SEQUENTIAL LEARNING FOR PASSIVE MONITORING OF MULTI-CHANNEL WIRELESS NETWORKS

A Thesis
Presented to
the Faculty of the Department of Electrical and Computer Engineering
University of Houston

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
in Electrical Engineering

by
Thanh Dang Le

May 2013

SEQUENTIAL LEARNING FOR PASSIVE MONITORING OF MULTI-CHANNEL WIRELESS NETWORKS

Thanh Dang Le

Approved:

Chair of the Committee
Dr. Zhu Han, Associate Professor
Electrical and Computer Engineering

Committee Members:

Dr. Rong Zheng, Associate Professor
Computer Science

Dr. Saurabh Prasad, Assistant Professor
Electrical and Computer Engineering

Dr. Suresh K. Khator, Associate Dean,
Cullen College of Engineering

Dr. Badrinath Roysam, Professor and Chairman,
Electrical and Computer Engineering

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my advisor, Dr. Zhu Han, for his guidance and continual support during the course of my degree. Working with him was a wonderful experience, his constructive advice and constant encouragement that he shared during my stay at the University of Houston has been invaluable.

I would also like to thank my co-advisor, Dr. Rong Zheng for her wise knowledge, insightful comments, and valuable discussions. She contributed significantly to both my research and my professional development. I am also honored to have Dr. Saurabh Prasad as a committee member. Your suggestions and accessibility were helpful for my thesis proposal and defense.

During my graduate studies at the University of Houston, I have had the pleasure of meeting my labmates, who have helped me directly or indirectly in completing my studies and have made my MS a rewarding experience. I owe my thanks to you, friends.

I am indebted to my parents and my younger brother who have been a constant source of support and love throughout my time in the United States and my life. Thank you for everything.

SEQUENTIAL LEARNING FOR PASSIVE MONITORING OF MULTI-CHANNEL WIRELESS NETWORKS

An Abstract
of a
Thesis
Presented to
the Faculty of the Department of Electrical and Computer Engineering
University of Houston

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
in Electrical Engineering

by
Thanh Dang Le
May 2013

Abstract

With the requirement for increasing efficiency of wireless spectrum usage, the cognitive radio technique has been emerging as an important solution. Passive monitoring over wireless channels in cognitive radio is an innovative approach in which the system attempts to locate channels with the highest activity over time. A huge amount of work has been contributed to this field when the reward of each channel is identical to observers. However, when the reward is different over observers, these algorithms perform poorly. In this thesis, we challenge this problem by considering this correlation as part of the reward. We develop one optimal online learning algorithm when a switching cost exists in the system. We also propose three approximation algorithms with competitive computation complexity but still guarantee to obtain a constant amount of reward compared to the optimal case. Theoretical analysis and simulation are conducted to prove the effectiveness of these approaches.

Table of Contents

List of Figures

List of Algorithms

Chapter 1

Introduction and Background

As the number of wireless devices increases, there emerges a major concern, the running out of frequency spectrum for wireless devices. This problem can be solved by applying cognitive radio technique, in which secondary users can use the channel when licensed users have no activity. In this thesis, we concentrate on passively monitoring the activity of licensed users over channels so that the system can make a choice based on the information provided.

1.1 Sequential Learning and Multi-Armed Bandit Problem

Sequential learning, in which wireless monitoring is an active branch, first came from neuro-scientists who studied the human's ability of learning things from sequences with order. This observation caught the attention of scientists, particularly those in computer science when Lashley [1] stated that sequential learning is not attributable to sensory feedback and that there are plans for behavior since the nervous system prepares for some behaviors but not others. This idea opened a new way of thinking for both neuro-scientists and computer scientists doing research on sequential learning. In computer science, sequential learning has been long studied by [2], [3], [4], [5].

Wireless monitoring in sequential learning is a technique where a dedicated set of hardware devices, called *sniffers*, are used to monitor activities in wireless networks. These devices capture transmissions of users' activity of interference source in their vicinity, and store packet level in trace files, which can be analyzed distributive or at a central location. Since most, if not all, infrastructure networks utilize multiple contiguous or non-contiguous channels or bands, an important issue is to determine which set of frequency bands each sniffer operates on to maximize the total amount of information gathered. This is called the *sniffer-channel assignment* problem or *channel assignment* problem for short.

This is a challenging problem. First, the system usually does not have enough devices to

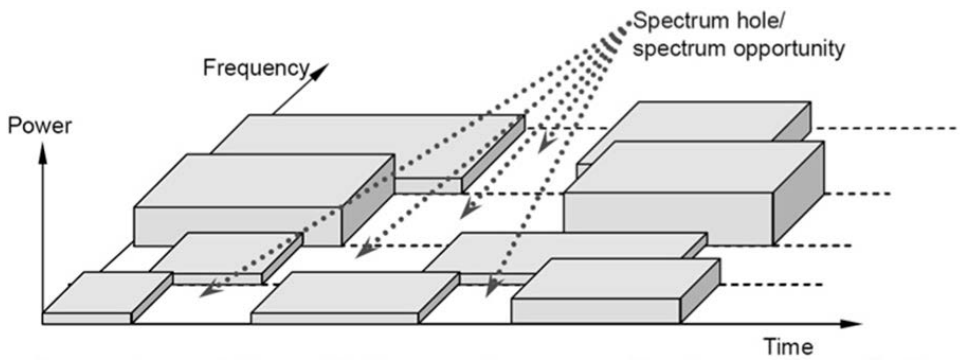

Figure 1.1: Objective of cognitive radio technique

monitor all resources at any time. Second, there is no *priori* - the knowledge of usage patterns or the likelihood of occurrence of interesting events. Therefore, the system needs to balance between exploring channels that are under-sampling and assigning sniffers to busiest ones with current knowledge. This trade-off is closely related to the multi-armed bandit problem (MAB) [2] [6]. In MAB problems, a gambler must choose one arm over N non-identical slot machines to play in a sequence of trials so as to maximize his payoff. At any time, the gambler chooses an arm based on past information. The efficiency of the policy that he uses can be measured in term of its associated *regret*, which is defined as the difference between the expected payoff gained by a "genie" who always uses the optimal stationary arm, and that obtained by a given policy. The regret achieved by a policy is evaluated in terms of its growth over time and how it scales with respect to the various problem parameters.

A large volume of work has been devoted to designing good strategies for variations of the MAB problem and to the understanding of the theoretical limits of such procedures. Lai *et al.* [7] established logarithmic upper and lower bounds for dent stochastic arms with parametric pay-off distributions.

While Agrawal [8] considered a class of sample-mean based policies, Auer *et al.* [9] analyzed upper confidence bound (UCB) based and ε-Greedy policies which both have the regret of $O(\log T)$ over time. Bandit problems with linear parameterized payoff were studied extensively in [10] [11] [12]. However, when we have no statistical assumptions about the payoff of the slot machines

(*adversarial environment*), Auer *et al.* proposed the well-known EXP3 set of algorithms [13] which are built up based on [14] and [15], and in turn is a variant of Littlestone and Warmuth's [16] weighted majority algorithm, and Vovk's [17] aggregating strategies. The EXP3 policy achieves a theoretical growth of regret $O(T^{1/2})$ over time with T is the total time that the algorithm is planning to played.

Recently, [18] considered the MAB problem in an unknown environment and proposed a policy with its regret achieves $O(\sqrt{nK}\log^{3/2}(n)\log K)$ in adversarial model and $O(\frac{K}{\Delta}\log^2(n)\log K)$ in stochastic model. MAB with switching costs was first considered in [19]. An excellent survey on MAB with switching costs can be found in [20]. Beside traditional multi-armed bandit problems, [21] proposed their algorithm when the set of arm is infinite and satisfies the metric space, and the payoff function satisfies a Lipschitz condition with respect to the metric.

1.2 Passive Monitoring of Multi-Channel Wireless Networks

In the field of wireless monitoring, from system-level point of view, [22], [23], [24], [25], [26], [27] attempted to design complete systems, and addressed the interactions among the components of such systems. The authors have argued both qualitatively and quantitatively the need for wireless side monitoring. Different from [28], [29] or [30] in which the certain statistics regarding the users' activity are given or can be inferred, our work has no assumption about such parameters.

In this case, a sequential approach should be applied to balance the trade-off between *exploration* and *exploitation*. In exploration phase, sniffers are assigned to channels with less information to gain further knowledge, while in exploitation; they are set to channels with highest traffic based on the gathered information of the system. If the system only concentrates on discovering the environment, it suffers the regret by not assigning its sniffers to the best channels to capture the traffic. However, if the exploration process is not enough, the system may choose a suboptimal choice to assign their sniffers. This trade-off is vividly illustrated by the famous multi-armed bandit problem (MAB).

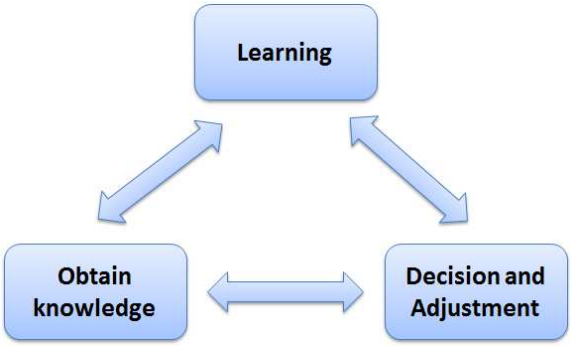

Figure 1.2: Characteristics and capabilities of of wireless monitoring system

Realizing the connection between the MAB and spectrum access in cognitive radio networks, [31] considered the problem of secondary user channel selection as the distributed multi-armed bandit problem, and presented a policy that achieved asymptotically logarithmic regret in time. Lai *et al.* applied the UCB1 algorithm [9] to single user-channel selection in [32], and later extended it to consider Markovian payoffs and for the case of multiple users in [33]. Two policies for distributed learning and access with order-optimal cognitive system throughput under self-play were proposed in [34]. In centralized model, [35] proposed two algorithms without switching costs which are proven to achieve logarithmic regret over time compared to an offline optimal solution. A distributed algorithm based on Gibbs sampler is proposed in [36]. Shin *et al.* [37] extended their earlier work and propose DA-OSCA, a distributed algorithm for the channel assignment problem while preserving the same approximation ratio as the centralized algorithm. It adapts to the changes in the network by determining the arrival and departure of sniffers/nodes.

In our problem, each of the S sniffers must be assigned to one over the set of K channels to monitor so as to maximize the total information gathered. Therefore, the total number of arms available each round is thus $N = K^S$. We assume that the payoff of each arm is proportional to the number of distinct users detected. For simplicity, we assume that a user's activity in a given channel can be described with a sequence of independent and identically distributed (i.i.d) Bernoulli random variables. However, as opposed to the standard MAB problem, the observation upon a single assignment is not only the reward associated with the assignment, but also the activity patterns observed

at each monitored channel. Note that the observed pattern may have correlated components, e.g., when two sniffers observe the transmission of the same set of users. The objective of our work is to design algorithms to assign sniffers to channels so as to maximize the probability of observing users' activity over time.

1.3 Contribution and Organization of This Thesis

In our thesis, we formulate the channel assignment problem as a multi-agent multi-arm partial information problem with linearly parameterized payoff. By formulating as such, two sniffers may observe the same set of users when they sniff the same channel. Hence, in the first part of our thesis, we formulate this correlation part as a visible set of rewards and propose one optimal online learning policy - our Upper Confidence Bound (UCB)-based algorithm. This algorithm achieves the theoretical logarithmic regret, but it is NP-hard, which is high computation complexity. From this point of view, an approximate online learning algorithm - ε-GREEDY-APPROX is designed to not only obtain a constant ratio of reward, but also take advantage in the computation complexity compare to the optimal algorithm.

By including the correlation reward between sniffers in the problem, the computation complexity of the system is increased. To reduce this amount of time, in the second part, we attempt to achieve approximate solutions without considering the correlation information. Following this direction, our algorithms have shorter running time but still keep a competitive performance compare to the previous two algorithms. These new algorithms without the correlation reward can be seen as multi-agent systems [38], in which all the agents work together to find its best assignment, but each one has different information about its surrounded area. We evaluate the trade-off between optimality, computation cost and rate of learning between algorithms from the simulation results of all algorithms we proposed.

The first algorithm using the multi-agent approach is the EXP3-APPROX. In this algorithm, channels seen by each agent are modeled as adversarial environment. With this assumption, we

apply the idea of MAB algorithms under adversarial setting [13] and obtain an $O(\sqrt{T})$ regret, and an out-performed computation time compare to the first two algorithms. This algorithm is later extended to the distributed version - DEXP3-APPROX - with the idea of having no central processing unit in real systems. However, by loosening the constraint of the environment, the regret of EXP3-APPROX has low convergence speed compare to previous algorithms. Under the observation of agents, we realize that channels are distorted at the beginning, but become "more stable" after each agent converges to its optimal assignment. As a result, we devise ε-GREEDY-AGENT-APPROX with the idea of using an stochastic algorithm for each agent when channel condition seen by agents is "stable enough" to be seen as nearly stochastic environment.

Finally, from what we have done so far, we summarize them into this thesis with the structure as follows. In Chapter 2, we formulate our channel assignment problem. Details, analysis, simulation results and conclusion for all the algorithms taking response of the correlation reward are introduced in Chapter 3, and the other policies are in Chapter 4. We also do here a small scale implementation for the wireless sniffing using both adversarial and stochastic algorithms in Chapter 5. Last but not least is our conclusion for this thesis and the future direction in Chapter 6.

Chapter 2

Problem Formulation

The objective of this chapter is to cast the problem that we are going to solve into mathematic model. The main idea of this thesis is to design algorithms to assign sniffers to channels with highest users' activity. In order to do that, we first model the offline problem in Section 2.1, in which a set of sniffers needs to sniff through all available channels to determine the best assignments (sniffer to channel) which contains the set of user with highest weights. Due to the property of the wireless monitoring problem, we have a high probability of observing the same set of users monitored by two or more sniffers when they are sniffing in the same channel. Therefore, in Section 2.2, we describe our tool to learn this correlation reward which is later used in Chapter 3. Section 2.3 formulates the optimal online learning problem with switching cost. Finally, we sketch a general view about the approximate online learning algorithms in section 2.4 and take a deeper look at them in both Chapter 3 and 4.

2.1 Offline Optimization Problem

We consider S sniffers monitoring user activities in K channels. A user u operates in one of K channels, $c(u) \in \mathcal{K} = \{1, ..., K\}$. Let p_u denote the transmission probability of user u. We represent the relationship between users and sniffers using an undirected bi-partite graph $G = (\mathcal{S}, U, E)$, where $\mathcal{S} = \{1, ..., S\}$ is the set of sniffer nodes and U is the set of users. An edge $e = (s, u)$ exists between sniffer $s \in \mathcal{S}$ and user $u \in U$ if s can capture the transmission from u. A channel assignment decision (or action) consists of an (unordered) set of S tuples $\mathbf{k} = \{\langle i_1, k_{i_1} \rangle, \langle i_2, k_{i_2} \rangle, ..., \langle i_S, k_{i_S} \rangle\}$, where the i_j^{th} sniffer is assigned the $k_{i_j}^{th}$ channel. The utility (of payoff) of an assignment is the average amount of user activity it monitors.

We formulate the optimal sniffer-channel assignment where the graph G and the user-activity probability $(p_u; u \in U)$ are both known. The objective here is to maximize the expected number

of activity users monitored. We denote MAX-EFFORT-COVER (MEC) the problem of finding the largest (weight) set of users that can be monitored by a set of sniffers, where each sniffer can monitor one of a set of K channels. The MEC problem can be cast as the following integer program (IP):

$$\begin{array}{lll} \max & \sum_{u \in U} p_u y_u & \\ \text{s.t.} & \sum_{k=1}^{K} z_{s,k} \leq 1 & \forall s \in S \\ & y_u \leq \sum_{s \in N(u)} z_{s,c(u)} & \forall u \in U \\ & y_u, z_{s,k} \in \{0,1\} & \forall u, s, k. \end{array} \tag{2.1}$$

In (2.1), each sniffer is associated with a set of binary decision variables, $z_{s,k} = 1$ if the sniffer is assigned to channel k; 0, otherwise. Further, y_u is a binary variable indicating whether or not user u is monitored, and p_u is the weight associated with user u. MEC has been proven to be NP-hard in [28].

2.2 Reward Learning Tool

In the online setting, in order to exploit the correlation among sniffer observations, MEC is reformulated and casted as a linearized multi-arm bandit (MAB) problem [35].

Consider *channel assignments* of sniffers to channels, $\mathbf{k} = (\langle 1, k_1 \rangle, \ldots, \langle S, k_S \rangle)$ (abbreviated as $(k_1, \ldots, k_S)$), where $1 \leq k_i \leq K$. Let $\mathbb{K} = \left\{ \mathbf{k} \mid \mathbf{k} : \mathcal{S} \to \{1, .., K\}^S \right\}$ be the set of all possible assignments. Let $U_{ik}(t)$ be a nonnegative, integer-valued random variable that denotes the index of the user whose activity sniffer i can observe in channel k at time t, or which takes the value of zero if there is no activity in the chosen channel. The instantaneous feedback (observations) received under the joint action $\mathbf{k}(t) = (k_1, \ldots, k_S))$ is $Y^{\circ}_{(k_1,\ldots,k_S)} = (U_{1,k_1}(t), U_{2,k_2}(t), \ldots, U_{S,k_S}(t))$. Note that $\mathbb{I}_{\left\{U_{i_1,k_{i_1}} = U_{i_2,k_{i_1}} = \ldots = U_{i_s,k_{i_1}} > 0\right\}}$ is a function of $Y^{\circ}_{(k_1,\ldots,k_S)}$, and hence can be taken as part of the observation. Thus, we define $Y_{(k_1,\ldots,k_S)}$ as the collection

$$\mathbb{I}_{\left\{U_{i_1,k_{i_1}} = U_{i_2,k_{i_1}} = \ldots = U_{i_s,k_{i_1}} > 0\right\}} \qquad 1 \leq s \leq S,\ 1 \leq i_1 < \ldots < i_s \leq S. \tag{2.2}$$

Note that spatial multiplexing is allowed such that multiple users can be active at the same time in one channel (as long as they are sufficiently far apart geographically). However, we assume

one sniffer can observe one user at a time. This is consistent with many existing multiple access mechanisms including FDMA and TDMA. As in (2.1), the payoff upon selecting the joint action is the number of distinct users that the sniffers observe. That is, the joint payoff of $\mathbf{k} = (k_1, k_2, \ldots, k_p)$ is

$$\begin{aligned} X_{\mathbf{k}}(t) &= |\{U_{1,k_1}(t), \ldots, U_{S,k_S}(t)\}| - \mathbb{I}_{\{U_{1,k_1}(t)=0,\ldots,U_{S,k_S}(t)=0\}} \\ &= \sum_{i=1}^{S} \mathbb{I}_{\{U_{1,k_i}(t)>0\}} \\ &- \sum_{i,j=1}^{S} \mathbb{I}_{\left\{U_{i,k_i}(t)=U_{j,k_j}(t)>0\right\}} \mathbb{I}_{\{k_i=k_j, i\neq j\}} \\ &\ldots \\ &- (-1)^S \mathbb{I}_{\left\{U_{1,k_1}(t)=U_{2,k_2}(t)=\ldots=U_{S,k_S}(t)>0\right\}} \times \mathbb{I}_{\{k_1=k_2=\ldots=k_S\}}. \end{aligned} \tag{2.3}$$

The expectation of joint payoff for channels $\mathbf{k} = (k_1, k_2, \ldots, k_S)$ is given by,

$$\begin{aligned} \mathbb{E}[X_{\mathbf{k}}(t)] &= \sum_{i=1}^{S} \mathbb{P}(U_{1,k_i}(t) > 1) \\ &- \sum_{i,j=1}^{S} \mathbb{P}(U_{i,k_i}(t) = U_{j,k_j}(t) > 0) \mathbb{I}_{\{k_i=k_j, i\neq j\}} \\ &\ldots \\ &- (-1)^S \mathbb{P}(U_{1,k_1}(t) = \ldots = U_{S,k_S}(t) > 0) \times \mathbb{I}_{\{k_1=k_2=\ldots=k_S\}}. \end{aligned} \tag{2.4}$$

Define an unknown vector θ with the following elements:

$$\begin{array}{rl} \mathbb{P}(U_{i,k} > 0), & 1 \leq i \leq S, 1 \leq k \leq K, \\ \mathbb{P}(U_{i_1,k} = U_{i_2,k} > 0), & 1 \leq i_1 < i_2 \leq S, 1 \leq k \leq K, \\ & \vdots \\ \mathbb{P}(U_{1,k} = U_{2,k} = \ldots = U_{S,k} > 0), & 1 \leq k \leq K. \end{array} \tag{2.5}$$

We introduce the "arm features," $\phi_{\mathbf{k}} \in \mathbb{R}^M$ as (2.6), where $M = K(2^S - 1)$. Note that the arm feature $\phi_{\mathbf{k},j}$ of the jth arm can be uniquely determined by $\mathbf{k} = (k_1, k_2, \ldots, k_S)$. Let $\mathcal{M}_{\mathbf{k}} = \{i : 1 \leq i \leq M, \phi_{\mathbf{k},i} \neq 0\}$ be the set of nonzero components of feature vector $\phi_{\mathbf{k}}$ and let

$M_{\mathbf{k}} = |\mathcal{M}_{\mathbf{k}}|$.

$$\phi_{\mathbf{k},i} = \begin{cases} \mathbb{I}_{\{k_1=i\}}\,, & \text{if } 1 \le i \le K; \\ \dots & \\ \mathbb{I}_{\{k_2=i-l\cdot K\}}\,, & \text{if } l\cdot K+1 \le i \le (l+1)\cdot K; \\ \dots & \\ -\mathbb{I}_{\{k_1=k_2=i-p\cdot K\}}\,, & \text{if } S\cdot K+1 \le i \le (S+1)\cdot K; \\ \dots & \\ -(-1)^S\mathbb{I}_{\{k_1=k_2=\ldots=k_S=i-K(2^S-2)\}}\,, & \text{if } K(2^S-2)+1 \le i \le K(2^S-1). \end{cases} \tag{2.6}$$

To this end, we can rewrite the expectation of the payoff in MEC as a linear function of the arm feature $\phi_{\mathbf{k}}$,

$$f(\mathbf{k}) = \mathbb{E}[X_{\mathbf{k}}(t)] = \theta^T\,\phi_{\mathbf{k}}, \tag{2.7}$$

where $(\cdot)^T$ denotes transposition.

Given θ, the system can play optimally: An arm with maximal payoff is chosen by $\mathbf{k}^* = \text{argmax}_{\mathbf{k}\in\mathbb{K}}\theta^\top\phi_{\mathbf{k}}$ (here, and in what follows, for the sake of simplicity, we assume that there is a unique optimal arm). A reasonable way to estimate the parameter vector θ is to keep a running average for the components of θ. If at time t the agent chose $\mathbf{k}(t) \in \mathbb{K}$ then the current estimate, $\hat{\theta}(t-1)$, can be updated by

$$\begin{aligned} \hat{\theta}_i(t) &= \hat{\theta}_i(t-1) + \frac{1}{N_i(t)}\left(Y_i(t) - \hat{\theta}_i(t-1)\right)\mathbb{I}_{\{i\in\mathcal{M}_{\mathbf{k}(t)}\}}, \\ N_i(t) &= N_i(t-1) + \mathbb{I}_{\{i\in\mathcal{M}_{\mathbf{k}(t)}\}}. \end{aligned} \tag{2.8}$$

Here $N_i(0) = 0$, $\hat{\theta}_i(0) = 0$. Thus, $N_i(t)$ counts the number of times data for component i has been observed up to time t.

Spanner arm ζ is a set of arms by playing which θ can be learned. In the monitoring problem ζ can be chosen to be $\zeta = \{(k,\dots,k) : 1 \le k \le K\}$, whose cardinality is $K \ll K^S = |\mathbb{K}|$. The set ζ is called a *spanning set* or a *spanner* and its elements are called *spanner arms*.

2.3 Optimal Online Learning with Switching Regret

In practice, it takes a sniffer short period - a tuning time - to tune to the new channel whenever the sniffer changes its channel. Hence, beside the sampling regret R_n^π due to playing suboptimal arms, a policy π may incur the switching regret SW_n^π. Sampling regret is given by

$$R_n^\pi = \mathbb{E}\left[\sum_{t=1}^{n}\left\{\max_{\mathbf{k}\in\mathcal{A}} \phi_{\mathbf{k}}^T\theta - \phi_{\mathbf{k}_t}^T\theta\right\}\right]. \tag{2.9}$$

Let $\mathcal{S}_n(j) = \sum_{t=1}^{n} \mathbb{I}_{\{\mathbf{k}_t=j,\mathbf{k}_{t+1}\neq j\}}$, where $\mathbf{k}_t$ denotes the joint action selected at time t. The switching regret is thus,

$$SW_n^\pi = C_{sw}\sum_{\mathbf{k}\in\mathcal{A}} \mathbb{E}[\mathcal{S}_n(\mathbf{k})], \tag{2.10}$$

where C_{sw} is the switching cost. Now we assume that the switching cost is constant across all joint actions. This is reasonable in a synchronous system where all sniffers coordinate the onsets of monitoring.

An optimal monitoring policy π determines a sequence of actions in $\mathbb{K}$ over time such that the expected total *regret*

$$Q_n^\pi = R_n^\pi + SW_n^\pi$$

is minimized. In Chapter 3 we propose our UCB-based optimal policy where Q_n^π grows sublinearly in n.

2.4 Approximate Online Learning

Using any well-known algorithm in MAB problem in [35] or [9] to select the optimal arm $\mathbf{k}(t) = \text{argmax}_{\mathbf{k}}\, \hat{\mu}_{\mathbf{k}}(t-1)$ is NP-hard due to its equivalence to the MEC problem. This high computation complexity suffers any systems in practical environment. To reduce computation complexity, we apply the idea of GREEDY in [29] with the guarantee of constant ratio of the reward compare to the optimal case. Denote $\mathbf{k}^g$ the assignment (or arm) chosen by GREEDY with com-

plete information. We define the regret of policy π relative to GREEDY as

$$R_g^\pi(n) = \mathbb{E}\left[\sum_{t=1}^{n} \left\{\phi_{\mathbf{k}^g}\theta - \phi_{\mathbf{k}_t}^T\theta\right\}\right]. \tag{2.11}$$

Note that by the property of GREEDY, $\frac{1}{2}\max_{\mathbf{k}\in\mathbb{K}} \phi_{\mathbf{k}}^T\theta - \phi_{\mathbf{k}^g}\theta \leq 0$. Thus, we have

$$R_g^\pi(n) \geq \mathbb{E}\left[\sum_{t=1}^{n} \left\{\frac{1}{2}\max_{\mathbf{k}\in\mathbb{K}} \phi_{\mathbf{k}}^T\theta - \phi_{\mathbf{k}_t}^T\theta\right\}\right].$$

A part in Chapter 3 and Chapter 4 are dedicated to solving this approximate problem.

Chapter 3

Online Learning Policies with Correlation Information

When two or more sniffers are assigned to the same channel, they have a high probability of observing the same set of users. Therefore, when the system calculates the received reward, it should take into account this correlation amount to calculate the assignment's gain correctly. In this chapter, we consider this correlation part as an element of our problem and propose two algorithms, one optimal and one approximate which explore and exploit the correlation reward between sniffers. In section 3.1, we proposed an optimal algorithm for the channel assignment problem. We then evaluate this approach, compare it with the UCB-based algorithm in [35]. We point out the drawbacks of optimal algorithms in channel assignment problems and devise our approximate algorithm to over come these drawbacks in the second section.

3.1 Optimal Online Learning Algorithm with Switching Cost

In reality, whenever a wireless device switches from one frequency to another one, it takes the device a period of time to tune to the new channel. In our experiment with USRP2, the tuning time to change between frequencies in the 802.11 standard is 8.02 milliseconds on average. As a result, we always lose some information for this unavoidable period. In order to include this part in to our problem, we cast it as a switching regret of the system and suppose that it is a constant number for any switching actions. This regret should also be considered as important as the sampling regret we face by choosing wrong assignments. In this section, we build up an algorithm try to minimize both the sampling and switching regret of the channel assignment problem.

3.1.1 An Upper Confidence Bound (UCB)-Based Policy

When the switching cost is not negligible, to limit the increasing of switching regret, an algorithm should not change the joint action too often. Most policies consider switching cost utilize

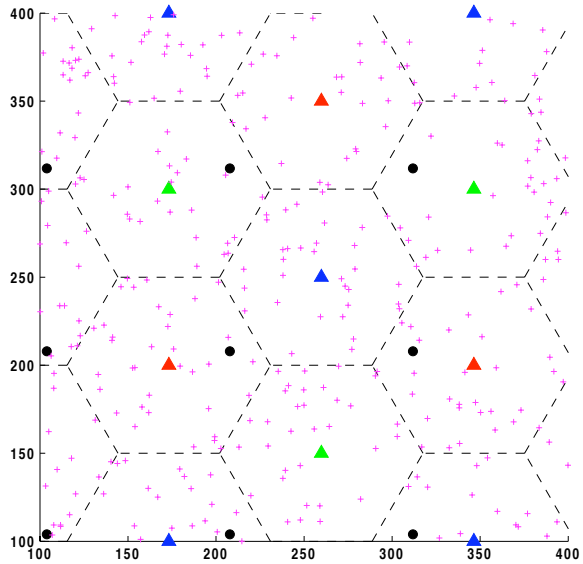

Figure 3.1: Hexagonal layout with users ('+'), sniffers (solid dots), and channels of each cell (in different colors)

"block" sampling, namely, an action once selected is played for a period of time, called an *epoch*. The block length should be short with uncertainty parameters and longer when more knowledge is gained. Define the epoch length of the rth epoch $\tau(r) = \lceil (1+\alpha)^r \rceil$, where $\alpha \geq 0$.

The policy that we consider is built up on UCB2 [9] with the difference in the consideration of dependent arms and that in the initialization stage, we only play each of the spanners ζ once. After that, the decision time instances for arm selection are denoted by t_j, $j = 1, \ldots, J_n$, where $t_1 = |\zeta| + 1$ and J_n is the number of decision time instances up to time n. $t_j, j = 1, \ldots, J_n$ divide the time into epochs of length $l_j = t_{j+1} - t_j$, $j = 1, \ldots, J_n - 1$ to be defined next. At time t_j, the algorithm chooses

$$\mathbf{k}(t_j) = \underset{\mathbf{k} \in \mathcal{E}}{\text{argmax}} V_{\mathbf{k}}(t_j - 1), \tag{3.1}$$

where

$$\begin{aligned} V_{\mathbf{k}}(t_j - 1) &= \hat{\mu}_{\mathbf{k}}(t_j - 1) + \textstyle\sum_{i \in \mathcal{M}_{\mathbf{k}}} \sqrt{\frac{\rho \log t_j}{N_i(t_j - 1)}}, \\ \hat{\mu}_{\mathbf{k}}(t_j - 1) &= \hat{\theta}(t_j - 1)^\top \phi_{\mathbf{k}}. \end{aligned} \tag{3.2}$$

the arm $\mathbf{k}(t_j)$ is played l_j times.

Let $I(t_j) = \underset{m \in \mathcal{M}_{\mathbf{k}(t)}}{\text{argmin}} N_m(t_j - 1)$, each component i is associated with an epoch counter $r_i(t)$

initialized to zero. At time t_j, the epoch counter of component $I(t_j)$ is updated as $r_{I(t_j)}(t_j) = r_{I(t_j)}(t_j - 1) + 1$, and remains the same for the rest of the epoch length $l_j = \tau(r_{I(t_j)}(t_j)) - \tau(r_{I(t_j)}(t_j) - 1)$. After playing $\mathbf{k}(t_j)$ and observing $(Y_i(t); i \in \mathcal{M}_{\mathbf{k}(t_j)}), t_j \leq t < t_j + l_j$ the parameter estimate is updated using (2.8). Then, the process is repeated. The pseudo-code for the algorithm can be seen in 3.1.

Theorem 3.1. *Choose any ρ that satisfies $\rho > 1/1.99$. Then, there exists a constant $C > 0$ (which may depend on ρ) such that for all $n \geq 1$, the expected regret of UCB satisfies*

$$Q_n^{\mathrm{UCB}} \leq 4M\Delta_{\max} \left(\max_{\mathbf{k}:\Delta_{\mathbf{k}}>0} \frac{M_{\mathbf{k}}}{\Delta_{\mathbf{k}}} \right)^2 \rho \log n + C,$$

where $\Delta_{\max} = \max_{\mathbf{k}} \Delta_{\mathbf{k}}$, and C scales linearly with $|\mathbb{K}|$ can be extracted from the proof.

Proof. The proof is similar to the original proof given by Auer *et al* [9], with some elements borrowed from the analysis technique of Audibert *et al* [39]. (see also, [40]) and Gai *et al* [41]. As the total regret consists of the sampling regret and the switching regret, we derive a bound of each term separately.

We start by introducing the necessary notation. We denote by $T_{\mathbf{k}}(n)$ the number of times arm $\mathbf{k}$ is chosen up to time n (including time n): $T_{\mathbf{k}}(n) = \sum_{t=1}^{n} \mathbb{I}_{\{\mathbf{k}(t)=\mathbf{k}\}}$. We let $\mu^* = \max_{\mathbf{k}} \mu_{\mathbf{k}}$, $\Delta_k =$

Algorithm 3.1 The UCB-based algorithm

Initialize:
- Play each arm in the spanner ζ once and update vector $\hat{\theta}$ and its components as (2.8);
- Update sampling and switching regret;

for all $j = 1, 2, ...$ **do**
- Choose arm $\mathbf{k}(t_j)$ that maximize $V_{\mathbf{k}}(t_j - 1)$ using (3.2);

  if $\mathbf{k}(t_j)$ is different from the previous arm **then**
  Adding switching regret;
  end if
  for $i = 1$ **to** l_j **do**
  - Play arm $\mathbf{k}(t_j)$;
  - Update vector $\hat{\theta}$ using (2.8);
  - Update sampling regret;

  end for

end for

$\mu^* - \mu_{\mathbf{k}}$. Then, it is easy see that $\mathbb{E}\left[R_n^{\mathrm{UCB1}}\right] = \sum_{\mathbf{k}} \Delta_{\mathbf{k}} \mathbb{E}\left[T_{\mathbf{k}}(n)\right] \leq (\max_{\mathbf{k}} \Delta_{\mathbf{k}})\ \mathbb{E}\left[\sum_{\mathbf{k}:\Delta_{\mathbf{k}}>0} T_{\mathbf{k}}(n)\right]$. Our goal is to develop a bound on $\mathbb{E}\left[\sum_{\mathbf{k}:\Delta_{\mathbf{k}}>0} T_{\mathbf{k}}(n)\right]$ which scales linearly with M rather that with $|\mathbb{K}|$.

Let $Z_i(t_j) = \mathbb{I}_{\{\mathbf{k}(t_j)\neq\mathbf{k}^*, I(t_j)=i\}}$, and $\tilde{T}_i(t) = \tilde{T}_i(t-1) + Z_i(t_j), t_j \leq t < t_{j+1}$.[1] Note that $\sum_{\mathbf{k}\neq\mathbf{k}^*} T_{\mathbf{k}}(n) = \sum_i \tilde{T}_i(n)$, since exactly one of the counters is incremented on both sides when a suboptimal arm is chosen. Thus, it suffices to bound $\tilde{T}_i(n)$.

Therefore pick any index $1 \leq i \leq M$ and let u be an integer to be chosen later. We have $Z_i(t_j) = Z_i(t_j)\mathbb{I}_{\{\tilde{T}_i(t_j-1)>\tau(u)\}} + Z_i(t_j)\mathbb{I}_{\{\tilde{T}_i(t_j-1)\leq\tau(u)\}}$. Since $\sum_{j=1}^{J_n} Z_i(t_j)\mathbb{I}_{\{\tilde{T}_i(t_j-1)\leq\tau(u)\}} l_j \leq \tau(u)$, it suffices to deal with the first term, which we bound as follows:

$$
\begin{aligned}
&Z_i(t_j)\mathbb{I}_{\{\tilde{T}_i(t_j-1)>\tau(u)\}} \\
&\leq \mathbb{I}_{\left\{V_{\mathbf{k}(t_j)}(t_j-1)>\mu^*, \tilde{T}_i(t_j-1)>\tau(u), I(t_j)=i\right\}} + \mathbb{I}_{\{V_{\mathbf{k}^*}(t_j-1)\leq\mu^*\}}.
\end{aligned}
$$

Now, let the $\hat{\mathbf{k}}_{ij}$ be the arm played in the jth epoch out of the epochs j's where $I(t_{j'}) = i$, and $\hat{t}_{ij}$ is the time when such an epoch starts. Clearly, $\hat{\mathbf{k}}_{ij} = \mathbf{k}(\hat{t}_{ij})$. Denote $\delta(j) = \tau(j) - \tau(j-1)$. Thus,

$$
\begin{aligned}
\tilde{T}_i(n) &= 1 + \textstyle\sum_{j=1}^{J_n-1} Z_i(t_j) l_j \\
&\leq 1 + \tau(u) + \sum_{j=u+1}^{J_{max}} Z_i(\hat{t}_{ij})\mathbb{I}_{\{\tilde{T}_i(\hat{t}_{ij}-1)>\tau(u)\}}\delta(j),
\end{aligned} \tag{3.3}
$$

where J_{max} is the maximum possible number of epochs where $Z_i(t_j) = 1$. Clearly, $J_{max} \leq \lceil \frac{\log n}{\log(1+\alpha)} \rceil$. Therefore,

$$
\begin{aligned}
&\mathbb{E}\left[\tilde{T}_i(n)\right] \leq \tau(u) + 1 \\
&\quad + \sum_{j=u+1}^{J_{max}} \mathbb{P}\Big(V_{\hat{\mathbf{k}}_{ij}}(\hat{t}_{ij}-1) > \mu^*, \tilde{T}_i(\hat{t}_{ij}-1) > \tau(u)\Big)\delta(j) \\
&\quad + \sum_{j=u+1}^{J_{max}} \mathbb{P}\big(V_{\mathbf{k}^*}(\hat{t}_{ij}-1) \leq \mu^*\big)\delta(j).
\end{aligned}
$$

We will now show that both sums can be bounded logarithmically with respect to n, provided that u is sufficiently large.

[1] We are using the assumption that there is a unique optimal arm $\mathbf{k}^*$. Note that this is assumed just for the sake of simplicity and the proof, at the price of a more complicated presentation, works without it.

The summand of the first sum is bounded as follows:

$$p_{1j} \triangleq \mathbb{P}\Big(V_{\hat{\mathbf{k}}_{ij}}(\hat{t}_{ij}-1) > \mu^*, \tilde{T}_i(\hat{t}_{ij}-1) > \tau(u), I(\hat{t}_{ij}) = i\Big)$$

$$\leq \mathbb{P}\Big\{\hat{\mu}_{\hat{\mathbf{k}}_{ij}}(\hat{t}_{ij}-1) > \mu_{\hat{\mathbf{k}}_{ij}} + \Delta_{\hat{\mathbf{k}}_{ij}} - c_{\hat{\mathbf{k}}_{ij},\hat{t}_{ij}-1},$$

$$\tilde{T}_i(t-1) > \tau(u), I(\hat{t}_{ij}) = i\Big\}$$

where $c_{\hat{\mathbf{k}}_{ij},\hat{t}_{ij}-1} = \sqrt{\rho \log \hat{t}_{ij}} \sum_{m\in\mathcal{M}_{\hat{\mathbf{k}}_{ij}}} \sqrt{\frac{1}{N_m(\hat{t}_{ij}-1)}} \triangleq \sqrt{\rho \log \hat{t}_{ij}}\; W_{\hat{\mathbf{k}}_{ij}}(\hat{t}_{ij}-1)$. Now,

$$\Delta_{\hat{\mathbf{k}}_{ij}} - c_{\hat{\mathbf{k}}_{ij},\hat{t}_{ij}-1}$$

$$= \sum_{m\in\mathcal{M}_{\hat{\mathbf{k}}_{ij}}} \left(\frac{\Delta_{\hat{\mathbf{k}}_{ij}}}{W_{\hat{\mathbf{k}}_{ij}}(\hat{t}_{ij}-1)} - \sqrt{\rho \log \hat{t}_{ij}}\right) \sqrt{\frac{1}{N_m(\hat{t}_{ij}-1)}}.$$

We claim that under the condition that $\tilde{T}_i(\hat{t}_{ij}-1) > \tau(u)$ the largest value $W_{\hat{\mathbf{k}}_{ij}}(\hat{t}_{ij}-1)$ can take is bounded from above by $M_{\hat{\mathbf{k}}_{ij}}/\sqrt{\tau(u)}$. To see this note that $\tilde{T}_m(t-1) \leq N_m(t-1)$ holds for any m and t, because $N_m(\cdot)$ is always incremented when $\tilde{T}_m(\cdot)$ is incremented. Further, since $I(t) = \operatorname{argmin}_{m\in\mathcal{M}_{\mathbf{k}(t)}} N_m(t-1)$, $N_{I(t)}(t-1) \leq N_m(t-1)$ holds for any $m \in \mathcal{M}_{\mathbf{k}(t)}$. Thus, for arbitrary $m \in \mathcal{M}_{\hat{\mathbf{k}}_{ij}}$, $\tau(u) < \tilde{T}_i(\hat{t}_{ij}-1) \leq N_i(\hat{t}_{ij}-1) \leq N_m(\hat{t}_{ij}-1)$. The claim then follows from the definition of $W_{\hat{\mathbf{k}}_{ij}}(\hat{t}_{ij}-1)$.

Hence,

$$\Delta_{\hat{\mathbf{k}}_{ij}} - c_{\hat{\mathbf{k}}_{ij},\hat{t}_{ij}-1}$$

$$\geq \sum_{m\in\mathcal{M}_{\hat{\mathbf{k}}_{ij}}} \left(\frac{\Delta_{\hat{\mathbf{k}}_{ij}}\sqrt{\tau(u)}}{M_{\hat{\mathbf{k}}_{ij}}} - \sqrt{\rho \log \hat{t}_{ij}}\right) \sqrt{\frac{1}{N_m(\hat{t}_{-}1)}}.$$

Further, $\frac{\Delta_{\hat{\mathbf{k}}_{ij}}\sqrt{\tau(u)}}{M_{\hat{\mathbf{k}}_{ij}}} - \sqrt{\rho \log \hat{t}_{ij}} \geq \sqrt{\rho \log n}$ holds for $1 \leq \hat{t}_{ij} \leq n$ if

$$\tau(u) \geq \left(2 \max_{\mathbf{k}:\Delta_{\mathbf{k}}>0} \frac{M_{\mathbf{k}}}{\Delta_{\mathbf{k}}}\right)^2 \rho \log n.$$

Then, $\Delta_{\hat{\mathbf{k}}_{ij}} - c_{\hat{\mathbf{k}}_{ij},t-1} \geq \sqrt{\rho \log n}\; W_{\hat{\mathbf{k}}_{ij}}(t-1)$ and thus

$$p_{1j} \leq \mathbb{P}\Big(\hat{\mu}_{\hat{\mathbf{k}}_{ij}}(t_{ij}-1) > \mu_{\hat{k}k_{ij}} + \sqrt{\rho \log n}\; W_{\hat{\mathbf{k}}_{ij}}(\hat{t}_{ij}-1)\Big) \tag{3.4}$$

$$\leq \sum_{\mathbf{k}} M_{\mathbf{k}} \lceil 4 \log n \rceil \exp\left(-1.99\rho \log n\right), \tag{3.5}$$

where the last inequality follows from the union bound and Lemma 3.3, which is presented later in subsection 3.1.2.

The summand of the second sum can be bounded as follows:

$$\begin{aligned}
p_{2j} &\triangleq \mathbb{P}\big(V_{\mathbf{k}^*}(\hat{t}_{ij}-1) \le \mu^*\big) \\
&= \mathbb{P}\Big(\hat{\mu}_{\mathbf{k}^*}(\hat{t}_{ij}-1) + c_{\mathbf{k}^*,\hat{t}_{ij}} \le \mu^*\Big) \\
&\le \sum_{t=\tau(j)}^{n} \mathbb{P}(\hat{\mu}_{\mathbf{k}^*}(t-1) + c_{\mathbf{k}^*,t} \le \mu^*) \\
&= \sum_{t=\tau(j)}^{n} \mathbb{P}\Big(\hat{\mu}_{\mathbf{k}^*}(t-1) \le \mu^* - \sqrt{\rho \log t} W_{k^*}(t)\Big).
\end{aligned}$$

The inequality is due to the fact that $t_{ij} \ge \tau(j)$ and the union bound. Using Lemma 3.3 again, we get that

$$\begin{aligned}
p_{2j} &\le \sum_{t=\tau(j)}^{n} M_{\mathbf{k}^*} \lceil 4 \log n \rceil t^{-1.99\rho} \\
&\le M_{\mathbf{k}^*} \lceil 4 \log n \rceil \int_{\tau(j)}^{\infty} t^{-1.99\rho} \\
&= M_{\mathbf{k}^*} \lceil 4 \log n \rceil \tau(j)^{-1.99\rho+1} \\
&\le M_{\mathbf{k}^*} \lceil 4 \log n \rceil (1+\alpha)^{(-1.99\rho+1)j}.
\end{aligned}$$

Putting together the inequalities, for n sufficiently large, we have

$$
\begin{aligned}
\mathbb{E}\left[\tilde{T}_i(n)\right] &\leq \tau(u) + \sum_{j=u+1}^{J_{max}} (p_{1j} + p_{2j})\delta(j) \\
&\leq \left(2 \max_{\mathbf{k}:\Delta_{\mathbf{k}}>0} \frac{M_{\mathbf{k}}}{\Delta_{\mathbf{k}}}\right)^2 \rho \log n + \sum_{\mathbf{k}} M_{\mathbf{k}} \lceil 4 \log n \rceil \, n^{-1.99\rho} \sum_{j=u+1}^{J_{max}} \delta(j) \\
&+ M_{\mathbf{k}^*} \lceil 4 \log n \rceil \sum_{j=u+1}^{J_{max}} (1+\alpha)^{(-1.99\rho+1)j} \delta(j) \\
&\leq \left(2 \max_{\mathbf{k}:\Delta_{\mathbf{k}}>0} \frac{M_{\mathbf{k}}}{\Delta_{\mathbf{k}}}\right)^2 \rho \log n + \sum_{\mathbf{k}} M_{\mathbf{k}} \lceil 4 \log n \rceil \, n^{-1.99\rho} \lceil (1+\alpha)^{J_{max}} - (1+\alpha)^u \rceil \\
&+ M_{\mathbf{k}^*} \lceil 4 \log n \rceil \sum_{j=u+1}^{J_{max}} (1+\alpha)^{(-1.99\rho+2)j} \\
&\leq \left(2 \max_{\mathbf{k}:\Delta_{\mathbf{k}}>0} \frac{M_{\mathbf{k}}}{\Delta_{\mathbf{k}}}\right)^2 \rho \log n + \sum_{\mathbf{k}} M_{\mathbf{k}} \lceil 4 \log n \rceil \, n^{-1.99\rho+1} + C' M_{\mathbf{k}^*} \lceil 4 \log n \rceil n^{(-1.99\rho+2)},
\end{aligned}
$$

where C' is a proper defined constant dependent on α. Clearly, if $\rho \geq 2/1.99$, the last two terms are $o(1)$.

To this end, we have proved that the sampling regret grows logarithmically with n. Next, we analyze the asymptotic property of the switching regret. Clearly, the number of switching is bounded by the number of times a suboptimal arm is played, which is clearly logarithmic in time. However, a tighter bound can be obtained by taking into account the epoch length. Let $\Psi_i(0) = 0$ and

$$
\Psi_i(t) = \begin{cases} \Psi_i(t-1) + Z_i(t), & t = t_1, t_2, \ldots \\ \Psi_i(t-1), & else \end{cases}
$$

Recall that $Z_i(t_j) = \mathbb{I}_{\{k(t_j) \neq k^*, I(t_j)=i\}}$. Namely, $\Psi_i(t)$ is the number of epochs the ith component incurred till time t, when it is the least visited component in the chosen arm. Clearly, the switching cost if bounded by $C_{sw} \sum_i \Psi_i$. Thus,

$$
\begin{aligned}
\Psi_i(n) &= 1 + \textstyle\sum_{j=1}^{J_n} Z_i(t_j) \\
&\leq 1 + u + \sum_{j=u+1}^{J_{max}} Z_i(\hat{t}_{ij}) \mathbb{I}_{\left\{\tilde{T}_i(\hat{t}_{ij}-1) > \tau(u)\right\}}.
\end{aligned} \tag{3.6}
$$

Therefore,

$$\begin{aligned}
&\mathbb{E}\left[\Psi_i(n)\right] \leq 1 + u \\
&\quad + \sum_{j=u+1}^{J_{max}} \mathbb{P}\Big(V_{\hat{\mathbf{k}}_{ij}}(\hat{t}_{ij}-1) > \mu^*, \tilde{T}_i(\hat{t}_{ij}-1) > \tau(u), I(t) = i\Big) \\
&\quad + \sum_{j=u+1}^{J_{max}} \mathbb{P}\big(V_{\mathbf{k}^*}(\hat{t}_{ij}-1) \leq \mu^*\big).
\end{aligned}$$

Following the same argument as the proof of sampling regret and picking

$$u = \log_{1+\alpha}\left(2 \max_{\mathbf{k}:\Delta_{\mathbf{k}}>0} \frac{M_{\mathbf{k}}}{\Delta_{\mathbf{k}}}\right)^2 \rho \log n,$$

we can prove that

$$\mathbb{E}\left[\Psi_i(n)\right] \leq \log_{1+\alpha} \log n + C''.$$

In summary, the sampling regret grows logarithmic with time, while the switching regret grows in $\log\log$ fashion. Combining the sampling and the switching regret, we complete the proof of the theorem. This algorithm is a part of our paper [42] that is in submission. □

3.1.2 Tail Probability Bounds

This subsection is conducted to support the proof in subsection 3.1.1.

The following lemma generalizes Hoeffding's inequality to sums with a random number of terms. The lemma in the form presented here can be found as Theorem 18 of [43] (a similar statement, generalizing Bernstein's inequality can be extracted from [39]).

Lemma 3.2. *Let $(\mathcal{F}_t; t \geq 0)$ be a filtration. Let $(X_t; t \geq 1)$ be an i.i.d. sequence taking values in some interval of length B. Let $\varepsilon_t \in \{0, 1\}$ be a binary sequence. Assume that X_t is $\mathcal{F}_t$-measurable and ε_t is $\mathcal{F}_{t-1}$-measurable ($t \geq 1$). Let $N_n = \sum_{t=1}^n \varepsilon_t$, $\overline{X}_n = \sum_{t=1}^n \varepsilon_t X_t / N_n$. Then, for any $n \geq 1$, $\eta > 0$,*

$$\mathbb{P}\left(\overline{X}_n > \mathbb{E}\left[X_1\right] + z\sqrt{\frac{1}{N_n}}, N_n \geq 1\right) \leq \frac{\log n}{\log(1+\eta)} \exp\left(-\frac{2z^2}{B^2}\left(1 - \frac{\eta^2}{16}\right)\right).$$

In particular, when $\eta = 0.3$*,*

$$\mathbb{P}\left(\overline{X}_n > \mathbb{E}\left[X_1\right] + \frac{z}{\sqrt{N_n}}, N_n \geq 1\right) \leq \lceil 4 \ln n \rceil \exp\left(-\frac{1.99 z^2}{B^2}\right).$$

Now, we consider a multi-dimensional generalization of this result:

Lemma 3.3. *Let* $(\mathcal{F}_t; t \geq 0)$ *be a filtration. Let* $(X_t; t \geq 1)$ *be an i.i.d. sequence taking values in* $\mathbb{R}^M$ *such that* X_{ti}*, the* i^{th} *component of* X_t*, takes values in some interval of length* B*. Define* $\mu = \sum_{i=1}^M \mathbb{E}\left[X_{1i}\right]$*. Let* $\varepsilon_t \in \{0,1\}^M$ *be an* M*-dimensional binary sequence. Assume that* X_t *is* $\mathcal{F}_t$*-measurable and* ε_t *is* $\mathcal{F}_{t-1}$*-measurable (*$t \geq 1$*). Let* $N_{ni} = \sum_{t=1}^n \varepsilon_{ti}$*,* $\overline{X}_{ni} = N_{ni}^{-1} \sum_{t=1}^n \varepsilon_{ti} X_{ti}$ *and* $\overline{X}_n = \sum_{i=1}^M \overline{X}_{ni}$*. Then, for any* $n \geq 1$*,*

$$\mathbb{P}\left[\overline{X}_n > \mu + z \sum_{i=1}^M \sqrt{\frac{1}{N_{ni}}},\ N_{n1}, \ldots, N_{nM} \geq 1\right] \leq M \lceil 4 \ln n \rceil \exp\left(-\frac{1.99 z^2}{B^2}\right).$$

Proof. Let p denote the probability to be bounded and let $\mu_i = \mathbb{E}\left[X_{1i}\right]$. Then,

$$p \leq \sum_{i=1}^M \mathbb{P}\left[\overline{X}_{ni} > \mu_i + z\sqrt{\frac{1}{N_{ni}}},\ N_{ni} \geq 1\right].$$

The result then follows by applying Lemma 3.2 to each of the M terms on the right-hand side. □

The next result can be extracted from [9] (with a slight improvement). The setting is similar to that of Lemma 3.2 with the deviation from the mean as a deterministic number.

Lemma 3.4. *Let* $(\mathcal{F}_t; t \geq 0)$ *be a filtration. Let* $(X_t; t \geq 1)$ *be an i.i.d. sequence taking values in some interval of length* 1*. Let* $\varepsilon_t \in \{0,1\}$ *be a binary sequence. Assume that* X_t *is* $\mathcal{F}_t$*-measurable and* ε_t *is* $\mathcal{F}_{t-1}$*-measurable (*$t \geq 1$*). Let* $N_n = \sum_{t=1}^n \varepsilon_t$*,* $\overline{X}_n = \sum_{t=1}^n \varepsilon_t X_t / N_n$*. Then, for any* $n \geq 1$*,* $x > 0$*,* $z > 0$*,*

$$\mathbb{P}\left[\overline{X}_n > \mathbb{E}\left[X_1\right] + \frac{z}{2}\right] \leq \mathbb{P}\left[N_n < x\right] + \frac{2}{z^2} \exp\left(-\frac{\lceil x \rceil z^2}{2}\right).$$

Proof. We have

$$\mathbb{P}\left[\overline{X}_n > \mathbb{E}\left[X_1\right] + \frac{z}{2}\right] \leq \mathbb{P}\left[N_n < x\right] + \mathbb{P}\left[N_n \geq x, \overline{X}_n > \mathbb{E}\left[X_1\right] + \frac{z}{2}\right].$$

Now,

$$\mathbb{P}\left[N_n \geq x, \overline{X}_n > \mathbb{E}\left[X_1\right] + \frac{z}{2}\right] = \sum_{s=\lceil x \rceil}^{n} \mathbb{P}\left[N_n = s, \overline{X}_n > \mathbb{E}\left[X_1\right] + \frac{z}{2}\right].$$

Let $S_n = \sum_{t=1}^{n} \varepsilon_t X_t$. Define $\tau(s)$ as the first time when s values of X are observed: $\tau(s) = \min\{t \geq 1 : N_t = s\}$. Further, let $S^{(1)} = S_{\tau(1)}$, $S^{(2)} = S_{\tau(2)}, \ldots$. Note that $S^{(k)}$ has exactly k terms and $S^{(k)}$ is an $\mathcal{F}^{(k)}$-adapted martingale, where $\mathcal{F}^{(k)} = \mathcal{F}_{\tau(k)-1}$ (the so-called the "optional skipping process"). Now, $\overline{X}_n = S_n/N_n = S^{(N_n)}/N_n$. Hence,

$$\begin{aligned}
&\mathbb{P}\left[N_n = s, \overline{X}_n > \mathbb{E}\left[X_1\right] + \frac{z}{2}\right] \\
&= \mathbb{P}\left[N_n = s, S^{(N_n)}/N_n > \mathbb{E}\left[X_1\right] + \frac{z}{2}\right] \\
&= \mathbb{P}\left[N_n = s, S^{(s)}/s > \mathbb{E}\left[X_1\right] + \frac{z}{2}\right] \\
&\leq \mathbb{P}\left[S^{(s)}/s > \mathbb{E}\left[X_1\right] + \frac{z}{2}\right].
\end{aligned}$$

By the Hoeffding-Azuma inequality, $\mathbb{P}\left[S^{(s)}/s > \mathbb{E}X_1 + \frac{z}{2}\right] \leq \exp(-s\, z^2/2)$. Using $\sum_{s=u}^{\infty} e^{-\kappa u} \leq \kappa^{-1}\, e^{-\kappa u}$, which holds for any integer u and $\kappa > 0$, we obtain the desired result. □

3.1.3 Simulation Results and Analysis

In this simulation, wireless users are placed randomly in 2-D plane. The area is partitioned into hexagon cells with circumcircle of radius 86 meters. Each cell is associated with a base station operating in a channel (and so are the users in the cell). The channel to base station assignment ensures that *no neighboring cells use the same channel*. Sniffers are deployed in a grid formation separated by distance 100 meters, with a coverage radius of 12 meters. A snap shot of the synthetic deployment is show in Figure 3.1. The transmission probability of users is selected uniformly from

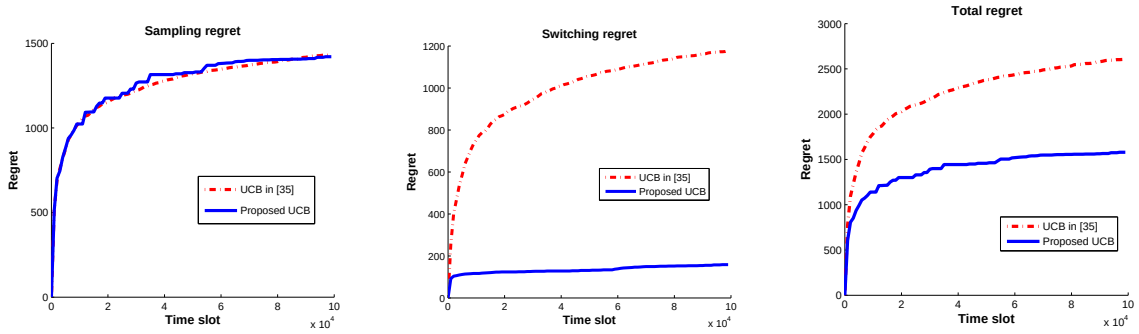

Figure 3.2: Regret of two algorithms over the configuration of 4 APs using 3 channels and 3 sniffers

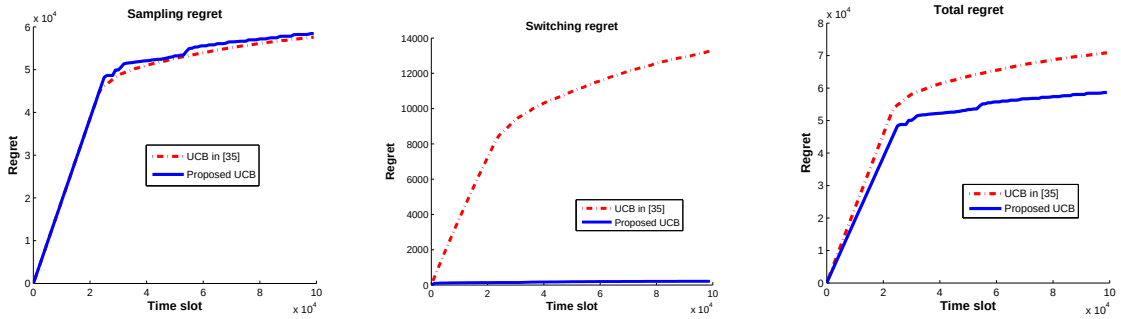

Figure 3.3: Regret of two algorithms over the configuration of 12 APs using 3 channels and 6 sniffers

$[0, 0.006]$, resulting in an average busy probability of 0.2685 in each cell. The switching regret here is set at 0.3. We vary the number of cells from 4 to 12, and the number of sniffers from 3 to 6.

Figure 3.2 and 3.3 show the regret of our proposed UCB and the UCB-based algorithm in [35] over time. In all scenarios, two algorithms have approximately the same sampling regret. It is consistent because they utilize the same equations to find the best arm to play whenever the system needs to make a decision. However, by using epoch, our policy surpasses the UCB-based algorithm in [35] in both switching regret (which directly leads to better total regret), and computation time (as showed in Table 3.1).

Table 3.1: Computation time (mins)

Configuration	Proposed UCB	UCB in [35]
4 APs, 3 channels, 3 sniffers	0.64	14.68
12APs, 3 channels, 6 sniffers	21.9	1868.5

3.2 Approximate Online Policy with Correlation Information

In the offline setting, as proved in [28], MEC problem is NP-hard, where graph G and user-activity probabilities are both known. Therefore, in order to find the optimal solution, our algorithm suffers a high computation complexity as we need to include all assignments in our computation. To reduce the complexity, we devise this approximation algorithm built up from offline Greedy algorithm and the reward learning tool as we introduced previously in Chapter 2. The Greedy algorithm is shown to be $\frac{1}{2}$-approximate in the work of [28]. In the online setting, when both G and user-activity probabilities are not known a prior, our ε-GREEDY-APPROX policy is well-fit in this configuration.

First we extend the definition of arm features to the case where only a subset of sniffers is in use. In particular, consider an action $\mathbf{k} = (k_1, k_2, ..., k_S)$, where $k_i = 0$ for some $i \in [1, S]$. In other words, some sniffers are not assigned any channel. Clearly, [35] is still valid as the arm feature for $\mathbf{k}$, and the expected payoff of arm $\mathbf{k}$ is $f(\mathbf{k}) = \theta_{\mathbf{k}}^T \phi$. Next, we rewrite GREEDY in terms of θ. GREEDY proceeds in L rounds. Let the arm chosen by round l be $\mathbf{k}_l = (\langle i_1, k_{i_1} \rangle, \langle i_2, k_{i_2} \rangle, ..., \langle i_l, k_{i_l} \rangle)$. Let $\oplus$ denote concatenation. In the $l + 1^{st}$ round, GREEDY picks sniffer i_{l+1} and assign it channel $k_{i_{l+1}}$ if and only if the following conditions holds,

$$i_{l+1} = \arg \max_{j \in \mathcal{S}/\{i_1, i_2, ..., i_l\}} \max_{c=1,2,...,K} (\phi_{\mathbf{k}_l \oplus \langle j,c \rangle}^T \theta - \phi_{\mathbf{k}_l}^T \theta), \tag{3.7}$$

and

$$k_{i_{l+1}} = \arg \max_{c=1,2,...,K} (\phi_{\mathbf{k}_l \oplus \langle i_{l+1},c \rangle}^T \theta - \phi_{\mathbf{k}_l}^T \theta). \tag{3.8}$$

In the $l + 1^{st}$ round, the total number of choices are $K(S - l)$. GREEDY needs to perform $K^2(S-l)(2^S-1)$ multiplications to compute the expected payoffs and make $K(S-l)$ comparisons

to find the optimal. Thus, the total computation complexity is $O(l(2S - l + 1)K^2(2^S - 1)/2)$ compared to $O(K^S)$ for the optimal assignment through enumeration. The actual computation time can reduce by only considering the non-zero entries in the arm feature. After having all required definitions, we are in the position to present the ε-GREEDY-APPROX algorithm that use GREEDY as a subroutine. The algorithm is summarized in Algorithm 3.2.

To establish the regret bound of the ε-GREEDY-APPROX algorithm, we first introduce the following lemma.

Lemma 3.5. *Given θ, there exists a non-empty $B \in [0,1]^{K\cdot(2^S-1)}$ centered at θ such that $\forall \hat{\theta} \in B$, $\mathbf{k}^g(\hat{\theta}) = \mathbf{k}^g(\theta)$.*

Lemma 3.5 implies that as long as $\hat{\theta}$, the estimate of θ, is sufficiently close to θ, the channel assignment of GREEDY is identical.

Proof. We prove by constructing a convex region C such that $\forall \hat{\theta} \in C$, the choice of GREEDY $\mathbf{k}^g(\theta) = \mathbf{k}^g(\hat{\theta})$.

Let the arm chosen in GREEDY given θ be $\mathbf{k}^g(\theta) = (k_{i_1}, k_{i_2}, \ldots, k_{i_S})$. Assume there is no tie in the execution of GREEDY. Consider applying GREEDY to $\hat{\theta}$. The sufficient condition that the same arm is chosen is given by,

$$\phi^T_{\mathbf{k}_l \oplus \left\langle i_{l+1}, k_{i_{l+1}} \right\rangle} \hat{\theta} > \phi^T_{\mathbf{k}_l \oplus \langle j,c \rangle} \hat{\theta},$$

$\forall j \neq i_1, i_2, \ldots, i_{l+1}, c \neq k_{i_{l+1}}, \forall l$. Or equivalently, $\forall j \neq i_1, \ldots, i_{l+1}, c \neq k_{i_{l+1}}, \forall l$,

$$(\phi_{\mathbf{k}_l \oplus \left\langle i_{l+1}, k_{i_{l+1}} \right\rangle} - \phi_{\mathbf{k}_l \oplus \langle j,c \rangle})^T \hat{\theta} > 0. \tag{3.9}$$

The above inequalities define a set of half planes with non-empty convex intersection since $\hat{\theta} = \theta$ satisfies all the inequalities. Therefore, there exists a ball B centered at θ, and thus, the conclusion in Lemma 3.5 holds. □

The regret bound of the ε-GREEDY-APPROX algorithm is summarized in Theorem 3.6.

Algorithm 3.2 The ε-GREEDY-APPROX algorithm

Define: the sequence $\varepsilon_t \in (0,1], t = 1, 2, \ldots$ by

$$\varepsilon_t \stackrel{def}{=} \min\left\{1, \frac{c}{t}\right\}. \tag{3.10}$$

for $t = 1$ **to** $Stoppingtime$ **do**

- Let i_t the arm picked by GREEDY;
- With probability $1 - \varepsilon_t$ play i_t and with probability ε_t play a random spanner arm;
- Observe the feedback and update the estimation of parameters using (2.8).

end for

Theorem 3.6. *Let*

$$\varepsilon_n = \min\left\{1, \frac{c}{n}\right\}, \; n > |\zeta|, \tag{3.11}$$

where $c > 0$ is a tuning parameter. Then, assuming that $c > \min(10|\zeta|, \frac{4|\zeta|}{d^2})$, where $d = \min_{\mathbf{k}:\Delta_{\mathbf{k}}>0} \Delta_{\mathbf{k}}$, the expected regret of ε-GREEDY satisfies

$$R_g^{\varepsilon\text{-greedy-approx}}(n) \leq c\log(n+1) + O(1). \tag{3.12}$$

Proof. The proof follows the steps of the proof in [35] with some modifications. We denote by $T_{\mathbf{k}}(n)$ the number of times arm $\mathbf{k}$ is chosen up to time n (including time n): $T_{\mathbf{k}}(n) = \sum_{t=1}^{n} \mathbb{I}_{\{\mathbf{k}(t)=\mathbf{k}\}}$.

Without loss of generality, we assume that $\varepsilon_n = 0$ if $n \leq |\zeta|$ (note that the algorithm does not depend on the values of $\varepsilon_1, \ldots, \varepsilon_{|\zeta|}$ and this assumption allows us to shorten the proof). Clearly, it suffices to bound $\mathbb{E}\left[T_{\mathbf{k}}(n)\right]$. For this purpose we will bound $\mathbb{P}\left[\mathbf{k}(n) \neq \mathbf{k}^g\right]$, where $\mathbf{k}^g$ is an action chosen by the GREEDY.

For $n > |\zeta|$, by Lemma 3.5, the probability of choosing $\mathbf{k}$ is bounded by

$$\mathbb{P}\left[\mathbf{k}(n) \neq \mathbf{k}^g\right] \leq \frac{\epsilon_n \, \mathbb{I}_{\{\mathbf{k}\in\zeta\}}}{|\zeta|} + (1-\epsilon_n)\,\mathbb{P}\left[\hat{\theta}(n-1) \notin B\right].$$

Let δ be the radius of B. We have

$$\mathbb{P}\left[\hat{\theta}(n-1) \notin B\right] \leq \sum_i \mathbb{P}\left[|\hat{\theta}_i(n-1) - \theta_i| > \frac{\delta}{\sqrt{M}}\right].$$

Define $x_0 = \frac{1}{2|\zeta|}\sum_{t=1}^{n-1}\epsilon_t$. By Lemma 7 in [35],

$$\mathbb{P}\left[|\hat{\theta}_i(n-1)-\theta_i| > \frac{\delta}{\sqrt{M}}\right]$$
$$\leq \mathbb{P}\left[N_i(n-1)\leq x_0\right] + \frac{M}{\delta^2}\exp\left(-\frac{2\lceil x_0\rceil\delta^2}{M}\right).$$

Let us now bound the first term of the right-hand side. Let $\mathbf{k}_e \in \zeta$ be such that $i \in \mathcal{M}_{\mathbf{k}_e}$. Let $N_i^R(n)$ be the number of times $\mathbf{k}_e$ has been selected up to time n in an exploration step: $N_i^R(n) = \sum_{t=1}^{n}\mathbb{I}_{\{\mathbf{k}(t)=\mathbf{k}_e, U_t\leq\varepsilon_t\}}$. Clearly, $N_i^R(n-1)\leq N_i(n-1)$. Hence, $\mathbb{P}\left[N_i(n-1)\leq x_0\right]\leq \mathbb{P}\left[N_i^R(n-1)\leq x_0\right]$. Furthermore, $\mathbb{E}\left[N_i^R(n-1)\right] = \frac{1}{|\zeta|}\sum_{t=1}^{n-1}\epsilon_t = 2x_0$, and $\mathrm{Var}[N_i^R(n-1)]\leq \frac{1}{|\zeta|}\sum_{t=1}^{n-1}\epsilon_t = 2x_0$. Therefore, by the Bernstein's inequality (for details see [9]), we have

$$\mathbb{P}\left[N_i^R(n-1)\leq x_0\right]\leq e^{-x_0/5}. \tag{3.13}$$

Since $x_0 = \frac{1}{2|\zeta|}\sum_{t=1}^{n-1}\epsilon_t \geq \frac{c}{2|\zeta|}\log n$, we have

$$\mathbb{P}\left[N_i^R(n-1)\leq x_0\right]\leq e^{-x_0/5}\leq n^{-\frac{c}{10|\zeta|}}.$$

Thus,

$$\mathbb{P}\left[\hat{\theta}(n-1)\notin B\right]\leq Mn^{-\frac{c}{10|\zeta|}} + \frac{M^2}{\delta^2}\,n^{\frac{c\delta^2}{M|\zeta|}}. \tag{3.14}$$

Therefore,

$$\mathbb{P}\left[\mathbf{k}(n)\neq\mathbf{k}\right]\leq \frac{\epsilon_n\,\mathbb{I}_{\{\mathbf{k}\in\zeta\}}}{|\zeta|} + Mn^{-\frac{c}{10|\zeta|}} + \frac{M^2}{\delta^2}\,n^{\frac{c\delta^2}{M^2|\zeta|}}.$$

Now, $\mathbb{E}\left[R_n^{\varepsilon-\mathrm{greedy}-\mathrm{approx}}\right]\leq |\zeta| + \sum_{t=|\zeta|+1}^{n}\mathbb{P}\left[\mathbf{k}(n)\neq\mathbf{k}\right]\leq |\zeta| + c\log n + M\sum_{t=1}^{n}t^{-\frac{c}{10|\zeta|}} + \frac{M^2}{\delta^2}\sum_{t=1}^{n}t^{\frac{c\delta_{\mathbf{k}}^2}{|\zeta|}}$. If $c > \min(10|\zeta|, \frac{|\zeta|}{\delta_{\mathbf{k}}^2})$ holds for any suboptimal $\mathbf{k}$ then the sum of the last two terms over $t = 1,\ldots,n$ becomes finite. This finishes the proof of the result. □

The empirical results of this algorithm is provided later with the other three approximate online algorithms. Our algorithm has been published in [44].

3.3 Conclusions

This chapter first proposed an UCB-based optimal approach for the online MEC problem. Our optimal algorithm explores the characteristic of the environment and exploits the channel by balancing between the confidence interval of each arm and the average reward of the arm. By observing the channel over each epoch, our algorithm outperforms the UCB-based algorithm in [35] in the switching regret, hence the total regret; and approaches the optimal upper bound performance. The future work follows this direction includes extension to non-stationarity environments, which could be done, e.g., along the line of work of [43], the consideration of an adversarial setting [45], [46] and/or switching costs [19].

Although our optimal approach can obtain a logarithmic regret, it suffers a high computation complexity due to MEC is a NP-hard problem. To make it implementable in reality, we devise ε-GREEDY approach to reduce the computation time. ε-GREEDY has an excellent learning rate due to extracting information from the correlated elements of arm featured vector. This approximate algorithm out-performs our optimal approach in running time, but still has a comparative computation complexity. In Chapter 4, we propose three approximate online learning algorithms without using the correlation information between sniffers to reduce the complexity. We will evaluate the regret, computation time, and rate of learning of all approximate algorithms.

Chapter 4

Approximate Online Learning Policies without Correlation Information

As we have seen from Chapter 3, the correlation reward can support previous algorithms finding the best assignment. However, using this information will add more parameters into the calculation process of the system. This factor mainly increases the complexity of all algorithms. To reduce this amount of computation time, we build up three approximate algorithms without concerning about the correlation part. We show that our algorithms are competitive in many systems which may requires low computation complexity, high rate of learning or minimum requirement of hardware.

4.1 Approximate Online Policy with Adversarial Assumption

In ε-GREEDY-APPROX, the algorithm needs to decide the sequence of sniffers to choose and their assignments based on the approximate vector $\hat{\theta}$. It makes the algorithm suffer a high computation in order to find the best arm to play each time. ε-GREEDY-APPROX can be classified as a *single agent* multi-arm learning as both learning and decision are made by one agent. To further improve the computation efficiency, we propose a *multi-agent* multi-arm learning strategy called EXP3-APPROX. The basic idea is to have S agents, each corresponding to one stage of GREEDY. Each agent maintains its own set of states and makes decision accordingly by treating decisions from agents prior to itself as a black box. Learning of the agents is coupled by how the payoff is partitioned among the agents.

Consider L agents $\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_L$. Agent $\mathcal{E}_i$ keeps a weight matrix W^i of dimension $S \times K$. At time t, agent $\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_L$ choose their respective action in action space $\mathcal{S} \times \mathcal{K}$ according to a modified version of the EXP3.1 algorithm [13] (Algorithm 4.1). Agent $\mathcal{E}_l$ knows about the decision of agent $\mathcal{E}_1$ through $\mathcal{E}_{l-1}$, namely, $\mathbf{k}_l = (\langle i_1, k_{i_1} \rangle, \langle i_2, k_{i_2} \rangle, \ldots, \langle i_{l-1}, k_{i_{l-1}} \rangle)$. Algorithm

4.1 summarizes the inner EXP3.1 policy executed by agent $\mathcal{E}_l$ at time t.

It has been proven in [13], EXP3.1 can achieve a uniform regret bound of $O(\sqrt{nK' \ln K'})$ at time n with respect to an optimal policy using a single action, where K' is the number of arms. Compared to EXP3.1 in [13], the main differences in the modified EXP3.1 is the consideration of only a subset of actions based on the previous agents' decisions. This requires normalization of $w^l_{s,k}$ accordingly. Similarly, we can adapt the EXP3.P algorithm in [13], which has less variances in the regrets. The EXP3-APPROX (Algorithm 4.2) uses the modified EXP3.1 as a subroutine for each agent. It is easy to see that EXP3-APPROX has a computation complexity of $O(LKS)$ per update.

To see the regret bound of EXP-APPROX, we need to first establish a few lemmas. Let $\tilde{g}$ be GREEDY when decisions are made with some additive errors. More specifically, $\tilde{g}$ chooses $a_{j+1} = \left\langle i_{j+1}, k_{i_{j+1}} \right\rangle$ in stage $j+1$ if and only if

$$f(\mathbf{k}^{\tilde{g}}_j \oplus a_{j+1}) - f(\mathbf{k}^{\tilde{g}}_j) \geq \max_{\langle s,c \rangle} \{ f(\mathbf{k}^{\tilde{g}}_j \oplus \langle s, c \rangle) - f(\mathbf{k}^{\tilde{g}}_j) \} - \varepsilon_{j+1}, \tag{4.1}$$

where ε_{j+1} is the additive error in stage $j+1$.

Algorithm 4.1 The MODIFIED EXP3.1 algorithm

Input: $\left\langle i_j, k_{i_j} \right\rangle, j = 1, 2, \ldots, l-1$;
Initialize:

- Let $t = 1$, and $\hat{G}_{i,j}(1) = 0$ for $i = 1, 2, \ldots, S, j = 1, 2, \ldots, K$;
- $C = \mathcal{S} \backslash \{i_1, i_2, \ldots, i_{l-1}\}$;
- $w^l_{s,k} = 1, s \in \mathcal{S}, k \in \mathcal{K}; r_l = 0$;
- $g = SK \ln SK/(e-1)4^{r_l}$;
- $\gamma = \min\left\{1, \frac{1}{2^{r_l}}\right\}$;
- $p_{s,k} = (1-\gamma)\frac{w^l_{s,k}}{\sum_{(i,j)\in C} w^l_{i,j}} + \frac{\gamma}{SK}, \forall s \in \mathcal{S}, k \in \mathcal{K}$;

for all $t = 1, 2, \ldots$ **do**
  repeat
    ♯♯Draw (i_l, k_{i_l}) randomly according to the probabilities $p_{s,k}$;
  until $i_l \notin C$
  - Receive reward $x_{i_l,k_{i_l}}(t)$;
  - $\hat{x}_{i_1,k_{i_l}} = \frac{x_{i_l,k_{i_l}}(t)}{p_{i_l,k_{i_l}}} \sum_{s \notin C, k \in \mathcal{K}} p_{s,k}$;
  - $w^l_{i_1,k_{i_l}} = w^l_{i_1,k_{i_l}} exp\left(\frac{\gamma \hat{x}_{i_l,k_{i_l}}}{SK}\right)$;
  - $\hat{G}_{i_l,k_{i_l}}(t+1) = \hat{G}_{i_l,k_{i_l}}(t) + x_{i_l,k_{i_l}}(t)$;

  if $\max_{i,j} \hat{G}_{i,j}(t) > g - SK/\gamma$ **then**
    $r_l \leftarrow r_l + 1$;
  end if
end for

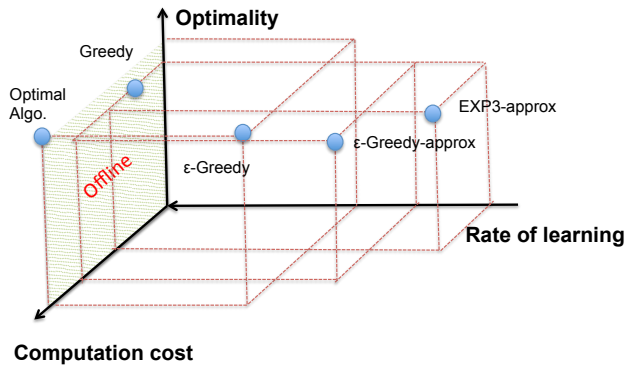

Figure 4.1: Trade-offs in computation complexity, optimality and rate of learning in offline and online algorithm

Lemma 4.1. *$\tilde{g}$ satisfies*

$$f(\mathbf{k}^{\tilde{g}}) \geq \frac{1}{2}\left(\max_{\mathbf{k}\in\mathbb{K}} f(\mathbf{k}) - \sum_{j=1}^{p} \varepsilon_j\right).$$

Lemma 4.1 states that with additive errors, GREEDY can achieve a utility no less than half of that of the optimal algorithm minus the sum of the additive errors.

Proof. Let S be a finite set. A function $h : 2^S \to R$ is submodular if for any $A \subset B \subset S$ and $x \in S/B$, $h(A \bigcup \{x\}) - h(A) \geq h(B \bigcup \{x\}) - h(B)$. First, we show that $f(\mathbf{k}) = \phi_{\mathbf{k}}^T \theta$ is non-decreasing and submodular under the generalized definition of arm features. To see so, consider $\mathbf{k}_L = (\langle i_1, k_{i_1}\rangle, \langle i_2, k_{i_2}\rangle, \ldots, \langle i_L, k_{i_L}\rangle)$ and $\mathbf{k}_M = (\langle i_1, k_{i_1}\rangle, \langle i_2, k_{i_2}\rangle, \ldots, \langle i_M, k_{i_M}\rangle)$, where $L < M$. Additionally, let $a = \langle q, k_q\rangle$, where $q \neq i_1, i_2, \ldots, i_M$. By definition, $f(\mathbf{k}_L) = \mathbb{E}\left[\left|\bigcup_{j\in(i_1,i_2,\ldots,i_L)} U_{j,k_j}(t)\right|\right]$. Clearly, $f(\mathbf{k}_L \oplus a) \geq f(\mathbf{k}_L)$ as $\bigcup_{j\in(i_1,i_2,\ldots,i_L)} U_{j,k_j}(t) \cup U_{q,k_q} \supseteq \bigcup_{j\in(i_1,i_2,\ldots,i_L)} U_{j,k_j}(t)$. Additionally, since $\bigcup_{j\in(i_1,i_2,\ldots,i_L)} U_{j,k_j}(t) \cup U_{q,k_q} / \bigcup_{j\in(i_1,i_2,\ldots,i_L)} U_{j,k_j}(t) \supseteq \bigcup_{j\in(i_1,i_2,\ldots,i_M)} U_{j,k_j}(t) \cup U_{q,k_q} / \bigcup_{j\in(i_1,i_2,\ldots,i_M)} U_{j,k_j}(t)$, $f(\mathbf{k}_L \oplus a) \geq f(\mathbf{k}_M \oplus a)$.

By renumbering the sniffers, we can assume that GREEDY picks the arm $(\left\langle 1, k_1^{\tilde{g}}\right\rangle, \left\langle 2, k_2^{\tilde{g}}\right\rangle, \ldots, \left\langle i, k_p^{\tilde{g}}\right\rangle)$ with errors ε_j's. Denote the optimal solution $(\langle 1, k_1^*\rangle, \langle 2, k_2^*\rangle, \ldots, \langle p, k_p^*\rangle)$. Define the action chosen up to stage j as $\mathbf{k}_j^{\tilde{g}} = (\left\langle 1, k_1^{\tilde{g}}\right\rangle, \left\langle 2, k_2^{\tilde{g}}\right\rangle, \ldots, \left\langle i, k_j^{\tilde{g}}\right\rangle)$. Similarly, we define

$\mathbf{k}^*_j = (\langle 1, k^*_1\rangle, \langle 2, k^*_2\rangle, \ldots, \left\langle j, k^*_j\right\rangle)$. We have

$$\begin{aligned} f(\mathbf{k}^{\tilde{g}}) &= \textstyle\sum_j \left(f(\mathbf{k}^{\tilde{g}}_j) - f(\mathbf{k}^{\tilde{g}}_{j-1})\right) \\ &\geq \textstyle\sum_j \left(f(\mathbf{k}^{\tilde{g}}_{j-1} \oplus \left\langle j, k^*_j\right\rangle) - f(\mathbf{k}^{\tilde{g}}_{j-1})\right) + \textstyle\sum_j \varepsilon_j \\ &\geq \textstyle\sum_j \left(f(\mathbf{k}^{\tilde{g}} \oplus \left\langle j, k^*_j\right\rangle) - f(\mathbf{k}^{\tilde{g}})\right) + \textstyle\sum_j \varepsilon_j \\ &\geq f(\mathbf{k}^{\tilde{g}} \oplus_j \left\langle j, k^*_j\right\rangle) - f(\mathbf{k}^{\tilde{g}}) + \textstyle\sum_j \varepsilon_j \\ &= f(\mathbf{k}^{\tilde{g}} \oplus \mathbf{k}^*) - f(\mathbf{k}^{\tilde{g}}) + \textstyle\sum_j \varepsilon_j \\ &\geq f(\mathbf{k}^*) - f(\mathbf{k}^{\tilde{g}}) + \textstyle\sum_j \varepsilon_j. \end{aligned} \tag{4.2}$$

The first inequality is due to the decision of GREEDY. The second and third inequalities are due to the submodularity of f. Note that we abuse the use of f here by generalizing to the case where a sniffer may be assigned to multiple channels.

From (4.2), we have

$$f(\mathbf{k}^{\tilde{g}}) \geq \frac{1}{2}\left(f(\mathbf{k}^*) + \sum_j \varepsilon_j\right). \tag{4.3}$$

□

Let r_i be the expected regret experienced by agent $\mathcal{E}_i$ and let $R = \sum_{i=1}^{L} r_i$. The following lemma relates the regret experienced by each agent to the regret of the original online problem.

Lemma 4.2. $R_g^{EXP3-approx}(n) > R/2$.

Algorithm 4.2 The EXP3-APPROX algorithm

for $n = 1$ **to** T **do**
  for $l = 1$ **to** L **do**
    - $\gamma = \min\{1, \sqrt{\frac{(S-(l-1))K\ln(S-(l-1))K}{(e-1)T}}\}$;
    - Run modified EXP3.1 to select an action $\langle i_l, k_{i_l}\rangle$;

  end for Play $\mathbf{k} = (\langle i_1, k_{i_1}\rangle, \langle i_2, k_{i_2}\rangle, \ldots, \langle i_L, k_{i_L}\rangle)$ and observe $Y^o_n = (U_{i_1,k_{i_1}}, U_{i_2,k_{i_2}}, \ldots, U_{i_L,k_{i_L}})$;
  for $l = 1$ **to** L **do**
    - Feedback $x_{i_l,k_{i_l}} = \left|\bigcup_{j=1}^{l} U_{i_j,k_{i_j}} - \bigcup_{j=1}^{l-1} U_{i_j,k_{i_j}}\right|$ to agent l;
    - Agent $\mathcal{E}_l$ updates W^l;

  end for
end for

Proof. Similar to the proof of [38], we view EXP3-APPROX as producing an approximate version of the offline greedy schedule. First, we view the sequence of actions selected by $\mathcal{E}_i$ as a single "meta-function" $\tilde{a}_i$ and define $f_t(\mathbf{k}\oplus\tilde{a}_i) = f(\mathbf{k}\oplus\tilde{a}^t_i)$. Define $h = \frac{1}{n}\sum_{t=1}^{n} f_t$, and let $\tilde{S}_i = \langle \tilde{a}_1, \tilde{a}_2, \ldots, \tilde{a}_i\rangle$.

By construction,

$$\begin{aligned} \frac{r_i(n)}{n} &= \max_{a \in A_i} \left\{ f\left(\tilde{S}_{i-1} \oplus a\right) - f\left(\tilde{S}_{i-1}\right) \right\} \\ &- \left(f\left(\tilde{S}_{i-1} \oplus \tilde{a}_i\right) - f\left(\tilde{S}_{i-1}\right) \right), \end{aligned} \tag{4.4}$$

where A_i is the set of valid actions agent $\mathcal{E}_i$ can take.

Thus, EXP3-APPROX behaves like GREEDY for the function h, where the ith decision is made with additive error $\frac{r_i}{n}$. From Lemma 4.1, we have $R_g^{EXP3-approx}(n) \leq \frac{1}{2}\sum_{i=1}^{p} r_i(n) = R(n)/2$. □

Note that in EXP3-APPROX, the feedback $x_{i_l,k_{i_l}}$ to agent $\mathcal{E}_l$ at time t satisfies $\mathbb{E}\left[x_{i_l,k_{k_l}}\right] = f(\mathbf{k}_{l-1} \oplus \langle i_l, k_{k_l}\rangle) - f(\mathbf{k}_{l-1})$ that depends on the choices of agents $\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_{l-1}$ and is independent of the choices of agent $\mathcal{E}_t$ in the previous round. Therefore, each agent faces an adversarial multi-arm bandit [13]. Using the same arguments as the proof of Theorem 12 in [38], we have:

Theorem 4.3. *For finite time horizon T, algorithm* EXP3-APPROX *has an expected regret,*

$$\mathbb{E}\left[R_g^{EXP3-approx}(T)\right] = O(\sqrt{TSK \ln SK}).$$

We are going to publish this algoritm in [47].

4.2 Distributed Approximate Online Policy with Adversarial Assumption

Until now, we have proposed two centralized approximation algorithms for online learning, both achieving sub-linear regrets compared to the offline GREEDY. Consider a synchronized network of S sniffers. In the centralized algorithm, in each slot, a control center needs to assign the new channel assignment to each sniffer and collect the observations (users observed) from each sniffer. The total number of messages generated is $\Theta(S)$. In addition, the central controller needs to store a total of $\Theta(LSK)$ amount of information. Distributed solutions have the advantage of distributed states and robustness to a single point of failure. In what follows, we present DEXP3-

Algorithm 4.3 The Distributed EXP3-APPROX algorithm

Initialize: $\alpha > 0$, $w^l_{s,k} \leftarrow 1$, and $Z_l \leftarrow SK$, $l = 1, 2, \ldots, L$;
for all $t = 1, 2, 3, \ldots$ **do**
  Initialize $\mathcal{S}^{'}_{s,k} \leftarrow \emptyset$, $R_l = 0, l = 1, 2, \ldots, L$;
  for agent $l = 1$ **to** L **do**
    Initialize $\mathcal{S}^{'} \leftarrow \emptyset$;
    for $s \notin \mathcal{S}^{'}$ in parallel **do**
      repeat
        Sample $X_s \sim \text{Poisson}\Big(\alpha \cdot \mathbb{P}(\sum_k w^l_{s,k}, Z_l, R_l, \gamma_l, |\mathcal{S}^{'}_{s,k}|)\Big)$;
        if $^{\dagger\dagger} X_s \geq 1$ **then**
          Broadcast $<$ sampled X_s, ID(s) $>$; Add $\mathcal{S}^{'} \leftarrow \mathcal{S}^{'} \cup s$; Receive messages from $\mathcal{S}^{'}$;
          if ID(s) = $\min_{s' \in \mathcal{S}'}$ ID($s^{'}$) **then**
            • Select exactly one element s_l from $\mathcal{S}^{'}$ such that each $s^{'}$ is selected with probability $X_{s'} / \sum_{s \in \mathcal{S}'} X_s$;
            • Broadcast $<$ select ID(s_l) $>$;
          end if
          if ID(s) = ID(s_l) **then**
            • Select exactly one channel k_l with probability $w^l_{s_l,k_l} / \sum_k w^l_{s_l,k}$;
            • Compute the reward $x_{\langle s_l, k_l \rangle}$ from (4.5);
            • Update $w^l_{s,k}$, $\hat{G}^l_{s,k}$;
            • $\hat{R}_i = \sum_k w^i_{s,k}, i = l+1, 2, \ldots, L$;
            • $\Delta_l = w^l_{s,k}(t) - w^l_{s,k}(t-1)$ and $Z_l \leftarrow Z_l + \Delta_l$;
            • Broadcast $< f_t(\mathcal{S}^{'}_{s,k} \oplus \langle s_l, k_l \rangle)$, ID$\langle s_l, k_l \rangle >$, $< \Delta_l$, ID$\langle s_l, k_l \rangle >$, and $< \hat{R}_{l+1}, \hat{R}_{l+2}, \ldots, \hat{R}_L$, ID$\langle s_l, k_l \rangle >$;
            if $\hat{G}^l_{s,k} > g_l - SK/\gamma_l$ **then**
              • $r_l \leftarrow r_l + 1$, recalculate g_l and γ_l;
              • Broadcast $< r_l, g_l, \gamma_l >$;
            end if
          end if
        end if
        if receive $< \Delta_l$, ID$\langle s_l, k_l \rangle >$ and $< \hat{R}_{l+1}, \hat{R}_{l+2}, \ldots, \hat{R}_L$, ID$\langle s_l, k_l \rangle >$ **then**
          • $\mathcal{S}^{'}_{s,k} \leftarrow \mathcal{S}^{'}_{s,k} \oplus \langle s_l, k_l \rangle$, $Z_l \leftarrow Z_l + \Delta_l$;
          • $R_i = R_i + \hat{R}_i, i = l+1, l+2, \ldots, L$;
        end if
        if receive $< r_l, g_l, \gamma_l >$ **then**
          Update r_l, g_l, and γ_l;
        end if
      until s receives a message of type $<$ select ID $>$
    end for
  end for
end for

APPROX, a distributed sniffer-channel selection scheme, where the ith sniffer maintains a local copy of $w_{i,k}, k \in \mathcal{K}$, which is updated based on the reward it receives.

At the core of DEXP3-APPROX, it implements a distributed multinomial sampling scheme for Line $^{\sharp\sharp}$ in Algorithm 4.1. Sampling from a multinomial distribution with parameters $\{p_{s,k}, s \in \mathcal{S}, k \in \mathcal{K}\}$ is equivalent to a two-step sampling process. Let $p_s \triangleq \sum_k p_{s,k}$. First, we pick a sniffer s according to the multinomial distribution with parameters $\{p_s, s \in \mathcal{S}\}$. Next, the chosen sniffer s *locally* selects a channel according to the multinomial distribution with parameters $\{p_{s,k}/p_s, k \in \mathcal{K}\}$. Note that the first step is equivalent to the distributed sensor selection problem in [48], while the second step can be implemented easily on the chosen sniffer. After L rounds of sampling, L sniffers are selected. They play their locally sampled channels and make observations. By the end of the procedure, the selected sniffer s is associated with a tuple (i_s, k_s), namely, its rank (acting as the i_s^{th} agent) and the channel it operates on. By default, all unassigned sniffers are given the rank of $L + 1$.

Determination of rewards is relatively straightforward if all sniffers are within a single broadcast domain. In a multi-hop setting where sniffers are spread out in a large geographical area, only neighboring sniffers operating in the same channel can have overlapping observations. Let N_s be the neighbors of sniffer s who may observe some common users as s. N_s can be conservatively estimated from the radio propagation model and the sensitivity threshold of sniffer s. Sniffer s's observation only needs to be disseminated to N_s. At sniffer s, upon receiving observations from its neighbors in N_s, it computes its reward as,

$$x_{s,k_s} = \left| \bigcup_{j \in N_s, i_j \leq i_s} U_{j,k_j} - \bigcup_{j \in N_s, i_j < i_s} U_{j,k_j} \right|. \tag{4.5}$$

The pseudo code of the DEXP3-approx is given in Algorithm 4.3. In the algorithm, a sniffer s stores weights for each of the L agent $w_{s,k}^l, k = 1, 2, \ldots, K$, a normalizing constant Z_l, an updating weight R_l, a cumulative reward $\hat{G}_{s,k}^l$, a round index r_l and a threshold γ_l, $l = 1, 2, \ldots, L$. We define

Table 4.1: Comparison of computation complexity and regret bounds of various online algorithms

	ε-GREEDY	ε-GREEDY-APPROX	EXP3-APPROX	DEXP3-APPROX‡
Computation complexity	$O(C(M,L)K^L))$	$O(LKS^2(2^S-1)$	$O(LKS)$	$O(L+K)$
Regret bound	$O(\log n)$	$O(\log n)^{\dagger}$	$O(\sqrt{n})^{\dagger}$	$O(\sqrt{n})^{\dagger}$
Message complexity	$\Theta(S)$	$\Theta(S)$	$\Theta(S)$	$\Theta(L)^{\sharp}$

† the regret bound is relative to the offline Greedy.
$C(S,L)$ is the combinatoric number.
‡ indicates per node complexity, $^{\sharp}$ indicates average complexity.

a function,

$$\mathbb{P}(w, Z, R, \gamma, L') \triangleq \frac{(1-\gamma)\dfrac{w}{Z} + \dfrac{\gamma}{SK}}{(1-\gamma)\dfrac{Z-R}{Z} + \dfrac{(S-L')\gamma}{S}},$$

It is easy to show that on average, a total of $\Theta(1)$ sensors are selected in Line †† in Algorithm 4.3. Thus, in each stage, an average of $\Theta(1)$ broadcasts are needed. Therefore, the total average number of messages generated are $\Theta(L)$. Note that since DEXP3-APPROX implements EXP3-APPROX in a distributed manner. The two share the same regret bounds.

Table 4.1 summarizes the computation complexity and regret bounds of channel assignment algorithms discussed thus far. Clearly, we observe a trade-off in the computation complexity (each iteration) and the growth in regret bounds over time. One may argue based on this observation that a unified metric should be introduced that characterizes the regret rate per computation unit. However, it should be noted that message exchanges in learning incur communication costs, while computation in each iteration incurs computation cost. Thus, a single metric is insufficient to capture both costs.

4.3 Approximate Online Policy without Correlation Information using stochastic algorithm

From what we have learned so far, our approximate online policy with correlation information achieves the logarithmic regret compare to the offline Greedy algorithm. However, the computation complexity that we have to pay to maintain the correlation structure is very expensive.

This drawback can be reduced the EXP3-APPROX with the cost of low convergence compare to ε-GREEDY-APPROX. Gaining experience from the drawback of two algorithms, we design this approach with the primitive objective is to obtain not only a competitive computation complexity like EXP3-APPROX but also an excellent convergence speed similar to ε-GREEDY-APPROX.

Though designed for the sniffer-channel selection problem, the idea of multi-agent learning behind EXP3-APPROX is of interest in its own right and is applicable to problems with sub-modular structure but subject to budget constraints. We found the slow convergence of the multi-agent algorithms is primarily due to the adversarial setting. Except the first agent of the system, which can obtain a perfect condition of all the channels (which is stochastic), the reward observed by the followed agents seems to be non-stochastic due to the property of Greedy algorithm. Hence, if each agent utilizes stochastic-setting algorithms (e.g., ε-Greedy) from the beginning, our system faces a high probability of having a linear regret.

However, as an agent converges to its optimal action, the reward observed over channels by its next colleague seems to be "more stochastic." Therefore, our idea here is to adapt the algorithm which each agent utilizes with the environment that it faces. Applying the idea of switching condition in [18], and later in [49], we design ε-GREEDY-AGENT-APPROX in which each agent starts using ε-GREEDY when the switching condition from it right previous agent is satisfied. In order to apply this idea, we define σ as the stability threshold of ε-GREEDY-AGENT-APPROX and assume that we know this parameter.

Algorithm 4.4 The ε-GREEDY-AGENT-APPROX algorithm

Initialize:

- The stability of the algorithm is σ.
- The sequences $\varepsilon_{l,t} \in (0, 1]$, with $t = 1, 2, \ldots$ by

$$\varepsilon_{l,t} \triangleq \min\left\{1, \frac{cK}{d_l^2(t - t_{l-1})}\right\}. \tag{4.6}$$

for $t = 1$ **to** $Stoppingtime$ **do**

- Play agent 1 using $\varepsilon_{1,t}$-GREEDY algorithm.
- t_l is the moment when $\varepsilon_{l,t} < \frac{\sigma}{2}$. At t_l, activate agent $l + 1$, play each arm in of this agent at least m times, then play agent $l + 1$ using $\varepsilon_{l+1,t}$-GREEDY algorithm.
- Observe the feedback and update the estimation of average reward matrix of all active agents.

end for

Each agent $\mathcal{E}_1$, $\mathcal{E}_2$,..., $\mathcal{E}_L$ keeps its average reward matrix $\bar{\mu}^l$ of dimension $S \times K$. At the beginning, agent 1 sniffs though all the channels by finding the appropriate assignment using the ε-GREEDY described in [9]. At any time t, the system checks for the stability threshold of the currently running agents. Whenever this parameter is crossed, the system starts activating the next agent. By using this algorithm, we can achieve a competitive logarithmic regret compare to ε-GREEDY with high probability. The full algorithm can be seen in Algorithm 4.4.

In our algorithm, we define $t_0 = 0$ with the meaning as agent 1 starts running from the beginning. Because each agent uses ε-GREEDY algorithm, the function of parameter c, and d_l must be the same as c and d in ε-GREEDY algorithm of [9], respectively. With agent l, c is chosen to be $c > 5$ for the convergence of the expected regret, and d_l is defined as:

$$0 < d_l \leq \min_{k:\mu_{k,l}<\mu_l^*} \Delta_{k,l}, \tag{4.7}$$

with

$$\Delta_{k,l} = \mu_l^* - \mu_{k,l}. \tag{4.8}$$

In 4.8, μ_l^* and $\mu_{k,l}$ are respectively the expected reward of the optimal and k^{th} assignments of agent l when all previous agents choose their optimal assignment to play.

The key idea of our algorithm is the stability σ of the system. This parameter helps the system determine the best time to trigger an agent with a high probability of $1 - \exp(-\frac{m\sigma}{6})$ of having logarithmic regret. In order to find this parameter, we first define the stability of each agent σ_1, σ_2,..., σ_{S-1} as

$$\sigma_l = \min_k \max \left(\frac{2\mu_{k,l+1}}{\Delta_{k,l+1}}, \frac{\Delta_{k,l+1}}{2(1-\mu_{k,l+1})} \right), \tag{4.9}$$

and

$$\sigma = \min_{l=1,\ldots,S-1} \sigma_l. \tag{4.10}$$

We can see that the stability of the system is chosen to be the smallest value of stability of all agents. This is consistent when the algorithm can only achieve the logarithmic regret with high probability when each agent in it achieves the local logarithmic regret. In reality, we do not let agents stay

Table 4.2: Computation time (s)

Configuration	L	ϵ-GREEDY-APPROX	EXP3-APPROX	DEXP3-APPROX†
4 APs, 3 channels, 3 sniffers	1	219.64	86.74	150.08
	3	916.04	248.30	446.02
9 APs, 3 channels, 6 sniffers	3	944.74	306.59	499.86
	6	3732.90	663.40	1062.1
12 APs, 6 channels, 6 sniffers	3	1409.70	419.74	625.47
	6	4075.80	927.13	1323.00

† The computation time of DEXP3-APPROX is the *total* time on all nodes.

idly before they are activated. To gain advantage, we let all agents choose random arms from the beginning without updating their average reward matrix before they are activated. This shortcut method can partially reduce the regret of the system.

4.4 Simulation Results and Analysis

In this section, we evaluate the performance of the proposed online approximation algorithms. Due to the high computation complexity of ε-GREEDY, we only compare the convergence and computation time of ε-GREEDY-APPROX, EXP3-APPROX, DEXP3-APPROX, and ε-GREEDY-AGENT-APPROX. We use the same environment as in the simulation of the UCB-based algorithm.

Figure 4.2 shows the regrets of ε-GREEDY-APPROX, EXP3-APPROX and DEXP3-APPROX over time. In all scenarios, ε-GREEDY-APPROX converges much faster than the other two algorithms. This is consistent with the analytical results that ε-GREEDY-APPROX converges logarithmically in $O(\log n)$; and EXP3-APPROX and DEXP3-APPROX converge in $O(\sqrt{n})$, where n is time. The slow convergence of EXP3-APPROX and DEXP3-APPROX can be attributed to two factors: i) they utilize multiple agents, each assuming adversarial payoffs, and ii) ε-GREEDY-APPROX utilizes spanner arms in exploitation stages, which allows fast learning.

Table 4.2 summarizes the computation time of a single execution of all algorithms under different scenarios. The algorithms are implemented in Matlab R2009b running on a Windows desktop PC with Intel core i7-2600 CPU@3.4GHz and 8GB RAM memory. As seen from Table 4.2,

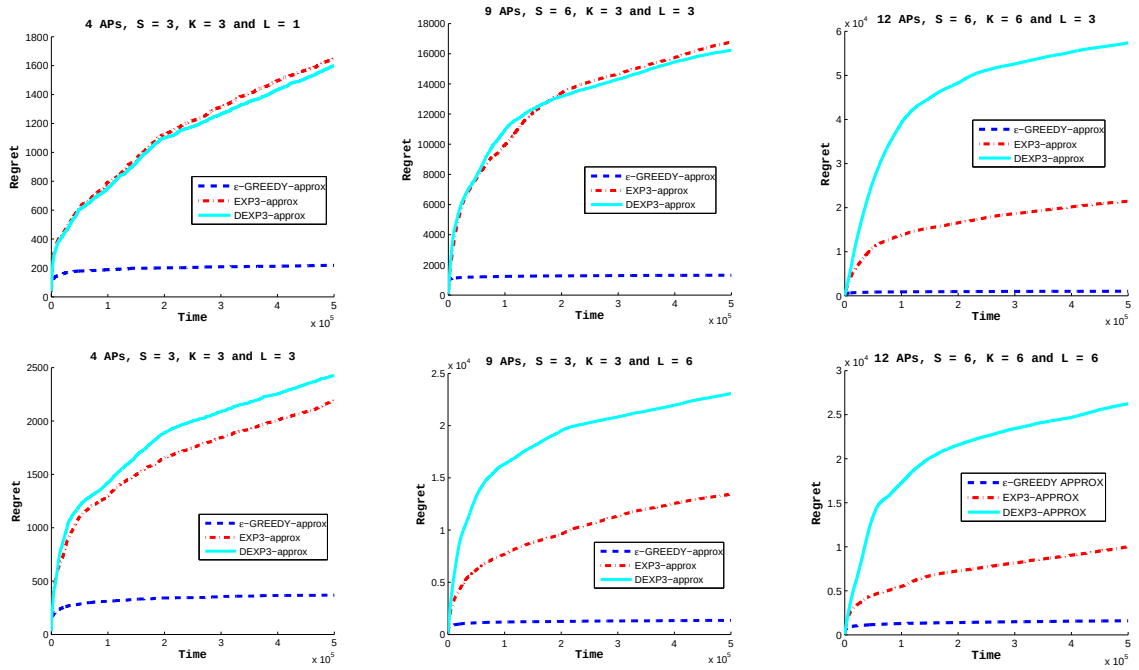

Figure 4.2: Regrets of ε-GREEDY-APPROX and EXP3-APPROX and DEXP3-APPROX

.

increasing the number of sniffers leads to a significant increase in the computation time in both algorithms. However, the increments with ε-GREEDY-APPROX are higher than that with EXP3-APPROX. In contrast, the computation time grows slower when increasing the number of channels. The total computation time of DEXP3-APPROX is higher than EXP3-APPROX. This is because DEXP-APPROX needs to resample when no sniffer is selected in a stage. However, the per-node computation time is expected to be shorter as the computation is done in parallel.

In the second part of this section, we compare three algorithms ε-GREEDY-APPROX, ε-GREEDY-AGENT-APPROX, and an algorithm by running the ε-GREEDY with each agent without caring about the stability of the channel - we call ε-GREEDY-SIMPLE. We evaluate these three approaches in two circumstances. In the first case, the environment is considered to be easy, in which the expected rewards seen by later agents are not very different when previous ones change their assignment. In this situation, the regret showed by three algorithms are approximately close to each other when we share the same probability of exploration between them. This result is in our prediction as all algorithms use ε-GREEDY method to determine the best assignment. The regret

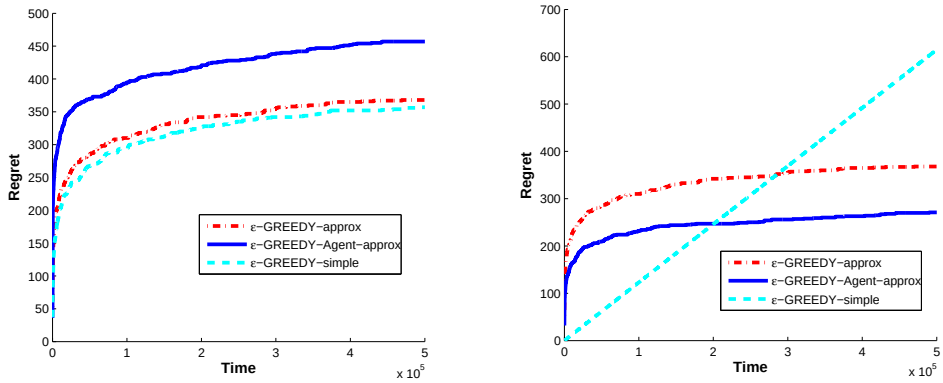

Figure 4.3: Regrets of ε-GREEDY-APPROX and ε-GREEDY-AGENT-APPROX and ε-GREEDY-SIMPLE when the configuration is 4 APs, S = 3, K = 3 and L = 3
.

in this case is the first picture of Figure 4.3.

In the second circumstance, the reward viewed by later agents in suboptimal arm seems to be better than it should be when previous agents choose the suboptimal assignment instead of their optimal one. Moreover, the reward of the optimal assignment of later agents is worse than it should be when previous agents choose the suboptimal assignment instead of the best one. In this case, the regret of both ε-GREEDY-APPROX and ε-GREEDY-AGENT-APPROX are still in logarithmic form while ε-GREEDY-SIMPLE's regret becomes linear. The behavior of these algorithms are showed in the second picture of Figure 4.3.

The linear regret of ε-GREEDY-SIMPLE is because at the time the exploitation rate surpasses the exploration rate of one agent, it previous agent is still busily exploring the environment. Therefore, the algorithm can easily make a mistake by choosing a suboptimal arm and follow it until the end of the experiment. This statement is not true with our ε-GREEDY-AGENT-APPROX when later agents only update their statistic information to find the optimal assignment when previous agents are stable enough. It makes ε-GREEDY-AGENT-APPROX perform as good as ε-GREEDY-APPROX in the achieved regret and out-perform ε-GREEDY-APPROX in the computation time as we can see in Table 4.3.

Table 4.3: Computation time (s)

Configuration	**L**	ϵ-GREEDY-APPROX	ε-GREEDY-AGENT-APPROX
4 APs, 3 channels, 3 sniffers	1	219.64	49.67
	3	916.04	143.05
9 APs, 3 channels, 6 sniffers	3	944.74	180.42
	6	3732.90	362.47
12 APs, 6 channels, 6 sniffers	3	1409.70	252.16
	6	4075.80	562.91

4.5 Conclusions

When we linearly increase the number of sniffers in our experiment, the computation complexity of optimal algorithms also increases exponentially. This problem makes optimal approaches unimplementable in the real environment with a huge set of assignments. In this section, we propose three approximate algorithms with low computation complexity but still guarantee a constant amount of reward compare to optimal algorithms. To achieve this goal, we combine the idea of offline GREEDY with the popular EXP3.1 algorithm to create EXP3-APPROX and DEXP3-APPROX. We also build up ε-GREEDY-AGENT-APPROX with a high probability of achieving logarithmic regret and surpassing ε-GREEDY algorithm in computation time. As showed in the results, our algorithms achieve the regret of the offline GREEDY. Moreover, we can see that all these algorithms have low complexity compare to the previously proposed approach with correlated reward.

Chapter 5

Implementation

The purpose of this chapter is to conduct a small scale implementation for what we have developed so far in this thesis. Therefore, we would like to start with a simple configuration and traditional algorithms to confirm the implementability of our direction over real environments. In the first section of this chapter, we describe IEEE 802.11 standard and the sniffing process we use in the experiment. The second part talks about algorithms that we want to implement and how to apply them in a real time environment. Last but not least are results, its analysis and our conclusions.

5.1 Sniffing Process

In every implementation, tools and the surrounded environment mainly contribute to the received results. Hence, we utilize this section to describe our sniffer, the sniffing process it uses, and IEEE 802.11 standard in which data packets are required to be captured. In our experiment, we use as a sniffer a Dell Latitude E6410 laptop with OS Ubuntu 10.04, Processor Intel(R) Core(TM) i5 CPU M520 @2.40 GHz, RAM 3GB, HDD 200 GB with WLAN 802.11 a/b/g Wireless cardbus adapter. This device uses the sniffing process which is built in library *libpcap* to captures data packets over the set of channel 1, 6, and 11 of IEEE 802.11 standard.

5.1.1 IEEE 802.11 Standard

IEEE LAN/MAN Standards Committee (IEEE 802) first created IEEE 802.11 in 1997. This family consist a set of standards for implementing wireless local area network computer communication in the 2.4, 3.6, 5.0 and 60 GHz frequency bands. It regulates the use of a series of half-duplex over-the-air modulation techniques that use the same basic protocol. The most popular are those defined by the 802.11b and 802.11g protocols which are in the range of our wireless card.

802.11b and 802.11g utilize the 2.4 - 2.5 GHz spectrum (ISM band). This band consists of

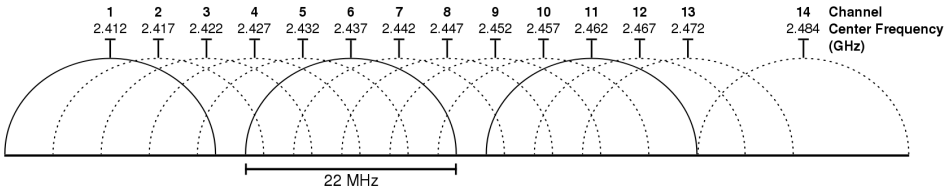

Figure 5.1: Wi-Fi channels in the 2.4 GHz band

13 channels begin with channel 1 centered on 2.412 GHz. The 14th band was proposed in Japan, but later dropped. All the bands space 5 MHz apart and have the bandwidth of 22 MHz regulated by 802.11b and 20 MHz by 802.11g. The availability of channels varies from country to country, constrained by how each country allocates its radio spectrum to various services. While Spain only permits channel 10 and 11, North America allows all channels from 1 through 11. Since 11 frequencies have some overlap parts over the others, three non-overlapping channels 1, 6, and 11 are widely in use in North America.

In order to capture data packets from one over these three Wi-Fi channels, we need to understand how IEEE 802.11 defines a packet as a data packet. This information is regulated in the Frame Control field of a MAC frame. A MAC sub-layer frame is described as a sequence of fields in a specific order as we can see in Figure 5.2. To distinguish a data packet from all packets, we have to take a deeper look inside the first byte of the Frame Control field. Based on IEEE 802.11 standard, the type of a packet is specified in Type and Subtype in Frame Control field. To find a data packet based on IEEE 802.11, a system must check for a frame with protocol version of $00b$ (in which "b" stand for bit counting), the value of Type field as $10b$ for data, and $0000b$ for the Subtype field for data. By using the library of *libpcap* to analyze our captured packets, a data packet should have the value of the first octet as $0x08$ written in hexadecimal system.

5.1.2 Sniffing Process

Our sniffing process *libpcap* on which a Dell laptop runs on was originally developed by the Network Research Group at Lawrence Berkeley Laboratory in *libpcap*. *libpcap* not only provides

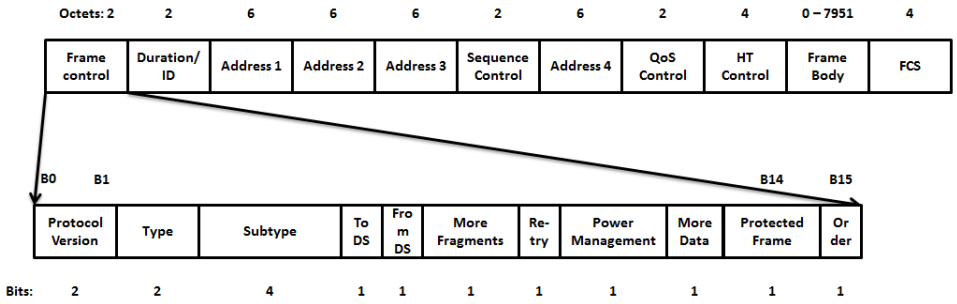

Figure 5.2: MAC frame structure of IEEE 802.11 standard

the packet-capture and filtering engines and protocol analyzers, but also supports saving captured packets to a file and processes it. In our thesis, we only concern about capturing a data packet from a chosen channel and analyze it.

The sniffing process here is divided in to five steps. The first step is to choose an interface, check for its availability, its mode and the frequency we want to use to sniff data packets. In this experiment, the interface *wlan1* which is the notation for the Wireless cardbus adapter is chosen, and set to the *monitor* mode in order to sniff. Finally, we choose a frequency from the set of Wi-Fi channels to tune to.

In the second step, we open the device for sniffing with all the parameters configured in the previous part. Next, a filter is applied to the sniffer so that we can extract the required type of packets we want. To refine a data frame from a sequence of frames, we need to set a filter to extract any MAC frame with the first byte in Frame Control field have the value of $0x08$. When everything is ready, the forth step is to sniff data packets, count it, and finally to close our session. Figure 5.3 generalizes our process.

5.2 Algorithms and Configurations

As we describe in the previous section, after having the sniffing process to capture data packets from Wi-Fi channels, we need to plug this tool into algorithms to observe the efficiency of each policy over real time environments. Therefore, the first part of this section is about algorithms that

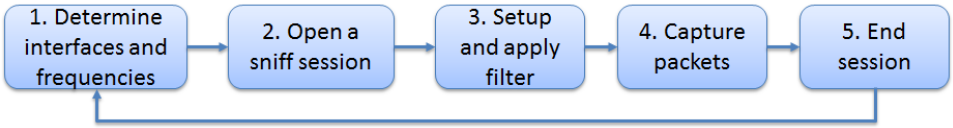

Figure 5.3: A sniffing process

we want to apply in our experiment. Later, we state requirements, our regulations and configurations in order to implement all mentioned approaches in the real world.

5.2.1 Algorithms

In our small scale implementation, we run 4 algorithms and compare their observed average packets over time. Our first algorithm is a naive one, in which it chooses a random channel in the set of $\{1, 6, 11\}$ Wi-Fi channels and observes the data packets on that frequency until the end of the session. This approach is the base line to compare the efficiency of the other algorithms over time. The second policy we want to implement here is EXP3 in [13] (which can be seen in Algorithm 5.1). This policy is used when the channel is supposed to be non-stationary over time.

In EXP3 policy, a constant exploration property is separated equally to all assignments over a system. Within one assignment, an up-to-date exploitation probability is calculated based on the ratio between the assignment's weights over the total one. An arm's weight is proportional to the observed reward, but vice versa to the probability of playing this arm at that time. As a result, the probability of playing one arm each time is the summation of exploration and exploitation probability at that moment. This scheme of finding an appropriate assignment is efficient in an

Algorithm 5.1 The EXP3 algorithm

Parameters: $\gamma \in (0, 1]$.
Initilize: $w_i(1) = 1$ for $i = 1, ..., K$ channels.
for t = 1 **to** Stopping time **do**

- Set $p_i = (1 - \gamma)\dfrac{w_i(t)}{\sum_{j=1}^{K} w_j(t)} + \dfrac{\gamma}{K}$.
- Draw i_t randomly according to $p_1(t)$,...,$p_K(t)$.
- Observe reward $x_{i_t}(t) \in [0, 1]$.
- Update the weight of the played arm as $w_{i_t}(t + 1) = w_{i_t}(t) \exp(\gamma x_{i_t}(t)/(K p_{i_t}(t)))$. The weight of the other arms does not change.

end for

adversarial environment when the weight system can extensively reflect the effect of the observed reward to the playing probability.

Although EXP3 can achieve the regret growth upper bounded by $O\left(T^{1/2}\right)$ in adversarial environments, this regret is only the lower bound when the channels are stochastic. This growth is considered to be poor as statisticians are proved that a upper bound of $O\left(\log T\right)$ can be obtained in this case. It is in [9] where two efficient approaches ε-GREEDY (as in Algorithm 5.2) and UCB1 (Algorithm 5.3) achieve this theoretical regret.

The key idea of ε-GREEDY is to have a decreasing exploration probability ε_t overtime. It also means that the exploitation part increases when we reveal more information about our surrounded environment. While in the exploitation mode, the arm with highest average reward is chosen to play, the system chooses a random arm in the exploration mode. This method of learning and utilizing is practical in stochastic setting when the rewards provided by channels are stationary.

Different from ε-GREEDY, UCB1 utilizes the idea of confident interval to choose an appropriate assignment at the beginning of each time slot. The confident interval is a brilliant generalization about experience a system has with its arms. The more times an assignment is played, the less obscurity it becomes to the system. As a result, the confident interval of a sparely played assignment is longer compare to the one of a frequently chosen assignment. By combining one assignment's confident interval with its average reward, we decide the assignment to observe is the one with this highest sum. In stochastic setting, both UCB1 and ε-GREEDY out-perform the other multi-armed bandits problems with their simplicity and efficiency.

Algorithm 5.2 The ε-GREEDY algorithm

Parameters: $c > 0$ and $0 < d < 1$.
Initialize: Define a sequence $\varepsilon_t \in (0, 1]$, $t = 1, 2, ...$ by
$\varepsilon_t \triangleq \min\left\{1, \frac{cK}{d^2 t}\right\}$.
for $t = 1$ **to** Stopping time **do**

- Let i_t be the machine with the highest current average reward.
- Play i_t with probability $1 - \varepsilon_t$ and play a random machine with probability ε_t.

end for

Table 5.1: The value of the divisor

Algorithm	Room 311 PGH	M. D. Anderson Library	Engineering building
UCB1	150	400	500
EXP3	225	600	750

5.2.2 Configurations

Because all these algorithms work on over time slots, we first need to determine the width of each time slot of our system. From experiments, we determine the width of a time slot to be 0.8 seconds. The width of the time slot affects the reward that we can observe over a time slot. Although this reward is directly related to the number of data frames the sniffer can catch, it is defined in our algorithms to be in the range of $[0, 1]$. This range is extremely important with the two algorithms UCB1 and EXP3 when the reward decides the way our algorithms update the confident interval/weight, which in turn determines the way an arm is chosen to play. Therefore, after counting the observed data frames, we should divide this number to a constant so that the reward is not larger than 1 every time. The chosen constant number depends on the maximum number of data frames that we can capture in a time slot in a place. It is chosen to be double the maximum number of packets in UCB1 algorithm, and three times in EXP3. The divisor's value varies over places and time, and can be seen in Table 5.1 in our experiment.

From what we have seen in theoretical research about the multi-armed bandit problem, an algorithm is evaluated by the growth of its *regret* over time. This is a brilliant and precise evaluation method in theory, but not practical in real implementations. Generally, we have no information about not only the activity of users over time, but also the channel with busiest traffic. As a result, we do not have the expected reward in stochastic setting or the real one in adversarial environment of the best channel. Therefore, the best way to evaluate an algorithm is to measure its *average*

Algorithm 5.3 The UCB1 algorithm

Initialize: Play each machine once.

Loop: Play machine j that maximizes $\bar{\mu}_j + \sqrt{\frac{2 \ln t}{t_j}}$, with $\bar{\mu}_j$ is the average reward of machine j, t_j is the number of time it is played, and t is the overall number of plays so far.

Table 5.2: Order of each algorithm in a period

PLACE		ALGORITHMS			
		Naive	ε-GREEDY	UCB1	EXP3
PGH building	10:00 AM	1	2	3	4
	01:15 PM	4	1	2	3
	02:45 PM	3	4	1	2
	07:30 PM	2	3	4	1
M. D. Anderson Library	10:00 AM	3	1	4	2
	01:15 PM	4	3	1	2
	02:45 PM	2	4	3	1
	07:30 PM	1	3	2	4
Engineering building 1	10:00 AM	1	2	3	4
	01:15 PM	4	1	2	3
	02:45 PM	3	4	1	2
	07:30 PM	2	3	4	1

reward over time which is calculated as the total packets observed by a sniffer over the total time it is active. Using this evaluation, we run both algorithms and test for the average reward these approaches.

5.3 Implementation Results and Analysis

The locations chosen to evaluate all four approaches over the set of $\{1, 6, 11\}$ Wi-Fi channels are: 1) Room 311 PGH building, 2) the second floor of M. D. Anderson Library, and 3) main hall of Engineering building 1 at University of Houston. We evaluate four policies ε-GREEDY, UCB1, EXP3, and the naive algorithm in each place whole day. The running time of an experiment in a place includes 4 periods: from 10:00 am to 11:15 am, from 1:15 pm to 2:30 pm, from 2:45 pm to 4:00 pm, and from 7:30 pm to 8:45 pm. Each period is chosen so as the traffic does not vary in a large scale (e.g., the period of 11:15 am to 1:15 pm is unstable because people usually leave for lunch). Results of each algorithm in one day are also sum up and divided by the number of time it is played to find the average reward of the algorithm over the place.

As in Figure 5.4, from the place of experiments, Room 311 PGH building is the most unstable

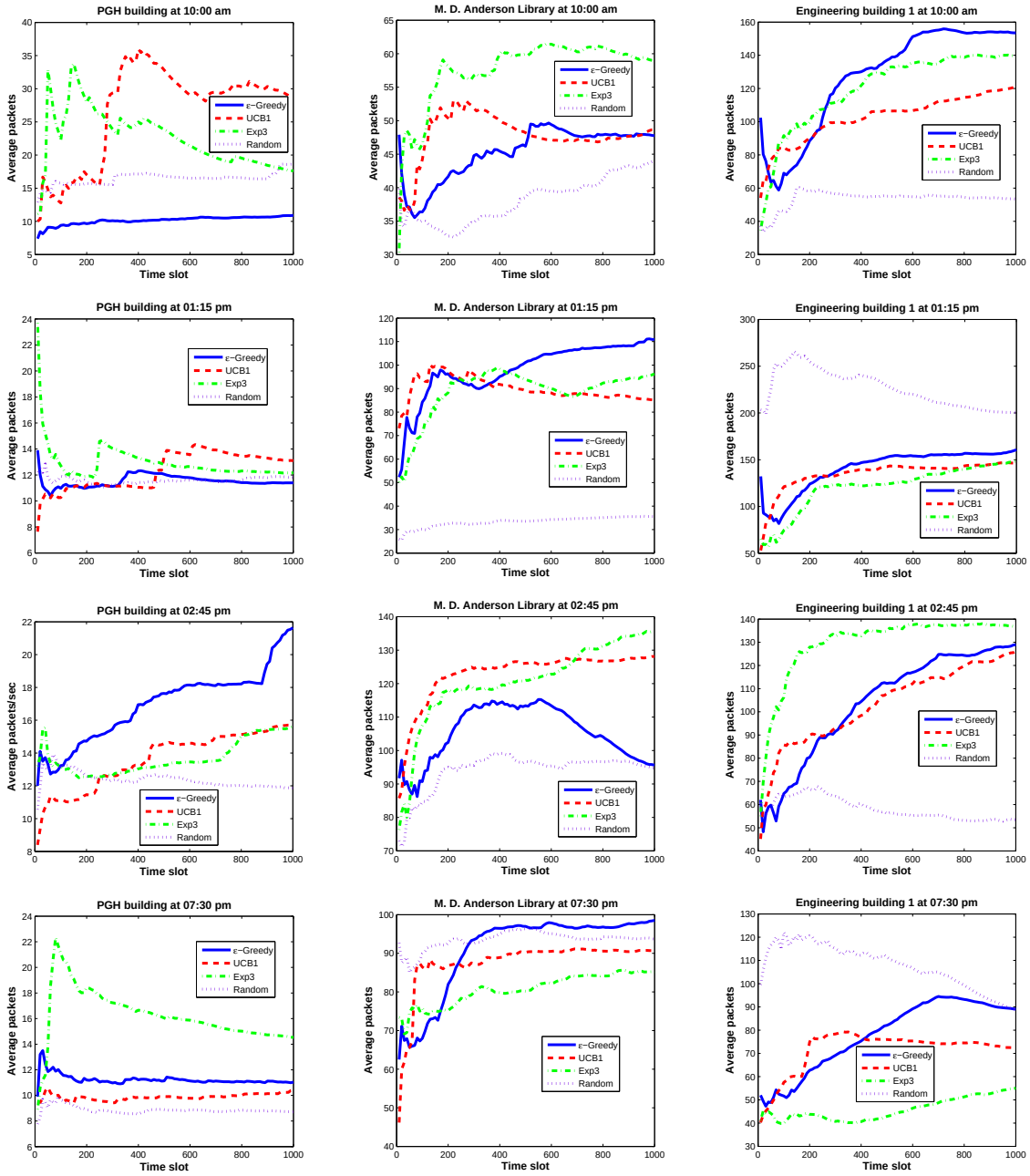

Figure 5.4: Average packets of four algorithms over places in each period of time

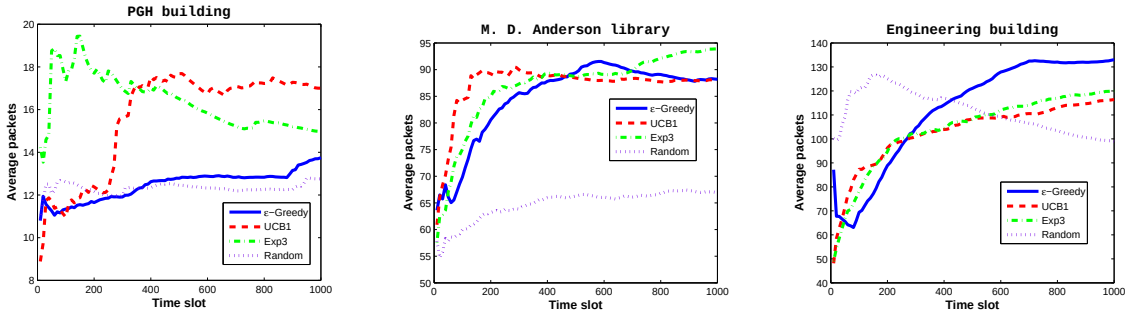

Figure 5.5: Average packets of four algorithms over places in one day

environment when the results of all algorithms vary in a big range. This is consistent because the stable low traffic within the room is largely affected by the unexpected data frames of graduate students passing outside. This phenomenon does not happen in our experiment at the second floor of M. D. Anderson Library or the main hall of Engineering building 1 as data packets created by one user only have a small contribution to the total traffic over experiment period.

Besides, looking at the time of each experiment, we can see that we have a big change in traffic in the third period from 02:45 pm to 04:00 pm. It is expressed as a big drop of average packets per second of the latest running algorithm in Table 5.2. In Engineering building 1, we do not see this phenomenon happens because the latest running algorithm - the naive policy - chose a suboptimal channel (the sixth frequency) to play. We relate this change to users' action of moving from classes to parking lots at University of Houston.

The average packets of each algorithm over a day in one place are plotted in Figure 5.5. In every case, the naive policy is the one observes the smallest average number of packets over time. This is understandable as it randomly access a channel without trying to explore the environment to find the best frequency. While in an unstable area, EXP3 and UCB1 outperform ε-GREEDY, three policies can obtained the same result under a mind environment at M. D. Anderson Library. In Engineering building 1 when the traffic is the most stable in cases, ε-GREEDY is the best choice given our configuration. We also included the number of time each channel is observed by our sniffer in Table 5.3.

Table 5.3: Number of observation over channel of three algorithms

PLACE		ε-GREEDY			UCB1			EXP3		
		1	6	11	1	6	11	1	6	11
PGH building	10:00 AM	75	67	858	142	660	198	143	708	149
	1:15 PM	739	81	180	309	365	326	275	367	358
	2:45 PM	848	74	78	500	221	279	538	190	272
	7:30 PM	60	93	847	289	273	438	671	112	217
M. D. Anderson Library	10:00 AM	84	842	74	264	574	162	136	724	140
	1:15 PM	840	79	81	551	309	140	770	108	122
	2:45 PM	821	95	84	714	191	95	732	165	103
	7:30 PM	856	64	80	458	416	126	342	533	125
M. D. Anderson Library	10:00 AM	830	89	81	808	121	71	751	141	108
	1:15 PM	850	85	65	817	126	57	729	157	114
	2:45 PM	852	66	82	823	111	66	754	139	107
	7:30 PM	79	64	857	266	124	610	201	153	646

5.4 Conclusions

As proved in [9], the growth of ε-GREEDY and UCB1 is $O(\log T)$ over time in the stochastic environment. However, these policies perform poorly when the environment is supposed to be adversarial. In this setting, EXP3 can achieve the regret of $O\left(\sqrt{T}\right)$ over time. Different from theoretical work, the convergence speed of algorithms is also decided by the width of a time slot, the divisor to calculate the reward of each algorithm and the time running these policies. Due to the limitation of time and devices, we only conducted experiments in three places at University of Houston with a fixed set of parameters. In the future, we would like to vary the location, the width of time slot, and the value of divisor parameters to observe the change in average reward seen by different algorithms. We also intend to increase the number of samples to more than one week at each place to get more precise information about the efficiency of these approaches over different environments.

Chapter 6

Conclusion and Future Work

6.1 Summary and Conclusion

In this work, we proposed one centralized optimal online learning algorithm that achieves sub-linear regret bound. Although the optimal algorithm has a well-behaved regret, it suffers a high computation complexity due to MEC is a NP-hard problem. We also proposed two centralized and one distributed approximate online learning algorithms with sub-linear regret bound compared to offline GREEDY algorithm with complete information.

With the first approximate online learning algorithm - ε-GREEDY-APPROX, the system try to obtain greedily the reward by exploiting the vector of correlation θ. By doing this, the computation time of the system can be reduced to only $O\left(LKS^2(2^S-1)\right)$ compare to $O\left(C(M,L)K^L\right)$ of the optimal online learning algorithm. This computation time still suffers high cost due to the structure of θ, and no longer can be reduced to achieve a logarithmic regret compare to the offline GREEDY algorithm. Therefore, we deduce EXP3-APPROX with the idea of multi-agents to have a better computation complexity with the adversarial setting.

Though designed for the sniffer-channel selection problem, the idea of multi-agent learning behind EXP3-APPROX is of interest in its own right and is applicable to problems with sub-modular structure but subject to budget constraints. We found the slow convergence of the multi-agent algorithms is primarily due to the adversarial settings. Except the first agent of the system, which can obtain a perfect condition of all the channels (which is stochastic), the reward which is observed by the followed agents seems to be non-stochastic due to the property of Greedy algorithm. Hence, if each agent runs stochastic-setting algorithms (e.g., ϵ-Greedy) from the beginning, our system faces a high probability of having a linear regret.

However, as each agent converges to its optimal actions the reward we can observe over each channel seems to be "more stochastic." Therefore, idea here is to adapt the algorithm which each

agent uses with the environment that it faces. It is the idea of the ε-GREEDY-AGENT-APPROX. The main challenge here is whenever the system decides the agent which is activating to have "enough convergence" so that the next agent can be activated. However, as we analyze previously in the summary of the ε-GREEDY-APPROX algorithm, by reducing the correlation reward between sniffers, we have to pay the price of "high probability" in the bound to get the advantage of computation time.

The proposed algorithms can provide a lot of benefit to society. As the number of mobile devices is increasing while the frequency spectrum is running out, conducting research into wireless monitoring can help scientists and investors to increase the usage of channels. These techniques will also assist the Internet advertisement and data mining by detecting the favor of users through their search behavior. By developing a huge range of algorithms, we hope that they can serve in a wide range of applications based only the designed advantage.

6.2 Future Work

As we discussed and analyzed in previous chapters, the stochastic multi-armed bandit algorithms cannot fully solve the partial information problem without having the correlation observed rewards. However, this conclusion is generally not true with adaptive algorithms in the reinforcement learning. In [50], Tokic presents an adaptive ε-GREEDY based on value differences. The idea of this paper is to control the exploration ratio by observing the difference between the current and previous average reward of each channel. In stochastic channel, when this difference converges to 0, it means that we are closing to the mean value of the reward. Hence, we can decrease an amount of ε that corresponds to the well explored channel.

The adaptive ε-GREEDY works excellently when user's activities over each channel are assumed to be stochastic. However, it performs poorly when we try to use it with each agent in a multi-agent algorithm. We explain the received bad result by the local structure of value differences function of this adaptive ε-GREEDY algorithm. With a traditional multi-armed bandit problem

with K arms, this algorithm requires to have K local functions to supervise the temporal difference of each channel. Each function only control the average reward each channel without caring about the other channels. This local information directly affects the change of exploration ratio ε. In multi-agent algorithm, these local functions usually converge to 0 before the agent can find its optimal assignment. This property is the result of the activity of the previous agents.

To overcome this draw back, we intend to build up a global value difference of each agent instead of using K local functions. This global value difference function supervises the average reward of the agent, hence only converges when the agent can find its optimal assignment. This property of the new function makes it slowly converge to the optimal assignment compare to K parallel local value difference functions in the stochastic environment, but overcome them when the channels are no long stochastic. The difficulties of the new algorithm and the one in [50] are first - how to choose the temporal value, the temperature of the algorithm so that ε can be guaranteed to converge, and second - how to prove the upper logarithmic regret of the algorithm.

All previous works, we suppose that the environment that we consider is stochastic, or adversarial. However, in reality, the action of the users over the channel has it own pattern. We can model the activity of users over channels as the Markovian chain with finite states. This problem has been considered in [51], [52]. In their problem, the reward is considered to change every time based on the Markov chain model with unknown parameters. In our problem, we suppose that the user with stay a fixed amount of time at a state before moving to the other states with some unknown probability, and each state has a fixed user active probability. By using this model, we can model the stochastic environment as a Markov chain with only one state and the adversarial environment as a Markov chain with infinite number of states. In every case, by using EXP3 we can guarantee to obtain a sub-linear regret. However, as we analyze previously, EXP3 has a low speed of convergence, hence we may not need to use this algorithm when the number of states in the Markov chain is small. When we have an upper bound of the number of stages, the best approach is to observe a channel in epochs to discover the pattern of the channel. Therefore, we would like to design a set of algorithms based on UCB2 in [9] to run on this environment and obtain a well-behaved regret

compares to running EXP3. This work is supposed to be done after the future work based on [50].

Last but not least, we want to extend the scale of our existing experiment in this thesis to a client-server model. In this model, a desktop is connected to a set of laptops through a switch. The desktop with a competitive configuration works as a server of the system. All the laptops behave as sniffers sniffing though channels of the Wi-Fi standard to capture data packets. These packets are then sent to the server to calculate the reward received by the assignment and receive control information from the server to tune to required channels. We expect to implement all of our algorithms proposed in this thesis in the experiment and check for their executability in real environment.

Bibliography

[1] C. I. of Technology. Hixon Fund, *Cerebral mechanisms in behavior the Hixom msymposium, ed. L.A. Jeffress*, Wiley, New York, NY, 1951.

[2] H. Robbins, Some aspects of the sequential design of experiments, Bulletin of the American Mathematical Society **58**(5), 527–535 (1952).

[3] S. Karlin and S. M. Johnson, *A Bayes model in sequential design*, RAND Paper, Santa Monica, CA, 1954.

[4] R. E. Bellman, A problem in the sequential design of experiments, Sankhya **16**, 221–229 (1956).

[5] J. H. Andreae, STELLA: A scheme for a learning machine, in *Proceedings of the 2nd IFAC congress, Basel*, pages 497–502, London, United Kingdom, 1963.

[6] D. A. Berry and B. Fristedt, *Bandit problems: sequential allocation of experiments*, Springer, New York, NY, 1985.

[7] T. L. Lai and H. Robbins, Asymptotically efficient adaptive allocation rules, Advances in Applied Mathematics **6**(1), 4–22 (1985).

[8] R. Agrawal, Sample mean based index policies with O(log n) regret for the multi-armed bandit problem, Advanced in Applied Probability **27**(4), 1054–1078 (1995).

[9] P. Auer, N. C. Bianchi, and P. Fischer, Finite-time analysis of the multi-armed bandit problem, Journal of Machine Learning **2–3**, 235–256 (2002).

[10] P. Rusmevichientong and J. N. Tsitsiklis, Linearly parameterized bandits, Mathematics of Operations Research **35**(2), 395–411 (2010).

[11] P. Auer, Using upper confidence bounds for online learning, in *Proceedings of the 41th Annual Symposium on Foundations of Computer Science*, pages 270–293, Washington DC, Nov. 2000.

[12] V. Dani, T. P. Hayes, and S. M. Kakade, Stochastic linear optimization under bandit feedback, in *Proceeding of Conference on Learning Theory*, pages 355–366, Helsinki, Finland, Jul. 2008.

[13] P. Auer, N. C. Bianchi, Y. Freund, and R. E. Schapire, The non-stochastic multi-armed bandit problem, SIAM Journal on Computing **32**(1), 48–77 (2003).

[14] Y. Freund and R. E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, in *Proceedings of the 2nd European Conference on Computational Learning Theory*, pages 23–37, London, United Kingdom, Mar. 1995.

[15] Y. Freund and R. E. Schapire, Adaptive game playing using multiplicative weights, Games and Economic Behavior **29**(1–2), 79–103 (1999).

[16] N. Littlestone and M. K. Warmuth, The weighted majority algorithm, Journal of Information and Computation **108**(2), 212–261 (1994).

[17] V. G. Vovk, Aggregating strategies, in *Proceedings of the 3rd Annual Workshop on Computational Learning Theory*, pages 371–383, San Francisco, CA, Aug. 1990.

[18] S. Bubeck and A. Slivkins, The best of both worlds: stochastic and adversarial bandits, in *Proceedings of the 25th Annual Conference on Learning Theory*, Edinburgh, Scotland, Jun. 2012.

[19] R. Agrawal, M. V. Hedge, and D. Teneketzis, Asymptotically efficient adaptive allocation rules for the multi-armed bandit problem with switching cost, IEEE Transactions on Automatic Control **33**(10), 899–906 (1988).

[20] T. Jun, A survey on the bandit problem with switching costs, De Economist **152**(4), 513–541 (2004).

[21] R. Kleinberg, A. Slivkins, and E. Upfal, Multi-armed bandits in metric spaces, in *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, pages 681–690, British Columbia, Canada, May 2008.

[22] A. Balachandran, G. M. Voelker, P. Bahl, and P. V. Rangan, Characterizing user behavior and network performance in a public wireless LAN, in *Proceedings of the 2002 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, pages 195–205, Marina Del Rey, CA, Jun. 2002.

[23] T. Henderson, D. Kotz, and I. Abyzov, The changing usage of a mature campus-wide wireless network, in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking*, pages 187–201, Philadelphia, PA, Sep. 2004.

[24] J. Yeo, M. Youssef, and A. Agrawala, A framework for wireless LAN monitoring and its applications, in *Proceedings of the 3rd ACM Workshop on Wireless Security*, pages 70–79, Philadelphia, PA, Oct. 2004.

[25] J. Yeo, M. Youssef, T. Henderson, and A. Agrawala, An accurate technique for measuring the wireless side of wireless networks, in *Proceedings of the 2005 Workshop on Wireless Traffic Measurements and Modeling*, pages 13–18, Seattle, WA, Jun. 2005.

[26] M. Rodrig, C. Reis, R. Mahajan, D. Wetherall, and J. Zahorjan, Measurement-based characterization of 802.11 in a hotspot setting, in *Proceedings of the 2005 ACM SIGCOMM Workshop on Experimental Approaches to Wireless Network Design and Analysis*, pages 5–10, Philadelphia, PA, Aug. 2005.

[27] Y. C. Cheng, J. Bellardo, P. Benko, A. C. Snoeren, G. M. Voelker, and S. Savage, Jigsaw: solving the puzzle of enterprise 802.11 analysis, in *Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pages 39–50, Pisa, Italy, Aug. 2006.

[28] A. Chhetry, H. Nguyen, G. Scalosub, and R. Zheng, On quality of monitoring for multi-channel wireless infrastructure networks, in *Proceedings of the ACM International Symposium and Mobile Ad Hoc Networking and Computing*, pages 111–120, Chicago, IL, Sep. 2010.

[29] C. Chekuri and A. Kumar, Maximum coverage problem with group budget constraints and applications, Approximation, Randomization, and Combinatorial Optimization **3122**, 72–83 (2004).

[30] D. H. Shin and S. Bagchi, Optimal monitoring in multi-channel multi-radio wireless mesh networks, in *Proceedings of the 10th ACM International Symposium on Mobile Ad hoc Networking and Computing*, pages 229–238, New Orleans, LA, May 2009.

[31] K. Liu and Q. Zhao, Distributed learning in multi-armed bandit with multiple players, IEEE Transactions on Signal Processing **58**(10), 5667–5681 (2010).

[32] L. Lai, H. E. Gamal, H. Jiang, and H. V. Poor, Cognitive medium access: Exploration, exploitation and competition, IEEE Transactions on Mobile Computing **10**(2), 239–253 (2011).

[33] L. Lai, H. Jiang, and H. V. Poor, Medium access in cognitive radio networks: A competitive multi-armed bandit framework, in *Proceedings of the 42nd Asilomar Conference on Signals, Systems and Computers*, pages 98–102, Pacific Grove, CA, Oct. 2008.

[34] A. Anandkumar, N. Michael, and A. K. Tang, Opportunistic spectrum access with multiple users: learning under competition, in *Proceedings of IEEE International Conference on Computer Communications*, pages 803–811, San Diego, CA, Mar. 2010.

[35] P. Arora, C. Szepesvari, and R. Zheng, Sequential learning for optimal monitoring of multi-channel wireless networks, in *Proceedings of IEEE International Conference on Computer Communications*, pages 1152–1160, Shanghai, China, Apr. 2011.

[36] P. Arora, N. Xia, and R. Zheng, A Gibbs sampler approach for optimal distributed monitoring of multi-channel wireless networks, in *Global Telecommunications Conference*, pages 1–6, Houston, TX, Dec. 2011.

[37] D. H. Shin, S. Bagchi, and C. C. Wang, Distributed online channel assignment toward optimal monitoring in multi-channel wireless networks, in *Proceedings of IEEE International Conference on Computer Communications*, pages 2626–2630, Orlando, FL, Mar. 2012.

[38] M. J. Streeter and D. Golovin, An online algorithm for maximizing sub-modular functions, in *Neural Information Processing System Foundation*, pages 1577–1584, Vancouver, Canada, Dec. 2008.

[39] J. Y. Audibert, R. Munos, and C. Szepesvari, Tuning bandit algorithms in stochastic environments, in *Proceedings of 18th International Conference on Algorithmic Learning Theory*, pages 150–165, Sendai, Japan, Oct. 2007.

[40] J. Y. Audibert, R. Munos, and C. Szepesvari, Exploration-exploitation trade-off using variance estimates in multi-armed bandits, Theoretical Computer Science **410**(19), 1876–1902 (2009).

[41] Y. Gai, B. Krishnamachari, and R. Jain, Learning multiuser channel allocations in cognitive radio networks: A combinatorial multi-armed bandit formulation, in *IEEE Symposium on New Frontiers in Dynamic Spectrum*, pages 1–9, Singapore, Singapore, Apr. 2010.

[42] T. Le, C. Szepesvari, and R. Zheng, Sequential learning for optimal monitoring of multi-channel wireless networks with switching costs, IEEE Transactions on Signal Processing (in submission) .

[43] A. Garivier and E. Moulines, On upper-confidence bound policies for switching bandit problems, in *Proceedings of the 22nd International Conference on Algorithmic Learning Theory*, pages 174–188, Espoo, Finland, Oct. 2011.

[44] R. Zheng, T. Le, and Z. Han, Approximate online learning for passive monitoring of multi-channel wireless networks, in *Proceedings of IEEE International Conference on Computer Communications*, page (to appear), Turin, Italy, Apr. 2013.

[45] N. C. Bianchi, G. Lugosi, and G. Stoltz, Regret minimization under partial monitoring, Mathematics of Operations Research **31**(3), 562–580 (2006).

[46] N. C. Bianchi and G. Lugosi, Combinatorial bandits, in *Proceedings of the 22th Annual Conference on Learning Theory*, pages 237–246, Montreal, Canada, Jun. 2009.

[47] R. Zheng, T. Le, and Z. Han, Approximate online learning for passive monitoring of multi-channel wireless networks, Journal of Selected Topics in Signal Processing , (in submission).

[48] D. Golovin, M. Faulkner, and A. Krause, Online distributed sensor selection, in *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, pages 220–231, Stockholm, Sweden, Mar. 2010.

[49] S. Bubeck and N. C. Bianchi, *Regret analysis of stochastic and non-stochastic multi-armed bandit problems*, Now Publishers, Norwell, MA, 2012.

[50] M. Tokic, Adaptive ε-Greedy exploration in reinforcement learning based on value differences, in *Proceedings of the 33rd annual German conference on advances in artificial intelligence*, pages 203–210, Karlsruhe, German, Sep. 2010.

[51] C. Tekin and M. Liu, Online algorithms for the multi-armed bandit problem with Markovian rewards, in *Proceeding of IEEE Annual Conference on Communication, Control, and Computing*, pages 1675–1682, Monticello, IL, Sep. 2010.

[52] Y. Gai, B. Krishnamachari, and M. Liu, On the combinatorial multi-armed bandit problem with Markovian rewards, in *Globe Telecommunications Conference*, pages 1–6, Houston, TX, Dec. 2011.