marginal increase on the system throughput is added to the set. The algorithm stops when the size of the set reaches $k$. In Figure 3.5, we vary $k$, the size of the sensing set, from 1 to $N$ and show the normalized throughput of SFS. When $k$ increases, the performance of SFS degrades. SFS on average achieves at least 0.8 of the optimal solution achieved by exhaustive search in our simulation although the normalized throughput is lower than 0.65 in one of the worst cases.

In Figure 3.6, we compare the performance of SFS with different rules, Greedy\_sum and random selection and identify the cases where the performance of Greedy\_sum is close to that of SFS or that of random selection. Greedy\_sum chooses $k$ SUs with the smallest $P_m^i + P_f^i$, which is much more efficient than SFS, since no system throughput calculations are needed. Random selection, by its name, randomly chooses $k$ SUs...
for sensing. We set $\gamma = 0.5$ and show the performance difference in Figure 3.6. In Figure 3.6(a), we fix the algorithm to be SFS and apply different decision rules to calculate system throughput at each step. Bayesian wins over other three rules and majority comes next as in Problem (A). AND and OR have poor performance and the system throughput is not even increasing with the number of SUs chosen in the sensing set. In Figure 3.6(b), for $P^i_m + P^i_f$ practically generated, we compare the performance of SFS, Greedy sum and random selection: Greedy sum gets closer to SFS as $k$ increases and both of them have significant increase over random selection. Next, we fix $k = 4$, restrict the range of $P^i_m + P^i_f$ but still keep the diversity of it in Figure 3.6(c). The gap between SFS and Greedy sum gradually decreases as $\beta$ increase, which means the variability in the sum $P^i_m + P^i_f$ for different values of $i$ increases. In Figure 3.6(d), when $\beta$ is small, diversity does not exist and the performance of Greedy sum is close to random selection. Bigger $\beta$ indicates better diversity and the performance of Greedy sum is close to SFS. When $P^i_m + P^i_f$ is not diverse enough, SFS is the best among the three algorithms. However, Greedy sum performs similarly well to SFS in most cases and it has much lower time complexity by avoiding the calculations of system throughput. When diversity of $P^i_m + P^i_f$ in the system is low, Greedy sum gracefully degrades to random selection in performance.

3.6.5 Sensitivity Analysis

So far, we have assumed that parameters such as $P^i_m$ and $P^i_f$ are accurate. However, they are collected using empirical data. Hence, the actual values could be different from those used in the calculations. We investigate the sensitivity of system throughput to these errors for Problems (A), (B), (C) in Figures 3.7, 3.8, and 3.9, respectively. We define $Efficiency$ as the throughput with inaccurate parameters over the throughput with accurate parameters. For sensitivity analysis to $P^i_m$ and $P^i_f$, the value of $P^i_m$
Figure 3.6: Performance comparisons with different rules/ranges of $P_m^i + P_f^i$ for Problem (C) with $\gamma = 0.5$. The restricted ranges show how diversity of SUs affect the performances of Greedy_sum.

used falls in the range of $[P_m^i - \epsilon, P_m^i + \epsilon]$ where $P_m^i$ is the actual value; similarly, the value of $P_f^i$ used falls in the range of $[P_f^i - \epsilon, P_f^i + \epsilon]$ where $P_f^i$ is the actual value. In Figures 3.7(a) and 3.8(a), we compare the performance of three scenarios in terms of the information accuracy of $P_m^i$, $P_f^i$: none of the SUs have accurate information; the one with lowest $P_m^i + P_f^i$ has accurate information; two SUs with lowest $P_m^i + P_f^i$ have accurate information. The efficiency is more than 96% even when the error range $\epsilon$ reaches 0.1 in both figures. As $\epsilon$ increases, the performance improvement increases.
when one more SU with smallest $P_m^i + P_f^i$ gets accurate information. The solution with inaccurate information is always feasible in all the 100 samples. In Figure 3.9(a), the sensitivity analysis is done for Problem (C) by comparing the performances of SFS and Greedy_sum. Greedy_sum is less sensitive to SFS and both of them have the efficiency over 97% as shown in the figure.

The sensitivity analysis of system throughput to $\pi_0$ is even more optimistic as in Figures 3.7(b), 3.8(b) and 3.9(b). The value of $\pi_0$ used falls in the range of $[\pi_0 - \epsilon, \pi_0 + \epsilon]$ where $\pi_0$ is the actual value. The efficiency is always greater than 99% when $\epsilon \leq 0.2$ in all figures, and for Problems (A) and (B), it is especially high. In all 100 samples, we have feasible solutions even with inaccurate information. The sensitivity analysis to $\pi_0$ is done on SFS and Greedy_sum where Greedy_sum is less sensitive to the inaccuracies in $\pi_0$. These results suggest that our solutions for all the problems are robust to inaccuracies in $P_m^i$, $P_f^i$ or $\pi_0$. 
3.7 Summary

In this chapter, we investigate three different problem settings for maximizing the system throughput using cooperative sensing in cognitive radio networks. The first problem we consider is to maximize the weighted sum of the PU and SU throughput in the cognitive radio system. We develop a Bayesian rule based algorithm to find the optimal decision. To guarantee a minimum PU throughput, we study a system throughput maximization problem with PU throughput constraint. We prove that the new problem is strongly NP-hard, and propose a greedy algorithm that achieves an approximation factor strictly greater than $\frac{1}{2}(1 - \epsilon)$ with the time complexity $O(\frac{N^5}{\log^2 \frac{1}{1-\epsilon}})$ where $N$ is the total number of SUs. We also characterize the feasibility gap which tends to 0 when $\epsilon$ goes to 0. Finally, we study the sensing set selection problem under a cardinality constraint. We establish the monotonicity of the system throughput function, i.e., more SUs lead to higher throughput, and propose a simple greedy heuristic to the subset selection problem.
Figure 3.9: Sensitivity analysis of Problem (C) where $\gamma = 0.5$. 

(a) Sensitivity to $P_m^i, P_f^i$. 

(b) Sensitivity to $\pi_0$. 


61
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S$</td>
<td>The set of all SUs in the secondary network</td>
</tr>
<tr>
<td>$M$</td>
<td>Total number of SUs in the secondary network. $</td>
</tr>
<tr>
<td>$S$</td>
<td>Set of SUs which cause interference to PU receivers</td>
</tr>
<tr>
<td>$N$</td>
<td>$</td>
</tr>
<tr>
<td>$S_0$</td>
<td>Set of SUs that are chosen to sense the channel. $S_0 \subseteq S$</td>
</tr>
<tr>
<td>$P_i^f$</td>
<td>False alarm probability of SU $i$</td>
</tr>
<tr>
<td>$P_i^m$</td>
<td>Mis-detection probability of SU $i$</td>
</tr>
<tr>
<td>$P_c^f$</td>
<td>False alarm probability of cooperative sensing</td>
</tr>
<tr>
<td>$P_c^m$</td>
<td>Mis-detection probability of cooperative sensing</td>
</tr>
<tr>
<td>$T_c$</td>
<td>Control slot</td>
</tr>
<tr>
<td>$T_d$</td>
<td>Data slot</td>
</tr>
<tr>
<td>$\pi_0$</td>
<td>Probability that the PU is idle</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>Average throughput of PUs in the interference range of a SU</td>
</tr>
<tr>
<td>$B$</td>
<td>PU activity: 0 is idle and 1 is active</td>
</tr>
<tr>
<td>$Z$</td>
<td>Fusion decision on the PU activity of cooperative sensing</td>
</tr>
<tr>
<td>$F$</td>
<td>Decision rule: ${0, 1}^{</td>
</tr>
</tbody>
</table>
Algorithm 3 Algorithm to Find the Joint Distribution of \( \log \frac{G(y)}{H(y)} \), \( \log H(y) \)

Input: \( N, P^i_j, P^i_m \) for all \( i, r \)

Output: \( C(N, j, j') \) for all \( j, j' \)

1. \( a_i \leftarrow \text{round}(\log \frac{1 - P^i_j}{P^i_m}, r) \times 10^r \) for all \( i \)
2. \( z_i \leftarrow \text{round}(\log \frac{1 - P^i_m}{P^i_j}, r) \times 10^r \) for all \( i \)
3. \( \lambda_i \leftarrow \text{round}(\log P^i_j, r) \times 10^r \) for all \( i \)
4. \( \mu_i \leftarrow \text{round}(\log (1 - P^i_m), r) \times 10^r \) for all \( i \)
5. \( Q \leftarrow \log \frac{(1 - T_e)\pi_0}{\gamma} + \sum_{i=1}^N \max \{a_i, z_i\} \)
6. \( q \leftarrow \log \frac{(1 - T_e)\pi_0}{\gamma} + \sum_{i=1}^N \min \{a_i, z_i\} \)
7. \( Q' \leftarrow \log \gamma + \max \{\max_i \lambda_i, \max_i \mu_i\} \)
8. \( q' \leftarrow \log \gamma + \sum_{i=1}^N \min \{\lambda_i, \mu_i\} \)
9. \( C(i, j, j') \leftarrow 0 \) for all \( i, j, j' \)
10. \( C(1, a_1 + \log \frac{(1 - T_e)\pi_0}{\gamma}, \lambda_1 + \log \gamma) \leftarrow 1 \)
11. \( C(1, z_1 + \log \frac{(1 - T_e)\pi_0}{\gamma}, \mu_1 + \log \gamma) \leftarrow 1 \)
12. \textbf{for} \( i = 1 \) \textbf{to} \( N - 1 \) \textbf{do}
13. \quad \textbf{for} \( j = q \) \textbf{to} \( Q \) \textbf{do}
14. \quad \quad \textbf{for} \( j' = m' \) \textbf{to} \( M' \) \textbf{do}
15. \quad \quad \quad \( C(i + 1, j, j') = C(i, j - y_{i+1}, j' - \lambda_{i+1}) + C(i, j - z_{i+1}, j' - \mu_{i+1}) \)
CHAPTER 4

MAXIMIZING SYSTEM THROUGHPUT WITH SENSING BUDGET CONSTRAINT

In Chapter 3, we investigate a series of system throughput maximization problems using cooperative sensing. However, the problem formulations place no limit on the number of channels that an SU can sense, which is impractical due to hardware and sensing duration constraints. In this chapter, we study the throughput maximization problem for a multi-channel CRN where each SU can only sense a limited number of channels based on the results we have achieved in Chapter 3. We show that this problem is strongly NP-hard, and propose an approximation algorithm with a factor at least $\frac{1}{2} \mu$ where $\mu \in [1, 2]$ is a system parameter reflecting the sensing capability of SUs across channels and their sensing budgets. This performance guarantee is achieved by exploiting a nice structural property of the objective function and constructing a particular matching. Our numerical results demonstrate the advantage of our algorithm compared with both a random and a greedy sensing assignment algorithms.

4.1 Introduction

While cooperative sensing improves sensing accuracy, it also incurs sensing and reporting overhead at the SU side, especially when an SU senses multiple channels in
a multi-channel CRN. In particular, requiring each SU to sense all the channels in a CRN may lead to long sensing durations, especially when the number of channels is large, which in turn reduces the average throughput of SUs. It is therefore reasonable to put a limit on the maximum sensing duration that an SU can afford, which translates to a budget on the number of channels that an SU can sense. Due to the hardware constraints, this budget could be different for different SUs. In this chapter, we study the problem of maximizing the system throughput in a multi-channel CRN, by deciding for each channel, a subset of SUs to sense the channel, subject to the sensing budget constraint at each SU. Our main contributions can be summarized as follows:

- We show that the throughput maximization problem is NP-hard in the strong sense and hence does not have a pseudo-polynomial time algorithm unless P = NP.

- We prove that the system throughput function satisfies a structural property, and based on this we propose a matching-based algorithm, which achieves an approximation factor at least \( \frac{1}{2} \mu \) where \( \mu \in [1, 2] \) is a system parameter depending on the sensing capability of SUs across channels and their sensing budgets.

This chapter is organized as follows. The system model and the problem formulation are introduced in Section 4.2. In Section 4.3, we prove that the optimization problem is NP-hard in the strong sense. We then prove the structural property of the system throughput function, and propose a matching based algorithm in Section 4.4. In Section 4.5, numerical results illustrate the performance of our algorithms. It is summarized in Section 4.6.
4.2 System Model

In this section, we present the system model in two parts: communication model and cooperative sensing model. Based on the models, we formulate our overall objective, which is to decide the channel sensing assignment to maximize the overall system throughput.

4.2.1 Communication Model

We consider a time-slotted cognitive radio network composed of $M$ orthogonal channels (each corresponding to a PU)\(^1\) and $N$ SUs (see Figure 4.1). In the figure, SUs in each circle are capable of sensing the corresponding PU (or channel). An SU may sense multiple PUs depending on its location. For example, SU 3 can sense both channels 1 and 3. When the channel is idle, SUs that do not interfere with each other can

---

\(^1\)Our model can be generalized to the scenario where multiple PUs access the same channel.
transmit over it. Since scheduling and channel assignment for SU transmission are not the focus of this chapter, we employ a simple policy: an SU is randomly selected for transmission over each available channel. Our model can readily be extended to practical models where conflict sets for a given interference model are known. We denote the set of SUs by $S = \{s_1, ..., s_N\}$ with $|S| = N$, and the set of channels by $C = \{c_1, ..., c_M\}$ with $|C| = M$.

### 4.2.2 Cooperative Sensing Model

We assume that a binary decision is made at an SU for each channel it senses. Let $P_f^i(k)$ represent the **probability of false alarm**, i.e., the probability that a SU $s_i$ senses channel $k$ to be occupied when in fact it is idle. Similarly, $P_m^i(k)$ represents the **probability of mis-detection**, i.e., the probability that $s_i$ senses channel $k$ to be idle when it is actually occupied. Note that SUs outside the sensing range, if selected for sensing, report random sensing results. For instance, in Figure 4.1, $P_m^1(3) = \frac{1}{2}$ and $P_f^1(3) = \frac{1}{2}$ since SU 1 is outside the sensing range of PU 3. We assume that these probabilities can be learned using historical data [22, 25, 40]. For instance, given the location information of SUs and hardware parameters such as energy detection threshold and time bandwidth product, etc., $P_m^i(k)$ and $P_f^i(k)$ can be calculated accordingly (see Section 4.5.1 for an example).

**Multi-Channel Cooperative Sensing**: SUs may sense the licensed channels cooperatively to reduce sensing errors. To encourage cooperative sensing, we assume that $\sum_i l_i \geq M$, which is common in cooperative sensing models [82], thus the expected number of SUs that sense a certain channel is at least 1. The sensing results of individual SUs are assumed to be independent. As mentioned earlier, due to practical constraints, SUs can sense a limited number of channels. We denote $l_i$ as the maximum number of channels that SU $s_i$ can sense in a time slot, $0 \leq l_i \leq M$, for
all $i = 1, \cdots, N$ and let $l_{\text{max}} = \max_{i=1}^{N} l_i$. Note that $l_i = 0$ means that the SU is in not in the sensing range of any channel, thus it cannot do any sensing and only guess the PU state randomly. In cooperative sensing under the multi-channel setting, multiple SUs choose to sense different channels and predict channel availability subject to the budget constraint, and different sensing set assignments lead to different system throughput across channels. We consider a centralized system model, where a central controller is responsible for (1) maintaining system parameters for PUs and SUs (2) in each time slot, deciding for each channel, a subset of SUs to sense the channel, and (3) making a global decision on channel availability based on the local binary decisions of SUs. Let $S_k$ denote the set of SUs that cooperatively sense channel $k$. The set of all feasible channel sensing assignment policies are denoted by $\mathcal{P}$, and defined as follows.

**Definition 4.2.1. Feasible assignment policy $\mathcal{P}$:**

A set of sensing sets $\{S_1, \cdots, S_M\}$ is a feasible assignment policy if $\sum_{k=1}^{M} 1_{\{s_i \in S_k\}} \leq l_i$ for all $i$, i.e., all SUs must be assigned to at most $l_i$ channels to sense.

Let $x_i(k)$ denote the observation of channel $k$ by SU $s_i \in S_k$. Further, $x_i(k) = 1$ represents that $s_i$ observes channel $k$ to be active, while $x_i(k) = 0$ represents that $s_i$ observes channel $k$ to be idle. We let $\mathbf{x}(S_k)$ denote the vector of observations for channel $k$. Let $\Omega = \{0, 1\}$, and let $f_A : \Omega^{|A|} \to \Omega$ denote a general decision rule that maps the local observations made by a set of SUs, $A \subseteq S$, to global decision on channel activity. As the domain of $f_A$ will be clear from the context, we drop the subscript and use $f$ instead. This decision rule applies per channel. Let $B(k)$ denote the activity of channel $k$ such that $B(k) = 1$ if channel $k$ is occupied, and $B(k) = 0$ otherwise. According to the definitions of false alarm and mis-detection, we define
the conditional probability of sensing channel $k$ to be idle when it is indeed idle as follows, where vector $\mathbf{y}$ denotes a particular instance of an observation vector:

$$P(f(\mathbf{x}(S_k)) = 0|B(k) = 0)$$

$$= \sum_{\mathbf{y}:f(\mathbf{y})=0} P(\mathbf{x}(S_k) = \mathbf{y}|B(k) = 0),$$

(4.2.1)

where

$$P(\mathbf{x}(S_k) = \mathbf{y}|B(k) = 0)$$

$$= \prod_{y_i=1,s_i\in S_k} P_i^i(k) \prod_{y_j=0,s_j\in S_k} (1 - P_j^j(k)),$$

Similarly, we define the conditional probability of sensing channel $k$ to be occupied when it is indeed occupied:

$$P(f(\mathbf{x}(S_k)) = 1|B(k) = 1)$$

$$= \sum_{\mathbf{y}:f(\mathbf{y})=1} P(\mathbf{x}(S_k) = \mathbf{y}|B(k) = 1),$$

(4.2.2)

where

$$P(\mathbf{x}(S_k) = \mathbf{y}|B(k) = 1)$$

$$= \prod_{y_i=1,s_i\in S_k} (1 - P_m^i(k)) \prod_{y_j=0,s_j\in S_k} P_m^j(k).$$

We assume that in each time slot, a control slot $T_c$ is assigned for cooperative sensing, during which time a central controller collects $P_m^i(k)$ and $P_j^j(k)$ from SUs, determines the channel sensing assignment, collects sensing results from SUs, and notifies an SU per channel to transmit if that channel is cooperatively sensed to be “idle.” Note that each SU $i$ only needs to send updates to the central controller of $P_m^i(k)$, $P_j^j(k)$ when their values change, e.g., when the location of the SU changes. Furthermore, the central controller only needs to compute a new assignment only when $P_m^i(k)$, $P_j^j(k)$ change. We assume $T_c$ to be a constant in the chapter. We further
assume that SUs can transmit at the same bit rate over each channel, and normalize this rate to 1. SUs are assumed to be always backlogged and only one of them is scheduled over channel \( k \) if sensed available in each time slot. Let \( \pi_0(k) \) denote the probability that channel \( k \) is idle, which is assumed to be acquired accurately over time. The capacity of channel \( k \) is denoted by \( \gamma(k) \) (after normalization), \( k = 1, \cdots, M \). We define \( \theta_1(k) = (1 - T_c)\pi_0(k) \) and \( \theta_2(k) = \gamma(k)(1 - \pi_0(k)) \). Following the logic in [44] and extending to the multi-channel case, we define the expected SU throughput over channel \( k \) sensed by \( S_k \).

\[
U_k^1(S_k) := (1 - T_c)P(B(k) = 0, f(x(S_k)) = 0) = \theta_1(k)P(f(x(S_k)) = 0|B(k) = 0) \quad (4.2.3)
\]

if \( S_k \neq \emptyset \);

\[
U_k^1(S_k) := 0 \quad \text{if} \quad S_k = \emptyset .
\]

where we assume that if \( S_k = \emptyset \), no sensing is conducted for channel \( k \) and the channel is never accessed. Likewise, the expected throughput of channel \( k \) can be represented by

\[
U_k^2(S_k) := \theta_2(k)P(f(x(S_k)) = 1|B(k) = 1) \quad (4.2.4)
\]

if \( S_k \neq \emptyset \);

\[
U_k^2(S_k) := \theta_2(k) \quad \text{if} \quad S_k = \emptyset .
\]

**Definition 4.2.2. System throughput:** For a channel assignment \( \{S_1, \cdots, S_M\} \), we define the throughput over channel \( k \) to be the sum of SU and PU throughput over channel \( k \), denoted as \( U_k(S_k) = U_k^1(S_k) + U_k^2(S_k) \). The system throughput is defined as

\[
\sum_{k=1}^{M} U_k(S_k).
\]

Note that for a given channel sensing assignment, the achievable system throughput is determined by the decision rule \( f \). In this chapter, we apply the optimal
Bayesian decision rule proposed in [44] to each channel respectively, to obtain the optimal expected system throughput. Formally, for each channel $k$ and an observation vector $y$ by $S_k$, if $\theta_2(k)P(x(S_k) = y|B(k) = 1) \geq \theta_1(k)P(x(S_k) = y|B(k) = 0)$, the decision on channel $k$ is “occupied”, and the contribution to throughput is $\theta_2(k)P(x(S_k) = y|B(k) = 1)$; otherwise, the decision on channel $k$ is “idle” and the contribution is $\theta_1(k)P(x(S_k) = y|B(k) = 0)$.

4.2.3 Problem Formulation

We formulate the optimization problem to maximize the system throughput, including PUs and SUs on all channels, as follows:

Problem (D): $\max_{\{S_1,\ldots,S_M\} \in \mathcal{P}} \sum_{k=1}^{M} U_k(S_k),$

where the Bayesian decision rule is implicit in the definition of $U_k(\cdot)$.

Our goal is to decide the optimal channel sensing assignment to maximize system throughput. We adopt a common assumption that PUs can tolerate interference to a certain extent, which may appear in the form of a constraint as in [22, 55] and our earlier paper [44] for the single channel setting. In the future, we plan to extend our solution presented in this paper to Problem (D) with explicit constraints on PU throughput.

We assume that the system is static and the optimization is done in a single time slot. Note that the solution of the static assignment would apply to multiple time slots if $P^i_m(k)$ and $P^i_f(k)$ do not change over time, or if changes occur over a much slower time scale.

4.3 Hardness of the Problem

In this section, we will show that Problem (D) is strongly NP-hard [74], by a reduction from Product Partition, which is NP-complete in the strong sense [9]. The Production
Partition problem is defined as follows: Given $N$ positive integers $a_1, a_2, \cdots, a_N$, is there a subset $X \subseteq \mathcal{N} := \{1, 2, \cdots, N\}$ such that $\prod_{i \in X} a_i = \prod_{i \in \mathcal{N} \setminus X} a_i$?

We reduce Product Partition to the following subproblem of Problem (D), with $M = 2$, $P^j_i(1) = P^j_i(2) = 0$ for all $i$, $P^i_m(1) = P^i_m(2) := P^i_m$ for all $i$, and $l_i = 1$ for all $i$, $\gamma(1) = \gamma(2) := \gamma$, $\pi_0(1) = \pi_0(2) := \pi_0$, $(1 - T_c)\pi_0 := \theta_1$, $\gamma(1 - \pi_0) := \theta_2$, and $\theta_1 = \theta_2$.

Let $(S_1, S_2)$ denote a solution to this subproblem. Without loss of optimality, we can assume $S_1$ and $S_2$ form a partition of the set of SUs, i.e., $S_1 \cup S_2 = S$ and $S_1 \cap S_2 = \emptyset$. The expected system throughput can then be easily determined using the Bayesian rule as $U_1(S_1) = \theta_1 + \theta_2(1 - \prod_{s_i \in S_1} P^i_m)$ and $U_2(S_2) = \theta_1 + \theta_2(1 - \prod_{s_i \in S_2} P^i_m)$. Problem (D) then becomes: $\max_{S_1 \subseteq S} \left[ 2\theta_1 + \theta_2(2 - (\prod_{s_i \in S_1} P^i_m + \prod_{s_i \in S \setminus S_1} P^i_m)) \right]$, which is further equivalent to $\min_{S_1 \subseteq S} \left( \prod_{s_i \in S_1} P^i_m + \prod_{s_i \in S \setminus S_1} P^i_m \right)$ since $2\theta_1 + 2\theta_2$ is a constant. We then establish the strong NP-hardness of Problem (D) by showing that this new problem is strongly NP-hard.

**Proposition 4.3.1.** Problem (D) is strongly NP-hard.

**Proof.** By the above argument, it suffices to prove that the subproblem, the following problem is strongly NP-hard.

$$\min_{S_1 \subseteq S} \left( \prod_{s_i \in S_1} P^i_m + \prod_{s_i \in S \setminus S_1} P^i_m \right)$$

Given an instance of Product Partition with parameters $a_1, \cdots, a_N$, we reduce it to an instance of this subproblem as follows: let $P^i_m = a_i / 10^r$, $i = 1, \cdots, N$, where $r$ is the smallest integer such that $P^i_m \leq 1$ for all $i = 1, ..., N$. This reduction can clearly be done in polynomial time. Furthermore, if there is a subset $X \subseteq \mathcal{N}$, such that $\prod_{i \in X} a_i = \prod_{i \in \mathcal{N} \setminus X} a_i = \sqrt[|\mathcal{N}|]{\prod_{i \in \mathcal{N}} a_i}$, then the optimal solution to the subproblem
is $2\sqrt{\prod_{s_i \in S} P^i_m}$, and vice-versa. Hence if there is polynomial time algorithm to the subproblem, the Product Partition problem can be determined in polynomial time as well, which contradicts the fact that Product Partition is strongly NP-complete.

Since Problem (D) is strongly NP-hard, no pseudo-polynomial time algorithms exist unless P = NP [74]. We will propose a matching-based approximation algorithm that has theoretical lower bound in Section 4.4.

4.4 Approximate Solutions

In this section, we propose an efficient approximation algorithm for Problem (D). We first investigate the random case where SUs are randomly assigned to channels for sensing subject to the budget constraint, and show the worst case performance ratio. We then propose a matching-based approximation algorithm. By exploiting the ratio of it over the worst case, we show a ratio at least $\frac{1}{2}\mu$ where $\mu \in [1, 2]$ is a system parameter and will be defined later.

4.4.1 Property of the System Throughput

We will show the range of the system throughput $U_k(\cdot)$ in the following lemma.

Lemma 4.4.1. For any SU $s_i$ and channel $c_k$, we have $\theta_1(k) + \theta_2(k) \geq U_k(s_i) \geq \max\{\theta_1(k), \theta_2(k)\}$.

Proof.

\[
U_k(\{s_i\}) = \max\left\{\theta_1(k)(1 - P^i_f(k)), \theta_2(k)P^i_m(k)\right\} \\
+ \max\left\{\theta_1(k)P^i_f(k), \theta_2(k)(1 - P^i_m(k))\right\} \\
\geq \max\left\{\theta_1(k), \theta_2(k)\right\}
\]
It is obvious from the form of $U_k(\cdot)$ that $\theta_1(k) + \theta_2(k) \geq U_k(s_i)$. It means that at most both PU and SU can achieve their full capacity.

\[\Box\]

### 4.4.2 A Matching-Based Approximation Algorithm

In this section, we propose a maximum weighted matching (MWM) [77] based algorithm to Problem (D). We first provided a detailed description of our algorithm (see Algorithm 4), and then establish its approximation factor.

The algorithm starts with constructing a complete and weighted bipartite graph (lines 2-4), where for each channel $k$, a vertex $c_k$ is constructed, and for each SU $s_i$, $l_i$ vertices are constructed corresponding to the $l_i$ copies of the SU, denoted as $s^j_i$, $j = 1, \ldots, l_i$, and for any pair of vertices $s^j_i$ and $c_k$, there is an edge connecting them. The weight of an edge $(s^j_i, c_k)$ is then defined as $w(s^j_i, c_k) = U_k(\{s_i\})$ (line 5).

A maximum weight matching in the bipartite graph is then found (line 6), and for each edge $(s^j_i, c_k)$ in the matching, SU $s_i$ is assigned to sense channel $c_k$. A greedy heuristic is applied for determining the assignment of the remaining copies of SUs to channels (lines 8-12). Basically, the remaining copies are first sorted in an arbitrary order, and a copy of $s_i$ is assigned to the channel that provides the maximum marginal improvement of the system throughput among all the channels not assigned to $s_i$ yet. This scheme is then compared with another scheme for which all SUs are assigned to a single channel that gives maximum throughput (line 13). The algorithm outputs whichever scheme provides a larger system throughput.

We then analyze the complexity of Algorithm 4, which is dominated by computing the maximum weighted matching and evaluating the throughput function $U_k(\cdot)$. It is shown in [44] that for a given sensing set $S_k$, $U_k(S_k)$ can be evaluated using a dynamic programming algorithm in pseudo-polynomial time. Let $Q$ denote the time
complexity for one evaluation of $U_k(\cdot)$. Note that the total number of such evaluations is bounded by $N_{l_{\text{max}}}M$. Therefore, the time complexity of Algorithm 4 is $O(N_{l_{\text{max}}}MQ + (N_{l_{\text{max}}} + M)^3)$.

The MWM in Algorithm 4 captures the system heterogeneity that includes: 1) SUs have different sensing abilities for each channel; 2) channels are competing for SUs due to their sensing budget. Before we show the approximation ratio of Algorithm 4, we construct a matching and show its lower bound performance. The matching is constructed as follows: 1) Partition the channel set $C$ into groups indexed by SU, and each group is labeled as $C_i$ that includes all channels $k$ with $U_k(\{s_i\}) \geq U_k(\{s_j\})$ where $j \neq i$. Ties are randomly broken. The size of $C_i$ is $r_i$. 2) Sort the channels $k$ in each group $C_i$ by $U_0^k$ in descending order, where $U_0^k = \min_i U_k(\{s_i\})$. 3) Pick the first $l_i$ channels from each group $C_i$ (the set is labeled as $C_{li}$) and assign SU $i$ to sense these channels. 4) Randomly assign an SU copy to each of the rest channels. We will next show lower bound of the performance of $M_{Gdy}$ in Lemma 4.4.2. We define $\lambda_i = \min\{l_i, r_i\}/r_i$, $\rho_i = \min_{k \in C_{li}} \frac{U^*_k}{U_k^0}$ where $U^*_k = \max_i U_k(\{s_i\})$, $\mu = 1 + \sum_i \lambda_i(\rho_i - 1)$ and $|M_{Gdy}|$ as the system throughput with sensing assignments in $M_{Gdy}$. Note that $\rho_i \in [1,2]$ for all $i$.

**Lemma 4.4.2.** $M_{Gdy}$ achieves at least $\mu \sum_k U_k^0$. 
Proof.

\[
\frac{|M_{Gdy}|}{\sum_k U_k^0} \geq \frac{\sum_{i \in S} \left( \sum_{k \in C_i^j} U_k^* + \sum_{k \in C_i \setminus C_i^j} U_k^0 \right)}{\sum_k U_k^0} \\
\geq \frac{\sum_{i \in S} \left( \sum_{k \in C_i^j} U_k^0 \rho_i + \sum_{k \in C_i \setminus C_i^j} U_k^0 \right)}{\sum_{i \in S} \sum_{k \in C_i} U_k^0} \\
= 1 + \frac{\sum_{i \in S} \left( \rho_i - 1 \right) \sum_{k \in C_i^j} U_k^0}{\sum_{i \in S} \sum_{k \in C_i} U_k^0} \\
\geq 1 + \frac{\sum_{i \in S} \left( \rho_i - 1 \right) \lambda_i \sum_{k \in C_i} U_k^0}{\sum_{i \in S} \sum_{k \in C_i} U_k^0} \\
\geq 1 + \min_{i \in S} \lambda_i (\rho_i - 1) \\
= \mu
\]

Based on Lemmas 4.4.1 and 4.4.2, we show the approximation ratio of Algorithm 4 in Proposition 4.4.3.

**Proposition 4.4.3.** Algorithm 4 achieves at least a fraction of \(\frac{1}{2} \mu\) of the optimal system throughput for Problem (D).

**Proof.** Let OPT be the optimal solution, and ALG be the solution by Algorithm 4 to Problem (D). By Lemma 4.4.1, we know that

\[
\frac{\sum_k U_k^0}{OPT} \geq \frac{\sum_k \max\{\theta_1(k), \theta_2(k)\}}{\sum_k \theta_1(k) + \theta_2(k)} \geq \frac{1}{2} \tag{4.4.1}
\]

Since ALG is an outcome at least as good as maximum weight matching and M_{Gdy} is a matching we construct in a greedy day, we have \(ALG \geq |M_{Gdy}|\). By Lemma 4.4.2, we can achieve

\[
\frac{ALG}{OPT} \geq \frac{1}{2} \mu. \tag{4.4.2}
\]
Remark 1: We note that when $\theta_1(k) \gg \theta_2(k)$ or $\theta_2(k) \gg \theta_1(k)$, we can achieve a solution close to the optimal by Algorithm 4 since Equation (4.4.1) becomes close to 1. Only when $\theta_1(k)$ and $\theta_2(k)$ for all $k$ are close, Equation (4.4.1) is only right above $\frac{1}{2}$. Also, if SU’s sensing abilities across channels vary in a large range, or the sensing budgets of SUs are large, by Equation (4.4.2), the ratio will be close to 1 since $\rho_i$, $\lambda_i$ will be large, respectively.

Remark 2: In the proof of Proposition 4.4.3, we have ignored the greedy heuristic applied to the copies of SUs not included in the matching. Hence the result established above only provides a lower bound on the performance of our algorithm. Proving a tighter bound for the algorithm that incorporates the greedy heuristic is part of our future work.

4.5 Simulations

In this section, we study the performance of our algorithm through simulations by comparing Algorithm 4 (MWM) with a random sensing assignment algorithm, and a greedy algorithm (defined next). In the random algorithm, the copies of SUs are randomly assigned to PUs. The greedy algorithm works as follows: for each PU $k$, the set of SUs are first sorted by $P^i_m(k) + P^i_f(k)$ in a non-decreasing order as its preference list. In each round, a random permutation of the set of PUs is applied. The algorithm then goes through the PU list, and for each PU $k$, a copy of the SU, say $s_i$, with the lowest $P^i_m(k) + P^i_f(k)$ among the remaining SUs, which has not been assigned to $k$ before and has remaining copies, is assigned to $k$. Repeat this procedure till all copies of SUs have been assigned.
4.5.1 Simulation Setting

The following parameters are fixed throughout the simulations. We consider a 100 × 100 area, where the locations of \( M \) PUs are randomly generated. For each PU \( k \), its maximum power level is randomly chosen between 1 and 10, and \( \pi_0(k) \) are randomly generated in \([0, 1]\). We also set \( T_c = 0.2 \) fixed.

In each of the 100 runs of the simulation, the following parameters are varied independently. First, the channel status of PU \( k \), either transmitting with the maximum power or idle, is randomly chosen according to \( \pi_0(k) \). The locations of \( N \) SUs (each SU is capable of sensing any of the \( M \) channels) are then randomly generated. Let \( d(k,i) (k = 1, \cdots, M; i = 1, \cdots, N) \) denote the distance between the \( k \)-th PU and the \( i \)-th SU. We then apply the model proposed in \([65]\) to generate \( P_m^i(k) \) and \( P_f^i(k) \) as follows. First, we calculate the SNR at SU \( i \) when PU \( k \) is transmitting as \( \frac{P(k)/d(k,i)^2}{\theta(i)} \), where \( \theta(i) \) is the normalized noise at SU \( i \) randomly generated between 0.01 and 0.1. The channel gain from PU \( k \) to SU \( i \) is given by \( \frac{1}{d(k,i)^2} \). The threshold of the energy detector at SU \( i \), denoted as \( \lambda(i) \), is randomly generated between 0 and 10. We then apply Equations (4.5.1) and (4.5.2) \([65]\) to generate \( P_m^i(k) \) and \( P_f^i(k) \), where the time bandwidth product \( u \) is set to 3:

\[
P_m^i(k) = 1 - e^{-\frac{\lambda(i)}{2}} \sum_{n=0}^{u-2} \frac{1}{n!} \left( \frac{\lambda(i)}{2} \right)^n - \left( \frac{1 + SNR(k,i)}{SNR(k,i)} \right)^{u-1} \times \left[ e^{-\frac{\lambda(i)}{2}} \sum_{n=0}^{u-2} \frac{1}{n!} \left( \frac{\lambda(i)SNR(k,i)}{2(1 + SNR(k,i))} \right)^n \right]
\]

\[
P_f^i(k) = \frac{\Gamma(u, \frac{\lambda(i)}{2})}{\Gamma(u)},
\]

where \( \Gamma(\cdot, \cdot) \) is the incomplete gamma function, \( \Gamma(\cdot) \) is the gamma function \([65]\) and
4.5.2 Simulation Results

The simulation results are shown in Figure 4.2. Note that we do not restrict \( \sum_i l_i \geq M \) in our simulations. If PU \( k \) is not assigned any SU for sensing (\( S_k = \emptyset \)), the system throughput on channel \( k \) is \( \theta_2(k) \) (Definition 4.2.2). In all the figures, we plot \( \sum_k \left[ \theta_1(k) + \theta_2(k) \right] \) as the upper bound for the optimal solution.

In Figure 4.2(a), we fix \( M = 20, l_{\max} = 3 \), and vary \( N \) from 4 to 20. For each PU \( k \), \( \gamma(k) \) in generated randomly in \([1,3]\) and then fixed over all 100 runs. We
choose this range since the average PU throughput is usually larger than the unit SU throughput. For each SU $i$, $l_i$ is randomly generated between 1 and $l_{\text{max}}$ and fixed over all the runs. The simulations results are averaged over all 100 runs. We observe that Algorithm 4 achieves significant improvement over the random and the greedy algorithms for all $N$, although the gap shrinks as $N$ increases. For instance, the system throughput of Algorithm 4 is 24% larger than that of the greedy algorithm when $N = 4$ and it decreases to 16% when $N = 20$. When more SUs join the network, the random and the greedy algorithms have more chance to choose “good” SUs. The greedy algorithm is comparable to the random algorithm when $N$ is small. However, it wins over the latter when $N \geq 12$. This indicates that the sorting step in the greedy algorithm helps PUs pick the “right” SUs, which is more useful when $N$ is large. Besides, the performance of Algorithm 4 reaches 95% of the upper bound of the optimal solution when $N = 20$.

In Figure 4.2(b), $M, N$ are fixed to be 20 and 8, respectively, and we vary $l_{\text{max}}$ from 1 to 5. $\gamma(k)$ is again generated randomly in $[1, 3]$ and fixed over all 100 runs. Similar to Figure 4.2(a), Algorithm 4 outperforms both the random and the greedy algorithms, and the greedy algorithm outperforms the random algorithm when $l_{\text{max}} \geq 3$. When $l_{\text{max}} = 4$, the system throughput of Algorithm 4 is 25% better than that of the greedy algorithm, which is the largest gap in the figure. An interesting observation is that the expected number of SU copies when $l_{\text{max}} = 4$ ($N = 8$) is equal to $M = 20$, thus every PU is assigned an SU on average. When there is more supply (SUs) than demand (PUs) or more demand than supply, the performance gap between Algorithm 4 and greedy algorithm, random algorithm is not so significant.

In Figure 4.2(c), we fix $M = 20$, $N = 8$, $l_{\text{max}} = 3$, and vary the range of the channel capacity $\gamma(k)$. For instance, $[1, 2]$ means all channel capacities are randomly generated between 1 and 2. Algorithm 4 is constantly better than the other two
algorithms. The gap first increases as the channel capacity increases (from 18% to 34%) till \(\gamma(k) \in [1, 2]\), and decreases thereafter (7% at \(\gamma(k) \in [1, 5]\)). When the channel capacity is comparable to unit SU capacity, the choice of SUs for sensing does not affect the system throughput significantly; When the channel capacity dominates the system throughput, the choice of SUs again loses its leading role. Thus the largest gap appears in the middle.

4.6 Summary

In this chapter, we investigate the problem of throughput maximization using cooperative sensing in multi-channel CRNs, where each SU can only sense a limited number of channels with various sensing capabilities, due to time or energy constraints. We show that under the optimal Bayesian decision rule, the channel sensing assignment problem is strongly NP-hard. A matching based algorithm is then proposed with an approximation ratio that is at least \(\frac{1}{2}\mu\) where \(\mu \in [1, 2]\) is a system parameter. Our numerical results demonstrate that our algorithm performs significantly better than the a random channel sensing assignment algorithm and a greedy algorithm.
Algorithm 4 A maximum weighted matching based algorithm for maximizing the system throughput across channels

Input: $N, M, T_c, \pi_0(k), \gamma(k)$ for all $k$; $l_i$ for all $i$; $P^i_m(k), P^i_f(k)$ for all $i$ and $k$

Output: $U$ and $S_k$ for all $k$

1: $S_k \leftarrow \emptyset$ for all $k$
2: $V \leftarrow \{s^1_1, \ldots, s^l_i, \ldots, s^l_N\} \cup \{c_1, \ldots, c_M\}$
3: $E \leftarrow \bigcup_{i=1, \ldots, N; \ k=1, \ldots, M} \bigcup_{j=1} \{(s^j_i, c_k)\}$
4: $G \leftarrow (V, E)$
5: $w(s^j_i, c_k) \leftarrow U_k(\{s_i\})$, $\forall i = 1, \ldots, N$, $j = 1, \ldots, l_i$, $k = 1, \ldots, M$
6: $\mathcal{M} \leftarrow$ a maximum weight matching in $G$
7: $S_k \leftarrow \{s_i : (s^j_i, c_k) \in \mathcal{M}\}$, $\forall k$
8: $R \leftarrow \{s^j_i : s^j_i$ is not matched in $\mathcal{M}\}$
9: for all $s^j_i \in R$ do
10: $k^* \leftarrow \arg \max_{k \in \{1, \ldots, M\}, s_i \not\in S_k} \left[ U_k(S_k \cup \{s_i\}) - U_k(S_k) \right]$
11: $S_{k^*} \leftarrow S_{k^*} \cup \{s_i\}$
12: $U \leftarrow \sum_{k=1}^M U_k(S_k)$
13: $U_1 \leftarrow \max_{k=1}^M U_k(S)$
14: if $U_1 > U$ then
15: $U \leftarrow U_1$
16: $k^* \leftarrow \arg \max_{k=1}^M U_k(S)$
17: $S_{k^*} \leftarrow S, S_k \leftarrow \emptyset \ \forall k \neq k^*$
CHAPTER 5
SOCIAL WELFARE MAXIMIZATION IN
OPERATOR-BASED CRNS

In this chapter, we explore the operator based model in CRNs and shift our attention from throughput to social welfare which reflects how much each SU values the spectrum. We study the spectrum allocation problem under spectrum uncertainty and sensing inaccuracy. Most existing works apply the framework of spectrum auctions assuming that the operator has perfect knowledge of PU activities. In practice, however, it is more likely that the operator only has statistical information of the PU traffic when trading a spectrum hole, and it is acquiring more accurate information over time. The operator may acquire channels from PUs out of its control and these channels have to be sensed before SUs use them for transmission. We first model the problem as a finite horizon Markov decision process when the operator knows all spectrum requests in advance, and propose an optimal dynamic programming based algorithm. We then investigate the case when spectrum requests are submitted online, and propose a greedy algorithm that is 1/2-competitive for homogeneous channels and is comparable to the offline algorithm for more general settings. We further extend the online algorithm to an online auction scheme.
5.1 Introduction

We propose a spectrum allocation framework that takes both spectrum uncertainty and sensing inaccuracy into account. In particular, we consider two types of spectrum resources: PU channels that are under the control of the operator, and the channels that the operator acquires from PUs out of its control. In practice, wireless service providers (WSP) act as operators, and they may cover areas that almost completely overlap. SUs registered with one of them may access spectrum from other WSPs as will be introduced in our model. In both types of channels, PU traffic on each channel is assumed to follow a known i.i.d. Bernoulli distribution. For the first type of channels, the real-time channel state can be learned accurately by the operator. For the second type of channels, a sense-before-use paradigm must be followed, where a collision with the PU traffic due to sensing inaccuracy incurs a penalty.

Using a fixed set of channels of each type, we study the joint spectrum sensing and allocation problem to serve spectrum requests with arbitrary valuations and arbitrary levels of delay tolerance. The objective of the operator is to maximize social welfare, i.e., the total valuations obtained from successfully served requests minus the cost due to collisions. We consider both the scenario where all spectrum requests are known in advance, and the setting when spectrum requests are submitted online. While our online setting is similar to the online spectrum auction schemes considered in [17, 79], the key difference is that sensing inaccuracy is not considered in these existing works. Hence, the approaches in [17, 79] can only be applied to cases where accurate real-time channel states are obtainable, which is not always the case.

Our contributions can be summarized as follows:

- We model the joint sensing and spectrum allocation problem as a finite horizon Markov decision process when all spectrum requests are revealed to the operator.
offline, i.e., ahead of time. We develop an optimal dynamic programming based algorithm, which serves as a baseline for the achievable social welfare.

- We propose a greedy algorithm for the case when spectrum requests are submitted online. We prove that the online algorithm is $1/2$-competitive for homogeneous channels, and we show that it achieves performance comparable to the offline algorithm for the more general heterogenous channel case by numerical results.

- We further extend the online algorithm by proposing an online auction scheme, which ensures incentive compatibility for SUs and also provides a way for trading off social welfare and revenue using a reservation price.

The chapter is organized as follows: The system model and problem formulation are introduced in Section 5.2. Our solutions to the problem with offline and online requests are presented in Sections 5.3 and 5.4, respectively. Our online auction scheme is then discussed in Section 5.5. In Section 5.6, numerical results are presented to illustrate the performance of the greedy online algorithm in general cases, and the tradeoff between social welfare and revenue. We summarize the chapter in Section 5.7.

5.2 System Model and Problem Formulation

We consider a cognitive radio network with a single operator and multiple SUs registered with it (see Figure 5.1). The operator manages multiple orthogonal channels and controls the corresponding network composed of PUs. We focus on downlink transmission at the operator with power control. A time slotted system is considered with all PU and SU transmissions synchronized. All SUs are assumed to be in the interference range of each other and that of PUs, hence, each channel can be assigned to at most one SU at any time when it is not used by PUs. In this chapter, we focus
on the temporal reuse of spectrum, and we will consider spatial reuse [17], [38], [79] to further improve allocation efficiency in the future.

The spectrum pool consists of two types of channels, those managed by the operator and those that are not. The operator is aware of the downlink activity of its own PUs at the beginning of each time slot. The set of the spectrum bands\(^1\) managed by the operator is denoted by \(T_1\). However, the activities of PUs not managed by the operator are unknown. Bands accessed by these PUs are denoted by \(T_2\). To access bands in \(T_2\), SUs cooperatively sense them and report their sensing results to the operator. The operator then makes a fusion decision on the activities of bands in \(T_2\) and selects a subset of channels sensed idle to serve the SUs. We only consider the set of PUs located in the coverage area of the operator so that all SUs in the system have the cognitive capability and can sense spectrum in \(T_2\). We assume that the sensing cost is low and even negligible. In practice, wireless service providers (WSP) act as operators, and they may cover areas that almost overlap. SUs registered with one of them may access spectrum from other WSPs as introduced in our model.

We assume that the spectrum bands in \(T_1\) and \(T_2\) have the same capacity, which is normalized to 1. PU activities on these channels follow an i.i.d. Bernoulli distribution in each time slot. For instance, in Figure 5.1, there are three channels in \(T_1\) and two channels in \(T_2\). In time slot 1, channels 2 and 3 in \(T_1\) are idle and channel 1 in \(T_2\) is idle. However, channel 1 in \(T_2\) is sensed busy and it will not be allocated. Also, channel 2 in \(T_2\) is incorrectly sensed to be idle and scheduling a request on this channel will lead to a collision. We let \(\pi_1(i)\) denote the probability that channel \(i\) in \(T_1\) is idle and \(\pi_2(j)\) the probability that channel \(j\) in \(T_2\) is idle. We also assume that the prior distribution of the PU activity is accurately acquired over time. We assume

\(^1\)We use channel and spectrum band interchangeably.
that state changes occur at the beginning of a time slot. Let $C \triangleq |T_1| + |T_2|$ denote the total number of channels, which remains constant over time.

The availabilities of channels in $T_1$ and $T_2$ at $t$ are denoted by binary vectors $\vec{I}_1(t) = (I_1^1(t), \cdots, I_1^k(t), \cdots)$ and $\vec{I}_2(t) = (I_2^1(t), \cdots, I_2^l(t), \cdots)$, respectively, where 0 represents idle and 1 represents busy states. Moreover, $\vec{I}_2^s(t)$ denotes the sensed availabilities of channels in $T_2$ at $t$. Let $P_f(k)$, $k \in T_2$, denote the probability of false alarm for channel $k$, i.e., the probability that SUs cooperatively sense channel $k$ to be busy given that it is actually idle. Let $P_m(k)$ represent the probability of misdetection for channel $k$, i.e., the probability that SUs cooperatively sense channel $k$ to be idle given that it is actually busy. Our problem formulation and solutions are independent of the cooperative sensing scheme used. We further define $P_l(k)$ as the probability that channel $k$ is sensed idle and $P_0(k)$ as the conditional probability of channel $k$ being idle given that it is sensed idle. Note that $P_l(k)$ =
\(\pi_2(k)(1 - P_f(k)) + (1 - \pi_2(k))P_m(k)\) and \(P_0(k) = \frac{\pi_2(k)(1 - P_f(k))}{P_f(k)}\). We assume that \(P_f(k)\) and \(P_m(k)\) are constant for any channel \(k \in T_2\), which occurs e.g. when SUs are static in the system. Some of our technical results apply to the special case when all channels in \(T_2\) are *homogenous*, that is, when the channels have the same \(\pi_2(i)\), \(P_m(i)\) and \(P_f(i)\). Thus, they also have the same \(P_0(i)\) and \(P_f(i)\).

We assume each spectrum request is for a single time-frequency chunk, i.e., a single time slot of any channel in \(T_1\) or \(T_2\). Each request \(i\) submitted at time \(t\) is of the form \((a_i, d_i, w_i)\), where \(a_i \geq t\) is the required service starting time, \(d_i\) is the deadline, at the beginning of which request \(i\) leaves the system, and \(w_i\) is the valuation of request \(i\), which will be added to the social welfare if request \(i\) is served by \(d_i\). We denote the set of requests by \(\mathcal{N} = \{1, \cdots, N\}\). \(H = \max_{i \in \mathcal{N}} d_i - \min_{i \in \mathcal{N}} a_i\) denotes the time period spectrum allocation needs to be made, and \(\min_{i \in \mathcal{N}} a_i\) is normalized to 1. The maximum number of outstanding requests in the system at any time is denoted as \(r\). Table 5.1 summarizes the notations used in the chapter.

We are interested in maximizing the social welfare of the operator and the SUs in the system, which is defined as the total valuations from the requests successfully served (by their deadlines) minus the collision cost to channels in \(T_2\). Let \(Q\) denote the penalty incurred per collision. Let \(x_{il}(t)\) \((i \in \mathcal{N}, l \in T_1 \cup T_2, t = 1, \cdots, H)\) denote the allocation indicator: \(x_{il}(t) = 1\) if request \(i\) is allocated to channel \(l\) at \(t\); \(x_{il}(t) = 0\) otherwise. Let \(y_i\) denote the service indicator: \(y_i = 1\) if request \(i\) is served by \(d_i\); \(y_i = 0\) otherwise. The social welfare maximization problem is then formulated as follows, where \(Z(\cdot)\) denotes the number of 0 elements in a vector:

**Problem (A):**

\[
\max_{x, y, \mathbf{I}_1, \mathbf{I}_2} \sum_{i \in \mathcal{N}} y_i w_i - Q \sum_{i \in \mathcal{N}} \sum_{l \in T_2} \sum_{t=1}^H x_{il}(t) I^2_l(t)
\]

88
\[
\text{s.t. } \sum_{t=a_i}^{d_i} \left( \sum_{l \in T_1} x_{il}(t) + \sum_{k \in T_2} x_{ik}(t)(1 - I_2^k(t)) \right) \geq y_i, \quad (5.2.1)
\]
for all \( i \in \mathcal{N} \)

\[
\sum_{i \in \mathcal{N}} \sum_{t \in T_1} x_{it}(t) \leq Z(\vec{I}_1(t)), \text{ for all } t = 1, \cdots, H \quad (5.2.2)
\]

\[
\sum_{i \in \mathcal{N}} \sum_{k \in T_2} x_{ik}(t) \leq Z(\vec{I}_2^s(t)), \text{ for all } t = 1, \cdots, H, \quad (5.2.3)
\]

where \( \mathbf{x} = (x_{it}(t))_{i,t}, \mathbf{y} = (y_i)_{i \in \mathcal{N}}, \mathbf{I}_1 = (\vec{I}_1(t))_{t=1,\cdots,H}, \mathbf{I}_2 = (\vec{I}_2(t))_{t=1,\cdots,H}, \mathbf{I}_2^s = (\vec{I}_2^s(t))_{t=1,\cdots,H} \). The cost \( Q \sum_{i \in \mathcal{N}} \sum_{t \in T_2} x_{it}(t)I_2^t(t) \) takes into account the current availabilities of channels in \( T_2 \). Inequality (5.2.1) reflects the relationship between the allocation indicator \( x_{it}(t) \) and the service indicator \( y_i \). Inequality (5.2.2) guarantees that a channel in \( T_1 \) will not be allocated unless it is observed idle. Likewise, Inequality (5.2.3) guarantees that a channel in \( T_2 \) will not be allocated unless it is sensed idle.

The challenges of Problem (A) are threefold: 1) The requests are uncertain since they may be submitted at different time slots; 2) Spectrum availabilities of \( T_1 \) and \( T_2 \) in the future are not known at the current time slot; 3) Sensing is not accurate for channels in \( T_2 \). In the following, we propose an offline optimal solution in Section 5.3 and an online solution in Section 5.4. We define the offline algorithm as an algorithm that decides the channel allocation for outstanding requests in each time slot with only the observed availabilities of channels in \( T_1 \) and sensed availabilities in \( T_2 \) of the current slot. All requests, including future arrivals, are assumed to be known. For instance, SUs submit their requests at the beginning of \( H \). The operator then knows the full arrival information. In each time slot, the operator has to make channel allocation decisions based on the observed availabilities of its own channels and the sensed availabilities of channels managed by other operators. The only difference between online and offline algorithms is that online algorithm does not assume the
full arrival information to be known ahead of time. *Both algorithms are designed under the challenges of spectrum uncertainty and sensing inaccuracy.*

### 5.3 Optimal Offline Algorithm

In this section, we study Problem (A) under the assumption that the operator has full knowledge of the spectrum requests in advance. By our assumptions on channel statistics, the problem can be modeled as a finite horizon Markov Decision Process (MDP) [57]. In this section, we propose an optimal dynamic programming based solution to the problem. We start with the simple case where \( T_1 = \emptyset \) and all the channels for serving SUs are in \( T_2 \), which models the case where all the channels owned by the operator are overloaded by PU traffic. Then, we proceed with the general case where both \( T_1 \) and \( T_2 \) channels are available in the system. In each time slot, based on the knowledge of the spectrum requests and the current channel state, the operator makes a joint decision including 1) which subset of requests to schedule; 2) which subset of channels to allocate; 3) which request to assign to which channel. In our solution, we consider all possible scenarios for each time slot and find the schedule that maximizes the expected social welfare. We show that our algorithm has a complexity of \( O(2^r3^C(\max\{C,r\})^{\min\{C,r\}}HCr) \). When \((a_i, d_i)\) of requests do not have a dense overlap, i.e., \( r = O(\log N) \) where \( N \) is the total number of requests in \([1, H]\), our algorithms are of polynomial complexity. We also provide important structural properties which further substantially reduce the complexity and help design a simple online greedy algorithm.

We first define \( F(D, t) \) as the maximum expected social welfare from the beginning of slot \( t \) till the end of slot \( H \) given that the set of outstanding requests at time \( t \) is \( D \). The expectation takes into account all possible channel realizations and sensing results. We define \( F(D, H + 1) = 0, \forall D \). Our goal is to calculate \( F(\{i : i \in \mathcal{N}, a_i = \)
We calculate it backward from \( t = H \) till \( t = 1 \) is reached since requests requiring service in future time slots have an impact on the current optimal scheduling decision. Note that at any time \( t \), it is sufficient to consider \( D \) in \( F(D, t) \)'s for being any subset of the requests that satisfy \( a_i \leq t < d_i \).

5.3.1 With no available channels in \( T_1 \)

When no channel is in \( T_1 \), the spectrum bands managed by the operator, SUs can only be served by channels in \( T_2 \). SUs may request spectrum in arbitrary time slots. The success of serving request \( i \) contributes \( w_i \) to the social welfare while the assignment of a request to a busy channel causes collisions, incurring a penalty of \( Q \).

We define \( X(D, S, t) \) as the maximum expected social welfare from \( t \) (\( t = 1, \cdots, H \)) to the end of the period, given that the set of outstanding requests is \( D \) and channels in \( S \) are sensed idle (\( S \subseteq T_2 \)). The expectation is taken over all possible realizations of \( I_2 \). Then,

\[
X(D, S, t) = \max_{x(t)} \left[ \sum_{S_1 \subseteq S} \prod_{l \in S_1} P_0(l) \prod_{m \in S \setminus S_1} (1 - P_0(m)) \right. \\
\left. (W(D, S, S_1, x(t), t) + F(D', t + 1)) \right], \tag{5.3.1}
\]

where \( W(D, S, S_1, x(t), t) \)

\[
= \sum_{n \in D} \left[ w_n \sum_{k \in S_1} x_{nk}(t) - Q \sum_{k \in S \setminus S_1} x_{nk}(t) \right]
\]
is defined as the social welfare achieved in time slot \( t \), for a given \( D \), the set of outstanding requests; \( S \subseteq T_2 \), the set of channels sensed idle; \( S_1 \subseteq S \), the set of channels that are sensed idle and actually idle; and \( x(t) \), the channel allocation at \( t \). Recall that \( x_{nk}(t) \) is the allocation indicator used to determine whether the SU is served by this allocation. We form \( D' \) based on \( D \) as follows: If request \( m \) is allocated to channels in \( S_1 \), then remove \( m \) from \( D \), which means it is served and the request
no long exists. If request $n$ satisfies $a_n = t + 1$, then add $n$ to $D$, which indicates it is a new request. Among the remaining requests, those that expire at the beginning of $t + 1$ are removed from $D$.

Based on $X(D, S, t)$, we calculate $F(D, t)$ as follows. The expectation in $F(D, t)$ in the form of the product of $P_I(l)$ and $(1 - P_I(m))$ takes into account all realizations of $I_2^*.$

$$F(D, t) = \sum_{S \subseteq T_2} \prod_{l \in S} P_I(l) \prod_{m \in T_2 \setminus S} (1 - P_I(m))X(D, S, t)$$

(5.3.2)

In Algorithm 5, our objective $F(\{i : i \in N, a_i = 1\}, 1)$ is calculated by dynamic programming. It first calculates the maximum social welfare and the corresponding schedule for each time slot, and then specifies the real time operations. Lines 1-5 calculate $F(D, t)$ backward from $H$ to 1 given the initial condition defined earlier $F(D, H + 1) = 0$ for all $D$. Line 4 calculates the optimal scheduling policy for time $t$ given $D$, the request set; $S$, the set of channels sensed idle; and $S_1$, the set of channels sensed idle and actually idle, according to Equation (5.3.1). The value of $F(D, t)$ is updated in Line 5 according to Equation (5.3.2). The complexity of the Equation (5.3.2) is $O(3^{\left|T_2\right|}(\max\{\left|T_2\right|, r\})^{\min\{\left|T_2\right|, r\}}):$ The number of possible channels realizations is $3^{\left|T_2\right|}$ since different social welfare values will be generated in the cases where the channel is sensed idle but actually busy, it is sensed idle and actually idle, and all other cases. It takes at most $(\max\{\left|T_2\right|, r\})^{\min\{\left|T_2\right|, r\}}$ combinations to find the optimal $x$ in Equation (5.3.1). The complexity for the calculation of $W(D, S, S_1, x(t), t)$ is $O(|T_2| r)$. On the other hand, given $t$, the number of possible argument combinations in $F(D, t)$ is $O(2^rH)$ by assumption. The total time complexity is $O(2^r3^C(\max\{C, r\})^{\min\{C, r\}}HC r).$ Note that $C$ is assumed to be a constant in our model. When there are only homogeneous $T_2$ channels, allocation to different channels in $T_2$ makes no difference. Then, we can replace $(\max\{C, r\})^{\min\{C, r\}}$ with $2^r$, resulting in a complexity of $O(2^{2^r}3^C HCr).$
Algorithm 5 Dynamic Programming based Optimal Algorithm for Social Welfare Maximization

Offline computation

1: for $t = H$ to 1 do
2:   for all $D \subseteq \{i : a_i \leq t < d_i\}$ do
3:     for all $S \subseteq T_2$ do
4:       $X(D, S, t) \leftarrow \max_{\mathbf{x}(t)} \left[ \sum_{S_1 \subseteq S \backslash S_1} \prod_{l \in S_1} P_0(l) \prod_{m \in S \backslash S_1} (1 - P_0(m)) \right.$
5:         $\left. \left( \sum_{n \in D} \left[ w_n \sum_{k \in S_1} x_{nk}(t) - Q \sum_{k \in S \backslash S_1} x_{nk}(t) \right] + F(D', t + 1) \right) \right]$
6:       $F(D, t) \leftarrow \sum_{S \subseteq T_2} \prod_{l \in S} P_I(l) \prod_{m \in T_2 \backslash S} (1 - P_I(m)) X(D, S, t)$

Real-time scheduling

1: At each time slot $t$ with a set of requests $D$ that are currently in the system and a set of channels $S$ that are sensed idle, allocate channels to the requests based on the schedule $\mathbf{x}(t)$ that maximizes $X(D, S, t)$.

5.3.2 With at least one channel in $T_1$

With channels in $T_1$, requests can be served by channels in both $T_1$ and $T_2$. Since the channel availabilities of $T_1$ are known at the beginning of each time slot, they can serve SU requests without any cost. Thus, once observed idle, channels in $T_1$ could be assigned to requests so as to maximize the sum of valuations. Our focus is still the allocation of channels in $T_2$ if they are sensed idle. The differences from the case without $T_1$ channels are as follows: 1) All realizations of $\mathbf{I}_1$ needs to be taken into account; 2) In a schedule $\mathbf{x}(t)$, assignment of any request to a channel in $T_1$ causes no cost.

We define $Y(D, S, t)$ as the maximum expected social welfare from $t$ to the end.
of the period, given that the set of outstanding requests is $D$ and channels in $S$ are sensed idle ($S \subseteq T_2$). We also define $\hat{X}(D, \Gamma, S, t)$ as the maximum expected social welfare from $t$ to the end of the period, given that the set of outstanding requests is $D$, channels in $\Gamma$ are observed to be idle ($\Gamma \subseteq T_1$), and channels in $S$ are sensed idle ($S \subseteq T_2$). The expectation in $Y(D, S, t)$ is for all realizations of $I_1$ and $I_2$. The expectation in $\hat{X}(D, \Gamma, S, t)$ is for all realizations of $I_2$. Then,

$$
\hat{X}(D, \Gamma, S, t) = \max_{X(t)} \left[ \sum_{S_1 \subseteq S} \prod_{l \in \Gamma} P_0(l) \prod_{m \in S \setminus S_1} (1 - P_0(m)) (\hat{W}(D, \Gamma, S, S_1, x(t), t) + F(D', t + 1)) \right],
$$

where

$$
\hat{W}(D, \Gamma, S, S_1, x(t), t) = \sum_{n \in D} \left[ w_n \left( \sum_{l \in \Gamma} x_{nl}(t) + \sum_{k \in S_1} x_{nk}(t) \right) - Q \sum_{k \in S \setminus S_1} x_{nk}(t) \right]
$$

is defined as the social welfare achieved in time slot $t$, given $D$, the set of outstanding requests; $\Gamma$, the set of channels in $T_1$ that are observed to be idle; $S$, the set of channels sensed idle; $S_1$, the set of channels sensed idle and actually idle; and $x(t)$, the channel allocation at $t$. The only difference between $W(D, S, S_1, x(t), t)$ and $\hat{W}(D, \Gamma, S, S_1, x(t), t)$ is the addition of the valuations contributed by the service on channels in $T_1$. We form $D'$ based on $D$ in a similar way to Equation (5.3.1): If a request $m$ is allocated to channels in $S_1 \cup \Gamma$, then remove $m$ from $D$, which means it is served and the request no long exists. If request $n$ satisfies $a_n = t + 1$, then add $n$ into $D$, which indicates it is a new request. All other SU $i$ are removed from $D$ only when $d_i = t + 1$. We then calculate $Y(D, S, t)$ as:

$$
Y(D, S, t) = \sum_{\Gamma \subseteq T_1} \prod_{l \in \Gamma} \pi_1(l) \prod_{m \in T_1 \setminus \Gamma} (1 - \pi_1(m)) \hat{X}(D, \Gamma, S, t).
$$

Hence,

$$
F(D, t) = \sum_{S \subseteq T_2} \prod_{l \in S} P_1(l) \prod_{m \in T_2 \setminus S} (1 - P_1(m)) Y(D, S, t),
$$
**Algorithm 6** Dynamic Programming based Optimal Algorithm for Social Welfare Maximization (at least one $T_1$ channels)

**Offline computation**

1: for $t = H$ to 1 do
2: for all $D \subseteq \{ i : a_i \leq t < d_i \}$ do
3: for all $S \subseteq T_2$ do
4: $Y(D, S, t) \leftarrow \max_{x(t)} \sum_{\Gamma \subseteq T_1} \prod_{l \in \Gamma} \pi_1(l) \prod_{m \in T_1 \setminus \Gamma} (1 - \pi_1(m)) \bar{X}(D, \Gamma, S, t)$
5: $F(D, t) \leftarrow \sum_{S \subseteq T_2} \prod_{l \in S} P_l(l) \prod_{m \in T_2 \setminus S} (1 - P_l(m)) Y(D, S, t)$

**Real-time scheduling**

1: At each time slot $t$ with a set of requests $D$ that are currently in the system, a set of channels $\Gamma$ that are observed idle, and a set of channels $S$ that are sensed idle, allocate channels to the requests based on the schedule $x(t)$ that maximizes $Y(D, S, t)$.

which takes into account all realizations of $I_1$, the availabilities of channels in $T_1$; $I_2^s$, the sensed availabilities of channels in $T_2$; and $I_2$, the actual availabilities of channels in $T_2$.

Correspondingly, the algorithm for the general case (Algorithm 6) is similar to Algorithm 5 except that in Line 4, the allocations of $T_1$ channels is included (replacing $X(D, S, t)$ by $Y(D, S, t)$) and in Line 5, $F(D, t)$ needs to take into account all realizations of $I_1$ (by Equation (5.3.5)). Following a similar argument as in the case where $|T_1| = 0$, the total time complexity is still $O(2^r3^C (\max \{C, r\})^{\min \{C, r\}} HC r)$.

### 5.3.3 Discussion

In this section, we prove some structural properties of the optimal solution, which helps to further reduce the time complexity of the algorithm and also provides insight to the design of the online algorithm discussed in Section 5.4. Note that at any time $t$,
for an active request $i$ and a channel $k \in T_2$ that is sensed idle, $P_0(k)w_i - Q(1 - P_0(k))$ is the expected immediate social welfare contributed by request $i$ if $i$ is assigned to $k$ in the current slot. Proposition 5.3.1 shows that a non-negative immediate social welfare is necessary for request $i$ to be served by channel $k$ in the optimal solution, which turns out to be a sufficient condition in certain scenario as stated in Proposition 5.3.2, as well.

**Proposition 5.3.1.** At any time $t$, if a request $i$ is scheduled on channel $k \in T_2$ in Algorithm 5, then $P_0(k)w_i \geq Q(1 - P_0(k))$.

**Proof.** We will prove Proposition 5.3.1 for the case without $T_1$ channels. It can be shown for the general case in a similar way. Suppose at time $t$, request $i$ is assigned to channel $k$ in the optimal solution, with the system state being $(D, S, S_1)$ as defined before. Note that $k$ may or may not be in $S_1$. Let $x(t)$ be the optimal schedule, $D_S$ be the set of requests scheduled in $x(t)$, and $D'_{S,S_1}$ be the set of outstanding requests for $t + 1$ with $D_S$ scheduled in $t$. Let $\hat{x}(t)$ be the same schedule as $x(t)$ except that $i$ is excluded. To simplify the notation, let $R(S,S_1) = \prod_{l \in S_1} P_0(l) \prod_{m \in S \setminus S_1} (1 - P_0(m))$.

Then we have

$$X(D,S,t) = \left[ \sum_{S_1 \subseteq S, k \in S_1} R(S,S_1)(W(D,S,S_1,\hat{x}(t),t) + w_i + F(D'_S,S_1,t + 1)) \right]$$

$$+ \left[ \sum_{S_1 \subseteq S, k \notin S_1} R(S,S_1)(W(D,S,S_1,\hat{x}(t),t) - Q + F(D'_{S,S_1},t + 1)) \right] \quad (5.3.6)$$

On the other hand, if $i$ is not scheduled, then the expected social welfare from $t$ to $H$ is

$$X'(D,S,t) = \left[ \sum_{S_1 \subseteq S, k \in S_1} R(S,S_1)(W(D,S,S_1,\hat{x}(t),t) + F(D'_{S,S_1} \cup \{i\},t + 1)) \right]$$

96
\[
= \left[ \sum_{S_1 \subseteq S, k \in S_1} R(S, S_1)(W(D, S, S_1, \mathbf{x}(t), t) + F(D_{S,S_1}', t+1)) \right]
\quad \text{(5.3.7)}
\]

Since \(D_S\) is the optimal solution for Equation (5.3.1), we have (5.3.6)-(5.3.7) ≥ 0. By rearranging the terms, we obtain

\[
\sum_{S_1 \subseteq S, k \in S_1} R(S, S_1)(F(D_{S,S_1}', t+1) - F(D_{S,S_1}' \cup \{i\}, t+1))
+ w_i \sum_{S_1 \subseteq S, k \in S_1} R(S, S_1) - Q \sum_{S_1 \subseteq S, k \notin S_1} R(S, S_1)
\]

\[
= \sum_{S_1 \subseteq S, k \in S_1} R(S, S_1)(F(D_{S,S_1}', t+1) - F(D_{S,S_1}' \cup \{i\}, t+1))
+ (w_i P_0(k) - Q(1 - P_0(k))) \sum_{S_1 \subseteq S \setminus \{k\}} R(S, S_1) \geq 0,
\quad \text{(5.3.8)}
\]

where \(F(D_{S,S_1}', t+1) - F(D_{S,S_1}' \cup \{i\}, t+1) \leq 0\) since the social welfare is monotonic over the set of requests. Hence, we must have \(P_0(k)w_i \geq Q(1 - P_0(k))\) for Inequality (5.3.8) to hold.

Proposition 5.3.2 shows that the condition \(P_0(k)w_i > Q(1 - P_0(k))\) is also sufficient for a request to be scheduled for homogenous channels. To simplify notation, we drop the index for channel related parameters for the homogeneous case.

**Proposition 5.3.2.** In a system with no channels in \(T_1\) and homogeneous channels in \(T_2\), if there exists at least one request \(i\) that satisfies \(P_0w_i > Q(1 - P_0)\) in a slot \(t\) and there is at least one channel sensed idle, then in the optimal solution at least one of the requests satisfying this condition will be scheduled, for all \(t\).

**Proof.** Given the system state \((D, S, S_1)\) at time \(t\), consider a subset of requests \(D_S \subseteq D\) to be scheduled where \(i \in D_S\). Let \(D_{S,S_1}'\) denote the set of the outstanding requests at \(t+1\) given that \(D_S\) is scheduled at \(t\), and \(k\) the channel assigned to \(i\) in the schedule. We want to show that the expected social welfare from \(t\) to the end of the time period with \(D_S\) scheduled at \(t\) is at least as large as that with \(D_S \setminus \{i\}\)
scheduled. We define $U(T_2, S) = \prod_{i \in S} P_{I}(l) \prod_{m \in T_2 \setminus S} (1 - P_{I}(m))$. We also define $F_1(D, t)$ as the expected social welfare from $t$ till the end of $H$ by scheduling $D_S$ at time $t$ and $F_2(D, t)$ as the expected social welfare from $t$ till the end of $H$ by scheduling $D_S \setminus \{i\}$ in time $t$. By Equations (5.3.6), (5.3.7), and (5.3.8) we obtain

$$F_1(D, t) - F_2(D, t) = \sum_{S \subseteq T_2} U(T_2, S) \left[ \sum_{S_1 \subseteq S, k \in S_1} P_0^{S_1} (1 - P_0)^{|S \setminus S_1|} \right]$$

$$\left[ F(D'_{S, S_1}, t + 1) - F(D'_{S, S_1} \cup \{i\}, t + 1) \right]$$

$$+ (w_i P_0 - Q(1 - P_0)) \sum_{S_1 \subseteq S \setminus \{k\}} P_0^{S_1} (1 - P_0)^{|S \setminus S_1| - 1}$$

$$= \sum_{S \subseteq T_2} U(T_2, S) \left[ P_0(F(D'_{S, S_1}, t + 1) - F(D'_{S, S_1} \cup \{i\}, t + 1)) \right]$$

$$+ (w_i P_0 - Q(1 - P_0)) \sum_{S_1 \subseteq S \setminus \{k\}} P_0^{S_1} (1 - P_0)^{|S \setminus S_1| - 1} \quad (5.3.9)$$

In the following we will show that $P_0(F(D'_{S_i S_1} \cup \{i\}, t + 1) - F(D'_{S_i S_1}, t + 1)) \leq w_i P_0 - Q(1 - P_0)$. We first observe that $F(D, t) \leq F(D \setminus \{i\}, t) + F(\{i\}, t)$ for any $t$, $D$ and $i \in D$ since $i$ is competing with requests in $D \setminus \{i\}$ for the spectrum in the former case but not in the latter case. Hence we only need to prove that $P_0(F(\{i\}, t + 1) - F(\emptyset, t + 1)) \leq w_i P_0 - Q(1 - P_0)$ for all $t$, that is, $P_0 F(\{i\}, t + 1) = w_i P_0 - Q(1 - P_0)$ for all $t$ since $F(\emptyset, t + 1) = 0$. We will prove it by induction. Let $\hat{P}_I$ denote the probability that at least one channel is sensed idle. For $t = H$, we have $P_0 F(\{i\}, H) = P_0 \hat{P}_I (w_i P_0 - Q(1 - P_0)) \leq w_i P_0 - Q(1 - P_0)$. Suppose $P_0 F(\{i\}, t) \leq w_i P_0 - Q(1 - P_0)$ for all $t > \tau$, then $F(\{i\}, \tau) = (1 - \hat{P}_I) F(\{i\}, \tau + 1) + \hat{P}_I \max \{w_i P_0 + (1 - P_0)(F(\{i\}, \tau + 1) - Q), F(\{i\}, \tau + 1)\}$. Note that

$$F(\{i\}, \tau + 1) - (w_i P_0 + (1 - P_0)(F(\{i\}, \tau + 1) - Q))$$

$$= P_0(F(\{i\}, \tau + 1) - (w_i P_0 - Q(1 - P_0)))^{(a)} \leq 0,$$
where (a) is by the induction assumption. Hence in the optimal solution, \( i \) should be scheduled at \( \tau \) if it is the only request. Therefore,

\[
\begin{align*}
P_0 F(\{i\}, \tau) &= P_0((1 - \hat{P}_t)F(\{i\}, \tau + 1) \\
+ &\hat{P}_t(w_i P_0 + (1 - P_0)(F(\{i\}, \tau + 1) - Q))) \\
= (1 - P_0 \hat{P}_t)P_0 F(\{i\}, \tau + 1) &+ P_0 \hat{P}_t(w_i P_0 - Q(1 - P_0))) \\
\leq w_i P_0 - Q(1 - P_0),
\end{align*}
\]

where (b) is by the induction assumption. Thus \( P_0 (F(D \cup \{i\}, t) - F(D, t)) \leq w_i P_0 - Q(1 - P_0) \) for all \( D \) and \( t \). Therefore, Equation (5.3.9) \( \geq 0 \), which means the expected social welfare from \( t \) to the end of the time period with \( D_S \) scheduled at \( t \) is always better than that with \( D_S \setminus \{i\} \) scheduled.

Based on these propositions, we can reduce the candidate set of requests for scheduling in each time slot. For instance, no requests should be scheduled if \( P_0(k)w_i \leq Q(1 - P_0(k)) \) for all existing requests \( i \) and all \( k \) sensed idle. Also, in a system with no channels in \( T_1 \) and homogeneous channels in \( T_2 \), the candidate set is composed of all requests that satisfy \( P_0(k)w_i > Q(1 - P_0(k)) \). We utilize these propositions in the design of our online algorithm.

### 5.4 Online Algorithm

In this section, we introduce a greedy online algorithm (Algorithm 7) that does not need future arrival information. For systems where requests are not submitted ahead of the required service starting time \( a_i \), the online algorithm makes decisions based on the information available in the current slot. An online algorithm for a maximization problem is \( \alpha \)-competitive \((\alpha \leq 1)\) if it achieves at least a fraction \( c \) of the objective value of an optimal offline algorithm for any finite input sequence \([7]\), where \( c \) is called
a competitive ratio. We show that the greedy online algorithm is 1/2-competitive for homogeneous $T_2$ channels in Proposition 5.4.2, and achieves performance comparable to the optimal offline algorithm for the more general heterogeneous channel case by numerical results (see Figure 5.2(b)).

In Algorithm 7, the main idea is to (greedily) offer requests with higher valuation channels with better quality. We define $c_k \triangleq Q(1 - P_0(k))/P_0(k)$, which is the expected cost of serving one request on channel $k$ (will be shown in Lemma 5.4.1). Note that $c_k = 0$ for $k \in T_1$. Lines 2 and 3 sort channels sensed idle by $c_k$ and current requests by $w_j$, respectively. Since accessing channels in $T_1$ causes no cost if observed idle, they are allocated first to requests with highest valuations (Lines 5-8). In Lines 11-13, the remaining requests are allocated to channels in $T_2$ sensed idle from highest valuation to lowest if they satisfy $w_n > \theta(k)$ where $\theta(k)$ serves as a threshold for using channel $k$. We set $\theta(k) = c_k$ in this section, which is motivated by Propositions 5.3.1 and 5.3.2. Setting different thresholds provides a way for trading off the social welfare and the revenue of the operator, which will be discussed in detail in Section 5.5.

The time complexity of Algorithm 7 is $O(C\log C + r\log r)$ since the complexity of sorting in Lines 2 and 3 dominates that of allocation in Lines 4-13. We then show that the greedy online algorithm is 1/2-competitive when $|T_1| = 0$ and channels in $T_2$ are homogeneous in Proposition 5.4.2. To establish this result, we first show that $c_k$ is the expected cost per a request served by channel $k$ in Lemma 5.4.1.

**Lemma 5.4.1.** For the greedy online policy, $c_k$ is the expected cost of serving a request on channel $k$ when $H \to \infty$.

**Proof.** Let $A'_k$ denote the the number of time slots after the last request is served by channel $k$ in Algorithm 7. We have $A'_k/H \to 0$ if $H \to \infty$. Consider the time interval right after a request is served by channel $k$ and before the next request is served by
Algorithm 7 Greedy Online Algorithm

In each time slot $t$:

1. if $D = \emptyset$ then exit
2. Sort channels in $S$ (sensed idle in $T_2$) by $c_k$ in ascending order
3. Sort requests in $D$ (outstanding ones) by $w_j$ in descending order
4. $i \leftarrow 1$
5. for all $l$ in $\Gamma$ (channels in $T_1$ observed idle) do
6. $x_{il}(t) \leftarrow 1; \ D \leftarrow D \setminus \{i\}$
7. if $D = \emptyset$ then exit
8. $i \leftarrow i + 1$
9. if $D = \emptyset$ then exit
10. $n \leftarrow |\Gamma| + 1$
11. for all $k$ in $S$ (channels sensed idle in $T_2$) do
12. if $w_n \leq \theta(k)$ or $D = \emptyset$ then break
13. $x_{nk} \leftarrow 1; \ D \leftarrow D \setminus \{n\}; \ n \leftarrow n + 1$

channel $k$. Remove all time slots in the interval when there are no requests in the system or channel $k$ is sensed but not allocated. Given that a channel is sensed idle, the probability that collision happens is $1 - P_0(k)$. Thus the number of slots where collisions happen follows a geometric distribution and the expected cost per a request service on channel $k$ is $Q(1 - P_0(k))/P_0(k)$.

Based on Lemma 5.4.1, we show the competitive ratio of Algorithm 7 for homogeneous $T_2$ channels.

Proposition 5.4.2. If $|T_1| = 0$ and channels in $T_2$ are homogeneous, Algorithm 7 is $1/2$-competitive when $H \to \infty$. 

101
Proof. Let the random variable $\gamma$ denote the set of requests that are eventually served by the algorithm. Let $P_0 = P_0(k)$ for any channel $k \in T_2$. Since the channels in $T_2$ are homogeneous, we have $c = Q(1 - P_0)/P_0$, which is the expected cost for serving a single request in Algorithm 7 when $H \to \infty$ by Lemma 5.4.1. Then the expected social welfare achieved by Algorithm 7 can be written as follows:

$$
N \sum_{k=1}^{N} \left( \sum_{|\gamma|=k} \Pr(\gamma) \sum_{i \in \gamma} w_i - kc \Pr(|\gamma| = k) \right) = N \sum_{k=1}^{N} \left[ \sum_{|\gamma|=k} \Pr(\gamma) \sum_{i \in \gamma} (w_i - c) \right]
$$

(5.4.1)

Let $\gamma'$ denote the set of requests that are eventually served by the optimal offline algorithm, then $c$ serves as an lower bound for the expected cost of serving a request by ignoring the time slots after the last request is served. Hence, Equation (5.4.1) with $\gamma$ replaced by $\gamma'$ serves as an upper bound for the expected social welfare achieved by the optimal offline algorithm.

Note that the greedy algorithm always chooses the active request with highest valuation. For any sample path, consider the set of requests served by the optimal offline algorithm and those by the greedy algorithm with $w_i' = w_i - c$ as the valuation. We follow the same argument as in [30]: We consider any request $i$ that is scheduled offline but not online. Since request $i$ is not scheduled online, it is present at time $t$ and the greedy algorithm schedules another request $j$ in that slot, the valuation of request $j$ should be as least as large as that of request $i$. For any request $i$ that is allocated offline and also online, it makes the same contribution to the social welfare. Then the offline solution achieves a social welfare at most twice that in the online solution since $\frac{w_j'}{w_i' + w_j'} \geq \frac{1}{2}$. Therefore, Algorithm 7 is 1/2-competitive when $H \to \infty$. \qed

Note that the factor 2 in Proposition 5.4.2 does not depend on request arrival patterns or channel related parameters. Algorithm 7 can always achieve at least $\frac{1}{2}$ of
the social welfare of the optimal offline algorithm (Algorithm 5) when the system is only composed of homogeneous $T_2$ channels.

### 5.5 Achieving Incentive Compatibility

When the available spectrum resource cannot satisfy all the requests, which is often the case, a selfish SU may choose to cheat on its valuation or arrival and deadline times to obtain some priority of being served. Such strategic behavior leads to a less efficient system. In this section, an online auction scheme is presented, which utilizes the online greedy algorithm (Algorithm 7) together with a payment scheme to suppress the cheating behavior. We first develop a natural payment scheme that uses $c_k$, the expected cost for serving a request by channel $k$, as the reservation price for using channel $k$, and show that the mechanism achieves incentive compatibility (formally defined below). For heterogeneous channels, however, such a variable reservation price has the weakness that the payment charged to an SU not only depend on the valuations of SUs but also the particular channel that serves the SU. Hence, two SUs with the same valuation served simultaneously may be charged different prices simply because they are served by different channels. This leads to arbitrary and unfair treatment of SUs. To avoid this issue, we also introduce a fixed reservation pricing scheme that is independent of channel assignment, which also provides a more straightforward way of trading off social welfare and revenue. The revenue of the operator is composed of two parts: payments collected from the SUs by serving their requests and the penalty paid for causing collisions. For an actual business model to be viable, it is important that the revenue of the operator is taken into account.
5.5.1 Online Auction with Variable Reservation Price

In this section, an online auction scheme using a variable reservation price is presented (see Auction 1) to suppress the cheating behavior of SUs. At any time slot $t$, the operator accepts bids of the form $(\hat{a}_i, \hat{d}_i, \hat{w}_i)$, where $\hat{a}_i = t$ and $\hat{d}_i$ denote the reported required service starting time and the deadline, respectively, and $\hat{w}_i$ denotes the reported valuation. All these values could be different from the true values of request $i$. We assume there is no early-arrival misreport and late-departure misreport in the system, that is, $\hat{a}_i \geq a_i$ and $\hat{d}_i \leq d_i$ in any bid. In practice, both of them can easily be detected since the request is no longer in the system when either misreport occurs.

Let $p_i$ denote the payment that the operator charges a SU for having its request $i$ served. The net utility for request $i$ is defined as: $u_i = w_i - p_i$ if request $i$ is served and $u_i = 0$ if not. A mechanism is said to be dominant-strategy incentive compatible (DSIC) if for any given sample path of channel state realizations and sensing realizations and a set of requests, each request maximizes its utility when it truthfully reveals the private information independent of the bids from other requests (adapted from Definition 16.5 in [51]).

In Auction 1, channels are assigned to requests according to Algorithm 7 by setting $\theta(k) = c_k$, where a request is assigned to channel $k$ only if its valuation is higher than $c_k$. Hence $c_k$ serves as a reservation price. For every request successfully served by its deadline using channel $k$, the charged price is the maximum of the reservation price $c_k$ and and a critical value. A critical value is defined as the maximum reported valuation under which it will not be served, assuming the other bids are fixed. Note that by the definition of payment, the net utility of a SU is always non-negative, and the revenue of the operator never exceeds the social welfare. Furthermore, since a payment is lower bounded by the corresponding reservation price, a non-negative expected revenue of the operator is obtained for large enough $H$ by Lemma 5.4.1.
**Auction 1:** Requests \((\hat{a}_i, \hat{d}_i, \hat{w}_i)\) are reported to the operator at time \(t = \hat{a}_i\).

(i) At the beginning of each \(t\), allocate requests according to Algorithm 7 by setting \(\theta(k) = c_k\).

(ii) A request successfully served by channel \(k\) pays \(\max(c_k, \text{the critical value})\), collected at its reported deadline.

According to Theorem 16.13 in [51], to show that Auction 1 is DISC, it is sufficient to show that the mechanism is monotonic in terms of both valuation and timing. That is, for a given sample path of channel realizations and sensing realizations and a set of requests, if request \(i\) submitting a bid \((\hat{a}_i, \hat{d}_i, \hat{w}_i)\) wins, then it continues to win if it instead submits a bid \((\hat{a}_i', \hat{d}_i', \hat{w}_i')\) with \(\hat{w}_i' > \hat{w}_i\), \(\hat{a}_i' \leq \hat{a}_i\), and \(\hat{d}_i' \geq \hat{d}_i\), assuming other bids are fixed. This condition can be easily verified. So, Auction 1 is DSIC.

By the definition of critical price, we propose Algorithm 8 that applies binary search to find the critical price for requests scheduled by Algorithm 7. Algorithm 8 runs in each slot \(t\) when there are requests scheduled. In the binary search from Lines 6-12, scheduling decisions must be remade from \(t = a_i\) to \(d_i\) with \(w_i\) updated by the new value of \(w_i\) (Line 7) till the critical price is found.

**Remark:** In a traditional VCG like auction [51], the payment charged to a winning bidder is only determined by the valuations of other bidders competing for the same resource. In Auction 1, due to the heterogeneity of spectrum resource, however, it also depends on the particular channel that serves the request. For instance, consider two heterogeneous \(T_2\) channels with \(c_1 \ll c_2\), both of which are sensed idle and really idle at time slot \(t\). Assume that two requests of same valuation \(w > c_2\) arrive at \(t\) and expire at the beginning of \(t + 1\). Since \(w > c_2\), the greedy algorithm will serve request 1 by channel 1 and request 2 by channel 2. For request 1, the critical value is \(c_1\) since given that request 2 reports a valuation of \(w\), request 1 will be served iff it reports a valuation higher than \(c_1\). Hence, the payment charged to request 1 is also
Algorithm 8 Critical Price Calculation for Requests

In each time slot $t$:

1: **for all** $i \in D$ **do**
2:  **if** $i$ is scheduled by Algorithm 7 **then**
3:  $w_{low} \leftarrow 0$
4:  $w_{high} \leftarrow w_i$
5:  **while** $w_{low} < w_{high}$ **do**
6:  $cp_i \leftarrow w_{low} + w_{high} \over 2$
7:  Run Algorithm 7 with the valuation of request $i$ updated by $cp_i$ from $t = a_i$ to $d_i$
8:  **if** $i$ is scheduled **then**
9:  $w_{high} \leftarrow cp_i$
10: **else**
11:  $w_{low} \leftarrow cp_i$
12: **Output** $cp_i$ as the critical price for $i$

$c_1$. Similarly, the payment charged to request 2 is $c_2$. The two requests are charged different prices even if they have the same valuation and face the same competitive environment. Ideally, we would like to hide such resource heterogeneity from SUs.

5.5.2 Online Auction with a Fixed Reservation Price

To overcome the weakness of the variable reservation pricing scheme for the heterogeneous channel case, we consider a fixed reservation price in this section. Let $\theta(k) = q$ in Auction 1, where $q$ is computed using channel related parameters, and is fixed for a given set of channels. Note that the mechanism is still monotonic, hence, DSIC is still ensured in this case. Setting $q$ to different values provide a way for trading off
social welfare and revenue. At a very low reservation price, the payment collected cannot recover the expected cost and hence the average revenue becomes negative. A low reservation price may also harm social welfare by our necessary condition for serving requests (Proposition 5.3.1). On the other hand, when the reservation price is too high, fewer requests will be accepted, which harms both social welfare and revenue. Furthermore, the optimal social welfare and revenue are usually achieved at very different reservation prices (see Figures 5.3(a) and 5.3(b)). In this section, we would like to find a reservation price that matches the average expected cost for serving a request, such that nearly optimal social welfare can be achieved while ensuring the expected revenue of the operator to be non-negative.

Consider any given set of requests. Let $n'_j$ denote the expected fraction of requests served by channel $j$, with the expectation taken over all possible channel and sensing realizations. Then by Lemma 5.4.1, for large enough $H$, the average expected cost per request can be represented as $q' \triangleq \sum_{j \in T_1 \cup T_2} c_j n'_j$. Note that, for homogenous $T_2$ channels, $q' = c_j$ for any $j$. For heterogenous channels, however, finding the accurate value of $q'$ is hard, if not impossible, without accurate knowledge of the request set. We therefore consider an upper bound of $q'$ that can be computed using channel related parameters only. To this end, let $v_j$ denote the probability that the channel $j$ is sensed idle and it is really idle. Then $v_j = \pi_1(j)$ if $j \in T_1$, and $v_j = P_I(j)P_0(j)$ if $j \in T_2$. We then define $m_j \triangleq \frac{v_j}{\sum_{l \in T_1 \cup T_2} v_l}$, and use it to estimate $n'_j$. Note that, $m_j = n'_j$ when the system is always overloaded, that is, if whenever a channel is (sensed) available, it will be used to serve some active request in the system. Now define $q_1 \triangleq \sum_{j \in T_1 \cup T_2} c_j m_j$. Since $q_1$ only depends on channel parameters, it can be easily computed. Note that $q_1 = q'$ for homogenous channels. We will show that $q_1 \geq q'$ and therefore a non-negative expected revenue is obtained using $q_1$ as the reservation
price in Proposition 5.5.2. Furthermore, using $q_1$ also achieves good performance in terms of social welfare as shown in numerical results (see Figures 5.3(a) and 5.3(b)).

In the following, we assume that channels have been sorted by a non-decreasing order of $c_j$. We start with Lemma 5.5.1 that provides a sufficient condition for $q_1 \geq q'$.

**Lemma 5.5.1.** If $\frac{m_j}{m_{j+1}} \leq \frac{n'_j}{n'_{j+1}}$ for all $j$, then $q_1 \geq q'$.

**Proof.** We calculate $q_1 - q' = \sum_{j \in T_1 \cup T_2} c_j(m_j - n'_j)$. In the following, we will show that there exists $i$ such that for all $j \leq i$ we have $m_j \leq n'_j$ and for all $k > i$ we have $m_k > n'_k$. Since $\frac{m_j}{m_{j+1}} \leq \frac{n'_j}{n'_{j+1}}$ for all $j$, it is easy to see that: if $m_j \geq n'_j$, then $m_k \geq n'_k$ by multiplying $\frac{m_{j+1}}{m_j} \cdots \frac{m_k}{m_{k-1}}$ and $\frac{n'_{j+1}}{n'_j} \cdots \frac{n'_k}{n'_{k-1}}$, respectively, on both sides. Then we can find such $i$. We divide $q_1 - q'$ into two parts:

$$q_1 - q' = \sum_{j \leq i} c_j(m_j - n'_j) + \sum_{k > i} c_k(m_k - n'_k) \tag{5.5.1}$$

If $i = |T_1| + |T_2|$, $q_1 - q' = \sum_{j \in T_1 \cup T_2} c_j(m_j - n'_j) \geq \left(\max_{j \in T_1 \cup T_2} c_j\right) \left(\sum_{j \in T_1 \cup T_2} m_j - \sum_{j \in T_1 \cup T_2} n'_j\right) = \left(\max_{j \in T_1 \cup T_2} c_j\right)(1 - 1) = 0$. If $i = 0$, then all terms in $q_1 - q'$ are positive. Next we consider the case where neither sums in Equation (5.5.1) has no terms. Since all terms in the first term in the sum are non-positive and all terms in the second term in the sum are positive, we obtain

$$q_1 - q' \geq \left(\max_{j \leq i} c_j\right) \sum_{j \leq i} (m_j - n'_j) + \left(\min_{k > i} c_k\right) \sum_{k > i} (m_k - n'_k)$$

$$\geq \left(\max_{j \leq i} c_j\right) \left(\sum_{j \leq i} (m_j - n'_j) + \sum_{k > i} (m_k - n'_k)\right) \tag{a}$$

$$= \left(\max_{j \leq i} c_j\right) \left(\sum_{j \in T_1 \cup T_2} m_j - \sum_{j \in T_1 \cup T_2} n'_j\right) = 0,$$

where (a) is by the assumption that $c_1 \leq \cdots \leq c_{|T_1| + |T_2|}$. Hence $q_1 \geq q'$ holds. \qed
Based on Lemma 5.5.1, we claim that a reservation price of $q_1$ results in a non-negative revenue for the operator.

**Proposition 5.5.2.** The operator obtains a non-negative expected revenue with reservation price $q_1$ when $H$ is large enough.

**Proof.** It suffices to show that $q_1 \geq q'$. Consider a given set of requests and any sample path of channel states. Without loss of generality, consider the first two channels in the sorted list. Let $n_1$ and $n_2$ denote the number of requests served by channels 1 and 2, respectively. Let $s_i$ denote the number of time slots that are sensed and allocated in the interval between $(i-1)$th and $i$-th requests served by channel 1, and define $b_i$ similarly for channel 2. Let $A$ denote the total number of time slots between 0 and $H$ that are not sensed or sensed but not allocated for channel 1, and $A'$ the number of time slots after the last request is served by channel 1. Define $B$ and $B'$ similarly for channel 2. Note that by the ordering of channels, when there is only one request in the system, and both channels are available, channel 1 will be used. It follows that $A \leq B$. We then have $H = \sum s_i + A + A' = \sum b_i + B + B'$. Therefore, $H = E(\sum s_i + A + A') = E(n_1)/m_1 + E(A) + E(A')$ (by geometric distribution) and $H = E(n_2)/m_2 + E(B) + E(B')$. Note that $E(A')/H \to 0$ and $E(B')/H \to 0$ when $H \to \infty$. Therefore $\frac{n_1}{n_2} = \frac{E(n_1)}{E(n_2)} = \frac{[H - E(A)]m_1}{[H - E(B)]m_2} \geq \frac{m_1}{m_2}$ since $H - E(A) \geq H - E(B)$. It then follows that $q_1 \geq q'$ by Lemma 5.5.1. Hence, a reservation price of $q_1$ leads to a non-negative revenue at the operator. \qed

### 5.6 Numerical Result

In this section, we evaluate the performance of the greedy online algorithm (Algorithm 7) and the tradeoff between social welfare and revenue for different reservation prices. We first show the performance of the greedy online algorithm compared with
the optimal offline algorithm under different channel settings and request related parameters, respectively. We then apply Auction 1 with varying reservation prices and show the performances of social welfare and revenue.

We let the arrivals $a_i$ of requests follow a Poisson distribution and the duration $d_i - a_i$ of the requests follow an exponential distribution. The valuations follow a uniform distribution in $[1, 15]$. We choose $Q = 10$, the penalty per collision, comparable to the valuations in all our simulations. We fix the number of requests as 20, and the inter-arrival mean as 3 slots, and vary the mean of request duration to adjust the density of requests. Given the means of inter-arrival and request durations, we generate 50 groups of requests and compare the average for the metrics we consider. We generate the channel availabilities in each time slot based on our assumption that channel states follow an i.i.d Bernoulli distribution and 100 samples of channel realizations are taken for our simulations. The channel parameters we use will be introduced in Section 5.6.1.

5.6.1 Performance of Greedy Online Algorithm

In Figure 5.2(a), we compare the performance of Algorithm 7 with that of Algorithm 6 when there are three homogeneous $T_2$ channels in the system with $\pi_2 = 0.6324$, $P_m = 0.2218$, $P_f = 0.6595$ and various number of $T_1$ channels. The $y$-axis denotes the achieved performance ratio, i.e., the ratio between the social welfare of the online algorithm and that of the optimal offline algorithm. When $|T_1| = 1$, we set $\pi_1 = 0.5058$; When $|T_1| = 2$, we set $\pi_1(1) = 0.8147$ and $\pi_1(2) = 0.1270$. We observe that the performance of Algorithm 7 degrades as $|T_1|$ increases, independent of the request duration mean. With a high number of $T_1$ channels, a wrong decision made by the greedy online algorithm to schedule a request affects the performance more. Also, the greedy online algorithm serves requests of a larger density better than requests.
of a smaller density. When the system is overloaded with requests, even the optimal offline algorithm can not satisfy all requests. Thus, those with larger valuations tend to be chosen, as in the greedy online algorithm. All ratios plotted are strictly above $\frac{1}{2}$, even for those with $|T_1| \neq 0$.

In Figure 5.2(b), we evaluate the performance of Algorithm 7 with heterogeneous $T_2$ channels. We use the same $T_1$ channel parameters as in the homogeneous case.

Figure 5.2: Performance of online algorithm versus offline algorithm over various request duration means with homogeneous and heterogeneous $T_2$ channels ($|T_2| = 3$), respectively.

(a) Homogeneous $T_2$ channels.

(b) Heterogeneous $T_2$ channels.
The parameters related to $T_2$ channels are as follows: $\pi_2(1) = 0.9134$, $\pi_2(2) = 0.6324$, $\pi_2(3) = 0.0975$, $P_m(1) = 0.1419$, $P_f(1) = 0.7922$, $P_m(2) = 0.2218$, $P_f(2) = 0.6595$, $P_m(3) = 0.6557$, $P_f(3) = 0.2157$. We observe similar results as in Figure 5.2(a): Algorithm 7 performs better with fewer $T_1$ channels and denser requests. Again all ratios are above $\frac{1}{2}$.

5.6.2 Tradeoff between Social Welfare and Revenue

We now study the tradeoff between social welfare and revenue generated by Auction 1 with a fixed reservation price. In Figure 5.3, the request duration mean is fixed at 3 time slots, and we vary the values of reservation price $q$. The channel related parameters are the same as in Section 5.6.1. We first show the tradeoff in a system with homogeneous $T_2$ channels and no $T_1$ channels in Figure 5.3(a). We observe that both social welfare and revenue first increase and then decrease as the reservation price increases. Note that the highest social welfare and revenue are obtained at very different reservation prices with $q = 3.8$ for the former and $q = 7$ for the latter. At a low reservation price ($< 3$), the payment collected cannot recover the expected cost and hence the average revenue becomes negative. We note that when the reservation price is $q_1 = 3.8$ (defined in Section 5.5.2), a non-negative revenue is obtained together with a high social welfare. Furthermore, at a very high reservation price ($\geq 8$), the social welfare and the revenue converge, where the payment actually becomes the same as the valuation for requests served.

In Figure 5.3(b), we show the tradeoff in a system with $T_1$ and heterogeneous $T_2$ channels. The trend of social welfare and revenue is similar to that in Figure 5.3(a). The highest social welfare and revenue are obtained at $q = 10$ and $q = 13$, respectively. Note that with the reservation price $q_1 = 6.9$, the revenue obtained is right above
0, which is consistent with Proposition 5.5.2 and also shows that $q_1$ is nearly a tight upper bound of the expected cost for this case.

5.7 Summary

In this chapter, we study the joint sensing and spectrum allocation problem for serving secondary users in cognitive radio networks with the objective of maximizing the
social welfare. Our problem formulation takes into account both spectrum uncertainty and sensing inaccuracy, which enables dynamic spectrum access at small time scales. Using only channel statistics and real time channel states, we develop an optimal solution for serving a given set of spectrum requests with various time elasticity. We then propose an online algorithm that achieves comparable performance as the offline algorithm. We further extend the online algorithm to an online auction scheme, which ensures incentive compatibility for the SUs and also provides a way for trading off social welfare and revenue.
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N$</td>
<td>Set of spectrum requests submitted to the operator</td>
</tr>
<tr>
<td>$T_1$</td>
<td>Set of channels managed by the operator</td>
</tr>
<tr>
<td>$T_2$</td>
<td>Set of channels not managed by the operator</td>
</tr>
<tr>
<td>$\pi_1(i)$</td>
<td>Probability that channel $i$ in $T_1$ is idle</td>
</tr>
<tr>
<td>$\pi_2(j)$</td>
<td>Probability that channel $j$ in $T_2$ is idle</td>
</tr>
<tr>
<td>$C$</td>
<td>The total number of channels ($C =</td>
</tr>
<tr>
<td>$\vec{I}_1(t)$</td>
<td>Availabilities of channels in $T_1$ at $t$</td>
</tr>
<tr>
<td>$\vec{I}_2(t)$</td>
<td>Availabilities of channels in $T_2$ at $t$</td>
</tr>
<tr>
<td>$\vec{I}_2^s(t)$</td>
<td>Sensed availabilities of channels in $T_2$ at $t$</td>
</tr>
<tr>
<td>$P_f(k)$</td>
<td>Probability of false alarm for channel $k \in T_2$</td>
</tr>
<tr>
<td>$P_m(k)$</td>
<td>Probability of misdetection for channel $k \in T_2$</td>
</tr>
<tr>
<td>$P_I(k)$</td>
<td>Probability that channel $k$ is sensed idle</td>
</tr>
<tr>
<td>$P_0(k)$</td>
<td>Probability of channel $k$ being idle given that it is sensed idle</td>
</tr>
<tr>
<td>$a_i$</td>
<td>Earliest service time for request $i$</td>
</tr>
<tr>
<td>$d_i$</td>
<td>Deadline of request $i$</td>
</tr>
<tr>
<td>$w_i$</td>
<td>Valuation of the request $i$</td>
</tr>
<tr>
<td>$H$</td>
<td>The time period where spectrum allocation has to be made</td>
</tr>
<tr>
<td>$r$</td>
<td>Maximum number of outstanding requests in the system at any time</td>
</tr>
<tr>
<td>$Q$</td>
<td>Penalty price per collision</td>
</tr>
</tbody>
</table>
CHAPTER 6
DISTRIBUTED QUEUE LENGTH BASED SCHEDULING ALGORITHM

In Chapter 3 through 5, we consider systems with overloaded traffic where our goal is to maximize the served traffic amount or sum of their service valuations. In this chapter, we focus on arrivals within the capacity region of the system and study scheduling policies for CRNs, since developing a distributed implementation that can fully utilize the spectrum opportunities for secondary users (SUs) has so far remained elusive. Although throughput optimal algorithms based on the well-known Maximal Weight Scheduling (MWS) algorithm exist for cognitive radio networks, they require central processing of network-wide SU information. In this chapter, we devise a new distributed algorithm that asymptotically achieve the capacity region of the cognitive radio systems.

6.1 Introduction

We develop a new distributed throughput optimal scheduling algorithm for CRNs. To this end, we introduce a new system state representation that includes channel state information, and design our algorithms to achieve throughput optimality without causing interference with PUs. In this chapter, we focus on the algorithm design for single-channel cognitive radio networks. Our analysis can be immediately extended
to a system with orthogonal channels, where scheduling is performed per channel. However, extensions to non-orthogonal channels are beyond the scope.

This chapter is organized as follows: In Section 6.2, the system model is introduced. In Section 6.3, we discuss some technical challenges that require a new state space representation and then describe the appropriate representation on which our new distributed queue-length based scheduling algorithm for CRNs is built. Numerical performance evaluations are presented in Section 6.4. The summary is in Section 6.5.

6.2 System Model

We consider a cognitive radio network consisting of \( N \) SUs coexisting with a PU network (Figure 6.1), where the PU network is represented as a single source of emission and the SUs communicate with their neighbors directly. The PU network has a single designated channel to transmit on. We adopt the exclusive communication approach to interference avoidance, i.e., an SU can transmit only when the PU network does not use the channel. The set \( S \) denotes all SUs that are outside the interference range of the PU network while \( \bar{S} \) is the set of all other SUs. i.e., SUs 1-3 are in \( \bar{S} \) and SUs 4-7 are in \( S \). SUs in \( S \) have access to the channel at any time since its transmission does not interfere with the PU network. SUs in \( \bar{S} \) can sense the channel and keep silent when the PU network is active.

The neighboring SUs of a given SU \( i \) (\( i = 1, \cdots, N \)) is denoted by \( C(i) \) such that \( i \) and any of its neighbors \( j \in C(i) \) cannot transmit at the same time. Note that both the \( k \)-hop (\( k \geq 1 \)) [37] and distance-based [29] interference models are included as special cases. Symmetry is assumed in the conflict set, i.e., if \( i \in C(j) \), then \( j \in C(i) \).

We consider a time-slotted system with unit capacity links. A feasible schedule includes SUs that can be active at the same time subject to the conflict set constraints.
Let $M$ be the set of all feasible schedules. A schedule is represented by a set $x (x \in M)$ and we denote the vector of schedule by $\vec{x} \in \{0, 1\}^N$ such that $x'_i = 1$ if SU $i$ is in the schedule $x$. So $i \in x$ indicates that $x'_i = 1$. The capacity region of the SUs, in the absence of PU activity is defined as $\Lambda = \{\vec{\lambda}|\vec{\lambda} \succeq 0 \text{ and } \exists \vec{\mu} \in Co(M), \vec{\lambda} \prec \vec{\mu}\}$, where $Co(\cdot)$ is the convex hull and $\succeq$, $\prec$ are component-wise “greater than or equal to” and “smaller than” operators, respectively. The actual capacity region of SUs is the interior of $\Lambda$ subject to the PU activity. The queue lengths at SUs evolve as a Markov Chain with the transitions caused by arrivals and departures in the current time slot. The system is said to be stable if this Markov Chain is positive recurrent. A throughput-optimal algorithm stabilizes the network for any arrival rate in the actual capacity region.

In this work, single-hop flows for both the primary and the secondary systems are considered. We assume that the PU has i.i.d. Bernoulli traffic in each time slot, where the PU is idle (channel is available) with probability $p$. $B(t)$ is defined as the channel state at time $t$, where $B(t) = 0$ if the channel is available (PU is idle) and $B(t) = 1$ if the channel is unavailable (PU is active).
6.3 The Distributed Scheduling Algorithm for CRNs

6.3.1 Q-CSMA Overview

As mentioned in the introduction, throughput optimal CSMA based distributed scheduling algorithms such as Q-CSMA [49] have been proposed in the recent past. It is tempting to apply such algorithms to achieve throughput optimality in a distributed manner. In the following, we first present an overview of the Q-CSMA algorithm. We then demonstrate why Q-CSMA cannot be directly applied to CRNs, which motivates our main contributions in this work.

In [49], a discrete-time distributed randomized algorithm is proposed to achieve the full capacity region in a single-channel network. The algorithm of [49] is based on a generalization of Glauber dynamics in statistical physics. In Glauber dynamics, only one link has a state update within a time slot. In scheduling, a state update can be interpreted as a transition of a link from “transmitting” to “idle” or from “idle” to “transmitting”. The incremental state update in every time slot leads to a scheduling policy sufficiently close to MWS, which guarantees the throughput optimality. In [49], multiple links are allowed to update their states in a single time slot. This minor change, which results in improved delay performance, does not affect throughput optimality.

A more detailed description of the Q-CSMA algorithm is as follows: Each time slot $t$ is divided into a control slot and a data slot, where the control slot is much smaller than the data slot. In the control slot, a collision-free transmission schedule is generated and used for data transmission in the data slot. Let $m(t)$ be a set of SUs that do not conflict with each other and selected randomly in the control slot (the scheme will be presented in Section 6.3.2). $M_0$ denotes the set of all $m(t)$ which is all possible schedules. So $M_0$ includes all feasible schedules that could be generated
by the randomized algorithm. Note that $M_0 \subseteq M$, the set of all feasible schedules. The network randomly selects a feasible schedule $m(t)$, which is called the decision schedule in [49]. $m(t)$ can be regarded as a candidate schedule. Note that $m(t)$ and $m(t - 1)$ are independent for all $t > 0$ because $m(t)$ is chosen independently in the subsequent control slot [49]. Each link within $m(t)$ will be checked to decide whether it will be included in the transmission schedule $x(t)$. Link $i \in m(t)$ may be included in $x(t)$ if $\forall j \in C(i), j \notin x(t - 1)$; otherwise, Link $i$ is not included in $x(t)$. Furthermore, link $k \notin m(t)$ is included in $x(t)$ if $k \in x(t - 1)$. The detailed algorithm is as follows: links in $m(t)$ that had no neighbors active in the previous data slot are allowed to update their states with a certain probability which is a function of their queue lengths; those outside the decision schedule $m(t)$ maintain their states. By explicitly taking into account collisions in the control slot, the algorithm generates collision-free transmission schedules $x(t)$ for the data slot. More importantly, the Discrete Time Markov Chain (DTMC) with the transmission schedule chosen as the state is shown to be time-reversible and has product-form stationary distribution, which are used to prove throughput optimality of this algorithm.

The operation of the Q-CSMA algorithm is illustrated in Figure 6.2 for a simple two-link topology, where the two links interfere with each other. State $(0, 0)$ represents that neither link transmits and state $(0, 1)$ indicates that only link 2 transmits. $\alpha_i$ is the probability that link $i$ ($i = 1, 2$) is chosen in the decision schedule $m(t)$ and $p_i$ is the probability that link $i$ is activated given it is in $m(t)$ and no neighbors were active in the previous data slot. The DTMC is time reversible and has the following product-form stationary distribution: $\pi(0, 0) = \frac{1}{Z}$, $\pi(0, 1) = \frac{1}{Z} \frac{p_2}{1 - p_2}$, $\pi(1, 0) = \frac{1}{Z} \frac{p_1}{1 - p_1}$ where $Z = 1 + \frac{p_1}{1 - p_1} + \frac{p_2}{1 - p_2}$ and $\pi(a, b)$ is the stationary distribution for state $(a, b)$ ($a, b = 0, 1$), which are in accordance with Proposition 1 of [49].
Figure 6.2: DTMC with the vector of transmission schedule $\vec{x}(t)$ of two links as the state. Two links interfere with each other.

We next illustrate why $x(t)$ is a poor choice for representing the state in CRNs, and leads to a non-reversible DTMC. For clarity, we use $\vec{x}(t)$, the vector of transmission schedule as defined in Section 6.2, as the state. Consider a simple example in Figure 6.3 for three interfering SU links, where 1 interferes with 2, 2 interferes with 3 and 1 does not interfere with 3. All SU links are within the interference range of the PU. Similar to the earlier example, 1 (0) indicates that an SU link is (not) transmitting. For instance, state (1, 0, 1) means that SU links 1 and 3 are transmitting and SU link 2 is not transmitting. Note that (0, 0, 0) includes two cases: 1) The channel is available and no SU link is transmitting; 2) The channel is unavailable. Let $\alpha_i$ be the probability that SU link $i$ is chosen in the decision schedule $m(t)$ ($i = 1, 2, 3$), $\alpha_4$ the probability that both 1 and 3 are chosen in $m(t)$, and $p_i$ the probability that SU link $i$ is activated ($i = 1, 2, 3$) given it is in $m(t)$ and no neighbors were active in the previous data slot. Now we examine the transitions from (0, 0, 0), (1, 0, 0) to (1, 0, 1) clockwise and counter-clockwise. These two probabilities are not equal so the DTMC is not time reversible by Kolmogorov’s criterion [39].
To treat the “available” and “unavailable” channel separately, we incorporate the channel state into the state space design. The new DTMC is illustrated in Figure 6.4 where we consider two SUs interfering with each other and both are in the interference range of the PU. The new system state is defined as $X' = (x'; B)$ where $x'$ is the vector of transmission schedule and $B$ is the channel state defined in Section 6.2 (0 represents that the channel is idle; 1 represents that the channel is busy). For instance, (0, 0; 1) indicates that the channel is unavailable and neither SU is transmitting; (0, 1; 0) means the channel is available and SU 2 is transmitting. Note that $\alpha_1$ and $\alpha_2$ have the same definitions as before; $\alpha_0$ is the probability that neither SU is selected in the decision schedule ($\alpha_0 + \alpha_1 + \alpha_2 = 1$). We now check to see whether the DTMC is time reversible by examining the transitions from (0, 0; 1), (0, 0; 0) to (0, 1; 0).
Figure 6.4: DTMC with both the vector of transmission schedule $\vec{x}'(t)$ of two SUs and channel state as the state. Two SUs are in the conflict set of each other.

clockwise and counter-clockwise. The product of clockwise transition probabilities is $p^2(1-p)p_2\alpha_2(\alpha_1(1-p_1) + \alpha_2(1-p_2) + \alpha_0)$ and the product of counter-clockwise transition probabilities is $p^2(1-p)\alpha_2^2p_2(1-p_2)$. These two are not equal so the DTMC is not time reversible by Kolmogorov’s criterion [39].

To generate a time reversible DTMC with a product form stationary distribution for a general topology where there are SUs in $S$ (outside the interference range of the PU network), we introduce two Markov chains with new state definitions. We define\(^1\)

$$\vec{y}^a(t) = \{\vec{x}'(\tau) : \text{the largest } \tau \leq t \text{ with } B(\tau) = 0\}$$

$$\vec{y}^b(t) = \{\vec{x}'(\tau) : \text{the largest } \tau \leq t \text{ with } B(\tau) = 1\}$$

\(^1\)We assume $t$ starts from $-\infty$ to make $\vec{y}^a$, $\vec{y}^b$ well-defined, especially for $\vec{y}^a(t)$ when $B(0) = \cdots = B(t) = 1$ and $\vec{y}^b(t)$ when $B(0) = \cdots = B(t) = 0$, respectively.
Note that $\vec{y}^a$ is the vector of transmission schedule in the most recent data slot (including time $t$) when the channel is ON; $\vec{y}^b$ is the vector of transmission schedule in the most recent data slot (including time $t$) when the channel is OFF. We also denote the corresponding transmission schedules by $y^a$ and $y^b$, respectively: $i \in y^a$ if and only if $\vec{y}^a_i = 1$ and $i \in y^b$ if and only if $\vec{y}^b_i = 1$. We then define $Y^a = (\vec{y}^a(t); B(t))$, $Y^b = (\vec{y}^b(t); B(t))$ as the states for the two chains, respectively. For instance, in Figure 6.5, two SUs are in the conflict set of each other with SU 1 in $\bar{S}$ and SU 2 in $S$: $(0, 1; 0)$ indicates that the channel is available and only SU link 2 is transmitting; $(1, 0; 1)$ means that the channel is unavailable and in the most recent available channel state, only SU link 1 is transmitting. The transitions in Figure 6.5 follows exactly from Algorithm 9 described later in Section 6.3.2. It is obvious that the transition to the next state depends only on the current state and the current input including the channel state $B$ and the decision schedule. Hence both chains are Markovian. It is easy to verify, as in Figure 6.4, that Markov chains with $Y^a$, $Y^b$ as the states are not time reversible.\(^2\)

We further define an aggregate state for each DTMC, which includes only $\vec{y}(t)$.

\[ DTMC^a : \bar{Y}^a = (\vec{y}^a(t)) \]
\[ DTMC^b : \bar{Y}^b = (\vec{y}^b(t)) \]

Note that in Figure 6.5, the rectangles in the first chain correspond to $\bar{Y}^a$ and those in the second chain correspond to $\bar{Y}^b$. In $DTMC^a$, the transition to the next state $\bar{Y}^a$ depends only on the current state $\bar{Y}^a$ and the current input including the channel state $B$ and the decision schedule. Thus $DTMC^a$ is Markovian. Similar arguments

\(^2\)Note that we also use Figure 6.5 to illustrate the Markov chains with the aggregate states $\bar{Y}^a$ and $\bar{Y}^b$ later. The transitioning probabilities for each single state are not labeled in Figure 6.5. By Algorithm 9, it is easy to find that the outgoing probabilities from $(\vec{y}^a; 0)$ and $(\vec{y}^a; 1)$ to $(\vec{y}'; 0)$ are the same. The incoming probabilities from $(\vec{y}'; B(t))$ to $(\vec{y}^a; 1)$ do not exist if $\vec{y}' \neq \vec{y}^a$. 

124
can be applied to $DTMC^b$. In Section 6.3.2, both DTMCs will be shown to be time reversible with a product form stationary distribution.

### 6.3.2 Scheduling Algorithm for Single Channel CRNs

Our new algorithm is based on the redefinition of the system state of Section 6.3.1 along with the differentiated treatment of SUs inside and outside the PU interference range. Each SU $i$ keeps a queue denoted by $q_i$. The transmission schedule for $q_i$ is denoted by $x'_i$. We define $A_i(t)$ as the arrivals to SU $i$ at time slot $t$, $i = 1, \cdots, N$ and we assume it to be bounded. We assume arrivals are i.i.d. over time and independent between users. $\lambda_n$ is defined to be $E(A_n(t))$. The decision schedule for all SUs is denoted by $m^a(t)$. The decision schedule for SUs in $S$, denoted by $m^b(t)$, does not consider the interference to or from SUs in $\bar{S}$. The set of all $m^a(t)$ and $m^b(t)$ are denoted by $M^a_0$ and $M^b_0$, respectively. We define $\alpha^a(m^a(t))$ and $\alpha^b(m^b(t))$ as the probability that $m^a(t)$ is chosen in the control slot when the channel is available, and the probability that $m^b(t)$ is chosen in the control slot when the channel is unavailable, respectively. For clarification, “available” means the PU is idle and “unavailable” means the PU is active although SUs in $S$ cannot sense it. Recall that $S$ is the set of all SUs that are outside the interference range of the PU network.

We first develop Algorithm 9 that characterizes the different behaviors of SUs in $S$ and $\bar{S}$ under different channel states. SUs in $\bar{S}$ acquire channel state information in every time slot by locally sensing the channel while SUs in $S$ are notified by SUs in $\bar{S}$ for the new channel state. Next we elaborate on the behaviors of the SUs on the channel state transitions. When the channel state is available, all SUs treat the most recently available slot as their previous slot ignoring the unavailable period, and schedule packets in a way similar to Q-CSMA (Lines 2-11) where $\bar{p}_i = 1 - p_i$ [49]. When the channel state is unavailable, SU $i$ in $S$ remains silent and SUs in $S$ treat
Algorithm 9 Schedule()
1: if channel is available /*$B(t) = 0$*/ then
2: 1. In the control slot, randomly select a decision schedule $m^a(t) \in M^a_0$ with probability $\alpha^a(m^a(t))$
3: if $i \in m^a(t)$ and $y^a_j(t-1) = 0$ for all $j \in C(i)$ then
4:   (a) $x'_i(t) = 1$ with probability $p_i$
5:   (b) $x'_i(t) = 0$ with probability $\bar{p}_i$
6: else if $i \in m^a(t)$ and $y^a_j(t-1) = 1$ for some $j \in C(i)$ then
7:     $x'_i(t) = 0$
8: else
9:     $x'_i(t) = y^a_i(t-1) /*i \notin m^a(t)*/$
10: 2. In the data slot, use $\vec{x}'(t)$ as the transmission schedule
11: else
12: Execute Lines 2-11 by replacing all a with b

the most recently unavailable slot as their previous slot ignoring the available period, and schedule packets in a way similar to Q-CSMA (Lines 13-22) [49]. In other words, SUs in $\bar{S}$ either retrieve or record information on the activities of SUs in $C(i)$ when channel state changes while SUs in $S$ have to record and retrieve on the channel state change.

To understand Algorithm 9 better, we consider the illustrative example (Figure 6.5) with SU 1 inside the interference range of the PU ($1 \in \bar{S}$) and SU 2 ($2 \in S$) outside. $(0, 1; 0)$ indicates that the channel is available and only SU 2 is transmitting; $(1, 0; 1)$ means the channel is unavailable and in the most recent available channel state, only SU 1 is transmitting. $(0; 1)$ indicates that the channel is unavailable and SU 2 is not transmitting; $(1; 0)$ means the channel is available and in the most recent
unavailable channel state, SU 2 is transmitting. When the channel is available, the first chain in Figure 6.5 transitions to the next state though the second chain only stays in the previous state (Lines 2-11 in Algorithm 9); when the channel is unavailable, the second chain transitions to the next state while first chain only stays in the previous state (Lines 13-22).

In the following, we will formally show that Algorithm 9 achieves throughput optimality. The transition probabilities are presented in Lemmas 6.3.1 and 6.3.3. Propositions 6.3.2 and 6.3.4 give the product-form of the stationary distribution. Proposition 6.3.6 claims the throughput optimality of Algorithm 9. We define $\pi(v) := \text{prob(state is } v)$.

**Lemma 6.3.1.** (a) A state $\bar{Y}^a = (\bar{y}^a(t))$ can make a transition to a state $\bar{\hat{Y}}^a = \hat{y}^a(t)$ ($\bar{y}^a \neq \hat{y}^a$) iff

$$y^a \cup \hat{y}^a \in M_0^a$$

and there exists a decision schedule $m^a \in M_0^a$ s.t.

$$y^a \triangle \hat{y}^a := (y^a \setminus \hat{y}^a) \cup (y^a \setminus \hat{y}^a) \subseteq m^a.$$  

(b) The transition probability $P(\bar{Y}^a, \bar{\hat{Y}}^a)$ from $\bar{Y}^a$ to $\bar{\hat{Y}}^a$ ($\neq \bar{Y}^a$).

$$P(\bar{Y}^a, \bar{\hat{Y}}^a) = \sum_{m^a \in M_0^a : y^a \triangle \hat{y}^a \subseteq m^a} p\alpha^a(m^a) \left( \prod_{l \in y^a \setminus \hat{y}^a} \tilde{p}_l \right) \left( \prod_{i \in y^a \cap (y^a \cup \hat{y}^a)} p_i \right) \left( \prod_{j \in m^a \setminus (y^a \cup \hat{y}^a) \cap C(y^a \cup \hat{y}^a)} \tilde{p}_j \right),$$

(6.3.1)

where $C(y^a \cup \hat{y}^a)$ denotes the neighbors of nodes in $y^a \cup \hat{y}^a$. 

127
Proof. Part (a) can be proven as Lemma 2 in [49]. To prove Part (b), we denote \( P^{sch}(\vec{y}^a, \vec{\tilde{y}}^a) \) as \( P(\vec{y}', \vec{y}') \) in Lemma 2 of [49] which is the transition probability from state \( \vec{y}' \) to state \( \vec{\tilde{y}}' \) with the always-available channel. We only need to show
\[
P((\vec{y}^a ; 0), \vec{\tilde{y}}^a ) = P((\vec{y}^a ; 1), \vec{\tilde{y}}^a ) = pP^{sch}(\vec{y}^a , \vec{\tilde{y}}^a ).
\]
The first equality is obvious by Algorithm 9.
\[
P((\vec{y}^a ; 0), \vec{\tilde{y}}^a ) = P((\vec{y}^a ; 0), (\vec{y}' ; 0)) + P((\vec{y}^a ; 0), (\vec{y}' ; 1)) = pP^{sch}(\vec{y}^a , \vec{\tilde{y}}^a ) + 0;
\]
\[
P((\vec{y}^a ; 1), \vec{\tilde{y}}^a ) = P((\vec{y}^a ; 1), (\vec{y}' ; 0)) + P((\vec{y}^a ; 1), (\vec{y}' ; 1)) = pP^{sch}(\vec{y}^a , \vec{\tilde{y}}^a ) + 0.
\]
By Lemma 2 in [49] which states the transition probability with the always-available channel, we can prove Part (b).

Based on the state transition probabilities, we show that \( DTMC^a \) has product-form stationary distributions and give the specific forms in Proposition 6.3.2.

**Proposition 6.3.2.** A necessary and sufficient condition for the \( DTMC^a \) to be irreducible and aperiodic is \( \cup_{m^a \in M^a} m^a = \{1, \cdots , N\} \) and in this case the DTMC is reversible and has the following product-form stationary distribution: \( \pi(\vec{Y}^a) = \frac{1}{Z^a} \prod_{i \in y^a} \frac{p_i}{\tilde{p}_i}, Z^a = \sum_{y^a \in M^a} \prod_{i \in y^a} \frac{p_i}{\tilde{p}_i}. \)

Proof. The necessary and sufficient condition can be proven as in the proof of Proposition 1 in [49] which states the product-form stationary distribution of the DTMC with the transmission schedule as the network state in an alway-available channel network. We only need to check \( \pi(\vec{Y}^a) P(\vec{Y}^a, \vec{\tilde{Y}}^a) = \pi(\vec{\tilde{Y}}^a) P(\vec{\tilde{Y}}^a, \vec{Y}^a) \). It follows that the DTMC is reversible and has such stationary distribution. \( \square \)
In a similar way, we show that $DTMC^b$ has product-form stationary distributions in Proposition 6.3.4 based on Lemma 6.3.3.

**Lemma 6.3.3.** (a) A state $\bar{Y}^b = (y^b(t))$ can make a transition to a state $\bar{Y}^b = y'(t)$ ($\bar{y}^b \neq y'$) iff

$$y^b \cup \bar{y}^b \in M^b_0$$

and there exists a decision schedule $m^b \in M^b_0$ s.t.

$$y^b \triangle \bar{y}^b := (y^b \setminus \bar{y}^b) \cup (\bar{y}^b \setminus y^b) \subseteq m^b.$$

(b) The transition probability $P(\bar{Y}^b, \bar{Y}^b)$ from $\bar{Y}^b$ to $\bar{Y}^b$ ($\neq \bar{Y}^b$).

$$P(\bar{Y}^b, \bar{Y}^b) = \sum_{m^b \in M^b_0 : y^b \triangle \bar{y}^b \subseteq m^b} p\alpha^b(m^b) \left( \prod_{i \in y^b \setminus \bar{y}^b} p_i \right)$$

$$(\prod_{k \in y^b \setminus \bar{y}^b} p_k) \left( \prod_{i \in m^b \cap (y^b \setminus \bar{y}^b)} p_i \right) \left( \prod_{j \in m^b \setminus (y^b \cup \bar{y}^b) \cup C(y^b \cup \bar{y}^b)} \bar{p}_j \right),$$

where $C(y^b \cup \bar{y}^b)$ denotes the neighbors of nodes in $y^b \cup \bar{y}^b$.

**Proof.** Part (a) can be proven as Lemma 2 in [49]. To prove Part (b), we denote $P^{sch}(y^b, y')$ as $P(\bar{y}^b, \bar{y}')$ in Lemma 2 of [49] which is the transition probability from state $\bar{y}^b$ to state $\bar{y}'$ with the always-available channel. We only need to show

$$P((\bar{y}^b; 0), \bar{y}') = P((\bar{y}^b; 1), \bar{y}') = pP^{sch}(y^b, y').$$

The rest of the proof is similar to that of Lemma 6.3.1. \hfill \Box

**Proposition 6.3.4.** A necessary and sufficient condition for the $DTMC^b$ to be irreducible and aperiodic is $\cup_{m^b \in M^b_0} m^b = S$ and in this case the DTMC is reversible and has the following product-form stationary distribution: $\pi(\bar{Y}^b) = \frac{1}{Z^b} \prod_{i \in y^b} \frac{p_i}{\bar{p}_i}$, $Z^b = \sum_{y^b \in M^b_0} \prod_{i \in y^b} \frac{p_i}{\bar{p}_i}$.  

129
Proof. It is similar to the proof of Proposition 6.3.2 except that we need to check
\[ \pi(\bar{Y}^b)P(\bar{Y}^b, \bar{Y}^b) = \pi(\bar{Y}^b)P(\bar{Y}^b, \bar{Y}^b). \]

Based on the product-form distribution, we use the following results established

**Theorem 6.3.5.** [43] We define
\[ w^*(t) := \max_{x \in M(t)} \sum_{i \in x} w_i(t) \]
where \( M(t) \) is the set of all feasible schedules at time \( t \). For a scheduling algorithm, if given any \( \epsilon \) and \( \delta \), \( 0 < \epsilon, \delta < 1 \), there exists a \( \beta > 0 \) such that: if \( w^*(t) > \beta \), the scheduling algorithm chooses a schedule \( x(t) \in M(t) \) that satisfies

\[ P\{ \sum_{i \in x(t)} w_i(t) \geq (1 - \epsilon)w^*(t) \} \geq 1 - \delta, \quad (6.3.3) \]

where \( w_i(t) = f_i(q_i(t)) \) is a function of queue lengths satisfying the following conditions:

1. \( f_i(q_i(t)) \) is a nondecreasing, continuous function with \( \lim_{q_i \to \infty} f_i(q_i) = \infty \);

2. Given any \( a \in \mathbb{R} \), \( \lim_{q_i \to \infty} \frac{f_i(q_i + a)}{f_i(q_i)} = 1 \).

Remark: The throughput optimality result in Theorem 6.3.5 holds for any scheduler as long as the conditions are satisfied. It does not depend on how the scheduling algorithm is designed.

Then the scheduling algorithm is throughput-optimal.

We choose \( p_i = e^{w_i(t)} \) as long as \( w_i \) satisfies the conditions in [21]. By choosing \( f_i \) wisely, \( w_i(t) \) evolves slowly over \( t \). For instance, we choose \( f_i(q_i) = \log(\log(q_i + 1)) \)

---

\(^{3}\)The proof of the theorem in [43] can be easily extended to general conflict graph, and is applicable to our cognitive radio model. However, the authors in [43] can only find a scheduling algorithm that satisfies (6.3.3) in the fully-connected conflict graph while we propose a scheduling algorithm satisfying (6.3.3) for the general conflict graph under the cognitive radio framework.
in Section 6.4. We assume that the DTMC is in steady-state in every time slot throughout this chapter (time-scale separation) \[36][49].

**Proposition 6.3.6.** Suppose \( \cup_{m^a \in M^a} m^a = \{1, \cdots, N\} \) and \( \cup_{m^b \in M^b} m^b = S \). Let \( p_i = \frac{e^{w_i(t)}}{e^{w_i(t)} + 1}, \forall i \in \{1, \cdots, N\} \) when \( B(t) = 0 \) and \( p_i = \frac{e^{w_i(t)}}{e^{w_i(t)} + 1}, \forall i \in S \) when \( B(t) = 1, \) where \( w_i(t) = f_i(q_i(t)) \) is a function of queue length satisfying the conditions established in Theorem 6.3.5. Then Algorithm 9 is throughput optimal.

**Proof.** By Propositions 6.3.2 and 6.3.4, we know that both DTMCs have product-form stationary distributions. Given any \( \epsilon \) and \( \delta \) s.t. \( 0 < \epsilon, \delta < 1. \) For \( DTMC^a, \) we define \( w^a(t) = \max_{x \in M^a_0} \sum_{i \in x} w_i(t). \) Based on this, four sets of states are defined as follows:

\[
\chi^a_0 := \{(\vec{y}^a; 0) | y^a \in M^a_0, \sum_{i \in y^a} w_i(t) < (1 - \epsilon)w^a(t)\}
\]

\[
\chi^a_1 := \{(\vec{y}^a; 1) | y^a \in M^a_0, \sum_{i \in y^a} w_i(t) < (1 - \epsilon)w^a(t)\}
\]

\[
\phi^a := \chi^a_0 \cup \chi^a_1
\]

\[
\psi^a := \{(\vec{y}^a) | y^a \in M^a_0, \sum_{i \in y^a} w_i(t) < (1 - \epsilon)w^a(t)\}
\]

where \( \chi^a_0 \) includes all states with the channel available and the sum of \( w_i(t) \) from SUs chosen in the schedule is at least a fraction of \( \epsilon \) away from \( w^a(t), \) \( \chi^a_1 \) includes all states with the channel unavailable and the sum of \( w_i(t) \) from SUs chosen in the schedule of the most recently available slot is at least a fraction of \( \epsilon \) away from \( w^a(t). \)
Note that if \( (\vec{y}^a; B) \in \varphi^a \), then \( \bar{Y}^a = (\vec{y}^a) \in \psi^a \). We then calculate the probability of a state in set \( \chi_0^a \). We define \( \pi(A) := \text{prob (state } v \in A) \).

\[
\pi(\chi_0^a) = \pi(\psi^a) = \sum_{\vec{y}^a \in \psi^a} \pi(\bar{Y}^a) = \sum_{\vec{y}^a \in \psi^a} e^{\sum_{i \in \vec{y}^a} w_i(t)} Z^a \\
\leq \frac{|\psi^a| e^{(1-\epsilon)w^a(t)}}{Z^a} < \frac{2^N}{e^\epsilon w^a(t)}
\]

where

\[
Z^a = \sum_{y^a \in M_0^a} e^{\sum_{i \in y^a} w_i(t)} > e^{\sum_{i \in M_0^a} \sum_{i \in y^a} w_i(t)} = e^{w^a(t)}.
\]

The first equality holds because \( 1_{\{\vec{y}^a; 0\}} \cup \{\vec{y}^a; 1\} = 1_{\{\vec{y}^a\}} \); The last inequality is true because \( |\psi^a| \leq |M_0^a| \leq 2^N \). Thus, \( \exists \beta^a > 0 \), such that: \( w^a(t) > \beta^a \) implies \( \pi(\chi_0^a) < \delta \min (p, 1 - p) \).

For \( DTMC^b \), we define \( w^b(t) = \max_{x \in M_0^b} \sum_{i \in x} w_i(t) \). Similar to \( DTMC^a \), four sets of states are defined.

\[
\chi_0^b := \{(\vec{y}^b; 0) | y^b \in M_0^b, \sum_{i \in y^b} w_i(t) < (1 - \epsilon)w^b(t) \}
\]

\[
\chi_1^b := \{(\vec{y}^b; 1) | y^b \in M_0^b, \sum_{i \in y^b} w_i(t) < (1 - \epsilon)w^b(t) \}
\]

\[
\varphi^b := \chi_0^b \cup \chi_1^b
\]

\[
\psi^b := \{\bar{Y}^b = (\vec{y}^b) | y^b \in M_0^b, \sum_{i \in y^b} w_i(t) < (1 - \epsilon)w^b(t) \}
\]
We then calculate the probability of a state in set $\chi^b_1$.

$$\pi(\chi^b_1) = \pi(\Phi^b) = \pi(\psi^b) = \sum_{\vec{y}^b \in \psi^b} \frac{\pi(y^b(t))}{Z^b} \leq \frac{|\psi^b| e^{(1-\epsilon)w^b(t)}}{Z^b} < \frac{2|S|}{e^\epsilon w^b(t)}$$

where

$$Z^b = \sum_{y^b \in M_0^b} \sum_{i \in y^b} w_i(t) e^\max_{i \in y^b} \sum_{i \in y^b} w_i(t) = e^{w^b(t)}.$$

The first equality holds because $1_{\{(y^b_{0})\cup(y^b_{1})\}} = 1_{\{(y^b_{1})\}}$. The last inequality is true because $|\psi^b| \leq |M_0^b| \leq 2|S|$. Thus, $\exists \beta^b > 0$, such that: $w^b(t) > \beta^b$, which implies that $\pi(\chi^b_1) < \delta \min(p, 1-p)$.

Since $w^*(t) = w^a(t)$ when $B(t) = 0$, $w^*(t) = w^b(t)$ when $B(t) = 1$, we have the following results:

$$P\{ \sum_{i \in y^a(t)} w_i(t) \geq (1-\epsilon)w^a(t) | B(t) = 0 \} = 1 - \pi(\chi^b_0)/p > 1 - \delta \min(p, 1-p)/p \geq 1 - \delta$$

Equation (6.3.4) implies that: If $B(t) = 0$, $P\{ \sum_{i \in y^a(t)} w_i(t) \geq (1-\epsilon)w^a(t) \} > 1 - \delta$ if $w^a(t) > \beta$. Similarly,
when $w^b(t) > \beta^b$. Equation (6.3.5) implies that: If $B(t) = 1$, $P\{ \sum_{i \in y^b(t)} w_i(t) \geq (1 - \epsilon)w^*(t) | B(t) = 1 \} > 1 - \delta$ when $w^b(t) > \beta$.

We use the total probability formula to calculate the unconditional probability:

$$P\{ \sum_{i \in x(t)} w_i(t) \geq (1 - \epsilon)w^*(t) \}
\overset{(a)}{=} P\{ \sum_{i \in y^a(t)} w_i(t) \geq (1 - \epsilon)w^*(t) | B(t) = 0 \} P(B(t) = 0)
+ P\{ \sum_{i \in y^b(t)} w_i(t) \geq (1 - \epsilon)w^*(t) | B(t) = 1 \} P(B(t) = 1)
\overset{(b)}{\geq} (1 - \delta)p + (1 - \delta)(1 - p) = 1 - \delta$$

if $w^*(t) > \beta$ where $\beta = \max(\beta^a, \beta^b)$. Note that (a) holds because $y^a(t) = x(t)$ when $B(t) = 0$ and $y^b(t) = x(t)$ when $B(t) = 1$ by definition, and (b) holds due to Equations (6.3.4) and (6.3.5). Hence Algorithm 9 satisfies the condition of Theorem 6.3.5 and is throughput optimal. Although Algorithm 9 involves the evolutions of two Markov chains where a combination of the transmission schedule and the channel state is the system state, it is throughput optimal as long as the conditions in Theorem 6.3.5 are satisfied.
Proposition 6.3.6 shows that the full capacity region can be achieved by Algorithm 9 in CRNs with SUs inside and outside the interference range of the PU. Note that Lines 2 and 13 in Algorithm 9 can be implemented in a distributed manner through contention similar to [49] - the information exchange is kept locally. At time slot \( t \): 1) SU \( i \) selects a random number \( T_i \) uniformly in \([0, W - 1]\) and waits for \( T_i \) control mini-slots; 2) If SU \( i \) hears an INTENT message from a SU in \( C(i) \) before the \((T_i + 1)\)-th control mini-slot, \( i \) will not be included in \( m(t) \) and will give up the transmission of the INTENT message in this control slot; 3) If SU \( i \) does not hear an INTENT message from any SU in \( C(i) \) before the \((T_i + 1)\)-th control mini-slot, it will broadcast an INTENT message at the beginning of the \((T_i + 1)\)-th control mini-slot. If there is no collision, SU \( i \) will be included in \( m(t) \); or else, no SU is included in \( m(t) \). The overhead of the algorithm only includes notifications of channel state changes for SUs in \( S \) that could be done by SUs in \( \bar{S} \).

6.4 Simulations

In this section, we conduct simulations to compare the performance of 802.11 (we use a similar algorithm as in [49]) and Algorithm 9. Since the 802.11 algorithm we compare does not follow the exact designs of contention window sizes, sensing slots, transmission slots, etc, we call it “simple 802.11” in all the figures. In Algorithm 9, \( W \), the contention window size in the control slot, is chosen to be the number of SUs in the network. SU weights are chosen to be of the form \( \log(\log(q + e)) \), where \( q \) is the queue length [49] [58] [27]. We set \( p = 0.6 \), that is, 60% of the time, the PU is not using the channel.

In “Network 1”, there are 6 SUs whose conflict graph is shown in Figure 6.6. A conflict graph is one where two SUs are neighbors if they cannot transmit simultaneously. SUs 3 and 6 are outside the interference range of the PU and others are
inside. Let $\lambda = 0.2 \times (1,0,1,0,0,0) + 0.3 \times (1,0,0,1,0,1) + 0.2 \times (0,1,0,1,0,1) + 0.3 \times (0,0,1,0,1,0) = (0.5,0.2,0.5,0.3,0.5,0.3)$, which is a convex combination of some maximal independent sets. We vary $\rho$ from 0 to 0.9 $p$ so that $\rho \times \lambda$ lies inside the capacity region. For each algorithm, for a fixed $\rho$, we run 10 independent experiments and take the average. We show the average queue length of the network over $\rho$ in Figure 6.7 where the running time is $10^5$ time slots. As we can see, Algorithm 9 outperforms simple 802.11, which does not take into account the queue length information. In [49], a hybrid algorithm is developed to reduce the delay while maintaining the property of throughput optimality. A hybrid algorithm based on Algorithm 9 can be similarly designed to further improve the delay performance, which is not the focus of this chapter.

There are 16 SUs in “Network 2”, where the conflict graph is a 4 by 4 grid (Figure 6.8). Note that the sizes we choose in the simulations are comparable to those in [49] and [36]. SUs 1 through 11 are within the interference range of the PU, while SUs 12 to 16 are not. We compare metrics specific in cognitive radio networks for simple 802.11 and Algorithm 9 in “Network 2”. First, we define channel usage as the percentage that any of the SUs in the interference range of the PU is using the channel. In Figure 6.9, we vary the load factors from 0 to close to capacity region boundary and compare the channel usage percentages. Both algorithms fluctuate but there are no significant drops or jumps.

We compare channel usage over different PU traffic loads in Figure 6.10 where the SU traffic load is 90% of the capacity region (in the comparisons of all the following metrics, we use the same load factor if not specifically mentioned). There is an obvious gap between the two algorithms.

To see how long it takes the algorithms to saturate the network, we define a queue length threshold, above which the SU is saturated. This threshold is set to be
500/6 ÷ 83 in the 6-SU network. Note that 500 is the average number of packets in the network reached by Algorithm 9 as shown in Figure 6.7. We plot the time for queue length of each SU to reach this threshold using both simple 802.11 and Algorithm 9 in Figure 6.11. The simulation time is $10^5$ times slot. If the queue length does not reach 83 packets within the simulation time, we set the saturation time to be $10^5$. The saturation times at SUs under Algorithm 9 are much more balanced than in simple 802.11, which may also explain why Algorithm 9 stabilizes the system with arrivals that simple 802.11 cannot.

In Algorithm 9, channel state changes need to be notified to the SUs outside the interference range of the PU, which is an extra overhead and may cause errors as well. In Figure 6.12, we show the robustness of the network throughput to notification errors. We assume that the wrong notification in the previous slot will be corrected in the current slot, which means, SUs outside the interference range of the PU would know they got the wrong notification in the previous slot. This assumption allows the design of an efficient correction algorithm on the notification errors, which is also our future work. The network throughput degrades gracefully when the error increases in Figure 6.12.

6.5 Summary

In this chapter, we develop a throughput optimal distributed queue length based CSMA/CA scheduling algorithm for cognitive radio networks. The algorithm needs to signal a group of SUs during channel state change, different from existing distributed queue-length based CSMA/CA algorithms. Our algorithm is designed to adapt to the channel availability dynamics caused by unknown PU activity. The performance of our algorithm and a simplified 802.11 are compared in simulations to show the efficacy of the former.
Figure 6.5: Two DTMCs evolutions for SUs inside and outside the interference range of the PU. Two SUs are in the conflict set of each other. Both DTMCs are time reversible and have product-form stationary distribution (by Propositions 6.3.2 and 6.3.4). $\alpha_2^a$ is the probability SU 2 is chosen the decision schedule $m^a$ when the channel is available. $\alpha_2^b$ is the probability SU 2 is chosen in the decision schedule $m^b$ when the channel is unavailable. Note that $|m^a| = 2$ and $|m^b| = 1$ in this example.
Figure 6.6: Conflict graph with 6 SUs.

Figure 6.7: Queue lengths of two algorithms with different loads in the 6-SU network.
Figure 6.8: Conflict graph with 16 SUs.

Figure 6.9: Channel usage percentage over load factor in the 4 by 4 grid network.
Figure 6.10: Channel usage percentage over PU traffic in the 4 by 4 grid network.

Figure 6.11: Time for each SU to saturate in the 6-SU network.
Figure 6.12: Network throughput over the probability of false notification of channel state in the 6-SU network.
CHAPTER 7
CONCLUSION

Cognitive Radio Networks allow secondary users to opportunistically access the spectrum of the primary users without disrupting their transmission. To detect the PU transmission, SUs sense the licensed channels and find underutilized “white spaces”. Designing efficient sensing and spectrum allocation algorithms is crucial for properly operating a CRN. In this thesis, we try to answer this question. We first investigate system throughput maximization problem using cooperative sensing and then add constraints on PU throughput (Chapter 3) and sensing budget (Chapter 4), respectively. Then we study the social welfare maximization problem in operator based CRNs and design spectrum allocation algorithms to serve the SU requests of various levels of delay (Chapter 5). We also design a distributed throughput optimal scheduling algorithm in CRNs assuming the sensing results are accurate (Chapter 6). We summarize our contributions in Section 7.1.

7.1 Research Contributions

In Chapter 3, we investigated three different problem settings for maximizing the system throughput by cooperative sensing in CRNs. As the first step, we considered the problem of maximizing the weighted sum of the PU and SU throughput and developed a Bayesian rule based algorithm which is shown to be optimal. Then we
studied a constrained throughput maximization problem to guarantee a minimum PU throughput. We showed the NP-hardness of the problem and proposed an approximation algorithm with provable bound. We also established the monotonicity of the system throughput function, and proposed a simple greedy heuristic to the subset selection problem.

Whereas Chapter 3 solves the system throughput maximization problem in different settings, it does not consider the sensing budget of SUs. For instance, each SU can only sense a limited number of channels with various sensing capabilities, due to time or energy constraints. We took into account this constraint in Chapter 4, and showed this problem to be strongly NP-hard. We proposed matching based algorithm with a provable approximation ratio.

In Chapter 5, we studied the joint sensing and spectrum allocation problem for serving SUs with various levels of delay tolerance. We formulated a social welfare maximization problem and took into account both spectrum uncertainty and sensing inaccuracy. We developed an offline algorithms which leads to an optimal solution and an online algorithm that achieves comparable performance to the offline algorithm. We further extended the online solution to the auction framework, which guarantees incentive compatibility. We also provided a way for trading off social welfare and revenue.

In Chapter 6, we developed a throughput optimal distributed scheduling algorithm for CRNs. It is designed to adapt to the channel availability dynamics caused by unknown PU activity. We compared the performance of our algorithm with a simplified 802.11 in simulations.
7.2 Future Directions

In this section, we point out interesting problems that have not been solved and new areas that researchers can explore in the future. We still categorize them by the three functions in a typical duty cycle of a CRN.

**Spectrum sensing and analysis:** Specifically related to our work in Chapter 3, a greedy heuristic is proposed for the system throughput maximization problem with the number of SUs chosen for sensing restricted. However, the characterization of the combinatorial problem remains elusive. Moreover, investigating cases where the observations of SUs are correlated is also of interest. In Chapter 4, the ratio of the matching based algorithm only takes into account the diversity of SUs. However, the effect of cooperative sensing is not fully considered. With this effect, a tighter performance bound may be found. Also, the same problem with a constraint on the PU throughput is worth investigating. From a larger perspective, soft combining rules and other detection methods can be explored to achieve a certain goal such as maximizing sensing accuracy. Also, novel sensing methods may be come up with to overcome the shortcomings of existing solutions.

**Spectrum management and handoff:** The switching delay to other candidate channels may be further reduced by improving the hardware performance. In addition, channel reservation schemes may incorporate more intelligent information not limited to location, such as channel statistics. Learning techniques can also be applied. Moreover, hybrid metrics can be designed to measure the channel quality and then to assist in the decision making.

**Spectrum allocation and sharing:** In Chapter 5, there are several open problems to be solved. First, in practice, a more flexible form of spectrum requests will be desirable. For instance, a request may ask for multiple chunks that may or may not be preemptive. Extending the current solutions to this more general setting will be
a future direction. Second, we plan to extend the problem formulation by including
the notion of spatial spectrum reuse in addition to the time dimension considered in
the paper. Third, we plan to relax the assumption on the i.i.d Bernoulli channels by
considering correlated channels. The memory then becomes an important resource
to exploit for cost estimation. The operator, however, tends to maximize the overall
long term objective. It involves solving an *exploration vs. exploitation* problem in the
context of an auction. Fourth, an operator may purchase spectrum from the donor
operator to avoid sensing overhead and penalty payment, which leads to a different
formulation. Fifth, multiple operator competing for SUs can be modeled as a game.
This falls into the framework of non-cooperative spectrum sharing between different
secondary networks. In Chapter 6, the time scale separation assumption may be
relaxed. An extension to multi-hop and multi-PU CRNs is a future direction. Also,
the impact of sensing errors on algorithm design is of future interest. There are other
subareas where contributions have been made by others. They still have open prob-
lems including cognitive relaying, power control, control channel management, upper
layer issues, security, etc.
BIBLIOGRAPHY


147


[34] R. Balakrishnan I. F. Akyildiz, B. F. Lo. Cooperative spectrum sensing in

2011.

[36] Libin Jiang and Jean Walrand. A distributed csma algorithm for throughput
and utility maximization in wireless networks. IEEE/ACM Transactions on

[37] Changhee Joo, Xiaojun Lin, and N.B. Shroff. Understanding the capacity region
of the greedy maximal scheduling algorithm in multi-hop wireless networks. In
INFOCOM 2008. The 27th Conference on Computer Communications. IEEE,

[38] Gaurav S. Kasbekar and Saswati Sarkar. Spectrum pricing games with spatial
reuse in cognitive radio networks. IEEE Journal on Selected Areas in Commun-

New York, NY, USA, 2011.

[40] Seung-Jun Kim and Georgios B. Giannakis. Sequential and cooperative sensing
for multi-channel cognitive radios. IEEE Transactions on Signal Processing,

[41] M. Kudo. Comparison of algorithms that select features for pattern classifiers.

[42] Mathieu Leconte, Jian Ni, and Rayadurgam Srikant. Improved bounds on the
throughput efficiency of greedy maximal scheduling in wireless networks. In Pro-
cedings of the tenth ACM international symposium on Mobile ad hoc networking

[43] Bin Li and Atilla Eryilmaz. A fast-csma algorithm for deadline constraint
scheduling over wireless fading channels. In Workshop on Resource Allocation
and Cooperation in Wireless Networks (RAWNET), May 2011.

cooperative sensing in cognitive radio networks. In Proceedings of INFOCOM

CDC, dec 2012.


