SECURITY AND INTELLIGENCE MEASURE IN ONLINE MACHINE LEARNING-BASED DYNAMIC SPECTRUM SHARING NETWORKS

by

Monireh Dabaghchian
A Dissertation
Submitted to the
Graduate Faculty
of
George Mason University
In Partial fulfillment of
The Requirements for the Degree
of
Doctor of Philosophy
Electrical and Computer Engineering

Committee:

_____ Dr. Kai Zeng, Dissertation Director

_____ Dr. Zhi Tian, Committee Member

_____ Dr. Brian Mark, Committee Member

_____ Dr. Jie Xu, Committee Member

_____ Dr. Monson H. Hayes, Department Chair

_____ Dr. Kenneth Ball, Dean, Volgenau School of Engineering

Date: _____ Summer Semester 2019
George Mason University
Fairfax, VA

Security and Intelligence Measure in Online Machine Learning-based Dynamic Spectrum Sharing Networks

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at George Mason University

By

Monireh Dabaghchian
Master of Science
University of Tabriz, 2009
Bachelor of Science
University of Tabriz, 2006

Director: Dr. Kai Zeng, Professor
Department of Volgenau School of Engineering

Summer Semester 2019
George Mason University
Fairfax, VA

Dedication

To the love of my life, Amir, who has made this journey full of joy by his unconditional love, support, sacrifice, and courage to me. Words fail me to express my gratitude to him.

Acknowledgments

I would like to thank my dissertation advisor, Professor Kai Zeng, for all his support and patience with me throughout my PhD studies. I would like to express my gratitude to him for all his guides and insightful discussions we have had during this period. I feel blessed of having Professor Kai Zeng as my adviser as not only I have learned from him many technical material and how to become an independent researcher, but by observing his decent behavior during these years, he has helped me to grow morally, as well. He has always been there to listen to my concerns or any challenges I have been facing in research and in personal life and helped me find my path.

I would like to extend my appreciation to my dissertation committee members, Professor Zhi Tian, Professor Brian Mark, and Professor Jie Xu for their support and guides to me throughout this period and for their insightful comments on the technical material. I have learned many great lessons from their manners and interactions with the students.

There are many other faculty members whom I owe my success to; people who will be in my heart for ever: Professor Peter Auer, Professor Amarda Shehu, Professor Monson H. Hayes, Professor Kathleen E. Wage, Professor Andre Manitius, Professor Tyros Berry, Professor Jim Jones, and many others. These are the people who truly supported me during my PhD studies and they kept me from falling during tough times. I would like to express my gratitude to them.

I would also like to thank my colleagues, Hengrun Zhang, Long Jiao, Jie Tang, Ning Wang, and Pu Wang, for their support and friendship and for making our lab a pleasant place to do research.

Last but not least, I would like to thank all my family members: my beloved husband, Amir, my parents for all their sacrifices and courage to me, my twin sister and brother, Maryam and Saeed. My love to them gives me the courage and strength I need to work hard and look forward to a brighter future. Thank you all for being there for me.

I would also like to acknowledge that this work has been partially supported by the U.S. National Science Foundation under grant No. CNS-1502584 and CNS-1464487.

Table of Contents

List of Tables

List of Figures

Abstract

SECURITY AND INTELLIGENCE MEASURE IN ONLINE MACHINE LEARNING-BASED DYNAMIC SPECTRUM SHARING NETWORKS

Monireh Dabaghchian, PhD

George Mason University, 2019

Dissertation Director: Dr. Kai Zeng

Cognitive radio (CR) as a spectrum sharing network is considered as a key enabling technology for dynamic spectrum access to improve spectrum efficiency. This dissertation studies the two aspects of spectrum sharing networks: Security and intelligence capabilities. In the first part, we consider a primary user emulation (PUE) attacker that can send falsified primary user signals and prevent the secondary user from utilizing the available channel. The best attacking strategies that an attacker can apply have not been well studied. In this thesis, for the first time, we study optimal PUE attack strategies by formulating an online learning problem where the attacker needs to dynamically decide the attacking channel in each time slot based on its attacking experience. The challenge in our problem is that since the PUE attack happens in the spectrum sensing phase, the attacker cannot observe the reward on the attacked channel. To address this challenge, we utilize the attacker's observation capability. We propose online learning-based attacking strategies based on the attacker's observation capabilities. Through our analysis, we show that with no observation within the attacking slot, the attacker loses on the regret order, and with the observation of at least one channel, there is a significant improvement on the attacking performance.

Observation of multiple channels does not give additional benefit to the attacker (only a constant scaling) though it gives insight on the number of observations required to achieve the minimum constant factor. Our proposed algorithms are optimal in the sense that their regret upper bounds match their corresponding regret lower-bounds. We show consistency between simulation and analytical results under various system parameters. In the second part of the dissertation, we study the intelligence measure of these spectrum sharing devices. Although the CR concept was invented with the core idea of realizing "cognition", the research on measuring CR cognition capabilities and intelligence is largely open. Deriving the intelligence capabilities of CR not only can lead to the development of new CR technologies, but also makes it possible to better configure the networks by integrating CRs with different intelligence capabilities in a more cost-efficient way. In this work, for the first time, we propose a data-driven methodology to quantitatively analyze the intelligence factors of the CR with learning capabilities. The basic idea of our methodology is to run various tests on the CR in different spectrum environments under different settings and obtain various performance results on different metrics. Then we apply factor analysis on the performance results to identify and quantify the intelligence capabilities of the CR. More specifically, we present a case study consisting of so many different types of CRs. CRs are different in terms of learning-based dynamic spectrum access strategies, number of sensors, sensing accuracy, and processing speed. Based on our methodology, we analyze the intelligence capabilities of the CRs through extensive simulations. Four intelligence capabilities are identified for the CRs through our analysis, which comply with the nature of the tested algorithms.

Chapter 1: Introduction to Security in Cognitive Radio Networks

Spectrum sharing is a means to optimally and efficiently share the same wireless spectrum bandwidth between multiple categories of users. Spectrum sharing has emerged to address the growing demand to spectrum bandwidth among IoT devices such as Unmanned Aerial Vehicles (UAVs) [1], connected cars [2], smart phones, with both industrial and military applications.

There are two methods for spectrum sharing: tiered access and coexistance. The first method, tiered access, has emerged to address the spectrum shortage problem by efficiently utilizing the underutilized spectrum bandwidth assigned to licensed users. As we can see in Fig. 1.1 the spectrum bandwidth assigned to licensed users is not busy all the time and at some time intervals, some frequency bandwidths are idle. To address this problem, Federal Communications Commission (FCC) has authorized opening spectrum bands (e.g., 3550-3700 MHz and TV white space) owned by licensed primary users (PU) to unlicensed secondary users (SU) when the primary users are inactive [3, 4].

Cognitive radio (CR) is a key technology that enables secondary users to learn the spectrum environment and dynamically access the best available channel. Cognitive radios are assumed to be smart, and are designed to identify the underutilized spectrum assigned to the licensed users and conduct data transmission on those channels while their interference to the licensed/primary users are controlled. This type of spectrum sharing is used in the Citizens Broadband Radio Service (CBRS). According to the second method, coexistence, several categories of users share the same frequency spectrum bandwidth, simultaneously. An example of such spectrum sharing systems is Bluetooth and 2.4-GHz Wi-Fi.

In this work, we consider the tiered access based on which cognitive radios only access

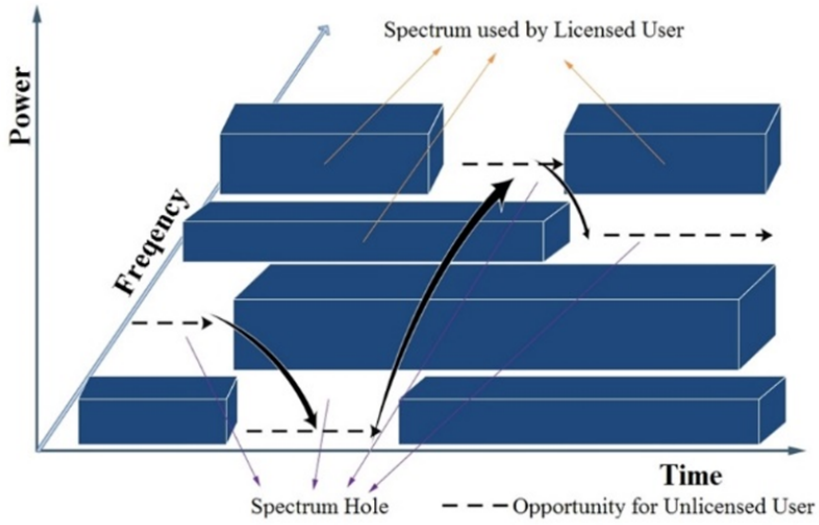


Figure 1.1: Spectrum bandwidth assigned to licensed users is not all the time busy on all the frequency channels.

the channel under controlled interference to primary users and tend to be more active when primary users are idle. Meanwhile, an attacker can send signals emulating the primary users to manipulate the spectrum environment, preventing a secondary user from utilizing the available channel. This attack is called primary user emulation (PUE) attack [5, 6, 7, 8, 9].

Existing works on PUE attacks mainly focus on PUE attack detection [10, 11] and defending strategies [7, 12]. However, there is a lack of study on the optimal PUE attacking strategies. Better understanding of the optimal attacking strategies will enable us to quantify the severeness or impact of a PUE attacker on the secondary user's throughput. It will also shed light on the design of defending strategies.

In practice, an attacker may not have any prior knowledge of the primary user activity characteristics or the secondary user dynamic spectrum access strategies. Therefore, it needs to learn the environment and attack at the same time. In this thesis, for the first time, we study the optimal PUE attacking strategies without any assumption on the prior knowledge of the primary user activity or secondary user accessing strategies. We formulate this problem as an online learning problem. We use the words *play*, *action taking*, and *attack* interchangeably throughout the manuscript.

Different from all the existing works on online learning based PUE attack defending strategies [7, 12], in our problem, an attacker cannot observe the reward on the attacked channel. Considering a time-slotted system, the PUE attack usually happens in the channel sensing period, in which a secondary user attempting to access a channel conducts spectrum sensing to decide the presence of a primary user. If a secondary user senses the attacked channel, it will believe the primary user is active so that it will not transmit the data in order to avoid interfering with the primary user. In this case, the PUE attack is effective since it disrupts the secondary user's communication and affects its knowledge of the spectrum availability. In the other case, if there is no secondary user attempting to access the attacked channel, the attacker makes no impact on the secondary user, so the attack is ineffective. However, the attacker cannot differentiate between the two cases when it launches a PUE attack on a channel because no secondary user will be observed on the attacked channel whether a secondary user has ever attempted to access the channel or not.

The key for the attacker to launch effective PUE attacks is to learn which channel or channels a secondary user is most likely to access. To do so, the attacker needs to make observations on the channels. As a result, the attacker's performance is dependent on its observation capability. We define the observation capability of the attacker as the number of the channels it can observe within the same time slot after launching the attack. We propose two attacking schemes based on the attacker's observation capability.

In the first scheme called *Attack-OR-Observe* (AORO), an attacker, if it attacks, cannot make any observation in the same time slot due to the short slot duration in highly dynamic systems [13] or when the channel switching time is long. We call this attacker an attacker with no observation capability within the same time slot (since no observation is possible if it attacks). To learn the most rewarding channel though, the attacker needs to dedicate some time slots for observation only without launching any attacks. On the other hand, an attacker could be able to attack a channel in the sensing phase and observe other channels in the data transmission phase within the same time slot if it can switch between channels fast enough. We call this attacking scheme, *Attack-But-Observe-Another* (ABOA).

For the AORO case, we propose an online learning algorithm called POLA -*Play or Observe Learning Algorithm*- to dynamically decide between attack and observation and then to choose a channel for the decided action, in a given time slot. Through theoretical analysis, we prove that POLA achieves a regret in the order of $\tilde{O}(\sqrt[3]{T^2})$ where T is the number of slots the CR network operates. The $\tilde{O}$ indicates there are some dependency on logarithmic factors, too. Its higher slope regret is due to the fact that it cannot attack (gain reward) and observe (learn) simultaneously in a given time slot. We show the optimality of our algorithm by matching its regret upper bound with its lower bound.

For the ABOA case, we propose EXP3-DO -*EXP3 with Deterministic Observation* and PROLA -*Play and Random Observe Learning Algorithm*- online learning algorithms to be applied by the PUE attacker. EXP3-DO deterministically chooses the attacking and observation channels dynamically and deterministically, respectively. EXP3-DO achieves a regret in the order of $\tilde{O}(\sqrt[3]{T^2})$. PROLA dynamically chooses the attacking and observing channels in each time slot. The PROLA learning algorithm's observation policy fits a specific group of graphs called time-varying partially observable feedback graphs [14]. It is derived in [14] that these feedback graphs lead to a regret in the order of $\tilde{O}(\sqrt[3]{T^2})$. However, our algorithm, PROLA, is based on a new theoretical framework and we prove its regret is in the order of $\tilde{O}(\sqrt{T})$. We then prove the regret lower bounds for the algorithms are in the order of $\tilde{\Omega}(\sqrt[3]{T^2})$ and $\tilde{\Omega}(\sqrt{T})$, respectively which match their upper bound and shows our algorithm's optimality.

All algorithms proposed address the attacker's observation capabilities and can be applied as optimal PUE attacking strategies without any prior knowledge of the primary user activity and secondary user access strategies.

We further generalize PROLA to multi-channel observations where an attacker can observe multiple channels within the same time slot. By analyzing its regret, we come to the conclusion that increasing the observation capability from one to multiple channels does not give additional benefit to the attacker in terms of regret order. It only improves the

constant factor of the regret.

Our main contributions are summarized as follows:

- We formulate the PUE attack as an online learning problem without any assumption on the prior knowledge of either primary user activity characteristics or secondary user dynamic channel access strategies.

- We propose two attacking schemes, AORO and ABOA, that model the behavior of a PUE attacker. For the AORO case, a PUE attacker, in a given time slot, dynamically decides either to attack or observe; then chooses a channel for the decision it made. While in the ABOA case, the PUE attacker dynamically chooses one channel to attack and chooses at least one other channel to observe both deterministically and dynamically within the same time slot.

- We propose an online learning algorithm POLA for the AORO case. POLA achieves an optimal learning regret in the order of $\tilde{\Theta}(\sqrt[3]{T^2})$. We show its optimality by matching its regret lower and upper bounds.

- For the ABOA case, we propose two online learning algorithms: EXP3-DO to choose the attacking and observing channels dynamically and deterministically respectively, and it achieves an optimal regret in the order of $\tilde{\Theta}(\sqrt[3]{T^2})$. Another proposed algorithm, PROLA, to dynamically decide the attacking and observing channels. We prove that PROLA achieves an optimal regret order of $\tilde{\Theta}(\sqrt{T})$ by deriving its regret upper bound and lower bound. This shows that:

  1) with a carefully designed observation capability of at least one within the attacking slot, there is a significant improvement on the performance of the attacker.

  2) Theoretical contribution: For PROLA, despite observing the actions partially, it achieves an optimal regret order of $\tilde{\Theta}(\sqrt{T})$ which is better than a known bound of $\tilde{\Theta}(\sqrt[3]{T^2})$. We accomplish it by proposing randomized time-variable feedback graphs.

- The algorithm and the analysis of the PROLA are further generalized to multi-channel observations.
- We conduct simulations to evaluate the performance of the proposed algorithms under various system parameters.

Through theoretical analysis and simulations under various system parameters, we have the following findings:

- With no observation at all within the attacking slot, the attacker loses on the regret order. While with the observation of at least one channel if designed optimally, there is a significant improvement on the attacking performance.
- Observation of multiple channels in the PROLA algorithm does not give additional benefit to the attacker in terms of regret order. The regret is proportional to $\sqrt{1/m}$, where m is the number of channels observed by the attacker. Based on this relation, the regret is a monotonically decreasing and a convex function of the number of observing channels. As more observing channels are added, the reduction in the regret becomes marginal. Therefore, a relatively small number of observing channels is sufficient to approach a small constant factor.
- The attacker's regret is proportional to $\sqrt{K \ln K}$, where K is the total number of channels.

1.1 Related Work to CRN Security and Online Learning

1.1.1 PUE Attacks in Cognitive Radio Networks

Primary User Emulation (PUE) attack is one of the unique attacks against cognitive radio networks. In a PUE attack, a secondary user tries to emulate the primary user's signal and pretend to be a primary user. The goal of such an attacker is either selfish to prevent other secondary users from accessing the channel, so it can itself gain exclusive access to the channels or to generate a Denial of Service (DoS) attack towards the secondary users.

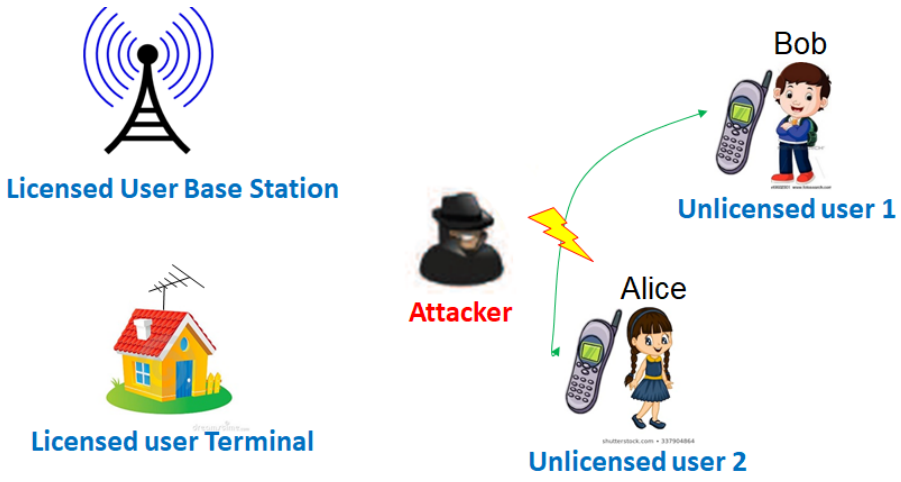


Figure 1.2: A PUE attacker in a cognitive radio network trying to disrupt the communication between secondary users, Alice and Bob.

Figure 1.2 represents a PUE attacker in the cognitive radio system. In order to have an effective PUE attack, the attacker has to have knowledge on the spectrum sensing technique applied by the secondary users, so it can emulate those features of the primary signal. As an example, if the SUs apply energy detectors, the attacker should create signals with the same power of the PUs. However, if the SUs are applying a feature-based detectors, then the attacker has to emulate the associated features to the PU.

Existing work on PUE attacks mainly focus on PUE attack detection [10, 11] and defending strategies [7, 12]. There are few works discussing attacking strategies under dynamic spectrum access scenarios. In [12], the attacker applies a partially observable Markov decision process (POMDP) framework to find the attacking channel in each time slot. It is assumed that the attacker can observe the reward on the attacking channel. That is, the attacker knows if a secondary user is ever trying to access a channel or not. In [7], it is assumed that the attacker is always aware of the best channel to attack. However, there is no methodology proposed or proved on how the best attacking channel can be decided.

The optimal PUE attack strategy without any prior knowledge of the primary user activity and secondary user access strategies is not well understood. In this thesis, we fill this gap by formulating this problem as an online learning problem. Our problem is also

unique in that the attacker cannot observe the reward on the attacking channel due to the nature of PUE attack.

1.1.2 Multi-armed Bandit Problems

There is a rich literature about online learning algorithms. The most related ones to our work are multi-armed bandit (MAB) problems [15, 16, 17, 18, 19]. The MAB problems have many applications in cognitive radio networks with learning capabilities [7, 20, 21, 22]. In such problems, an agent plays a machine repeatedly and obtains a reward when it takes a certain action at each time. Any time when choosing an action the agent faces a dilemma of whether to take the best rewarding action known so far or to try other actions to find even better ones. Trying to learn and optimize his actions, the agent needs to trade off between exploration and exploitation. On one hand the agent needs to explore all the actions often enough to learn which is the most rewarding one and on the other hand he needs to exploit the believed best rewarding action to minimize his overall regret.

For most existing MAB frameworks as explained, the agent needs to observe the reward on the taken action. Therefore, these frameworks cannot be directly applied to our problem where a PUE attacker cannot observe the reward on the attacking channel. Most recently, Alon et al. generalize the cases of MAB problems into feedback graphs [14]. In this framework, they study the online learning with side observations. They show in their work that if an agent takes an action without observing its reward, but observes the reward of all the other actions, it can achieve an optimal regret in the order of $\tilde{O}(\sqrt{T})$. However, the agent can only achieve a regret of $\tilde{O}(\sqrt[3]{T^2})$ if it cannot observe the rewards on all the other actions simultaneously. In other words, even one missing edge in the feedback graph leads to the regret of $\tilde{O}(\sqrt[3]{T^2})$.

In this thesis, we propose two novel online learning policies. The first one, POLA, for the case when only either observation or attack is possible within a time slot, is shown to achieve a regret in the order of $\tilde{O}(\sqrt[3]{T^2})$. *We then advance this theoretical study by proposing a strategy, PROLA, which is suitable for an attacker (learning agent) with higher*

observation capabilities within the acting time step. We prove that PROLA achieves an optimal regret in the order of $\tilde{O}(\sqrt{T})$ without observing the rewards on all the channels other than the attacking one simultaneously. In PROLA, the attacker uniformly randomly selects at least one channel other than the attacking one to observe in each time slot. Our framework is called randomized time-variable feedback graph.

1.1.3 Jamming Attacks

There are several works formulating jamming attacks and anti-jamming strategies as online learning problems [21, 23, 24, 25]. In jamming attacks, an attacker can usually observe the reward on the attacking channel where an ongoing communication between legitimate users can be detected. Also it is possible for the defenders to learn whether they are defending against a jammer or a PUE attacker by observing the reward on the accessed channel. PUE attacks are different in that the attacker attacks the channel sensing phase and prevents a secondary user from utilizing an available channel. As a result, a PUE attacker cannot observe the instantaneous reward on the attacked channel. That is, it cannot decide if an attack is effective or not.

1.1.4 Adaptive Opponents

Wang et al. study the outcome/performance of two adaptive opponents when each of the agents applies a no regret learning-based access strategy [21] . One agent tries to catch the other on a channel and the other one tries to evade it. Both learning algorithms applied by the opponents are no regret algorithms and are designed for oblivious environments. In other words, each of the learning algorithms is designed for benign environments and non of them assumes an adaptive/learning-based opponent. It is shown in this work, that both opponents applying a no-regret learning-based algorithm, reach an equilibrium which is in fact a Nash equilibrium in that case. In other words, despite the fact that the learning-based algorithms applied are originally proposed for oblivious environments, the two agents applying these algorithms act reasonably well to achieve an equilibrium. Motivated by

this work, in our simulations, we have considered a learning-based secondary user. More specifically, even though we proposed learning-based algorithms for oblivious environments, in the simulations, we evaluate its performance in an adaptive environment. The simulation results show the rationality of the attacker that even against an adaptive opponent (learning-based cognitive radio), it performs well and achieves a regret below the derived upper bound.

Chapter 2: System Model and Problem Formulation

We consider a cognitive radio network consisting of several primary users, multiple secondary users, and one attacker. There are K ($K > 1$) channels in the network. We assume the system is operated in a time-slotted fashion.

2.1 Primary User

In each time slot, each primary user is either active (on) or inactive (off). We assume the on-off sequence of PUs on the channels is unknown to the attacker a priori. In other words, the PU activity can follow any distribution or can even be arbitrary.

2.2 Secondary User

The secondary users may apply any dynamic spectrum access policy [26, 27, 28]. In each time slot, each SU conducts spectrum sensing, data transmission, and learning in three consecutive phases as shown in Fig. 2.1(a).

At the beginning of each time slot, each secondary user senses a channel it attempts to access. If it finds the channel idle (i.e., the primary user is inactive), then accesses this channel; otherwise, it remains silent till the end of the current slot in order to avoid interference to the primary user. At the end of the slot, it applies a learning algorithm to decide which channel it will attempt to access in the next time slot based on its past channel access experiences.

We assume the secondary users cannot differentiate the attacking signal from the genuine primary user signal. That is, in a time slot, when the attacker launches a PUE attack on the channel a secondary user attempts to access, the secondary user will not transmit any data on that channel.

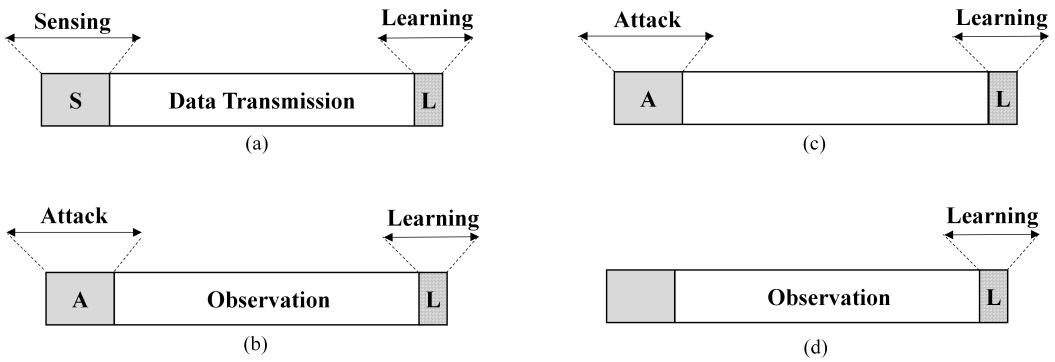


Figure 2.1: Time slot structure of a) an SU and b,c,d) a PUE attacker.

2.3 Attacker

We assume a smart PUE attacker with learning capability. We do not consider attack on the PUs. The attacker may have different observation capabilities. Observation capability is defined as the number of channels the attacker can observe after the attack period within the same time-slot. Observation capabilities of the attacker are impacted by the overall number of it's antennas, time slot duration, switching time, etc.

Within a time-slot, an attacker with at least one observation capability conducts three different operations. First in the attacking phase, the attacker launches the PUE attack by sending signals emulating the primary user's signals [12] to attack the SUs. Note that, in this phase, the attacker has no idea if its attack is effective or not. That is, it does not know if a secondary user is ever trying to access the attacking channel or not. In the observation phase, the attacker is supposed to observe the communication on at least one other channel. The attacker can even observe the attacked channel in the observation phase, however, in that case it will sense nothing since it has attacked the channel by emulating the PU signal and scared away any potential SU attempting to access the channel. So from learning point of view, observing the attacked channel does not provide any useful information on the SUs' activity. Therefore, it only gets useful information when observing channels other than the attacked one. Observing a different channel, the attacker may detect a primary

user signal, a secondary user signal, or nothing on the observing channel. The attacker applies its past observations in the learning phase to optimize its future attacks. At the end of the learning period, it decides which channels it attempts to attack and observe in the next slot. Figure 2.1(b) shows the time slot structure for an attacker with at least one observation capability.

If the attacker has no observation capability within the attacking time slot, at the end of each time slot it still applies a learning strategy based on which, it decides whether to dedicate the next time slot for attack or observation, then chooses a channel for either of them. Figure 2.1(c,d) show the time slot structure for an attacker with no observation capability in the attacking time slot. As shown in these two figures, the attacker conducts either attack or observation at each time slot.

2.4 Problem Formulation

Since the attacker needs to learn and attack at the same time and it has no prior knowledge of the primary user activity or secondary user access strategies, we formulate this problem as an online learning problem.

We consider T as the total number of time slots the network operates. We define $x_t(j)$ as the attacker's reward on channel j at time slot t ($1 \leq j \leq K$, $1 \leq t \leq T$). Without loss of generality, we normalize $x_t(j) \in [0, 1]$. More specifically:

$$x_t(j) = \begin{cases} 1, & \textit{SU is on channel } j \textit{ at time } t \\ 0, & \textit{o.w.} \end{cases}. \tag{2.1}$$

Suppose the attacker applies a learning policy φ to select the attacking and observing channels. The aggregated expected reward of attacks by time slot T is equal to

$$G_\varphi(T) = \mathbf{E}_\varphi \left[\sum_{t=1}^{T} x_t(I_t) \right], \tag{2.2}$$

Table 2.1: Main Notation

T	total number of time slots
K	total number of channels
I_t	index of the channel to be attacked at time t
J_t	index of the channel to be observed at time t
R	total regret of the attacker in Algorithm 1
γ	exploration rate
η	learning rate
$p_t(i)$	attack distribution on channels at time t
$q_t(i)$	observation distribution on channels at time t in Algorithm 1
$\omega_t(i)$	weight assigned to channel i at time t
δ_t	observation probability at time t in Algorithm 1

where I_t indicates the channel chosen at time t to be attacked. The attacker's goal is to maximize the expected value of the aggregated attacking reward, thus to minimize the throughput of the secondary user,

$$\text{maximize} \quad G_{\varphi}(T). \tag{2.3}$$

For a learning algorithm, regret is commonly used to measure its performance. The regret of the attacker can be defined as follows

$$Regret = G_{max} - G_{\varphi}(T), \tag{2.4}$$

where

$$G_{max} = \max_{j} \sum_{t=1}^{T} x_t(j). \tag{2.5}$$

The regret measures the gap between the accumulated reward achieved applying a learning algorithm and the maximum accumulated reward the attacker can obtain when it keeps attacking the single best channel. Single best channel is the channel with highest accumulated reward up to time T [17]. Then the problem can be transformed to minimize the

regret

$$\text{minimize} \quad G_{max} - G_{\varphi}(T) . \tag{2.6}$$

Table I summarizes the main notation used.

Chapter 3: Online Learning-Based Attacking Strategy

In this section, we propose three novel online learning algorithms for the attacker. These algorithms do not require the attacker to observe the reward on the attacked channel. Moreover, our algorithms can be applied in any other application with the same requirements.

The first algorithm proposed, POLA, is suitable for an attacker with no observation capability in the attacking time slot . For this attacker, either attack or observation is feasible within each time slot. Based on this learning strategy, the attacker decides at each time slot whether to attack or observe and chooses a channel for either of them, dynamically. The assumption here is that, any time the attacker decides to make observation, it observes only one channel since time slot duration has been considered short or switching costs are high compared to the time slot duration.

The second algorithm proposed is called EXP3-DO. EXP3-DO is a non-stochastic online learning algorithm for the attacker to decide which channel to attack and observe in each time slot. The algorithm does not require the attacker to observe the reward on the attacked channel, but assuming the attacker can observe the reward on one other channel.

The third algorithm, PROLA, is proposed for an attacker with at least one observation capability. Based on this learning policy, at each time, the attacker chooses channels dynamically for both attack and observation. In the following, we assume the attacker's observation capability is one. We generalize it to multiple channel observation capability in Section 3.4.

All algorithms are considered as no-regret online learning algorithms in the sense that the incremental regret between two consecutive time slots diminishes to zero as time goes to infinity. The first one, POLA, achieves an optimal regret in the order of $\tilde{O}(\sqrt[3]{T^2})$. This higher slope arises from the fact that at each time slot, the attacker gains only some reward by attacking a channel without updating its learning or it learns by making observation without

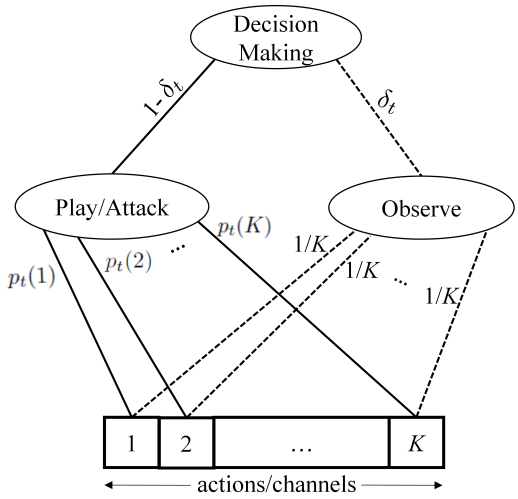


Figure 3.1: Decision Making process based on the POLA strategy: First the agent decides between play and observe, then for either of them chooses a channel dynamically.

being able to gain any reward. The EXP3-DO Algorithm, achieves a regret in the order of $O(T^{\frac{2}{3}})$. PROLA, is also optimal with the regret proved in the order of $\tilde{O}(\sqrt{T})$. Comparing these algorithms and the generalization of the latter together shows that, changing from no observation capability in the same time slot to one observation capability results in a significant improvement in the regret order; this is in comparison to changing from one observation capability to multiple observation capability which does not give any benefits in terms of regret order. However, multiple channel observation capability, provides insight to find the appropriate number of observations required to achieve the minimum constant factor in the regret upper bound.

3.1 Attacking Strategy 1: POLA Learning Algorithm

POLA is an online learning algorithm for the learning of an agent with no observation capability within the attacking time slot. In other words, the agent cannot play and observe simultaneously. If modeled properly as a feedback graph as is shown in Appendix A, this problem can be solved based on the EXP3.G algorithm presented in [14]. However, our

proposed learning algorithm, POLA, leads to a smaller regret constant compared to the one in [14] and it is much easier to understand. Based on the POLA attacking strategy, the attacker at each time slot decides between attacking and observation, dynamically. If it decides to attack, it applies an exponential weighting distribution based on the previous observations it has made. It chooses a channel uniformly at random if it decides to make an observation. Figure 3.1 represents the decision making structure of POLA. The proposed algorithm is presented in Algorithm 1.

Since the attacker is not able to attack and observe within the same time slot simultaneously, it needs to trade off between attacking and observation cycles. From one side the attacker needs to make observations to learn the most rewarding channel to attack. From the other side, it needs to attack often enough to minimize his regret. So, δ_t which represents the trade off between attack and observation needs to be chosen carefully.

In step 2 of Algorithm 1, $\hat{x}_t(j)$ represents an unbiased estimate of the reward $x_t(j)$. In order to derive $\hat{x}_t(j)$, we divide the $x_t(j)$ by the probability this channel is chosen to be observed which is equal to δ_t/K. The term δ_t/K indicates that in order for each channel to be chosen to be observed, first the algorithm needs to decide to make observation with probability δ_t and then to choose that channel for observation with probability $1/K$.

For any agent that acts based on the Algorithm 1, the following theorem holds.

Theorem 1. *For any $K \geq 2$ and for any $\eta \leq \sqrt[3]{\frac{\ln K}{K^2 T}}$, the upper bound on the expected regret of Algorithm 1*

$$G_{max} - \mathbf{E}\left[G_{POLA}\right] \leq (e-2)\eta K \left(\frac{3}{4}\sqrt[3]{\frac{(T+1)^4}{K \ln K}} + \frac{K \ln K}{4} \right) + \frac{\ln K}{\eta} + \frac{3}{2}\sqrt[3]{KT^2 \ln K} \quad (3.1)$$

holds for any assignment of rewards for any $T > 0$.

Algorithm 1: POLA, Play or Observe Learning Algorithm

Parameter: $\eta \in \left(0, \sqrt[3]{\frac{\ln K}{K^2 T}}\right]$.

Initialization: $\omega_1(i) = 1, \quad i = 1, ..., K$.

For each $t = 1, 2, ..., T$

1. Set $\delta_t = \min\left\{1, \sqrt[3]{\frac{K \ln K}{t}}\right\}$.

   Observe with probability δ_t and go to step 2.

   Attack with probability $1 - \delta_t$ and go to step 3.

2. Set

$$q_t(i) = \frac{1}{K}, \quad i = 1, ..., K$$

   Choose $J_t \sim \ q_t$ and observe the reward $x_t(J_t)$.

   For $j = 1, 2, ..., K$

$$\hat{x}_t(j) = \begin{cases} \frac{x_t(j)}{\delta_t(1/K)}, & j = J_t \\ 0, & o.w., \end{cases}$$

$$\omega_{t+1}(j) = \omega_t(j) \exp(\eta \hat{x}_t(j)),$$

   Go back to step 1.

3. Set

$$p_t(i) = \frac{\omega_t(i)}{\sum_{j=1}^{K} \omega_t(j)}, \quad i = 1, ..., K$$

   Attack channel $I_t \sim \ p_t$ and accumulate the unobservable reward $x_t(I_t)$.

   For $j = 1, 2, ..., K$

$$\hat{x}_t(j) = 0, \quad \omega_{t+1}(j) = \omega_t(j),$$

   Go back to step 1.

Proof of Theorem 1. The regret at time t is a random variable equal to,

$$r(t) = \begin{cases} x_t(j^*) - \mathbf{E}_{POLA,A}\left[x_t(I_t)\right] & \text{, Attack} \\ x_t(j^*) - \mathbf{E}_{POLA,O}\left[x_t(I_t)\right] = x_t(j^*) & \text{, Observe} \end{cases}$$

where j^* is the index of the best channel. The expected value of regret at time t is equal to

$$\mathbf{E}\left[r(t)\right] = (1-\delta_t)\left(x_t(j^*) - \mathbf{E}_{POLA,A}\left[x_t(I_t)\right]\right) + \delta_t\, x_t(j^*), \tag{3.2}$$

where the expectation is w.r.t. to the randomness in the attacker's attack policy. Expected value of accumulated regret, R, is

$$\mathbf{E}\left[R\right] = \mathbf{E}\left[\sum_{t=1}^{T} r(t)\right] \leq \sum_{t=1}^{T} x_t(j^*) - \sum_{t=1}^{T} \mathbf{E}_{POLA,A}\left[x_t(I_t)\right] + \sum_{t=1}^{T} \delta_t. \tag{3.3}$$

The inequality results from the fact that $\delta_t \geq 0$ and $x_t(j^*) \leq 1$. It is also assumed that $x_t(i) \leq x_t(j^*)$ for all i. The regret in equation (3.3) consists of two parts. The first one consists of the first two terms in this equation and it arises as a result of the attacker not attacking the most rewarding channel all the time but attacking some other low rewarding channels. The second part which consists of the last term in the equation is due to the observations made by the attacker in which it gains no reward. We derive an upper bound on each part separately, then add them together.

δ_t plays the key role in minimizing the regret. First of all, δ_t needs to be decaying since otherwise it leads to a linear growth of regret. So the key idea in designing a no-regret algorithm is to choose an appropriate decaying function for δ_t.

From one side, slowly decaying δ_t is desired from learning point of view; however, it

results in a larger value of regret since staying more in the observation phase precludes the attacker from launching attacks and gaining rewards. On the other hand, if δ_t decays too fast, the attacker is very likely to settle in a wrong channel since it does not have enough time to learn the most rewarding channel. We choose $\delta_t = \min\left\{1, \sqrt[3]{\frac{K \ln K}{t}}\right\}$ and our analysis shows the optimality of this function in minimizing the regret.

Below is the derivation of the upper bound for the first term of regret in equation (3.3). For a single t

$$\begin{aligned}
\frac{W_{t+1}}{W_t} &= \sum_{i=1}^{K} \frac{\omega_t(i)}{W_t} \exp(\eta \hat{x}_t(i)) \\
&\leq \sum_{i=1}^{K} p_t(i)[1 + \eta \hat{x}_t(i) + (e-2)\eta^2 \hat{x}_t^2(i)] \\
&\leq \exp\left(\eta \sum_{i=1}^{K} p_t(i)\hat{x}_t(i) + (e-2)\eta^2 \sum_{i=1}^{K} p_t(i)\hat{x}_t^2(i)\right), \qquad (3.4)
\end{aligned}$$

where the equality follows from the definition of $W_{t+1} = \sum_{i=1}^{K} \omega_{t+1}(i)$ and $\omega_{t+1}(i)$ in Algorithm 1. Also the last inequality follows from the fact that $e^x \geq 1 + x$. Finally, the first inequality holds by definition of $p_t(i)$ in Algorithm 1 and since $e^x \leq 1 + x + (e-2)x^2$ for $x \leq 1$. In this case, we need $\eta \hat{x}_t(i) \leq 1$. Based on our algorithm, $\eta \hat{x}_t(i) = 0$ if $i \neq J_t$ and for $i = J_t$ we have $\eta \hat{x}_t(i) = \eta \frac{x_t(i)}{\delta_t \frac{1}{K}} \leq \eta K \sqrt[3]{\frac{T}{K \ln K}}$, since $x_t(i) \leq 1$ and $\delta_t \geq \sqrt[3]{\frac{K \ln K}{T}}$ for $T \geq K \ln K$. This is equivalent to $\eta \leq \frac{1}{K \sqrt[3]{\frac{T}{K \ln K}}} = \sqrt[3]{\frac{\ln K}{K^2 T}}$.

By taking the ln and summing over $t = 1$ to T on both sides of equation (3.4), the left hand side (LHS) of the equation will be equal to

$$\sum_{t=1}^{T} \ln \frac{W_{t+1}}{W_t} = \ln \frac{W_{T+1}}{W_1} \geq \ln \omega_{T+1}(j) - \ln K = \eta \sum_{t=1}^{T} \hat{x}_t(j) - \ln K. \tag{3.5}$$

By combining (3.5) with (3.4),

$$\eta \sum_{t=1}^{T} \hat{x}_t(j) - \ln K \leq \eta \sum_{t=1}^{T} \sum_{i=1}^{K} p_t(i)\hat{x}_t(i) + (e-2)\eta^2 \sum_{t=1}^{T} \sum_{i=1}^{K} p_t(i)\hat{x}_t^2(i). \tag{3.6}$$

We take the expectation w.r.t. the randomness in $\hat{x}$, substitute j by j^* since j can be any of the actions, use the definition of $E_{POLA,A}[x_t(I_t)]$, and with a little simplification and rearranging the equation we get the following,

$$\sum_{t=1}^{T} x_t(j^*) - \sum_{t=1}^{T} E_{POLA,A}\left[x_t(I_t)\right] \leq (e-2)\eta K \sum_{t=1}^{T} \frac{1}{\delta_t} + \frac{\ln K}{\eta}. \tag{3.7}$$

The upper bound on $\sum_{t=1}^{T} \frac{1}{\delta_t}$ is equal to $\frac{3}{4}\sqrt[3]{\frac{(T+1)^4}{K \ln K}} + \frac{K \ln K}{4}$ which gives us,

$$\begin{aligned} &\sum_{t=1}^{T} x_t(j^*) - \sum_{t=1}^{T} \mathbf{E}_{POLA,A}\left[x_t(I_t)\right] \\ &\qquad \leq \frac{\ln K}{\eta} + (e-2)\eta K \left(\frac{3}{4}\sqrt[3]{\frac{(T+1)^4}{K \ln K}} + \frac{K \ln K}{4} \right). \end{aligned} \tag{3.8}$$

The upper bound on the second term of equation (3.3) is

$$\sum_{t=1}^{T} \delta_t = \sum_{t=1}^{T} \min \left\{ 1, \sqrt[3]{\frac{K \ln K}{t}} \right\} \leq \frac{3}{2}\sqrt[3]{K T^2 \ln K}. \tag{3.9}$$

Summing up the equation (3.8) and equation (3.9) gives us the regret upper bound. □

By choosing appropriate values for η, the above upper bound on the regret can be minimized.

Corollary 1. *For any $T > 2.577K \ln K$, we consider the input parameter*

$$\eta = \sqrt{\frac{\ln K}{(e-2)K\left(\frac{3}{4}\sqrt[3]{\frac{(T+1)^4}{K\ln K}} + \frac{K\ln K}{4}\right)}}.$$

Then

$$G_{max} - \mathbf{E}[G_{POLA}] \leq (\sqrt{3(e-2)} + \frac{3}{2})\sqrt[3]{T^2 K \ln K}$$

holds for any arbitrary assignment of rewards.

Proof of Corollary 1. We sketch the proof as follows. By getting the derivative from the statement in Theorem 1 with respect to η, we find the optimal value for η. Since $\eta \leq \sqrt[3]{\frac{\ln K}{K^2 T}}$, in order for the regret bound to hold, we need $T \geq \frac{8}{3\sqrt{3(e-2)^3}} K \ln K = 2.577 K \ln K$. By plugging in the value of η and some simplifications the regret bound in the corollary is achieved. □

Next is a theorem on the regret lower bound for this problem under which attacking and observation are not possible simultaneously.

Theorem 2. *For $K \geq 2$ and for any player strategy A, the expected weak regret of algorithm A is lower bounded by*

$$G_{max} - \mathbf{E}\left[G_A\right] \geq v\sqrt[3]{KT^2}$$

for some small constant v and it holds for some assignment of rewards for any $T > 0$.

Proof of Theorem 2. The proof follows from the lower bound analysis in [14]. The construction is given in the Appendix. □

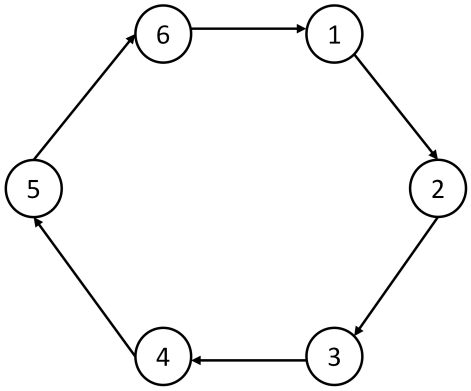


Figure 3.2: Channel observation strategy Deterministic, $K = 6$.

Based on Theorems 1 and 2, the algorithm's regret upper bound matches its regret lower bound which indicates POLA is an optimal online learning algorithm.

3.2 Attacking Strategy 2: EXP3-DO Learning Algorithm

In this algorithm, the attacking channel selection is based on the accumulated reward distribution on all the channels. While the observing channel is deterministically dependent on the attacking channel. The attacker always observes the channel next to the attacking one. It rounds to channel 1 when it attacks channel K. So we call this algorithm EXP3-DO (EXP3 with deterministic observation). This is in comparison to EXP3 [17] in which the rewards are observed on the same chosen action. Fig. 3.2 shows the observation strategy employed by the attacker when $K = 6$. The Algorithm shows the online learning-based attack strategy employed by the attacker.

The design of the EXP3-DO is motivated by [14] in which the idea of bandit graphs is presented. A bandit graph is a generalization of EXP3 algorithm in which the actions and the following observations from the chosen action are presented by the nodes and the edges of a graph, respectively.

Algorithm 2: EXP3-DO, EXP3 with Deterministic Observation

Parameters: $\gamma \in [0, e-2]$,

$\eta \in (0, \gamma/K]$

Initialization: $\omega_1(i) = 1, \quad i = 1, ..., K$

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

1. Set $p_t(i) = (1-\gamma)\frac{\omega_t(i)}{\sum_{j=1}^{K} \omega_t(j)} + \frac{\gamma}{K}, \quad i = 1, ..., K$

2. Attack channel $I_t \sim p_t$ and accumulate the unobservable reward $x_t(I_t)$.

3. In the observation phase, choose channel $J_t := 1 + (I(t) \ mod \ K)$, and observe its reward $x_t(J_t)$ based on equation (2.1).

4. In the learning phase, for $j = 1, 2, ..., K$

$$\hat{x}_t(j) = \begin{cases} \frac{x_t(j)}{p_t(I_t)}, & j = J_t \\ 0, & o.w. \end{cases}$$

$$\omega_{t+1}(j) = \omega_t(j) \exp(\eta \hat{x}_t(j))$$

Based on the theoretical analysis in [14], EXP3-DO is categorized as a weakly observable feedback graph. and it is an attacking strategy as a baseline of applying current technologies for the PUE attacker's attack strategy. In other words, since the PUE attacker cannot scan through all the channels in each time slot due to the limited duration of a time slot, achieving an optimal regret order for the attacker is not possible by applying the current theoretical frameworks. As a result, in this algorithm the observation strategy is designed such that at least no action is left un-observed. The observation strategy stated here in not the only observation structure and any permutation of channels that creates a partially observable graph leads to the same regret bound.

We note that in Step 4 of Algorithm 2, only the weight of the observed channel is updated. The estimated reward $\hat{x}_t(j)$ is an unbiased estimate of the actual reward $x_t(j)$, i.e., conditional on all previously chosen channels before t, we have $E[\hat{x}_j(t)|I_1,\ldots,I_{t-1}] = \frac{x_t(j)}{p_t(j')}p_t(j') = x_t(j)$ where $j' = (j-2)\ mod\ K+1$, i.e., the neighboring channel chosen for attack.

Theorem 3. *For any $K \geq 2$ and for any $\eta \leq \frac{\gamma}{K}$, the upper bound on the expected regret of Algorithm 2*

$$G_{max} - E[G_{EXP3-DO}] \leq (\gamma + (e-2)\frac{\eta K}{\gamma})G_{max} + \frac{\ln K}{\eta}.$$

holds for any assignment of rewards for any $T > 0$.

By choosing appropriate values for γ and η, the above upper bound on the regret can be minimized.

Proof of Theorem 3. Our proposed observing strategy can be categorized as a partially observable graph in [14]. Because by choosing an action, the reward on only one other channel is observed not on all the actions. Also no node is left un-observed. In [14] the upper-bound of the regret of weakly observable graphs is proved. □

Corollary 2. *For any $T > 0$ and the following values*

$\eta = \frac{\gamma^2}{K(e-2)}$, *and* $\gamma = \sqrt[3]{(e-2)K \ln K/g}$

where g is an upper-bound on the G_{max}. Then

$$G_{max} - E[G_{EXP-DO}] \leq 3\sqrt[3]{(e-2)}\sqrt[3]{T^2 K \ln K}$$

holds for any arbitrary assignment of rewards.

Proof. We sketch the proof as follows. Since $\gamma \leq (e-2)$, in order for the regret bound to be non-trivial, we need $g \geq \frac{K \ln K}{(e-2)^2}$. Then by getting the derivative, we find the optimal values for η and γ. Also T is an upper bound on the g since all the rewards are in $[0, 1]$ and the network runs for T time slots, which gives us the result. □

Theorem 4. *For any $K \geq 2$ and for any player strategy A, the expected weak regret of algorithm 2 is lower bounded by*

$$G_{max} - \mathbf{E}[G_A] \geq c\sqrt[3]{KT^2}$$

for some small constant c and it holds for some assignment of rewards for any $T > 0$.

Proof of Theorem 4. The proof of the lower bound analysis is provided in [14]. □

3.3 Attacking Strategy 3: PROLA Learning Algorithm

In this section, we propose another novel online learning algorithm that can be applied by the PUE attacker with one observation capability within the attacking slot. The proposed optimal online learning algorithm, called PROLA, at each time, chooses two actions dynamically to play and observe, respectively. The action to play is chosen based on an exponential weighting, while the other action for observation is chosen uniformly at random excluding the played action. The proposed algorithm is presented in Algorithm 3.

Algorithm 3 : PROLA, Play and Random Observe Learning Algorithm

Parameters: $\gamma \in (0,1)$, $\eta \in \left(0, \frac{\gamma}{2(K-1)}\right]$.

Initialization: $\omega_1(i) = 1, \quad i = 1, ..., K.$

For each $t = 1, 2, ..., T$

1. Set $p_t(i) = (1-\gamma)\frac{\omega_t(i)}{\sum_{j=1}^{K}\omega_t(j)} + \frac{\gamma}{K}, \qquad i = 1, ..., K.$
2. Attack channel $I_t \sim p_t$ and accumulate the unobservable reward $x_t(I_t)$.
3. Choose a channel J_t other than the attacked one uniformly at random and observe its reward $x_t(J_t)$ based on equation (2.1).
4. For $j = 1, ..., K$

$$\hat{x}_t(j) = \begin{cases} \frac{x_t(j)}{(1/(K-1))(1-p_t(j))}, & j = J_t \\ 0, & o.w., \end{cases}$$

$$\omega_{t+1}(j) = \omega_t(j)\exp(\eta\hat{x}_t(j)).$$

Figure 3.3 shows the observation policy governing the actions for $K = 4$ in a feedback graph format. In this figure, $Y_{ij}(t)$ is an observation indicator of channel j when channel i is attacked at time t. We define $Y_{ij}(t) \in \{0, 1\}$ such that at each time for the chosen action i to be played,

$$\sum_{j=1, j\neq i}^{K} Y_{ij}(t) = 1, \quad i = 1, \ldots, K \quad t = 1, \ldots, T. \tag{3.10}$$

In other words, there is a policy based on which for any channel i played, only one of the

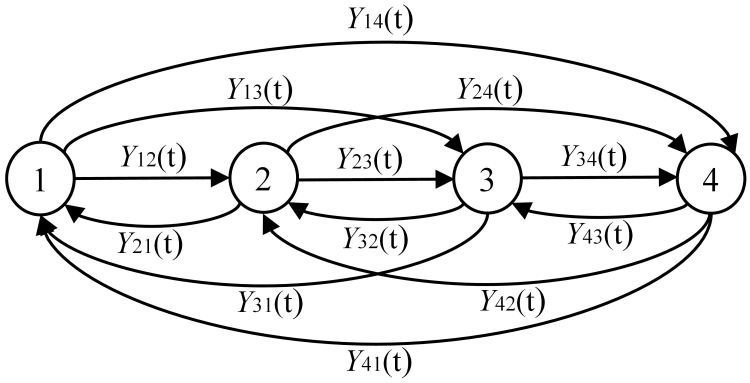


Figure 3.3: Channel observation strategy, $K = 4$

observation indicators takes a value of one and the rest take a value of zero. For example, if channel 2 is attacked ($i = 2$), only one of the three outgoing edges from 2 will be equal to one. This edge selection policy represents the channel selection process for observation. We define it as a uniform random distribution equal to $\frac{1}{K-1}$. We call this feedback graph, a time-variable random feedback graph. Our feedback graph fits into the time-variable feedback graphs introduced in [14] and based on the results derived in that work, the regret upper bound of our algorithm is $\tilde{O}(\sqrt[3]{T^2})$. However, based on our analysis, the upper bound on the attacker's regret is in the order of $\tilde{O}(\sqrt{T})$ which shows a significant improvement. In other words, despite the fact that the agent makes only partial observation on the channels, it achieves a significantly improved regret order compared to the no observation in the attacking slot case. This has been possible due to the new property we considered in the partially observable graphs which is adding randomness. In the long run, randomness makes full observation possible to the agent.

In Step 4 of Algorithm 3, in order to create $\hat{x}_t(j)$, an unbiased estimate of the actual reward $x_t(j)$, we divide the observed reward, $x_t(J_t)$, by $(1/(K-1))(1-p_t(J_t))$which is the probability of choosing channel J_t to be observed. In other words, channel J_t will be chosen to be observed if it has not been chosen for attacking, with probability $(1-p_t(J_t))$, and

second if it gets chosen uniformly at random from the rest of the channels, with probability $(1/(K-1))$.

Theorem 5. *For any $K \geq 2$ and for any $\eta \leq \frac{\gamma}{2(K-1)}$, for the given randomized observation structure for the attacker the upper bound on the expected regret of Algorithm 3,*

$$G_{max} - \mathbf{E}[G_{PROLA}] \leq (e-2)(K-1)\frac{\eta}{1-\gamma}T + \frac{\ln K}{\eta(1-\gamma)}$$

holds for any assignment of rewards for any $T > 0$.

Proof of Theorem 5. By using the same definition for $W_t = \omega_t(1) + \cdots + \omega_t(K) = \sum_{i=1}^{K} \omega_t(i)$ as in Algorithm 1, at each time t,

$$\begin{aligned}\frac{W_{t+1}}{W_t} &= \sum_{i=1}^{K} \frac{p_t(i) - \gamma/K}{1-\gamma} \exp(\eta \hat{x}_t(i)) \\ &\leq \exp\left(\frac{\eta}{1-\gamma}\sum_{i=1}^{K} p_t(i)\hat{x}_t(i) + \frac{(e-2)\eta^2}{1-\gamma}\sum_{i=1}^{K} p_t(i)\hat{x}_t^2(i)\right). \qquad (3.11)\end{aligned}$$

The equality follows from the definition of W_{t+1}, $\omega_{t+1}(i)$, and $p_t(i)$ respectively in Algorithm 3. Also, the inequality follows from the fact that $e^x \leq 1 + x + (e-2)x^2$ for $x \leq 1$ and $e^x \geq 1 + x$. When $\eta \leq \frac{\gamma}{2(K-1)}$, the result, $\eta\hat{x}_t(i) \leq 1$, follows from the observation that either $\eta\hat{x}_t(i) = 0$ or $\eta\hat{x}_t(i) = \eta\frac{x_t(i)}{\frac{1}{K-1}(1-p_t(i))} \leq \eta(K-1)\frac{2}{\gamma} \leq 1$, since $x_t(i) \leq 1$ and $p_t(i) = (1-\gamma)\frac{\omega_t(i)}{\sum_{j=1}^{K}\omega_t(j)} + \frac{\gamma}{K} \leq 1 - \gamma + \frac{\gamma}{2} \leq 1 - \frac{\gamma}{2}$.

By taking the logarithm of both sides of equation (3.11) and summing over t from 1 to T, we derive the following inequality on the LHS of the equation,

$$\sum_{t=1}^{T} \ln \frac{W_{t+1}}{W_t} = \ln \frac{W_{T+1}}{W_1} \geq \eta \sum_{t=1}^{T} \hat{x}_t(j) - \ln K. \tag{3.12}$$

Combining (3.11) and (3.12), we can get

$$\sum_{t=1}^{T} \hat{x}_t(j) - \sum_{t=1}^{T} \sum_{i=1}^{K} p_t(i) \hat{x}_t(i)$$

$$\leq \gamma \sum_{t=1}^{T} \hat{x}_t(j) + (e-2)\eta \sum_{t=1}^{T} \sum_{i=1}^{K} p_t(i) \hat{x}_t^2(i) + \frac{\ln K}{\eta}. \tag{3.13}$$

Let $\acute{x}_t(i) = \hat{x}_t(i) - f_t$ where $f_t = \sum_{i=1}^{K} p_t(i) \hat{x}_t(i)$. We make the pivotal observation that (3.13) also holds for $\acute{x}_t(i)$ since $\eta \acute{x}_t(i) \leq 1$, which is the only key to obtain (3.13).

We also note that,

$$\begin{aligned}
\sum_{i=1}^{K} p_t(i) \acute{x}_t^2(i) &= \sum_{i=1}^{K} p_t(i) (\hat{x}_t(i) - f_t)^2 \\
&= \sum_{i=1}^{K} p_t(i) \hat{x}_t^2(i) - f_t^2 \\
&\leq \sum_{i=1}^{K} p_t(i) \hat{x}_t^2(i) - \sum_{i=1}^{K} p_t^2(i) \hat{x}_t^2(i) \\
&= \sum_{i=1}^{K} p_t(i)(1 - p_t(i)) \hat{x}_t^2(i).
\end{aligned} \tag{3.14}$$

Substituting $\acute{x}_t(i)$ in equation (3.13) and combining with (3.14),

$$\sum_{t=1}^{T} (\hat{x}_t(j) - f_t) - \sum_{t=1}^{T} \sum_{i=1}^{K} p_t(i)(\hat{x}_t(i) - f_t) = \sum_{t=1}^{T} \hat{x}_t(j) - \sum_{t=1}^{T} \sum_{i=1}^{K} p_t(i) \hat{x}_t(i)$$

$$\leq \gamma \sum_{t=1}^{T} \left(\hat{x}_t(j) - \sum_{i=1}^{K} p_t(i)\hat{x}_t(i) \right) + (e-2)\eta \sum_{t=1}^{T}\sum_{i=1}^{K} p_t(i)(1-p_t(i))\hat{x}_t^2(i) + \frac{\ln K}{\eta}. \quad (3.15)$$

Observe that $\hat{x}_t(j)$ is similarly designed as an unbiased estimate of $x_t(j)$. Then for the expectation with respect to the sequence of channels attacked by the horizon T,

$$\mathbf{E}[\hat{x}_t(j)] = x_t(j), \quad \mathbf{E}\left[\sum_{i=1}^{K} p_t(i)\hat{x}_t(i)\right] = \mathbf{E}[x_t(I_t)],$$

and

$$\mathbf{E}\left[\sum_{i=1}^{K} p_t(i)(1-p_t(i))\hat{x}_t^2(i)\right] = \mathbf{E}\left[\sum_{i=1}^{K} p_t(i)(K-1)x_t(i)\hat{x}_t(i)\right] \leq K-1.$$

We now take the expectation with respect to the sequence of channels attacked by the horizon T in both sides of the last inequality of (3.15). For the left hand side,

$$\mathbf{E}\left[\sum_{t=1}^{T} \hat{x}_t(j)\right] - \mathbf{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{K} p_t(i)\hat{x}_t(i)\right] = G_{max} - \mathbf{E}[G_{PROLA}]. \quad (3.16)$$

and for the right hand side,

$$\mathbf{E}[\text{R.H.S}] \leq \gamma(G_{max} - \mathbf{E}[G_{PROLA}]) + (e-2)(K-1)\eta T + \frac{\ln K}{\eta}$$

Combining the last two equations we get,

$$(1-\gamma)(G_{max} - \mathbf{E}[G_{PROLA}]) \leq (e-2)(K-1)\eta T + \frac{\ln K}{\eta},$$

and since G_{max} can be substituted by T, by rearranging the relation above the regret upper bound is achieved. □

Similarly, we can minimize the regret bound by choosing appropriate values for η and γ.

Corollary 3. *For any $T \geq \frac{8(K-1)\ln K}{e-2}$ and $\gamma = \frac{1}{2}$, we consider the following value for η*

$$\eta = \sqrt{\frac{\ln K}{2(e-2)(K-1)T}}.$$

Then

$$G_{max} - \mathbf{E}[G_{PROLA}] \leq 2\sqrt{2(e-2)}\sqrt{T(K-1)\ln K}$$

holds for any arbitrary assignment of rewards.

Proof of Corollary 3. We sketch the proof as follows. By getting the derivative from the statement in Theorem 3 with respect to η, we find the optimal value for η. Since $\eta \leq \frac{\gamma}{2(K-1)}$, in order for the regret bound to hold, we need $T \geq \frac{8(K-1)\ln K}{e-2}$. Replacing the value of η gives us the result on the regret upper bound. □

The important observation is that, based on [14] such an algorithm, Algorithm 3, is expected to achieve a regret in the order of $\tilde{O}\left(\sqrt[3]{T^2}\right)$ since it can be categorized as a partially observable graph. However, our analysis gives a tighter bound and shows not only it is tighter but also it achieves the regret order of fully observable graphs. This significant improvement has been accomplished by introducing randomization into feedback graphs.

The following theorem provides the regret lower bound for this problem.

Theorem 6. *For any $K \geq 2$ and for any player strategy A, the expected weak regret of algorithm A is lower bounded by*

$$G_{max} - \mathbf{E}[G_A] \geq c\sqrt{KT}$$

for some small constant c and it holds for some assignment of rewards for any $T > 0$.

Proof of Theorem 6. The proof follows closely the lower bound analysis in [17]. Details are provided in the Appendix. □

3.4 Extension to Multiple Action Observation Capability

We generalize Algorithm PROLA to the case of an agent with multiple observation capability. This is suitable for an attacker with multiple observation capability when the attacker after the attack phase is able to observe multiple other channels within the same time-slot. At least one and at most $K-1$ observations are possible by the agent (attacker). In this case, m indicates the number of possible observations and $1 \leq m \leq K-1$. Then, for m observations, we modify equation (3.10) as follows,

$$\sum_{j=1,j\neq i}^{K} Y_{ij}(t) = m, \quad i = 1, \ldots, K \quad t = 1, \ldots, T. \tag{3.17}$$

The probability of a uniform choice of m observations at each time, $\sum_{j=1,j\neq i}^{K} Y_{ij}(t) = m$ is equal to $\frac{1}{\binom{K-1}{m}}$. Corollary 3 shows the result of the analysis for m observations.

Corollary 4. *If the agent can observe m actions at each time, then the regret upper bound is equal to*

$$G_{max} - \mathbf{E}\left[G_A\right] \leq 4\sqrt{(e-2)}\sqrt{T\frac{K-1}{m}\ln K}.$$

This regret upper bound shows a faster convergence rate when making more observations.

Proof of Corollary 4. We substitute $1/(K-1)$ by $m/(K-1)$ in Step 4 of Algorithm PROLA since at each time, m actions are being chosen uniformly at random. Then the regret upper bound can be derived by similar analysis. □

As our analysis shows, making multiple observations does not improve the regret in terms of its order in T compared to the case of making only one observation. This means that only

one observation and not more than that is sufficient to make a significant improvement in terms of regret order reduction compared to the case of no observation within the attacking slot. The advantage of making more observations however is in reducing the constant coefficient in the regret. Making more observations leads to a smaller constant factor in the regret upper bound. This relationship is non-linear though. i.e., the regret upper bound is proportional to $\sqrt{1/m}$ which means that in order to make a large reduction in the regret in terms of its constant coefficient impact, only a few observations would suffice. Our simulation results in Section 4.3 provide more details on this non-linear relationship.

Chapter 4: Performance Evaluation

In this section, we present the simulation results to evaluate the validity of the proposed online learning algorithms applied by the PUE attacker. All the simulations are conducted in MATLAB and the results achieved are averaged over 10,000 independent random runs.

We evaluate the performance of the proposed learning algorithms, POLA and PROLA, and compare them with their theoretical regret upper bounds $\tilde{O}(\sqrt[3]{T^2})$ and $\tilde{O}(\sqrt{T})$, respectively. POLA and PROLA correspond to an attacker with no observation and one observation capability within the attacking time slot, respectively. We then examine the impact of different system parameters on the attacker's performance. The parameters include the number of time slots, total number of channels in the network, and the distribution on the PU activities. We also examine the performance of an attacker with multiple observation capabilities. Finally we evaluate a secondary user's accumulated traffic with and without the presence of a PUE attacker.

K primary users are considered, each acting on one channel. The primary users' on-off activity follows a Markov chain or i.i.d. distribution in the network. Also, the PU activities on different channels are independent from each other. K idle probabilities are generated using MATLAB's rand function, each denoting one PU activity on each channel if PUs follow an i.i.d. Bernoulli distributions. pI = [0.85 0.85 0.38 0.51 0.21 0.13 0.87 0.7 0.32 0.95] is a vector of K elements each of which denotes the corresponding PU activity on the K channels. If the channels follow Markov chains, for each channel, we generate three probabilities, $p01$, $p10$, and $p1$ as the transition probabilities from state 0 (on) to 1 (off), from 1 to 0, and the initial idle probability, respectively. The three vectors below, are examples considered to represent PU activities. $p01$ = [0.76 0.06 0.3 0.24 0.1 0.1 0.01 0.95 0.94 0.55], $p10$ = [0.14 0.43 0.23 0.69 0.22 0.59 0.21 0.58 0.34 0.73], and $p1$ = [0.53 0.18 0.88 0.66 0.23 0.87 0.48

0.44 0.45 0.88].

The PUE attacker employs either of the proposed attacking strategies, POLA, or PROLA. Throughout the simulations, when we talk about an attacker employing PROLA, we assume the attacker's observation capability is one within the attacking slot, unless otherwise stated.

Since the goal here is to evaluate the PUE attacker's performance, for simplicity we consider one SU in the network. Throughout the simulations, we assume the SU employs an online learning algorithm called Hedge [16]. The assumption on the Hedge algorithm is that the secondary user is able to observe the rewards on all the channels in each time slot. Hedge provides the minimum regret in terms of both order and the constant factor among all optimal learning-based algorithms. As a result the performance of our proposed learning algorithms can be evaluated in the worst case scenario for the attacker. As explained in Section 1.1, even though in our analysis we considered an oblivious environment, in the simulations we can consider an SU that runs Hedge (an adaptive opponent). This experimental setup adds value since it shows that our proposed online learning-based algorithms perform reasonably even against adaptive opponents by keeping the occurred regret between the theoretical upper and lower bounds derived.

4.1 Performance of POLA and the impact of the number of channels

In this section, we evaluate the performance of the proposed algorithm, POLA and compare it with the theoretical analysis. Figure 4.2(a) shows the overall performance of the attacker for both cases of i.i.d. and Markovian chain distribution of PU activities on the channels for $K = 10$. Regret upper bound and lower bound from the analysis in Corollary 1 and Theorem 2, respectively are also plotted in this figure. Then we examine the impact of the number of channels in the network on it's performance. We consider K variable from 10 to 50. The attacker's regret when PUs follow i.i.d. distribution and Markovian chain for

different number of actions are shown in Fig. 4.2(b) and (c), respectively. Since POLA has a regret with higher slope, in order to better observe the results, we have plotted the figures for T from 1 to $20,000$. We can observe the following from the figures.

- Regardless of the PUs activity type, the occurred regret is below the regret upper bound achieved from the theoretical analysis.
- The regret on all three figures has a higher slope compared to the results for PROLA in Fig. 4.2(d)-(f) which also complies with our analysis. The higher slope in these figures can be seen by comparing their x and y axises.
- As the number of channels increases the regret increases as is expected based on the analysis from Corollary 1.
- As the number of channels increases, the regret does not increase linearly with it. Instead, the increment in the regret becomes marginal which complies with the theoretical analysis. Based on Corollary 1 the regret is proportional to $(\sqrt[3]{K \ln K})$. The dependency on K can be represented by plotting the regret versus K as is shown in the next subsection.

4.2 Performance of PROLA and the impact of the number of channels

We compare the performance of the proposed learning algorithm, PROLA, with the theoretical analysis from section 3. We consider a network of $K = 10$ channels. Figure 4.2(d) shows the simulation results as well as analytical results. From the simulations, we observe that the actual regret occurred in the simulations, is between the bounds achieved from the analysis regardless of the type of PU activity which complies with our analysis. We also note that when we derived the theoretical upper bound we did not make any assumption on the PU activity. The regret is only dependent on the K and T.

Then, we examine the impact of the number of channels in the network on the attacker's performance when it applies PROLA. Figure 4.2(e) and (f) show the attacker's regret when PUs follow i.i.d. distribution and Markovian chain respectively for K variable from 10 to 50. The same discussion on the system parameters and the results hold as in subsection 4.1. In another representation, we plot the regret versus the number of channels, K, as the x axis in Fig. 4.2(g).

Moreover, comparing regret values in Fig. 4.2(a) with those in Fig. 4.2(d), we observe the huge difference in regret amount between no observation capability within the attacking slot and one observation capability.

4.3 Impact of the number of observations in each time slot

We consider PROLA algorithm with $K = 40$ channels in the network. The number of observing channels, m, varies from 1 to 35. Figure 4.2(h) and (i) show the performance of the PROLA for $m = 1, 3, 8, 18, 35$ when the PUs follow i.i.d. distribution and Markovian chain on the channels, respectively. We can observe the following from the simulation results.

- As the observation capability of the attacker increases, it achieves a lower regret. This observation complies with the Corollary 3 provided in Section 3.4 based on which we expect smaller constant factor as the observation capability increases.
- In the beginning, even adding a couple of more observing channels (from $m = 1$ to $m = 3$), the regret decreases dramatically. The decrement in the regret becomes marginal as the number of observing channels becomes sufficiently large (e.g., from $m = 18$ to $m = 35$). This observation implies that, in order to achieve a good attacking performance (smaller constant factor in regret upper bound), the attacker does not need to be equipped with high observation capability. In the simulation, when the number of observing channels ($m = 10$) is $\frac{1}{4}$ of the number of all channels ($K = 40$), the regret is approaching to the optimal.

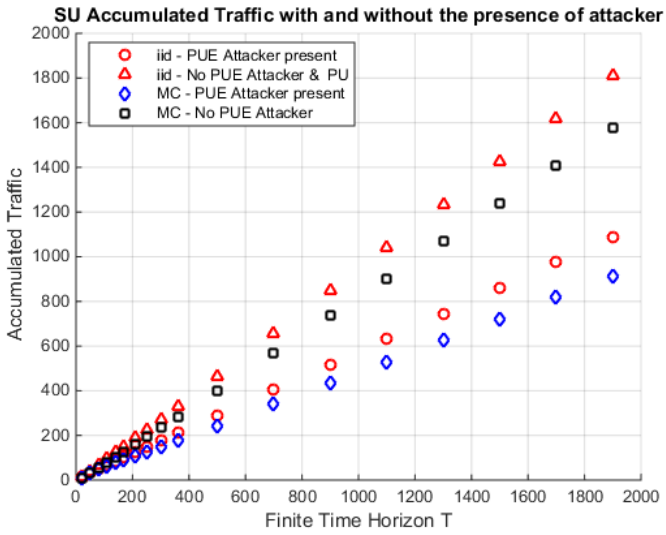
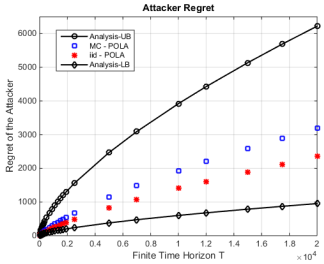


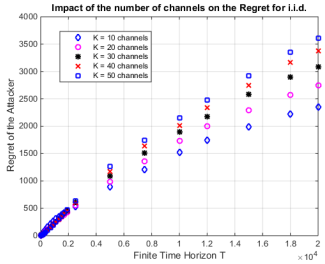
Figure 4.1: Accumulated Traffic of an SU

4.4 Accumulated Traffic of SU with and without attacker

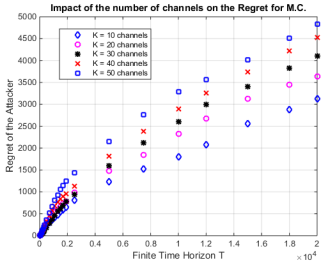
We set $K = 10$ and measure the accumulated traffic achieved by the SU with and without the presence of the attacker. The attacker employs the PROLA algorithm. Figure 4.1 shows that the accumulated traffic of the SU is largely decreased when there is a PUE attacker in the network for both types of PU activities.



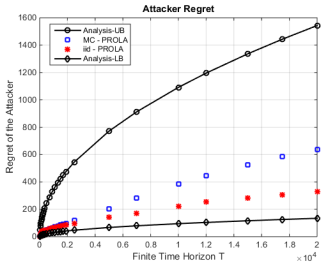
(a) POLA Learning



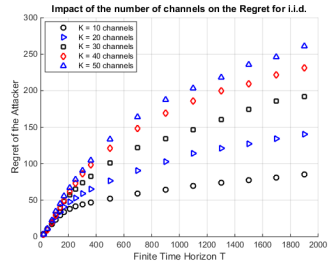
(b) POLA, channel impact, i.i.d. PU



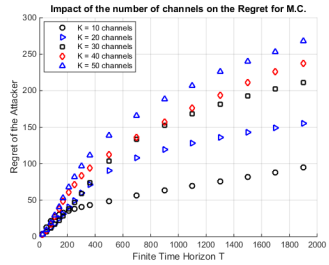
(c) POLA, channel impact, M.C. PU



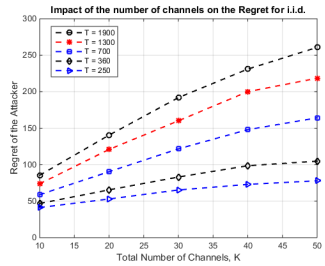
(d) PROLA Learning



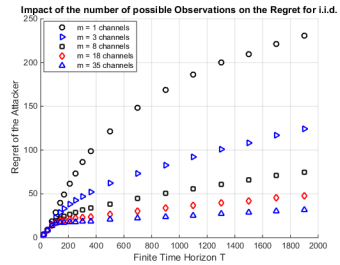
(e) PROLA, channel impact, i.i.d. PU



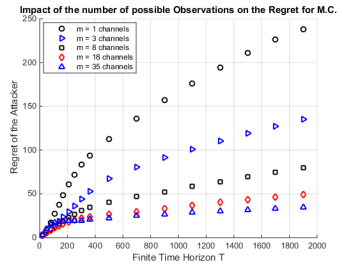
(f) PROLA, channel impact, M.C. PU



(g) PROLA, channel impact, i.i.d. PU



(h) PROLA, observation impact, i.i.d. PU



(i) PROLA, observation impact, M.C. PU

Figure 4.2: Simulation results under different PU activity assumptions

Chapter 5: Intelligence Measure of Cognitive Radios

In order to resolve the imminent spectrum shortage problem, sharing spectrum with legacy systems has attracted intensive research during the past decade. Cognitive radio (CR), which has the capability to sense, learn, and adapt to the spectrum environment [9, 29, 30], can significantly improve spectrum efficiency and guarantee the unharmful coexistence with the legacy systems [31, 32, 33, 34, 35]. Nevertheless, the complex and uncertain spectrum environment makes spectrum sharing extremely challenging. The uncertainty may come from the radio propagation environment, the legacy system activity, or the complex behavior of the CR itself.

Just like human being, sophisticated cognitive capabilities are essential for the CR to cope with the uncertainty of spectrum environment. The cognitive capabilities collectively define the intelligence of CR. Although the CR concept was born with the core idea of realizing "cognition" [36], the research on measuring CR cognitive capabilities or intelligence is largely open.

Being able to quantitatively measure the intelligence of CR can bring us a lot of benefits.

1. With the intelligence model and measuring methodology, we will gain deeper insight about the key factors that affect the intelligence of a CR which can be used to guide the development of new CRs with high intelligence.

2. A CR vendor may advertise and price their CR products based on CR intelligence as a metric. A CR with higher intelligence tends to achieve better performance in practically uncertain spectrum environments, thus will be priced higher.

3. With the knowledge of the intelligence of individual CRs, a service provider or network manager can better configure their networks by integrating CRs with different intelligence levels in a more cost-efficient way. For example, a CR with higher intelligence

leading a set of CRs with lower intelligence may achieve a desirable performance with low network deployment cost.

4. Last but not the least, the investigation of CR intelligence will shed light on the intelligence measure of other smart systems, such as connected cars [2, 37], unmanned aerial vehicles [1, 38, 39], smart grid [40], smart cities [41], etc.

This work is an extension of our previous work [20], in which we proposed a data-driven methodology to derive the intelligence measure. We construct a CR intelligence model following human intelligence theory, specifically the widely accepted Cattell-Horn-Carroll (CHC) intelligence model [42]. Based on this model, we develop psychometric techniques to measure the CR intelligence. The basic idea of our methodology is to use simulations to test different CRs in various spectrum environments under different settings. Based on the obtained performance data, we apply the factor analysis (FA) technique [43] to extract and measure the intelligence factors of CR.

More specifically, we present a case study consisting of 144 different types of CRs. We provide each CR with different levels of capabilities including learning-based algorithms [15, 17, 30, 44] for dynamic spectrum access, number of sensors, sensing accuracy, processing speed, and algorithmic complexity. With our methodology, five intelligence factors are identified for the CRs through our analysis, which are shown to comply with the nature of the tested algorithms. This validates our proposed methodology of measuring CR intelligence.

We summarize the contributions of this paper as follows:

- For the first time, we propose the idea of identifying the cognitive capabilities of CR and introduce an intelligence model for the CR.
- We propose a methodology to extract the CR's intelligence factors and apply factor analysis as a theoretical framework for this purpose.
- The proposed methodology is verified through a case study where we identify the

intelligence factors of learning-based CRs under dynamic spectrum access scenarios and show these factors comply with the nature of the CRs.

In the rest of this work, first we propose our intelligence model for CR in Section 5.2. Section 5.3 presents our methodology of deriving CR intelligence factors. In section 6, we present a case study in which we measure the intelligence of learning based CRs under a dynamic spectrum access scenarios. Section 5.1 discusses the related work and compares them with our approach. In particular, work on human intelligence measure and the difference between CR intelligence measure and human intelligence measure are highlighted.

5.1 Related Work

5.1.1 Reinforcement Learning

Reinforcement learning (RL) is a branch of machine learning, designed for online learning. Similar to MAB problems, the RL methods need to trade off between exploration and exploitation. Q-Learning is a well studied topic and is categorized as a reinforcement learning technique that can be used to find the optimal action selection policy [44, 45]. The environment is usually assumed to follow Markov Chain Process.

5.1.2 CR Intelligence

Intelligence measure of CRs has not been well studied in the literature. However, there are various studies on evaluating the performance of CRs. A cognitive radio test methodology to test a CR system is presented in [46]. The effect of cognitive engine on both SU and PU performance is measured and evaluated. It is suggested that the cognition may be measured based on the SU's capability to improve its throughput and at the same time to decrease PU interference. The authors call their method behavior-based testing. In other words, their goal is to measure SU cognition based on the evaluation of both SU and PU performances

instead of evaluating the SU cognition itself. The testing scenarios are defined as narrow-band or wide-band environments. The PU workloads and SU cognition considered in this work are limited and the authors suggest more research as a required step to justify the behavior-based cognition testing. Statistical tools and the psychometrics are not utilized in contrast to our work that considers those methods. This indicates that our approach is completely different from this work.

The performance of cognitive radios is studied in [47] which considers four cognitive radio algorithms and intends to distinguish those that perform better than the others often enough. They also study how sensitive different algorithms are to suboptimal parameters. It is shown through simulations that, usually those algorithms that outperform others are highly sensitive to sub-optimal parameters. While the others that show lower performance, represent a more steady performance and are more resistant to sub-optimality in the parameters. The conclusion is that there is a trade-off between performance and consistency. The difference of this work with ours is that their goal is to compare the performance of different learning based algorithms and to distinguish those that show consistent performance and have less dependency on the parameter values. However, we derive the cognitive capabilities of CRs which is a totally different aspect of CR intelligence measurement.

5.1.3 Cognitive Capabilities of Humans

The cognitive capabilities and the intelligence model of human beings have been studied extensively in psychology [48]. Human cognition capabilities include sensing, learning, memory, problem solving, etc. Intelligence is defined as the ability to learn and perform cognitive tasks [48]. Cattel-Horn-Carrot [42] is the most widely accepted model of human intelligence [35, 48].

At the top layer of the CHC model is the Stratum III, which defines a unique general intelligence factor g. People with high loadings on the g factor are more intelligent in general. The middle layer is the Stratum II in which eight broad cognitive capabilities are defined. For example, in stratum II, there is a factor modeling knowledge acquired through

education and memory. Older people intend to have higher loadings on this factor. In contrast, the fluid intelligence is a factor that denotes the capability of solving problems never seen before. People with higher loadings on this factor, usually have strong learning and problem solving capabilities. Finally, at the bottom is Stratum I with more than eighty narrow abilities. The narrow capabilities are more specific cognition capabilities such as induction, or reduction.

The practical measurement of mental abilities has been considered as a pivotal development in the behavioral sciences and the theories and techniques formed a field called "psychometrics". The first attempts of a mathematically more rigorous study of intelligence measure occurred in the 1940s, with statistical techniques such as correlation and FA. Overall, FA is used in multiple areas including psychology and economics.

Factor Analysis (FA) is an statistical method used in psychology. FA is able to extract the cognitive capabilities of the test taker. It can also be used to test a theory on possible cognitive capabilities. In other words, to determine whether the designed questions of the test measure the same factors that the questions were designed for [43, 49] .

There have been some efforts trying to develop comprehensive benchmark frameworks to evaluate the cognitive radio network performance [50], or to evaluate the performance of more general wireless networks [51, 52, 53]. Since benchmarking wireless network is challenging, simulation has been adopted widely as a tool in the literature. However, such benchmark studies are proposed not to test CR intelligence, but to evaluate CR performance. Our focus here is on the factor analysis and intelligence model construction, which is different than the conventional benchmarking research.

It is helpful to identify the differences between human and CR intelligence capabilities. One is that for human beings, the age of the test taker is an important factor that needs to be considered when designing the test questions, such as at the childhood stages in which the brain is still developing. However, with respect to the CRs, a testing scenario can be tested by all types of CRs.

Another important difference is that a human being can get tired by the long duration

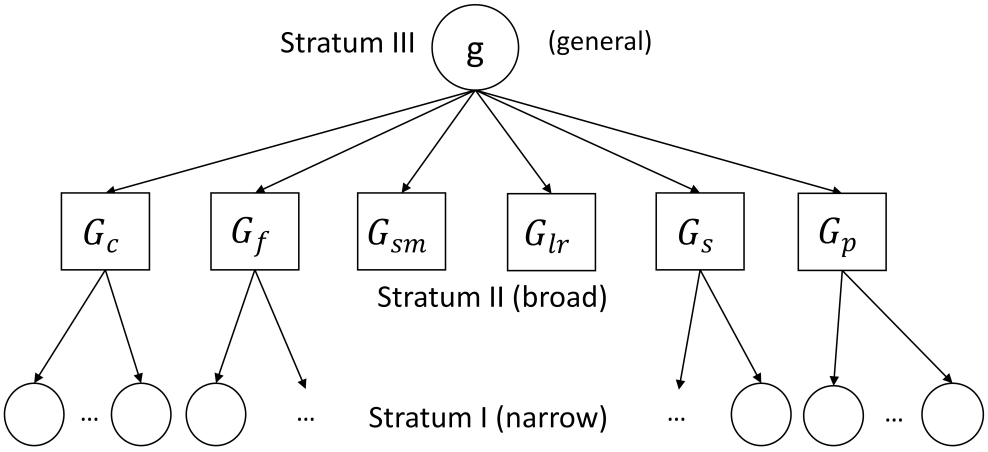


Figure 5.1: Intelligence model for the cognitive radio

of the test or may not be able to focus on the test day. This can make the test results unreliable. However, this is not a problem for CR and the test results can always be correct, unbiased and reliable.

The other difference is that for human beings the answer to each test question is considered either as zero or one. However for CR the score for each test scenario can be any real number not necessarily zero or one.

For human being we measure only one output as the score of the test taker when we aim to measure the general intelligence factor. However, for cognitive radios we generalize this notion to measure several kinds of output which in our case study include achieved throughput, delay, and the violation ratio.

5.2 Quantitative Intelligence Model of CR

In this section, first we propose an intelligence model for the CR. Then, in the next section, we propose a data-driven methodology to derive the intelligence capabilities of cognitive radios.

Motivated by the CHC model [42] that is widely used to describe human intelligence, we propose an intelligence model for the CR. Our model is structured with three strata (or

stages) as shown in Fig. 5.1. At the top stage lies the stratum III, which defines a unique general intelligence factor g. CRs with high values in the g factor are more intelligent in general. That is, they tend to achieve better performance in various dynamic spectrum environments.

The stratum II represents more broad abilities in terms of cognition capabilities contributing to intelligence, which may be modeled as the following ones:

1. Comprehension-Knowledge (G_c): includes the breadth and depth of a CR's acquired knowledge and the ability to reason using previously learned experiences or procedures.

2. Fluid reasoning (G_f): includes the broad ability to reason, form concepts, and perform dynamic spectrum access using unfamiliar information or novel procedures.

3. Short-Term Memory (G_{sm}): is the ability to apprehend and hold information in immediate awareness and then use it within a short period (e.g., a few seconds or the time the CR is on).

4. Long-Term Storage and Retrieval (G_{lr}): is the ability to store information and retrieve it later in the process of communication or dynamic spectrum access.

5. Spectrum Sensing (G_s): is the ability to sense the spectrum environment, e.g., sensing the availability of white space or presence of primary users.

6. Processing Speed (G_p): measures the information processing time, which may include channel sensing, accessing and switching delay, computing, reasoning, and information retrieval delay, etc.

Within each stratum II broad ability, we can further define stratum I which is at the bottom with more narrow abilities. These abilities are more specific cognition capabilities. For example, fluid reasoning can include inductive reasoning, sequential reasoning, deductive reasoning, and speed of reasoning. Spectrum sensing can include number of sensors and

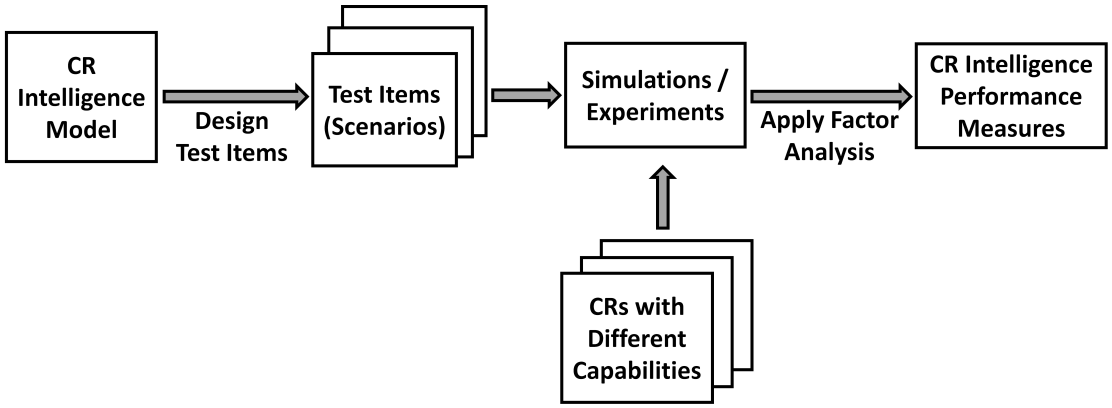


Figure 5.2: A data-driven methodology to measure the intelligence of CR.

accuracy of sensing capability. Processing speed can include the speed of processing on the received data, the speed of reasoning and decision making, and the speed of switching among channels.

5.3 Proposed Methodology to Measure the intelligence capabilities of CR

In this section, we propose a data-driven methodology to measure the intelligence of CRs. The basic idea of this methodology is illustrated in Fig. 5.2.

For a pool of N different CRs called CR_1, CR_2, ..., CR_N, we design a set of K test items to evaluate their performance. CRs are different in terms of learning based spectrum access strategy, number of sensors, processing speed, computational complexity, etc. Various test environments arise from different primary user activity types or statistics, channel rates, frame delivery ratio, etc. Through testing each CR in the testing scenarios, we obtain a vector of performance data $\boldsymbol{Y}_k(n)$ for each CR_n $(1 \leq n \leq N)$ at each test scenario k $(1 \leq k \leq K)$. The dimension of $\boldsymbol{Y}_k(n)$ equals to the number of performance metrics used. In our case study, $\boldsymbol{Y}_k(n)$ is an array of length three for each cognitive radio performing in a given test scenario since we measure three performance metrics for each CR. Then we apply

the FA method [43] on the measured data to derive the intelligence factors as latent factors. These factors are then matched to the broad cognitive capabilities described in Section 5.2 through analyzing the nature of the CR functions.

FA technique is applied on the data matrix $\boldsymbol{Y} = \{\boldsymbol{Y}_k(n), \quad 1 \leq n \leq N, 1 \leq k \leq K\}$, which identifies the latent factors as intelligence factors. The latent factors are then matched to the right cognitive capabilities by analyzing the functions of the CRs.

There are two types of FA in the literature: exploratory FA and confirmatory FA [43, 49]. Exploratory FA is used to identify the potential latent factors when both the number and the loading of the latent factors are unknown. Meanwhile, confirmatory FA is used when the number of latent factors are known. Then by applying the confirmatory FA we can decide whether the model and FA results match with each other or not. It can also be used to test a theory on possible cognitive capabilities. In other words, it determines whether or not the designed questions of the test measure the same factors that the questions were designed for. In this thesis, we use confirmatory FA to test our theory on the possible intelligence factors.

To describe the details about the intelligence model and the latent factors, consider the performance of a test taker modeled as

$$y_k(n) = a_k g(n) + z_k(n), \tag{5.1}$$

where $y_k(n)$ is the measured performance of the cognitive radio n on the testing scenario k, $g(n)$ is the general intelligence factor (see the stratum III of the intelligence model in Fig. 5.1) of the cognitive radio n. The parameter $g(n)$ is called the "common factor", whose value determines how smart the CR n is to achieve high performance value $y_k(n)$. The weighting coefficient a_k denotes the loading, i.e., the importance, of the intelligence factor $g(n)$ on achieving high score $y_k(n)$ on the testing scenario k. The value of $z_k(n)$ summarizes performance deviation from the simplified model $a_k g(n)$, which is unique to the specific performance measurement and is thus called the "unique factor". Equation

(5.1) also shows how cognitive capabilities or intelligence factors can be modeled by the common factor $g(n)$ [43]. Having all the measured data $y_k(n)$, we can use FA to determine whether the data fit the model (Eq. (5.1) and if so to estimate the loading a_k and the intelligence factor $g(n)$.

For more detailed cognitive capability analysis, we can consider the list of broad cognitive capabilities in stratum II. Let $x_i(n)$ denote the ith intelligence factor (or latent factor), where $1 \leq i \leq I$. The performance data vector $\boldsymbol{Y}_k(n)$ can be modeled as

$$\boldsymbol{Y}_k(n) = \boldsymbol{a}_{k,1}x_1(n) + \boldsymbol{a}_{k,2}x_2(n) + ... + \boldsymbol{a}_{k,I}x_I(n) + \boldsymbol{Z}_k(n), \tag{5.2}$$

where $\boldsymbol{a}_{k,1}, \cdots, \boldsymbol{a}_{k,I}$ and $\boldsymbol{Z}_k(n)$ are the weights (loadings) and the unique factor, respectively. Note that since it is possible to measure several metrics, the single value $y_k(n)$ in (5.1) is substituted by the vector performance measurement $\boldsymbol{Y}_k(n)$. In this case, with all the measured data $\boldsymbol{Y}_k(n)$, we can verify the validity of the model (5.2) and determine the weighting coefficients $\boldsymbol{a}_{k,i}$ as well as the latent factors $x_i(n)$. By analyzing the CR functioning, we can match the latent factors $x_i(n)$ with the CR stratum II cognitive capabilities listed Section 5.2.

The FA technique [43] is applied to extract the group of latent factors $x_i(n)$ and then construct the CR intelligence model. To apply the FA method, we rewrite Eq. (5.2) into the matrix form

$$\boldsymbol{Y} = \boldsymbol{\Lambda}\boldsymbol{X} + \boldsymbol{\Psi}, \tag{5.3}$$

where $\boldsymbol{X}$ and $\boldsymbol{\Psi}$ are the matrices of common and the unique latent factors, respectively, and $\boldsymbol{\Lambda}$ is the matrix of weights $\boldsymbol{a}_{k,i}$. Specifically,

$$\boldsymbol{Y} = \begin{bmatrix} \boldsymbol{Y}_1(1) & \cdots & \boldsymbol{Y}_1(N) \\ \vdots & & \vdots \\ \boldsymbol{Y}_K(1) & \cdots & \boldsymbol{Y}_K(N) \end{bmatrix},$$

$$\mathbf{\Lambda} = \begin{bmatrix} \boldsymbol{a}_1(1) & \cdots & \boldsymbol{a}_1(I) \\ \vdots & & \vdots \\ \boldsymbol{a}_K(1) & \cdots & \boldsymbol{a}_K(I) \end{bmatrix}, \tag{5.4}$$

and the other matrices can be obtained similarly.

From Eq. (5.3), we can obtain the correlation matrix of the observation $\boldsymbol{Y}$ as

$$\mathbf{\Sigma} = \boldsymbol{E}\left(\boldsymbol{Y}\boldsymbol{Y}'\right) = \mathbf{\Lambda}\mathbf{\Phi}\mathbf{\Lambda}' + \boldsymbol{E}\left(\mathbf{\Psi}\mathbf{\Psi}'\right) \tag{5.5}$$

where $\mathbf{\Phi} = E\left(\boldsymbol{X}\boldsymbol{X}'\right)$, and $E(\cdot)$ and $(\cdot)'$ denotes expectation and transposition, respectively. The Eq. (5.5) is derived based on the assumption that the common factor and unique factor are uncorrelated which yields $E\left(\boldsymbol{X}\mathbf{\Psi}'\right) = 0$. Similarly, based on the uncorrelation assumption, $\boldsymbol{E}\left(\mathbf{\Psi}\mathbf{\Psi}'\right)$ can be substituted by a diagonal positive definite matrix $\mathbf{\Gamma^2}$. Therefore, Eq. (5.5) can be rewritten as

$$\mathbf{\Sigma} = \mathbf{\Lambda}\mathbf{\Phi}\mathbf{\Lambda}' + \mathbf{\Gamma^2}. \tag{5.6}$$

Without loss of generality, it is assumed that the latent factors $x_i(n)$ are orthogonal in the model. As a result $\mathbf{\Phi} = \boldsymbol{I}$. Then we subtract $\mathbf{\Gamma^2}$ from both sides of Eq. (5.6) to derive

$$\mathbf{\Sigma} - \mathbf{\Gamma^2} = \mathbf{\Lambda}\mathbf{\Lambda}'. \tag{5.7}$$

In this model, $\mathbf{\Sigma} - \mathbf{\Gamma^2}$ is called "the reduced correlation matrix" [49].

The next step is to determine both $\mathbf{\Gamma^2}$ and $\mathbf{\Lambda}$. Note that $\mathbf{\Gamma^2}$ is a diagonal matrix. If both $\mathbf{\Sigma}$ and $\mathbf{\Gamma^2}$ are known, then $\mathbf{\Lambda}$ can be estimated as $\mathbf{\Lambda} = \boldsymbol{A}\boldsymbol{D}^{\frac{1}{2}}$, where $\boldsymbol{A}$ is the eigenvector matrix and $\boldsymbol{D}$ is the diagonal eigenvalue matrix of the matrix $\mathbf{\Sigma} - \mathbf{\Gamma^2}$. On the other hand, if $\mathbf{\Lambda}$ has been estimated, then we can calculate $\mathbf{\Gamma^2}$ as

$$\mathbf{\Gamma^2} = \mathbf{\Sigma} - \mathbf{\Lambda}\mathbf{\Lambda}'. \tag{5.8}$$

Therefore, with an initial estimate of $\mathbf{\Gamma}^2$, the Eq. (5.7) can be solved iteratively where each iteration involves the following three steps:

1. Find the eigenvector and eigenvalue matrices $\boldsymbol{A}$ and $\boldsymbol{D}$ of "the reduced correlation matrix": $\mathbf{\Sigma} - \mathbf{\Gamma^2} = \boldsymbol{ADA'}$;

2. Find $\mathbf{\Lambda} = \boldsymbol{AD}^{\frac{1}{2}}$;

3. Find $\mathbf{\Gamma^2} = \mathbf{\Sigma} - \mathbf{\Lambda\Lambda'}$.

This procedure runs iteratively until the maximum difference of the last two round of $\mathbf{\Gamma^2}$ is less than certain small threshold [49].

Let $\boldsymbol{S} = \mathbf{\Sigma} - \boldsymbol{D}$, then $\mathbf{\Sigma} - \boldsymbol{S}^2$ will generate the unrotated factors matrix. Normally, we will pick up as latent factors those entries in $\boldsymbol{D}$ that are large enough, e.g., greater than 1. In practice, we may simply use principal component analysis [49] to estimate $\mathbf{\Lambda}$, which just considers the latent factors influencing the performance and ignores the unique factors.

Chapter 6: Case Study: Intelligence Measure of CR with Learning Capabilities

In this section, we present a case study consisting of different types of CRs. By designing a set of testing environments, we apply our methodology presented in Section 5.3 to derive the latent factors and analyze them as intelligence factors as well as cognitive capabilities contributing to the CR intelligence.

6.1 Settings

We consider a single hop scenario where there is only one CR and one PU. Therefore, we can focus on each CR's performance without considering channel contention. There are several channels in the network. The PU can appear on some or all of the channels simultaneously. We also assume a time slot based network. Figure 6.1 shows the time slot structure used by the CR.

As shown in the figure, the first part of the time slot is assigned for channel sensing. During this period, the CR senses the chosen channel and at the end of this period decides whether the channel is idle or not. If the CR finds the channel idle, it begins data transmission. Otherwise, it keeps silent to avoid interfering with the PU.

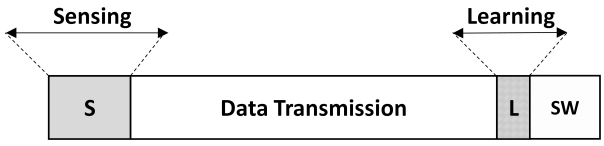


Figure 6.1: Time slot structure applied by the CR.

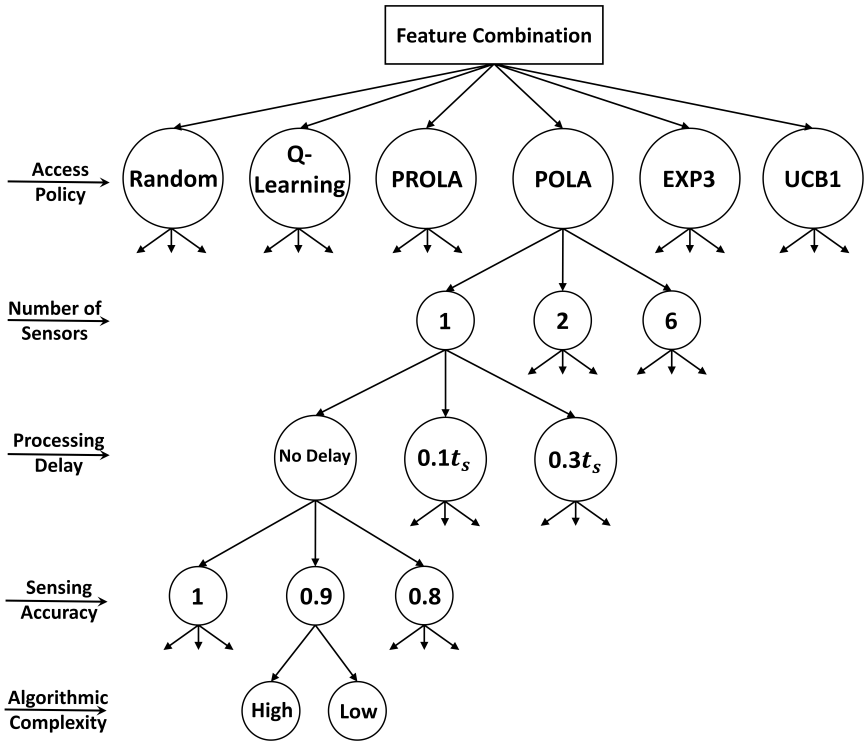


Figure 6.2: CRs consist of combinations of different features and parameters

During the third part of the time slot, the CR learns from its observation. No matter the channel was idle or busy, both of them provide useful information for the CR to learn and optimize its decisions in the future. The last part is the switching period which indicates the amount of time that it takes the CR to switch from one channel to another one. Switching period is dependent on the hardware limitations of each CR.

We have conducted extensive simulations with 144 different types of CRs. 10 channels are considered in the network. 18 testing scenarios are designed, such that each CR performs on each of them one by one. We run the simulations in MATLAB. For each CR performing in one single testing scenario we run the algorithm 10000 times and get the average.

6.2 Cognitive Radio Capabilities

Figure 6.2 shows the capabilities of CRs considered in this case study in terms of their features and parameters. Combinations of all these features gives us 144 different types of CRs as explained in the following. The CR features are described as follows.

- Channel access strategy (Access Policy) employed by the CR to learn and adapt to the environment. It can be a learning-based method, deterministic or just a random strategy. We consider five types of learning-based access strategies known as UCB1 [15], EXP3 [17], POLA[30], PROLA[30], and Q-Learning [44] and one random access strategy. Details of the strategies will be described in the sequel.

- Number of sensors. Possessing more sensors, the CR observes more channels at each time slot. Then depending on the reasoning it employs, the CR may adapt better to the environment. This is probably equal to higher loads in cognitive capabilities. In this case study, we consider the number of sensors (m) to be either 1, 2, or 6.

- Sensing accuracy which indicates the detecting probability when the PU is present. There are several methods of channel sensing including energy detection and feature extraction [54, 55, 56]. We consider three values of 1, 0.9, 0.8 as the probability of the correct sensing. The values are relatively large because in practice, the CRs usually have high sensing accuracy.

- Processing speed is another feature of a CR that occurs during sensing, learning, and switching parts of the time slot. Learning delay occurs due to two reasons, the hardware limitations and due to algorithmic complexity of the learning algorithm. We add up the delay due to hardware limitations that happen in different parts of a time slot as one single total delay. We assume this total delay to be either 0, $0.1t_s$, or $0.3t_s$ in which t_s indicates the time slot duration.

- Algorithmic complexity. The delay occurred due to the time required by the computations in the algorithmic side is different than the delay due to hardware limitations.

It depends on the efficiency of the learning algorithm and for this reason it is called algorithmic complexity. This type of delay depends on how well the learning algorithm has been designed algorithmic-wise and it is inherent to the learning technique.

As to the six channel access strategies we employ in this work, the random access strategy does not utilize any learning-based algorithm. The other learning based algorithms mentioned are described below.

Algorithm: UCB1 with multiple observations

Initialization: Play each machine once. Per each play make m observations including the played one. The m observations are made on the m subsequent actions beginning from the action played.

For each $t = 2, ..., T$: Play each machine that maximizes a given deterministic policy. The decision criteria is based on the upper confidence bound concept from statistics. Make m observations on the m subsequent channels beginning from the taken action.

The UCB1 and EXP3 algorithms [15, 17] are slightly modified from their original version for the case with m observations to address the more general case of observation of more than one channel. The modified UCB1 and EXP3 algorithms are described in the following. Note that UCB1 is a deterministic access policy designed for well behaved environments, while EXP3 is designed for adversarial environments.

Algorithm: EXP3 Algorithm with multiple observations

Initialization: Assign a uniform random distribution on action selection.

For each $t = 1, ..., T$

1. Update the distribution on action selection based on the observations made so far plus adding some randomness. Randomness is added to make sure the agent makes enough explorations.
2. Choose an action randomly based on the distribution defined above.
3. Observe the reward on m subsequent channels beginning from the taken action.
4. Update the observation history on all the channels. The observation history will be utilized in step one to optimize the channel selection distribution.

Algorithms POLA and the PROLA are presented below for convenience [30]. Both algorithms are designed for adversarial environments. PROLA as explained in Section 5.1 is similar to the EXP3 algorithm in the sense that at each time step, the agent is able to both gain reward and also to make an observation utilized in its learning process. The difference between PROLA and EXP3 is that in EXP3, the agent observes the reward on the same action it takes and gains reward; however, in PROLA, the agent makes observation on a channel other than the one it takes.

POLA is similar to the PROLA algorithm since both algorithms are designed to address the case when agent does not have the capability to observe the reward on the action it takes. However, POLA has a major difference from the PROLA and EXP3 based on which at each time step, it can either take action or make observation. This scenario, happens when the agent has limited capabilities and it cannot take action and switch to another channel for observation, during the periot of the same time step [30].

Algorithm: POLA with multiple Observations

Initialization: Assign uniform random distribution on the channels.

For each $t = 1, 2, ..., T$

1. With small probability ϵ decaying in time, choose an action uniformly at random to observe its reward. Otherwise, take an action.
2. If it is decided to make observation, choose m channels to observe then update the channel selection probability based on the channel observation history. Otherwise, choose a channel to access (take action) and accumulate the unobservable reward.

Algorithm : PROLA with multiple Observations

Initialization: Assign random uniform distribution on channel selection.

For each $t = 1, 2, ..., T$

1. Assign a distribution on action selection based on the channel observation history.
2. Choose a channel based on the above distribution to play and accumulate the unobservable reward.
3. Choose m channels other than the played one uniformly at random to observe their reward during the same time slot.
4. Update the channel observation history to optimize the distribution on channel selection policy.

The last learning algorithm we apply is Q-Learning algorithm [45] as described in the following Algorithm. Q-Learning is similar to the UCB1 algorithm in the sense that they

both are designed for well behaved environments. More specifically, Q-learning algorithm is usually applied in the environments that follow a Markovian Chain. One major difference between the Q-learning Algorithm and the UCB1 is that, Q-learning algorithm solves an optimization problem at each time step to optimize the action selection distribution.

In order to implement Q-Learning in MATLAB and to solve the optimization problems of this algorithm, CVX toolbox [57, 58] is used. More specifically, CVX toolbox is designed to solve convex optimization problems in MATLAB.

Considering all the combinations of the features as shown in Fig. (6.2), 162 different types of CRs are generated. However, for random access strategy, no learning capability is utilized. So the number of channels being observed makes no impact on the CR's performance. By removing eighteen redundant combinations, 144 CRs remain. Different features and their assigned values are shown in Fig. 6.2.

Algorithm 5: Q-Learning with multiple observations

Initialization: Assign a random uniform distribution on channel selection.

For each $t = 1, 2, ..., T$

1. With an small probability choose an action uniformly at random to play.
   Otherwise, choose an action with the distribution assigned based on the observation history.
2. Receive the reward on the action. Make $m - 1$ more observations on the subsequent channels other than the played one.
3. Use linear programming to optimize the action selection distribution.

6.3 Testing Scenarios

We consider several parameters to design the testing scenarios:

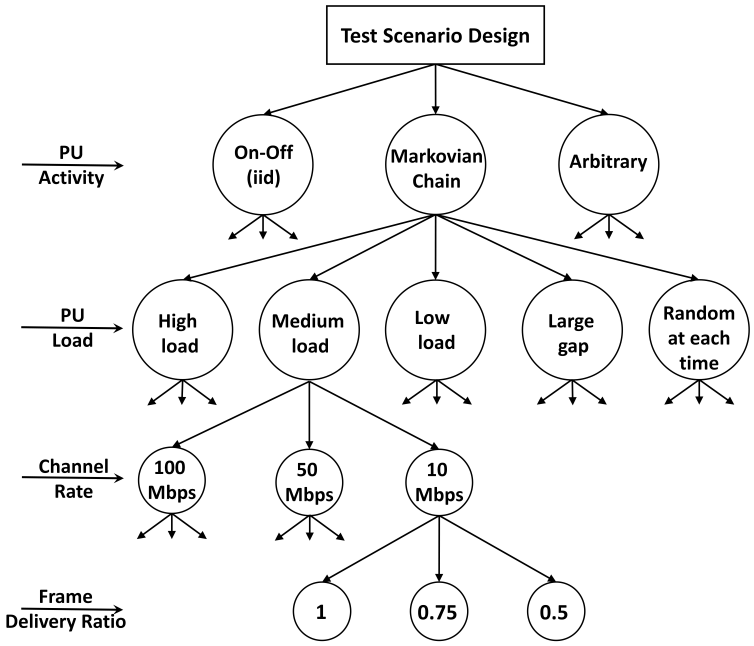


Figure 6.3: Designing Test Scenarios

- Type of PU Activity. We consider three types of activities for the PU which consists of i.i.d. distribution, Markovian Chain, and arbitrary where no well defined distribution exists.

- PU Load which indicates the probability of the PU to be active on each channel. PU may have a high load on all the channels or may have a light load on only one channel and a heavy load on all other channels (large gap). This testing scenario can discriminate among learning and nonlearning-based access strategies since by utilizing the observations and learning one can discriminate the good channel from low rewarding ones. We have considered several combinations of PU activity on the channels.

- Channel Rate. Three different values are considered as channel rates as shown in Fig. 6.3. If we assume all other characteristics of the channels to be identical, a CR that learns the high rate channel may be considered as having high load in the

corresponding cognitive capability.

- Frame Delivery Ratio (FDR) which includes the impact of channel quality and noise on a given channel. Three possible values for FDR are considered in this case study.

Figure 6.3 shows a summary of the parameters considered. Combining these parameters, we create 18 test scenarios. Each CR needs to perform on each testing scenario so that its cognitive capabilities can be derived.

6.4 Performance Metrics

We measure the performance of the CRs based on three different metrics:

- Throughput which is stored as $y_{1k}(n)$ where k and n indicate the testing scenario and the CR indices, respectively.
- Delay which indicates total delay occurred in the time slot and is stored as $y_{2k}(n)$.
- Violation ratio which represents the average number of times the CR interfered with the PU due to wrong sensing result called miss detection. It is assumed if the CR interferes with the PU, there will be a penalty for the CR and its data will be blocked, so there will be no throughput for the CR. Violation ratio is stored in $y_{3k}(n)$.

The performance measure data vector $\boldsymbol{Y}_k(n)$ is equal to $\boldsymbol{Y}_k(n) = [y_{1k}(n)\ y_{2k}(n)\ y_{3k}(n)]$ for $n = 1, \ldots, 144$ and $k = 1, \ldots, 18$.

6.5 Simulation Results

In this section we represent the simulation results, and analyze the intelligence factors as well as the cognitive capabilities of the CRs. We divide our simulations into several phases. In the first phase, we consider the UCB1, EXP3, and Random access based CRs. Associated with each of UCB1 and EXP3 policies, there are twenty-seven CRs according to Fig. 6.2. There are nine CRs utilizing the random access strategy.

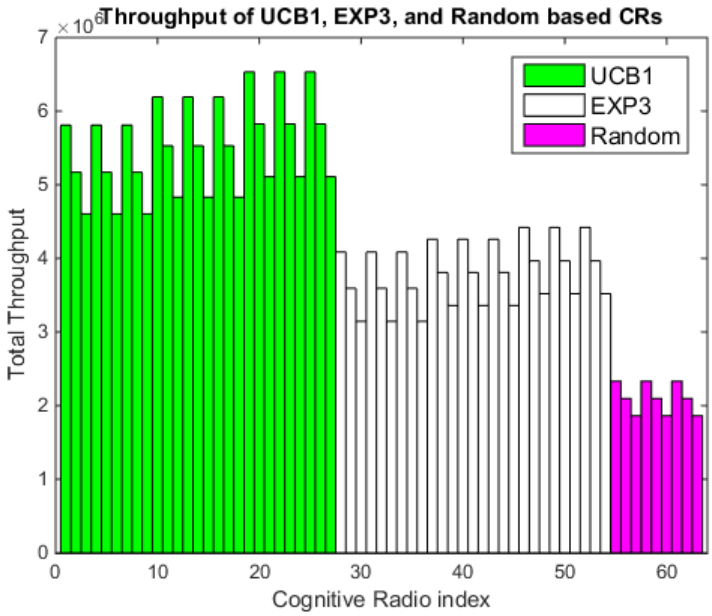


Figure 6.4: Total throughput of each CR achieved from all testing scenarios when the UCB1, EXP3, and Random access strategies are applied.

Figure 6.4 shows the simulation result of the first metric, throughput. This is the total throughput obtained by aggregating the throughput achieved from all the testing scenarios for each CR applying the mentioned access strategies.

From this figure, three clusters can be identified. The first cluster (for cognitive radio index 1 to 27) represents CRs employing UCB1 learning-based access strategy. The second cluster (for cognitive radio index 28 to 54) belongs to the CRs employing EXP3 learning-based access strategy. The last cluster (for cognitive radio index 55 to 63) represents CRs utilizing random access strategies.

One observation is that, within each cluster, as the number of sensors increases, the overall throughput increases as well. Next, the total throughput of CRs employing UCB1 is higher than those employing EXP3 since most of the testing scenarios designed are well behaved (stochastic) in which UCB1 performs better [15, 17]. The third cluster illustrates those CRs employing random access methods. Since random strategy never utilizes the

previous observations, it achieves the lowest throughput among others. The graphs also show that for each three consecutive CRs (i.e., three consecutive bars in the graph), the throughput is decreasing since the sensing accuracy is decreasing.

In the next step, we conduct data analysis via FA. From the simulations, three 63×18 matrices are generated for three metrics we measure. They all together create the data matrix $\boldsymbol{Y}$ with the dimension of 63×54. FA is applied on this matrix using the software IBM SPSS [59].

The analysis identifies four latent factors as shown in Fig. 6.5. Only four factors are distinguishable and the rest are negligible which are almost zero. Due to limited space we skip the detailed output data corresponding to the FA results. Even though the number of latent factors are identified, it is not yet clear which cognitive capabilities these factors correspond to. We need to examine the data thoroughly and find out the corresponding cognitive capabilities by matching them to the CR functions.

By examining the data, the four latent factors (cognitive capabilities) are found as follows: Spectrum sensing capability, processing speed capability, environment recognition capability, and environment adaptation capability. The results are summarized in the first four rows of the Table 6.1.

As we study the results achieved by applying FA technique, the data of the first factor provides information on the violation ratio which is impacted by the sensing accuracy and the number of sensors. As a result we conclude that the first latent factor corresponds to the spectrum sensing capability. The second latent factor addresses the delay, which is associated with the processing speed capability due to the hardware limitations of the CR. The third factor is related to the learning capability, or specifically the environment recognition capability. The forth factor shows a better performance for EXP3 and random access strategy than the UCB1 when the sensing accuracy decreases. The same thing happens when the environment is not well behaved. This indicates that the EXP3 and random access strategy adapt better to non-well behaved environments. The reason is because they utilize randomness in their access strategy which makes them more resilient

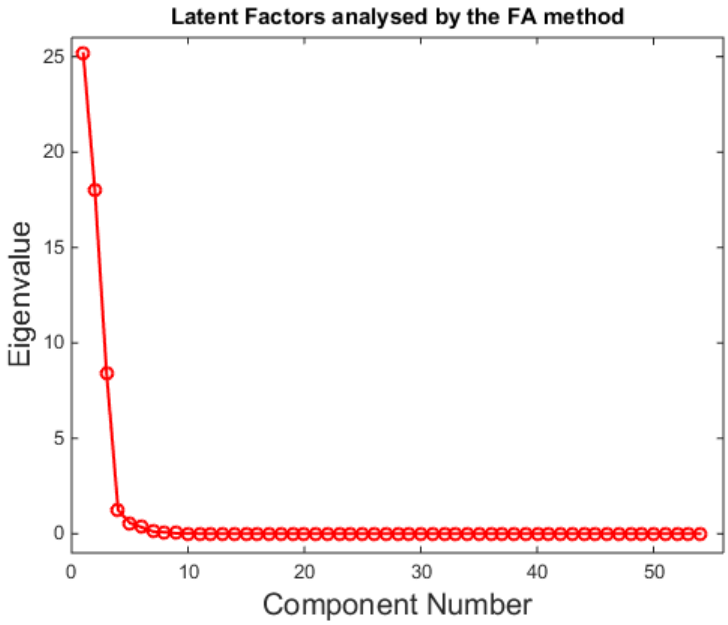


Figure 6.5: Latent factors identified for the UCB1, EXP3, and random-access based CRs based on the three metrics of throughput, delay, and violation ratio.

to changes in the environment. Deterministic based approaches assume a stable environment which makes them vulnerable to modifications in the environment. As a result this latent factor addresses the environment adaptation capability.

Comparing to the intelligence model proposed in Section 5.2, the processing speed capability matches the broad cognitive capability G_p, the spectrum sensing matches G_s, and the two others correspond to G_c or G_f as shown in Table 6.1. In addition, all the CRs used in this case-study have high load on the G_{sm} factor.

Next, we plot the components obtained through the analysis. Component plot shows how the scenarios in the case study belong to each of the four latent factors. Since it is not possible to plot four dimensional figures, we plot the components for factors 1, 2 and 3 as shown in Fig. 6.6. The whole data is divided into three clusters, each corresponding to one latent factor.

In order to get a deeper insight from the results, we also apply the FA technique to

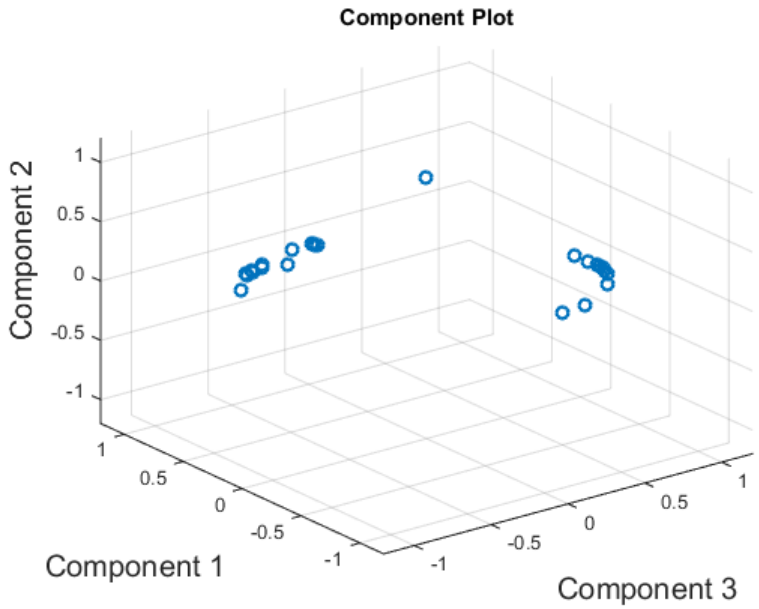


Figure 6.6: Component plot of the latent factors achieved by applying FA on all the three metrics.

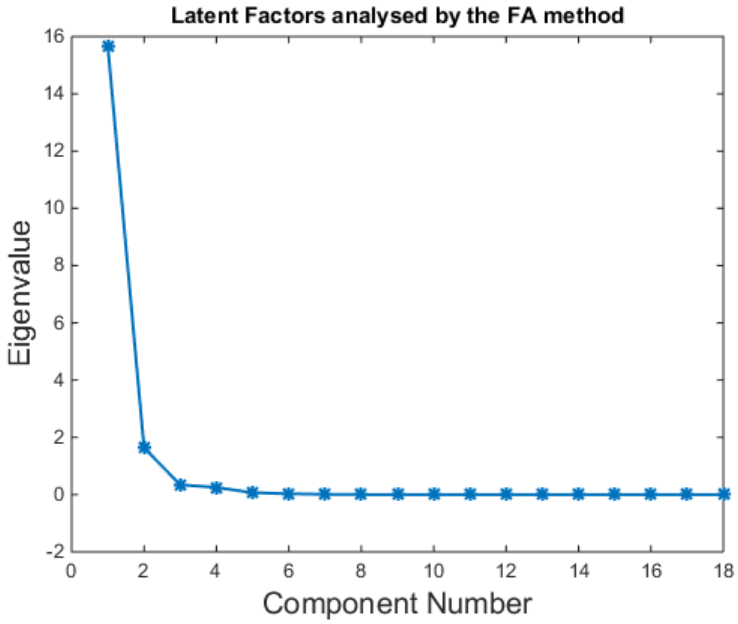


Figure 6.7: Latent factors identified for the UCB1, EXP3, and random-access based CRs based on only one metric, throughput.

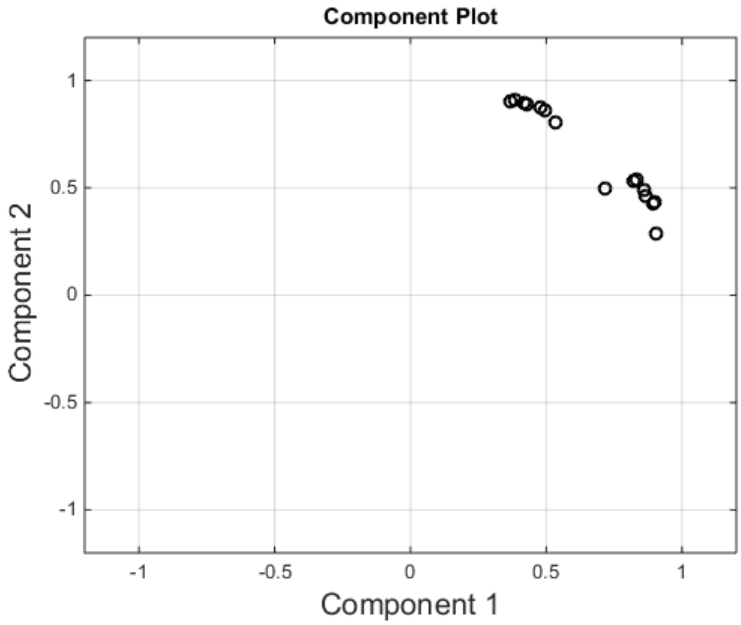


Figure 6.8: Component plot of the latent factors achieved by applying FA on the throughput metric.

Table 6.1: Latent factors identified that contribute to intelligence

Factor I	Sensing Capability, G_s
Factor II	Processing Speed Capability, G_p
Factor III	Environment Recognition Capability, G_c or G_f
Factor IV	Environment Adaptation Capability, G_c or G_f
Factor V	Algorithmic Processing Time, G_a

only one of the performance metrics called throughput. In this case which is a limited case than the previous one, only two factors are identified as shown in Fig. 6.7. One of them corresponds to the learning capability and the other one corresponds to the environment adaptation capability. Figure 6.8 shows the components of the analyzed data in which the whole data is divided into two clusters, each corresponding to one latent factor.

In the next phase of our simulation, we add the rest of the learning based CRs applying

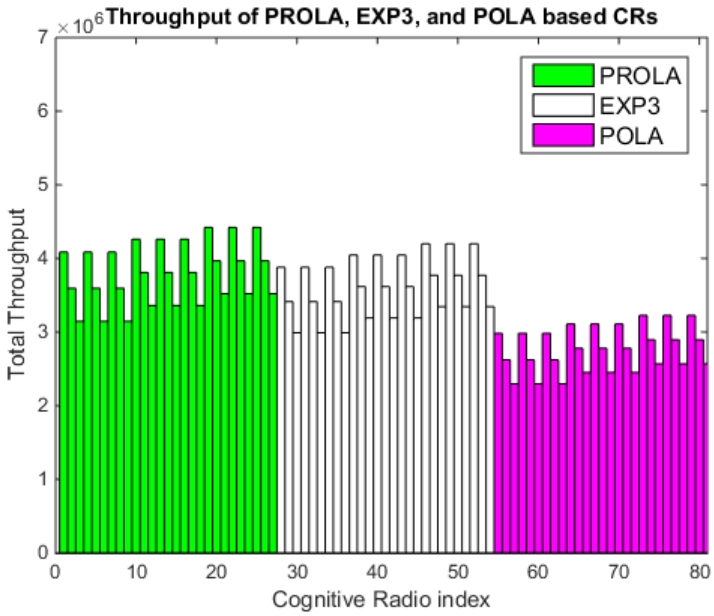


Figure 6.9: Total throughput of each CR achieved from all testing scenarios when the PROLA, EXP3, and POLA access strategies are applied.

POLA, PROLA, and Q-Learning to the ones we considered earlier to make a comprehensive list of CRs with different capabilities. Each of the 144 CRs performs in the testing scenarios one by one. Three performance metrics are measured. This means that three matrices are generated, each with a dimension of 144×18. The combination of these matrices results in the data matrix $\boldsymbol{Y}$ with dimension 144×54.

As shown in Fig. 6.9, the performance of the PROLA is similar to the performance of the EXP3. Algorithmic wise, the only difference between these two algorithms is that in EXP3, the agent observes the reward on the same action it takes; while in the PROLA, the agent makes an observation on one other action different than the one it takes. Our analysis shows that the cognitive capabilities of the PROLA is almost the same as the ones for EXP3. All the three algorithms are designed for the non-stochastic environments. As shown in the figure the POLA algorithm achieves a lower throughput compared to the two others. This is because the POLA algorithm is not able to take action and make observation simultaneously at each time step. Instead, it decides at each time step to do either of

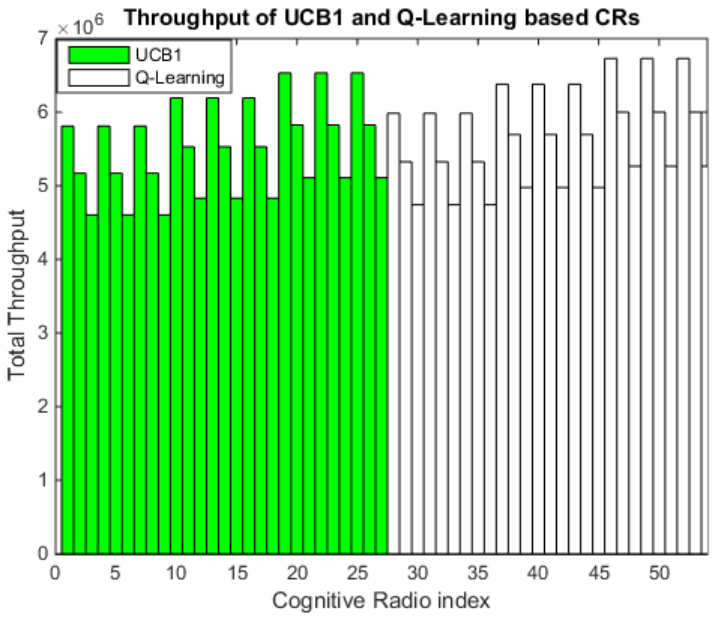


Figure 6.10: Total throughput of each CR achieved from all testing scenarios when the UCB1, Q-Learning access strategies are applied.

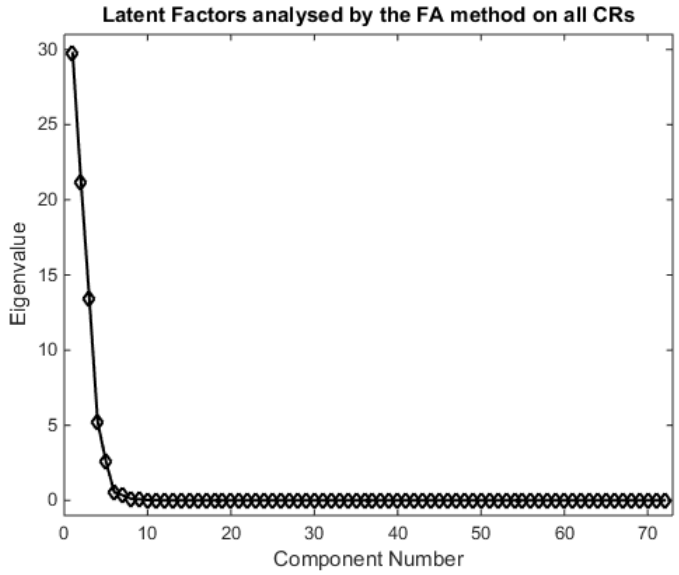
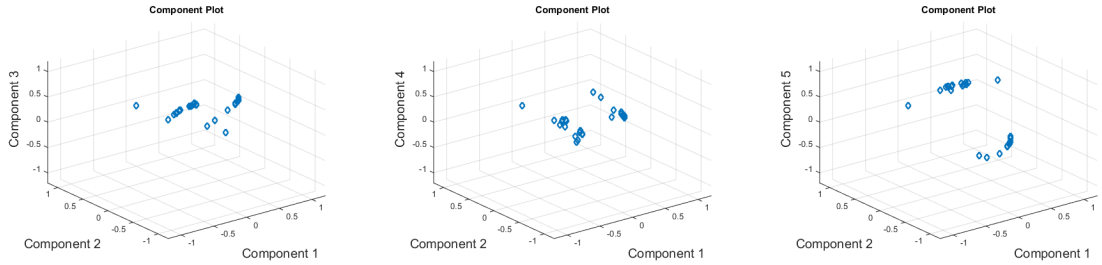


Figure 6.11: Latent factors identified considering all the CRs based on the three metrics of throughput, delay, and violation ratio.



(a) Latent factors one, two, and three (b) Latent factors one, two, and four (c) Latent factors one, two, and five

Figure 6.12: Component plot of all five latent factors achieved by applying FA on all the three metrics.

them. This leads to a lower environment recognition capability and as a result POLA has a lower load in this cognitive capability compared to others. In contrast, EXP3 and PROLA demonstrate almost equal loads with respect to this cognitive capability. This indicates that non-stochastic based online learning algorithms do not necessarily demonstrate the same cognitive capabilities and should not be categorized into the same group.

Similarly, Fig. 6.10 shows the performance comparison of Q-learning and UCB1. These two algorithms are both designed for stochastic environments. As deterministic algorithms, they do not consider randomness in their policies. Our results indicate that both algorithms show high loads in the cognitive capability of environment recognition. However, their environment adaptability cognitive capability is low. Q-Learning demonstrates low load in the cognitive capability of algorithmic processing. This is because at each time slot, in order to update the action policy, the Q-learning algorithm solves an optimization problem. In contrast, the UCB1 algorithm updates action policy at each time slot by a simple sum and multiplication operations.

Finally, we derive the latent factors as shown in Fig. 6.11. Five cognitive factors are identified with the fifth factor as the algorithmic processing time. Table 6.1 shows the whole list of factors identified in our case study.

We also show the component plot for the whole data set used in this case study in

Fig. 6.12. Since there are five latent factors, the component plot is five dimensional. In order to represent the five dimensional data, we fix two of the latent factors, then plot three figures considering third, fourth, and fifth latent factors, respectively.

Table 6.2: Cognitive Radios each with a different capability

Index	Strategy	Sensors	Total Delay	Accuracy	Factor 1	Factor 2	Factor 3	Factor 4
1	UCB1	1	0	1	-0.99854	-1.06053	0.72124	-0.67929
2	UCB1	1	0	0.9	0.07582	-1.06053	0.42578	-1.09318
3	UCB1	1	0	0.8	1.19144	-1.06053	0.22652	-1.35620
4	UCB1	1	0.1 t_s	1	-0.99854	-0.26513	0.72124	-0.67929
5	UCB1	1	0.1 t_s	0.9	0.07582	-0.26513	0.42578	-1.09318
6	UCB1	1	0.1 t_s	0.8	1.19144	-0.26513	0.22652	-1.35620
7	UCB1	1	0.3 t_s	1	-0.99854	1.32566	0.72124	-0.67929
8	UCB1	1	0.3 t_s	0.9	0.07582	1.32566	0.42578	-1.09318
9	UCB1	1	0.3 t_s	0.8	1.19144	1.32566	0.22652	-1.35620
10	UCB1	2	0	1	-0.91662	-1.06053	1.30529	0.37787
11	UCB1	2	0	0.9	0.19451	-1.06053	0.91472	-0.51420
12	UCB1	2	0	0.8	1.35054	-1.06053	0.54353	-1.27140
13	UCB1	2	0.1 t_s	1	-0.91662	-0.26513	1.30529	0.37787
14	UCB1	2	0.1 t_s	0.9	0.19451	-0.26513	0.91472	-0.51420
15	UCB1	2	0.1 t_s	0.8	1.35054	-0.26513	0.54353	-1.27140
16	UCB1	2	0.3 t_s	1	-0.91662	1.32566	1.30529	0.37787
17	UCB1	2	0.3 t_s	0.9	0.19451	1.32566	0.91472	-0.51420
18	UCB1	2	0.3 t_s	0.8	1.35054	1.32566	0.54353	-1.27140
19	UCB1	6	0	1	-0.91619	-1.06053	1.98175	2.26214
20	UCB1	6	0	0.9	0.23026	-1.06053	1.54077	1.20849
21	UCB1	6	0	0.8	1.43294	-1.06053	1.11334	0.13636
22	UCB1	6	0.1 t_s	1	-0.91619	-0.26513	1.98175	2.26214
23	UCB1	6	0.1 t_s	0.9	0.23026	-0.26513	1.54077	1.20849
24	UCB1	6	0.1 t_s	0.8	1.43294	-0.26513	1.11334	0.13636
25	UCB1	6	0.3 t_s	1	-0.91619	1.32566	1.98175	2.26214
26	UCB1	6	0.3 t_s	0.9	0.23026	1.32566	1.54077	1.20849
27	UCB1	6	0.3 t_s	0.8	1.43294	1.32566	1.11334	0.13636
28	EXP3	1	0	1	-1.32402	-1.06053	-0.40954	-0.46258
29	EXP3	1	0	0.9	-0.12030	-1.06053	-0.59928	-0.16071
30	EXP3	1	0	0.8	1.11229	-1.06053	-0.73378	0.26967
31	EXP3	1	0.1 t_s	1	-1.32402	-0.26513	-0.40954	-0.46258
32	EXP3	1	0.1 t_s	0.9	-0.12030	-0.26513	-0.59928	-0.16071
33	EXP3	1	0.1 t_s	0.8	1.11229	-0.26513	-0.73378	0.26967
34	EXP3	1	0.3 t_s	1	-1.32402	1.32566	-0.40954	-0.46258
35	EXP3	1	0.3 t_s	0.9	-0.12030	1.32566	-0.59928	-0.16071
36	EXP3	1	0.3 t_s	0.8	1.11229	1.32566	-0.73378	0.26967
37	EXP3	2	0	1	-1.29286	-1.06053	-0.30537	-0.52965
38	EXP3	2	0	0.9	-0.09329	-1.06053	-0.44370	-0.30205
39	EXP3	2	0	0.8	1.12099	-1.06053	-0.57601	-0.03410
40	EXP3	2	0.1 t_s	1	-1.29286	-0.26513	-0.30537	-0.52965
41	EXP3	2	0.1 t_s	0.9	-0.09329	-0.26513	-0.44370	-0.30205
42	EXP3	2	0.1 t_s	0.8	1.12099	-0.26513	-0.57601	-0.03410
43	EXP3	2	0.3 t_s	1	-1.29286	1.32566	-0.30537	-0.52965
44	EXP3	2	0.3 t_s	0.9	-0.09329	1.32566	-0.44370	-0.30205
45	EXP3	2	0.3 t_s	0.8	1.12099	1.32566	-0.57601	-0.03410
46	EXP3	6	0	1	-1.26007	-1.06053	-0.18426	-0.52123
47	EXP3	6	0	0.9	-0.07156	-1.06053	-0.32114	-0.38695
48	EXP3	6	0	0.8	1.11776	-1.06053	-0.45533	-0.23831
49	EXP3	6	0.1 t_s	1	-1.26007	-0.26513	-0.18426	-0.52123
50	EXP3	6	0.1 t_s	0.9	-0.07156	-0.26513	-0.32114	-0.38695
51	EXP3	6	0.1 t_s	0.8	1.11776	-0.26513	-0.45533	-0.23831
52	EXP3	6	0.3 t_s	1	-1.26007	1.32566	-0.18426	-0.52123
53	EXP3	6	0.3 t_s	0.9	-0.07156	1.32566	-0.32114	-0.38695
54	EXP3	6	0.3 t_s	0.8	1.11776	1.32566	-0.45533	-0.23831
55	Random	1	0	1	-1.61363	-1.06053	-1.54870	-0.27258
56	Random	1	0	0.9	-0.28221	-1.06053	-1.58403	1.09547
57	Random	1	0	0.8	1.06272	-1.06053	-1.61180	2.47243
58	Random	1	0.1 t_s	1	-1.61363	-0.26513	-1.54870	-0.27258
59	Random	1	0.1 t_s	0.9	-0.28221	-0.26513	-1.58403	1.09547
60	Random	1	0.1 t_s	0.8	1.06272	-0.26513	-1.61180	2.47243
61	Random	1	0.3 t_s	1	-1.61363	1.32566	-1.54870	-0.27258
62	Random	1	0.3 t_s	0.9	-0.28221	1.32566	-1.58403	1.09547
63	Random	1	0.3 t_s	0.8	1.06272	1.32566	-1.61180	2.47243

Chapter 7: Conclusions and Future Work

7.1 Security in CRN

In this thesis, we studied the optimal online learning algorithms that can be applied by a PUE attacker without any prior knowledge of the primary user activity characteristics and secondary user channel access policies. We formulated the PUE attack as an online learning problem. We identified the uniqueness of PUE attack that a PUE attacker cannot observe the reward on the attacking channel, but is able to observe the reward on another channel. We proposed a novel online learning strategy called POLA for the attacker with no observation capability in the attacking time slot. In other words, this algorithm is suitable when simultaneous attack and observation is not possible within the same time slot. POLA dynamically decides between attack and observation, then chooses a channel for the decision made. POLA achieves a regret in the order of $\tilde{\Theta}(\sqrt[3]{T^2})$. We showed POLA's optimality by matching its regret upper and lower bounds. We then proposed another online learning algorithm, EXP3-DO to dynamically choose attacking and observing channels for a PUE attacker in order to minimize its regret. EXP3-DO is based on the existing theoretical frameworks and it is regret in the order of $O(T^{\frac{2}{3}})$. We proposed a third novel online learning algorithm called PROLA for an attacker with at least one observation capabilities. For such an attacker, the attack period is followed by an observation period during the same time slot. PROLA introduces a new theoretical framework under which the agent achieves an optimal regret in the order of $\tilde{\Theta}(\sqrt{T})$.

One important conclusion of our study is that with no observation at all in the attacking slot in the POLA case, the attacker loses on the regret order, and with the observation of at least one channel in the PROLA case, there is a significant improvement on the performance

of the attacker. This is in contrast to the case where increasing the number of observations from one to $m \geq 2$, does not make that much difference, only improving the constant factor. Though, this observation can be utilized to study the approximate number of observations required to get the minimum constant factor. The attacker's regret upper bound has a dependency on the number of observations m as $\sqrt{1/m}$. That is, the regret decreases overall for an attacker with higher observation capability (larger m). However, when the number of observing channels is small, the regret decreases more if we add a few more observing channels. While, the decreased regret will become marginal when more observing channels are added. This finding implies that an attacker may only need a small number of observing channels to achieve a good constant factor. The regret upper bound also is proportional to $\sqrt{K \ln K}$ which means the regret increases when there are more channels in the network. The proposed optimal learning algorithm, PROLA, also advances the study of online learning algorithms. It deals with the situation where a learning agent cannot observe the reward on the action that is taken but can partially observe the reward of other actions. Before our work, the regret upper bound is proved to be in the order of $\tilde{O}(\sqrt[3]{T^2})$. Our algorithm achieves a tighter regret bound of $\tilde{O}(\sqrt{T})$ by randomizing the observing actions which introduces the concept of time-variable random feedback graphs. We show this algorithm's optimality by deriving its regret lower bound which matches with its upper bound.

As for future work, we believe that our work can serve as a stepping stone to study many other problems. How to deal with multiple attackers will be interesting, especially when the attackers perform in a distributed manner. One other interesting direction is to study the equilibrium between the PUE attacker(s) and secondary user(s) when both of them employ learning based algorithms. Integrating non-perfect spectrum sensing and the ability of PUE attack detection into our model will also be interesting especially that the PUE attacker may interfere with the PU during spectrum sensing period which can make it detectable by the PU. From theoretical point of view, our result shows only one possible case that the feedback graph despite being partially observable, achieves a tighter bound. A

particular reward structure may allow for $\tilde{O}(\sqrt{T})$ regret even in partially observable graphs.

7.2 Intelligence in CR

In the second part of this study, for the first time, we have proposed the idea of deriving the intelligence measure and analyzing the cognitive capabilities of the CR. An intelligence model is proposed for the CR, and a data-driven methodology is proposed which applies FA techniques to identify CR intelligence factors and cognitive capabilities. A case study is presented in which through extensive simulations, five latent factors are identified for the CR that comply well with the nature of the tested CRs.

Our ongoing effort is focused on measuring the intelligence quotient (IQ) for each CR. IQ can be considered as the general intelligence factor that indicates how well a CR performs in uncertain environments. We will also expand our methods to measure CR intelligence in multi-user and multi-hop networks. More specifically, the following can be considered as future research directions to pursue.

7.2.1 Cognitive Capabilities of Routing Algorithms

Our current work is on the intelligence measure of CRs while they act in the MAC layer. Intelligence measure of CRs in the routing layer is an interesting future research direction. There has been some preliminary work done on the learning-based routing methods [60], where the authors try to answer the question of whether machine learning including deep reinforcement learning can replace the traditional network protocol design. It is shown that data driven based routing methods that extract information from the traffic history achieve better performance. For any learning based routing algorithm designed for cognitive radio networks, we can measure their intelligence and cognitive capabilities, similarly. This leads to designing better routing algorithms and better network configurations to maximize network throughput while minimizing costs.

7.2.2 Item Response Theory and IQ Measure

After extracting intelligence factors and identifying cognitive capabilities of CRs, the next step would be to combine these capabilities and assign a quantitative value to it called Intelligence Quotient (IQ). This is in fact the unique general intelligence factor g in Stratum III shown in Fig. 5.1. In order to do so, one needs to first make sure that the test scenarios are comprehensive and standardized. In other words, the testing items shown in Fig. 5.2 should include all types of test scenarios from easy to hard ones. Item Response Theory (IRT) [61] which is a design, analysis, and scoring paradigm for tests, is the tool that needs to be used to quantify the easy and difficult test scenarios. Using IRT to design the optimal test scenarios and to develop the IQ measurement methods is another interesting future research direction.

7.2.3 Configuring the Network with Combination of CRs with Different Intelligence

As explained in the introduction, cognitive radio networks can be configured by integrating CRs with different intelligence and cognitive capabilities. This may lead to the optimal use of resources and would also be more cost efficient. More comprehensive research is needed in order to quantitatively measure the performance of such networks and to rigorously show how one or a few number of CRs with higher intelligence can lead and network with other CRs with lower intelligence.

Appendix A: POLA Algorithm's Regret Lower Bound

Proof of Theorem 2. We sketch the proof as follows. The problem of the PUE attacker with no observation capability within the attacking time slot can be modeled as a feedback graph. Feedback graphs are introduced in [14] based on which the observations governing the actions are modeled as graphs. More specifically, in a feedback graph, the nodes represent the actions and the edges connecting them demonstrate the observations made associated with taking a specific action. Figure A.1 shows how our problem can be modeled as a feedback graph.

In this figure, nodes 1 to K represent the overall number of actions. There are K more actions however in this figure. These K extra actions are required to be able to model our problem with feedback graphs. Actions $K+1$ up to $2K$ represent the observations; i.e., any time the agent decides to make an observation, it is modeled as an extra action. There are K channels and we model any observation on each channel with a new action which adds up to $2K$ actions overall. The agent gains a reward of zero if it chooses the observation action since it is not a real action. Instead, it makes an observation on the potential reward of its associated real action.

So, this problem can be modeled as a feedback graph and it in fact turns out to be a partially observable graph [14]. In [14], it is proved that the regret lower bound for partially observable graphs is $\Omega(\iota\sqrt[3]{KT^2})$ which completes the proof. □

Appendix B: PROLA Algorithm's Regret Lower Bound

The proof here is similar to the lower bound analysis of EXP3 given in proof of Theorem 5.1, Appendix, in [17]. We mention the differences here. For this analysis the problem setup is exactly the same as the one in [17]. The notations and definitions are also the same as given in the first part of the analysis in [17]. Below we bring some of the important notations and definitions used in the analysis here to keep the clarity of our analysis and

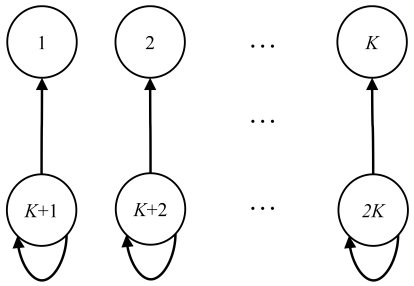


Figure A.1: Modeling the Attacker with no observation capability within the attacking slot with a feedback graph

to make our analysis self contained; however, the reader is referred to [17] for more details on the definitions of notations.

The reward distribution on the actions are defined as follows. One action is chosen uniformly at random to be the good action. Below is the reward distribution on the good action, i for all $t = 1, ..., T$,

$$x_t(i) = \begin{cases} 1, & 1/2 + \epsilon \\ 0, & o.w., \end{cases} \tag{B.1}$$

where $\epsilon \in (0, 1/2]$. The reward distribution on all other actions is defined to be one or zero with equal probabilities. $\mathbf{P}_* \{\cdot\}$ is used to denote the probability w.r.t. this random choice of rewards to play. $\mathbf{P}_i \{\cdot\}$ represents the probability conditioned on i being the good action. Also, $\mathbf{P}_{unif} \{\cdot\}$ shows the probability with respect to uniformly random choice of rewards on all actions. $\mathbf{O}$ notation is also used to denote the probability w.r.t. observation. Analogous observation probability $\mathbf{O}_* \{\cdot\}$, $\mathbf{O}_i \{\cdot\}$, $\mathbf{O}_{unif} \{\cdot\}$ and expectation notations, $\mathbf{E}_*[\cdot]$, $\mathbf{E}_i[\cdot]$, $\mathbf{E}_{unif}[\cdot]$ are used.

The agent's access/attack policy is denoted by A. $r_t = x_t(i_t)$ is a random variable which its value shows the reward gained at time t. $\mathbf{r}^t =< r_1, \ldots, r_t >$ is a sequence of rewards

received till t and $\mathbf{r}$ is the entire sequence of rewards. $G_A = \sum_{t=1}^{T} r_t$ is the gain of the agent and $G_{max} = \max_j \sum_{t=1}^{T} x_t(j)$. The number of times action i is chosen by A is a random variable denoted by N_i.

Lemma 1 *Let* $f : \{0,1\}^T \longrightarrow [0, M]$ *be any function defined on reward sequences* $\mathbf{r}$. *Then for any action* i,

$\mathbf{E}_i[f(\mathbf{r})] \leq \mathbf{E}_{unif}[f(\mathbf{r})] + \frac{M}{2}\sqrt{-\mathbf{E}_{unif}[\mathbf{O}_i]\ln(1-4\epsilon^2)}$.

Proof.

$$\begin{aligned}
\mathbf{E}_i[f(\mathbf{r})] - \mathbf{E}_{unif}[f(\mathbf{r})] &= \sum_{\mathbf{r}} f(\mathbf{r})(\mathbf{O}_i\{\mathbf{r}\} - \mathbf{O}_{unif}\{\mathbf{r}\}) \\
&\leq \sum_{\mathbf{r}:\mathbf{O}_i\{\mathbf{r}\} \geq \mathbf{O}_{unif}\{\mathbf{r}\}} f(\mathbf{r})(\mathbf{O}_i\{\mathbf{r}\} - \mathbf{O}_{unif}\{\mathbf{r}\}) \\
&\leq M \sum_{\mathbf{r}:\mathbf{O}_i\{\mathbf{r}\} \geq \mathbf{O}_{unif}\{\mathbf{r}\}} (\mathbf{O}_i\{\mathbf{r}\} - \mathbf{O}_{unif}\{\mathbf{r}\}) \\
&= \frac{M}{2}\left\|\mathbf{O}_i - \mathbf{O}_{unif}\right\|_1 . \qquad \text{(B.2)}
\end{aligned}$$

We also know from [17, 62] that,

$$\left\|\mathbf{O}_{unif} - \mathbf{O}_i\right\|_1^2 \leq (2\ln 2) KL(\mathbf{O}_{unif} \,||\, \mathbf{O}_i). \qquad \text{(B.3)}$$

From chain rule for relative entropy we derive the following,

$$\begin{aligned}
KL(\mathbf{O}_{unif} \,||\, \mathbf{O}_i) &= \sum_{t=1}^{T} KL(\mathbf{O}_{unif}\left\{r_t|\mathbf{r}^{t-1}\right\} \,||\, \mathbf{O}_i\left\{r_t|\mathbf{r}^{t-1}\right\}) \\
&= \sum_{t=1}^{T} (\mathbf{O}_{unif}\{i_t \neq i\}\, KL(\frac{1}{2} \,||\, \frac{1}{2}))
\end{aligned}$$

$$+ \left(\mathbf{O}_{unif}\left\{i_t = i\right\} KL(\frac{1}{2} \,||\, \frac{1}{2} + \epsilon)\right)$$

$$= \sum_{t=1}^{T} \mathbf{O}_{unif}\left\{i_t = i\right\} (-\frac{1}{2}\lg(1 - 4\epsilon^2))$$

$$= \mathbf{E}_{unif}[\mathbf{O}_i](-\frac{1}{2}\lg(1 - 4\epsilon^2)). \tag{B.4}$$

The lemma follows by combining (B.2),(B.3), and (B.4). □

Proof of Theorem 4. The rest of the analysis in this part is similar to the analysis in Theorem A.2 in [17], except that when we apply lemma 1 to N_i, we reach the following inequality,

$\mathbf{E}_i[N_i] \leq \mathbf{E}_{unif}[N_i] + \frac{T}{2}\sqrt{-\mathbf{E}_{unif}[\mathbf{O}_i]\ln(1 - 4\epsilon^2)}$.

where $\sum_{i=1}^{K} \sqrt{\mathbf{E}_{unif}[\mathbf{O}_i]} \leq \sqrt{KT}$. By considering the observation probability and making a little simplification the upper bound is achieved. Following similar steps as in Theorem A.2 in [17] for the rest of the analysis gives us the regret lower bound equal to $\omega(c\sqrt{KT})$ which completes the proof. □

Bibliography

[1] A. Alipour-Fanid, M. Dabaghchian, N. Wang, P. Wang, L. Zhao, and K. Zeng, "Machine learning-based delay-aware UAV detection over encrypted Wi-Fi traffic," in *to appear in IEEE CNS - International Workshop on Cyber-Physical Systems Security (CPS-SEC) 2019*, 2019.

[2] A. Alipour-Fanid, M. Dabaghchian, H. Zhang, and K. Zeng, "String stability analysis of cooperative adaptive cruise control under jamming attacks," in *2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE)*, Jan 2017, pp. 157–162.

[3] "Report and order and second further notice of proposed rulemaking, federal communications commission 15-47 gn docket no. 12-354," 2015.

[4] I. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "A survey on spectrum management in cognitive radio networks," *Communications Magazine, IEEE*, vol. 46, no. 4, pp. 40–48, April 2008.

[5] Z. Gao, H. Zhu, S. Li, S. Du, and X. Li, "Security and privacy of collaborative spectrum sensing in cognitive radio networks," *Wireless Communications, IEEE*, vol. 19, no. 6, pp. 106–112, 2012.

[6] Q. Yan, M. Li, T. Jiang, W. Lou, and Y. Hou, "Vulnerability and protection for distributed consensus-based spectrum sensing in cognitive radio networks," in *INFOCOM, 2012 Proceedings IEEE*, March 2012, pp. 900–908.

[7] H. Li and Z. Han, "Dogfight in spectrum: Combating primary user emulation attacks

in cognitive radio systems; part ii: Unknown channel statistics," *Wireless Communications, IEEE Transactions on*, vol. 10, no. 1, pp. 274–283, January 2011.

[8] ——, "Blind dogfight in spectrum: Combating primary user emulation attacks in cognitive radio systems with unknown channel statistics," in *Communications (ICC), 2010 IEEE International Conference on*, May 2010, pp. 1–6.

[9] M. Dabaghchian, A. Alipour-Fanid, K. Zeng, and Q. Wang, "Online learning-based optimal primary user emulation attacks in cognitive radio networks," in *Communications and Network Security, 2016 CNS 2016. IEEE*, Oct 2016.

[10] R. Chen, J.-M. Park, and K. Bian, "Robust distributed spectrum sensing in cognitive radio networks," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, April 2008, pp. –.

[11] R. Chen, J.-M. Park, and J. Reed, "Defense against primary user emulation attacks in cognitive radio networks," *Selected Areas in Communications, IEEE Journal on*, vol. 26, no. 1, pp. 25–37, Jan 2008.

[12] H. Li and Z. Han, "Dogfight in spectrum: Combating primary user emulation attacks in cognitive radio systems, part i: Known channel statistics," *Wireless Communications, IEEE Transactions on*, vol. 9, no. 11, pp. 3566–3577, November 2010.

[13] M. Clark and K. Psounis, "Can the privacy of primary networks in shared spectrum be protected?" in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, April 2016, pp. 1–9.

[14] N. Alon, N. Cesa-Bianchi, O. Dekel, and T. Koren, "Online learning with feedback graphs: Beyond bandits," in *Proceedings of The 28th Conference on Learning Theory*, ser. Proceedings of Machine Learning Research, vol. 40. Paris, France: PMLR, 03–06 Jul 2015, pp. 23–35. [Online]. Available: http://proceedings.mlr.press/v40/Alon15.html

[15] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Mach. Learn.*, vol. 47, no. 2-3, pp. 235–256, May 2002. [Online]. Available: http://dx.doi.org/10.1023/A:1013689704352

[16] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "Gambling in a rigged casino: The adversarial multi-armed bandit problem," in *Foundations of Computer Science, 1995. Proceedings., 36th Annual Symposium on*, Oct 1995, pp. 322–331.

[17] ——, "The nonstochastic multiarmed bandit problem," *SIAM J. Comput.*, vol. 32, no. 1, pp. 48–77, Jan. 2003. [Online]. Available: http://dx.doi.org/10.1137/S0097539701398375

[18] K. Wang, Q. Liu, and L. Chen, "Optimality of greedy policy for a class of standard reward function of restless multi-armed bandit problem," *Signal Processing, IET*, vol. 6, no. 6, pp. 584–593, August 2012.

[19] Y. Gai and B. Krishnamachari, "Distributed stochastic online learning policies for opportunistic spectrum access," *Signal Processing, IEEE Transactions on*, vol. 62, no. 23, pp. 6184–6193, Dec 2014.

[20] M. Dabaghchian, S. Liu, A. Alipour-Fanid, K. Zeng, X. Li, and Y. Chen, "Intelligence measure of cognitive radios with learning capabilities," in *2016 IEEE Global Communications Conference (GLOBECOM)*, Dec 2016, pp. 1–6.

[21] Q. Wang and M. Liu, "Learning in hide-and-seek," *IEEE/ACM Transactions on Networking*, vol. PP, no. 99, pp. 1–14, 2015.

[22] M. Dabaghchian, A. Alipour-Fanid, S. Liu, K. Zeng, X. Li, and Y. Chen, "Who is smarter? Intelligence measure of learning-based cognitive radios," *CoRR*, vol. abs/1712.09315, 2017. [Online]. Available: http://arxiv.org/abs/1712.09315

[23] H. Su, Q. Wang, K. Ren, and K. Xing, "Jamming-resilient dynamic spectrum access

for cognitive radio networks," in *Communications (ICC), 2011 IEEE International Conference on*, June 2011, pp. 1–5.

[24] Q. Wang, K. Ren, and P. Ning, "Anti-jamming communication in cognitive radio networks with unknown channel statistics," in *19th IEEE International Conference on Network Protocols*, 2011, pp. 393–402.

[25] Q. Wang, P. Xu, K. Ren, and X. y. Li, "Delay-bounded adaptive UFH-based anti-jamming wireless communication," in *INFOCOM, 2011 Proceedings IEEE*, April 2011, pp. 1413–1421.

[26] K. Ezirim, S. Sengupta, and E. Troia, "Multiple channel acquisition and contention handling mechanisms for dynamic spectrum access in a distributed system of cognitive radio networks," in *Computing, Networking and Communications (ICNC), 2013 International Conference on*, Jan 2013, pp. 252–256.

[27] P. Lin and T. Lin, "Optimal dynamic spectrum access in multi-channel multi-user cognitive radio networks," in *21st Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, Sept 2010, pp. 1637–1642.

[28] Y. Liao, T. Wang, K. Bian, L. Song, and Z. Han, "Decentralized dynamic spectrum access in full-duplex cognitive radio networks," in *2015 IEEE International Conference on Communications (ICC)*, June 2015, pp. 7552–7557.

[29] H. Li and Z. Han, "Dogfight in spectrum: Combating primary user emulation attacks in cognitive radio systems; part ii: Unknown channel statistics," *Wireless Communications, IEEE Transactions on*, vol. 10, no. 1, pp. 274–283, January 2011.

[30] M. Dabaghchian, A. Alipour-Fanid, K. Zeng, Q. Wang, and P. Auer, "Online learning with randomized feedback graphs for optimal PUE attacks in cognitive radio networks," *IEEE/ACM Transactions on Networking*, vol. 26, no. 5, pp. 2268–2281, Oct 2018.

[31] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Computer Networks*, vol. 50, no. 13, pp. 2127 – 2159, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128606001009

[32] Q. Zhao and B. M. Sadler, "A survey of dynamic spectrum access," *IEEE Signal Processing Magazine*, vol. 24, no. 3, pp. 79–89, May 2007.

[33] K. G. Shin, H. Kim, A. W. Min, and A. Kumar, "Cognitive radios for dynamic spectrum access: from concept to reality," *IEEE Wireless Communications*, vol. 17, no. 6, pp. 64–74, December 2010.

[34] M. J. Marcus, "Spectrum policy for radio spectrum access," *Proceedings of the IEEE*, vol. 100, no. Special Centennial Issue, pp. 1685–1691, May 2012.

[35] S. D. R. Forum, "Cognitive radio definition and nomenclature," *SDRF-06-P-0009*, 2008.

[36] I. C. A. Organization, "Potential for radio frequency interference to aeronautical surveillance systems for new terrestrial communications," *Montreal, Canada*, Apr. 2012.

[37] A. Alipour-Fanid, M. Dabaghchian, and K. Zeng, "Platoon stability and safety analysis of cooperative adaptive cruise control under wireless Rician fading channels and jamming attacks," *CoRR*, vol. abs/1709.10128, 2017. [Online]. Available: https://arxiv.org/abs/1710.08476

[38] D. W. Matolak, "Unmanned aerial vehicles: Communications challenges and future aerial networking," in *2015 International Conference on Computing, Networking and Communications (ICNC)*, Feb 2015, pp. 567–572.

[39] A. Alipour-Fanid, M. Dabaghchian, N. Wang, P. Wang, L. Zhao, and K. Zeng, "Machine learning-based delay-aware UAV detection and operation mode identification over encrypted Wi-Fi traffic," in *arXiv*, 2019.

[40] V. C. Gungor, D. Sahin, T. Kocak, S. Ergut, C. Buccella, C. Cecati, and G. P. Hancke, "Smart grid technologies: Communication technologies and standards," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 4, pp. 529–539, Nov 2011.

[41] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, Feb 2014.

[42] K. S. Mcgrew, "Editorial: Chc theory and the human cognitive abilities project: Standing on the shoulders of the giants of psychometric intelligence research," *Intelligence*, p. 10, 2009.

[43] H. KESTELMAN, "The fundamental equation of factor analysis," *British Journal of Statistical Psychology*, vol. 5, no. 1, pp. 1–6, 1952. [Online]. Available: http://dx.doi.org/10.1111/j.2044-8317.1952.tb00106.x

[44] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3, pp. 279–292, May 1992. [Online]. Available: https://doi.org/10.1007/BF00992698

[45] M. L. Littman, "Markov games as a framework for multi-agent reinforcement learning," in *Proceedings of the Eleventh International Conference on International Conference on Machine Learning*, ser. ICML'94. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1994, pp. 157–163. [Online]. Available: http://dl.acm.org/citation.cfm?id=3091574.3091594

[46] J. J. Thompson, K. M. Hopkinson, and M. D. Silvius, "A test methodology for evaluating cognitive radio systems," *IEEE Transactions on Wireless Communications*, vol. 14, no. 11, pp. 6311–6324, Nov 2015.

[47] A. Hess, F. Malandrino, N. J. Kaminski, T. K. Wijaya, and L. A. DaSilva, "Cognitive radio algorithms coexisting in a network: Performance and parameter sensitivity," *IEEE Transactions on Cognitive Communications and Networking*, vol. 2, no. 4, pp. 381–396, Dec 2016.

[48] R. J. Sternberg and S. B. Kaufman, Eds., *The Cambridge Handbook of Intelligence*. Cambridge University Press, 2011, cambridge Books Online. [Online]. Available: http://dx.doi.org/10.1017/CBO9780511977244

[49] S. A. Mulaik, *Foundations of factor analysis*. CRC press, 2009.

[50] Y. Zhao, S. Mao, J. O. Neel, and J. H. Reed, "Performance evaluation of cognitive radios: Metrics, utility functions, and methodology," *Proceedings of the IEEE*, vol. 97, no. 4, pp. 642–659, April 2009.

[51] N. Patwari and S. K. Kasera, "Crawdad utah CIR measurements."

[52] S. Rehman, T. Turletti, and W. Dabbous, "A roadmap for benchmarking in wireless networks," Technical Report, INRIA 2011.

[53] G. Jourjon, T. Rakotoarivelo, C. Dwertmann, and M. Ott, "Executable paper challenge: Labwiki: an executable paper platform for experiment-based research," Procedia Computer Science, 2011.

[54] S. Atapattu, C. Tellambura, and H. Jiang, "Energy detection based cooperative spectrum sensing in cognitive radio networks," *IEEE Transactions on Wireless Communications*, vol. 10, no. 4, pp. 1232–1241, April 2011.

[55] W. Zhang, R. K. Mallik, and K. B. Letaief, "Optimization of cooperative spectrum sensing with energy detection in cognitive radio networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 12, pp. 5761–5766, December 2009.

[56] W. l. Chin, H. c. Kuo, and H. h. Chen, "Features detection assisted spectrum sensing in

wireless regional area network cognitive radio systems," *IET Communications*, vol. 6, no. 8, pp. 810–818, May 2012.

[57] M. C. Grant and S. P. Boyd, "Graph implementations for nonsmooth convex programs," in *Recent Advances in Learning and Control*, V. D. Blondel, S. P. Boyd, and H. Kimura, Eds. London: Springer London, 2008, pp. 95–110.

[58] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014.

[59] IBM SPSS. [Online]. Available: http://www.ibm.com/analytics/us/en/technology/spss/

[60] A. Valadarsky, M. Schapira, D. Shahaf, and A. Tamar, "A machine learning approach to routing," *CoRR*, vol. abs/1708.03074, 2017. [Online]. Available: http://arxiv.org/abs/1708.03074

[61] S. P. R. Susan E. Embretson, *Item Response Theory for Psychologists.* LEA, 2000.

[62] T. M. Cover and J. A. Thomas, *Elements of Information Theory.* Wiley, 2006.

Biography

Monireh Dabaghchian graduated from Alzahra High School, Tabriz, Iran, in 2001. She received her Master and Bachelor of Electrical Engineering from University of Tabriz in 2009, and 2006, respectively. She was employed as a teaching faculty in University College of NabiAkram from 2009 till January 2013.