Online Learning in Bandit Problems

by

Cem Tekin

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering: Systems)
in The University of Michigan
2013

Doctoral Committee:

Professor Mingyan Liu, Chair
Professor Satinder Singh Baveja
Professor Demosthenis Teneketzis
Assistant Professor Ambuj Tewari

To my family, Hülya Tekin and Nuh Tekin

ACKNOWLEDGEMENTS

My years in Ann Arbor have been a valuable experience for me. During these years I have seen tremendous improvement in my problem solving and critical thinking skills. Apart from my hard work and commitment, I owe this to the admirable people I met in the University of Michigan.

First and foremost, I would like to thank my advisor, Professor Mingyan Liu, who has been a great mentor for me. I appreciate her wonderful personality, broad technical knowledge and generous support. It was her interest in my research and her motivation that led me to explore many interesting and challenging problems that forms this thesis. I have enjoyed and learned a lot from our discussions which have made a big impact on my career.

I would also like to thank Professor Demosthenis Teneketzis for being in my committee, and for being an excellent teacher. The many courses I have taken from him not only broadened my technical knowledge, but also extended my vision. I would also like to express my gratitude to my committee members Professor Satinder Singh and Professor Ambuj Tewari. Their interest in my research and their expertise in learning problems has been a great incentive for me to write this thesis. Additionally, thanks to all the professors and colleagues from whom I learned a lot.

Finally, my very special thanks to my parents Hülya Tekin and Nuh Tekin for their lifelong support and love. They always believed in me and encouraged me to pursue my goals. Without their guidance and help I would not be at the point where I am today.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

LIST OF APPENDICES

Appendix

ABSTRACT

Online Learning in Bandit Problems

by

Cem Tekin

Chair: Mingyan Liu

In a bandit problem there is a set of arms, each of which when played by an agent yields some reward depending on its internal state which evolves stochastically over time. In this thesis we consider bandit problems in an online framework which involves sequential decision-making under uncertainty. Within the context of this class of problems, agents who are initially unaware of the stochastic evolution of the environment (arms), aim to maximize a common objective based on the history of actions and observations. The classical difficulty in a bandit problem is the exploration-exploitation dilemma, which necessitates a careful algorithm design to balance information gathering and best use of available information to achieve optimal performance. The motivation to study bandit problems comes from its diverse applications including cognitive radio networks, opportunistic spectrum access, network routing, web advertising, clinical trials, contract design and many others. Since the characteristics of agents for each one of these applications are different, our goal is to provide an agent-centric approach in designing online learning algorithms for bandit problems.

When there is a single agent, different from the classical work on bandit problems which assumes IID arms, we develop learning algorithms for Markovian arms by

considering the computational complexity. Depending on the computational power of the agent, we show that different performance levels ranging from optimality in weak regret, to strong optimality can be achieved.

Apart from classical single-agent bandits, we also consider the novel area of multi-agent bandits which has informational decentralization and communication aspects not present in single-agent bandits. For this setting, we develop distributed online learning algorithms that are optimal in terms of weak regret depending on communication and computation constraints.

CHAPTER I

Introduction

1.1 Description and Applications of Bandit Problems

The bandit problem is one of the classical examples of sequential decision making under uncertainty. There is a set of discrete time stochastic processes (also referred to as arms) with unknown statistics. At any discrete time step ($t = 1, 2, \ldots$), each arm is in one of a set of states where the state process is generally assumed to be an independent and identically distributed (IID) or Markovian process. There is a set of players (agents), each selecting an arm at each time step in order to get a reward depending on the state of the selected arm. The evolution of the state of an arm may be independent of the actions of the agents or it may depend on the actions of the agents who select that arm. An agent has partial information about the system, which means that at any time step t, the agent only knows a subset of what has happened up to time t, and is limited to base its decision at time t on its partial information.

As an example, at any time step t, an agent can only observe the state of the arm it selects but not the states of the other arms. Another example is a multi-agent system in which an agent cannot observe the actions of other agents, or can only observe the actions of agents that are located close to it.

The focus of this thesis is to design learning algorithms for the agents that satisfy

various performance objectives by taking into consideration a series of constraints inherent in the system.

Constraints inherent in the system include decentralization, limited computational power of the agents, and strategic behavior. We study various degrees of decentralization ranging from no communication between the agents at any time to full communication between the agents, in which each agent knows all the past observations and actions of all the agents at any time. Computational power of the agents bounds the computational complexity of the algorithms we design for the agents. Strategic behavior makes an agent act in a selfish way to maximize its own total reward, which may not coincide with the actions that will maximize the collective performance of all agents.

While the details are different under different constraints, general idea is to balance exploration and exploitation, i.e., reducing the uncertainty about the statistics of the underlying stochastic process and state of the system by playing the infrequently selected arms, and exploiting the best arms selected according to an optimality criterion based on estimates of the statistics and the state of the system.

Our motivation to study bandit problems comes from its strength in modeling many important sequential decision making scenarios such as those occur in communication networks, web advertising, clinical trials and economics. Several applications of bandit problems are given below.

1.1.1 Random Access in Fading Channels

Consider a network of M decentralized agents/users and K arms/channels. Each user can be seen as a transmitter-receiver pair. Let S^k be the set of fading states of channel k. At each time step channel k is in one of the fading states $s \in S^k$ which evolves according to an IID or Markovian process with statistics unknown to the users. If user i is the only one using channel k at time t with the channel in fading state s,

then it gets a certain (single-user) channel quality given by some $q_k(s)$, where without loss of generality $q_k : S^k \to [0, 1]$; for instance this could be the received SNR, packet delivery ratio or data throughput. When there are n users simultaneously using the channel, then under a collision model in each time step each user has a probability $\frac{1}{n}$ of obtaining access, which results in a channel quality/reward of

$$r_k(s, n) = \frac{1}{n} q_k(s) \ .$$

Although the system is decentralized, each user knows the number of users on the channel it selects by using some signal detection method such as an energy detector or a cyclostationary feature detector. The goal of the users is to maximize the total cumulative expected reward of all users up to any finite time horizon T.

1.1.2 Cognitive Radio Code Division Multiple Access

Consider a network of M decentralized (secondary) agents/users and K (licensed) arms/channels. Primary users who are the licensed users of the channels have the priority in using the channels. Let $s \in \{0, 1\}$ denote the primary user activity on channel k: $s = 1$ if there is no primary user on channel (or channel is available) and $s = 0$ otherwise. A secondary user is only allowed to access the channel if $s = 1$. The primary user activity on channel k is modeled as a two-state Markov chain with state transition probabilities p_{01}^k and p_{10}^k unknown to the users. Multiple secondary users share access to the channel using code division multiple access (CDMA). When channel k is not occupied by a primary user, the rate a secondary user i gets can be modeled as (see, e.g. *Tekin et al.* (2012)),

$$\log\left(1 + \gamma \frac{h_{ii}^k P_i^k}{N_o + \sum_{j \neq i} h_{ji}^k P_j^k}\right),$$

where h_{ji}^k is the channel gain between the transmitter of user j and the receiver of user i, P_j^k is the transmit power of user j on channel k, N_o is the noise power, and $\gamma > 0$ is the spreading gain. If we assume the rate function to be user-independent, i.e., $h_{ii}^k = \hat{h}^k, \forall i \in \mathcal{M}$, $h_{ji}^k = \tilde{h}^k, \forall i \neq j \in \mathcal{M}$, $P_i^k = P^k$, $\forall i \in \mathcal{M}$, which is a reasonable approximation in a homogeneous environment, then we obtain

$$r_k(s,n) = s \log \left(1 + \gamma \frac{\hat{h}^k P_i^k}{N_o + (n-1)\tilde{h}^k P^k} \right).$$

The goal of the users is to maximize the total cumulative reward by minimizing the number of wasted time slots due to primary user activity, and minimizing the congestion due to multiple secondary users using the same channel.

1.1.3 Adaptive Clinical Trials

Clinical trials are one of the main motivations in development of bandit problems. In an adaptive clinical trial several treatments are applied to patients in a sequential manner, and patients are dynamically allocated to the best treatment. Consider K treatments/arms which represent K different doses of a drug. Consider a population $\mathcal{P}$ for which this drug is going to be used. $\mathcal{P}$ may be the citizens of a country, or it can be the set of people with a specific health condition such as diabetes. The effectiveness of a dose can vary from patient to patient in population $\mathcal{P}$ depending on genetic factors, lifestyle, etc. Let S be the space of genetic factors, lifestyle preferences, etc., that determines the effectiveness of the drug. Then, for any $s \in S$, the effectiveness of dose K is given by $r^k(s)$. Let F be the distribution of population $\mathcal{P}$ over S. Then the effectiveness of dose k is

$$\mu^k = \int r^k(s) F(ds).$$

F may be unknown since S can be very large, and not all the people in population $\mathcal{P}$ is categorized based on the factors that determine the effectiveness of the drug. S may be even unknown since all factors that determine the drug efficiency may not be discovered yet. Consider a subset $\mathcal{P}_T$ of population $\mathcal{P}$, which will participate in a clinical trial. We assume that $\mathcal{P}_T$ represents the characteristics of $\mathcal{P}$ well, i.e., the distribution of $\mathcal{P}_T$ over S is also (approximately) F. Patients in $\mathcal{P}_T$ are randomly ordered, and sequentially treated. At each time step one of the doses is given to a patient, then its efficiency is observed at the end of that time step. The goal of the clinical trial is to find the most effective dose for population $\mathcal{P}$ while minimizing the number of patients in $\mathcal{P}_T$ that are given less effective doses.

1.1.4 Web Advertising

A significant portion of the revenues of Internet search engines such as Google, Yahoo! and Microsoft Bing come from advertisements shown to a user based on its search query. Usually, the search engine shows a list of results based on the query with the top result being an advertisement related to that query. If the user clicks on the ad, then the search engine receives a payment from the advertiser. Assume that the search engine has K different ads/arms for a specific query. Let $\mathcal{P}$ denote the set of internet users that use the search engine in a specific country. The search engine does not know the percentage of users in $\mathcal{P}$ which will click on ad k if it is shown for query q. Moreover, this percentage can change over time, due to changing trends, consumer behavior, etc. The goal of the search engine is to display ads in a way that will maximize the number of clicks, hence the revenue. The search engine can maximize the number of clicked ads by balancing exploration of potentially relevant ads and exploitation of estimated best ads.

1.1.5 Online Contract Design

Consider a seller, who offers a set of m contracts $\boldsymbol{x}_t = (x_1, x_2, \ldots, x_m)_t \in \mathcal{X}$ at each time step t to sequentially arriving buyers, where $\mathcal{X}$ is the set of feasible offers. Without loss of generality, we assume that $\mathcal{X}$ includes all offers $\boldsymbol{x} = (x_1, x_2, \ldots, x_m)$ for which $x_i \in (0, 1]$ and $x_i \leq x_{i+1}$ for all i. Let θ_t be the type of buyer arriving at time t, which is sampled from a distribution F that is unknown to the seller. Based on its type, the buyer will accept a single contract in $\boldsymbol{x}_t$ or it may reject all contracts in $\boldsymbol{x}_t$. The preference of a type θ buyer is given in terms of its utility function $U_B(x, \theta)$ which is the payoff the buyer receives when it accepts contract x. In this problem, the seller knows $U_B(x, \theta)$, but it does not know the buyer's type θ or its distribution F. At time t, the seller receives a payoff $U_s(\boldsymbol{x}_t) = u_s(x)$ if the buyer accepts contract $x \in \boldsymbol{x}_t$. For a set of contracts $\boldsymbol{x}$, $E[U_s(\boldsymbol{x})]$ depends on $U_B(x, \theta)$ and the distribution of buyers type F. The goal of the seller is to maximize its expected payoff up to T which is

$$\sum_{t=1}^{T} E[U_s(\boldsymbol{x}_t)].$$

This contract design problem is equivalent to the following bandit problem: Each $\boldsymbol{x} = (x_1, x_2, \ldots, x_m) \in \mathcal{X}$ is an arm. At each time step the seller selects one of the arms $\boldsymbol{x} \in \mathcal{X}$, and receives a random reward $U_s(\boldsymbol{x})$. Let

$$\boldsymbol{x}^* = \arg\max_{\boldsymbol{x} \in \mathcal{X}} E[U_s(\boldsymbol{x})],$$

be the arm (or set of arms) that gives the highest expected reward to the seller. Then, the goal of the seller can be restated as minimizing the regret which is given by

$$TE[U_s(\boldsymbol{x}^*)] - \sum_{t=1}^{T} E[U_s(\boldsymbol{x}_t)].$$

Note that online contract design is different from the previous applications in the following ways. Firstly, $\mathcal{X}$ is an uncountable set. Therefore, exploring each arm separately to assess its average reward is not feasible. Secondly, dimension of $\mathcal{X}$ increases with the number of simultaneous offers m. This implies that the problem has a combinatorial structure. Despite these difficulties, this problem is tractable because the arms are correlated. The expected reward from each arm depends on the buyer's type distribution F.

Due to its different structure, we formulate this problem separately in Chapter VIII, while the formulation for all other bandit problems we consider is given in the next section.

1.2 Problem Definition and Preliminaries

In a bandit problem there are K arms indexed by the set $\mathcal{K} = \{1, 2, \ldots, K\}$. There is a set of agents who select a subset of the arms in a sequential manner. We assume a discrete time model, $t = 1, 2, \ldots$, where state transitions and decisions occur at the beginning of each time slot. Let S^k be the state space of arm k which can be either finite or infinite. In the centralized setting, there is an agent who selects $M \leq K$ of the arms at each time step. In the decentralized setting there are M agents indexed by the set $\mathcal{M} = \{1, 2, \ldots, M\}$, each of which selects a single arm at each time step. An agent has a partial observation of the state of the system which consists of observations of the rewards of the states of the arms that the agent has selected up to the current time.

Upon selecting an arm, an agent receives a reward depending on the state of that arm. The reward of arm k at time t is denoted by $r^k(t)$. This reward depends on the state of the arm, as well as the selections of the other agents.

Initially, an agent does not have any information about how the rewards of the arms are generated. The goal of the agents is to maximize a global objective. To

do this, they should learn how the rewards of the arms are generated and which arms yield highest rewards based on the current state of the system. In the following subsections we propose models for evolution of the arms, interaction between the agents, and give the definitions of performance metrics.

1.2.1 Arm Evolution Models

We consider the following stochastic models for arm state evolution.

Definition I.1. *IID model*: At each time step the kth arm is in a state s which is drawn from a distribution P^k over S^k, independently from other time steps and other arms.

Although, IID model is a simple yet elegant mathematical model for which sharp results can be derived, realistic modeling of many real-world applications require incorporation of temporal information. A more complicated, yet analytically tractable model is the Markovian model. In a Markovian model, the quality of an arm is reflected by its state, which evolves in a Markovian fashion. Below we give several Markovian models.

Definition I.2. *Rested Markovian model*: An arm has two modes, *active* and *passive*. An arm is active if it is selected by an agent, otherwise it is passive. In the active mode the kth arm is modeled as a discrete-time, irreducible and aperiodic Markov chain with a finite state space S^k. When an arm is played, transition to the next state occurs according to the transition probability matrix $P^k = \left(p^k_{xy}\right)_{x,y\in S^k}$, where p^k_{xy} is the transition probability from state x to state y. In the passive mode the state of the arm remains frozen, i.e., it does not change. State changes of different arms are independent.

Definition I.3. *Restless Markovian model*: An arm has two modes, *active* and *passive*. An arm is active if it is selected by an agent, otherwise it is passive. Arm k is

modeled the same way as its rested counterpart in the active mode, independent of other arms. However, in the passive mode the state of the arm changes arbitrarily.

While in the restless Markovian model, we allow a very general model for the passive mode of an arm, some of our results require a stronger assumption which requires a restless arm to be *uncontrolled*, i.e., the state transition of the arm is independent of the actions of agents.

Definition I.4. *Uncontrolled restless Markovian model*: The state evolution of an arm is independent of actions of the agents. An arm is modeled the same way as a rested arm in the active mode.

Many real-world problems can be modeled as a restless Markovian bandit. One example is a patient for which a new treatment is applied. The health condition of the patient can be modeled with a finite set of states which evolve differently when she is under the treatment and when she is not. Moreover many real-world problems are uncontrolled. For example, in a cognitive radio network, the primary user activity is independent of the secondary users' actions.

Under the Markovian models, since each arm in the active mode is a finite state, irreducible, aperiodic Markov chain, a unique stationary distribution exists. Let $\boldsymbol{\pi}^k = \{\pi_x^k, x \in S^k\}$ denote the stationary distribution of arm k. For both the IID and the Markovian models let $\boldsymbol{P} = (P^1, P^2, \ldots, P^K)$.

1.2.2 Reward Models

In this section we list the models which describe how the agents receive rewards from the arms.

1.2.2.1 A Single-agent Reward Model

Reward the agent gets from arm k at time t only depends on the state x_t^k of arm k at time t. We assume without loss of generality that reward from state x of arm k

is

$$r_x^k \in [0, 1],$$

while our results will hold for any reward function that is bounded. The mean reward of arm k is given by

$$\mu^k := \int_{S^k} r_x^k P^k(dx),$$

in the IID arm evolution model, and by

$$\mu^k := \sum_{x \in S^k} r_x^k \pi_x^k,$$

in the Markovian arm evolution model.

1.2.2.2 Multi-agent Reward Models

In the decentralized multi-agent setting interaction between the agents plays an important role in the performance. Agents interact when they select the same arm at the same time. The interaction between agents may change their ability to collect rewards. For example, in a communication network, a transmitter-receiver pair can be regarded as an agent. If two transmitters use the same channel, due to the signal interference the receivers of both transmitters may fail to correctly decode the signal. As a result, communication is delayed, and the energy used in transmitting the signal is wasted. Below we list different interaction models.

Definition I.5. *Collision model*: If more than one agent selects the same arm at the same time step, all of the agents who selected the same arm gets zero reward. A collision only affects the ability of an agent to collect the reward. It does not affect the ability of an agent to observe the reward.

The communication network scenario described above fits to the collision model.

Definition I.6. *Random sharing model*: If more than one agent selects the same arm at the same time step, they share the reward in an arbitrary way. Random sharing only affects the ability of an agent to collect the reward. It does not affect the ability of an agent to observe the reward.

An example of the random sharing model is the *random access scheme* in a communication network. Upon selecting a channel, an agent generates an exponential backoff time (which is negligible compared to the transmission/time slot length). The agent waits, and senses the channel again at the end of the backoff time. If sensed idle (there is no other agent transmitting on the channel), the agent transmits on that channel. Otherwise it does not transmit and selects another channel in the next time slot. Note that in both of the multi-agent models defined above, the mean reward of arm k is the same as the mean reward in the single-agent model.

Definition I.7. *General symmetric interaction model*: An agent who selects arm k at time t gets reward $r_k(x_k^t, n_k^t)$, where x_k^t is the state of arm k at time t and n_k^t is the number of agents on arm k at time t. Without loss of generality we assume that

$$r_k : S^k \times \mathcal{M} \to [0, 1],$$

while our results will also hold for any bounded function.

Let (k, n) denote an arm-activity pair where k denotes the arm's index, and n denotes the number of agents selecting that arm. In the general symmetric interaction model, the mean reward of an arm depends on the agents' selections therefore it is not only a function of the stochastic evolution of the states of that arm. However, mean reward of the pair (k, n) does not depend on the agents' selection and can be written only in terms of the stochastic model of the states of arm k. For the general

symmetric interaction model with the IID arm evolution model the mean reward of arm-activity pair (k, n) is given by

$$\mu_{k,n} := \int_{S^k} r_k(x, n) P^k(dx),$$

and with the Markovian arm evolution model it is given by

$$\mu_{k,n} := \sum_{x \in S^k} r_k(x, n) \pi_x^k.$$

It can be the case that the reward an agent gets decreases as more agents select the same arm with it. In this case the interaction model is called the *interference model*. An example of this is the random access in fading channels scheme given in Section 1.1.

Definition I.8. *Agent-specific interaction model*: An agent who selects arm k at time t gets reward $r_k^i(x_k^t, n_k^t)$, where x_k^t is the state of arm k at time t and n_k^t is the number of agents on arm k at time t. Without loss of generality we assume that

$$r_k^i : S^k \times \mathcal{M} \to [0, 1],$$

while our results will also hold for any bounded function.

For the agent-specific interaction model with the IID arm evolution model the mean reward of arm-activity pair (k, n) for agent i is given by

$$\mu_{k,n}^i := \int_{S^k} r_k^i(x, n) P^k(dx),$$

and with the Markovian arm evolution model it is given by

$$\mu_{k,n}^{i} := \sum_{x \in S^k} r_k^i(x,n)\pi_x^k.$$

In the agent-specific interaction model, the reward an agent gets not only depends on the state and the actions of other agents, but it is also depends on the type of the agent itself. An example of this is the cognitive radio CDMA scheme given in Section 1.1.

1.2.3 Performance Models

In this thesis, unless otherwise stated, we assume that the agents are cooperative. Their goal is to maximize a performance objective such as the sum of the expected total rewards of all agents. Therefore, our goal is to design online learning algorithms for the agents to reach their goal. The loss in performance can be due to the decentralization of agents, the unknown stochastic arm rewards and partial observability of the state of the system.

We compare the performance of the learning algorithms with performance of policies which are given hindsight information or information about the distribution of arm states, or with centralized policies in which agents can agree at each time step on which set of arms to select. The performance loss of an algorithm with respect to such a policy is called the *regret* of the algorithm. We can extend the definition of regret to capture computation, switching and communication costs. Although these costs are not directly related with the stochastic evolution of the states and the agents' ability to collect rewards, they reduce the benefit of the collected reward to an agent. For example when an agent needs to solve an NP-hard problem to find which arms to select in the next time step, we can associate a cost C_{cmp} with such a computationally hard operation. Moreover, we can add switching cost C_{swc}, which

is incurred when an agent changes the arm it selects, and communication cost C_{com}, which is incurred when an agent communicates with other agents to share its information about the system or receive information from other agents. For example, in opportunistic spectrum access, switching cost models the energy and time spent in changing the operating frequency of a radio, while communication cost captures the energy and other resources used to transmit signals between the agents. Without loss of generality, we assume that computation, switching and communication costs are agent-independent, i.e., the cost of each of these is the same for all agents.

Online learning algorithms used by the agents choose the arms based on the past actions and observations. In the single-agent model for an agent using algorithm α, the set of arms selected at time t is denoted by $\boldsymbol{\alpha}(t) = \{\alpha_1(t), \ldots, \alpha_M(t)\}$. In the multi-agent model, when agent i uses algorithm α_i, the arm selected by agent i at time t is denoted by $\alpha_i(t)$, and the set of arms selected by all M agents at time t is denoted by $\boldsymbol{\alpha}(t) = \{\alpha_1(t), \ldots, \alpha_M(t)\}$. Let $E_{\alpha}^{\boldsymbol{P}}$ denote the expectation operator when algorithm α is used, and the set of arm state distributions is $\boldsymbol{P}$. Whenever the definition of expectation is clear from the context, we will drop superscript and the subscript and simply use E for the expectation operator.

Below we give definitions of various performance measures used in single-agent and multi-agent models.

1.2.4 Single-agent Performance Models

In a single-agent bandit, there is no decentralization of information. Therefore, the contribution to regret comes from unknown stochastic arm rewards and partially observable states.

Let $\boldsymbol{\sigma} = \{\sigma_1, \sigma_2, \ldots, \sigma_K\}$ be a permutation of arms such that the mean arm rewards are ordered in a non-increasing way, i.e., $\mu^{\sigma_1} \geq \mu^{\sigma_2} \geq \ldots \geq \mu^{\sigma_K}$. Let $r^k(t)$ be the random variable representing the reward from arm k at time t.

Below is the definition of weak regret for a single agent (stochastic) bandit.

Definition I.9. *Weak regret in the single-agent model*: For an agent using algorithm α, selecting $M \leq K$ arms at each time step, the weak regret up to time T is

$$R^{\alpha}(T) := T \sum_{k=1}^{M} \mu^{\sigma_k} - E_{\alpha}^{\boldsymbol{P}} \left[\sum_{t=1}^{T} \sum_{k \in \boldsymbol{\alpha}(t)} r^k(t) \right],$$

where $\boldsymbol{\alpha}(t)$ is the set of arms selected by the agent at time t.

When we add computation and switching costs, the weak regret becomes

$$R^{\alpha}(T) := T \sum_{k=1}^{M} \mu^{\sigma_k} - E_{\alpha}^{\boldsymbol{P}} \left[\sum_{t=1}^{T} \sum_{k \in \boldsymbol{\alpha}(t)} r^k(t) - C_{cmp} m_{cmp}(T) - C_{swc} m_{swc}(T) \right], \quad (1.1)$$

where $m_{cmp}(T)$ denotes the number of NP-hard computations by the agent by time T, and $m_{swc}(T)$ denotes the number of times the agent switched arms by time T.

Under the IID model, weak regret compares the performance of the algorithm with respect to the optimal policy given full information about the stochastic dynamics. This is also true under the rested Markovian model with a large time horizon T. This is because under these models the optimal policy is a static one. However, under the restless Markovian model, the optimal policy is no longer a static one. In general, the optimal policy dynamically switches between the arms based on the perceived state of the system by the agent. The regret with respect to such an optimal policy is called *strong regret*.

Definition I.10. *Strong regret in the uncontrolled restless single-agent model*: Let Γ be the set of admissible policies which can be computed using the stochastic dynamics of the arms. Those are the policies for which the action of agent at any time depends on the set of transition probabilities $\boldsymbol{P} = \{P^1, P^2, \ldots, P^K\}$, initial belief about the state of the system ψ_0, and past observations and actions. For an agent using a

learning algorithm α, selecting $M \leq K$ arms at each time step, the strong regret by time T is

$$R^{\alpha}(T) := \sup_{\gamma' \in \Gamma} \left(E^{\boldsymbol{P}}_{\psi_0,\gamma'} \left[\sum_{t=1}^{T} \sum_{k \in \boldsymbol{\gamma}'(t)} r^k(t) \right] \right) - E^{\boldsymbol{P}}_{\psi_0,\alpha} \left[\sum_{t=1}^{T} \sum_{k \in \boldsymbol{\alpha}(t)} r^k(t) \right].$$

In the definition of strong regret, we compare the performance of the learning algorithm α with the optimal policy computed without taking computation and switching costs into consideration. Note that when $\boldsymbol{P}$ is known, the optimal policy is computed once, at the beginning, and the agent plays according to that policy. Therefore the optimal policy given $\boldsymbol{P}$ does not depend on the cost of computation. However, a policy which is optimal when there are no switching costs may not be optimal when there are switching costs. When we introduce the switching cost C_{swc}, the strong regret becomes

$$\begin{aligned} R^{\alpha}(T) := \sup_{\gamma' \in \Gamma} & \left(E^{\boldsymbol{P}}_{\psi_0,\gamma'} \left[\sum_{t=1}^{T} \sum_{k \in \boldsymbol{\gamma}'(t)} r^k(t) - C_{swc} m_{swc}(T) \right] \right) \\ & - E^{\boldsymbol{P}}_{\psi_0,\alpha} \left[\sum_{t=1}^{T} \sum_{k \in \boldsymbol{\alpha}(t)} r^k(t) - C_{swc} m_{swc}(T) - C_{cmp} m_{cmp}(T) \right]. \end{aligned} \tag{1.2}$$

Note that the strong regret is a stronger performance measure than infinite horizon average reward. Firstly, it holds for all finite T. Secondly, even if two algorithms have the same average reward, the difference between their asymptotic regret (as $T \to \infty$) can be unbounded. In fact, strong regret specifies how fast the algorithm converges to the optimal average reward. An alternative performance measure is to compare the average reward of the algorithm with the average reward of the optimal policy with known stochastic dynamics. This performance measure is called *approximate optimality*.

Definition I.11. *Approximate optimality in the single-agent model*: Let OPT be the

average reward of the optimal policy given the stochastic dynamics. The learning algorithm α for the agent is ϵ approximately optimal if

$$\liminf_{T\to\infty} E_\alpha^{\boldsymbol{P}}\left[\frac{\sum_{t=1}^{T}\sum_{k\in\boldsymbol{\alpha}(t)} r^k(t)}{T}\right] \geq \epsilon OPT.$$

Similar to the strong regret model, we can incorporate computation and switchings costs to the approximate optimality criterion. Note that any algorithm with sublinear number of computations and switchings in time will have the same average reward as the optimal policy computed without computational and switching costs.

1.2.5 Multi-agent Performance Models

For the multi-agent model we only consider weak regret. For definition of the weak regret we need to consider the interaction between the agents. Firstly, we define the regret for the collision model given in Definition I.5. For all agent interaction models except the agent-dependent interaction model, we assume that agents cannot communicate with each other. Therefore, we do not include to cost of communication for these models.

Definition I.12. *Weak regret in the collision model*: With M decentralized agents, under the agent interaction model given in Definition I.5, when agent i uses algorithm α_i to select a single arm at each time step, the weak regret up to time T is

$$R^\alpha(T) := T\sum_{k=1}^{M}\mu^{\sigma_k} - E_\alpha^{\boldsymbol{P}}\left[\sum_{t=1}^{T}\sum_{i=1}^{M} r^{\alpha_i(t)}(t) I(n_{\alpha_i(t)}^t = 1)\right],$$

where $\alpha_i(t)$ is the arm selected by agent i at time t, and $I(n_{\alpha_i(t)}^t = 1)$, which is the indicator function of the event $\{n_{\alpha_i(t)}^t = 1\}$, indicates that the reward from arm $\alpha_i(t)$

is collected only when agent i is the only agent on that arm, or equivalently

$$R^{\alpha}(T) = T\sum_{k=1}^{M}\mu^{\sigma_k} - E_{\alpha}^{\boldsymbol{P}}\left[\sum_{t=1}^{T}\sum_{k=1}^{K} r^k(t)I(n_k^t = 1)\right].$$

Definition I.13. *Weak regret in the random sharing model*: With M decentralized agents, under the agent interaction model given in Definition I.6, when agent i uses algorithm α_i to select a single arm at each time step, the weak regret up to time T is

$$R^{\alpha}(T) := T\sum_{k=1}^{M}\mu^{\sigma_k} - E_{\alpha}^{\boldsymbol{P}}\left[\sum_{t=1}^{T}\sum_{k=1}^{K} r^k(t)I(n_k^t \geq 1)\right].$$

Even though we will not analyze the random sharing model directly, for the algorithms we propose, the upper bounds on regret we prove in the collision model will also hold for the random sharing model. This is because the observations of the agents remains the same (since we assume that collision does not affect an agents ability to observe the reward, but only effects its ability to collect the reward), while at each time step the collected reward in the random sharing model is greater than or equal to the collected reward in the collision model.

Note that for all the definitions of the weak regret above, we compare the algorithms performance with respect to the strategy that always selects the M best arms, which is an orthogonal configuration, i.e., all agents select different arms. The computation and switching costs can be added to the weak regret in a similar way with the single agent model. For example, the weak regret of the collision model becomes

$$\begin{aligned} R^{\alpha}(T) := T\sum_{k=1}^{M}\mu^{\sigma_k} - E_{\alpha}^{\boldsymbol{P}}\Bigg[\sum_{t=1}^{T}\sum_{k=1}^{K} r^k(t)I(n_k^t = 1) - C_{cmp}\sum_{i=1}^{M} m_{cmp}^i(T) \\ - C_{swc}\sum_{i=1}^{M} m_{swc}^i(T)\Bigg], \end{aligned} \tag{1.3}$$

where $m_{cmp}^i(T)$ is the number of NP-hard computations by agent i by time T and

$m^i_{swc}(T)$ is the number of switchings by agent i by time T.

For the general symmetric interaction model given in Definition I.7, we need to take into account the fact that there may be more than one agent on each arm in the optimal allocation. In the general symmetric interaction model, let $r_{k,n}(t)$ be the random variable which denotes the reward of arm-activity pair (k, n) at time t. It is important not to confuse this random variable with $r_k(x, n)$ which denotes the reward from state x of arm k when there are n agents on it.

Definition I.14. *Weak regret in the general symmetric interaction model*: Given $\mu_{k,n}, \forall k \in \mathcal{K}, n \in \mathcal{M}$, the optimal allocation of arms to agents is the set of allocations

$$\mathcal{A}^* := \arg\max_{\boldsymbol{a}\in\mathcal{A}} \sum_{i=1}^{M} \mu_{a_i, n_{a_i}(\boldsymbol{a})},$$

where $\boldsymbol{a} = (a_1, a_2, \ldots, a_M) \in \mathcal{A}$ is the vector of arms chosen by the agents $1, 2, \ldots, M$ respectively, $n_k(\boldsymbol{a})$, $k \in \mathcal{K}$ is the number of agents on arm k under vector $\boldsymbol{a}$, and $\mathcal{A} := \{\boldsymbol{a} : a_i \in \mathcal{K}, \forall i \in \mathcal{M}\}$ is the set of possible arm selections by the agents. Let $\mathcal{N} := \{\boldsymbol{n} = (n_1, n_2, \ldots, n_K) : n_k \geq 0, n_1 + n_2 + \ldots + n_K = M\}$ be the set of possible number of agents on (agents selecting) each arm, where n_k is the number of agents on arm k. An equivalent definition of the optimal allocation in terms of elements of $\mathcal{N}$ is

$$\mathcal{N}^* := \arg\max_{\boldsymbol{n}\in\mathcal{N}} \sum_{k=1}^{K} n_k \mu_{k,n_k}.$$

Let v^* be the value of the optimal allocation. Then the weak regret by time T is

$$R^\alpha(T) := Tv^* - E^{\boldsymbol{P}}_\alpha \left[\sum_{t=1}^{T} \sum_{i=1}^{M} r_{\alpha_i(t), n^t_{\alpha_i(t)}}(t) \right].$$

It turns out that in the general symmetric interaction model, an agent should form an estimate of the best combination of agents and arms. Forming such an estimate

is a combinatorial optimization problem which is NP-hard in general. Thus, adding computation and switching costs, the weak regret becomes

$$R^{\alpha}(T) := Tv^* - E_{\alpha}^{\boldsymbol{P}}\left[\sum_{t=1}^{T}\sum_{i=1}^{M} r_{\alpha_i(t),n^t_{\alpha_i(t)}}(t) - C_{cmp}\sum_{i=1}^{M} m^i_{cmp}(T) - C_{swc}\sum_{i=1}^{M} m^i_{swc}(T)\right]. \tag{1.4}$$

A generalization of the general symmetric interaction model is the agent-specific interaction model which is given in Definition I.8. In this model, let $r^i_{k,n}(t)$ be the random variable which denotes the reward of arm-activity pair (k,n) to agent i at time t. Since the observations of an agent in this case does not provide any information about the rewards of other agents, agents should share their perception of the arm rewards with other agents in order to cooperatively achieve some performance objective. We assume that whenever an agent communicates with any other agent, it incurs cost C_{com}.

Definition I.15. *Weak regret in the agent-specific interaction model*: Given $\mu^i_{k,n}, \forall k \in \mathcal{K}, i, n \in \mathcal{M}$, the set of optimal allocations is

$$\mathcal{A}^* = \arg\max_{\boldsymbol{a}\in\mathcal{A}} \sum_{i=1}^{M} \mu^i_{a_i,n_{a_i}(\boldsymbol{a})},$$

where $\boldsymbol{a} = (a_1, a_2, \ldots, a_M) \in \mathcal{A}$ is the vector of arms selected by agents $1, 2, \ldots, M$ respectively, and $\mathcal{A} = \mathcal{K}^M$ is the set of possible arm selections by the agents. Let v^* be the value of the optimal allocation. Then the weak regret by time T is

$$R^{\alpha}(T) := Tv^* - E_{\alpha}^{\boldsymbol{P}}\left[\sum_{t=1}^{T}\sum_{i=1}^{M} r^i_{\alpha_i(t),n^t_{\alpha_i(t)}}(t)\right].$$

When computation, switching and communication costs are added, the weak re-

gret becomes

$$R^{\alpha}(T) := Tv^{*} - E_{\alpha}^{\boldsymbol{P}}\left[\sum_{t=1}^{T}\sum_{i=1}^{M} r^{i}_{\alpha_i(t),n^{t}_{\alpha_i(t)}}(t) - C_{cmp}\sum_{i=1}^{M} m^{i}_{cmp}(T) - C_{swc}\sum_{i=1}^{M} m^{i}_{swc}(T) - C_{com}\sum_{i=1}^{M} m^{i}_{com}(T)\right]. \tag{1.5}$$

1.2.6 Degree of Decentralization

In this section we define various degrees of decentralization that may be possible in a multi-agent system. These settings are ordered according to increasing feedback and communication among the agents. Let $r^{i}_{k}(t)$ denote the reward agent i receives from arm k at time t.

Model I.16. No feedback. Under this model, upon selecting arm k at time t, agent i only observes the reward $r^{i}_{k}(t)$, but not the number of other agents selecting arm k (n^{t}_{k}) or state of arm k (x^{t}_{k}).

For example, in a dynamic spectrum access problem where arms are channels, Model I.16 applies to systems of relatively simple and primitive radios that are not equipped with threshold or feature detectors.

Model I.17. Binary feedback. Under this model, upon selecting arm k at time t, agent i observes the reward $r^{i}_{k}(t)$. At the end of the time slot t, agent i receives a binary feedback $z \in \{0, 1\}$ where $z = 0$ means that agent i is the only agent who selected arm k at time t, and $z = 1$ means that there is at least one other agent who also selected arm k at time t.

Similar to the previous example, in a dynamic spectrum access problem, agents who are equipped with a threshold detector can infer if there are other agents who used the same channel with them.

Model I.18. Partial feedback. Under this model, upon selecting arm k at time t, agent i observes the reward $r_k^i(t)$ and acquires the value n_k^t. Moreover, each agent knows the total number of agents M.

Again, Model I.18 applies to a system of more advanced radios, those that are equipped with a threshold or feature detector. Based on the interference received, a user/agent can assess the simultaneous number of users in the same channel. The same can be achieved by each radio broadcasting its presence upon entering the network, and upon selecting a channel.

Model I.19. Partial feedback with synchronization. Under this model, upon selecting arm k at time t, agent i observes the reward $r_k^i(t)$ and acquires the value n_k^t. Each agent knows the total number of agents M. Moreover, the agents can coordinate and pre-determine a joint allocation rule during initialization.

As an example, Model I.19 applies to a system of more advanced radios as in the previous model. Moreover, each radio is equipped with sufficient memory to keep an array of exploration sequences based on its identity, i.e, a sequence of channels that should be selected consecutively.

Model I.20. Costly communication. In this model, agents can communicate with each other, but communication incurs some cost $C_{com} > 0$. Upon selecting arm k at time t, agent i observes the reward $r_k^i(t)$ but nothing more.

In Model I.20, we generally assume that the agents can exchange messages over a common control channel. However, if arms are channels as in the dynamic spectrum access problem, agents can communicate over the arms. A common control channel is not needed in this case.

1.3 Literature Review

In this section we review the literature on bandit problems. We classify the bandit problems according to the number and reward generation process of the arms, correlation between the reward processes of the arms, and the number of agents. We also mention the bandit optimization problems, in which the goal is find computationally efficient methods to calculate the optimal policy. These problems do not involve any learning.

1.3.1 Classical Single-agent Models

The single-agent bandit problem is investigated by many researchers over the past decades. Motivated by clinical trials this problem is first studied in *Thompson* (1933), and the seminal work *Robbins* (1952) set the foundations of the bandit problem. In the classical single-agent models, there is a finite set of independent arms.

Most of the existing literature assumes an IID model for the reward process of each arm. Below we discuss accomplishments in the IID model. Since the optimal policy is a static policy in the IID model, weak regret which is given in Definition I.9 compares the performance of the learning algorithm with respect to the optimal policy. Therefore for the IID model, weak regret is the same as strong regret which is given in Definition I.10. Since they are equivalent we simply call it the regret.

In *Lai and Robbins* (1985) the problem where there is a single agent that plays one arm at each time step is considered, assuming an IID reward process for each arm whose probability density function (pdf) is unknown to the agent, but lies in a known parametrized family of pdfs. Under some regularity conditions such as the denseness of the parameter space and continuity of the Kullback-Leibler divergence between two pdfs in the parametrized family of pdfs, the authors provide an asymptotic lower bound on the regret of any *uniformly good* policy. This lower bound is logarithmic in time which indicates that at least logarithmic number of samples should be taken

from each arm to decide on the best arm with a high probability. They define a policy to be *asymptotically optimal* if it achieves this lower bound, and then construct such a policy. This result is extended in *Anantharam et al.* (1987a) to single agent and multiple plays, in which the agent selects multiple arms at each time step. The policies proposed in these two papers are *index policies*, which assign an index to each arm based on the observations from that arm only, and select the arm with the highest index at each time step. However, complexity of deciding on which arm to select increases linearly in time both in *Robbins* (1952) and *Anantharam et al.* (1987a) which makes the learning policies computationally infeasible. This problem is addressed in *Agrawal* (1995a) where sample mean based index policies are constructed. The complexity of a sample mean based policy does not depend on time since the decision at each time step only depends on the average of the rewards in the previous time steps, not on the reward sequence itself. The policies proposed in *Agrawal* (1995a) are order optimal, i.e., they achieve the logarithmic growth of regret in time, which is shown to be the best possible rate of growth. However, they are not in general optimal because the constant term which multiplies the logarithmic expression in time is not the best possible term.

In all the papers mentioned above, the limiting assumption is that there is a known single parameter family of pdfs for arm reward processes in which the correct pdf resides. Such an assumption virtually reduces the arm quality estimation problem into a parameter estimation problem. This assumption is relaxed in *Auer et al.* (2002), which requires that the reward of an arm is drawn from an unknown distribution with a bounded support. Under this condition, the authors propose an index policy called *the upper confidence bound* (UCB1) similar to the one in *Agrawal* (1995a) which only uses the sample means of the reward sequences. They prove order-optimal regret bounds that hold uniformly over time, not just asymptotically.

In *Auer and Ortner* (2010) a modified version of UCB1 with an order-optimal

regret bound that has a smaller constant than the regret bound of UCB1 is proposed. In *Garivier and Cappé* (2011), the authors propose an index policy, KL-UCB, which is uniformly better than UCB1. Moreover, this policy is shown to be asymptotically optimal for Bernoulli arm rewards. Authors in *Audibert et al.* (2009) consider the same problem with *Auer et al.* (2002), but in addition take into account empirical variance of the arm rewards for arm selection. They provide a logarithmic upper bound on regret with better constants under the condition that the suboptimal arms have low reward variance. In addition, they derive probabilistic bounds on the variance of the regret by studying its tail distribution.

In the papers that are mentioned above, the uniform regret bounds hold for bounded reward distributions. Online learning in bandit problems is extended to heavy-tailed reward distributions in *Liu and Zhao* (2011) using deterministic exploration and exploitation sequences. Specifically, when the reward distributions have central moments up to any order, logarithmic regret uniform in time is achievable. It is also shown that even if the reward distributions have central moments up to a finite order p, sublinear regret uniform in time, in the order $O(T^{1/p})$, is achievable.

Another part of the literature is concerned with the case when the reward process for each arm is Markovian. This offers a richer framework for the analysis, especially more suitable to real-world applications including opportunistic spectrum access. Results in the Markovian reward model can be divided into two groups

The first group is the rested Markovian model given in Definition I.2, in which the state of an arm evolves according to a Markov rule when it is played by the agent, and remains frozen otherwise. Similar to the IID model, it can also be shown that the weak regret and strong regret is the same for the rested Markovian model (after some finite number of time steps) and the optimal policy is a static policy that plays the arms with the highest expected rewards. A usual assumption under this model is that the reward process for each arm is modeled as a finite-state, irreducible,

aperiodic Markov chain. This problem is first addressed in *Anantharam et al.* (1987b) assuming a single parametrized transition probability model, where *asymptotically optimal* index policies with logarithmic regret is proposed. In *Tekin and Liu* (2010), we relax the parametric assumption on transition probabilities, and prove that a slight modification of UCB1 in *Auer et al.* (2002) achieves logarithmic regret uniformly in time for the rested Markovian problem. Different from *Anantharam et al.* (1987b) our result holds for any finite time, and our algorithm, which needs only the sample mean of the collected rewards, is computationally simpler. However, the constant that multiplies the logarithmic term is not optimal. This work constitutes Chapter II of the thesis.

The second group is the *restless Markovian model* in which the state of an arm evolves according to two different Markov rules depending on whether the agent played that arm or not. Clearly, the optimal policy is not necessarily a static policy, thus weak regret and strong regret are different performance measures for this model. This problem is significantly harder than the *rested Markovian* case, and even when the transition probabilities of the arms are known by the agent, it is PSPACE hard to approximate as shown in *Papadimitriou and Tsitsiklis* (1999). Because of this difficulty, most of the authors focus on algorithms whose performance can be evaluated in terms of the weak regret. Specifically, in *Tekin and Liu* (2011b) we propose a learning algorithm which estimates mean rewards of the arms by exploiting the regenerative cycles of the Markov process. This algorithm is computationally simple and requires storage which is linearly increasing in the number of arms. We prove that this algorithm has logarithmic weak regret for a more general restless Markovian model given in Definition I.3. This work constitutes Chapter III of the thesis. In a parallel work, *Liu et al.* (2010), the idea of geometrically growing exploration and exploitation block lengths is used to prove a logarithmic weak regret bound. The difference is that the block lengths in *Liu et al.* (2010) is deterministic, while the block lengths in *Tekin*

and Liu (2011b) are random.

Since weak regret does not provide a comparison with respect to the optimal policy for the restless Markovian model, we study stronger measures of performance in *Tekin and Liu* (2012a) and *Tekin and Liu* (2011a), which forms Chapters V and IV of this thesis. Specifically, in *Tekin and Liu* (2012a) we consider an approximately optimal, computationally efficient algorithm for a special case of the restless Markovian bandit problem which is called the *feedback bandit* problem. This problem is studied in *Guha et al.* (2010) in an optimization setting rather than a learning setting. We combine learning and optimization by using a threshold variant of the optimization policy proposed in *Guha et al.* (2010) in exploitation steps of the proposed learning algorithm. Because of the computational complexity result in *Papadimitriou and Tsitsiklis* (1999), it is not possible to achieve logarithmic strong regret with a computationally feasible algorithm for the restless Markovian model. However, the existence of a learning algorithm with logarithmic strong regret is still an interesting open problem. In *Tekin and Liu* (2011a) we consider this problem, and propose a learning algorithm with logarithmic strong regret uniform in time for the uncontrolled restless Markovian model given in Definition I.4. This result can be seen as a step towards optimal adaptive learning in partially observable Markov decision processes.

Different from our work, in *Dai et al.* (2011) strong regret is considered with a policy-based approach. The authors assume that the agent is given a set of policies which includes the optimal policy. They provide an algorithm where there is a predetermined sequence of blocks with increasing lengths, and during a block the policy with the highest sample mean reward up to date is selected. In essence, the algorithm treats each policy as a policy-arm, and keeps track of the sample mean rewards of each policy-arm. This algorithm achieves $G(T)O(logT)$ strong regret, where $G(T)$ can be an arbitrary slowly diverging sequence. Although this result is promising because it is computationally simple, in a general restless bandit problem the number of policies is

infinite, and the policy space is exponential in the number of arms. Moreover, most of the policies can be highly correlated but the algorithm considers them independently.

Other than the bounds on regret, another interesting study is to derive probably approximately correct (PAC) bounds on the number of explorations required to identify a near-optimal arm. In other words, find the expected number of exploration steps such that at the end of explorations the algorithm finds a near-optimal arm with high probability. *Even-Dar et al.* (2002) and *Mannor and Tsitsiklis* (2004) are some examples of the research in this direction.

Similar to learning in bandit problems, learning in unknown Markov decision processes (MDPs) is considered by several researchers. In a finite, irreducible MDP with bounded rewards, logarithmic regret bounds with respect to the best deterministic policy is considered in *Burnetas and Katehakis* (1997); *Ortner* (2007); *Tewari and Bartlett* (2008); *Ortner* (2008). In *Burnetas and Katehakis* (1997) the authors propose an index-based learning algorithm, where the indices are the inflations of right-hand sides of the estimated average reward optimality equations based on Kullback-Leibler (KL) divergence. Although not computationally feasible, assuming that the support of the transition probabilities are known by the agent, they show that this algorithm achieves logarithmic regret asymptotically, and it is optimal both in terms of the order and the constant.

The same problem is also studied in *Tewari and Bartlett* (2008), and a learning algorithm that uses l_1 distance instead of KL divergence with the same order of regret but a larger constant is proposed. Different form *Burnetas and Katehakis* (1997), knowledge about support of the transition probabilities is not required. A learning algorithm with logarithmic regret and reduced computation, which solves the average reward optimality equation only when a confidence interval is halved is considered in *Auer et al.* (2009), and learning in an MDP with deterministic transitions is studied in *Ortner* (2008).

1.3.2 Classical Multi-agent Models

Unlike classical single-agent models, multi-agent bandit problems became a popular area of research recently. Many practical applications involving multi-agent dynamic resource allocation can be analyzed using the bandit framework. The properties of the multi-agents models that are not present in the single-agent models include informational decentralization, strategic/selfish behavior and communication between the agents. In a classical multi-agent bandit problem there is a finite set of independent arms.

Most of the relevant work in multi-agent bandit problems assumes that the best static configuration of agents on arms is such that at any time step there is at most one agent on an arm. We call such a configuration an *orthogonal configuration*. Multi-agent bandits with IID reward model is considered in *Liu and Zhao* (2010) and *Anandkumar et al.* (2011), and distributed learning algorithms with logarithmic regret are proposed assuming that the best static configuration is an orthogonal configuration. Specifically, the algorithm in *Liu and Zhao* (2010) uses a mechanism called *time division fair sharing*, where an agent shares the best arms with the others in a predetermined order. Whereas in *Anandkumar et al.* (2011), the algorithm uses randomization to settle to an orthogonal configuration, which does not require predetermined ordering, at the cost of fairness. In the long run, each agent settles down to a different arm, but the initial probability of settling to the best arm is the same for all agents.

In addition to the IID reward model, some researchers considered the restless Markovian model. In *Tekin and Liu* (2012d), we design a distributed learning algorithm with logarithmic weak regret for the restless Markovian model. Our approach is based on a distributed implementation of the regenerative cycle algorithm we proposed for the single-agent bandits. This work forms Chapter VI of the thesis. Different from our work, in *Liu et al.* (2011) the authors propose a learning algorithm based

on deterministic exploration and exploitation blocks, which also achieves logarithmic weak regret for the restless Markovian model.

Although the assumption on optimality of orthogonal configuration is suitable for applications in communication systems such as random access or collision models, it lacks the generality for applications like code division multiple access and power control in wireless systems. In a general resource sharing problem, agents may still get some reward by sharing the same resource. This motivates us to have an agent-centric approach to online resource sharing problems. Specifically, based on the characteristics of the agents including computation power, switching costs, communication ability and degree of decentralization, we propose online learning algorithms with variaous performance guarantees in *Tekin and Liu* (2011c, 2012b). This work constitutes Chapter VII of the thesis. Specifically, in *Tekin and Liu* (2011c) we propose a randomized algorithm with sublinear regret with respect to the optimal configuration of agents for the IID model, and in *Tekin and Liu* (2012b) we propose algorithms based on deterministic sequencing of exploration and exploitation with logarithmic weak regret with respect to the optimal configuration which work for both the IID and restless Markovian models.

1.3.3 Models with Correlation

Both the classical single-agent and multi-agent models consider finite number of independent arms. Therefore using the algorithms designed for these models will not result in optimal performance when the arms are correlated, i.e., the reward of an arm at time step t can depend on the reward of another arm at time step t. In these settings, learning algorithms that exploit the correlation between the arms perform better. In the literature there are many different assumptions about the correlation structure. However, we can group these into two main areas: bandits with finite number of arms with a combinatorial structure and bandits with infinite number of

arms.

Usually the classical models focus on achieving logarithmic regret in time, without considering the dependency of regret on the number of arms (which is linear in most of the cases), while models that exploit correlation between the arms also try to reduce the growth of regret with the number of arms (sublinear or logarithmic for finite number of arms). When there are infinitely many arms sublinear regret bounds (in time) can be achieved by exploiting the correlation. Unless otherwise stated, the results for the models with correlation is usually restricted to a single-agent.

When the number of arms is finite, an IID model where expected rewards of the arms are correlated through a linear function of an unknown scalar whose distribution is known by the agent is considered in *Mersereau et al.* (2009), where the agent knows the distribution of the unknown scalar. Under some assumptions on the structure of known coefficients, they prove that a greedy algorithm that chooses the arm with the highest posterior mean reward is optimal in the infinite horizon discounted setting, and the play settles to the best arm with probability one. This is in contrast to the incomplete learning theorem stated in *Brezzi and Lai* (2000), which says that in the classical bandit setting, the agent needs to indeterminately switch between exploration and exploitation in order to avoid the possibility of settling down to a suboptimal arm. The authors also show that under the undiscounted setting the asymptotic weak regret of the greedy policy is finite, contrary to the unbounded $O(\log T)$ regret results in the classical bandit problems. In *Pandey et al.* (2007) a model where the arms are separated into clusters, with the correlation between the arms in a cluster is described by a generative model with unknown parameters is considered. Authors propose a two stage algorithm that first chooses a cluster, and then an arm within that cluster. Numerical results show that exploiting the dependencies in a cluster reduces the regret.

A combinatorial bandit framework is studied in *Gai et al.* (2012a, 2011, 2012b)

with IID, rested Markovian and restless Markovian models respectively. In a combinatorial bandit an agent chooses a set of arms. Reward the agent receives is a linear combination of the rewards of the individual arms. In addition to receiving the combination of rewards the agent observes the individual rewards generated by each selected arm. The number of arm combinations that the agent can choose from is exponential in the number of arms, therefore classical bandit algorithms such as UCB1 suffers a regret exponential in the number of arms for this problem. Moreover UCB1 has to store an index for each combination, which also makes the storage exponential in the number of arms. The authors overcome this problem by proposing algorithms that update the index of each arm separately, and compute the optimal pair based on a bipartite matching. Both the storage and the regret of these algorithms are polynomials in the number of arms.

When there are infinitely many arms, the correlation is usually given by a distance metric that relates the distance between the arms with the distance between their expected rewards. Examples of this line of work are given in *Rusmevichientong and Tsitsiklis* (2010); *Bartlett et al.* (2008); *Dani et al.* (2008); *Jiang and Srikant* (2011). Specifically, linearly parameterized bandits in which the expected reward of each arm is a linear function of an r-dimensional random vector is considered in *Rusmevichientong and Tsitsiklis* (2010). A similar linear optimization formulation of the bandit problem is considered in *Dani et al.* (2008). A high probability regret bound is considered in *Bartlett et al.* (2008) for the problem studied in *Dani et al.* (2008). In *Jiang and Srikant* (2011) the authors propose another parametric model, in which an arm is associated with a finite dimensional attribute vector. In all of the papers mentioned above, learning algorithms with sublinear regret bounds are proposed.

In *Kleinberg et al.* (2008) the authors study *Lipschitz* bandits, where agents' actions form a metric space, and the reward function satisfies a *Lipschitz* condition with respect to this metric. They provide a sublinear lower bound on regret, and propose

a *zooming* algorithm which achieves regret arbitrarily close to the lower bound.

In *Tekin and Liu* (2012c), we model an online contract selection problem as a bandit problem with uncountable number of arms and a combinatorial structure. Different from the work on bandits with infinitely many arms, we exploit the combinatorial structure to prove a sublinear regret bound that has linear dependence on the dimension of the problem. This work forms Chapter VIII of the thesis.

Another type of bandits in which the correlation structure is exploited is contextual bandits (bandits with side information). In a contextual bandit there is a predetermined unknown sequence of context arrivals. Different from the classical bandit setting, the decision of the agent is also based on the newly arrived context. The goal of the agent is to choose the best arm given the context. One of the early notable work in this area is *Wang et al.* (2005) in which exploitation of side information for a two-armed bandit setting is considered. Usually, the number of arms is infinite and the agent is given a similarity metric by which it can deduce the correlation between different context-arm pairs. Important papers under large arm sets are *Langford and Zhang* (2007), which provides an epoch-greedy algorithm with sublinear regret, *Slivkins* (2009), which gives tight upper and lower bounds on the regret when the agent is provided with the similarity information, and *Rosin* (2011), which considers episodic context arrivals.

1.3.4 Non-stationary and Adversarial Models

In this section we review the literature on bandit problems with rewards generated either by a non-stationary process or by an adversary. In non-stationary bandits either the number of arms or arm reward distributions vary dynamically over time, while in adversarial bandits the arm rewards are generated by an adversary whose goal is to minimize the total reward of the agent. The adversary can be an *oblivious* adversary, which means that at the beginning it can choose a sequence of arm rewards according

to the agent's algorithm to minimize the agent's total reward, but once the agent starts playing it cannot adaptively change the sequence of arm rewards based on the history of play of the agent. The adversary capable of doing this is called the *adaptive* adversary. Since the set of possible strategies for an adaptive adversary includes the set of strategies for an oblivious adversary, the performance of the agent will be worse for an adaptive adversary compared to an oblivious adversary.

The seminal work *Auer et al.* (2003) considers the adversarial bandit problem, and proves sublinear upper and lower bounds on weak regret for an oblivious adversary. The weak regret in this problem compares the performance of the agent's learning algorithm with the arm that yields the highest expected reward given the hindsight. Their proof of the lower bound shows that there exists reward distributions for which no learning algorithm can have a weak regret better than $O(\sqrt{T})$. Their upper bound matches the lower bound up to a logarithmic factor. Since the oblivious adversary can be seen as the worst-case reward distribution, their algorithm yields the same weak regret result under both IID and Markovian reward models. In *Bianchi and Lugosi* (2009) the authors study an adversarial-type combinatorial problem where at each time step an agent chooses a binary vector from a subset $\mathcal{S}$ of K-dimensional vectors and an adversary chooses a K-dimensional loss vector. They provide $O(\sqrt{TK \log |\mathcal{S}|})$ weak regret bound with respect to the best fixed binary vector for this case. A special case of this problem is considered in *Kale et al.* (2010), and a similar but more general adversarial linear optimization version is considered in *McMahan and Blum* (2004), where $O(T^{3/4}\sqrt{\log T})$ weak regret bound is proved (ignoring the dependence on size of $\mathcal{S}$ which may be infinite in this case). In *Hazan and Kale* (2009) the authors propose a novel $O(\sqrt{Q})$ regret bound, where Q is the total observed variation in total loss. Although the regret increases with increasing Q, this bound is never worse than $O(\sqrt{T})$. When the adversary is adaptive, it is shown in *Arora et al.* (2012) that the weak regret with respect to the best constant action sequence cannot be sublinear

in time. Therefore the authors consider a memory-bounded adaptive adversary, who can set arm rewards adaptively only depending on the recent actions of the agent, and show that it is possible to achieve sublinear weak regret.

Different from the research on adversarial bandits described above, in *Garivier and Moulines* (2008) the authors consider a non-stationary model where arm reward distributions are IID and fixed through an interval and change at random time instants. They provide $O(\sqrt{T})$ lower bound on regret and propose two algorithms: *discounted UCB* with $O(\log T\sqrt{T})$ weak regret and *sliding window UCB* with $O(\sqrt{\log TT})$ weak regret. A non-stationary IID model where mean reward changes according to a Brownian motion with reflecting boundaries is studied in *Slivkins and Upfal* (2008). They try to minimize the average cost per step compared to an algorithm which selects the arm with the highest expected reward at each time step. Specifically, they consider a *state oblivious* case where the agent only observes the reward and a *state informed* case where the agent not only observes the reward but also observes the expected reward of the selected arm. They propose algorithms with $O(K\sigma^2)$ and $O(K\sigma)$ average cost per step for the *state informed* and *state oblivious* cases respectively, based on the volatility $\sigma \in [0, 1)$ of the Brownian motion, where K is the number of arms.

1.3.5 Bandit Optimization Problems

In the bandit optimization problems agents are given the stochastic dynamics of the system. Hence the difficulty is the computational complexity of the optimal solution. The original rested multi-armed bandit optimization problem is proposed in the seminal work *Gittins and Jones* (1972) under the infinite horizon discounted setting with independent arms. They showed that the problem of finding the optimal policy reduces to computing K dynamic allocation indices, one for each arm based on the history of play only on that arm. The optimal policy selects the arm with the highest index at each time step. Due to the rested nature of the problem, only the

index of the selected arm is updated at each time step, while the indices of other arms remain constant. *Whittle* (1980) showed the optimality of dynamic allocation indices via a dynamic programming approach by computing *retirement* values for each arm. Alternative proofs of the optimality of dynamic allocation indices, and extensions are given by many others, including *Whittle* (1981); *Varaiya et al.* (1985); *Weber* (1992); *Tsitsiklis* (1994); *Frostig and Weiss* (1999). Specifically, *Whittle* (1981) considers *arm-acquiring bandits*, where new branches of arms are formed from a selected (root) arm.

The restless bandit optimization problem is first proposed in *Whittle* (1988), under infinite horizon average reward (undiscounted) setting. A heuristic index policy (*Whittle's index policy*) is formed by relaxing the constraint of selecting a single arm at each time step, to selecting a single arm on average. The linear program formed this way is later named *Whittle's LP*, and arms are decoupled via a Lagrangian argument. Although no optimality result is established for Whittle's index policy, for most of the instances its near-optimality is justified by numerical results. A special case of this problem called *monotone bandits* is considered in *Guha et al.* (2010), and an approximately optimal index policy based on a *global* Lagrange multiplier is proposed. A special case of two-state, identical, positively correlated arms is studied by *Ahmad et al.* (2009). They showed that a myopic policy which chooses the arm with the highest probability of being in a good state is optimal for both the discounted and undiscounted settings.

General conditions for indexability of bandit problems are studied in *Bertsimas and Niño-Mora* (1996); *Nino-Mora* (2001). Comprehensive discussion on bandit optimization problems and their applications can be found in *Gittins et al.* (1989); *Mahajan and Teneketzis* (2008); *Bergemann and Valimaki* (2006).

1.4 Our Contributions

In most of the systems, agents do not know the dynamics of the system a priori. They learn the dynamics over time as a result of interaction with the system. In this thesis we do not assume any a priori knowledge about the dynamics of the arms. An agent's task is to learn the stochastic dynamics by estimating system parameters based on its own information, while incurring minimal loss in terms of system payoff with respect to an agent which knows the dynamics of each arm perfectly. Some researchers focus on unknown but stationary dynamics by assuming that each arm evolves according to an independent and identically distributed (IID) process. In this case, the problem is reduced to accurately estimating the expectation of the distribution of each arm. We mainly focus on Markovian dynamics, which captures the fact that the average quality of an arm may change over time. For example, in web advertising the preferences of customers may change over time, or in clinical trials a virus may mutate so that its response to a drug may change over time.

Our contributions can be divided into two main sections. The first is the design of algorithms with various complexity and performance constraints for the single-agent bandits. The second is the design of distributed algorithms for the decentralized multi-agent bandits by taking into account additional constraints such as the degree of decentralization and communication requirements.

1.4.1 Algorithms for Single-agent Bandits

For a single agent and Markovian arm rewards we can specify our achievements as follows. Our first class of algorithms are index policies based on the agent's sample mean estimate of the quality of each arm using a regenerative cycle approach to transform a Markovian observation process into an IID one. For this class of algorithms we prove logarithmic weak regret results that holds uniformly in time, which implies that the agent can run these algorithms for an indeterminate amount of time, with-

out knowing the time horizon *a priori*. Moreover, these algorithms are very simple to implement since the agent can calculate the indices of arms in a recursive manner by only keeping track of the sample mean of the observations from each arm.

The second class of algorithms we propose is optimal in the strong sense. That is, they achieve logarithmic strong regret uniformly in time. We provide a characterization of arm statistics under which strong regret is achievable. The main problem with strong regret algorithms is that they need to solve average reward optimality equations based on the estimated statistics, which is computationally intractable in general.

Our third class of algorithms provides a bridge between the first two classes. Instead of logarithmic regret with respect to the optimal policy, they are guaranteed to accrue reward within a constant factor of the optimal policy for the infinite horizon average reward problem. This tradeoff in performance allows us to design algorithms with linear complexity in the number of arms, and polynomial complexity in time.

We summarize our achievements for single-agent models in the list below:

(1) For the rested Markovian model (given in Definition I.2) with $M \leq K$ plays at each time step, UCB (given in Figure 2.1) has weak regret (given in Definition I.9) of $O(K \log T)$, uniformly in T (see Theorem II.3).

(2) For the restless Markovian model (given in Definition I.3) with $M \leq K$ plays at each time step, RCA (given in Figure 3.3) has weak regret (given in Definition I.9) of $O(K \log T)$, uniformly in T (see Theorem III.2).

(3) For the uncontrolled restless Markovian model (given in Definition I.4) with a single play at each time step, IRCEP (given in Figure 4.1) has strong regret (given in Definition I.10) of $O(\log T)$, uniformly in T (see Theorem IV.22).

(4) For a special case of the uncontrolled restless Markovian model (given in Definition I.4) called the *feedback bandit* model, with a single play at each time step,

ABA (given in Figure 5.5) is $(2+\epsilon)$ *approximately optimal* (given in Definition I.11) (see Theorem V.12).

(5) For the contract selection problem proposed in Chapter VIII, in which the number of arms is exponential in the number of contracts, and each contract can be selected from an uncountable set, when a continuity property holds for the seller's expected reward from the bundles of contracts, algorithms TLVO and TLFO (given in Figures 8.3 and 8.5), which require the time horizon T as an input, achieve sublinear regret uniformly in T (see Corollaries VIII.7 and VIII.9).

1.4.2 Algorithms for Multi-agent Bandits

As we mentioned before, multi-agent bandits have informational decentralization and communication aspects which are not present in single-agent bandits. Similar to the single-agent bandits, our regret bounds for the multi-agent bandits hold uniformly over time, and the agents do not need to know the time horizon to adjust their learning rates.

In the fully-decentralized multi-agent model without any communication and feedback, agents can converge to an equilbirium strategy of a well defined game, by using a distributed learning algorithm.

In the decentralized multi-agent model with restless Markovian arms, when at each time step a binary feedback about other agents' activity is available to the agents, we show that logarithmic regret with respect to the best static (centralized) policy with known statistics is achievable with a distributed regenerative cycle algorithm. This model is restricted in the sense that the optimal policy is the one where each arm is selected by at most one agent, i.e., the optimal policy is an orthogonal policy.

In the partially decentralized model (where each agent observes the number of agents using the same arm with it) with IID arms, we show that sublinear weak regret with respect to the optimal policy (with known statistics) is achieved with a

distributed randomized algorithm using a decreasing exploration rate. For both IID and restless Markovian arms, if synchronization between the agents is possible, then logarithmic weak regret with respect to the best static policy is achievable by using distributed sequencing of exploration and exploitation.

We summarize our achievements for multi-agent bandits in the list below:

(1) For the IID model (given in Definition I.1), with $M \leq K$ agents, each playing a single arm at each time step, when agents receive rewards according to a special case of the general symmetric interaction model (given in Definition I.7), in which the reward an agent gets is the quality of the arm multiplied by the interference from other agents, when no communication or feedback is available to the agents, using Exp3 (given in Figure 7.1), the joint play of agents converge to an equilibrium of a well defined congestion game (see Theorem VII.7).

(2) For the restless Markovian model (given in Definition I.3) with $M \leq K$ agents, each playing a single arm at each time step, when the agents receive rewards according to the collision model (given in Definition I.5) or the random sharing model (given in Definition I.6), if binary feedback (given in Model I.17) is available to the agents, then DRCA (given in Figure 6.1) has weak regret (given in Definitions I.12 and I.13) of $O(M^3 K \log^2 T)$, uniformly in T (see Theorem VI.5).

(3) For the IID model (given in Definition I.1) with $M \leq K$ agents, each playing a single arm at each time step, when agents receive rewards according to the general symmetric interaction model (given in Definition I.7), if partial feedback (given in Model I.18) is available to the agents, then RLOF (given in Figure 7.2) has weak regret (given in Definition I.14) of $O(T^{\frac{2M-1+2\gamma}{2M}})$ for some $\gamma > 0$ which can be chosen arbitrarily small with a tradeoff in finite time regret, uniformly in T (see Theorem VII.12).

(4) For the IID model (given in Definition I.1) and the restless Markovian model

(given in Definition I.3) with $M \leq K$ agents, each playing a single arm at each time step, when the agents receive rewards according to the general symmetric interaction model (given in Definition I.7), if partial feedback with synchronization (given in Model I.19) is available to the agents, then DLOE (given in Figure 7.3) has weak regret (given in Definition I.14) of $O(M^3 K \log T)$, uniformly in T (see Theorems VII.19 and VII.24).

(5) For the IID model (given in Definition I.1) and the restless Markovian model (given in Definition I.3) with $M \leq K$ agents, each playing a single arm at each time step, when the agents receive rewards according to the agent-specific interaction model (given in Definition I.8), if costly communication (given in Model I.20) is available to the agents, then DLC (given in Figure 7.4) has weak regret (given in Definition I.15) of $O(M^3 K \log T)$, uniformly in T (see Theorems VII.28 and VII.30).

1.5 Organization of the Thesis

The organization of this thesis is as follows. In Chapter II we study the single-agent rested Markovian bandit problem with single and multiple plays, and in Chapter III we study the single-agent restless Markovian bandit problem with single and multiple plays. For both of these chapters, we consider algorithms in the weak regret setting. We study the single-agent uncontrolled restless Markovian bandit problem with a single play in Chapter IV in the strong regret setting. A special case of the uncontrolled restless Markovian bandit problem, called the feedback bandit problem, with a single agent and single play is considered in Chapter V in the approximate optimality setting. In Chapter VI we consider the multi-agent restless bandit problem with a collision model in the weak regret setting. Then, we study multi-agent bandit problems in a general resource sharing setting in Chapter VII. An online

contract selection problem is studied in Chapter VIII, and learning algorithms with subliner regret are proposed. Finally in Chapter IX, we give concluding remarks and emphasize possible future research directions.

CHAPTER II

Single-agent Rested Bandits

As mentioned in the introduction, in a single-agent rested bandit there is an agent which sequentially selects M of K arms to maximize its total reward, while the arms evolve in a Markovian fashion when selected by the agent, and remains frozen otherwise. In this chapter we show that there exists a learning algorithm whose weak regret is uniformly logarithmic in time. Previously, *Anantharam et al.* (1987b) showed that for this problem asymptotic logarithmic weak regret is achievable when the transition probabilities are parameterized. Our results in this chapter extends the previous results by providing finite time regret bounds for a more general non-parametric transition probability model.

The organization of this chapter is as follows. Problem definition and notations are given in Section 2.1. Rested bandit problem with a single play is investigated, and an algorithm with logarithmic regret is proposed in Section 2.2. Extension to multiple plays is done in Section 2.3. A gambling application and numerical results are given in Section 2.4. Finally, a discussion is given in Section 2.5.

2.1 Problem Formulation and Preliminaries

Consider K mutually independent rested Markovian arms described in Definition I.2. For simplicity of presentation, without loss of generality, the arms are ordered

according to their mean reward, $\mu^1 \geq \mu^2 \geq \ldots \geq \mu^K$ (ordering not known by the agent). Let $(P^k)'$ denote the *adjoint* of P^k on $l_2(\pi)$ where

$$(p^k)'_{xy} = (\pi^k_y p^k_{yx})/\pi^k_x, \ \forall x, y \in S^i,$$

and $\dot{P}^k = (P^k)'P$ denote the *multiplicative symmetrization* of P^k. We will assume that the P^k's are such that $\dot{P}^k$'s are irreducible. To give a sense of how weak or strong this assumption is, we first note that this is a weaker condition than assuming the arms are reversible Markov chains. This technical assumption is required in a large deviation bound (Lemma A.1) that we will frequently use in the proofs.

There is an agent which plays M of the K arms at each time step. Although not required, for simplicity of presentation in this chapter, we will make the additional assumption that the mean reward of arm M is strictly greater than the mean reward of arm $M+1$, i.e., we have $\mu^1 \geq \mu^2 \geq \cdots \geq \mu^M > \mu^{M+1} \geq \cdots \geq \mu^K$. The results will still hold for $\mu^M \geq \mu^{M+1}$.

We will refer to the set of arms $\{1, 2, \cdots, M\}$ as the M-best arms and say that each arm in this set is *optimal*, while referring to the set $\{M+1, M+2, \cdots, K\}$ as the M-worst arms, and say that each arm in this set is *suboptimal*.

For a policy α its weak regret $R^\alpha(T)$ is the difference between the expected total reward that can be obtained by only playing the M-best arms and the expected total reward obtained by policy α up to time T, which is given in Definition I.9. The objective is to examine how the regret $R^\alpha(T)$ behaves as a function of T for a given policy α, and to construct a policy whose regret is order-optimal, through appropriate bounding. As we will show and as is commonly done, the key to bounding $R^\alpha(T)$ is to bound the expected number of plays of any suboptimal arm. Let $N^{\alpha,k}(t)$ be the number of times arm k is played by policy α at the end of time t, and $\bar{r}^k(N^{\alpha,k}(t))$ be the sample mean of the rewards observed from the first $N^{\alpha,k}(t)$ plays of arm k. When

the policy used is clear from the context we will suppress the superscript α from the above expressions. Although throughout our discussion we will consider a horizon of T time slots, our regret bounds hold uniformly for all T. Time horizon T is not an input to our algorithms, so agent does not need to know T.

The notation given in Table 2.1 is frequently used throughout this and the following chapters.

$\beta := \sum_{t=1}^{\infty} 1/t^2$
$\pi^k_{\min} = \min_{x \in S^k} \pi^k_x$
$\pi_{\min} := \min_{k \in \mathcal{K}} \pi^k_{\min}$
$r_{\max} := \max_{x \in S^k, k \in \mathcal{K}} r^k_x$
$S_{\max} := \max_{k \in \mathcal{K}} \lvert S^k \rvert$
$\hat{\pi}_{\max} := \max_{x \in S^k, k \in \mathcal{K}} \left\{ \pi^k_x, 1 - \pi^k_x \right\}$
ϵ^k: the eigenvalue gap (the difference between 1 and the second largest eigenvalue) of the multiplicative symmetrization $\dot{P}^k$ of P^k
$\epsilon_{\min} := \min_{k \in \mathcal{K}} \epsilon^k$

Table 2.1: frequently used expressions

In the next two sections we present algorithms with logarithmic weak regret for the problem stated in this section for a single play $M = 1$, and multiple plays $M > 1$, respectively.

2.2 Rested Bandit Problem with a Single Play

In this section we show that there exists an algorithm that achieves logarithmic weak regret uniformly over time for the rested bandit problem with a single play. The algorithm we consider is called *the upper confidence bound* (UCB), which is a slight modification of UCB1 from *Auer et al.* (2002) with an unspecified exploration constant L instead of fixing it at 2.

As shown in Figure 2.1, UCB selects the arm with the highest index at each time step and updates the indices according to the rewards observed. Let $y^k(t)$ be the reward from the tth play of arm k. The index given on line 4 of Figure 2.1 depends

on the sample mean reward and an exploration term which reflects the relative uncertainty about the sample mean of an arm. We call L in the exploration term *the exploration constant*. It can be shown that the exploration term grows logarithmically when the arm is not played in order to guarantee that sufficient samples are taken from each arm to approximate the mean reward.

The Upper Confidence Bound (UCB) Algorithm

1: Initialize: Play each arm once in the first K slots
2: **while** $t \geq K$ **do**
3: $\quad \bar{r}^k(t) = \frac{y^k(1)+y^k(2)+\ldots+y^k\left(N^k(t)\right)}{N^k(t)},\ \forall k$
4: $\quad$ calculate index: $g^k_{t,N^k(t)} = \bar{r}^k(N^k(t)) + \sqrt{\frac{L \ln t}{N^k(t)}},\ \forall k$
5: $\quad$ randomly select $i^* \in \arg\max_{k \in \mathcal{K}} g^k_{t,N^k(t)}$
6: $\quad t := t + 1$
7: $\quad$ play the arm $\alpha(t) = i^*$, receive reward $r^{i^*}(t)$, $N^{i^*}(t) = N^{i^*}(t-1) + 1$.
8: **end while**

Figure 2.1: pseudocode for the UCB algorithm

To upper bound the regret of the above algorithm logarithmically, we proceed as follows. We begin by relating the regret to the expected number of plays of the arms and then show that each suboptimal arm is played at most logarithmically in expectation. These steps are illustrated in the following lemmas.

Lemma II.1. *Assume that all arms are finite-state, irreducible, aperiodic, rested Markov chains. Then using UCB we have:*

$$\left| R(T) - \left(T\mu^1 - \sum_{k=1}^{K} \mu^k E[N^k(T)] \right) \right| \leq C_{\mathbf{S},\mathbf{P},\mathbf{r}}, \tag{2.1}$$

where $C_{\mathbf{S},\mathbf{P},\mathbf{r}}$ is a constant that depends on the state spaces, rewards, and transition probabilities but not on time.

Proof. We have,

$$\left| R(T) - \left(T\mu^1 - \sum_{k=1}^{K} \mu^k E[N^k(T)] \right) \right|$$

$$
\begin{aligned}
&= \left| E\left[\sum_{k=1}^{K}\sum_{x\in S^k} r_x^k \sum_{t=1}^{N^k(T)} I(y^k(t)=x)\right] - \sum_{k=1}^{K}\sum_{x\in S^k} r_x^k \pi_x^k E[N^k(T)]\right| \\
&= \left|\sum_{k=1}^{K}\sum_{x\in S^k} r_x^k (E[U^k(x,N^k(T))] - \pi_x^i E[N^k(T)])\right| \\
&\leq \sum_{k=1}^{K}\sum_{x\in S^k} r_x^k C_{P^k} = C_{\mathbf{S},\mathbf{P},\mathbf{r}} \ , \qquad (2.2)
\end{aligned}
$$

where

$$
U^k(x,N^k(T)) = \sum_{t=1}^{N^k(T)} I(y^k(t)=x),
$$

and (2.2) follows from Lemma A.3 using the fact that $N^k(T)$ is a stopping time with respect to the σ-field generated by the arms played up to time T. □

Lemma II.2. *Under UCB with* $L \geq 112 S_{\max}^2 r_{\max}^2 \hat{\pi}_{\max}^2/\epsilon_{\min}$*, for any suboptimal arm* k*, we have*

$$
E[N^k(T)] \leq 1 + \frac{4L\ln T}{(\mu^1-\mu^k)^2} + \frac{(|S^k|+|S^1|)\beta}{\pi_{\min}}
$$

Proof. see Appendix B. □

Theorem II.3. *With constant* $L \geq 112 S_{\max}^2 r_{\max}^2 \hat{\pi}_{\max}^2/\epsilon_{\min}$ *the regret of UCB is upper bounded by*

$$
R(T) \leq 4L\ln T \sum_{k>1} \frac{1}{(\mu^1-\mu^k)} + \sum_{k>1}(\mu^1-\mu^k)\,(1+C_{k,1}) + C_{\mathbf{S},\mathbf{P},\mathbf{r}},
$$

where $C_{k,1} = \frac{(|S^k|+|S^1|)\beta}{\pi_{\min}}$.

Proof.

$$
R(T) \ \leq \ T\mu^1 - \sum_{k=1}^{K} \mu^k E[N^k(T)] + C_{\mathbf{S},\mathbf{P},\mathbf{r}} \qquad (2.3)
$$

$$
\begin{aligned}
&\leq \sum_{k>1}(\mu^1-\mu^k)E[N^k(T)]+C_{\mathbf{S},\mathbf{P},\mathbf{r}} \\
&\leq \sum_{k>1}(\mu^1-\mu^k)\left(1+\frac{4L\ln T}{(\mu^1-\mu^k)^2}+\frac{(|S^k|+|S^1|)\beta}{\pi_{\min}}\right)+C_{\mathbf{S},\mathbf{P},\mathbf{r}} \qquad (2.4)\\
&= 4L\ln T\sum_{k>1}\frac{1}{(\mu^1-\mu^k)}+\sum_{k>1}(\mu^1-\mu^k)\left(1+C_{k,1}\right)+C_{\mathbf{S},\mathbf{P},\mathbf{r}},
\end{aligned}
$$

where (2.3) follows from Lemma II.1 and (2.4) follows from Lemma II.2. □

The above theorem says that provided that L satisfies the stated sufficient condition, UCB results in logarithmic weak regret. This sufficient condition does require certain knowledge on the underlying Markov chains. This requirement may be removed if the value of L is adapted over time. More is discussed in Section 2.5.

2.3 Rested Bandit Problem with Multiple Plays

In this section we consider the case where the agent selects $M > 1$ arms at each time step. The multiple-play extension to UCB1, referred to as UCB-M given in Figure 2.2 below, is straightforward: initially each arm is played M times in the first K slots (M arms in each slot, in arbitrary order); subsequently at each time slot the algorithm plays M of the K arms with the highest current indices.

The Upper Confidence Bound - Multiple Plays (UCB-M):

1: Initialize: Play each arm M times in the first K slots
2: **while** $t \geq K$ **do**
3: $\quad \bar{r}^k(N^k(t)) = \frac{y^k(1)+y^k(2)+\ldots+y^k(N^k(t))}{N^k(t)},\ \forall k$
4: $\quad$ calculate indices: $g^k_{t,N^k(t)} = \bar{r}^k(N^k(t)) + \sqrt{\frac{L\ln t}{N^k(t)}},\ \forall k$
5: $\quad$ Let M^* be the vector of M arms with the highest indices
6: $\quad t := t+1$
7: $\quad$ play arms $\boldsymbol{\alpha(t)} = M^*$, receive the reward $r^k(t)$ for $k \in M^*$, $N^k(t) = N^k(t-1)+1$ for $k \in M^*$.
8: **end while**

Figure 2.2: pseudocode for the UCB-M algorithm

Similar to the lemmas in the previous section, we first relate the regret with

expected number of plays of the arms, then bound the expected number of plays of the suboptimal arms.

Lemma II.4. *For an agent using UCB-M, we have*

$$\left| R(T) - \left(T \sum_{l=1}^{M} \mu^l - \sum_{k=1}^{K} \mu^k E[N^k(T)] \right) \right| \leq C_{\mathbf{S},\mathbf{P},\mathbf{r}},$$

where $C_{\mathbf{S},\mathbf{P},\mathbf{r}}$ is a constant that depends on the state spaces, rewards, and transition probabilities but not on time.

Proof. The proof is similar to the proof of Lemma II.1. □

Lemma II.5. *For an agent using UCB-M with $L \geq 112 S_{\max}^2 r_{\max}^2 \hat{\pi}_{\max}^2 / \epsilon_{\min}$, for any suboptimal arm k, we have*

$$E[N^k(T)] \leq M + \frac{4L \ln T}{(\mu^M - \mu^k)^2} + \sum_{j=1}^{M} \frac{(|S^k| + |S^j|)\beta}{\pi_{\min}}.$$

Proof. The proof is similar to the proof of Lemma II.2. □

Theorem II.6. *For an agent using UCB-M with $L \geq 112 S_{\max}^2 r_{\max}^2 \hat{\pi}_{\max}^2 / \epsilon_{\min}$, we have*

$$R(T) \leq 4L \ln T \sum_{k>M} \frac{(\mu^1 - \mu^k)}{(\mu^M - \mu^k)^2} + \sum_{k>M} (\mu^1 - \mu^k) \left(M + \sum_{j=1}^{M} C_{k,j} \right) + C_{\mathbf{S},\mathbf{P},\mathbf{r}},$$

where $C_{k,j} = \frac{(|S^k| + |S^j|)\beta}{\pi_{\min}}$.

Proof.

$$\begin{aligned} T \sum_{l=1}^{M} \mu^l - \sum_{k=1}^{K} \mu^k E[N^k(T)] &= \sum_{k=1}^{M} \mu^k (T - E[N^k(T)]) - \sum_{k>M} \mu^k E[N^k(T)] \\ &\leq \sum_{k=1}^{M} \mu^1 (T - E[N^k(T)]) - \sum_{k>M} \mu^k E[N^k(T)] \\ &= \sum_{k>M} (\mu^1 - \mu^k) E[N^k(T)] \end{aligned}$$

Thus,

$$R(T) \leq T\sum_{j=1}^{M}\mu^j - \sum_{k=1}^{K}\mu^k E[N^k(T)] + C_{\mathbf{S},\mathbf{P},\mathbf{r}} \tag{2.5}$$

$$\leq \sum_{k>M}(\mu^1-\mu^k)E[N^k(T)] + C_{\mathbf{S},\mathbf{P},\mathbf{r}}$$

$$\leq \sum_{k>M}(\mu^1-\mu^k)\left(M + \frac{4L\ln T}{(\mu^M-\mu^k)^2} + \sum_{j=1}^{M}\frac{(|S^k|+|S^j|)\beta}{\pi_{\min}}\right) + C_{\mathbf{S},\mathbf{P},\mathbf{r}} \tag{2.6}$$

$$= 4L\ln T\sum_{k>M}\frac{(\mu^1-\mu^k)}{(\mu^M-\mu^k)^2} + \sum_{k>M}(\mu^1-\mu^k)\left(M+\sum_{j=1}^{M}C_{k,j}\right) + C_{\mathbf{S},\mathbf{P},\mathbf{r}},$$

where (2.5) follows from Lemma II.4 and (2.6) follows from Lemma II.5. □

2.4 Numerical Results

In this section we study the following gambling problem as an application of the rested bandit problem. Consider a player (agent) who plays one of 5 gambling machines (arms) at each time ($\mathcal{K} = \{1, 2, \ldots, 5\}$). The goal of the player is to maximize its returns from gamble. Each machine can be in one of two states g or b, and is modeled as an irreducible and aperiodic Markov chain, which changes its state only when it is played. State g yields reward 1, while state b yields reward 0.001, which is consistent with the assumption of non-negative rewards. We assume that the state transition probabilities of each machine is parameterized by $\theta \in (0, 10)$ such that

$$p_{gb}^k(\theta) = 1 - (\frac{\theta}{10})^2, \tag{2.7}$$

$$p_{bg}^k(\theta) = (\frac{\theta}{10})^3, \tag{2.8}$$

for all $k \in \mathcal{K}$. The parameter set, which is unknown by the player, is $\boldsymbol{\theta} = [7, 5, 3, 1, 0.5]$ where kth element corresponds to the parameter of machine k. The stationary dis-

tribution of machine k is

$$\boldsymbol{\pi}^k = [\pi_b^k, \pi_g^k] = \left[\frac{p_{gb}^k}{p_{gb}^k + p_{bg}^k}, \frac{p_{bg}^k}{p_{gb}^k + p_{bg}^k} \right] .$$

The transition probabilities, stationary distributions and expected rewards of the machines under the parameter set $\boldsymbol{\theta}$ is given in Table 2.2. The machines are ordered in terms of their expected rewards such that $\mu^1 > \mu^2 > \ldots > \mu^5$.

We compare the performance of UCB with the index policy given in *Anantharam et al.* (1987b). Irreducible multiplicative symmetrization assumption holds since $p_{gb}^k > 0, p_{bg}^k > 0$, for $\theta_k \in (0, 10), k \in \mathcal{K}$. Any policy α for which $R^\alpha(T) = o(T^\gamma)$ for every $\gamma > 0$ is called a *uniformly good* policy in *Anantharam et al.* (1987b). It was shown that for any *uniformly good* policy α,

$$\liminf_{T\to\infty} \frac{R^\alpha(T)}{\ln T} \geq \sum_{k:\mu^k<\mu^1} \frac{\mu^1 - \mu^k}{I(k,1)}, \tag{2.9}$$

where

$$I(k,1) := \sum_{x\in S^k} \pi_x^k \sum_{y\in S^k} p_{xy}^k(\theta^k) \ln \frac{p_{xy}^k(\theta^k)}{p_{xy}^k(\theta^1)} .$$

Furthermore, they showed that the index policy α^* in *Anantharam et al.* (1987b) satisfies

$$\limsup_{T\to\infty} \frac{R^{\alpha^*}(T)}{\ln T} \leq \sum_{k:\mu^k<\mu^1} \frac{\mu^1 - \mu^k}{I(k,1)} ,$$

when some regularity conditions are satisfied, and the transition probabilities of the arms are known functions of a single parameter θ, whose value is not known by the player. One can check that these assumptions hold for the transition probabilities given in (2.7) and (2.8) since $\mu^k(\theta)$ is increasing in θ, and $p_{bg}^k(\theta)$ and $p_{gb}^k(\theta)$ are log-

concave in θ.

In the two state model, the eigenvalue gap of the multiplicative symmetrization of the transition probability matrix of arm k is given by

$$\epsilon^k = p^k_{gb} + p^k_{bg} \ .$$

When the parameter set is $\boldsymbol{\theta}$, $112 S^2_{\max} r^2_{\max}/\epsilon_{\min} = 525.07$.

Figure 2.3 compares the regret of the index policy of *Anantharam et al.* (1987b) (labeled as Anantharam policy in the figure) with UCB under different values of the exploration constant L. Note that the index policy of *Anantharam et al.* (1987b) assumes that the player knows the functions $p^k_{gb}(\theta)$ and $p^k_{bg}(\theta)$, while in UCB these functions are unknown to the player. The player using UCB with $L = 530 > 112 S^2_{\max} r^2_{\max}/\epsilon_{\min}$ satisfies the sufficient condition for the logarithmic regret bound in Theorem II.3. The logarithmic term in Theorem II.3 for this case is $46.11 L \ln T = 24438 \ln n$, while Anantharam's bound has the logarithmic term $4.406 \ln T$, which is significantly better in terms of the constant.

The first thing to note is the gap between the bound we derived for UCB and the bound of *Anantharam et al.* (1987b) given in (2.9). The second thing to note is that for $L = 0.05$, UCB has smaller regret than the index policy of *Anantharam et al.* (1987b), as well as the bound in (2.9), for the given time horizon. Note that *Anantharam et al.* (1987b) proved that the performance of any *uniformly good* policy cannot be better than the bound in (2.9) asymptotically. Since uniformly good policies have the minimum growth of regret among all policies, this bound also holds for UCB. This however is not a contradiction because this bound holds asymptotically; we indeed expect the regret of UCB with $L = 0.05$ to be very close to this bound in the limit. These results show that while the bound in *Anantharam et al.* (1987b) is better than the bound we proved for UCB in this paper, in reality the UCB policy can perform

Arm	p_{01}, p_{10}	π_1	μ
1	0.0270, 0.9100	0.0288	0.4027
2	0.1250, 0.7500	0.1429	0.1437
3	0.3430, 0.5100	0.4021	0.0298
4	0.0010, 0.9900	0.0010	0.0020
5	0.0001, 0.9975	0.0001	0.0011

Table 2.2: parameters of the arms for $\boldsymbol{\theta} = [7, 5, 3, 1, 0.5]$

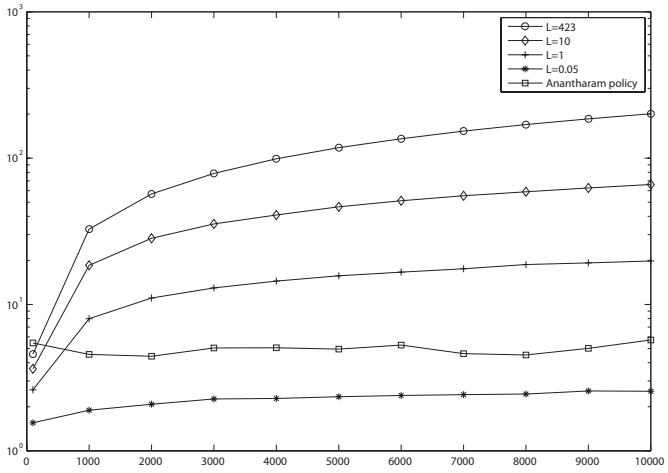

Figure 2.3: regrets of UCB and Anantharam's policy

very close to the tighter bound (uniformly, not just asymptotically).

2.5 Discussion

Although $L \geq 112 S_{\max}^2 r_{\max}^2 / \epsilon_{\min}$ is a sufficient condition for the upper bound in Theorem II.3 to hold, from the numerical results in Section 2.4, we conclude that it is not necessary. Results in Figure 2.3 imply that, for the rested bandit model of Section 2.4 the regret is monotonically decreasing in L. This is because the rate of exploration depends on the inflation term $\sqrt{L \ln t / N^k(t)}$ of the index of arm k. Large values of L imply more exploration, which in turn implies more plays of suboptimal arms. Note that the condition on L comes from the large deviation bound in Lemma A.1. This observation opens up the question that which large deviation bound should

we use to obtain tighter regret bounds.

It can be shown that instead of using Lezaud's large deviation bound given in Lemma A.1, if we use Gillman's large deviation bound given in Lemma A.2, logarithmic regret results in Theorems II.3 and II.6 still hold with different constants given that the exploration constant $L \geq 90 S_{\max}^2 r_{\max}^2 / \tilde{\epsilon}_{\min}$, where $\tilde{\epsilon}_{\min} = \min_{k \in \mathcal{K}} \tilde{\epsilon}^k$, and $\tilde{\epsilon}^k$ is the eigenvalue gap of the transition probability matrix of arm k. The Gillman bound requires Markov chains to reversible, instead of the irreducible multiplicative symmetrization assumption of Lezaud's bound. This trivially holds for irreducible two state Markov chains. In the two state model in Section 2.4, the eigenvalue gap of the transition probability matrix of arm k is given by

$$\tilde{\epsilon}^k = p_{gb}^k + p_{bg}^k \ ,$$

which is same as the eigenvalue gap of the multiplicative symmetrization of the transition probability matrix of arm k. For the setting in Section 2.4, $90 S_{\max}^2 r_{\max}^2 / \tilde{\epsilon}_{\min} = 422.04$, which is smaller than $L = 525.07$ found according to Lezaud's bound.

CHAPTER III

Single-agent Restless Bandits with Weak Regret

In this chapter we study the single-agent restless bandit problem in which the state of on arm may also change when it is not played by the agent. Specifically, we show that when the stochastic rewards from the arms are generated by unknown Markov chains, the agent can achieve uniformly logarithmic weak regret over time. Although in general the optimal policy computed based on the known transition probabilities of the arms is a dynamic policy which may play different arms according to the *information state* of the system, there are various reasons to approach the restless bandit problem in terms of the weak regret which compares the performance of the algorithm with the best static policy.

First of all, even in the optimization setting the restless bandit problem is $P-SPACE$ hard to approximate which is shown in *Papadimitriou and Tsitsiklis* (1999). Since we are taking an agent-centric approach to the bandit problems, we also consider computational power and memory requirements when designing online learning algorithms. We consider weak regret because the algorithms to achieve order-optimal (logarithmic) weak regret just requires storage linear in the number of arms, and runs a few simple arithmetic operations at each time step. In real-world applications, agents may be wireless sensor nodes with limited energy, computational power and memory, therefore they may not be able to run complex algorithms.

Secondly, the agent might have costs associated with switching arms frequently. For example, when the arms are frequency bands and the agent is a transmitter-receiver pair, then switching to different bands frequently may not be a viable option because of the associated energy cost. As another example, if the arms are products which yields stochastic payoffs changing with time, and the agent is an investor, the agent may not want to switch too many times because of the friction in the market due to transaction costs. The algorithms we propose in this chapter switches arms only logarithmically many times in expectation, thus advantageous for the agents with high switching costs.

Thirdly, weak regret is a commonly used performance measure in the bandit literature, especially for adversarial bandits, in which the rewards of the arms are generated by an adversary to minimize to payoff to the agent. The bounds derived for these problems are worst-case in the sense that they hold for all stochastic processes that generates the arm rewards not just the Markovian ones. For our weak regret results to hold, the state of an arm should change in a Markovian way when it is played by the agent. However, it might change adversarially when not played.

The organization of this chapter is as follows. Problem definition and notations are given in Section 3.1. Restless bandit problem with single play is investigated, and an algorithm with logarithmic weak regret is proposed in Section 3.2. Extension to multiple plays is done in Section 3.3. An opportunistic spectrum access application and numerical results are given in Section 3.4. Finally, discussion is given in Section 3.5.

3.1 Problem Formulation and Preliminaries

In this chapter we study the restless Markovian model. Consider K mutually independent restless Markovian arms described in Definition I.3. For simplicity of presentation, WLOG, the arms are ordered according to their mean reward, $\mu^1 \geq$

$\mu^2 \geq \ldots \geq \mu^K$ (ordering not known by the agent). Let $(P^k)'$ denote the *adjoint* of P^k on $l_2(\pi)$ where

$$(p^k)'_{xy} = (\pi^k_y p^k_{yx})/\pi^k_x, \ \forall x, y \in S^i,$$

and $\dot{P}^k = (P^k)'P$ denote the *multiplicative symmetrization* of P^k. We have the same assumption as in Chapter II that the P^k's are such that $\dot{P}^k$'s are irreducible. We note that one condition that guarantees the $\hat{P}^i$'s are irreducible is $p_{xx} > 0, \forall x \in S^i, \forall i$. This assumption thus holds naturally for two of our motivating applications, opportunistic spectrum access in fading channels and cognitive radio dynamic spectrum access, as it's possible for channel condition to remain the same over a single time step (especially if the unit is sufficiently small). It also holds for a very large class of Markov chains and applications in general. Consider for instance a queueing system scenario where an arm denotes a server and the Markov chain models its queue length, in which it is possible for the queue length to remain the same over one time unit.

There is an agent which plays M of the K arms at each time step. Similar to Chapter II, we will make the additional assumption that the mean reward of arm M is strictly greater than the mean reward of arm $M+1$, i.e., we have $\mu^1 \geq \mu^2 \geq \cdots \geq \mu^M > \mu^{M+1} \geq \cdots \geq \mu^K$. Contrary to Chapter II, strict inequality between μ^M and μ^{M+1} is needed because otherwise there can be a large number of arm switchings between the Mth and the $(M+1)$th arms (possibly more than logarithmic). Strict inequality prevents this from happening. We note that this assumption is not in general restrictive; in our motivating applications mentioned above, distinct channel conditions typically mean different data rates. Possible relaxation of this condition is given in Section 3.5.

For a policy α its weak regret $R^\alpha(T)$ is the difference between the expected total reward that can be obtained by only playing the M-best arms and the expected total

reward obtained by policy α up to time T, which is given in Definition I.9. The objective is to examine how the regret $R^{\alpha}(T)$ behaves as a function of T for a given policy α and to construct a policy whose regret is order-optimal, through appropriate bounding. Similar to the rested Markovian model, the key to bounding $R^{\alpha}(T)$ is to bound the expected number of plays of any suboptimal arm. Let $N^{\alpha,k}(t)$ be the number of times arm k is played by policy α at the end of time t, and $\bar{r}^{k}(N^{\alpha,k}(t))$ be the sample mean of the rewards observed from the first $N^{\alpha,k}(t)$ plays of arm k. When the policy used is clear from the context we will suppress the superscript α from the above expressions. Although, throughout our discussion we will consider a horizon of T time slots, our regret bounds hold uniformly for all T. Time horizon T is not an input to our algorithms, so agent does not need to know T.

In the next two sections we present algorithms with logarithmic weak regret for the problem stated in this section for a single play $M = 1$, and multiple plays $M > 1$, respectively. While the multiple-play case is more general, the analysis in the single-play case is more intuitive to illustrate with less cumbersome notations.

3.2 Restless Bandit Problem with a Single Play

In this section we study the restless bandit problem, where an agent chooses a single arm at each time step. We construct an algorithm called the *regenerative cycle algorithm* (RCA), and prove that this algorithm guarantees logarithmic regret uniformly over time under the same mild assumptions on the state transition probabilities as in the rested Markovian model. Below we first present the key conceptual idea behind RCA, followed by a more detailed pseudocode. We then prove the logarithmic regret result.

As the name suggests, RCA operates in regenerative cycles. In essence RCA uses the observations from sample paths within regenerative cycles to estimate the sample mean reward of an arm in the form of an index similar to that used in UCB

while discarding the rest of the observations (only for the computation of the index; they contribute to the total reward). Note that the rewards from the discarded observations are collected but are not used to make decisions. The reason behind such a construction has to do with the restless nature of the arms. Since each arm continues to evolve regardless of the agent's action, the probability distribution of the reward the agent gets by playing an arm is a function of the amount of time that has elapsed since the last time the agent played the same arm. Since the arms are not played continuously, the sequence of observations from an arm which is not played consecutively does not correspond to a discrete time homogeneous Markov chain. While this certainly does not affect the agent's ability to collect rewards, it becomes hard to analyze the estimated quality (the index) of an arm calculated based on rewards collected this way.

However, if instead of the actual sample path of observations from an arm, we limit ourselves to a sample path constructed (or rather stitched together) using only the observations from regenerative cycles, then this sample path essentially has the same statistics as the original Markov chain due to the renewal property and one can now use the sample mean of the rewards from the regenerative sample paths to approximate the mean reward under stationary distribution.

Under RCA the agent maintains a block structure; a block consists of a certain number of slots. Within a block the agent plays the same arm continuously till a certain pre-specified state (say γ^k) is observed. Upon this observation the arm enters a regenerative cycle and the agent continues to play the same arm till state γ^k is observed for the second time, which denotes the end of the block. For the purpose of index computation and subsequent analysis, each block is further broken into three sub-blocks (SBs). SB1 consists of all time slots from the beginning of the block to right before the first visit to γ^k; SB2 includes all time slots from the first visit to γ^k up to but excluding the second visit to state γ^i; SB3 consists of a single time slot with

the second visit to γ^k. Figure 3.1 shows an example sample path of the operation of RCA.

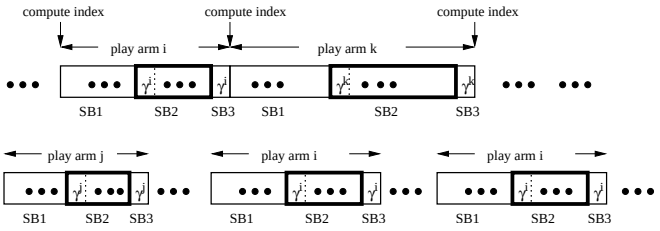

Figure 3.1: example realization of RCA

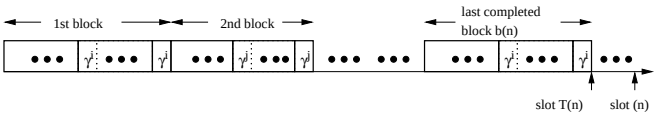

Figure 3.2: the block structure of RCA

The key to the RCA algorithm is for each arm to single out only observations within SB2's in each block and virtually assemble them. In addition to the notation given in Table 2.1, we will also use the notation given in Table 3.2. Let $y^k(t)$ denote the reward from the tth play of arm k in SB2's.

The block structure along with some of the definitions above are presented in Figure 3.2. RCA computes and updates the value of an *index* g^k for each arm k at the end of block b based on the total reward obtained from arm k during all SB2's as follows:

$$g^k_{t_2(b),N_2^k(t_2(b))} = \bar{r}^k(N_2^k(t_2(b))) + \sqrt{\frac{L \ln t_2(b)}{N_2^k(t_2(b))}}, \tag{3.1}$$

where L is a constant, and

$$\bar{r}^k(N_2^k(t_2(b))) = \frac{y^k(1) + y^k(2) + ... + y^k(N_2^k(t_2(b)))}{N_2^k(t_2(b))}$$

$\Omega^k_{x,y}$ is the mean hitting time of state y given the initial state x for arm k and P^k
$\Omega^k_{\max} = \max_{x,y \in S^k}$.
γ^k: the state that determines the regenerative cycles for arm k.
$\tilde{\alpha}(b)$: the arm played in the bth block.
$b(T)$: the number of completed blocks up to time T.
$N(T)$: the time at the end of the last completed block (see Figure 3.2).
$B^k(b)$: the total number of blocks within the first completed b blocks in which arm k is played.
$X^k_1(b)$: the vector of observed states from SB1 of the bth block in which arm k is played; this vector is empty if the first observed state is γ^i.
$X^k_2(b)$: the vector of observed states from SB2 of the bth block in which arm k is played.
$X^k(b)$: the vector of observed states from the bth block in which arm k is played. Thus we have $X^k(b) = [X^k_1(b), X^k_2(b), \gamma^k]$.
$t(b)$: time at the end of block b.
$t_2(b)$: the number of time slots that lie within an SB2 of any completed block up to and including block b.
$N^k_2(t)$: the number of time slots arm k is played during SB2's when the number of time steps that lie within an SB2 is t.

Table 3.1: frequently used expressions

denotes the sample mean of the reward collected during SB2. It is also worth noting that under RCA rewards are also collected during SB1's and SB3's. However, the computation of the indices only relies on SB2. The pseudocode of RCA is given in Figure 3.3.

Proving the existence of a logarithmic upper bound on the regret for restless arms is a non-trivial task since the blocks may be arbitrarily long and the frequency of arm selection depends on the length of the blocks. In the analysis that follows, we first show that the expected number of blocks in which a suboptimal arm is played is at most logarithmic. By the regenerative property of the arms, all the observations from SB2's of an arm can be combined together and viewed as a sequence of continuous observations from a rested arm. Therefore we can use a large deviation result to bound the expected number of times the index of a suboptimal arm exceeds the index of an optimal arm. Using this result, we show that the expected number of blocks in which a suboptimal arm is played is at most logarithmic in time. We then relate

Regenerative Cycle Algorithm (RCA):

Initialize: $b = 1, t = 0, t_2 = 0, N_2^k = 0, r^k = 0, \forall k = 1, \cdots, K$
for $b \leq K$ **do**
  play arm b; set γ^b to be the first state observed
  $t := t + 1$; $t_2 := t_2 + 1$; $N_2^b := N_2^b + 1$; $r^b := r^b + r^b_{\gamma^b}$
  play arm b; denote observed state as x
  while $x \neq \gamma^b$ **do**
    $t := t + 1$; $t_2 := t_2 + 1$; $N_2^b := N_2^b + 1$; $r^b := r^b + r^b_x$
    play arm b; denote observed state as x
  end while
  $b := b + 1$; $t := t + 1$
end for
for $k = 1$ to K **do**
  compute index $g^k := \frac{r^k}{N_2^k} + \sqrt{\frac{L \ln t_2}{N_2^k}}$
  $k++$
end for
$k := \arg\max_j g^j$
while (1) **do**
  play arm k; denote observed state as x
  while $x \neq \gamma^k$ **do**
    $t := t + 1$
    play arm k; denote observed state as x
  end while
  $t := t + 1$; $t_2 := t_2 + 1$; $N_2^k := N_2^k + 1$; $r^k := r^k + r^k_x$
  play arm k; denote observed state as x
  while $x \neq \gamma^k$ **do**
    $t := t + 1$; $t_2 := t_2 + 1$; $N_2^k := N_2^k + 1$; $r^k := r^k + r^k_x$
    play arm k; denote observed state as x
  end while
  $b := b + 1$; $t := t + 1$
  for $k = 1$ to K **do**
    compute index $g^k := \frac{r^k}{N_2^k} + \sqrt{\frac{L \ln t_2}{N_2^k}}$
    $k++$
  end for
  $k := \arg\max_j g^j$
end while

Figure 3.3: pseudocode of RCA

the expected number of blocks in which a suboptimal arm is played to the expected number of time slots in which a suboptimal arm is played using the positive recurrence property of the arms. Finally, we show that the regret due to arm switching is at

most logarithmic, and the regret from the last, incomplete block is finite due to the positive recurrence property of the arms.

Below, we first bound the expected number of plays from a suboptimal arm.

Lemma III.1. *For an agent using RCA with constant* $L \geq 112 S_{\max}^2 r_{\max}^2 \hat{\pi}_{\max}^2 / \epsilon_{\min}$, *we have*

$$E[N^k(N(T))] \leq D_k \left(\frac{4L \ln T}{(\mu^1 - \mu^k)^2} + C_{k,1} \right),$$

where,

$$C_{k,1} = \left(1 + \frac{(|S^k| + |S^1|)\beta}{\pi_{\min}} \right), \ \beta = \sum_{t=1}^{\infty} t^{-2},$$
$$D_k = \left(\frac{1}{\pi_{\min}^k} + \Omega_{\max}^k + 1 \right).$$

Proof. See Appendix C. □

We now state the main result of this section.

Theorem III.2. *For an agent using RCA with constant* $L \geq 112 S_{\max}^2 r_{\max}^2 \hat{\pi}_{\max}^2 / \epsilon_{\min}$, *the regret is upper bounded by*

$$R(T) < 4L \ln T \sum_{k>1} \frac{1}{\mu^1 - \mu^k} \left(D_k + \frac{E_k}{\mu^1 - \mu^k} \right)$$
$$+ \sum_{k>1} C_{k,1} \left((\mu^1 - \mu^k) D_k + E_k \right) + F$$

where

$$C_{k,1} = \left(1 + \frac{(|S^k| + |S^1|)\beta}{\pi_{\min}} \right), \ \beta = \sum_{t=1}^{\infty} t^{-2}$$
$$D_k = \left(\frac{1}{\pi_{\min}^k} + \Omega_{\max}^k + 1 \right),$$
$$E_k = \mu^k (1 + \Omega_{\max}^k) + \mu^1 \Omega_{\max}^1,$$

$$F = \mu^1 \left(\frac{1}{\pi_{\min}} + \max_{k \in \{1,\ldots,K\}} \Omega^k_{\max} + 1 \right).$$

Proof. See Appendix D. □

Theorem III.2 suggests that given minimal information about the arms such as an upper bound for $S^2_{\max} r^2_{\max} \hat{\pi}^2_{\max} / \epsilon_{\min}$ the agent can guarantee logarithmic regret by choosing an L in RCA that satisfies the stated condition. As in the rested case, this requirement on L can be completely removed if the value of L is adapted over time; more is discussed in Section 3.5.

We conjecture that the order optimality of RCA holds when it is used with any index policy that is order optimal for the rested bandit problem. Because of the use of regenerative cycles in RCA, the observations used to calculate the indices can be in effect treated as coming from rested arms. Thus, an approach similar to the one used in the proof of Theorem III.2 can be used to prove order optimality of combinations of RCA and other index policies. We comment more on this in Section 3.5.

3.3 Restless Bandit Problem with Multiple Plays

In this section we extend the results of the previous section to the case of multiple plays. The multiple-play extension to the regenerative cycle algorithm will be referred to as the RCA-M. As in the rested case, even though our basic model is one of single-agent with multiple plays, our description is in the equivalent form of multiple coordinated agents each with a single play.

As in RCA, RCA-M maintains the same block structure, where the agent plays the same arm till it completes a regenerative cycle. Since M arms are played (by M agents) simultaneously in each slot, different blocks overlap in time. Multiple blocks may or may not start or end at the same time. In our analysis below blocks will be ordered; they are ordered according to their start time. If multiple blocks start at the

same time then the ordering among them is randomly chosen. Figure 3.4 shows an example sample path of the operation of RCA-M. The block structure of two plays and the ordering of the blocks are shown.

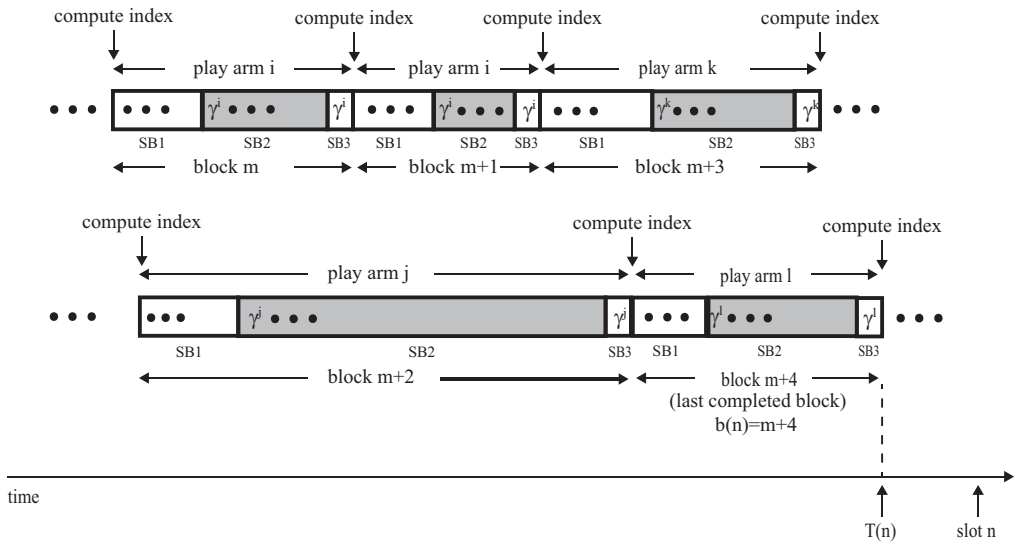

Figure 3.4: example realization of RCA-M with $M = 2$ for a period of n slots

The pseudocode of RCA-M is given in Figure 3.5. The analysis is similar to that in Section 3.2, with careful accounting of the expected number of blocks in which a suboptimal arm is played. The details are given in Theorem III.3.

Theorem III.3. *For an agent using RCA-M with constant* $L \geq 112 S_{\max}^2 r_{\max}^2 \hat{\pi}_{\max}^2 / \epsilon_{\min}$ *the regret is upper bounded by*

$$R(T) < 4L \ln T \sum_{k>M} \frac{1}{(\mu^M - \mu^k)^2} \left((\mu^1 - \mu^k) D_k + E_k \right)$$
$$+ \sum_{k>M} \left((\mu^1 - \mu^k) D_k + E_k \right) \left(1 + M \sum_{j=1}^{M} C_{k,j} \right) + F,$$

The Regenerative Cycle Algorithm - Multiple Plays (RCA-M):

Initialize: $b = 1, t = 0, t_2 = 0, N_2^k = 0, r^k = 0, I_{SB2}^k = 0, I_{IN}^k = 1, \forall k \in \mathcal{K}$, $A = \emptyset$
//I_{IN}^k indicates whether arm k has been played at least once
//I_{SB2}^k indicates whether arm k is in an SB2 sub-block
while (1) **do**
  for $k = 1$ to K **do**
    if $I_{IN}^k = 1$ and $|A| < M$ **then**
      $A \leftarrow A \cup \{k\}$ //arms never played is given priority to ensure all arms are sampled initially
    end if
  end for
  if $|A| < M$ **then**
    Add to A the set
    $\left\{k : g^k \text{ is one of the } M - |A| \text{ largest among } \{g^j, j \in \mathcal{K} - A\}\right\}$
    //for arms that have been played at least once, those with the largest indices are selected
  end if
  for $k \in A$ **do**
    play arm k; denote state observed by x^k
    if $I_{IN}^k = 1$ **then**
      $\gamma^k = x^k$, $N_2^k := N_2^k + 1$, $r^k := r^k + r_{x^k}^k$, $I_{IN}^k = 0$, $I_{SB2}^k = 1$
      //the first observed state becomes the regenerative state; the arm enters SB2
    else if $x^k \neq \gamma^k$ and $I_{SB2}^k = 1$ **then**
      $N_2^k := N_2^k + 1$, $r^k := r^k + r_{x^k}^k$
    else if $x^k = \gamma^k$ and $I_{SB2}^k = 0$ **then**
      $N_2^k := N_2^k + 1$, $r^k := r^k + r_{x^k}^k$, $I_{SB2}^k = 1$
    else if $x^k = \gamma^k$ and $I_{SB2}^k = 1$ **then**
      $r^k := r^k + r_{x^k}^k$, $I_{SB2}^k = 0$, $A \leftarrow A - \{k\}$
    end if
  end for
  $t := t + 1$, $t_2 := t_2 + \min\left\{1, \sum_{k \in A} I_{SB2}^i\right\}$ //t_2 is only accumulated if at least one arm is in SB2
  for $k = 1$ to K **do**
    $g^k = \frac{r^k}{N_2^k} + \sqrt{\frac{L \ln t_2}{N_2^k}}$
  end for
end while

Figure 3.5: pseudocode of RCA-M

where

$$C_{k,j} = \frac{(|S^k| + |S^j|)\beta}{\pi_{\min}},\ \beta = \sum_{t=1}^{\infty} t^{-2}$$
$$D_k = \left(\frac{1}{\pi^k_{\min}} + \Omega^k_{\max} + 1\right),$$
$$E_k = \mu^k(1 + \Omega^k_{\max}) + \sum_{j=1}^{M} \mu^j \Omega^j_{\max},$$
$$F = \sum_{j=1}^{M} \mu^j \left(\frac{1}{\pi_{\min}} + \max_{k \in \mathcal{K}} \Omega^k_{\max} + 1\right).$$

Proof. See Appendix E. □

3.4 Numerical Results

In this section we give numerical results for the algorithms we proposed under the Gilbert-Elliot channel model in which each channel/arm has two states, *good* and *bad* (or 1, 0, respectively). For any channel k the rewards are given by $r^k_1 = 1$, $r^k_0 = 0.1$. We consider four opportunistic spectrum access (OSA) scenarios, denoted S1-S4, each consisting of 10 channels with different state transition probabilities. The state transition probabilities and mean rewards of the channels in each scenario are given in Tables 3.2 and 3.3, respectively. The four scenarios are intended to capture the following differences. In S1 channels are bursty with mean rewards not close to each other; in S2 channels are non-bursty with mean rewards not close to each other; in S3 there are bursty and non-bursty channels with mean rewards not close to each other; and in S4 there are bursty and non-bursty channels with mean rewards close to each other. All simulations are done for a time horizon $T = 10^5$, and averaged over 100 random runs. Initial states of the channels are drawn from their stationary distributions. For each algorithm that requires a regenerative state, the regenerative state of an arm for an agent is set to be the first state the agent observes from that

arm, and is kept fixed throughout a single run.

We first compute the normalized regret values, i.e., the regret per play $R(T)/M$, for RCA-M. In Figures 3.6, 3.8, 3.10, 3.12, we observe the normalized regret of RCA-M for the minimum values of L such that the logarithmic regret bound holds. However, comparing with Figures 3.7, 3.9, 3.11, 3.13 we see that the normalized regret is smaller for $L = 1$. Therefore it appears that the condition on L we have for the logarithmic bound, while sufficient, may not be necessary.

We next compute the regret of UCB with single play under the OSA model. We note that our theoretical regret bound for UCB is for rested channels but the numerical results are given for a special case of restless channels. Results in Figure 3.14 show that when $L = 1$, for S1, S3 and S4, UCB has negative regret, which means that it performs better than the best single action policy, while for S2 it has a positive regret, which is also greater than the regret of RCA with single play under S2 with $L = 1$. In Figure 3.15, we see the regret of UCB for larger values of L. As expected, the regret of UCB increases with L due to the increase in explorations. However, comparing the regret of UCB with that of RCA under the same value of L, we see that UCB outperforms RCA for all scenarios considered here. These results imply that although there is no theoretical bounds for the regret of UCB, its performance is comparable to RCA under the presented setting. This is because (1) RCA has a smaller update rate due to the random length of the regenerative cycles; thus it takes longer to use the latest observations in arm selection, and (2) even though there is no guarantee that UCB produces accurate estimates on the mean rewards, the simple structure of the problem helps UCB keep track of the shorter-term (not the stationary) quality of each arm.

Instead of its original index given in (3.1), we can also use RCA with the index

$$g^k_{t_2(b),N_2^k(t_2(b))} = \bar{r}^k(N_2^k(t_2(b))) + \sqrt{\frac{L \ln b}{B^k(b)}}. \tag{3.2}$$

The difference between the index given in (3.1) and (3.2) is that the exploration term in (3.2) depends on the number of blocks completed by an agent, while in (3.1) it depends on the number of time steps spent in SB2's of an agent. Using the fact that the average reward collected during each regenerative cycle of an arm can be modeled as an IID process, we can exploit the well known result for the IID problem *Auer et al.* (2002) which says that setting $L = 2$ is enough to get a logarithmic regret bound. The regret for single play under different scenarios is given in Figure 3.16. Comparing them with their counterparts using RCA with an L such that the logarithmic regret bound holds, we observe that the modified index results in better performance. This is because L is smaller, and the exploration is more *balanced* in a way that the growth of the exploration term does not depend on the randomness of the block lengths.

channel	1	2	3	4	5	6	7	8	9	10
S1, p_{01}	0.01	0.01	0.02	0.02	0.03	0.03	0.04	0.04	0.05	0.05
S1, p_{10}	0.08	0.07	0.08	0.07	0.08	0.07	0.02	0.01	0.02	0.01
S2, p_{01}	0.1	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
S2, p_{10}	0.9	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1
S3, p_{01}	0.01	0.1	0.02	0.3	0.04	0.5	0.06	0.7	0.08	0.9
S3, p_{10}	0.09	0.9	0.08	0.7	0.06	0.5	0.04	0.3	0.02	0.1
S4, p_{01}	0.02	0.04	0.04	0.5	0.06	0.05	0.7	0.8	0.9	0.9
S4, p_{10}	0.03	0.03	0.04	0.4	0.05	0.06	0.6	0.7	0.8	0.9

Table 3.2: transition probabilities of all channels

channel	1	2	3	4	5	6	7	8	9	10
S1	0.20	0.21	0.28	0.30	0.35	0.37	0.70	0.82	0.74	0.85
S2	0.19	0.19	0.28	0.37	0.46	0.55	0.64	0.73	0.82	0.91
S3	0.19	0.19	0.28	0.37	0.46	0.55	0.64	0.73	0.82	0.91
S4	0.460	0.614	0.550	0.600	0.591	0.509	0.585	0.580	0.577	0.550

Table 3.3: mean rewards of all channels

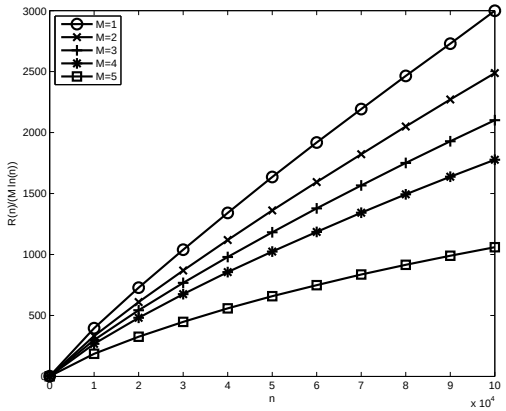

Figure 3.6: normalized regret of RCA-M: S1, $L = 7200$

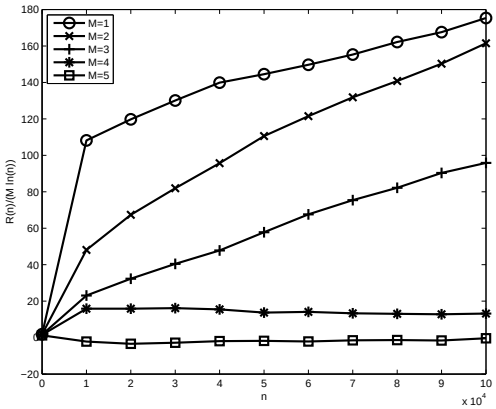

Figure 3.7: normalized regret of RCA-M: S1, $L = 1$

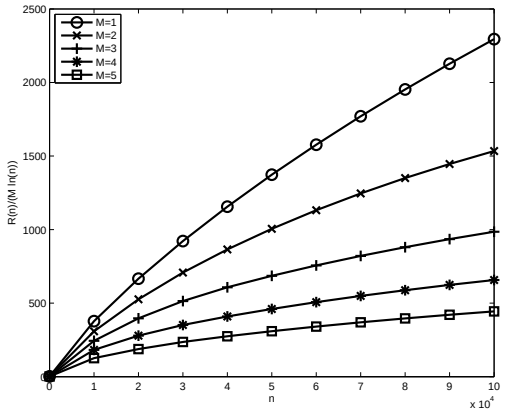

Figure 3.8: normalized regret of RCA-M: S2, $L = 360$

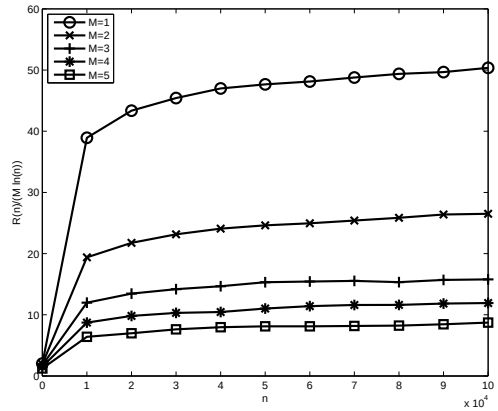

Figure 3.9: normalized regret of RCA-M: S2, $L = 1$

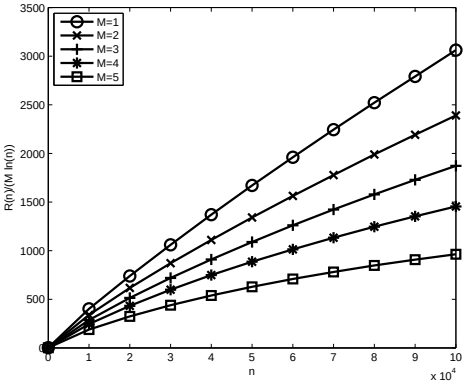

Figure 3.10: normalized regret of RCA-M: S3, $L = 3600$

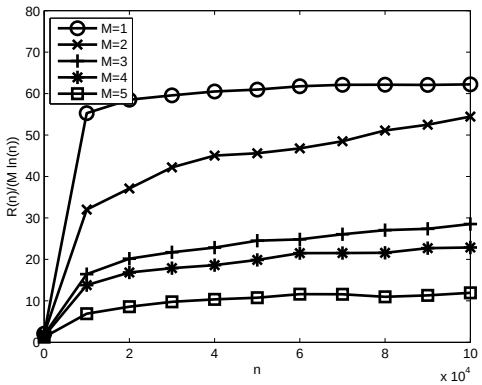

Figure 3.11: normalized regret of RCA-M: S3, $L = 1$

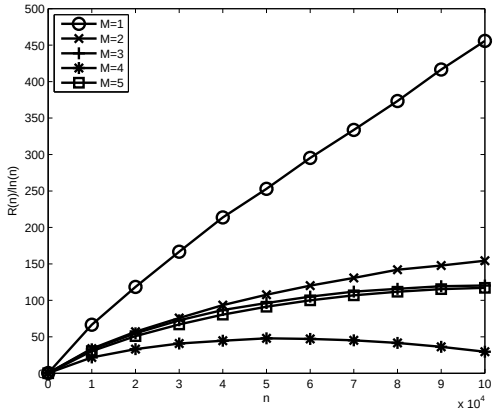

Figure 3.12: normalized regret of RCA-M: S4, $L = 7200$

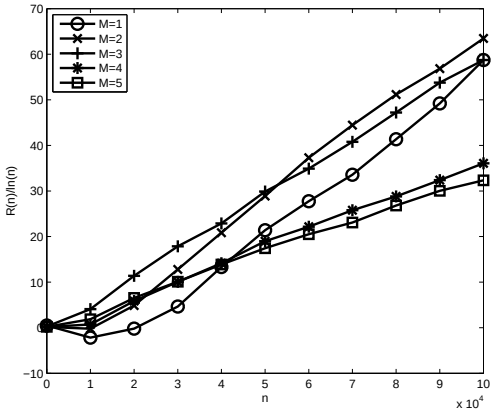

Figure 3.13: normalized regret RCA-M: S4, $L = 1$

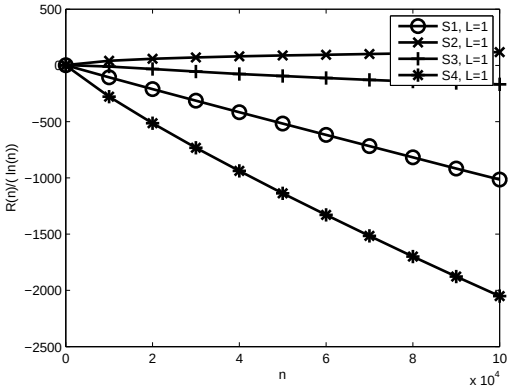

Figure 3.14: regret of UCB, $M = 1$

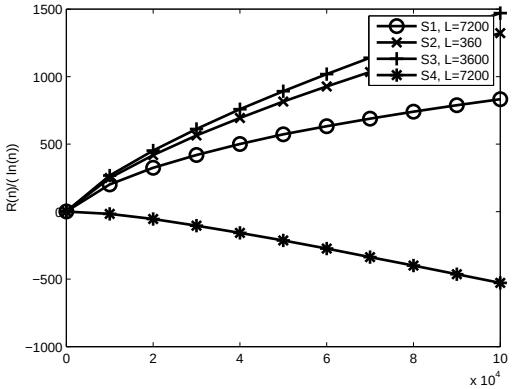

Figure 3.15: regret of UCB, $M = 1$

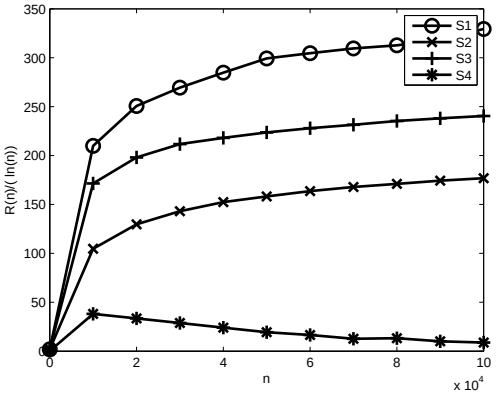

Figure 3.16: regret of RCA with modified index

3.5 Discussion

In this section we discuss how the performance of RCA-M and its special case RCA may be improved (in terms of the constants and not in order), and possible relaxation and extensions.

3.5.1 Applicability and Performance Improvement

We note that the same logarithmic bound derived in this chapter holds for the general restless bandit problem independent of the state transition law of an arm when it is not played. Indeed, the state transitions of an arm when it is not played can even be adversarial. This is because the reward to the agent from an arm is determined only by the active transition probability matrix and the first state after a discontinuity in playing the arm. Since the number of plays from any suboptimal arm is logarithmic and the expected hitting time of any state is finite, the regret is at most logarithmic independent of the first observed state of a block.

The regenerative state for an arm under RCA is chosen based on the random initial observation. It's worth noting that the selection of the regenerative state γ^k in each block in general can be arbitrary: within the same SB2, we can start and end in different states. As long as we guarantee that two successive SB2's end and start with the same state, we will have a continuous sample path for which our analysis in Section 2.2 holds.

It is possible that RCA may happen upon a state with long recurrence time which results in long SB1 and SB2 sub-blocks. Consider now the following modification: RCA records all observations from all arms. Let $N^k(s,t)$ be the total number of observations from arm k up to time t that are *excluded* from the computation of the index of arm k when the regenerative state is s. Recall that the index of an arm is computed based on observations from regenerative cycles; this implies that $N^k(s,t)$ is the total number of slots in SB1's when the regenerative state is s. Let t_b be the

time at the end of the bth block. If the arm to be played in the bth block is k then the regenerative state is set to $\gamma^k(b) = \arg\min_{s \in S^k} N^k(s, t_{b-1})$. The idea behind this modification is to estimate the state with the smallest recurrence time and choose the regenerative cycles according to this state. With this modification the number of observations that does not contribute to the index computation and the probability of choosing a suboptimal arm can be minimized over time.

3.5.2 Universality of the Block Structure

We note that any index policy used under the IID reward model can be used in the restless bandit problem with a Markovian reward model by exploiting the regenerative cycles. This is because the normalized rewards collected in each regenerative cycle of the same arm can be seen as IID samples from that arm whose expectation is equal to the mean reward of that arm. Thus, any upper bound for the expected number of times an arm is played in an IID problem will hold for the expected number of blocks an arm is played for the restless bandit problem under the block structure proposed in RCA. Specifically, we have shown via numerical results in Section 2.4 that if RCA is used with the index given in (3.2), logarithmic regret is achieved assuming that the regenerative state for each arm is kept fixed and the rewards are in the unit interval $[0, 1]$. We do not provide a technical analysis here since the details are included in the analysis of the IID model *Auer et al.* (2002) and our analysis of RCA. Instead, we illustrate the generality of the block structure by using the KL-UCB algorithm proposed in *Garivier and Cappé* (2011) for IID rewards inside our block structure. KL-UCB is shown to outperform most of the other index policies for IID rewards including UCB. For simplicity we only consider single play, i.e., $M = 1$.

Lemma III.4. *Assume P^k is such that arm k is irreducible (multiplicative symmetrization of P^k need not be irreducible), and $r_x^k \leq 1, \forall k \in \mathcal{K}, \forall x \in S^k$. Then, using*

KL-UCB in the regenerative block under RCA, we have for any suboptimal arm k

$$\limsup_{T\to\infty} \frac{E\left[B^k(b(T))\right]}{\log b(T)} \leq \frac{1}{d(\mu^k,\mu^1)},$$

where

$$d(p,q) = p\log\left(\frac{p}{q}\right) + (1-p)\log\left(\frac{1-p}{1-q}\right).$$

Proof. The normalized reward during a block (sum of the rewards collected during an SB2 divided by the length of the SB2) forms an IID process with support in $[0,1]$. Thus, the result follows from Theorem 2 of *Garivier and Cappé* (2011). □

We see as before, bounding the expected number of blocks a suboptimal arm is played is a key step in bounding the regret. The main result is given in the following theorem.

Theorem III.5. *Assume P^k is such that arm k is irreducible (multiplicative symmetrization of P^k need not be irreducible), and $r_x^i \leq 1, \forall i \in \mathcal{K}, \forall x \in S^i$. Then, using KL-UCB in the regenerative block under RCA, we have*

$$\limsup_{T\to\infty} \frac{R(T)}{\log T} \leq \sum_{k>1}\left(\frac{1}{d(\mu^k,\mu^1)}\right)\left((\mu^1-\mu^k)D_k + E_k\right),$$

where

$$D_k = \left(\frac{1}{\pi^k_{\min}} + \Omega^k_{\max} + 1\right),$$
$$E_k = \mu^i(1+\Omega^k_{\max}) + \mu^1\Omega^1_{\max}.$$

Proof. The result follows from Lemma III.4 and using steps similar to the proof of Theorem III.2. □

3.5.3 Extension to Random State Rewards

So far we considered the case where each state $x \in S^k$ corresponds to a deterministic reward r_x^k. An interesting extension is to consider the case where each state $x \in S^k$ has a random reward with a fixed, IID distribution. That is, after observing the state x, the agent receives reward r_x^k which is drawn from a probability distribution F_x^k. In our application context, this may correspond to a situation where the agent/user observes the received signal to noise ratio (SNR) which gives the probability of correct reception, but not the actual bit error rate. When the distribution (or its expectation) $F_x^k, \forall x \in S^k, k \in \mathcal{K}$ is known to the agent, the agent can use the expectation of the distribution of each state instead of the actual observed rewards to update the indices. In doing so logarithmic regret will be achieved by using RCA.

A more complex case is when the reward distribution of each state is unknown to the agent but has a bounded support. Then, to estimate the quality of each arm, playing logarithmic number of blocks from each arm may not be sufficient because there may be cases where the number of samples from some state of a sufficiently sampled arm may not be enough to accurately estimate the quality of that state. This may result in an inaccurate estimate of the expected reward received during a regenerative cycle. To avoid this, we can use arbitrary regenerative states, and modify RCA as follows: At the end of each block in which arm k is played, we record the least sampled state x of arm k up to that point. Whenever arm k is played in a block, the state x is then used as the regenerative state to terminate that block. This guarantees that x is sampled at least once during that block. Of course, to preserve the regenerative property the agent needs to set the first state of its next SB2 to x in the next block it plays arm k. This way *fairness* between the states of each arm is guaranteed. By logarithmically playing each arm, the agent can guarantee logarithmic number of samples taken from each state, thus the sample mean estimate of the expected reward of each state will be accurate. Then, the agent can use the

sample mean of the rewards for each state in calculating the index with RCA to obtain good performance. In order to have theoretical results we will need to use two large deviation bounds; one for the sample mean estimates of the rewards of each state of each arm, the other for bounding the deviation of the index from the expected reward of an arm.

3.5.4 Relaxation of Certain Conditions

As observed in Section 3.4 that the condition on L, while sufficient, does not appear necessary for the logarithmic regret bound to hold. Indeed our examples show that smaller regret can be achieved by setting $L = 1$. Note that this condition on L originates from the large deviation bound by Lezaud given in Lemma A.1. If we use an alternative bound, e.g., the large deviation bound in Lemma A.2, then $L \geq 90 S_{\max}^2 r_{\max}^2 / \epsilon_{\min}$ will be sufficient, and our theoretical results will hold for smaller L, provided that $\hat{\pi}_{\max}^2 \geq 90/112$ and the arms are reversible Markov chains.

We further note that even if no information is available on the underlying Markov chains to derive this sufficient condition on L, $O(log(T)f(T))$ regret is achievable by letting L grow slowly with time where $f(T)$ is any increasing sequence. Such approach has been used in other settings and algorithms, see e.g., *Anandkumar et al.* (2011); *Liu et al.* (2010).

We have noted earlier that the strict inequality $\mu^M > \mu^{M+1}$ is required for the restless bandit problem because in order to have logarithmic regret, we can have no more than a logarithmic number of discontinuities from the optimal arms. When $\mu^M = \mu^{M+1}$ the rankings of the indices of arms M and $M+1$ can oscillate indefinitely resulting in a large number of discontinuities. Below we briefly discuss how to resolve this issue if indeed $\mu^M = \mu^{M+1}$. Consider adding a threshold ϵ to the algorithm such that a new arm will be selected instead of an arm currently being played only if the index of that arm is at least ϵ larger than the index of the currently played arm

which has the smallest index among all currently played arms. Then given that ϵ is sufficiently small (with respect to the differences of mean rewards) indefinite switching between the Mth and the $M+1$th arms can be avoided. Further analysis is needed to verify that this approach will result in logarithmic regret.

3.5.5 Definition of Regret

We have used the weak regret measure throughout this chapter, which compares the learning strategy with the best single-action strategy. When the statistics are known a priori, it is clear that in general the best policy is not a single-action policy (in principle one can derive such a policy using dynamic programming). Ideally one could try to adopt a stronger regret measure with respect to this optimal policy. Under some conditions on the structure of the optimal policy, in Chapter IV, we propose a learning algorithm with logarithmic regret with respect to the optimal policy, which is defined as strong regret in Definition I.10. However, in general such an optimal policy is PSPACE-hard even to approximate in the restless case (see e.g., *Whittle* (1988); *Papadimitriou and Tsitsiklis* (1999)), which makes the comparison intractable, except for some very limited cases when such a policy happens to be known (see e.g., *Ahmad et al.* (2009); *Dai et al.* (2011)) or special cases when approximation algorithms with guaranteed performance are known (see e.g., *Guha et al.* (2010); *Tekin and Liu* (2012a)).

CHAPTER IV

Single-agent Restless Bandits with Strong Regret

In this chapter we study the single-agent uncontrolled restless bandit problem given in Definition I.4, and provide an algorithm whose strong regret given in Definition I.10 grows logarithmically over time. Different from the previous chapter in which we focused on computationally simple algorithms, the learning algorithm we propose in this chapter is computationally intractable. However, if we want to have logarithmic regret with respect to the optimal solution, we cannot hope to get a polynomial complexity algorithm since the optimization problem itself is PSPACE-hard to approximate as shown in *Papadimitriou and Tsitsiklis* (1999).

Therefore, in this chapter we only focus on the performance, ignoring computation and implementation issues. Since the uncontrolled restless bandit problem is a subclass of partially observable Markov decision processes (POMDPs), our results can be seen as an extension of optimal adaptive learning in finite Markov decision processes (MDPs) developed in *Agrawal et al.* (1989) and *Burnetas and Katehakis* (1997) to optimal adaptive learning in a subclass of POMDPs. The main difficulty in this extension is dealing with infinite state spaces and lack of recurrence conditions that are present in most of the finite MDPs.

The organization of the remainder of this chapter is as follows. In Section 4.1, we present the problem formulation. In Section 4.2, we introduce the average reward

optimality equation, which gives the optimal solution in terms of the average reward when the transition probabilities of the arms are known, and state the conditions under which it has a continuous solution. In Section 4.3, we give an equivalent countable representation of the information state. In Section 4.4, a learning algorithm is given. Then, in Section 4.5 we define finite partitions of the information state which is used to bound the strong regret of the learning algorithm. We analyze the strong regret in Section 4.6, and prove that it increases logarithmically in time. In Section 4.7, we propose a variant of our learning algorithm which does not require a bound on the accuracy of transition probability estimates. Then, in Section 4.8, we propose a second variant of our learning algorithm which achieves logarithmic regret without any assumptions on the structure of the optimal policy.

4.1 Problem Formulation

Consider K mutually independent uncontrolled restless Markovian arms, indexed by the set $\mathcal{K} = \{1, 2, \ldots, K\}$ whose states evolve in discrete time steps $t = 1, 2, \ldots$ according to a finite-state Markov chain with unknown transition probabilities.

Let S^k be the state space of arm k. For simplicity of presentation, we assume that for state $x \in S^k$, $r_x^k = x$, i.e., the state of an arm also represents its reward under that state. This is without loss of generality as long as one of the following is true: either the state is perfectly observed when played, or that the reward is perfectly observed when received and a reward uniquely identifies a state for a given arm (i.e., no two states have the same reward). It follows that the state space of the system is the Cartesian product of the state spaces of individual arms, denoted by $\boldsymbol{S} = S^1 \times \ldots \times S^K$. Let p_{ij}^k denote the transition probability from state i to state j of arm k. The transition probability matrix of arm k is denoted by P^k, whose ijth element is p_{ij}^k. The set of transition probability matrices is denoted by $\boldsymbol{P} = (P^1, \ldots, P^K)$. We assume that P^ks are such that each arm is ergodic. This

implies that, for each arm there exists a unique stationary distribution which is given by $\boldsymbol{\pi}^k = (\pi_x^k)_{x \in S^k}$. At each time step, the state of the system is a K-dimensional vector of states of arms which is given by $\boldsymbol{x} = (x^1, \ldots, x^K) \in \boldsymbol{S}$.

The following notation will be frequently used throughout the chapter. Let e_x^k represent the unit vector with dimension $|S^k|$, whose xth element is 1, and all other elements are 0. $\mathbb{N} = \{1, 2, \ldots\}$ denotes the set of natural numbers, $\mathbb{Z}_+ = \{0, 1, \ldots\}$ the set of non-negative integers, $(\boldsymbol{v} \bullet \boldsymbol{w})$ the standard inner product of vectors $\boldsymbol{v}$ and $\boldsymbol{w}$, $||\boldsymbol{v}||_1$ and $||\boldsymbol{v}||_\infty$ respectively the l_1 and l_∞ norms of vector $\boldsymbol{v}$, and $||P||_1$ the induced maximum row sum norm of matrix P. For a vector $\boldsymbol{v}$, $(\boldsymbol{v}_{-u}, v')$ denotes the vector whose uth element is v', while all other elements are the same as in $\boldsymbol{v}$. For a vector of matrices $\boldsymbol{P}$, $(\boldsymbol{P}_{-u}, P')$ denotes the vector of matrices whose uth matrix is P', while all other matrices are the same as in $\boldsymbol{P}$. The transpose of a vector $\boldsymbol{v}$ or matrix P is denoted by $\boldsymbol{v}^T$ or P^T, respectively. In addition, the following quantities frequently appear in this chapter:

- $\beta = \sum_{t=1}^{\infty} 1/t^2$, $\pi_{\min}^k = \min_{x \in S^k} \pi_x^k$;
- $\pi_{\min} = \min_{k \in \mathcal{K}} \pi_{\min}^k$;
- $r_{\max} = \max_{x \in S^k, k \in \mathcal{K}} r_x^k$;
- $S_{\max} = \max_{k \in \mathcal{K}} |S^k|$.

There is an agent who selects one of the K arms at each time step t, and gets a bounded reward depending on the state of the selected arm at time t. Without loss of generality, we assume that the state rewards are non-negative. Let $r^k(t)$ be the random variable which denotes the reward from arm k at time t. The objective of the agent is to maximize the undiscounted sum of the rewards over any finite horizon $T > 0$. However, the agent does not know the set of transition probability matrices $\boldsymbol{P}$. In addition, at any time step t the agent can only observe the state

of the arm it selects but not the states of the other arms. Intuitively, in order to maximize its reward, the agent needs to explore/sample the arms to estimate their transition probabilities and to reduce the uncertainty about the current state $\boldsymbol{x} \in \boldsymbol{S}$ of the system, while it also needs to exploit the information it has acquired about the system to select arms that yield high rewards. The exploration and exploitation need to be carefully balanced to yield the maximum reward for the agent. In a more general sense, the agent is learning to play optimally in an uncontrolled POMDP.

We denote the set of all possible stochastic matrices with $|S^k|$ rows and $|S^k|$ columns by Ξ^k, and let $\boldsymbol{\Xi} = (\Xi^1, \Xi^2, \ldots, \Xi^K)$. Since $\boldsymbol{P}$ is unknown to the agent, at time t the agent has an estimate of $\boldsymbol{P}$, denoted by $\hat{\boldsymbol{P}}_t \in \boldsymbol{\Xi}$. For two vectors of transition probability matrices $\boldsymbol{P}$ and $\tilde{\boldsymbol{P}}$, the distance between them is defined as $||\boldsymbol{P} - \tilde{\boldsymbol{P}}||_1 := \sum_{k=1}^{K} ||P^k - \tilde{P}^k||_1$. Let X_t^k be the random variable representing the state of arm k at time t. Then, the random vector $\boldsymbol{X}_t = (X_t^1, X_t^2, \ldots, X_t^K)$ represents the state of the system at time t.

The action space U of the agent is equal to $\mathcal{K}$ since it chooses an arm in $\mathcal{K}$ at each time step, and the observation space Y of the agent is equal to $\cup_{k=1}^{K} S^k$, since it observes the state of the arm it selects at each time step. Since the agent can distinguish different arms, for simplicity we will assume $S^k \cap S^l = \emptyset$ for $k \neq l$, so that these states may be labeled distinctly. Let $u_t \in U$ be the arm selected by the agent at time t, and $y_t \in Y$ be the state/reward observed by the agent at time t. The history of the agent at time t consists of all the actions and observations of the agent by time t, which is denoted by $\boldsymbol{z}^t = (u_1, y_1, u_2, y_2, \ldots, u_t, y_t)$. Let H^t denote the set of histories at time t. A learning algorithm $\alpha = (\alpha(1), \alpha(2), \ldots)$ for the agent, is a sequence of mappings from the set of histories to actions, i.e., $\alpha(t) : H^t \to U$. Since the history depends on the stochastic evolution of the arms, let U_t and Y_t be the random variables representing the action and the observation at time t, respectively.

Let $Q_{\boldsymbol{P}}(y|u)$ be the sub-stochastic transition probability matrix such that

$$(Q_{\boldsymbol{P}}(y|u))_{\boldsymbol{x}\boldsymbol{x}'} = P_{\boldsymbol{P}}(\boldsymbol{X}_t = \boldsymbol{x}', Y_t = y|\boldsymbol{X}_{t-1} = x, U_t = u),$$

I changed X_t to X_{t-1}. The previous version was incorrect because it is taken from work on POMDP, which is a little different from the work on bandits because when you take action at time t, you observe its result in $t+1$. where $P_{\boldsymbol{P}}(.|.)$ denotes the conditional probability with respect to distribution $\boldsymbol{P}$. For URBP, $Q_{\boldsymbol{P}}(y|u)$ is the zero matrix for $y \notin S^u$, and for $y \in S^u$, only nonzero entries of $Q_{\boldsymbol{P}}(y|u)$ are the ones for which $x^u = y$.

Let Γ be the set of admissible policies, i.e., policies γ' for which $\gamma'(t) : H^t \to U$. Note that the set of admissible policies include the set of optimal policies which are computed by dynamic programming based on $\boldsymbol{P}$. Let ψ_0 be the initial belief of the agent, which is a probability distribution over $\boldsymbol{S}$. Since we assume that the agent knows nothing about the state of the system initially, ψ_0 can be taken as the uniform distribution over $\boldsymbol{S}$.

Let $E^{\boldsymbol{P}}_{\psi,\gamma}[.]$ denote the expectation taken with respect to an algorithm or policy γ, initial state ψ, and the set of transition probability matrices $\boldsymbol{P}$. I think here by "algorithm" you mean a policy in the optimization context? This needs to be clarified as it can be confused with an "algorithm" in the learning context which is denoted by the same notation. Technically they are the same, i.e., they both produce a sequence of actions, but we don't want a reader to all of a sudden think we are talking about the learning problem... The performance of an algorithm α can be measured by its strong regret, whose value at time t is the difference between performance of the algorithm and performance of the optimal policy by time t. It is given by

$$R^{\alpha}(T) = \sup_{\gamma' \in \Gamma} \left(E^{\boldsymbol{P}}_{\psi_0,\gamma'} \left[\sum_{t=1}^{T} r^{\gamma'(t)}(t) \right] \right) - E^{\boldsymbol{P}}_{\psi_0,\alpha} \left[\sum_{t=1}^{T} r^{\alpha(t)}(t) \right]. \tag{4.1}$$

It is easy to see that the time average reward of any algorithm with sublinear regret, i.e., regret $O(T^\rho)$, $\rho < 1$, converges to the time average reward of the optimal policy. For any algorithm with sublinear regret, its regret is a measure of its convergence rate to the average reward. In Section 4.4, we will give an algorithm whose regret grows logarithmically in time, which is the best possible rate of convergence.

4.2 Solutions of the Average Reward Optimality Equation

As mentioned earlier, if the transition probability matrices of the arms are known by the agent, then the URBP becomes an optimization problem (POMDP) rather than a learning problem. In this section we discuss the solution approach to this optimization problem. This approach is then used in subsequent sections by the agent in the learning context using *estimated* transition probability matrices.

A POMDP problem is often presented using the belief space (or information state), i.e., the set of probability distributions over the state space. For the URBP with the set of transition probability matrices $\boldsymbol{P}$, the belief space is given by

$$\boldsymbol{\Psi} := \{\psi : \psi^T \in \mathbb{R}^{|\boldsymbol{S}|}, \psi_{\boldsymbol{x}} \geq 0, \forall \boldsymbol{x} \in \boldsymbol{S}, \sum_{\boldsymbol{x} \in \boldsymbol{S}} \psi_{\boldsymbol{x}} = 1\},$$

which is the unit simplex in $\mathbb{R}^{|\boldsymbol{S}|}$. Let ψ_t denote the belief of the agent at time t. Then the probability that the agent observes y given it selects arm u when the belief is ψ is given by

$$V_{\boldsymbol{P}}(\psi, y, u) := \psi Q_{\boldsymbol{P}}(y|u)\mathbf{1},$$

where $\mathbf{1}$ is the $|\boldsymbol{S}|$ dimensional column vector of 1s. Given arm u is chosen under

belief state ψ and y is observed, the next belief state is

$$T_{\boldsymbol{P}}(\psi, y, u) := \frac{\psi Q_{\boldsymbol{P}}(y|u)}{V_{\boldsymbol{P}}(\psi, y, u)} .$$

The average reward optimality equation (AROE) is

$$g + h(\psi) = \max_{u \in U} \left\{ \bar{r}(\psi, u) + \sum_{y \in S^u} V_{\boldsymbol{P}}(\psi, y, u) h(T_{\boldsymbol{P}}(\psi, y, u)) \right\}, \tag{4.2}$$

where g is a constant and h is a function from $\boldsymbol{\Psi} \to \mathbb{R}$,

$$\bar{r}(\psi, u) = (\psi \bullet r(u)) = \sum_{x^u \in S^u} x^u \phi_{u,x^u}(\psi)$$

is the expected reward of action u under belief ψ, $\phi_{u,x^u}(\psi)$ is the probability that arm u is in state x^u given belief ψ, $r(u) = (r(\boldsymbol{x}, u))_{\boldsymbol{x} \in S}$ and $r(\boldsymbol{x}, u) = x^u$ is the reward when arm u is chosen in state $\boldsymbol{x}$. Is there a significance to this statement? If so we should spell it out as in e.g., this fact is used later, etc..

Assumption IV.1. $p_{ij}^k > 0, \forall k \in \mathcal{K}, i, j \in S^k$.

When Assumption IV.1 holds, the existence of a bounded, convex continuous solution to (4.2) is guaranteed.

Let V denote the space of bounded real-valued functions on $\boldsymbol{\Psi}$. Next, we define the undiscounted dynamic programming operator $F : V \to V$. Let $v \in V$, we have

$$(Fv)(\psi) = \max_{u \in U} \left\{ \bar{r}(\psi, u) + \sum_{y \in S^u} V_{\boldsymbol{P}}(\psi, y, u) v(T_{\boldsymbol{P}}(\psi, y, u)) \right\}. \tag{4.3}$$

In the following lemma, we give some of the properties of the solutions to the average reward optimality equation and the dynamic programming operator defined above.

Lemma IV.2. *Let* $h_+ = h - \inf_{\psi\in\mathbf{\Psi}}(h(\psi))$, $h_- = h - \sup_{\psi\in\mathbf{\Psi}}(h(\psi))$ *and*

$$h_{T,\boldsymbol{P}}(\psi) = \sup_{\gamma\in\Gamma}\left(E^{\boldsymbol{P}}_{\psi,\gamma}\left[\sum_{t=1}^{T} r^{\gamma}(t)\right]\right).$$

Given that Assumption IV.1 is true, the following holds:

S-1 Consider a sequence of functions $v_0, v_1, v_2, \ldots$ *in* V *such that* $v_0 = 0$, *and* $v_l = Fv_{l-1}$, $l = 1, 2, \ldots$. *This sequence converges uniformly to a convex continuous function* v^* *for which* $Fv^* = v^* + g$ *where* g *is a finite constant. In terms of (4.2), this result means that there exists a finite constant* $g_{\boldsymbol{P}}$ *and a bounded convex continuous function* $h_{\boldsymbol{P}} : \mathbf{\Psi} \to \mathbb{R}$ *which is a solution to (4.2).*

S-2 $h_{\boldsymbol{P}-}(\psi) \leq h_{T,\boldsymbol{P}}(\psi) - Tg_{\boldsymbol{P}} \leq h_{\boldsymbol{P}+}(\psi)$, $\forall\psi \in \mathbf{\Psi}$.

S-3 $h_{T,\boldsymbol{P}}(\psi) = Tg_{\boldsymbol{P}} + h_{\boldsymbol{P}}(\psi) + O(1)$ *as* $T \to \infty$.

Proof. Sufficient conditions for the existence of a bounded convex continuous solution to the AROE are investigated in *Platzman* (1980). According to Theorem 4 of *Platzman* (1980), if reachability and detectability conditions are satisfied then S-1 holds. Below, we directly prove that reachability condition in *Platzman* (1980) is satisfied. To prove that detectability condition is satisfied, we show another condition, i.e., subrectangular substochastic matrices, holds which implies the detectability condition.

We note that $P(\boldsymbol{X}_{t+1} = \boldsymbol{x}'|\boldsymbol{X}_t = \boldsymbol{x}) > 0$, $\forall \boldsymbol{x}, \boldsymbol{x}' \in \boldsymbol{S}$ since by Assumption IV.1, $p^k_{ij} > 0$ $\forall i, j \in S^k, \forall k \in \mathcal{K}$.

Condition IV.3. (Reachability) There is a $\rho < 1$ and an integer ξ such that for all $\boldsymbol{x} \in \boldsymbol{S}$

$$\sup_{\gamma\in\Gamma}\max_{0\leq t\leq\xi} P(\boldsymbol{X}_t = \boldsymbol{x}|\psi_0) \geq 1 - \rho,\ \forall\psi_0 \in \mathbf{\Psi}.$$

Set $\rho = 1 - \min_{\boldsymbol{x},\boldsymbol{x}'} P(\boldsymbol{X}_{t+1} = \boldsymbol{x}' | \boldsymbol{X}_t = \boldsymbol{x})$, $\xi = 1$. Since the system is uncontrolled, state transitions are independent of the arm selected by the agent. Therefore,

$$\begin{aligned}\sup_{\gamma \in \Gamma} P(\boldsymbol{X}_1 = \boldsymbol{x} | \psi_0) &= P(\boldsymbol{X}_1 = \boldsymbol{x} | \psi_0) \\ &\geq \min_{\boldsymbol{x},\boldsymbol{x}'} P(\boldsymbol{X}_{t+1} = \boldsymbol{x}' | \boldsymbol{X}_t = \boldsymbol{x}) = 1 - \rho.\end{aligned}$$

Condition IV.4. (Subrectangular matrices) For any substochastic matrix $Q(y|u), y \in Y, u \in U$, and for any $i, i', j, j' \in \boldsymbol{S}$,

$$(Q(y|u))_{ij} > 0 \text{ and } (Q(y|u))_{i'j'} > 0 \;\Rightarrow (Q(y|u))_{ij'} > 0 \text{ and } (Q(y|u))_{i'j} > 0.$$

$Q(y|u)$ is subrectangular for $y \notin S^u$ since it is the zero matrix. For $y \in S^u$ all entries of $Q(y|u)$ is positive since $P(\boldsymbol{X}_{t+1} = \boldsymbol{x}' | \boldsymbol{X}_t = \boldsymbol{x}) > 0$, $\forall \boldsymbol{x}, \boldsymbol{x}' \in \boldsymbol{S}$. Yes for any t since the arms are time homogeneous Markov chains.

S-2 holds by Lemma 1 in *Platzman* (1980), and S-3 is a consequence of S-2 and the boundedness property in S-1. □

4.3 Countable Representation of the Information State

The belief space, which is the set of probability distributions over the state space, is uncountable. Since the problem we consider in this chapter is a learning problem, it is natural to assume that the agent does not have an initial belief about the state of the system, or the initial belief is just the uniform distribution over the state space. Therefore, instead of considering initial beliefs which are arbitrary distributions over the state space, we lose nothing by considering initial beliefs which are formed by playing all the arms at least once. Assume that the initial K steps are such that the agent select arm k at the kth step. Then, the POMDP for the agent can be written as a countable state MDP. Specifically, the information state at time t can

be represented by

$$(\boldsymbol{s}_t, \boldsymbol{\tau}_t) = ((s_t^1, s_t^2 \ldots, s_t^K), (\tau_t^1, \tau_t^2 \ldots, \tau_t^K)),$$

where s_t^k and τ_t^k are the last observed state of arm k and how long ago (from t) the last observation of arm k was made, respectively. Note that the countable state MDP obtained this way is a subset of the POMDP for the bandit problem in which the agent can only be in one of the countably many points in the belief space $\boldsymbol{\Psi}$ at any time step t. Our approach in this chapter is to exploit the continuity property of the AROE to bound the regret of the agent. In order to do this we need to work in $\boldsymbol{\Psi}$, thus we make a distinction between ψ_t, which is a probability distribution over the state space, and $(\boldsymbol{s}_t, \boldsymbol{\tau}_t)$ which is a sufficient information for the agent to calculate ψ_t when $\boldsymbol{P}$ is given. Therefore we call $\psi \in \boldsymbol{\Psi}$, *the belief*, and $(\boldsymbol{s}, \boldsymbol{\tau})$ *the information state*.

The contribution of the initial K steps to the regret is at most $Kr_{\max}$, which we will subsequently ignore in our analysis. We will only analyze the time steps after this initialization, and set $t = 0$ upon the completion of the initialization phase. The initial information state of the agent can be written as $(\boldsymbol{s}_0, \boldsymbol{\tau}_0)$. Let $\mathcal{C}$ be the set of all possible information states that the agent can be in. Since the agent selects a single arm at each time step, at any time $\tau^k = 1$ for the last selected arm k. I'm guessing you are reserving the term "belief state" for the ψ you previously defined, and the term "information state" for the above countable representation. If so we should make this distinction more explicit, because in the two are in general used interchangeably...

The agent can compute its belief state $\psi_t \in \Psi$ by using its transition probability estimates $\hat{\boldsymbol{P}}$ together with the information state $(\boldsymbol{s}_t, \boldsymbol{\tau}_t)$. We let $\psi_{\boldsymbol{P}}(\boldsymbol{s}_t, \boldsymbol{\tau}_t)$ be the belief that corresponds to information state $(\boldsymbol{s}_t, \boldsymbol{\tau}_t)$ when the set of transition prob-

ability matrices is $\boldsymbol{P}$. The agent knows the information state exactly, but it only has an estimate of the belief that corresponds to the information state, because it does not know the transition probabilities. The true belief computed with the knowledge of exact transition probabilities and information state at time t is denoted by ψ_t, while the estimated belief computed with estimated transition probabilities and information state at time t is denoted by $\hat{\psi}_t$.

When the belief is ψ and the set of transition probability matrices is $\boldsymbol{P}$, the set of optimal actions which are the maximizers of (4.2) is denoted by $O(\psi; \boldsymbol{P})$. When the information state is $(\boldsymbol{s}_t, \boldsymbol{\tau}_t)$, and the set of transition probability matrices is $\boldsymbol{P}$, we denote the set of optimal actions by $O((\boldsymbol{s}, \boldsymbol{\tau}); \boldsymbol{P}) := O(\psi_{\boldsymbol{P}}((\boldsymbol{s}, \boldsymbol{\tau})); \boldsymbol{P})$.

4.4 Average Reward with Estimated Probabilities (AREP)

In this section we propose the algorithm *average reward with estimated probabilities* (AREP) given in Figure 4.1, as a learning algorithm for the agent. AREP consists of exploration and exploitation phases. In the exploration phase the agent selects each arm for a certain time to form estimates of the transition probabilities, while in the exploitation phase the agent selects an arm according to the optimal policy based on the estimated transition probabilities. At each time step, the agent decides if it is an exploration phase or exploitation phase based on the accuracy of transition probability estimates. Let $N^k(t)$ be the number of times arm k is selected by time t, $N^k_{i,j}(t)$ be the number of times a transition from state i to state j of arm k is observed by the agent by time t, and $C^k_i(t)$ be the number of times a transition from state i of arm k to any state of arm k is observed by time t. Clearly,

$$C^k_i(t) = \sum_{j \in S^k} N^k_{i,j}(t).$$

Let $f(t)$ be a non-negative, increasing function which sets a condition on the accuracy

Average Reward with Estimated Probabilities (AREP)

Initialize: $f(t)$ given for $t \in \{1, 2, \ldots\}$, $t = 1$, $N^k = 0, N^k_{i,j} = 0, C^k_i = 0$, $\forall k \in \mathcal{K}, i, j \in S^k$. Play each arm once to set the initial information state $(\boldsymbol{s}, \boldsymbol{\tau})_0$. Pick $\alpha(0)$ randomly.
while $t \geq 1$ **do**
  $\bar{p}^k_{ij} = (I(N^k_{i,j} = 0) + N^k_{i,j})/(|S^k| I(C^k_i = 0) + C^k_i)$
  $\hat{p}^k_{ij} = (\bar{p}^k_{ij})/(\sum_{l \in S^k} \bar{p}^k_{il})$
  $W = \{k \in \mathcal{K} :$ there exists $i \in S^k$ such that $C^k_i < f(t)\}$.
  if $W \neq \emptyset$ **then**
    EXPLORE
    if $\alpha(t-1) \in W$ **then**
      $\alpha(t) = \alpha(t-1)$
    else
      select $\alpha(t) \in W$ arbitrarily
    end if
  else
    EXPLOIT
    solve $\hat{g}_t + \hat{h}_t(\psi) = \max_{u \in U} \{\bar{r}(\psi, u) + \sum_{y \in S^u} V(\psi, y, u) \hat{h}_t(T_{\hat{\boldsymbol{P}}_t}(\psi, y, u))\}, \forall \psi \in \Psi$.
    Let $\hat{\psi}_t$ be the estimate of the belief at time t based on $(\boldsymbol{s}_t, \boldsymbol{\tau}_t)$ and $\hat{\boldsymbol{P}}_t$.
    compute the indices of all actions at $\hat{\psi}_t$:
    $\forall u \in U$, $\mathcal{I}_t(\hat{\psi}_t, u) = \bar{r}(\hat{\psi}_t, u) + \sum_{y \in S^u} V(\hat{\psi}_t, y, u) \hat{h}_t(T_{\hat{\boldsymbol{P}}_t}(\hat{\psi}_t, y, u))$.
    Let u^* be the arm with the highest index (arbitrarily select one if there is more than one such arm).
    $\alpha(t) = u^*$.
  end if
  Receive reward $r^{\alpha(t)}(t)$, i.e., state of $\alpha(t)$ at t
  Compute $(\boldsymbol{s}_{t+1}, \boldsymbol{\tau}_{t+1})$
  if $\alpha(t-1) = \alpha(t)$ **then**
    for $i, j \in S^{\alpha(t)}$ **do**
      if State j is observed at t, state i is observed at $t-1$ **then**
        $N^{\alpha(t)}_{i,j} = N^{\alpha(t)}_{i,j} + 1$, $C^{\alpha(t)}_i = C^{\alpha(t)}_i + 1$.
      end if
    end for
  end if
  $t := t + 1$
end while

Figure 4.1: pseudocode for the Average Reward with Estimated Probabilities (AREP)

of estimates. If $C^k_i(t) < f(t)$ for some $k \in \mathcal{K}$, $i \in S^k$, the agent explores at time t. Otherwise, the agent exploits at time t. In other words, the agent concludes that the sample mean estimates are accurate enough to compute the optimal action correctly when $C^k_i(t) \geq f(t)$ $\forall k \in \mathcal{K}$, $i \in S^k$. In an exploration step, in order to update the estimate of $p^k_{ij}, j \in S^k$, the agent does the following. It selects arm k until state i is observed, then selects arm k again to observe the next state after i. Then, the agent

forms the following sample mean estimates of the transition probabilities:

$$\bar{p}_{ij,t}^{k} := \frac{N_{i,j}^{k}(t)}{C_{i}^{k}(t)} \ , i, j \in S^{k}.$$

In order for this estimates to form a probability distribution, the agent should have $\sum_{j \in S^k} \bar{p}_{ij,t}^{k} = 1$. Therefore, instead of the estimates $\bar{p}_{ij,t}^{k}$, the agent uses the normalized estimates, i.e.,

$$\hat{p}_{ij,t}^{k} := \frac{\bar{p}_{ij,t}^{k}}{\sum_{l \in S^k} \bar{p}_{il,t}^{k}} \ .$$

If AREP is in the exploitation phase at time t, the agent first computes $\hat{\psi}_t$, the estimated belief at time t, using the set of estimated transition probability matrices $\hat{\boldsymbol{P}}_t$. Then, it solves the average reward optimality equation using $\hat{\boldsymbol{P}}_t$, for which the solution is given by $\hat{g}_t$ and $\hat{h}_t$. We assume that the agent can compute the solution at every time step, independent of the complexity of the problem. Even if the agent cannot exactly solve it, it can use value iteration and belief state space discretization to compute an approximate solution. Evaluation of the performance of approximate solutions is out of the scope of this chapter. This solution is used to compute the indices (given on line 18 of AREP) as

$$\mathcal{I}_t(\hat{\psi}_t, u) = \bar{r}(\hat{\psi}_t, u) + \sum_{y \in S^u} V(\hat{\psi}_t, y, u) \hat{h}_t(T_{\hat{\boldsymbol{P}}_t}(\hat{\psi}_t, y, u)),$$

for each action $u \in U$ at estimated belief $\hat{\psi}_t$. $\mathcal{I}_t(\hat{\psi}_t, u)$ represents the advantage of choosing action u starting from information state $\hat{\psi}_t$, i.e, the sum of gain and bias. After computing the indices for each action, the agent selects the action with the highest index. In case of a tie, one of the actions with the highest index is randomly selected. Note that it is possible to update the state transition probabilities even in the exploitation phase given that the arms selected at times $t-1$ and t are the

same. Thus $C_i^k(t)$ may also increase in an exploitation phase, and the number of explorations may be smaller than the number of explorations needed in the worst case, in which the transition probability estimates are only updated at exploration steps.

In the subsequent sections we bound the strong regret of AREP by bounding the number of times a suboptimal arm selection is made at the information states visited by the agent. Since there are infinitely many information states, in order to bound the sum of the numbers of suboptimal plays we need to form a finite partition of the information states. We do this in the next subsection. In the next section, we define partitions of the agent's belief state space that are used to bound the regret. From now on we will denote AREP by α.

4.5 Finite Partitions of the Information State

Note that even when the agent is given the optimal policy as a function of the belief state for any time horizon T, it may not be able to play optimally because it does not know the exact belief ψ_t at time t. In this case, one way to ensure that the agent plays optimally is to have an $\epsilon > 0$ such that if $||\psi_t - \hat{\psi}_t||_1 < \epsilon$, the set of actions that are optimal in $\hat{\psi}_t$ is a subset of the set of actions that are optimal in ψ_t. This is indeed the case, and we prove it by exploiting the continuity of the solution to (4.2) under Assumption IV.1.

We start by defining finite partitions of the set of information states $\mathcal{C}$.

Definition IV.5. Let $\tau_{\text{tr}} > 0$ be an integer which denotes the threshold in time lag. This threshold is used to group information states in a way that treats all states of an arm which is not played for more than this threshold as a single group. Consider a vector $\boldsymbol{i} = (i^1, \ldots, i^K)$ such that either $i^k = \tau_{\text{tr}}$ or $i^k = (s^k, \tau^k), \tau_k < \tau_{\text{tr}}, s^k \in S^k$. Each vector will define a set in the partition of $\mathcal{C}$ so we call $\boldsymbol{i}$ a *partition vector*. The

sets defined by different partition vectors form a partition of $\mathcal{C}$ such that each set in this partition either consists of a single information state, or it includes infinitely many information states of the arms for which $i^k = \tau_{\text{tr}}$. Maybe we should clarify that each vector defines a point in the set $\mathcal{C}$, except for vectors containing at least one τ_{tr}; such a vector denotes a set of $\mathcal{C}$. I think we did it below. Let $\mathcal{G}_{\tau_{\text{tr}}}$ denote the partition formed by τ_{tr}. Let $\boldsymbol{s}'(\boldsymbol{i}) = \{s^k : i^k \neq \tau_{\text{tr}}\}$ and $\boldsymbol{\tau}'(\boldsymbol{i}) = \{\tau^k : i^k \neq \tau_{\text{tr}}\}$. Let $\mathcal{M}(\boldsymbol{i}) := \{k : i^k = \tau_{\text{tr}}\}$ be the set of arms that are played at least τ_{tr} time steps ago. Let $\bar{\mathcal{M}}(\boldsymbol{i}) := \mathcal{K} - \mathcal{M}(\boldsymbol{i})$. Vector $\boldsymbol{i}$ forms the following set in the partition $\mathcal{G}_{\tau_{\text{tr}}}$.

$$G_{\boldsymbol{i}} = \{(\boldsymbol{s}, \boldsymbol{\tau}) \in \mathcal{C} : (\boldsymbol{s}_{\bar{\mathcal{M}}(\boldsymbol{i})} = \boldsymbol{s}'(\boldsymbol{i}), \boldsymbol{\tau}_{\bar{\mathcal{M}}(\boldsymbol{i})} = \boldsymbol{\tau}'(\boldsymbol{i})), s^k \in S^k, \tau^k \geq \tau_{\text{tr}}, \forall k \in \mathcal{M}(\boldsymbol{i})\}.$$

Let $A(\tau_{\text{tr}})$ be the number of sets in partition $\mathcal{G}_{\tau_{\text{tr}}}$. Re-index the sets in $\mathcal{G}_{\tau_{\text{tr}}}$ as $G_1, G_2, \ldots, G_{A(\tau_{\text{tr}})}$.

Consider a set of transition probability matrices $\tilde{\boldsymbol{P}}$ for which Assumption IV.1 holds. Since each arm is ergodic, when we map a set G_l with infinitely many information states to the belief space using $\psi_{\tilde{\boldsymbol{P}}}$, for any $\delta > 0$, only a finite number of information states in G_l will lie outside the radius-δ ball centered at the joint stationary distribution of arms for which $i^k = \tau_{\text{tr}}$. Is this a unique point? these arms will be in stationary distribution, but the other arms are not...

For a set $G_l \in \mathcal{G}_{\tau_{\text{tr}}}$, given a set of transition probability matrices $\boldsymbol{P}$, we define its center as follows. If G_l only contains a single information state, then the belief corresponding to that information state is the center of G_l. If G_l contains infinitely many information states, then the center of G_l is the belief in which all arms for which $i^k = \tau_{\text{tr}}$ are in their stationary distribution based on $\boldsymbol{P}$. In both cases, the belief which is the center of G_l is denoted by $\psi^*(G_l; \boldsymbol{P})$. Let $O^*(G_l; \boldsymbol{P})$ be the set of optimal actions at this belief. Note that as τ_{tr} increases, the number of sets with infinitely many elements increases, and each of these sets are centered around the joint

stationary distribution. Similarly, as τ_{tr} increases, the number of sets with a single information state increases. I think this is true. For example when $\tau_{tr} = 2$, only the information states with $\tau_k = 1$ for all k can form a set with a single information state. However when $\tau_{tr} = 3$ informations states for which $\tau_k = 1$ or $\tau^k = 2$ for all k form a set with a single information state. However, the number of sets with infinitely many elements remains always the same, because these can only be the sets that are around the joint stationary distributions of the arms, and the number of joint stationary distributions of the arms cannot change. The points in belief space corresponding to these sets are shown in Figure 4.2. Example IV.6 shows the partition of $\mathcal{C}$ formed by $\tau_{\text{tr}} = 3$ when $K = 2$.

Example IV.6. Let $K = 2$, $S^1 = \{0, 2\}$, $S^2 = \{1\}$ and $\tau_{\text{tr}} = 3$. For convinicence let $(\boldsymbol{s}, \boldsymbol{\tau}) = ((s^1, \tau^1), (s^2, \tau^2))$. Then the partition formed by τ_{tr}, i.e., $\mathcal{G}_{\tau_{\text{tr}}}$ contains the following sets:

$$
\begin{aligned}
&G_1 = \{((0,1),(1,2))\}, \ G_2 = \{((2,1),(1,2))\}, \\
&G_3 = \{((0,2),(1,1))\}, \ G_4 = \{((2,2),(1,1))\}, \\
&G_5 = \{((0,1),(1,3)), \ ((0,1),(1,4)), \ldots\}, \\
&G_6 = \{((2,1),(1,3)), \ ((2,1),(1,4)), \ldots\}, \\
&G_7 = \{((0,3),(1,1)), \ ((2,3),(1,1)), \ ((0,4),(1,1)), \ ((2,4),(1,1)), \ldots\}
\end{aligned}
$$

Next, we define extensions of the sets G_l on the belief space. For a set $B \in \boldsymbol{\Psi}$ let $B(\epsilon)$ be the ϵ *extension* of that set, i.e.,

$$B(\epsilon) = \{\psi \in \boldsymbol{\Psi} : \psi \in B \text{ or } d_1(\psi, B) < \epsilon\},$$

where $d_1(\psi, B)$ is the minimum l_1 distance between ψ and any element of B. The ϵ-extension of $G_l \in \mathcal{G}_{\tau_{\text{tr}}}$ corresponding to $\boldsymbol{P}$ is the ϵ-extension of the convex-hull of the

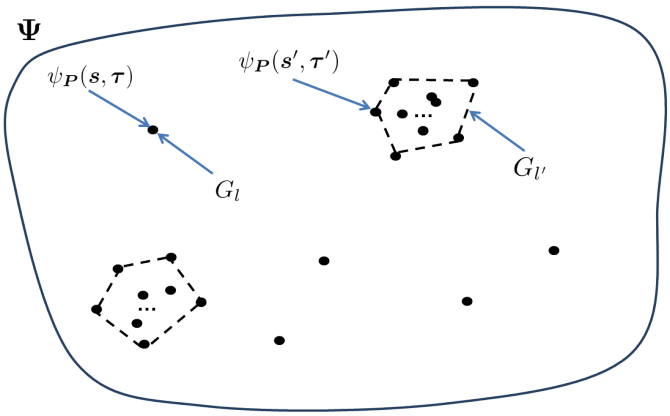

Figure 4.2: Partition of $\mathcal{C}$ on $\boldsymbol{\Psi}$ based on $\boldsymbol{P}$ and τ_{tr}. G_l is a set with a single information state and $G_{l'}$ is a set with infinitely many information states.

points $\psi_{\boldsymbol{P}}(\boldsymbol{s},\boldsymbol{\tau})$ such that $(\boldsymbol{s},\boldsymbol{\tau}) \in G_l$. Let $J_{l,\epsilon}$ denote the ϵ-extension of G_l. Examples of $J_{l,\epsilon}$ on the belief space is given in Figure 4.5.

Let the diameter of a set B be the maximum distance between any two elements of that set. Another observation is that when τ_{tr} increases, the diameter of the convex-hull of the points of the sets in $\mathcal{G}_{\tau_{\text{tr}}}$ that contains infinitely many elements decreases. In the following lemma, we show that when τ_{tr} is chosen large enough, there exists $\epsilon > 0$ such for all $G_l \in \mathcal{G}_{\tau_{\text{tr}}}$, we have non-overlapping ϵ-extensions in which only a subset of the actions in $O^*(G_l;\boldsymbol{P})$ is optimal.

Lemma IV.7. *For any $\boldsymbol{P}$, $\exists\ \tau_{tr} > 0$ and $\epsilon > 0$ such that for all $G_l \in \mathcal{G}_{\tau_{tr}}$, its ϵ-extension $J_{l,\epsilon}$ has the following properties:*

i For any $\psi \in J_{l,\epsilon}$, $O(\psi;\boldsymbol{P}) \subset O^*(G_l;\boldsymbol{P})$.

ii For $l \neq l'$, $J_{l,\epsilon} \cap J_{l',\epsilon} = \emptyset$.

Proof. For $G_l \in \mathcal{G}_{\tau_{\text{tr}}}$ consider its center $\psi^*(G_l;\boldsymbol{P})$. For any $\psi \in \boldsymbol{\Psi}$ the suboptimality gap is defined as

$$\Delta(\psi,\boldsymbol{P}) = \max_{u\in U}\left\{\bar{r}(\psi,u) + \sum_{y\in S^u} V_{\boldsymbol{P}}(\psi,y,u)h(T_{\boldsymbol{P}}(\psi,y,u))\}\right\}$$

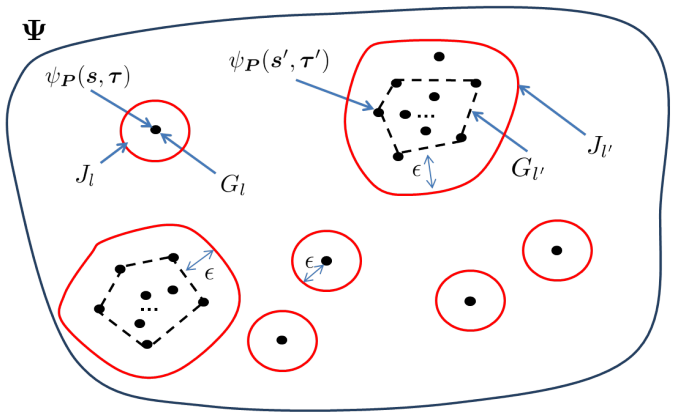

Figure 4.3: ϵ-extensions of the sets in $\mathcal{G}_{\tau_{\text{tr}}}$ on the belief space.

$$- \max_{u \in U - O(\psi;\boldsymbol{P})} \left\{ \{\bar{r}(\psi,u) + \sum_{y \in S^u} V_{\boldsymbol{P}}(\psi,y,u) h(T_{\boldsymbol{P}}(\psi,y,u)) \right\}. \tag{4.4}$$

Since r, h, V and T are continuous in ψ, we can find an $\epsilon > 0$ such that for any $\psi \in B_{2\epsilon}(\psi^*(G_l;\boldsymbol{P}))$ and for all $u \in U$,

$$\left| \bar{r}(\psi^*(G_l;\boldsymbol{P}),u) + \sum_{y \in S^u} V_{\boldsymbol{P}}(\psi^*(G_l;\boldsymbol{P}),y,u) h(T_{\boldsymbol{P}}(\psi^*(G_l;\boldsymbol{P}),y,u)) \right.$$
$$\left. -\bar{r}(\psi,u) + \sum_{y \in S^u} V_{\boldsymbol{P}}(\psi,y,u) h(T_{\boldsymbol{P}}(\psi,y,u)) \right| < \Delta(\psi^*(G_l;\boldsymbol{P}),\boldsymbol{P})/2, \tag{4.5}$$

and $B_{2\epsilon}(\psi^*(G_l;\boldsymbol{P})) \cap B_{2\epsilon}(\psi^*(G_{l'};\boldsymbol{P})) = \emptyset$ for $l \neq l'$. Therefore, any action u which is not in $O^*(G_l;\boldsymbol{P})$ cannot be optimal for any $\psi \in B_{2\epsilon}(\psi^*(G_l;\boldsymbol{P}))$. Since the diameter of the convex-hull of the sets that contains infinitely many information states decreases with τ_{tr}, there exists $\tau_{\text{tr}} > 0$ such that for any $G_l \in \mathcal{G}_{\tau_{\text{tr}}}$, the diameter of the convex-hull $J_{l,0}$ is less than ϵ. Let τ_{tr} be the smallest integer such that this holds. Then, the ϵ-extension of the convex hull $J_{l,\epsilon}$ is included in the ball $B_{2\epsilon}(\psi^*(G_l;\boldsymbol{P}))$ for all $G_l \in \mathcal{G}_{\tau_{\text{tr}}}$. This concludes the proof. □

Remark IV.8. According to Lemma IV.7, although we can find an ϵ-extension in

which a subset of $O^*(G_l; \boldsymbol{P})$ is optimal for any $\psi, \psi' \in J_{l,\epsilon}$, the set of optimal actions for ψ may be different from the set of optimal actions for ψ'. Note that agent's estimated belief $\hat{\psi}_t$ is different from the true belief ψ_t. If no matter how close $\hat{\psi}_t$ to ψ_t, the set of optimal actions for the two is different, then the agent can make a suboptimal decision even if it knows the optimal policy. The performance loss of the agent, which can be bounded by the number of suboptimal decisions, may grow linearly over time. It appears that this is a serious problem in the design of an efficient learning algorithm. In this chapter, we present two different approaches which will make performance loss (regret) grow logarithmically in time. The first approach is based on an assumption about the structure of the optimal policy, while the second approach is to construct an algorithm that will almost always choose near-optimal actions, whose suboptimality can be bounded by a function of the time horizon T.

Assumption IV.9. *There exists $\tau_{tr} \in \mathbb{N}$ such that for any $G_l \in \mathcal{G}_{\tau_{tr}}$, there exists $\epsilon > 0$ such that the same subset of $O^*(G_l; \boldsymbol{P})$ is optimal for any $\psi \in J_{l,\epsilon} - \psi^*(G_l; \boldsymbol{P})$.*

When this assumption is correct, if ψ_t and $\hat{\psi}_t$ are sufficiently close to each other, then the agent will always chose an optimal arm. Assume that this assumption is false. Consider the *stationary* information states for which $\tau^k = \infty$ for some arm k. Then for any $\tau_{\text{tr}} > 0$, there exists a set $G_l \in \mathcal{G}_{\tau_{\text{tr}}}$ and a sequence of information states $(\boldsymbol{s}, \boldsymbol{\tau})_n$, $n = 1, 2, \ldots$, such that $\psi_{\boldsymbol{P}}((\boldsymbol{s}, \boldsymbol{\tau})_n)$ converges to $\psi^*(G_l; \boldsymbol{P})$ but there exists infinitely many n's for which $O((\boldsymbol{s}, \boldsymbol{\tau})_n; \boldsymbol{P}) \neq O((\boldsymbol{s}, \boldsymbol{\tau})_{n+1}; \boldsymbol{P})$.

For simplicity of analysis, we focus on the following version of Assumption IV.9, although our results in Section 4.6 will also hold when Assumption IV.9 is true.

Assumption IV.10. *There exists $\tau_{tr} \in \mathbb{N}$ such that for any $G_l \in \mathcal{G}_{\tau_{tr}}$, a single action is optimal for $\psi^*(G_l; \boldsymbol{P})$.*

Corollary IV.11. *Let $\tau_{tr} \in \mathbb{N}$ be the minimum integer such that Assumption IV.10 holds. Then, there exists $\bar{\epsilon} > 0$, depending on τ_{tr}, such that for all $\epsilon \leq \bar{\epsilon}$, and any*

$\psi \in J_{l,\epsilon}$, *a single action is optimal.*

Proof. Result follows from Assumption IV.10 and Lemma IV.7. □

Remark IV.12. Although we don't know a way to check if Assumption IV.10 holds given a set of transition probability matrices $\boldsymbol{P}$, we claim that it holds for a large set of $\boldsymbol{P}$s. To explain better we may need a rigorous proof but I don't know how we can do it know. I wanted to say that at any time step the reward distributions and the expected rewards of all the arms will be different no matter in which information state we are in. Therefore at each time step there will only be a single optimal arm. The agent's selection does not affect state transitions of the arms; it only affects the agent's reward by changing the information state. Moreover, each arm evolves independently from each other. Assume that $\boldsymbol{P}$ is arbitrarily selected from $\boldsymbol{\Xi}$, and the state reward r_x^k, $x \in S^k$ is arbitrarily selected from $[0, r_{\max}]$. Then at any information state $(\boldsymbol{s}, \boldsymbol{\tau}) \in \mathcal{C}$, the probability that the reward distribution of two arms are the same will be zero. Based on this, we claim that Assumption IV.10 holds with probability one, if the arm rewards and $\boldsymbol{P}$ is chosen from the uniform distribution on $\boldsymbol{\Psi} \times [0, r_{\max}]$. In other words, the set of arm rewards and transition probabilities for which Assumption IV.10 does not hold is a measure zero subset of $\boldsymbol{\Psi} \times [0, r_{\max}]$.

4.6 Analysis of the Strong Regret of AREP

In this section we show that when $\boldsymbol{P}$ is such that Assumptions IV.1 and IV.10 hold, if the agent uses AREP with $f(t) = L \log t$ with L sufficiently large, i.e., the exploration constant $L \geq C(\boldsymbol{P})$, where $C(\boldsymbol{P})$ is a constant that depends on $\boldsymbol{P}$, then the regret due to explorations will be logarithmic in time, while the regret due to all other terms are finite, independent of t. No it does not only depend on the dimension. It is true that Lemma 8 depends only on the dimension, but $C_P(\epsilon)$ on Lemma 8 is different form the constant here. Because the value of ϵ we want depends on P Note

that since the agent does not know $\boldsymbol{P}$, it cannot know how large it should chose L. For simplicity we assume that the agent starts with an L that is large enough without knowing $C(\boldsymbol{P})$. We also prove a near-logarithmic regret result when the agent sets $f(t) = L(t)\log t$, where $L(t)$ is a positive increasing function over time such that $\lim_{t\to\infty} L(t) = \infty$.

In the following lemma, using Lemma A.7, we show that the probability that an estimated transition probability is significantly different from the true transition probability given AREP is in an exploitation phase is very small.

Lemma IV.13. *For any $\epsilon > 0$, for an agent using AREP with constant $L \geq C_{\boldsymbol{P}}(\epsilon)$, we have*

$$P\left(|\hat{p}^k_{ij,t} - p^k_{ij}| > \epsilon, \mathcal{E}_t\right) := P\left(\{|\hat{p}^k_{ij,t} - p^k_{ij}| > \epsilon\} \cap \mathcal{E}_t\right) \leq \frac{2S_{\max} + 2}{t^2},$$

for all $t > 0$, $i, j \in S^k$, $k \in \mathcal{K}$, where $C_{\boldsymbol{P}}(\epsilon)$ is a constant that depends on $\boldsymbol{P}$ and ϵ.

Proof. See Appendix F. □

4.6.1 An Upper Bound on the Strong Regret

For any admissible policy α, the regret with respect to the optimal T horizon policy is given in (4.1), which we restate below:

$$\sup_{\gamma\in\Gamma}\left(E^{\boldsymbol{P}}_{\psi_0,\gamma}\left[\sum_{t=1}^{T} r^{\gamma(t)}(t)\right]\right) - E^{\boldsymbol{P}}_{\psi_0,\alpha}\left[\sum_{t=1}^{T} r^{\alpha(t)}(t)\right].$$

First, we derive the regret with respect to the optimal policy as a function of the number of suboptimal plays. Before proceeding, we define expressions to compactly represent the right hand side of the AROE. Let

$$\mathcal{L}(\psi, u, h, \boldsymbol{P}) := \bar{r}(\psi, u) + (V(\psi, ., u) \bullet h(T_{\boldsymbol{P}}(\psi, ., u)))$$

$$\mathcal{L}^*(\psi, \boldsymbol{P}) := \max_{u \in U} \mathcal{L}(\psi, u, h_{\boldsymbol{P}}, \boldsymbol{P}).$$

Let

$$\Delta(\psi, u; \boldsymbol{P}) := \mathcal{L}^*(\psi, \boldsymbol{P}) - \mathcal{L}(\psi, u, h_{\boldsymbol{P}}, \boldsymbol{P}), \tag{4.6}$$

denote the degree of suboptimality of action u at information state ψ when the set of transition probability matrices is $\boldsymbol{P}$. From Proposition 1 in *Burnetas and Katehakis* (1997), we have for all $\gamma \in \Gamma$

$$R^{\gamma}_{(\psi_0;\boldsymbol{P})}(T) = \sum_{t=1}^{T} E^{\boldsymbol{P}}_{\psi_0,\gamma}[\Delta(\psi_t, U_t; \boldsymbol{P})] + \bar{C}_{\boldsymbol{P}}, \tag{4.7}$$

for some constant $\bar{C}_{\boldsymbol{P}}$, depending on $\boldsymbol{P}$. We have used the subscript $(\psi_0; \boldsymbol{P})$ to denote the dependence of regret to the initial belief and the transition probabilities. We assume that initially all the arms are sampled once thus the initial belief is $\psi_0 = \psi_{\boldsymbol{P}}((\boldsymbol{s}_0, \boldsymbol{\tau}_0))$. For the true set of transition probability matrices $\boldsymbol{P}$, let τ_{tr} and ϵ be such that Corollary IV.11 is true. Specifically, let τ_{tr} be the minimum over all possible values so that Corollary IV.11 is true, and ϵ be the maximum over all possible values given τ_{tr} so that Corollary IV.11 is true. Then, denote the ϵ-extension of the set $G_l \in \mathcal{G}_{\tau_{\text{tr}}}$ by $J_{l,\epsilon}$. Note that at any t, the belief $\psi_t \in J_{l,\epsilon}$ for some l. When ϵ is clear from the context, we simply write $J_{l,\epsilon}$ as J_l.

Let

$$\bar{\Delta}(J_l, u; \boldsymbol{P}) := \sup_{\psi \in J_l} \Delta(\psi, u; \boldsymbol{P}).$$

Recall that U_t is the random variable that denotes the arm selected by the agent at time t, which depends on the policy used by the agent. Note that if $U_t \in O(\psi_t; \boldsymbol{P})$ then $\Delta(\psi_t, U_t; \boldsymbol{P})$=0, else $U_t \notin O(\psi_t; \boldsymbol{P})$ then $\Delta(\psi_t, U_t; \boldsymbol{P}) \leq \bar{\Delta}(J_l, U_t; \boldsymbol{P})$ with probability

one. Let

$$N_T(J_l, u) := \sum_{t=1}^{T} I(\psi_t \in J_l, U_t = u).$$

We have the following lemma.

Lemma IV.14. *For any admissible policy γ,*

$$R^{\gamma}_{(\psi_0;\boldsymbol{P})}(T) \leq \sum_{l=1}^{A(\tau_{tr})} \sum_{u \notin O(J_l;\boldsymbol{P})} E^{\boldsymbol{P}}_{\psi_0,\gamma}[N_T(J_l, u)]\bar{\Delta}(J_l, u; \boldsymbol{P}) + \bar{C}_{\boldsymbol{P}}.$$

Proof.

$$\begin{aligned} R^{\gamma}_{(\psi_0;\boldsymbol{P})}(T) &\leq \sum_{t=1}^{T} E^{\boldsymbol{P}}_{\psi_0,\gamma}\left[\sum_{l=1}^{A} \sum_{u \notin O(J_l;\boldsymbol{P})} I(\psi_t \in J_l, U_t = u)\bar{\Delta}(J_l, u; \boldsymbol{P})\right] + \bar{C}_{\boldsymbol{P}} \\ &= \sum_{l=1}^{A} \sum_{u \notin O(J_l;\boldsymbol{P})} E^{\boldsymbol{P}}_{\psi_0,\gamma}\left[\sum_{t=1}^{T} I(\psi_t \in J_l, U_t = u)\right] \bar{\Delta}(J_l, u; \boldsymbol{P}) + \bar{C}_{\boldsymbol{P}} \\ &= \sum_{l=1}^{A} \sum_{u \notin O(J_l;\boldsymbol{P})} E^{\boldsymbol{P}}_{\psi_0,\gamma}[N_T(J_l, u)]\bar{\Delta}(J_l, u; \boldsymbol{P}) + \bar{C}_{\boldsymbol{P}}. \end{aligned}$$

□

Now consider AREP, which is denoted by α. We will upper bound $N_T(J_l, u)$ for suboptimal actions u by a sum of expressions which we will upper bound individually. Let $\mathcal{E}_t$ be the event that AREP is in an exploitation step at time t and

$$\mathcal{F}_t(\epsilon) := \left\{ \left\| \hat{h}_t - h_{\boldsymbol{P}} \right\|_{\infty} \leq \epsilon \right\}.$$

For an event $\mathcal{F}$, denote its complement by $\mathcal{F}^c$. Consider the following random variables which count the number of times some event has happened by time T. Since

they all depend on T, we drop the time script in the notation for convenience.

$$
\begin{aligned}
D_{1,1}(\epsilon, J_l, u) &:= \sum_{t=1}^{T} I\left(\hat{\psi}_t \in J_l, U_t = u, \mathcal{E}_t, \mathcal{F}_t(\epsilon)\right), \\
D_{1,2}(\epsilon) &:= \sum_{t=1}^{T} I(\mathcal{E}_t, \mathcal{F}_t^c(\epsilon)), \\
D_1(\epsilon, J_l, u) &:= D_{1,1}(\epsilon, J_l, u) + D_{1,2}(\epsilon), \\
D_{2,1}(\epsilon) &:= \sum_{t=1}^{T} I(||\psi_t - \hat{\psi}_t||_1 > \epsilon, \mathcal{E}_t), \\
D_{2,2}(\epsilon, J_l) &:= \sum_{t=1}^{T} I(||\psi_t - \hat{\psi}_t||_1 \leq \epsilon, \hat{\psi}_t \notin J_l, \psi_t \in J_l, \mathcal{E}_t), \\
D_2(\epsilon, J_l) &:= D_{2,1}(\epsilon) + D_{2,2}(\epsilon, J_l).
\end{aligned}
$$

Lemma IV.15. *For any $\boldsymbol{P}$ satisfying Assumption, we have IV.9*

$$
\begin{aligned}
E^{\boldsymbol{P}}_{\psi_0,\gamma}[N_T(J_l, u)] \leq E^{\boldsymbol{P}}_{\psi_0,\gamma}[D_1(\epsilon, J_l, u)] + E^{\boldsymbol{P}}_{\psi_0,\gamma}[D_2(\epsilon, J_l)] \\
+ E^{\boldsymbol{P}}_{\psi_0,\gamma}\left[\sum_{t=1}^{T} I(\mathcal{E}_t^c)\right].
\end{aligned} \tag{4.8}
$$

Yes u is any action optimal or suboptimal

Proof.

$$
\begin{aligned}
&N_T(J_l, u) = \sum_{t=1}^{T} (I(\psi_t \in J_l, U_t = u, \mathcal{E}_t) + I(\psi_t \in J_l, U_t = u, \mathcal{E}_t^c)) \\
&\leq \sum_{t=1}^{T} I(\psi_t \in J_l, \hat{\psi}_t \in J_l, U_t = u, \mathcal{E}_t) + \sum_{t=1}^{T} I(\psi_t \in J_l, \hat{\psi}_t \notin J_l, U_t = u, \mathcal{E}_t) \\
&+ \sum_{t=1}^{T} I(\mathcal{E}_t^c) \\
&\leq \sum_{t=1}^{T} I(\hat{\psi}_t \in J_l, U_t = u, \mathcal{E}_t) + \sum_{t=1}^{T} I(\psi_t \in J_l, \hat{\psi}_t \notin J_l, \mathcal{E}_t) + \sum_{t=1}^{T} I(\mathcal{E}_t^c) \\
&\leq D_{1,1}(\epsilon, J_l, u) + D_{1,2}(\epsilon) + D_{2,1}(\epsilon) + D_{2,2}(\epsilon, J_l)
\end{aligned}
$$

$$+ \sum_{t=1}^{T} I(\mathcal{E}_t^c).$$

The result follows from taking the expectation of both sides. □

4.6.2 Bounding the Expected Number of Explorations

The following lemma bounds the number of explorations by time T.

Lemma IV.16.

$$E_{\psi_0,\alpha}^{\boldsymbol{P}}\left[\sum_{t=1}^{T} I(\mathcal{E}_t^c)\right] \leq \left(\sum_{k=1}^{K} |S^k|\right) L \log T(1 + T_{\max}), \tag{4.9}$$

where $T_{\max} = \max_{k\in\mathcal{K},i,j\in S^k} E[T_{ij}^k] + 1$ *and* T_{ij}^k *is the hitting time of state* j *of arm* k *starting from state* i *of arm* k*. Since all arms are ergodic* $E[T_{ij}^k]$ *is finite for all* $k \in \mathcal{K}, i, j \in S^k$*.*

Proof. Assume that state i of arm k is *under-sampled*, i.e., $C_i^k(t) < L \log t$. Since each arm is an ergodic Markov chain, if the agent keeps playing arm k, the expected number of time steps until a single transition out of state i of arm k is observed is at most $(1 + T_{\max})$. If by time T transitions out of state i of arm k is observed at least $L \log T$ times, for all states i of all arms k, then the agent will not explore at time T. Therefore there can be at most $\sum_{k=1}^{K} \sum_{i\in S^k} L \log T$ such transitions by time T that take place in an exploration step. In the worst-case each of these transitions takes $(1 + T_{\max})$ expected time steps. □

4.6.3 Bounding $E_{\psi_0,\alpha}^{\boldsymbol{P}}[D_1(\epsilon, J_l, u)]$ for a suboptimal action $u \notin O(J_l; \boldsymbol{P})$

We will first bound $E_{\psi_0,\alpha}^{\boldsymbol{P}}[D_{1,1}(\epsilon, J_l, u)]$ for any suboptimal u. Let

$$\underline{\Delta}(J_l; \boldsymbol{P}) := \min_{\psi\in J_l, u\notin O(J_l;\boldsymbol{P})} \Delta(\psi, u; \boldsymbol{P}).$$

By Corollary IV.11, $\underline{\Delta}(J_l; \boldsymbol{P}) > 0$ for all $l = 1, \ldots, A(\tau_{\text{tr}})$. Let

$$\underline{\Delta} := \min_{l=1,\ldots,A(\tau_{\text{tr}})} \underline{\Delta}(J_l; \boldsymbol{P}).$$

In the following lemma we show that when the transition probability estimates are accurate enough and the estimated solution to the AROE is close enough to the true solution, a suboptimal action cannot be chosen by the agent.

Lemma IV.17. *Let $\delta_e > 0$ be the greatest real number such that*

$$||\hat{\boldsymbol{P}}_t - \boldsymbol{P}||_1 < \delta_e \Rightarrow \left|\mathcal{L}(\psi, u, h_{\boldsymbol{P}}, \boldsymbol{P}) - \mathcal{L}(\psi, u, h_{\boldsymbol{P}}, \hat{\boldsymbol{P}}_t)\right| \leq \underline{\Delta}/4,$$

for all $\psi \in \boldsymbol{\Psi}$. Such δ_e exists because $T_{\boldsymbol{P}}(\psi, y, u)$ is continuous in $\boldsymbol{P}$, and $h_{\boldsymbol{P}}(\psi)$ is continuous in ψ. Then, for an agent using AREP with $L \geq C_{\boldsymbol{P}}(\delta_e/(KS_{\max}^2))$, for any suboptimal action $u \notin O(J_l; \boldsymbol{P})$, we have

$$E_{\psi_0,\alpha}^{\boldsymbol{P}}[D_{1,1}(\epsilon, J_l, u)] \leq 2KS_{\max}^2(S_{\max} + 1)\beta,$$

for $\epsilon < \underline{\Delta}/4$, where $\beta = \sum_{t=1}^{\infty} 1/t^2$ and $C_{\boldsymbol{P}}(.)$ is the constant given in Lemma IV.13.

Proof. See Appendix G. □

Next, we consider bounding $E_{\psi_0,\alpha}^{\boldsymbol{P}}[D_{1,2}(\epsilon)]$. To do this we introduce the following lemma which implies that when the estimated transition probabilities get closer to the true transition probabilities, the difference between the functions which are solutions to the AROE based on the estimated and true transition probabilities diminishes.

Lemma IV.18. *For any $\epsilon > 0$, there exists $\varsigma > 0$ depending on ϵ such that if $\left\|P^k - \hat{P}^k\right\|_1 < \varsigma, \forall k \in \mathcal{K}$ then $\left\|h_{\boldsymbol{P}} - h_{\hat{\boldsymbol{P}}}\right\|_\infty < \epsilon$.*

Proof. See Appendix H. □

The following lemma bounds $E^{\boldsymbol{P}}_{\psi_0,\alpha}[D_{1,2}(\epsilon)]$.

Lemma IV.19. *For any $\epsilon > 0$, let $\varsigma > 0$ be such that Lemma IV.18 holds. Then for an agent using AREP with $L \geq C_{\boldsymbol{P}}(\varsigma/S^2_{\max})$, we have*

$$E^{\boldsymbol{P}}_{\psi_0,\alpha}[D_{1,2}(\epsilon)] \leq 2KS^2_{\max}(S_{\max}+1)\beta \ , \tag{4.10}$$

where $C_{\boldsymbol{P}}(.)$ is the constant given in Lemma IV.13.

Proof. See Appendix I. □

4.6.4 Bounding $E^{\boldsymbol{P}}_{\psi_0,\alpha}[D_2(\epsilon, J_l)]$

Lemma IV.20. *For an agent using AREP with exploration constant*

$$L \geq C_{\boldsymbol{P}}(\epsilon/(KS^2_{\max}|S^1|\dots|S^K|C_1(\boldsymbol{P}))),$$

we have

$$E^{\boldsymbol{P}}_{\psi_0,\alpha}[D_{2,1}(\epsilon)] \leq 2KS^2_{\max}(S_{\max}+1)\beta, \tag{4.11}$$

where $C_{\boldsymbol{P}}(.)$ is the constant given in Lemma IV.13, $C_1(\boldsymbol{P}) = \max_{k\in\mathcal{K}} C_1(P^k, \infty)$ and $C_1(P^k, t)$ is a constant that can be found in Lemma A.6.

Proof. See Appendix J. □

Next we will bound $E^{\boldsymbol{P}}_{\psi_0,\alpha}[D_{2,2}(\epsilon, J_l)]$.

Lemma IV.21. *Let τ_{tr} be such that Assumption IV.10 holds. Then for $\epsilon < \bar{\epsilon}/2$, where $\bar{\epsilon}$ is given in Corollary IV.11, $E^{\boldsymbol{P}}_{\psi_0,\alpha}[D_{2,2}(\epsilon, J_l)] = 0, l = 1, \dots, A(\tau_{tr})$.*

Proof. By Corollary IV.11, any $\psi_t \in J_l$ is at least $\bar{\epsilon}$ away from the boundary of J_l. Thus given $\hat{\psi}_t$ is at most ϵ away from ψ_t, it is at least $\bar{\epsilon}/2$ away from the boundary of J_l. □

4.6.5 A Logarithmic Strong Regret Upper Bound

Theorem IV.22. *Assume that Assumptions IV.1 and IV.10 are true. Let τ_{tr} be the minimum threshold, and $\epsilon = \bar{\epsilon}$ be the number given in Corollary IV.11. Under these assumptions, for an agent using AREP with L sufficiently large (depending on $\boldsymbol{P}$ and ϵ), for any arm (action) $u \in U$ which is suboptimal for the belief vectors in J_l, we have*

$$E^{\boldsymbol{P}}_{\psi_0,\alpha}[N_T(J_l,u)] \leq \left(\sum_{k=1}^{K}|S^k|\right) L\log T(1+T_{\max}) + 6KS^2_{\max}(S_{\max}+1)\beta \ ,$$

for some $\delta > 0$ depending on L. Therefore,

$$R^{\alpha}_{\psi_0;\boldsymbol{P}}(T) \leq \left(\left(\sum_{k=1}^{K}|S^k|\right) L\log T(1+T_{\max}) + 6KS^2_{\max}(S_{\max}+1)\beta\right) \times \sum_{l=1}^{A(\tau_{tr})} \sum_{u\notin O(J_l;\boldsymbol{P})} \bar{\Delta}(J_l,u;\boldsymbol{P}) + \bar{C}_{\boldsymbol{P}}.$$

When the arm rewards are in $[0,1]$, strong regret at time T given as $R^{\alpha}_{\psi_0;\boldsymbol{P}}(T)$ can also be upper bounded by

$$\left(\left(\sum_{k=1}^{K}|S^k|\right) L\log T(1+T_{\max}) + 6KS^2_{\max}(S_{\max}+1)\beta\right)(KA(\tau_{tr})) + \bar{C}_{\boldsymbol{P}} \ .$$

Proof. The result follows from Lemmas IV.14, IV.16, IV.19 IV.20 and IV.21. □

Remark IV.23. Our regret bound depends on $A(\tau_{\text{tr}})$. However, the agent does not need to know the value of τ_{tr} for which Corollary IV.11 is true. It only needs to choose L large enough so that the number of exploration steps is sufficient to ensure a bounded number of errors in exploitation steps.

4.7 AREP with an Adaptive Exploration Function

In this section, we consider an adaptive exploration function for AREP, by which the agent can achieve near-logarithmic regret without knowing how large it should chose the exploration constant L, which depends on $\boldsymbol{P}$. We note that the analysis in Section 4.6 holds when AREP is run with a sufficiently large exploration constant L such that at each exploitation step the estimated transition probabilities $\hat{\boldsymbol{P}}$ is close enough to $\boldsymbol{P}$ to guarantee that all regret terms in (4.8) is finite except the regret due to explorations which is logarithmic in time. In other words, there is an $C(\boldsymbol{P}) > 0$ such that when AREP is run with $L \geq C(\boldsymbol{P})$, at the end of each exploration step we have $||\hat{\boldsymbol{P}} - \boldsymbol{P}||_1 \leq \delta(\boldsymbol{P})$, where $\delta(\boldsymbol{P}) > 0$ is a constant for which Theorem IV.22 holds.

However, in our learning model we assumed that the agent does not know the transition probabilities initially, therefore it is impossible for the agent to check if $L \geq C(\boldsymbol{P})$. Let $\tilde{\boldsymbol{\Xi}} \subset \boldsymbol{\Xi}$ be the set of transition probability matrices where $\boldsymbol{P}$ lies in. If the agent knows $\tilde{\boldsymbol{\Xi}}$, then it can compute $\tilde{C} = \sup_{\tilde{\boldsymbol{P}} \in \tilde{\boldsymbol{\Xi}}} C(\tilde{\boldsymbol{P}})$, and choose $L > \tilde{C}$.

In this section, we present another exploration function for AREP such that the agent can achieve near-logarithmic regret even without knowing $C(\boldsymbol{P})$ or $\tilde{C}$. Let $f(t) = L(t) \log t$ where $L(t)$ is an increasing function such that $L(1) = 1$ and $\lim_{t\to\infty} L(t) = \infty$. The intuition behind this exploration function is that after some time T_0, $L(t)$ will be large enough so that the estimated transition probabilities are sufficiently accurate, and the regret due to incorrect calculations is a constant independent of time.

Theorem IV.24. *When $\boldsymbol{P}$ is such that Assumptions IV.1 and IV.10 hold, if the agent uses AREP with $f(t) = L(t) \log t$, for some increasing $L(t)$ such that $L(1) = 1$ and $\lim_{t\to\infty} L(t) = \infty$, then there exists $\tau_{tr}(\boldsymbol{P}) > 0$, $T_0(L, \boldsymbol{P}) > 0$ such that the strong*

regret is upper bounded by

$$
\begin{aligned}
R^{\alpha}_{\psi_0;\boldsymbol{P}}(T) \leq r_{\max} & \left(T_0(L,\boldsymbol{P}) + \left(\sum_{k=1}^{K} |S^k| \right) L(T) \log T (1+T_{\max}) \right. \\
& \left. +6KS^2_{\max}(S_{\max}+1)\beta \left(\sum_{l=1}^{A(\tau_{tr})} \sum_{u \notin O(J_l;\boldsymbol{P})} \bar{\Delta}(J_l,u;\boldsymbol{P}) \right) \right) \\
\leq r_{\max} & \left(T_0(L,\boldsymbol{P}) + \left(\sum_{k=1}^{K} |S^k| \right) L(T) \log T (1+T_{\max}) \right. \\
& \left. +6KS^2_{\max}(S_{\max}+1)\beta(\tau_{tr})^M \left(\sum_{k=1}^{K} |S^k| \right) \max_{l \in \{1,\ldots,A(\tau_{tr})\}} \bar{\Delta}(J_l,u;\boldsymbol{P}) \right) + \bar{C}_{\boldsymbol{P}}
\end{aligned}
$$

Proof. The regret up to $T_0(L,\boldsymbol{P})$ can be at most $r_{\max}T_0(L,\boldsymbol{P})$. After $T_0(L,\boldsymbol{P})$, since $L(t) \geq C(\boldsymbol{P})$, transition probabilities at exploitation steps sufficiently accurate so that all regret terms in (4.8) except the regret due to explorations is finite. Since time t is an exploration step whenever $C_i^k(t) < L(t)\log t$, the regret due to explorations is at most

$$
r_{\max} \left(\sum_{k=1}^{K} |S^k| \right) L(T) \log T (1+T_{\max}) \ .
$$

□

Remark IV.25. There is a tradeoff between choosing a rapidly increasing L or a slowly increasing L. The regret of AREP up to time $T_0(L,\boldsymbol{P})$ is linear. Since $T_0(L,\boldsymbol{P})$ is decreasing in function L, a rapidly increasing L will have better performance when the considered time horizon is small. However, in terms of asymptotic performance, i.e., as $T \to \infty$, L should be a slowly diverging sequence. For example if $L = \log(\log t)$, then the asymptotic regret will be $O(\log(\log t)\log t)$.

4.8 AREP with Finite Partitions

In this section we present a modified version of AREP, and prove that it can achieve logarithmic regret without Assumption IV.9 if the agent knows the time horizon T. We call this variant AREP with finite partitions (AREP-FP).

Basically, AREP-FP takes as input the threshold/mixing time τ_{tr}, then forms $\mathcal{G}_{\tau_{\text{tr}}}$ partition of the set of information states $\mathcal{C}$. At each exploitation step (at time t) AREP-FP solves the estimated AROE based on the transition probability estimate $\hat{\boldsymbol{P}}_t$, and if the belief $(\boldsymbol{s}_t, \boldsymbol{\tau}_t)$ is in G_l, the agent arbitrarily picks an arm in $O^*(G_l; \hat{\boldsymbol{P}}_t)$, instead of picking an arm in $O(\psi_{\hat{\boldsymbol{P}}_t}((\boldsymbol{s}_t, \boldsymbol{\tau}_t)); \hat{\boldsymbol{P}}_t)$.

When the arm in $O^*(G_l; \hat{\boldsymbol{P}}_t)$ selected by the agent is indeed in $O(\psi_{\boldsymbol{P}}((\boldsymbol{s}_t, \boldsymbol{\tau}_t)); \boldsymbol{P})$, then it ends up playing optimally at that time step. Else if the selected arm is in $O^*(G_l; \boldsymbol{P})$ but not in $O(\psi_{\boldsymbol{P}}((\boldsymbol{s}_t, \boldsymbol{\tau}_t)); \boldsymbol{P})$ such that $(\boldsymbol{s}_t, \boldsymbol{\tau}_t) \in G_l$, it plays near-optimally. Else, it plays suboptimally if the selected arm is neither in $O(\psi_{\boldsymbol{P}}((\boldsymbol{s}_t, \boldsymbol{\tau}_t)); \boldsymbol{P})$ nor in $O^*(G_l; \boldsymbol{P})$. By Lemma IV.7 we know that when τ_{tr} is chosen large enough, for any $G_l \in \mathcal{G}_{\tau_{\text{tr}}}$, and $(\boldsymbol{s}, \boldsymbol{\tau})$, $O(\psi_{\boldsymbol{P}}((\boldsymbol{s}, \boldsymbol{\tau})); \boldsymbol{P})$ is a subset of $O^*(G_l; \boldsymbol{P})$. Since the solution to the AROE is a continuous function, by choosing a large enough τ_{tr}, we can control the regret due to near-optimal actions. The regret due to suboptimal actions can be bounded the same way as in Theorem IV.22. The following theorem gives a logarithmic upper bound on the regret of AREP-FP.

Theorem IV.26. *When the true set of transition probabilities $\boldsymbol{P}$ is such that Assumption IV.1 is true, for an agent using AREP-FP with exploration constant L, and mixing time τ_{tr} sufficiently large such that for any $(\boldsymbol{s}, \boldsymbol{\tau}) \in G_l$, $G_l \in \mathcal{G}_{\tau_{tr}}$, we have $|h_{\boldsymbol{P}}(\psi_{\boldsymbol{P}}((\boldsymbol{s}, \boldsymbol{\tau}))) - h_{\boldsymbol{P}}(\psi^*(G_l; \boldsymbol{P}))| < C/2T$, where $C > 0$ is a constant and T is the time horizon, the regret of AREP-FP is upper bounded by*

$$C + \left(L \log T(1 + T_{\max}) + 6KS_{\max}^2 \beta(S_{\max} + 1)\right) \times \sum_{l=1}^{A(\tau_{tr})} \sum_{u \notin O(J_l; \boldsymbol{P})} \bar{\Delta}(J_l, u; \boldsymbol{P}) + \bar{C}_{\boldsymbol{P}},$$

for some $\delta > 0$ which depends on L and τ_{tr}.

Proof. The regret at time T is upper bounded by Lemma IV.14. Consider any t which is an exploitation step. Let l be such that $(\boldsymbol{s}_t, \boldsymbol{\tau}_t) \in G_l$. If the selected arm $\alpha(t) \in O(\psi_{\boldsymbol{P}}((\boldsymbol{s}_t, \boldsymbol{\tau}_t)); \boldsymbol{P})$, then an optimal decision is made at t, so the contribution to regret in time step t is zero. Next, we consider the case when $\alpha(t) \notin O(\psi_{\boldsymbol{P}}((\boldsymbol{s}_t, \boldsymbol{\tau}_t)); \boldsymbol{P})$. In this case there are two possibilities: either $\alpha(t) \in O^*(G_l; \boldsymbol{P})$ or not. We know that when $O^*(G_l; \hat{\boldsymbol{P}}_t) \subset O^*(G_l; \boldsymbol{P})$ we have $\alpha(t) \in O^*(G_l; \boldsymbol{P})$. Since $|h_{\boldsymbol{P}}(\psi_{\boldsymbol{P}}((\boldsymbol{s}, \boldsymbol{\tau}))) - h_{\boldsymbol{P}}(\psi^*(G_l; \boldsymbol{P}))| < C/2T$ for all $(\boldsymbol{s}, \boldsymbol{\tau}) \in G_l$, we have by (4.6),

$$\Delta(\psi_t, \alpha(t); \boldsymbol{P}) = \mathcal{L}^*(\psi_t, \boldsymbol{P}) - \mathcal{L}(\psi_t, \alpha(t), h_{\boldsymbol{P}}, \boldsymbol{P}) \leq C/T. \tag{4.12}$$

Therefore, contribution of a near-optimal action to regret is at most C/T.

Finally, consider the case when $\alpha(t) \notin O^*(G_l; \boldsymbol{P})$. This implies that either the estimated belief $\hat{\psi}_t$ is not close enough to ψ_t or the estimated solution to the AROE, i.e., $\hat{h}_t$, is not close enough to $h_{\boldsymbol{P}}$. Due to the non-vanishing suboptimality gap at any belief vector $\psi^*(G_l; \boldsymbol{P})$, and since decisions of AREP-FP is only based on belief vectors corresponding to $(\boldsymbol{s}, \boldsymbol{\tau}) \in \mathcal{C}$, the regret due to suboptimal actions can be bounded by Theorem IV.22. We get the regret bound by combining all these results. □

Note that the regret bound in Theorem IV.26 depends on τ_{tr} which depends on T since τ_{tr} should be chosen large enough so that for every G_l in the partition created by τ_{tr}, function $h_{\boldsymbol{P}}$ should vary by at most $C/2T$. Clearly since $h_{\boldsymbol{P}}$ is a continuous function, the variation of $h_{\boldsymbol{P}}$ on G_l, i.e., the difference between the maximum and minimum values of $h_{\boldsymbol{P}}$ on G_l, decreases with the diameter of G_l on the belief space. Note that there is a term in regret that depends linearly on the number of sets $A(\tau_{\text{tr}})$ in the partition generated by τ_{tr}, and $A(\tau_{\text{tr}})$ increases proportional to $(\tau_{\text{tr}})^K$. This tradeoff is not taken into account in Theorem IV.26. For example, if $(\tau_{\text{tr}})^K \geq T$ then the regret bound in Theorem IV.26 is useless. Another approach is to jointly

optimize the regret due to suboptimal and near-optimal actions by balancing the number of sets $A(\tau_{\text{tr}})$ and the variation of $h_{\boldsymbol{P}}$ on sets in partition $\mathcal{G}_{\tau_{\text{tr}}}$. For example, given $0 < \theta \leq 1$, we can find a $\tau_{\text{tr}}(\theta)$ such that for any $(\boldsymbol{s}, \boldsymbol{\tau}) \in G_l$, $G_l \in \mathcal{G}_{\tau_{\text{tr}}(\theta)}$, and $C > 0$, we have

$$|h_{\boldsymbol{P}}(\psi_{\boldsymbol{P}}((\boldsymbol{s}, \boldsymbol{\tau}))) - h_{\boldsymbol{P}}(\psi^*(G_l; \boldsymbol{P}))| < \frac{C}{2T^{\theta}} \ .$$

Then, the regret due to near-optimal decisions will be proportional to $CT^{1-\theta}$, and the regret due to suboptimal decision will be proportional to $(\tau_{\text{tr}}(\theta))^K$. Let

$$C = \sup_{\psi \in \boldsymbol{\Psi}} h_{\boldsymbol{P}}(\psi) - \inf_{\psi \in \boldsymbol{\Psi}} h_{\boldsymbol{P}}(\psi) \ .$$

Since $T^{1-\theta}$ is decreasing in θ and $\tau_{\text{tr}}(\theta)$ is increasing in θ, there exists $\theta \in [0, 1]$, such that

$$\theta = \arg\min_{\theta' \in [0,1]} |T^{1-\theta} - (\tau_{\text{tr}}(\theta))^K| \ ,$$

and

$$|T^{1-\theta} - (\tau_{\text{tr}}(\theta))^K| \leq (\tau_{\text{tr}}(\theta) + 1)^K - (\tau_{\text{tr}}(\theta))^K \ .$$

If the optimal value of θ is in $(0, 1)$, then given θ, the agent can balance the tradeoff, and achieve sublinear regret proportional to $T^{1-\theta}$. However, since the agent does not know $\boldsymbol{P}$ initially, it may not know the optimal value of θ. Online learning algorithms for the agent to estimate the optimal value of θ is a future research direction.

CHAPTER V

Single-agent Feedback Bandit with Approximate Optimality

In this chapter we study a special case of the single-agent uncontrolled restless bandit problem which is called the *feedback bandit* problem. In a feedback bandit problem, each arm has two states; a bad state which yields zero reward when played, and a good state which yields positive reward when played. In the previous chapters we considered algorithms with logarithmic weak regret which are very simple to implement, and algorithms with logarithmic strong regret which are computationally intractable. However, in general optimality in terms of the weak regret does not provide any information about how well the learning algorithm performs compared to the best policy. This chapter's goal is to explore algorithms that maximize the performance of the agent without sacrificing computational tractability. The negative result in *Papadimitriou and Tsitsiklis* (1999) motivates us to focus on special cases. Especially, the special structure of the feedback bandit problem allows us to develop a computationally tractable learning algorithm that is approximately optimal in terms of the average reward. As opposed to the learning algorithms in Chapter IV whose complexity increases exponentially with the number of arms, this algorithm's complexity increases linearly with the number of arms, and it requires polynomial number of operations to select an arm at each time step. Therefore, it can be used

in practical applications that can be modeled as a feedback bandit, such as target tracking systems and dynamic spectrum access.

The organization of this chapter is as follows. Problem definition and notations are given in Section 5.1. Feedback bandit problem with single play is investigated, and an approximately optimal algorithm is proposed in Section 5.2. Discussion is given in Section 5.3.

5.1 Problem Formulation and Preliminaries

In this chapter we study a special case of the uncontrolled restless Markovian model under the approximate optimality criterion. Consider K mutually independent uncontrolled restless Markovian arms described in Definition I.4 with the additional property that $S^k = \{g, b\}$, $\forall k \in \mathcal{K}$, and $r_g^k = r^k > 0$, $r_b^k = 0$. Thus, each arm has two states, a *good* state g which yields some positive reward, and a *bad* state b which yields zero reward. This definition is called *the feedback bandit* in *Guha et al.* (2010), which solves the optimization version of this problem. The arms are *bursty*, i.e., $p_{gb}^k + p_{bg}^k < 1, \forall k \in \mathcal{K}$, and $p_{gb}^k > \delta$, $\forall k \in \mathcal{K}$ for some $\delta > 0$. If arm k is played τ steps ago and the last observed state is $s \in S^k$, let (s, τ) be *the information state* for that arm. Let v_τ^k be the probability that arm k will be in state g given that it is observed τ steps ago in state b, and u_τ^k be the probability that arm k will be in state g given that it is observed τ steps ago in state g. These probabilities can be written in terms of the state transition probabilities as

$$v_\tau^k = \frac{p_{bg}^k}{p_{bg}^k + p_{gb}^k}(1 - (1 - p_{bg}^k - p_{gb}^k)^\tau), \; u_\tau^k = \frac{p_{bg}^k}{p_{bg}^k + p_{gb}^k} + \frac{p_{gb}^k}{p_{bg}^k + p_{gb}^k}(1 - p_{bg}^k - p_{gb}^k)^\tau,$$

and by the burstiness assumption ($p_{gb}^k + p_{bg}^k < 1, \forall k \in \mathcal{K}$), v_τ^k, $1 - u_\tau^k$ are monotonically increasing concave functions of τ. $\mathbb{Z}_+$ denotes the set of non-negative integers, and $I(.)$ is the indicator function.

There is an agent whose goal is to maximize its infinite horizon average reward by playing at most one of the arms at each time step. We assume that there is a *dummy* arm which yields no reward, so the agent has the option to not play by selecting this arm. The agent initially does not know the transition probability matrices $P^k, k \in \mathcal{K}$, but knows the bound δ on p^k_{gb}. Without loss of generality, we assume that the agent knows the rewards of bad and good states, thus observing the reward of an arm is equivalent to observing the state of the arm from the agent's perspective. Let α be an admissible algorithm for the agent. We represent the expectation with respect α when the transition probability matrices are $\boldsymbol{P} = (P^1, P^2, \ldots, P^K)$ and initial state of the agent is ψ_0 by $E^{\boldsymbol{P}}_{\psi_0,\alpha}[.]$. Many subsequent expressions depend on the algorithm α used by the agent, but we will explicitly state this dependence only when it is not clear from the context.

Let $\alpha(t)$ denote the arm selected by the agent at time t, when it uses algorithm α. We define a *continuous play* of arm k starting at time t with state s as a pair of plays in which arm k is selected at times t and $t+1$ and state s is observed at time t. Let

$$N^k_T(s,s') = \sum_{t=1}^{T-1} I(\alpha(t) = \alpha(t+1) = k, X^k_t = s, X^k_{t+1} = s') \ ,$$

be the number of times transition from s to s' is observed in continuous plays of arm k up to time T. Let

$$C^k_T(s) = \sum_{s' \in \{g,b\}} N^k_T(s,s') \ ,$$

be the number of continuous plays of arm k starting with state s up to time T. These quantities will be used to estimate the state transition probabilities.

5.2 Algorithm and Analysis

5.2.1 Guha's Policy

For the optimization version of the problem we consider, where P^k's are known by the agent, *Guha et al.* (2010) propose a $(2+\epsilon)$ approximate policy for the infinite horizon average reward problem. Under this approach, Whittle's LP relaxation in *Whittle* (1988) is used, where the constraint that exactly one arm is played at each time step is replaced by an average constraint that on average one arm is played. The value (average reward) of the optimal solution to Whittle's LP is denoted by OPT. In *Guha et al.* (2010) it is shown that OPT is at least the average reward of the optimal policy in the original problem, since Whittle's LP is a relaxation. Then, the arms are decoupled by considering the Lagrangian of Whittle's LP. Thus instead of solving the original problem which has a size exponential in K, K individual optimization problems are solved, one for each arm. The Lagrange multiplier $\lambda > 0$ is treated as penalty per play and it was shown that the optimal single arm policy has the structure of the policy $\mathcal{P}_\tau^k$ given in Figure 5.1: whenever an arm is played and a good state is observed, it will also be played in the next time; if a bad state is observed then the agent will wait $\tau - 1$ time steps before playing that arm again. Thus, τ is called the *waiting time*. Let R_τ^k and Q_τ^k be the average reward and rate of play for policy $\mathcal{P}_\tau^k$ respectively. Q_τ^k is defined as the average number of times arm k will be played under a single arm policy with waiting time τ. Then from Lemma A.1 of *Guha et al.* (2010) we have

$$R_\tau^k = \frac{r^k v_\tau^k}{v_\tau^k + \tau p_{gb}^k},\ Q_\tau^k = \frac{v_\tau^k + p_{gb}^k}{v_\tau^k + \tau p_{gb}^k}.$$

Then, if playing arm k is penalized by λ, the gain of $\mathcal{P}_\tau^k$, i.e., the average reward subtracted by penalty times the rate of play is

$$F_{\lambda,\tau}^k = R_\tau^k - \lambda Q_\tau^k \ . \tag{5.1}$$

The optimal single arm policy for arm k under penalty λ is thus $\mathcal{P}_{\tau^k(\lambda)}^k$, where

$$\tau^k(\lambda) = \min \arg\max_{\tau \geq 1} F_{\lambda,\tau}^k \ ,$$

and the optimal gain, i.e, the gain under the optimal waiting time is

$$H_\lambda^k = \max_{\tau \geq 1} F_{\lambda,\tau}^k.$$

H_λ^k is a non-increasing function of λ by Lemma 2.6 in *Guha et al.* (2010). Let $G_\lambda = \sum_{k=1}^K H_\lambda^k$ be the sum of the gains of the optimal single arm policies. In *Guha et al.* (2010) the algorithm in Figure 5.2 is proposed, and it is shown that the infinite horizon average reward of this algorithm is at least $OPT/(2+\epsilon)$, where $\epsilon > 0$ is the performance parameter given as an input by the agent which we will refer to as the *stepsize*. The instantaneous and the long term average reward are balanced by viewing λ as an amortized reward per play and H_λ^k as the per step reward. This balancing procedure is given in Figure 5.3. After computing the balanced λ, the optimal single arm policy for this λ is combined with the priority scheme in Figure 5.2 so that at all times at most one arm is played. Denote the gain and the waiting time for the optimal arm k policy at the balanced λ by H^k and τ^k.

Note that it is required that at any t one and only one arm must be in good state in Guha's policy. This can be satisfied by initially sampling from $K-1$ arms until a bad state is observed and sampling from the last arm until a good state is observed. Such an initialization will not change the infinite horizon average reward,

so we assume that such an initialization is completed before the agent starts using its learning algorithm.

At time t: 1. If arm k is just observed in state g, also play arm k at $t+1$. 2. If arm k is just observed in state b, wait $\tau - 1$ steps, and then play arm k.

Figure 5.1: policy $\mathcal{P}_\tau^k$

Choose a balanced λ by the procedure in Figure 5.3. Let $\mathcal{Q} = \{k : H_\lambda^k > 0\}$, $\tau^k = \tau^k(\lambda)$. Only play the arms in $\mathcal{Q}$ according to the following priority scheme: At time t: 1. Exploit: If $\exists k \in \mathcal{Q}$ in state $(g, 1)$, play arm k. 2. Explore: If $\exists k \in \mathcal{Q}$ in state $(b, \tau) : \tau \geq \tau^k$, play arm k. 3. Idle: If 1 and 2 do not hold do not play any arm.

Figure 5.2: Guha's policy

Input: ϵ. Perform binary search to find the balanced $\lambda = \lambda(\epsilon)$: 1. Start with $\lambda = \sum_{k=1}^K r^k$, Calculate $G_\lambda = \sum_{k=1}^K H_\lambda^k$. 2. While $\lambda > G_\lambda$ 2.1 $\lambda = \lambda/(1+\epsilon)$, 2.2 Calculate G_λ. 3. Output $\lambda, \tau^k, k \in \mathcal{K}$

Figure 5.3: procedure for the balanced choice of λ

5.2.2 A Threshold Policy

In this section we consider a threshold variant of Guha's policy, called the ϵ_1-threshold policy. The difference between the two is in balancing the Lagrange multiplier λ. Pseudocode of this policy is shown in Figure 5.4. Let $\tilde{H}_\lambda^k$, $\tilde{\tau}_\lambda^k$ denote the optimal gain and the optimal waiting time for arm k calculated by the ϵ_1-threshold policy when the penalty per play is λ. For any λ if the optimal single arm policy for arm k has gain $H_\lambda^k < \epsilon_1$, that arm is considered not worth playing and $\tilde{H}_\lambda^k = 0, \tilde{\tau}_\lambda^k = \infty$. For any λ and any arm k with the optimal gain greater than or equal to ϵ_1, the optimal

waiting time after a bad state and the optimal gain are the same as Guha's policy. We denote the stepsize used by the ϵ_1-threshold policy by ϵ_2, which is also an input to the policy by the agent.

ϵ_1-threshold policy

1: Input: ϵ_1, ϵ_2
2: Initialize: $\lambda = \sum_{k=1}^{K} r^k$.
3: Compute $H_\lambda^k, \tau_\lambda^k, \forall k \in \mathcal{K}$.
4: **for** $k = 1, 2, \ldots, K$ **do**
5: **if** $H_\lambda^k < \epsilon_1$ **then**
6: Set $\tilde{H}_\lambda^k = 0, \tilde{\tau}_\lambda^k = \infty$.
7: **else**
8: Set $\tilde{H}_\lambda^k = H_\lambda^k, \tilde{\tau}_\lambda^k = \tau_\lambda^k$.
9: **end if**
10: **end for**
11: $\tilde{G}_\lambda = \sum_{k=1}^{K} \tilde{H}_\lambda^k$.
12: **if** $\lambda < \tilde{G}_\lambda$ **then**
13: Play Guha's policy with $\tau^1 = \tilde{\tau}_\lambda^1, \ldots, \tau^K = \tilde{\tau}_\lambda^K$.
14: **else**
15: $\lambda = \lambda/(1+\epsilon_2)$. Return to Step 3
16: **end if**

Figure 5.4: pseudocode for the ϵ_1-threshold policy

Note that at any λ, any arm k which will be played by the ϵ_1-threshold policy will also be played by Guha's policy with $\tau_\lambda^k = \tilde{\tau}_\lambda^k$. Arm k with $H_\lambda^k < \epsilon_1$ in Guha's policy will not be played by the ϵ_1-threshold policy. The following Lemma states that the average reward of an ϵ_1-threshold policy cannot be much less than $OPT/2$.

Lemma V.1. *Consider the ϵ_1-threshold policy shown in Figure 5.4 with step size ϵ_2. The average reward of this policy is at least*

$$\frac{OPT}{2(1+\epsilon_2)} - K\epsilon_1 \ .$$

Proof. Let λ^* be the balanced Lagrange multiplier computed by Guha's policy with

an input of ϵ_2. Then from Figure 5.3 we have,

$$\lambda^* < \sum_{k=1}^{K} H_{\lambda^*}^k \leq (1+\epsilon_2)\lambda^* \ .$$

For any λ we have

$$\sum_{k=1}^{K} H_{\lambda}^k - K\epsilon_1 \leq \sum_{k=1}^{K} \tilde{H}_{\lambda}^k \leq \sum_{k=1}^{K} H_{\lambda}^k \ . \tag{5.2}$$

We consider two cases:

Case 1: $\lambda^* < \sum_{k=1}^{K} \tilde{H}_{\lambda^*}^k$. Then, λ^* is also the balanced Lagrange multiplier computed by the ϵ_1-threshold policy.

Case 2: $\lambda^* \geq \sum_{k=1}^{K} \tilde{H}_{\lambda^*}^k$. Then, ϵ_1-threshold policy will continue the process of decreasing λ and recomputing $\tilde{G}_{\lambda}$ until it reaches some λ' such that

$$\lambda' < \sum_{k=1}^{K} \tilde{H}_{\lambda'}^k \leq (1+\epsilon_2)\lambda' \ .$$

Since $\tilde{H}_{\lambda}^k$ is non-increasing in λ we have

$$\sum_{k=1}^{K} \tilde{H}_{\lambda'}^k \geq \sum_{k=1}^{K} \tilde{H}_{\lambda^*}^k \ .$$

Thus by (5.2),

$$(1+\epsilon_2)\lambda' \geq \sum_{k=1}^{K} \tilde{H}_{\lambda^*}^k \geq \sum_{k=1}^{K} H_{\lambda^*}^k - K\epsilon_1 \ .$$

By Guha's policy $\sum_{k=1}^{K} H_{\lambda^*}^k \geq OPT/2$. Therefore,

$$\sum_{k=1}^{K} \tilde{H}_{\lambda'}^k \geq OPT/2 - K\epsilon_1, \ \lambda' \geq OPT/(2(1+\epsilon_2)) - K\epsilon_1 \ .$$

The result follows from Theorem 2.7 of *Guha et al.* (2010). □

The following lemma shows that computing $\tilde{\tau}^k$ for the ϵ_1-threshold policy can be done by considering waiting times in a finite window.

Lemma V.2. *For any λ, in order to compute $\tilde{\tau}^k, k \in \mathcal{K}$, the ϵ_1-threshold policy only requires to evaluate $F^k_{\lambda,\tau}$ for $\tau \in [1, \tau^*(\epsilon_1)]$, where $\tau^*(\epsilon_1) = \lceil r_{\max}/(\delta\epsilon_1) \rceil$.*

Proof. For any λ, $F^k_{\lambda,\tau} \leq R^k_\tau$. For $\tau \geq \tau^*(\epsilon_1)$,

$$R^k_\tau = r^k \frac{v^k_\tau}{v^k_\tau + \tau p^k_{gb}} \leq \frac{r_{\max}}{\tau p^k_{gb}} \leq \frac{r_{\max}}{\delta\tau} \ .$$

□

The following lemma shows that the procedure of decreasing λ can only repeat a finite number of times.

Lemma V.3. *Assume that there exists an arm k such that for some $\lambda > 0$, $\tilde{H}^k_\lambda \geq \epsilon_1$. Otherwise, no arm will be played by the ϵ_1-threshold policy. Let*

$$\hat{\lambda} = \sup\{\lambda : \tilde{H}^k_\lambda \geq \epsilon_1\} \ ,$$

$\lambda^* = \min\{\hat{\lambda}, \epsilon_1\}$. *Let $z(\epsilon_2)$ be the number of cycles, i.e., the number of times λ is decreased until the computation of $\tilde{\tau}^k, k \in \mathcal{K}$ is completed. We have*

$$z(\epsilon_2) \leq \min \left\{ z' \in \mathbb{Z}_+ \text{ such that } (1+\epsilon_2)^{z'} \geq \sum_{k=1}^K r^k/\lambda^* \right\} \ .$$

Proof. Since $\tilde{H}^k_\lambda$ is non-increasing in λ, $\tilde{H}^k_{\lambda^*} \geq \lambda^*$. The result follows from this observation. □

Let $\Theta(\epsilon_2) = \{\sum_{k=1}^K r^k, \sum_{k=1}^K r^k/(1+\epsilon_2), \ldots, \sum_{k=1}^K r^k/(1+\epsilon_2)^{z(\epsilon_2)}\}$ be the set of

values λ takes in $z(\epsilon_2)$ cycles, and

$$T_k(\lambda) = \arg\max_{\tau \geq 1} R_\tau^k - \lambda Q_\tau^k,$$
$$T_k'(\lambda) = \arg\max_{\tau \geq 1, \tau \notin T_k(\lambda)} R_\tau^k - \lambda Q_\tau^k \ ,$$

be the set of optimal waiting times, and best suboptimal waiting times under penalty λ respectively. Let

$$\delta(k,\lambda) = (R_{\tau^k}^k - \lambda Q_{\tau^k}^k) - (R_{\tau'^k}^k - \lambda Q_{\tau'^k}^k),\ \tau^k \in T_k(\lambda), \tau'^k \in T_k'(\lambda),$$

and $\delta_2 = \min_{k \in \mathcal{K}, \lambda \in \Theta(\epsilon_2)} \delta(k,\lambda)$. Consider a different set of transition probabilities $\hat{\boldsymbol{P}} = (\hat{P}^1, \ldots, \hat{P}^K)$. Let $\hat{R}_\tau^k$, $\hat{Q}_\tau^k$ and $\tilde{\hat{\tau}}_k$ denote the average reward, average number of plays and the optimal waiting time for arm k under ϵ_1-threshold policy and $\hat{\boldsymbol{P}}$ respectively.

Lemma V.4. *For* $\epsilon_3 = \delta_2 / \left(2(1 + \sum_{k=1}^K r^k)\right)$, *the event*

$$\left\{ |R_\tau^k - \hat{R}_\tau^k| < \epsilon_3, |Q_\tau^k - \hat{Q}_\tau^k| < \epsilon_3,\ \forall \tau \in [1, \tau^*(\epsilon_1)] \right\} \tag{5.3}$$

implies the event $\{\tilde{\tau}^k = \tilde{\hat{\tau}}^k, \forall k \in \mathcal{K}\}$.

Proof. By (5.3), for any $\lambda \in \Theta$, $\tau \in [1, \tau^*(\epsilon_1)]$,

$$\begin{aligned} |(R_\tau^k - \lambda Q_\tau^k) - (\hat{R}_\tau^k - \lambda \hat{Q}_\tau^k)| &\leq |R_\tau^k - \hat{R}_\tau^k| + \lambda |Q_\tau^k - \hat{Q}_\tau^k| \\ &\leq |R_\tau^k - \hat{R}_\tau^k| + \sum_{k=1}^K r^k |Q_\tau^k - \hat{Q}_\tau^k| \\ &< (1 + \sum_{k=1}^K r^k)\epsilon_3 = \frac{\delta_2}{2}. \end{aligned}$$

Thus, $\hat{F}_{\lambda,\tilde{\tau}^k}^k$ can be at most $\delta_2/2$ smaller than $F_{\lambda,\tilde{\tau}^k}^k$, while for any other $\tau \neq \tilde{\tau}^k$, $\hat{F}_{\lambda,\tau}^k$ can be at most $\delta_2/2$ larger than $F_{\lambda,\tau}^k$ for any λ. Thus the maximizers are the same

for all λ and the result follows. $\square$

The following lemma shows that $\tilde{\tau}^1, \ldots, \tilde{\tau}^K$ for the ϵ_1-threshold policy can be efficiently computed. We define a mathematical operation to be the computation of $R_\tau^k - \lambda Q_\tau^k$. We do not count other operations such as additions and multiplications.

Lemma V.5. *Finding the balanced λ and $\tilde{\tau}^1, \ldots, \tilde{\tau}^K$ requires at most*

$$K \lceil \log(z(\epsilon_2)) \rceil \tau^*(\epsilon_1)$$

mathematical operations.

Proof. Since $\tilde{G}_\lambda = \sum_{k=1}^K \tilde{H}_\lambda^k$ is decreasing in λ, the balanced λ can be computed by binary search. By Lemma V.3 the number of cycles required to find the optimal λ by binary search is $\lceil \log(z(\epsilon_2)) \rceil$. For each λ and each arm k, $\tilde{H}_\lambda^k$ and τ_λ^k can be calculated by at most $\tau^*(\epsilon_1)$ mathematical operations. $\square$

5.2.3 The Adaptive Balance Algorithm (ABA)

We propose the Adaptive Balance Algorithm (ABA) given in Figure 5.5 as a learning algorithm which is based on the ϵ_1-threshold policy instead of Guha's policy. This choice has several reasons. The first concerns the union bound we will use to relate the probability that the adaptive algorithm deviates from the ϵ_1-threshold policy with the probability of accurately calculating the average reward and the rate of play for the single arm policies given the estimated transition probabilities. In order to have finite number of terms in the union bound, we need to evaluate the gains $F_{\lambda,\tau}^k$ at finite number of waiting times τ. We do this by the choice of a finite time window $[1, \tau^*]$, for which we can bound our loss in terms of the average reward. Thus, the optimal single arm waiting times are computed by comparing $F_{\lambda,\tau}^k$'s in $[1, \tau^*]$. The second is due to the non-monotonic behavior of the gain $F_{\lambda,\tau}^k$ with respect to the waiting time τ. For example, there exists transition probabilities satisfying the

burstiness assumption such that the maximum of $F^k_{\lambda,\tau}$ occurs at $\tau > \tau^*$, while the second maximum is at $\tau = 1$. Then, by considering the time window $[1, \tau^*]$, it will not be possible to play with the same waiting times as in Guha's policy independent of how much we explore. The third is that for any $OPT/(2+\epsilon)$ optimal Guha's policy, there exists ϵ_1 and ϵ_2 such that the ϵ_1-threshold policy is $OPT/(2+\epsilon)$ optimal. Thus, any average reward that can be achieved by Guha's policy can also be achieved by the ϵ_1-threshold policy.

Let $\hat{p}^k_{bg,t}, \hat{p}^k_{gb,t}, k \in \mathcal{K}$, and $\hat{\boldsymbol{P}}_t = (\hat{P}^1_t, \ldots, \hat{P}^K_t)$ be the estimated transition probabilities and the estimated transition probability matrices at time t, respectively. We will use $\hat{\cdot}$ to represent the quantities computed according to $\hat{\boldsymbol{P}}_t$.

ABA consists of exploration and exploitation phases. Exploration serves the purpose of estimating the transition probabilities. If at time t the number of samples used to estimate the transition probability from state g or b of any arm is less than $L \log t$, for some $L > 0$ which we call the exploration constant, ABA explores to increase the accuracy of the estimated transition probabilities. In general it should be chosen large enough (depending on $\boldsymbol{P}, r^1, \ldots, r^K$) so that the regret bounds hold. We will describe an alternative way to choose L (independent of $\boldsymbol{P}, r^1, \ldots, r^K$) in Section 5.3. If all the transition probabilities are accurately estimated, then ABA exploits by using these probabilities in the ϵ_1-threshold policy to select an arm. Note that the transition probability estimates can also be updated after an exploitation step, depending on whether a continuous play of an arm occurred or not. In this section we denote ABA by α.

In the next section, we will show that the expected number of times in which ABA deviates from the ϵ_1-threshold policy given $\boldsymbol{P}$ is logarithmic in time.

Adaptive Balance Algorithm (ABA)

Input: ϵ_1, ϵ_2 $\tau^*(\epsilon_1)$, $L > 0$.
Initialize: Set $t = 1$, $N^k(i,j) = 0, C^k(i) = 0, \forall k \in \mathcal{K}, i, j \in S^k$. Play each arm once so the initial information state can be represented as an element of countable form $(s^1, \tau^1), \ldots, (s^K, \tau^K)$, where only one arm is observed in state g one step ago while all other arms are observed in state b, $\tau^k > 1$ steps ago.
while $t \geq 1$ **do**
  $\hat{p}^k_{gb} = \frac{1I(N^k(g,b)=0)+N^k(g,b)}{2I(C^k(g)=0)+C^k(g)}$,
  $\hat{p}^k_{bg} = \frac{1I(N^k(b,g)=0)+N^k(b,g)}{2I(C^k(b)=0)+C^k(b)}$,
  $W = \{(k,i), k \in \mathcal{K}, i \in S^k : C^k(i) < L \log t \}$.
  if $W \neq \emptyset$ **then**
    EXPLORE
    if $u(t-1) \in W$ **then**
      $\alpha(t) = u(t-1)$
    else
      select $\alpha(t) \in W$ arbitrarily.
    end if
  else
    EXPLOIT
    Start with $\lambda = \sum_{k=1}^K r^k$.
    Run the procedure for the balanced choice λ given by the ϵ_1-threshold policy with step size ϵ_2 and transition matrices $\hat{\boldsymbol{P}}_t$.
    Obtain $\hat{\tau}^1, \ldots, \hat{\tau}^K$.
    Play according to Guha's Policy with parameters $\hat{\tau}^1, \ldots, \hat{\tau}^K$ for only one time step.
  end if
  if $u(t-1) = \alpha(t)$ **then**
    for $i, j \in S^{\alpha(t)}$ **do**
      if State j is observed at t, state i is observed at $t-1$ **then**
        $N^{\alpha(t)}(i,j) = N^{\alpha(t)}(i,j) + 1$, $C^{\alpha(t)}(i) = C^{\alpha(t)}(i) + 1$.
      end if
    end for
  end if
  $t := t + 1$
end while

Figure 5.5: pseudocode for the Adaptive Balance Algorithm (ABA)

5.2.4 Number of Deviations of ABA from the ϵ_1-threshold policy

Let $\gamma^{\epsilon_1, \boldsymbol{P}}$ be the arm selection rule determined by the ϵ_1-threshold policy given ϵ_2 and $\boldsymbol{P} = (P^1, \ldots, P^K)$; and $\tilde{\tau}^1, \ldots, \tilde{\tau}^K$ be the waiting times after a bad state for

$\gamma^{\epsilon_1,\boldsymbol{P}}$. Let N_T be the number of times $\gamma^{\epsilon_1,\boldsymbol{P}}$ is not played up to T. Let E_t be the event that ABA exploits at time t. Then,

$$\begin{aligned}
N_T &\leq \sum_{t=1}^{T} I(\hat{\tau}^k(t) \neq \tilde{\tau}^k \text{ for some } k \in \mathcal{K}) \\
&\leq \sum_{t=1}^{T} I(\hat{\tau}^k(t) \neq \tilde{\tau}^k \text{ for some } k \in \mathcal{K}, E_t) + \sum_{t=1}^{T} I(E_t^C) \\
&\leq \sum_{k=1}^{K}\sum_{t=1}^{T} I(\hat{\tau}^k(t) \neq \tilde{\tau}^k, E_t) + \sum_{t=1}^{T} I(E_t^C) \\
&\leq \sum_{k=1}^{K}\sum_{t=1}^{T} I(|R_\tau^k - \hat{R}_\tau^k(t)| \geq \epsilon_3 \text{ or } |Q_\tau^k - \hat{Q}_\tau^k(t)| \geq \epsilon_3 \\
&\quad \text{for some } \tau \in [1, \tau^*(\epsilon_1)], E_t) + \sum_{t=1}^{T} I(E_t^C) \\
&\leq \sum_{k=1}^{K}\sum_{t=1}^{T}\sum_{\tau=1}^{\tau^*(\epsilon_1)} \Big(I(|R_\tau^k - \hat{R}_\tau^k(t)| \geq \epsilon_3, E_t) \\
&\quad + I(|Q_\tau^k - \hat{Q}_\tau^k(t)| \geq \epsilon_3, E_t) \Big) + \sum_{t=1}^{T} I(E_t^C) \ . \qquad (5.4)
\end{aligned}$$

We first bound the regret due to explorations.

Lemma V.6.

$$E_{\psi_0,\alpha}^{\boldsymbol{P}}\left[\sum_{t=0}^{T-1} I(E_t^C)\right] \leq 2KL\log T(1+T_{\max}),$$

where $T_{\max} = \max_{k\in\mathcal{K},i,j\in S^k} E[T_{ij}^k] + 1$, T_{ij}^k *is the hitting time of state* j *of arm* k *starting from state* i *of arm* k. *Since all arms are ergodic* $E[T_{ij}^k]$ *is finite for all* $k \in \mathcal{K}$, $i, j \in S^k$.

Proof. The number of transition probability updates that results from explorations up to time $T-1$ is at most $\sum_{k=1}^{K}\sum_{i\in S^k} L\log T$. The expected time spent in exploration during a single update is at most $(1+T_{\max})$. □

Using Lemma A.7, next we show that the probability that an estimated transition

probability is significantly different from the true transition probability given ABA is in an exploitation phase is very small. Let $C_1(P^k, \tau^k), k \in \mathcal{K}, \tau^k \in [1, \tau^*(\epsilon_1)]$ be the constant given in Lemma A.6, $C_1(\boldsymbol{P}) = \max_{k\in\mathcal{K},\tau^k\in[1,\tau^*(\epsilon_1)]} C_1(P^k, \tau^k)$.

Lemma V.7. *For an agent using ABA with constant $L \geq 3/(2\epsilon^2)$, we have*

$$P\left(|\hat{p}^k_{ss',t} - p^k_{ss'}| > \epsilon, E_t\right) \leq \frac{2}{t^2} \ ,$$

for all t, $s, s' \in S^k$, $k \in \mathcal{K}$.

Proof. Let $t(l)$ be the time $C^k_{t(l)}(s) = l$. We have,

$$\hat{p}^k_{ss',t} = \frac{N^k_t(s,s')}{C^k_t(s)} = \frac{\sum_{l=1}^{C^k_t(s)} I(X^k_{t(l)-1} = s, X^k_{t(l)} = s')}{C^k_t(s)}.$$

Note that $I(X^k_{t(l)-1} = s, X^k_{t(l)} = s'), l = 1, 2, \ldots, C^k_t(s)$ are i.i.d. random variables with mean $p^k_{ss'}$. Then

$$\begin{aligned}
P\left(|\hat{p}^k_{ss',t} - p^k_{ss'}| > \epsilon, E_t\right) &= P\left(\left|\frac{\sum_{l=1}^{C^k_t(s)} I(X^k_{t(l)-1} = s, X^k_{t(l)} = s')}{C^k_t(s)} - p^k_{ss'}\right| \geq \epsilon, E_t\right) \\
&= \sum_{b=1}^{t} P\left(\left|\sum_{l=1}^{C^k_t(s)} I(X^k_{t(l)-1} = s, X^k_{t(l)} = s') - C^k_t(s)p^k_{ss'}\right| \geq C^k_t(s)\epsilon, C^k_t(s) = b, E_t\right) \\
&\leq \sum_{b=1}^{t} 2e^{\frac{-2(L\log t\epsilon)^2}{L\log t}} = \sum_{b=1}^{t} e^{-2L\log t(\epsilon)^2} = 2\sum_{b=1}^{t}\frac{1}{t^{2L\epsilon^2}} = \frac{1}{t^{2L\epsilon^2-1}} \leq \frac{2}{t^2} \ ,
\end{aligned}$$

where we used Lemma A.7 and the fact that $C^k_t(s) \geq L\log t$ w.p.1. in the event E_t. □

The following Lemma which is an intermediate step in proving that if time t is an exploitation phase then the difference between $R^k\tau$, $\hat{R}^k_\tau$ and Q^k_τ,$\hat{Q}^k_\tau$ will be small with high probability, is proved using Lemma V.7.

Lemma V.8. *For an agent using ABA with constant* $L \geq 3/(2(\min\{\epsilon C_1(\boldsymbol{P})/4, \epsilon/2\})^2)$, *we have*

$$P(|v_\tau^k \hat{p}_{gb,t}^k - p_{gb}^k \hat{v}_\tau^k(t)| \geq \epsilon, E_t) \leq \frac{18}{t^2} \ .$$

Proof.

$$\begin{aligned}
&P(|v_\tau^k \hat{p}_{gb,t}^k - p_{gb}^k \hat{v}_\tau^k(t)| \geq \epsilon, E_t) \\
&\leq P(|v_\tau^k \hat{p}_{gb,t}^k - p_{gb}^k \hat{v}_\tau^k(t)| \geq \epsilon, |p_{gb}^k - \hat{p}_{gb,t}^k| < \eta, E_t) + P(|p_{gb}^k - \hat{p}_{gb,t}^k| \geq \eta, E_t),
\end{aligned}$$

for any η. Letting $\eta = \epsilon/2$ and using Lemma V.7 we have

$$\begin{aligned}
&P(|v^k \tau \hat{p}_{gb,t}^k - p_{gb}^k \hat{v}_\tau^k(t)| \geq \epsilon, E_t) \\
&\leq 4\left(P\left(|p_{gb}^k - \hat{p}_{gb,t}^k| \geq \frac{\epsilon}{4C_1(\boldsymbol{P})}, E_t\right) + P\left(|p_{bg}^k - \hat{p}_{bg,t}^k| \geq \frac{\epsilon}{4C_1(\boldsymbol{P})}, E_t\right)\right) \\
&+ P(|p_{gb}^k - \hat{p}_{gb,t}^k| \geq \epsilon/2, E_t) \\
&\leq \frac{18}{t^2} \ .
\end{aligned}$$

□

The next two lemmas bound the probability of deviation of $\hat{R}_\tau^k(t)$ and $\hat{Q}_\tau^k(t)$ from R_τ^k and Q_τ^k respectively.

Lemma V.9. *For an agent using ABA with constant*

$$L \geq \frac{3}{2(\min\{\frac{C_1(\boldsymbol{P})\epsilon\delta^2}{4r_{\max}}, \frac{\epsilon\delta^2}{2r_{\max}}\})^2},$$

we have on the event E_t *(here we only consider deviations in exploitation steps)*

$$P(|R_\tau^k - \hat{R}_\tau^k(t)| \geq \epsilon) \leq \frac{18}{t^2}.$$

Proof.

$$
\begin{aligned}
P(|R_\tau^k - \hat{R}_\tau^k(t)| \geq \epsilon) &= P\left(\left|\frac{r^k v_\tau^k}{v_\tau^k + \tau p_{gb}^k} - \frac{r^k \hat{v}_\tau^k(t)}{\hat{v}_\tau^k(t) + \tau \hat{p}_{gb,t}^k}\right|\right) \\
&= P\left(\tau r^k |v_\tau^k \hat{p}_{gb,t}^k - p_{gb}^k \hat{v}_\tau^k(t)| \geq \epsilon |v_\tau^k + \tau p_{gb}^k||\hat{v}_\tau^k(t) + \tau \hat{p}_{gb,t}^k|\right) \\
&\leq P\left(\tau r^k |v_\tau^k \hat{p}_{gb,t}^k - p_{gb}^k \hat{v}_\tau^k(t)| \geq \epsilon \tau^2 \delta^2\right) \\
&\leq P\left(|v_\tau^k \hat{p}_{gb,t}^k - p_{gb}^k \hat{v}_\tau^k(t)| \geq \frac{\epsilon \delta^2}{r_{\max}}\right) \\
&\leq \frac{18}{t^2},
\end{aligned}
$$

where the last inequality follows form Lemma V.8 since

$$
L \geq 3 / \left(2\left(\min\left\{\frac{C_1(\boldsymbol{P})\epsilon\delta^2}{4r_{\max}}, \frac{\epsilon\delta^2}{2r_{\max}}\right\}\right)^2\right).
$$

□

Lemma V.10. *For an agent using ABA with constant*

$$
L \geq \frac{3}{2(\min\{\frac{\epsilon\delta^2 C_1(\boldsymbol{P})}{4}, \frac{\epsilon\delta^2}{2}\})^2},
$$

we have on the event E_t

$$
P(|Q_\tau^k - \hat{Q}_\tau^k(t)| \geq \epsilon) \leq \frac{18}{t^2}\ .
$$

Proof.

$$
\begin{aligned}
P(|Q_\tau^k - \hat{Q}_\tau^k(t)| \geq \epsilon) &= P\left(\left|\frac{v_\tau^k + p_{gb}^k}{v_\tau^k + \tau p_{gb}^k} - \frac{\hat{v}_\tau^k(t) + \hat{p}_{gb,t}^k}{\hat{v}_\tau^k(t) + \tau \hat{p}_{gb,t}^k}\right|\right) \\
&= P\left((\tau - 1)|v_\tau^k \hat{p}_{gb,t}^k - p_{gb}^k \hat{v}_\tau^k(t)| \geq \epsilon |v_\tau^k + \tau p_{gb}^k||\hat{v}_\tau^k(t) + \tau \hat{p}_{gb,t}^k|\right) \\
&\leq P\left((\tau - 1)|v_\tau^k \hat{p}_{gb,t}^k - p_{gb}^k \hat{v}_\tau^k(t)| \geq \epsilon \tau^2 \delta^2\right) \\
&\leq P\left(|v_\tau^k \hat{p}_{gb,t}^k - p_{gb}^k \hat{v}_\tau^k(t)| \geq \epsilon \delta^2\right)
\end{aligned}
$$

$$\leq \frac{18}{t^2} \ ,$$

where the last inequality follows form Lemma V.8 since

$$L \geq 3/\left(2\left(\min\left\{\epsilon\delta^2 C_1(\boldsymbol{P})/4, (\epsilon\delta^2)/2\right\}\right)^2\right) .$$

□

The lower bound on the exploration constant L in Lemmas V.9 and V.10 is sufficient to make the estimated transition probabilities at an exploitation step close enough to the true transition probabilities to guarantee that the estimated waiting time is equal to the exact waiting time with very high probability, i.e., the probability of error at any time t is $O(1/t^2)$. The following theorem bounds the expected number of times ABA differs from $\gamma^{\epsilon_1,\boldsymbol{P}}$.

Theorem V.11. *For an agent using ABA with constant*

$$L \geq \frac{3}{2\min\left\{\frac{C_1(\boldsymbol{P})\epsilon_3\delta^2}{4r_{\max}}, \frac{\epsilon_3\delta^2}{2r_{\max}}, \frac{C_1(\boldsymbol{P})\epsilon_3\delta^2}{4}, \frac{\epsilon_3\delta^2}{2}\right\}} \ , \tag{5.5}$$

expected number of deviations from the ϵ_1-threshold procedure is bounded by

$$E[N_T] \leq 36K\tau^*(\epsilon_1)\beta + 2KL\log T(1 + T_{\max}) \ ,$$

where

$$\beta = \sum_{t=1}^{\infty} \frac{1}{t^2} \ .$$

Proof. Taking the expectation of (5.4) and using Lemma V.6

$$E[N_T] \leq \sum_{k=1}^{K} \sum_{t=1}^{T} \sum_{\tau=1}^{\tau^*(\epsilon_1)} \left(P(|R_\tau^k - \hat{R}_\tau^k(t)| \geq \epsilon_3, E_t) + P(|Q^k\tau - \hat{Q}_\tau^k| \geq \epsilon_3, E_t) \right) + 2KL \log T(1 + T_{\max}) \ .$$

Then, by the results of Lemmas V.9 and V.10, we have

$$\begin{aligned} E[N_T] &\leq \sum_{k=1}^{K} \sum_{t=1}^{T} \sum_{\tau=1}^{\tau^*(\epsilon_1)} \frac{20}{t^2} + 2KL \log T(1 + T_{\max}) \\ &\leq 36K\tau^*(\epsilon_1)\beta + 2KL \log T(1 + T_{\max}) \ . \end{aligned}$$

□

5.2.5 Performance of ABA

In this section we consider the performance of ABA. First we show that the performance of ABA is at most ϵ worse than $OPT/2$. Since each arm is an ergodic Markov chain, the ϵ_1-threshold policy is ergodic. Thus, after a single deviation from the ϵ_1-threshold policy only a finite difference in reward from the ϵ_1-threshold policy can occur.

Theorem V.12. *For an agent using ABA with $\delta, \epsilon_1, \epsilon_2$ and L as in (5.5), the infinite horizon average reward is at least*

$$\frac{OPT}{2(1+\epsilon_2)} - K\epsilon_1 = \frac{OPT}{2} - \epsilon \ ,$$

for

$$\epsilon = \frac{\epsilon_2 OPT}{2(1+\epsilon_2)} + K\epsilon_1 \ .$$

Moreover, the number of mathematical operations required to select an arm at any time is at most

$$K \lceil \log(z(\epsilon_2)) \rceil \tau^*(\epsilon_1) \ .$$

Proof. Since, after each deviation from the ϵ_1-threshold policy only a finite difference in reward from the ϵ_1-threshold policy can occur and the expected number of deviations of ABA is logarithmic (even sublinear is sufficient), ABA and the ϵ_1-threshold policy have the same infinite horizon average reward. Computational complexity follows from Lemma V.5. □

ABA has the fastest rate of convergence (logarithmic in time) to the ϵ_1-threshold policy given $\boldsymbol{P}$, i.e., $\gamma^{\epsilon_1,\boldsymbol{P}}$. This follows from the large deviation bounds where in order to logarithmically upper bound the number of errors in exploitations, at least logarithmic number of explorations is required. Although finite time performance of Guha's policy and $\gamma^{\epsilon_1,\boldsymbol{P}}$ is not investigated, minimizing the number of deviations will keep the performance of ABA very close to $\gamma^{\epsilon_1,\boldsymbol{P}}$ for any finite time. We define the regret of ABA with respect to $\gamma^{\epsilon_1,\boldsymbol{P}}$ at time T as the difference between the expected total reward of $\gamma^{\epsilon_1,\boldsymbol{P}}$ and ABA at time T. Next, we will show that this regret is logarithmic, uniformly over time.

Theorem V.13. *Let $r^\gamma(t)$ be the reward obtained at t by policy γ. For an agent using ABA with $\delta, \epsilon_1, \epsilon_2$ and L as in (5.5),*

$$\left| E^{\boldsymbol{P}}_{\psi_0,\alpha}\left[\sum_{t=1}^{T} r^\alpha(t)\right] - E^{\boldsymbol{P}}_{\psi_0,\gamma^{\epsilon_1,\boldsymbol{P}}}\left[\sum_{t=1}^{T} r^{\gamma^{\epsilon_1,\boldsymbol{P}}}(t)\right] \right| \leq Z(36K\tau^*(\epsilon_1)\beta + 2KL\log T(1+T_{\max})),$$

where Z is the maximum difference in expected reward resulting from a single deviation from $\gamma^{\epsilon_1,\boldsymbol{P}}$.

Proof. A single deviation from $\gamma^{\epsilon_1,\boldsymbol{P}}$ results in a difference at most Z. The expected

number of deviations from $\gamma^{\epsilon_1,\boldsymbol{P}}$ is at most $(36K\tau^*(\epsilon_1)\beta + 2KL\log T(1+T_{\max}))$ from Theorem V.11. □

5.3 Discussion

We first comment on the choice of the exploration constant L. Note that in computing the lower bound for L given by (5.5), ϵ_3 and $C_1(\boldsymbol{P})$ are not known by the agent. One way to overcome this is to increase L over time. Let L^* be the value of the lower bound. Thus, instead of exploring when $C_t^k(s) < K\log t$ for some $k \in \mathcal{K}, s \in S^k$, ABA will explore when $C_t^k(s) < L(t)\log t$ for some $k \in \mathcal{K}, s \in S^k$, where $L(t)$ is an increasing function such that $L(1) = 1$, $\lim_{t\to\infty} L(t) = \infty$. Then after some t_0, we will have $L(t) > L^*, t \geq t_0$ so our proofs for the number of deviations from the ϵ_1-threshold policy in exploitation steps will hold. Clearly, the number of explorations will be in the order $L(t)\log t$ rather than $\log t$. Given that $L(t)\log t$ is sublinear in t, Theorem V.12 will still hold. The performance difference given in Theorem V.13 will be bounded by $L(T)\log T$ instead of $\log T$.

Secondly, we note that our results hold under the burstiness assumption, i.e., $p_{gb}^k + p_{bg}^k < 1, \forall k \in \mathcal{K}$. This is a sufficient condition for the approximate optimality of Guha's policy and the ABA. It is an open problem to find approximately optimal algorithms under weaker assumptions on the transition probabilities.

Thirdly, we compare the results obtained in this chapter with the results in Chapters III and IV. The algorithm in Chapter III, i.e., the *regenerative cycle algorithm* (RCA) assigns an index to each arm which is based on the sample mean of the rewards from that arm plus an exploration term that depends on how many times that arm is selected. Indices in RCA can be computed recursively since they depend on the sample mean, and the computation may not be necessary at every t since RCA operates in blocks. Thus, RCA is computationally simpler than ABA. It is shown that for any t the regret of RCA with respect to the best single-arm policy (policy which

always selects the arm with the highest mean reward) is logarithmic in time. This result holds for general finite state arms. However, the best single-arm policy may have linear regret with respect to the optimal policy which is allowed to switch arms at every time *Auer et al.* (2003). Another algorithm is the *inflated reward computation with estimated probabilities* (IRCEP) proposed in Chapter IV. IRCEP assigns an index to each arm based on an inflation of the right hand side of the estimated average reward optimality equation. At any time step, if the transition probability estimates are accurate, IRCEP exploits by choosing the arm with the highest index. Otherwise, it explores to re-estimate the transition probabilities. Thus, at each exploitation phase IRCEP needs to solve the estimated average reward optimality equations for a POMDP which is intractable. However, under some assumptions on the structure of the optimal policy for the infinite horizon average reward problem, IRCEP is shown to achieve logarithmic regret with respect to the optimal policy for the finite horizon undiscounted problem. Thus, we can say that ABA lies in between the two algorithms discussed above. It is both efficient in terms of computation and performance.

Finally, we note that the adaptive learning approach we used here can be generalized for learning different policies, whenever the computation of actions are related to the transition probability estimates in such a way that it is possible to exploit some large deviation bound. As an example, we can develop a similar adaptive algorithm with logarithmic regret with respect to the myopic policy. Although myopic policy is in general not optimal for the restless bandit problem it is computationally simple and its optimality is shown under some special cases in *Ahmad et al.* (2009).

CHAPTER VI

Multi-agent Restless Bandits with a Collision Model

In this chapter we study the decentralized multi-agent restless bandit model, in which each agent receives a binary feedback about the activity on the arm it selects. Specifically we consider the collision model given in Definition I.5, in which an agent gets zero reward if there is another agent who selected the same arm with it. Although such an agent can not receive the reward, we assume that it still observes the reward. We assume that the agents cannot communicate with each other. Therefore an agent does not perfectly know the actions of other agents. However, we assume that there is a feedback structure such as the one given in Model I.17 in which an agent receives binary feedback about the other agents who selected the same arm with it. Basically, this binary feedback provides the information about whether the agent is the only agent who selected that arm or not.

For example, if the agents are cognitive radio devices which are transmitter-receiver pairs, at the beginning of each time step they can first sense a channel to learn about its reward, and then transmit on that channel. If more than one device transmit on the same channel at the same time, there will be a collision and the receivers of the agents will not receive the transmitted signal correctly. In this setting, an agent will only receive the sensed reward/rate if its transmission is successful,

which is signaled to the agent by the binary feedback (ACK/NACK) at the end of the time slot. This example justifies our assumption that the agent observes the reward even when there is a collision.

This problem is studied for the IID arm rewards in *Liu and Zhao* (2010) and *Anandkumar et al.* (2011), and distributed learning algorithms with logarithmic weak regret are proposed. In both of these works, the main motivation is multi-user dynamic spectrum access. However, spectrum measurements (see, e.g., *López-Benítez and Casadevall* (2011)) show that a discrete time Markov chain better models the change in channel conditions. This motivates us to study the restless bandit problem with the binary feedback model.

Since our main motivation is dynamic spectrum access in which wireless devices are limited in terms of computational power and memory, similar to Chapter III, we consider weak regret as the performance measure. The learning algorithm we propose in this chapter is a distributed extension of the regenerative cycle algorithm given in Chapter III, and it achieves logarithmic weak regret with respect to the best centralized static policy.

The organization of this chapter is as follows. Problem definition and notations are given in Section 6.1. Decentralized restless bandit problem with the collision model is investigated, and an algorithm with logarithmic weak regret is proposed in Section 6.2. A cognitive radio network application and numerical results are given in Section 6.3. Finally, discussion is given in Section 6.4.

6.1 Problem Formulation and Preliminaries

In this chapter we study the restless Markovian model given in Definition I.3 with K arms, and M decentralized agents indexed by the set $\mathcal{M} = \{1, 2, \ldots, M\}$. All the assumptions given for the arms in Section 3.1 of Chapter III also holds in this chapter. To summarize: (i) transition probability matrix of each arm has an

irreducible multiplicative symmetrization (this is not necessary with our alternative proof), (ii) $\mu^1 \geq \mu^2 \geq \ldots \geq \mu^K$ without loss of generality, (iii) $\mu^M > \mu^{M+1}$.

At each time t, agent i selects a single arm based on the algorithm α_i it uses. Let $\alpha_i(t)$ be the arm selected by agent i at time t when it uses algorithm α_i. Let $\boldsymbol{\alpha(t)} = \{\alpha_1(t), \alpha_2(t), \ldots, \alpha_K(t)\}$ be the vector of arm selections at time t. The collision model between the agents is given in Definition I.5, where if more than one agent selects the same arm at the same time step, then none of these agents get any reward. The weak regret for the multi-agent model is given in Definition I.12 which we restate below.

$$R^{\boldsymbol{\alpha}}(T) = T \sum_{k=1}^{M} \mu^{\sigma^k} - E\left[\sum_{t=1}^{T}\sum_{i=1}^{M} r^{\alpha_i(t)}(t) I(n^t_{\alpha_i(t)} = 1)\right], \tag{6.1}$$

where n^t_k is the number of agents on arm k at time t.

Although an agent does not receive any reward when there is collision, the agent observes the reward process of the arm it selects perfectly. Therefore, learning is affected by a collision only in an indirect way. Within the context of our motivating application, this problem models a decentralized multi-user dynamic spectrum access scenario, where multiple users compete for a common set of channels. Each user performs channel sensing and data transmission tasks in each time slot. Sensing is done at the beginning of a slot; the user observes the quality of a selected channel. This is followed by data transmission in the same channel. The user receives feedback at the end of the slot (e.g., in the form of an acknowledgement) on whether the transmission is successful. If more than one user selects the same channel in the same slot, then a collision occurs and none of the users gets any reward.

6.2 A Distributed Algorithm with Logarithmic Weak Regret

The algorithm we construct and analyze in this section is a decentralized extension to RCA-M and will be referred to as the Decentralized Regenerative Cycle Algorithm or DRCA. This algorithm works similarly as RCA-M, using the same block structure. However, since agents are uncoordinated, each agent keeps its own locally computed indices for all arms, and they may vary from agent to agent. As before, an agent continues to play the same arm till it completes a block, upon which it updates the indices for the arms using state observations from SB2's. Within this completed block it may experience collision in any of the time slots; for these slots it does not receive any reward. At the end of a block, if the agent did not experience a collision in the last slot of the block, it continues to play the arm with the same rank in the next block after the index update. If it did experience a collision, then the agent updates the indices for the arms, and then randomly selects an arm within the top M arms, based on the indices it currently has for all the arms, to play in the next block. The pseudocode of DRCA is given in Figure 6.1.

We see that compared to RCA-M, the main difference in DRCA is the randomization upon completion of a block. This is because if all agents choose the arm with the highest index, then the number of collisions will be very high even if agents do not have exactly the same local indices; this in turn leads to large regret. Letting an agent randomize among its M highest-ranked arms can help alleviate this problem, and aims to eventually orthogonalize the M agents in their choice of arms. This is the same idea used in *Anandkumar et al.* (2011). The difference is that, in *Anandkumar et al.* (2011) the randomization is done each time a collision occurs under an IID reward model, whereas in our case the randomization is done at the end of a completed block and is therefore less frequent as block lengths are random. The reason for this is because with the Markovian reward model, index updates can only be done after a regenerative cycle; switching before a block is completed will waste the state

Decentralized Regenerative Cycle Algorithm (DRCA) for agent j:

Initialize: $b = 1, t = 0, B^{k,j} = 0, N_2^{k,j} = 0, r^k = 0, \forall k = \mathcal{K}$. Select θ_j uniformly from $\{1, \ldots, M\}$
for $b \leq K$ **do**
  play arm b; set γ^b to be the first state observed
  $t := t + 1$; $N_2^{b,j} := N_2^{b,j} + 1$; $r^b := r^b + r_{\gamma^b}^b$
  play arm b; denote observed state as x
  while $x \neq \gamma^b$ **do**
    $t := t + 1$; $N_2^{b,j} := N_2^{b,j} + 1$; $r^b := r^b + r_x^b$
    play arm b; denote observed state as x
  end while
  if $Z(t) = 0$ **then**
    Select θ_j uniformly from $\{1, \ldots, M\}$
  else
    Do not change θ
  end if
  $b := b + 1$; $B^{b,j} := B^{b,j} + 1$; $t := t + 1$
end for
for $k = 1$ to K **do**
  compute index $g^{k,j} := \frac{r^j}{N_2^{k,j}} + \sqrt{\frac{2 \ln b}{B^{k,j}}}$
end for
$k := \sigma(\theta, \boldsymbol{g}^j)$
while (1) **do**
  play arm k; denote observed state as x
  while $x \neq \gamma^k$ **do**
    $t := t + 1$
    play arm k; denote observed state as x
  end while
  $t := t + 1$; $N_2^{k,j} := N_2^{k,j} + 1$; $r^k := r^k + r_x^k$
  play arm k; denote observed state as x
  while $x \neq \gamma^k$ **do**
    $t := t + 1$; $N_2^{k,j} := N_2^{k,j} + 1$; $r^k := r^k + r_x^k$
    play arm k; denote observed state as x
  end while
  if $Z(t) = 0$ **then**
    Select θ_j uniformly from $\{1, \ldots, M\}$
  else
    Do not change θ
  end if
  $b := b + 1$; $B^{k,j} := B^{k,j} + 1$ $t := t + 1$
  for $k = 1$ to K **do**
    compute index $g^{k,j} := \frac{r^k}{N_2^{k,j}} + \sqrt{\frac{2 \ln b}{B^{k,j}}}$
  end for
  $k := \sigma(\theta, \boldsymbol{g}^j)$
end while

Figure 6.1: pseudocode of DRCA

observations made within that incomplete block.

In the remainder of this section we show that using the above algorithm, the regret summed over all agents with respect to the optimal centralized (coordinated) solution, where M agents always play the M-best arms, is logarithmic in time. Our

analysis follows a similar approach as in *Anandkumar et al.* (2011), adapted to blocks rather than time slots and with a number of technical differences. In particular, the proof of Lemma VI.3 is significantly different because a single block of some agent may collide with multiple blocks of other agents; thus we need to consider the actions of the agents jointly in order to bound the regret.

Let $Y^{k,j}(b)$ be the sample mean of the rewards inferred from observations (not the actual rewards received since in this case reward is zero when there is collision) by agent j during its bth block in which it plays arm k. Let $B^{k,j}(b)$ be the number of blocks in which arm k is played by agent j at the end of its bth block. Let $b_j(t)$ be the number of agent j's completed blocks up to time t. Then the index of arm k computed (and perceived) by agent j at the end of its bth completed block is given by

$$g^{k,j}(b) = \frac{\sum_{v=1}^{B^{k,j}(b)} Y^{k,j}(v)}{B^{k,j}(b)} + \sqrt{\frac{2\ln b}{B^{k,j}(b)}} \ . \tag{6.2}$$

The difference between the index given in (3.1) and (6.2) is that the exploration term in (6.2) depends on the number of blocks completed by an agent, while in (3.1) it depends on the number of time steps spent in SB2's of an agent.

Let $J(t)$ be the number of slots involving collisions in the M optimal arms in the first t slots, and let $N^{k,j}(t)$ denote the number of slots agent j plays arm k up to time t. Then from Proposition 1 in *Anandkumar et al.* (2011), we have

$$R^{\boldsymbol{\alpha}}(T) \leq \mu^1 \left(\sum_{j=1}^{M} \sum_{k=M+1}^{K} E[N^{k,j}(T)] + E[J(T)] \right) \ . \tag{6.3}$$

This result relates the regret to the amount of loss due to collision in the optimal arms, and the plays in the suboptimal arms.

Lemma VI.1. *When all agents use DRCA, for any agent j and any suboptimal arm*

k we have

$$E[B^{k,j}(b_j(T))] \leq \frac{8 \ln T}{(\mu^M - \mu^k)^2} + 1 + M\beta \ .$$

Proof. See Appendix K. □

The next lemma shows that provided all agents have the correct ordering of arms, the expected number of blocks needed to reach an orthogonal configuration by randomization at the end of blocks is finite.

Lemma VI.2. *Given all agents have the correct ordering of the arms and do not change this ordering anymore, the expected number of blocks needed summed over all agents to reach an orthogonal configuration is bounded above by*

$$O_B = M \left[\binom{2M-1}{M} + 1 \right] \ .$$

Proof. The proof is similar to the proof of Lemma 2 in *Anandkumar et al.* (2011), by performing randomization at the end of each block instead of at every time step. Consider a genie aided scheme. At the end of each block a genie checks if there is collision in any of the arms. If this is the case, then genie orders all agents to randomize (even the agents with incomplete blocks will randomize and their incomplete block will also be counted in the number of blocks). Thus, in the worst case for each block with a collision, M blocks (for all agents) is counted when calculating the number of blocks spent to reach an orthogonal configuration. Then using the proof of Lemma 2 in *Anandkumar et al.* (2011), the expected number of blocks (which ends at different times) until an orthogonal configuration is reached is $(\binom{2M-1}{M} - 1)$. Thus the expected number of blocks until an orthogonal configuration is $M(\binom{2M-1}{M} - 1))$. Now consider the model without the genie, in which any agent whose block is not complete or who does not have a collision at the end of his block will not randomize. This means

that the number of configurations that can be reached in any randomization is less then the one in the genie aided model. Since the orthogonal configuration is in the set of configurations that can be reached for any randomization, the probability of reaching an orthogonal configuration at the end of each block is at least the one in the genie aided model. Thus the expected number of blocks to reach an orthogonal configuration is at most the one in the genie aided model. □

Let $B'(t)$ be the number of completed blocks up to t, in which at least one of the top M estimated ranks of the arms at some agent is wrong. Let $p_{xy}^k(t)$ be the t step transition probability from state x to y of arm k. Since all arms are ergodic, there exists $N > 0$ such that $p_{xy}^k(N) > 0$, for all $k \in \mathcal{K}, x, y \in S^k$. We now bound the expectation of $B'(t)$.

Lemma VI.3. *When all agents use DRCA, we have*

$$E[B'(T)] < M\left[2N(M-1)\left(1+\frac{1}{\lambda}(\ln T+1)\right)+1\right] \times\left(\sum_{a=1}^{M}\sum_{c=a+1}^{K}\left(\frac{8\ln T}{(\mu^a-\mu^c)^2}+1+\beta\right)\right),$$

where N is the minimum integer such that $p_{xy}^k(N) > 0$ for all $k \in \mathcal{K}, x, y \in S^k$, $\lambda = \ln\left(\frac{1}{1-p^*}\right)$ and $p^* = \min_{k\in\mathcal{K},x,y\in S^k} p_{xy}^k(N)$.

Proof. See Appendix L. □

Next we show that the expected number of collisions in the optimal arms is at most $\log^2(.)$ in time. Let $H(t)$ be the number of completed blocks in which some collision occurred in the optimal arms up to time t.

Lemma VI.4. *Under DRCA, we have*

$$E[H(T)] \leq O_B E[B'(T)]$$

Proof. See Appendix M. □

Combining all the above lemmas and using the fact that the expected block length is finite we have the following result.

Theorem VI.5. *When all agents use DRCA, we have*

$$R(T) \leq \mu^1 D_{\max} \left[\sum_{j=1}^{M} \sum_{k=M+1}^{K} \left(\frac{8 \ln T}{(\mu^M - \mu^k)^2} + M\beta + 2 \right) \right.$$
$$\left. + O_B M \left[2N(M-1) \left(1 + \frac{1}{\lambda}(\ln T + 1) \right) + 1 \right] \left(\sum_{a=1}^{M} \sum_{c=a+1}^{K} \left(\frac{8 \ln T}{(\mu^a - \mu^c)^2} + 1 + \beta \right) \right) \right],$$

where

$$D_{\max} = \frac{1}{\pi_{\min}} + \Omega_{\max} + 1, \ \Omega_{\max} = \max_{k \in \mathcal{K}} \Omega^k_{\max}, \ \pi_{\min} = \min_{k \in \mathcal{K}} \pi^k_{\min},$$

and N is the minimum integer such that $p^k_{xy}(N) > 0$ for all $k \in \mathcal{K}, x, y \in S^k$, $\lambda = \ln\left(\frac{1}{1-p^*}\right)$ and $p^* = \min_{k \in \mathcal{K}, x, y \in S^k} p^k_{xy}(N)$.

Proof. Since the expected length of each block is at most $D_{\max}$ and the expected number of time steps between current time T and the time of the last completed block is at most the expected block length, we have $E[N^{k,j}(T)] \leq D_{\max}(E[B^{k,j}(b_j(T))] + 1)$ and $E[J(T)] \leq D_{\max}(E[H(T)] + 1)$. The result follows from substituting these into (6.3) and using results of Lemmas VI.1 and VI.4. □

It is worth mentioning that our proof of the logarithmic regret upper bound in this section is based on the regenerative cycles but does not rely on a large deviation bound for Markov chains as in the main results of Chapter III. The main idea is that the sample mean rewards observed within regenerative cycles with the same regenerative state form an IID random process; our results are easier to prove by exploiting the IID structure. The same method can be used in the previous sections as well by choosing a constant regenerative state for each arm. Moreover, under this

method we no longer need the assumption that $p_{xx}^k > 0$ for any $k \in \mathcal{K}, x \in S^k$ (or the irreducible multiplicative symmetrization assumption). Indeed, with this method the same results can be derived for arbitrary non-Markovian discrete time renewal processes with finite mean cycle time and bounded rewards. However, we note that the previous method based on the large deviation bound for Markov chains is still of importance because it works when the regenerative states are adapted over time. In this case the cycles are no longer IID and the expected average reward in a cycle is not necessarily the mean reward of an arm. We give applications where there is a need to change the regenerative state of an arm over time in Section 6.4.

6.3 Numerical Results

In this section we evaluate the performance of DRCA with $M = 2$ agents in the opportunistic spectrum access (OSA) scenario described in Section 3.4 of Chapter III. This scenario is equivalent to a cognitive radio network, where there are M decentralized secondary users (agents), each of which senses a single channel at each time step, and transmits on that channel if there is no primary user on that channel. If an agent observes a primary user on the channel it senses, it does not transmit on that channel. State of channel k is 0 when there is a primary user, and 1 when there is no primary user. The rewards an agent gets from the states are $r_1^k = 1$, $r_0^k = 0.1$, $\forall k \in \mathcal{K}$. We assume that if more than one secondary user transmits on the same channel at the same time, then they collide. This means that the transmission is unsuccessful for all of them, and they get 0 reward. Note that the reward a secondary user gets when there is collision is 0 which is smaller than the reward it will get when it sees the channel occupied by the primary user. This is because, the secondary user will not attempt to transmit, hence will not consume energy in the latter case, whereas it will consume some energy for the failed transmission in the former case. We consider 4 different channel conditions S1-S4, each consisting of 10 channels with

different state transition probabilities. The state transition probabilities and mean rewards of the channels in each scenario are given in Tables 3.2 and 3.3, respectively. We present the regret of DRCA with 2 agents in Figure 6.2. The results are similar to that of RCA with the index given in (3.2), but with a larger regret due to collisions.

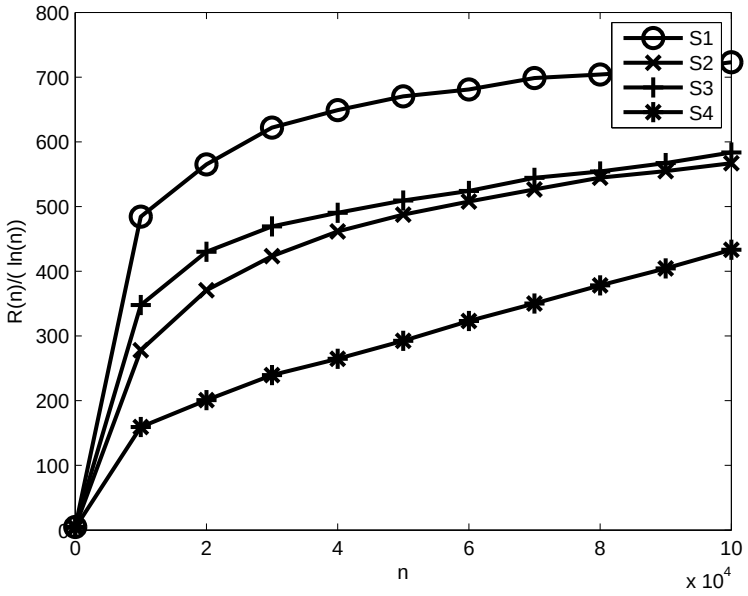

Figure 6.2: regret of DRCA with 2 users

6.4 Discussion

Since DRCA is the decentralized extension of RCA, they share many properties. Therefore the improvements and extensions on RCA discussed in Section 3.5, can also be applied to DRCA. Especially if the reward from state x of arm k, i.e., r_x^k is a random variable, the method of adapting the regenerative state, which is given in Section 3.5 can also be used in DRCA to achieve logarithmic regret.

CHAPTER VII

Online Learning in Decentralized Multi-agent Resource Sharing

In this chapter we consider a decentralized multi-agent online learning problem in a resource sharing setting, where the reward an agent gets from using a resource depends on not only the random realization of the resource quality, but also how many other agents are simultaneously using that resource, both of which are unknown a priori to the agent. Similar to the rest of the thesis, in this chapter we use the mathematical framework of bandit problems. Specifically, we consider the resource sharing setting as a bandit problem, where a resource corresponds to an arm, which generates random rewards depending on the number of agents using that resource, at discrete time steps. The goal of the agents is to achieve a system objective, such as maximization of the expected total reward over time. One of the major challenges in a decentralized system is the asymmetry in information possessed by different agents. In the absence of communication or feedback, each agent acts based on its own history of observations and actions, making it impossible to achieve an arbitrary system objective.

The main contribution of this chapter is to establish certain relationship between the degree of decentralization and the achievable performance of a learning algorithm. The different degree of decentralization is captured in a sequence of four settings with

increasing feedback and communication. Specifically, we consider the *no feedback*, *partial feedback*, *partial feedback with synchronization* and *costly communication* models given in Section 1.2.6. As the feedback and communication level between the agents increase, certain performance objectives will be achieved for a wider class of resource rewards, ranging from a special case of the general symmetric interaction model given in Definition I.7 to the agent-specific interaction model given in Definition I.8.

The performance measure we use in this chapter is the weak regret of a learning algorithm with respect to the *best static resource allocation rule*. At time T, this is the difference up to time T between the total expected reward of the best static allocation rule and that of the algorithm. Note that a static allocation rule is an offline rule in which a fixed allocation is selected at each time step. The regret quantifies the rate of convergence to the best static allocation. As T goes to infinity the performance of any algorithm with sublinear regret will converge in average reward to the optimal static allocation, while the convergence is faster for an algorithm with smaller regret.

For the first three settings in which communication between the agents is not possible, we consider the general symmetric interaction model. For the fourth setting in which we assume costly communication structure, our results also hold for the agent-specific interaction model. Specifically, for a special case of the general symmetric interaction model, we show that when the agents are fully decentralized without any communication or coordination, there exist learning algorithms under which individual actions will converge to an equilibrium point of an equivalent congestion game. Although this equilibrium may not be socially optimal, it has some performance guarantees.

In the second setting, we show that if agents are given partial feedback about the number of agents with whom they share the same resource, then they can achieve sublinear regret with respect to the optimal static allocation, provided that rewards obtained by agents sharing the same resource are governed by a *general symmetric*

interaction function, whereby the reward an agent gets from a resource is affected due to sharing and this effect is the same for all agents sharing the same resource. In the third setting, we show that if initial synchronization among agents is allowed which may require a small amount of communication at the beginning, then the agents can achieve logarithmic regret in the general symmetric interaction model. Finally, we introduce the "costly communication model", in which agents can communicate with each other and share their past observations at a cost. We show that there exist a distributed online learning algorithm that achieves logarithmic regret even when rewards from resources are agent-specific.

The logarithmic regret results above hold under the condition that a lower bound on the performance gap between the best and the second-best allocations is known by the agents; if this gap is unknown, we prove that agents can achieve near-logarithmic regret. Furthermore, our algorithms achieve the same order of regret when we introduce computation and switching costs, with the former modeling the time and resources it takes for the agents to compute the estimated optimal allocation, and the latter modeling the cost of changing resource.

The organization of this chapter is as follows. Problem definition and notations are given in Section 7.1. Achievable performance without feedback and communication is considered in Section 7.2, and a convergent algorithm is proposed. Then, the partial feedback model is considered in Section 7.3, and an algorithm with sublinear regret is proposed for IID resource rewards. The model with initial synchronization is studied in Section 7.4, and an algorithm with logarithmic regret for both the IID and Markovian resource rewards is proposed. Then, we study the achievable performance when costly communication is possible in Section 7.5. Finally, a discussion of models we proposed in this chapter is given in Section 7.6.

7.1 Problem Formulation and Preliminaries

We consider M distributed agents indexed by the set $\mathcal{M} = \{1, 2, \ldots, M\}$, and K mutually independent resources indexed by the set $\mathcal{K} = \{1, 2, \ldots, K\}$ in a discrete time setting with time index $t = 1, 2, \ldots$. At each time step t, an agent chooses a single resource from the set $\mathcal{K}$. The reward the agent gets from a resource depends on the *internal* state of the resource which varies stochastically over time and the interaction between the agents using that resource. For example, in a dynamic spectrum access problem agents are transmitter-receiver pairs, while resources are channels. The channel conditions vary stochastically over time due to reasons such as fading, primary user activity, etc.

The objective is to maximize the total system reward, which is the sum of the reward of all agents up to some T. The agents have no knowledge of the resource reward statistics, either as an average or as a prior, so cannot simply solve a distributed optimization problem to find the optimal allocation. Instead, they need to learn the resource rewards over time, and estimate the optimal allocation in a distributed way. The resource selected by agent i at time t depends on the algorithm α_i used by the agent. An agent's decision at time t is based on the history of decisions and observations it has by time t.

Our goal in this chapter is to design distributed algorithms $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_M)$ whose performance converges to the optimal allocation as fast as possible. Let $\alpha_i(t)$ be the resource selected by agent i at time t when it uses algorithm α_i. Let $\boldsymbol{\alpha}(t) = (\alpha_1(t), \alpha_2(t), \ldots, \alpha_K(t))$ be the vector of resource selections at time t.

7.1.1 Factors Determining the Resource Rewards

The quality of a resource perceived by an agent depends on two factors: (1) the state of the resource, and (2) the congestion or activity level in the resource, i.e., its number of simultaneous users. Specifically, we assume that when agent i selects

resource k at time t, it receives (and observes) a reward $r_k^i(s,n)$, where n denotes the total number of users on channel k at time t, and $s \in S^k$ the state of channel k at time t with S^k being the state space of channel k. Let

$$r_k^i : S^k \times \mathcal{M} \to [0,1], \forall k \in \mathcal{K}.$$

This resource rewards correspond to the agent-specific interaction model given in Defition I.8. When the resource rewards are agent-independent, i.e., in the general symmetric interaction model, we drop the superscript on the resource reward and denote it by r_k. In this case, The quantity $r_k(s,1)$ will also be referred to as the *single-occupancy reward* of the resource k in state s. For our application, r_k is in general non-increasing in n, i.e., more agents using the same resource leads to performance degradation due to increased congestion or interference. However, all our analysis holds regardless of this property. Below we show two examples of this type of multi-agent resource sharing problems with these type of resource rewards which are taken from Section 1.1.

Example VII.1. Random access. If user i is the only one using channel k at time t with the channel in fading state s, it gets a single-occupancy channel quality given by some $q_k(s)$, where $q_k : S^k \to [0,1]$. For instance this could be the received SNR, packet delivery ratio or data throughput. When there are n users simultaneously using the channel, then under a collision model in each time step each user has a probability $\frac{1}{n}$ of obtaining access, which results in a channel quality of

$$r_k(s,n) = \frac{1}{n} q_k(s) \ .$$

Example VII.2. Code division multiple access (CDMA). In this case, let $s \in \{0,1\}$ denote the primary user activity on channel k: $s = 1$ if there is no primary user on channel (or channel is available) and $s = 0$ otherwise. A secondary user is

only allowed to access the channel if $s = 1$. Multiple secondary users share access to the channel using CDMA. When channel k is not occupied by a primary user, the rate a secondary user i gets can be modeled as (see, e.g. *Tekin et al.* (2012)),

$$\log\left(1+\gamma\frac{h_{ii}^k P_i^k}{N_o+\sum_{j\neq i} h_{ji}^k P_j^k}\right),$$

where h_{ji}^k is the channel gain between the transmitter of user j and the receiver of user i, P_j^k is the transmit power of user j on channel k, N_o is the noise power, and $\gamma > 0$ is the spreading gain. If we assume the rate function to be user-independent, i.e., $h_{ii}^k = \hat{h}^k, \forall i \in \mathcal{M}$, $h_{ji}^k = \tilde{h}^k, \forall i \neq j \in \mathcal{M}$, $P_i^k = P^k$, $\forall i \in \mathcal{M}$, which is a reasonable approximation in a homogeneous environment, then we obtain

$$r_k(s,n) = s\log\left(1+\gamma\frac{\hat{h}^k P_i^k}{N_o+(n-1)\tilde{h}^k P^k}\right).$$

Note that in both examples above the effects of congestion and channel state on the received reward are separable, i.e., $r_k(s,n) = g_k(n)q_k(s)$, for some functions $g_k(\cdot)$ and $q_k(\cdot)$. Our results are not limited to this case, and holds for any general function $r_k(\cdot,\cdot)$.

7.1.2 Optimal Allocations and the Regret

In this subsection, we define the key properties of optimal resource allocations in both the general symmetric and agent-specific interaction models. Then, we provide the definitions of regret used in this chapter. We use these properties to design learning algorithms with low regret.

7.1.2.1 General Symmetric Interaction Model

Consider the general symmetric interaction model given in Definition I.7. For a resource-activity pair (k, n), the mean reward is given by

$$\mu_{k,n} := \int_{S^k} r_k(x, n) P^k(dx),$$

when the resource rewards evolve according to an IID model given in Definition I.1, and

$$\mu_{k,n} := \sum_{x \in S^k} r_k(x, n) \pi_x^k,$$

when the resource rewards evolve according to the uncontrolled Markovian model given in Definition I.4[1]. Note that P^k is the reward distribution of resource k, and π_x^k is the stationary probability of state x of arm k. The set of optimal allocations is given by

$$\mathcal{A}^* = \arg\max_{\boldsymbol{a} \in \mathcal{A}} \sum_{i=1}^{M} \mu_{a_i, n_{a_i}(\boldsymbol{a})},$$

where $\boldsymbol{a} = (a_1, a_2, \ldots, a_M)$ is an allocation of resources to agents, a_i is the resource used by agent i, $\mathcal{A} = \{\boldsymbol{a} = (a_1, a_2, \ldots, a_M) : a_i \in \mathcal{K}\}$ is the set of possible allocations, and $n_k(\boldsymbol{a})$ is the number of agents using resource k under allocation $\boldsymbol{a}$. The value of an allocation $\boldsymbol{a} \in \mathcal{A}$ is given by

$$v(\boldsymbol{a}) := \sum_{i=1}^{M} \mu_{a_i, n_{a_i}}.$$

[1] Although for simplicity we only consider the uncontrolled restless Markovian model in this chapter, all result which holds for this model also holds for the more general restless Markovian model given in Definition I.3

Let

$$v^* := \max_{\boldsymbol{a} \in \mathcal{A}} v(\boldsymbol{a}),$$

be the value of the optimal allocation, and

$$\mathcal{A}^* := \arg\max_{\boldsymbol{a} \in \mathcal{A}} v(\boldsymbol{a}),$$

be the set of optimal allocations. Since rewards are agent-independent, the value of an allocation will not change as long as the number of agents using each resource remains the same. As a next step, we give the characterization of the optimal allocation in terms of resource-activity vectors. A resource-activity vector is a vector $\boldsymbol{n} = (n_1, n_2, \ldots, n_K)$, where n_k denotes the number of agents using resource k. Let $\mathcal{A}(\boldsymbol{n})$ be the set of allocations that result in resource-activity vector $\boldsymbol{n}$. The value of resource-activity vector $\boldsymbol{n}$ is given by

$$w(\boldsymbol{n}) := \sum_{k=1}^{K} n_k \mu_{k,n_k}.$$

Note that for any $\boldsymbol{a} \in \mathcal{A}(\boldsymbol{n})$, $v(\boldsymbol{a}) = w(\boldsymbol{n})$. A resource-activity vector whose value is larger than or equal to the value of any resource-activity vector is called a number-optimal allocation. Let w^* denote the value of a number-optimal allocation. By the above argument $w^* = v^*$. The set of number-optimal allocations is given by

$$\mathcal{N}^* := \arg\max_{\boldsymbol{n} \in \mathcal{N}} \sum_{k=1}^{K} n_k \mu_{k,n_k},$$

where $\mathcal{N}$ is the set of feasible resource-activity vectors. For any $\boldsymbol{n} \in \mathcal{N}$, its suboptimality gap is defined as

$$\Delta(\boldsymbol{n}) := v^* - \sum_{k=1}^{K} n_k \mu_{k,n_k}.$$

Then the minimum suboptimality gap, i.e., the difference between the best and the second-best allocations, is

$$\Delta_{\min} := \min_{\boldsymbol{n} \in \mathcal{N} - \mathcal{N}^*} \Delta(\boldsymbol{n}). \tag{7.1}$$

We adopt the following assumption on the set of number-optimal allocations

Assumption VII.3. ***Uniqueness.*** *There is a unique optimal allocation in terms of the number of agents on each resource, i.e., the cardinality of* $\mathcal{N}^*$, $|\mathcal{N}^*| = 1$.

Let $\boldsymbol{n}^*$ denote the unique number-optimal allocation when Assumption VII.3 holds, and let $\mathcal{O}^*$ be the set of resources used by at least one agent under the optimal allocation. This assumption guarantees convergence by random selections over the optimal resources, when each agent knows the number-optimal allocation. Without this uniqueness assumption, even if all agents know all number-optimal allocations, simple randomizations cannot ensure convergence unless the agents agree upon a specific allocation. In Section 7.6 we discuss how the uniqueness assumption can be relaxed. The uniqueness assumption implies the following stability condition.

Lemma VII.4. ***Stability.*** *When Assumption VII.3 holds, for a set of estimated mean rewards* $\hat{\mu}_{k,n_k}$, *if* $|\hat{\mu}_{k,n_k} - \mu_{k,n_k}| < \Delta_{\min}/2M$, $\forall k \in \mathcal{K}, n_k \in \mathcal{M}$, *then*

$$\arg\max_{\boldsymbol{n} \in \mathcal{N}} \sum_{k=1}^{K} n_k \hat{\mu}_{k,n_k} = \mathcal{N}^*.$$

Proof. Let $\hat{v}(\boldsymbol{n})$ be the estimated value of resource-activity vector $\boldsymbol{n}$ computed using

the estimated mean rewards $\hat{\mu}_{k,n_k}$. Then, $|\hat{\mu}_{k,n_k} - \mu_{k,n_k}| < \Delta_{\min}/(2M)$, $\forall k \in \mathcal{K}, n_k \in \mathcal{M}$ implies that for any $\boldsymbol{n} \in \mathcal{N}$, we have $|\hat{v}(\boldsymbol{n}) - v(\boldsymbol{n})| \leq \Delta_{\min}/2$. This implies that

$$v^* - \hat{v}(\boldsymbol{n}^*) < \Delta_{\min}/2, \tag{7.2}$$

and, for any suboptimal $\boldsymbol{n} \in \mathcal{N}$

$$\hat{v}(\boldsymbol{n}) - v(\boldsymbol{n}) < \Delta_{\min}/2. \tag{7.3}$$

Combining (7.2) and (7.3), and using (7.1), we have for any suboptimal $\boldsymbol{n}$

$$\hat{v}(\boldsymbol{n}^*) - \hat{v}(\boldsymbol{n}) > \Delta_{\min} - 2\Delta_{\min}/2 = 0.$$

□

The stability condition suggests that when an agent estimates sufficiently accurately the mean rewards of resource-activity pairs, it can find the optimal allocation. In Sections 7.3 and 7.4 we study algorithms under the assumption that a lower bound on $\Delta_{\min}$ is known by the agents. This assumption may seem strong with unknown statistics of the resource rewards. However, if the resource reward represents a discrete quantity such as the data rate of a channel in bytes or revenue from a business in dollars, then all agents will know that $\Delta_{\min} \geq 1$ byte per second or dollars. Extension of our results to the case when $\Delta_{\min}$ is unknown to the agents can be done by increasing the number of samples that are used to form estimates $\hat{\mu}_{k,n}$ of $\mu_{k,n}$ over time at a specific rate. In Section 7.6 we investigate this extension in detail.

We measure the performance of our learning algorithm by calculating the weak regret given in Definition I.14, which is

$$\textbf{I:}\ \ R^{\alpha}(T) := Tv^* - E_{\alpha}^{\boldsymbol{P}}\left[\sum_{t=1}^{T}\sum_{i=1}^{M} r_{\alpha_i(t),n^t_{\alpha_i(t)}}(t)\right]. \tag{7.4}$$

where

$$r_{\alpha_i(t),n^t_{\alpha_i(t)}}(t) := r_{\alpha_i(t)}(s^t_{\alpha_i(t)}, n^t_{\alpha_i(t)}),$$

s^t_k is the state of channel k at time t, and n^t_k is the number of agents on channel k at time t. We also consider switching cost C_{swc} and computation cost C_{cmp}. When these terms are added the regret at time T becomes

$$\textbf{II: } R^\alpha(T) = Tv^* - E^{\boldsymbol{P}}_\alpha \left[\sum_{t=1}^{T} \sum_{i=1}^{M} r_{\alpha_i(t),n^t_{\alpha_i(t)}}(t) - C_{cmp} \sum_{i=1}^{M} m^i_{cmp}(T) - C_{swc} \sum_{i=1}^{M} m^i_{swc}(T) \right], \tag{7.5}$$

where $m^i_{cmp}(T)$ and $m^i_{swc}(T)$ denote the number of computations of the optimal allocation, and the number of resource switchings by agent i by time T, respectively. With this definition, the problem becomes balancing the loss in the performance and the loss due to the NP-hard computation and switchings that results from changes in strategy. We will denote by $\mathcal{O}_i(t)$ the set of resources that are used by at least one agent in the estimated optimal allocation of agent i, by $N^i_{k,n}(t)$ the number of times agent i selected resource k and observed n agents using it by time t, and $\hat{\mu}^i_{k,n}(t)$ the sample mean of the rewards collected by agent i from resource-activity pair (k, n) by its t-th observation of that pair.

7.1.2.2 Agent-specific Interaction Model

Consider the agent-specific interaction model given in Definition I.15. For a resource-activity pair (k, n), the mean reward perceived by agent i is given by

$$\mu^i_{k,n} := \int_{S^k} r^i_k(x, n) P^k(dx),$$

when the resource rewards evolve according to an IID model given in Definition I.1, and

$$\mu_{k,n}^{i} := \sum_{x \in S^k} r_k^i(x,n)\pi_x^k,$$

when the resource rewards evolve according to the uncontrolled Markovian model given in Definition I.4. Similar to the definitions in the previous subsection, $v(\boldsymbol{a})$ is the value of allocation $\boldsymbol{a}$, v^* is the value of an optimal allocation, $\Delta(\boldsymbol{a})$ is the suboptimality gap of allocation $\boldsymbol{a}$ and $\Delta_{\min}$ is the minimum suboptimality gap, i.e.,

$$\Delta_{\min} := v^* - \arg\max_{\boldsymbol{a} \in \mathcal{A} - \mathcal{A}^*} \Delta(\boldsymbol{a}),$$

where $\mathcal{A}^*$ is the set of optimal allocations. For a learning algorithm

$$\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_M),$$

we consider the weak regret in the agent-specific interaction model given in Definition I.15, which is

$$\textbf{III:}\ \ R^{\alpha}(T) := Tv^* - E_{\alpha}^{\boldsymbol{P}}\left[\sum_{t=1}^{T}\sum_{i=1}^{M} r^{i}_{\alpha_i(t),n^t_{\alpha_i(t)}}(t)\right]. \tag{7.6}$$

In order to estimate the optimal allocation, an agent must know how resource qualities are perceived by the other agents. This is not possible in general, since the reward function of an agent is its private information which is not known by the other agents. Therefore, when designing online learning algorithms for agent-specific resource rewards, we assume that communication between the agents is possible. This can either be done by broadcasting every time communication is needed, or broadcasting the next time to communicate on a specific channel initially, and then

using time division multiple access on that channel to transmit information about the resource estimates, and next time step to communicate. Every time an agent communicates with other agents, it incurs cost C_{com}. Considering the computation cost C_{cmp} and the switching cost C_{swc} the regret at time T becomes

$$\textbf{IV:}\ R^{\alpha}(T) = Tv^{*} - E_{\alpha}^{\boldsymbol{P}}\left[\sum_{t=1}^{T}\sum_{i=1}^{M} r^{i}_{\alpha_i(t),n^{t}_{\alpha_i(t)}}(t) - C_{cmp}\sum_{i=1}^{M} m^{i}_{cmp}(T) - C_{swc}\sum_{i=1}^{M} m^{i}_{swc}(T) - C_{com}\sum_{i=1}^{M} m^{i}_{com}(T)\right], \tag{7.7}$$

where $m^{i}_{com}(T)$ is the number of times agent i communicated (exchanged information) with other agents by time T.

The following stability condition is the analogue of Lemma VII.4. Note that due to sharing of information we do not require uniqueness of the number-optimal allocations under the costly-communication model. The agents can coordinate on an allocation to be played by communication.

Lemma VII.5. ***Stability.*** *For a set of estimated mean rewards $\hat{\mu}^{i}_{k,n}$, if $|\hat{\mu}^{i}_{k,n} - \mu^{i}_{k,n}| < \Delta_{\min}/(2M)$, $\forall i \in \mathcal{M}, k \in \mathcal{K}, n \in \mathcal{M}$ then,*

$$\arg\max_{\boldsymbol{a}\in\mathcal{A}} \sum_{i=1}^{M} \hat{\mu}^{i}_{a_i,n_{a_i}(\boldsymbol{a})} = \mathcal{A}^{*}.$$

Proof. The proof is similar to the proof of Lemma VII.4, noting that the value of each allocation is the sum of M resource-activity pairs. □

7.2 Achievable Performance without Feedback

We begin with the scenario of IID resources given in Definition I.1 and the no feedback model given in Model I.16; there is no communication among agents, and they cannot differentiate the effect of congestion from that of resource condition. The

resource rewards are in separable form, i.e., $r_k(s,n) = q_k(s)g_k(n)$, for some functions q_k and g_k. Note that two examples of this type of resource rewards are given earlier in Examples VII.1 and VII.2. We assume that each agent selects resources according to the Exp3 algorithm proposed in *Auer et al.* (2003), whose pseudocode is reproduced in Figure 7.1. Exp3 is a randomized algorithm, whereby each resource has some probability of being chosen, based on the history of resource selections and observed rewards. The probability of agent i choosing resource k depends on the exploration constant γ and weights w_{ik} that depend exponentially on past observations. Even though each agent runs an instance of Exp3 independently, as resource rewards are affected by the number of agents selecting a resource, every agent's action affects all other agents' subsequent actions.

At each time step t before the resource state and agent actions are drawn from their respective distributions, let $Q_k(t) = q_k(S_t^k)$ denote the random variable corresponding to the single-occupancy reward of the kth resource, where S_t^k is the random variable corresponding to the state of resource k at time t. Let $G_{ik}(t) = g_k(1 + N_k^i(t))$ be the random variable representing the reward or payoff agent i gets from resource k where $N_k^i(t)$ is the random variable representing the number of agents on resource k other than agent i. Let $U_{ik}(t) = Q_k(t)G_{ik}(t)$ and $\bar{u}_{ik}(t) = E_k[E_{-i}[U_{ik}(t)]]$ be the expected payoff to agent i by using resource k where E_{-i} represents the expectation taken with respect to the randomization of the agents other than i, E_k represents the expectation taken with respect to the random state realization of resource k. Since the resource reward is in separable form, we have $\bar{u}_{ik}(t) = \bar{q}_k(t)\bar{g}_{ik}(t)$ where $\bar{q}_k(t) = E[Q_k(t)]$ and $\bar{g}_{ik}(t) = E_{-i}[G_{ik}(t)]$.

We are interested in the asymptotic performance of agents when they are all using the Exp3 algorithm. We will show that the resource selection probabilities of a single agent converges to a point, where only a single resource will be selected with very high probability, while all other resources have a very small probability

(proportional to γ/K) to be selected. We prove this by writing the dynamics of Exp3 as a replicator equation, and showing that this replicator equation converges to a set of points which is equivalent to a pure Nash equilibrium of a congestion game played by the agents when all agents have complete knowledge about the mean payoffs and resource selections of all other agents. As noted earlier, the agents in our decentralized system are not assumed to be *strategic*. The above equivalence simply suggests that their asymptotic behavior coincides with an equilibrium point of a well-defined game.

Below, we give definitions of the replicator equation, the congestion game, and pure Nash equilibrium.

The replicator equation is widely studied in evolutionary game theory, such as in *Sandholm* (2011); *Smith* (1982); it models the dynamics of the survival of a particular type in a population. Intuitively, if a type yields high rewards, then the proportion of members in the population which has the characteristics of that type will increase over time. Consider the distribution vector of a population $\boldsymbol{x} = (x_1, x_2, \ldots, x_K)$, where x_k denotes the ratio of type k members of the population. The replicator equation is given by

$$\dot{x}_k = x_k(f_k(x) - \bar{f}(x)),$$

where f_k denotes the fitness of type k, which can be viewed as the survival rate of x_k in $\boldsymbol{x}$, and

$$\bar{f}(x) = \sum_{k=1}^{K} x_k f_k(x),$$

is the average population fitness.

A congestion game, which is defined in *Monderer and Shapley* (1996); *Rosenthal* (1973), (with agent independent payoffs) is given by the tuple $(\mathcal{M}, \mathcal{K}, (\Sigma_i)_{i\in\mathcal{M}}, (h_k)_{k\in\mathcal{K}})$, where $\mathcal{M}$ denotes a set of players (agents), $\mathcal{K}$ a set of resources (resources), $\Sigma_i \subset 2^{\mathcal{K}}$

the strategy space of player i, and $h_k : \mathbb{N} \to \mathbb{R}$ a payoff function associated with resource k, which is a function of the number of players using that resource. In essence, a congestion game models the resource competition among a set of agents, where the presence of an agent poses a negative externality to other agents.

Consider a strategy profile $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \ldots, \sigma_M)$ for the players in a congestion game $(\mathcal{M}, \mathcal{K}, (\Sigma_i)_{i\in\mathcal{M}}, (h_k)_{k\in\mathcal{K}})$. Let $(\boldsymbol{\sigma}_{-i}, \sigma_i')$ denote the strategy profile in which player i's strategy is σ_i', while any player $j \neq i$ has strategy σ_j. A pure Nash equilibrium of the congestion game is any strategy profile $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \ldots, \sigma_M)$ such that for $\sigma_i \in \mathcal{K}$ and for all $i \in \mathcal{M}$, we have

$$h_{\sigma_i}(n_{\sigma_i}(\boldsymbol{\sigma})) \geq h_{\sigma_i'}(n_{\sigma_i'}((\boldsymbol{\sigma}_{-i}, \sigma_i'))),$$

for any $i \in \mathcal{M}, \sigma_i' \in \mathcal{K}$, where $n_{\sigma_i}(\boldsymbol{\sigma})$ denotes the number of agents using σ_i under profile $\boldsymbol{\sigma}$. This means that there exists no player who can unilaterally deviate from $\boldsymbol{\sigma}$ and increase its payoff.

It is well known that the above congestion game (with agent independent payoff) is an exact potential game with an exact potential function, a local maxima of the potential function corresponds to a pure Nash equilibrium (PNE), and every sequence of asynchronous improvement steps is finite and converges to a PNE.

The next lemma shows that the evolution of the resource selection probabilities under Exp3 in time can be written as a replicator equation.

Lemma VII.6. *When all agents use Exp3, the derivative of the continuous-time limit of Exp3 is the replicator equation given by*

$$\chi_{ik} = \frac{1}{K}(\bar{q}_k p_{ik}) \sum_{l=1}^{K} p_{il}(\bar{g}_{ik} - \bar{g}_{il}) \ .$$

Exp3 (for agent i)

Initialize: $\gamma \in (0,1)$, $w_{ik}(t) = 1, \forall k \in \mathcal{K}$, $t = 1$
while $t > 0$ **do**

$$p_{ik}(t) = (1-\gamma)\frac{w_{ik}(t)}{\sum_{l=1}^{K} w_{il}(t)} + \frac{\gamma}{K}$$

  Sample $\sigma_i(t)$ according to the distribution $p_i(t) = [p_{i1}(t), p_{i2}(t), \ldots, p_{iK}(t)]$
  Select resource $\sigma_i(t)$ and receive reward $U_{i,\sigma_i(t)}(t)$
  for $k = 1, 2, \ldots, K$ **do**
    if $k = \sigma_i(t)$ **then**
      Set $w_{ik}(t+1) = w_{ik}(t) \exp\left(\frac{\gamma U_{i,\sigma_i(t)}(t)}{p_{ik}(t)K}\right)$
    else
      Set $w_{ik}(t+1) = w_{ik}(t)$
    end if
  end for
  $t = t + 1$
end while

Figure 7.1: pseudocode of Exp3

Proof. By the definition of the Exp 3 algorithm we have

$$(1-\gamma)w_{ik}(t) = \sum_{l=1}^{K} w_{il}(t)\left(p_{ik}(t) - \frac{\gamma}{K}\right) \ . \tag{7.8}$$

Now consider the effect of agent i's action $\sigma_i(t)$ on his probability update on resource k. We have two cases: $\sigma_i(t) = k$ and $\sigma_i(t) \neq k$. Let $A_{i,k}^{\gamma,t} = \exp\left(\frac{\gamma U_{ik}(t)}{p_{ik}(t)K}\right)$.

Consider the case $\sigma_i(t) = k$:

$$p_{ik}(t+1) = \frac{(1-\gamma)w_{ik}(t)A_{i,k}^{\gamma,t}}{\sum_{l=1}^{K} w_{il}(t) + w_{ik}(t)\left(A_{i,k}^{\gamma,t} - 1\right)} + \frac{\gamma}{K} \ . \tag{7.9}$$

Substituting (7.8) into (7.9), we get

$$p_{ik}(t+1) \ = \ \frac{\sum_{l=1}^{K} w_{il}(t)\left(p_{ik}(t) - \frac{\gamma}{K}\right)A_{i,k}^{\gamma,t}}{\sum_{l=1}^{K} w_{il}(t)\left(1 + \frac{p_{ik}(t) - \frac{\gamma}{K}}{1-\gamma}\left(A_{i,k}^{\gamma,t} - 1\right)\right)} + \frac{\gamma}{K}$$

$$= \frac{\left(p_{ik}(t) - \frac{\gamma}{K}\right) A_{i,k}^{\gamma,t}}{1 + \frac{p_{ik}(t) - \frac{\gamma}{K}}{1-\gamma}\left(A_{i,k}^{\gamma,t} - 1\right)} + \frac{\gamma}{K} .$$

The continuous time process is obtained by taking the limit $\gamma \to 0$, i.e., the rate of change in p_{ik} with respect to γ as $\gamma \to 0$. Then, dropping the discrete time script t, we have

$$\begin{aligned}
\dot{p_{ik}} &= \lim_{\gamma \to 0} \frac{dp_{ik}}{d\gamma} \\
&= \lim_{\gamma \to 0} \frac{\left(\frac{-1}{K} A_{i,k}^{\gamma,t} + \left(p_{ik} - \frac{\gamma}{K}\right) \frac{U_{ik}}{p_{ik} K} A_{i,k}^{\gamma,t}\right)\left(1 + \frac{p_{ik} - \frac{\gamma}{K}}{1-\gamma}\left(A_{i,k}^{\gamma,t} - 1\right)\right)}{\left(1 + \frac{p_{ik} - \frac{\gamma}{K}}{1-\gamma}\left(A_{i,k}^{\gamma,t} - 1\right)\right)^2} \\
&+ \frac{\left(p_{ik} - \frac{\gamma}{K}\right) A_{i,k}^{\gamma,t}\left(\frac{p_{ik} - \frac{1}{K}}{(1-\gamma)^2} A_{i,k}^{\gamma,t} + \frac{p_{ik} - \frac{1}{K}}{1-\gamma}\left(\frac{\gamma}{K} A_{i,k}^{\gamma,t}\right)\right)}{\left(1 + \frac{p_{ik} - \frac{\gamma}{K}}{1-\gamma}\left(A_{i,k}^{\gamma,t} - 1\right)\right)^2} + \frac{1}{K} \\
&= \frac{U_{ik}(1 - p_{ik})}{K} . \qquad (7.10)
\end{aligned}$$

Consider the case $\sigma_i(t) = \bar{k} \neq k$:

$$\begin{aligned}
p_{ik}(t+1) &= \frac{(1-\gamma) w_{ik}(t)}{\sum_{l=1}^{K} w_{il}(t) + w_{i\bar{k}}(t)\left(A_{i,\bar{k}}^{\gamma,t} - 1\right)} + \frac{\gamma}{K} \\
&= \frac{p_{ik}(t) - \frac{\gamma}{K}}{1 + \frac{p_{i\bar{k}}(t) - \frac{\gamma}{K}}{1-\gamma}\left(A_{i,\bar{k}}^{\gamma,t} - 1\right)} + \frac{\gamma}{K}.
\end{aligned}$$

Thus

$$\begin{aligned}
\dot{p_{ik}} &= \lim_{\gamma \to 0} \frac{\frac{-1}{K}\left(1 + \frac{p_{i\bar{k}} - \frac{\gamma}{K}}{1-\gamma}\left(A_{i,\bar{k}}^{\gamma,t} - 1\right)\right)}{\left(1 + \frac{p_{i\bar{k}} - \frac{\gamma}{K}}{1-\gamma}\left(A_{i,\bar{k}}^{\gamma,t} - 1\right)\right)^2} \\
&+ \frac{\left(p_{ik} - \frac{\gamma}{K}\right)\left(\frac{p_{i\bar{k}} - \frac{1}{K}}{(1-\gamma)^2} A_{i,\bar{k}}^{\gamma,t} + \frac{p_{i\bar{k}} - \frac{1}{K}}{1-\gamma}\left(\frac{\gamma}{K} A_{i,\bar{k}}^{\gamma,t}\right)\right)}{\left(1 + \frac{p_{i\bar{k}} - \frac{\gamma}{K}}{1-\gamma}\left(A_{i,\bar{k}}^{\gamma,t} - 1\right)\right)^2} + \frac{1}{K} \\
&= -\frac{p_{i\bar{k}} U_{i\bar{k}}}{K} . \qquad (7.11)
\end{aligned}$$

Then from (7.10) and (7.11), the expected change in p_{ik} with respect to the probability distribution p_i of agent i over the resources is

$$E_i[\dot{p}_{ik}] = \frac{1}{K} p_{ik} \sum_{l \in \mathcal{K}-\{k\}} p_{il}(U_{ik} - U_{il}).$$

Taking the expectation with respect to the randomization of resource rates and other agents' actions we have

$$\begin{aligned} \chi_{ik} &= E_k[E_{-i}[E_i[\dot{p}_{ik}]]] \\ &= \frac{1}{K} p_{ik} \sum_{l \in \mathcal{K}-\{j\}} p_{il} \left(E_k[E_{-i}[U_{ik}]] - E_k[E_{-i}[U_{il}]] \right) \\ &= \frac{1}{K} (\bar{q}_k p_{ik}) \sum_{l=1}^{K} p_{il} (\bar{g}_{ik} - \bar{g}_{il}) \ . \end{aligned}$$

□

Lemma VII.6 shows that the dynamics of an agent's probability distribution over the actions is given by a replicator equation which is commonly studied in evolutionary game theory, such as in *Sandholm* (2011); *Smith* (1982). With this lemma we can establish the following theorem.

Theorem VII.7. *For all but a measure zero subset of* $[0,1]^{2K}$ *from which the* $\bar{q}_k$*'s and* g_k*'s are selected, when* γ *in Exp3 is arbitrarily small, the action profile converges to the set of PNE of the congestion game* $(\mathcal{M}, \mathcal{K}, (\mathcal{S}_i)_{i \in \mathcal{M}}, (\bar{q}_k g_k)_{k \in \mathcal{K}})$.

Proof. Because the equation in Lemma VII.6 is identical to the replicator equation in *Kleinberg et al.* (2009), the proof of convergence to a PNE follows from *Kleinberg et al.* (2009). Here, we briefly explain the steps in the proof. Using a *potential function* approach it can be shown that the solutions to the replicator equation converge to the set of fixed points. Then, the stability analysis using the Jacobian matrix yields that every stable fixed point corresponds to a NE. Then, one can prove that for any stable

fixed point the eigenvalues of the Jacobian must be zero. This implies that every stable fixed point corresponds to a *weakly stable* NE strategy in the game theoretic sense. Then using tools from algebraic geometry one can show that almost every weakly stable NE is a PNE of the congestion game.

We also need to investigate the error introduced by treating the discrete time update rule as a continuous time process. However, by taking γ infinitesimal we can approximate the discrete time process by the continuous time process. For a discussion when γ is not infinitesimal one can define *approximately stable equilibria* as given in *Kleinberg et al.* (2009). □

The main difference between Exp3 and Hedge algorithm proposed in *Kleinberg et al.* (2009) is that in Exp3 agents do not need to observe the payoffs from the resources that they do not select, whereas Hedge assumes complete observation. In addition, in our analysis we have considered dynamic resource states which is not considered in *Kleinberg et al.* (2009).

In this section we showed that convergence is possible under a completely decentralized setting. The equilibrium may be suboptimal compared to the allocation that maximizes the sum of expected rewards of all agents. The inefficiency of the equilibrium can be measured by using the notion of *price of anarchy*. However, owing to the construction of Exp3, each agent is guaranteed sublinear regret with respect to the worst-case reward distribution. Within this context if we define the regret of an agent as the difference between the expected total reward the agent can obtain by always selecting the best resource, calculated based on the IID resource rewards and conditioned on the random resource selection by other agents, and the expected total reward of the agent by using Exp3[2], then a result from *Auer et al.* (2003) shows that this regret is $O(\sqrt{T})$ for all agents.

[2]Note that this is *not* the same as the weak regret measure used everywhere else in this chapter, which is with respect to the optimal static allocations for all agents in the system.

7.3 Achievable Performance with Partial Feedback

In this section we study the scenario of IID resources given in Definition I.1 and the partial feedback model given in Model I.18. We propose an algorithm, the Randomized Learning with Occupancy Feedback (RLOF), whose weak regret with respect to the optimal static allocation is $O(T^{\frac{2M-1+2\gamma}{2M}})$ for $\gamma > 0$ arbitrarily small. Clearly, this regret is sublinear (it approaches linear as the number of agents M increases). This means that the time average of the sum of rewards of all agents converges to the average reward of the optimal static allocation.

Each agent independently runs an instance of RLOF; its pseudocode is given in Figure 7.2. In running RLOF an agent keeps sample mean estimates of the rewards for each resource-activity pair. A time step t is either assigned as an exploration step with probability $1/(t^{\frac{1}{2M}-\frac{\gamma}{M}})$, or an exploitation step with probability $1-1/(t^{\frac{1}{2M}-\frac{\gamma}{M}})$. In an exploration step, the agent explores by randomly choosing one of the resources. If time t is an exploitation step for agent i, it exploits by first calculating an estimated optimal allocation $\hat{\boldsymbol{n}}^i(t) = \{\hat{n}^i_1(t), \ldots, \hat{n}^i_K(t)\}$ based on the sample mean reward estimates of the resource-activity pairs given by

$$\hat{\boldsymbol{n}}^i(t) = \arg\max_{\boldsymbol{n}\in\mathcal{N}} \sum_{k=1}^{K} n_k \hat{\mu}_{k,n_k}(N^i_{k,n_k}(t)),$$

and then selecting a resource from the set $\mathcal{O}_i(t)$ which is the set of resources selected by at least one agent in $\hat{\boldsymbol{n}}^i(t)$. $N^i_{k,n}(t)$ which is defined in Section 7.1, denotes the number of times agent i selected resource k and observed n agents on it by time t.

When choosing a resource from the set $\mathcal{O}_i(t)$, agent i follows a specific rule so that the joint resource selections by all agents can converge to the optimal allocation if all agents have correctly estimated the optimal allocation. If $\alpha_i(t-1) \in \mathcal{O}_i(t)$ and $n^{t-1}_{\alpha_i(t-1)} \leq \hat{n}^i_{\alpha_i(t-1)}(t)$ (i.e., the actual occupancy/congestion level in resource $\alpha_i(t-1)$ is below or at the estimated optimal congestion level), agent i will remain in the

resource it selected in the previous time slot, i.e., $\alpha_i(t) = \alpha_i(t-1)$. Otherwise, agent i randomizes within $\mathcal{O}_i(t)$: it selects resource $k \in \mathcal{O}_i(t)$ with probability $\hat{n}_k^i(t)/M$. Note that due to this randomization there may be a period of time in which the collective actions of all agents are not optimal even though they each has the correct estimated optimal allocation. This type of randomization guarantees that when agents have estimated the optimal allocation correctly in consecutive time steps, they will converge to the optimal allocation in finite expected time.

For notational convenience we will let $l_i(t-1) = n_{\alpha_i(t-1)}^{t-1}$. The following lemma on partial sum of series will be useful in the proof of the main theorem of this section.

Lemma VII.8. *For $p > 0, p \neq 1$*

$$\frac{(T+1)^{1-p} - 1}{1-p} < \sum_{t=1}^{T} \frac{1}{t^p} < 1 + \frac{T^{1-p} - 1}{1-p} \tag{7.12}$$

Proof. See *Chlebus* (2009). □

There are three factors contributing to the regret. The first is the regret due to exploration steps, the second is the regret due to incorrect computation of the optimal allocation by some agent, and the third is the regret due to the randomization steps after each agent has computed the optimal allocation correctly, in which at least one agent randomizes its selection due to higher-than-optimal congestion level in its current resource.

In order to provide a bound on the regret of RLOF, we first bound the expected number of time steps in which there exists at least one agent who computed the socially optimal allocation incorrectly.

Lemma VII.9. *When all agents use RLOF with parameter $\gamma > 0$, the expected number of time steps by time T, in which there exists at least one agent who computed*

Randomized Learning with Occupancy Feedback (RLOF) for agent i

Initialize: $0 < \gamma << 1$, $\hat{\mu}^i_{k,n} = 0, N^i_{k,n} = 0, \forall k \in \mathcal{K}, n \in \{1, 2, \ldots, M\}$, $t = 1$.
while $t \geq 1$ **do**
  Draw i_t randomly from Bernoulli distribution with $P(i_t = 1) = \frac{1}{t^{(1/2M)-\gamma/M}}$.
  if $i_t = 0$ **then**
    Compute the estimated optimal allocation.
    $\hat{\boldsymbol{n}}^i = \arg\max_{\boldsymbol{n} \in \mathcal{N}} \sum_{k=1}^{K} n_k \hat{\mu}_{k,n_k}$.
    Set $\mathcal{O}_i$ to be the set of resources in $\hat{\boldsymbol{n}}^i$ with at least one agent.
    if $l_i(t-1) > \hat{n}^i_{\alpha_i(t-1)}$ **then**
      Pick $\alpha_i(t)$ randomly from $\mathcal{O}_i$ with $P(\alpha_i(t) = k) = \hat{n}^i_k / M$.
    else
      $\alpha_i(t) = \alpha_i(t-1)$
    end if
  else
    Select $\alpha_i(t)$ uniformly at random from $\mathcal{K}$.
  end if
  Select resource $\alpha_i(t)$, observe $l_i(t)$ the number of agents using resource $\alpha_i(t)$ and the reward $r_{\alpha_i(t),l_i(t)}(t)$.
  Set $N^i_{\alpha_i(t),l_i(t)} = N^i_{\alpha_i(t),l_i(t)} + 1$.
  Set $\hat{\mu}^i_{\alpha_i(t),l_i(t)} = \frac{(N^i_{\alpha_i(t),l_i(t)} - 1)\hat{\mu}^i_{\alpha_i(t),l_i(t)} + r_{\alpha_i(t),l_i(t)}(t)}{N^i_{\alpha_i(t),l_i(t)}}$.
  $t = t + 1$
end while

Figure 7.2: pseudocode of RLOF

the optimal allocation incorrectly is upper bounded by

$$M^2 K \left(\tau(M, K, \Delta_{\min}, \gamma) + 3\beta\right),$$

where $\beta = \sum_{t=1}^{\infty} \frac{1}{t^2}$, τ *is a number which depends on* M, K, $\Delta_{\min}$ *and* γ, *and* τ *is non-decreasing in* $\Delta_{\min}$ *and non-increasing in* γ.

Proof. Let $H(t)$ be the event that at time t there exists at least one agent that computed the socially optimal allocation incorrectly. Let $\epsilon = \Delta_{\min}/2$, and let ω

denote a sample path. Then

$$
\begin{aligned}
&\sum_{t=1}^{T} I(\omega \in H(t)) \\
&\leq \sum_{t=1}^{T} \sum_{i=1}^{M} I(\hat{\boldsymbol{n}}^i(t) \neq \boldsymbol{n}^*) \\
&\leq \sum_{(t,i,k,n)=(1,1,1,1)}^{(T,M,K,M)} I(|\hat{\mu}^i_{k,n}(N^i_{k,n}(t)) - \mu_{k,n}| \geq \epsilon) \\
&= \sum_{(t,i,k,n)=(1,1,1,1)}^{(T,M,K,M)} I\left(|\hat{\mu}^i_{k,n}(N^i_{k,n}(t)) - \mu_{k,n}| \geq \epsilon, N^i_{k,n}(t) \geq \frac{a \ln t}{\epsilon^2}\right) \\
&+ \sum_{(t,i,j,l)=(1,1,1,1)}^{(T,M,K,M)} I\left(|\hat{\mu}^i_{k,n}(N^i_{k,n}(t)) - \mu_{k,n}| \geq \epsilon, N^i_{k,n}(t) < \frac{a \ln t}{\epsilon^2}\right), \qquad (7.13)
\end{aligned}
$$

for some $a > 0$. Let $\epsilon^i_{k,n}(t) = \sqrt{\frac{a \ln t}{N^i_{k,n}(t)}}$. Then, we have

$$
N^i_{k,n}(t) \geq \frac{a \ln t}{\epsilon^2} \Rightarrow \epsilon \geq \sqrt{\frac{a \ln t}{N^i_{k,n}(t)}} = \epsilon^i_{k,n}(t).
$$

Therefore,

$$
\begin{aligned}
&I\left(|\hat{\mu}^i_{k,n}(N^i_{k,n}(t)) - \mu_{k,n}| \geq \epsilon, N^i_{k,n}(t) \geq \frac{a \ln t}{\epsilon^2}\right) \\
&\leq I\left(|\hat{\mu}^i_{k,n}(N^i_{k,n}(t)) - \mu_{k,n}| \geq \epsilon^i_{k,n}(t)\right) \ ,
\end{aligned}
$$

and

$$
I\left(|\hat{\mu}^i_{k,n}(N^i_{k,n}(t)) - \mu_{k,n}| \geq \epsilon, N^i_{k,n}(t) < \frac{a \ln t}{\epsilon^2}\right) \leq I\left(N^i_{k,n}(t) < \frac{a \ln t}{\epsilon^2}\right) \ .
$$

Then, continuing from (7.13),

$$
\sum_{t=1}^{T} I(\omega \in H(t))
$$

$$\leq \sum_{(t,i,k,n)=(1,1,1,1)}^{(T,M,K,M)} \left(I\left(|\hat{\mu}^i_{k,n}(N^i_{k,n}(t)) - \mu_{k,n}| \geq \epsilon^i_{k,n}(t)\right) + I\left(N^i_{k,n}(t) < \frac{a \ln t}{\epsilon^2}\right)\right). \tag{7.14}$$

Taking the expectation over (7.14),

$$E\left[\sum_{t=1}^{T} I(\omega \in H(t))\right] \leq \sum_{(t,i,k,n)=(1,1,1,1)}^{(T,M,K,M)} P\left(|\hat{\mu}^i_{k,n}(N^i_{k,n}(t)) - \mu_{k,n}| \geq \epsilon^i_{k,n}(t)\right) + \sum_{(t,i,k,n)=(1,1,1,1)}^{(T,M,K,M)} P\left(N^i_{k,n}(t) < \frac{a \ln t}{\epsilon^2}\right). \tag{7.15}$$

We have

$$\begin{aligned} & P\left(|\hat{\mu}^i_{k,n}(N^i_{k,n}(t)) - \mu_{k,n}| \geq \epsilon^i_{k,n}(t)\right) \\ = \;& P\left(\hat{\mu}^i_{k,n}(N^i_{k,n}(t)) - \mu_{k,n} \geq \epsilon^i_{k,n}(t)\right) + P\left(\hat{\mu}^i_{k,n}(N^i_{k,n}(t)) - \mu_{k,n} \leq -\epsilon^i_{k,n}(t)\right) \\ \leq \;& 2\exp\left(-\frac{2(N^i_{k,n}(t))^2(\epsilon^i_{k,n}(t))^2}{N^i_{k,n}(t)}\right) = 2\exp\left(-\frac{2N^i_{k,n}(t)a\ln t}{N^i_{k,n}(t)}\right) = \frac{2}{t^{2a}}\ , \end{aligned} \tag{7.16}$$

where (7.16) follows from a Chernoff-Hoeffding inequality.

We next bound $P\left(N^i_{k,n}(t) < \frac{a\ln t}{\epsilon^2}\right)$. Let $TR^i_{k,n}(t)$ be the number of time steps in which agent i picked resource k and observed n agents on resource k among the time steps in which all agents explored up to time t. Then

$$\{\omega : N^i_{k,n}(t) < \frac{a \ln t}{\epsilon^2}\} \subset \{\omega : TR^i_{k,n}(t) < \frac{a \ln t}{\epsilon^2}\}.$$

Hence

$$P\left(N^i_{k,n}(t) < \frac{a \ln t}{\epsilon^2}\right) \leq P\left(TR^i_{k,n}(t) < \frac{a \ln t}{\epsilon^2}\right)\ . \tag{7.17}$$

We now define Bernoulli random variables $X^i_{k,n}(t)$ as follows: $X^i_{k,n}(t) = 1$ if all agents

explore at time t and agent i selects resource k and observes n agents on it; $X^i_{k,n}(t) = 0$ otherwise. Then $TR^i_{k,n}(t) = \sum_{\zeta=1}^{t} X^i_{k,n}(\zeta)$. $P(X^i_{k,n}(\zeta) = 1) = \rho_\zeta p_l$ where $p_l = \frac{\binom{M-1}{n-1}\binom{M+K-n-2}{K-2}}{\binom{M+K-1}{K-1}}$ and $\rho_\zeta = \frac{1}{\zeta^{(1/2)-\gamma}}$. Let $\zeta_t = \sum_{\zeta=1}^{t} \frac{1}{\zeta^{(1/2)-\gamma}}$ Then

$$\begin{aligned}
&P\left(TR^i_{k,n}(t) < \frac{a\ln t}{\epsilon^2}\right) \\
&= P\left(\frac{TR^i_{k,n}(t)}{t} - \frac{p_n\zeta_t}{t} < \frac{a\ln t}{t\epsilon^2} - \frac{p_n\zeta_t}{t}\right) \\
&\leq P\left(\frac{TR^i_{k,n}(t)}{t} - \frac{p_n\zeta_t}{t} < \frac{a\ln t}{t\epsilon^2} - \frac{p_n(t+1)^{(1/2)+\gamma} - 1}{t((1/2)+\gamma)}\right), \qquad (7.18)
\end{aligned}$$

where (7.18) follows from Lemma VII.8. Let $\tau(M, K, \epsilon, \gamma, \gamma', a)$ be the time that for all $n \in \{1, 2, \ldots, M\}$,

$$\frac{p_n(t+1)^{(1/2)+\gamma} - 1}{t((1/2)+\gamma)} - \frac{a\ln t}{t\epsilon^2} \geq t^{(1/2)+\gamma'}, \qquad (7.19)$$

where $0 < \gamma' < \gamma$. Then for all $t \geq \tau(M, K, \epsilon, \gamma, \gamma', a)$ (7.19) holds since the right hand side increases faster than the left hand side. Clearly, $\tau(M, K, \epsilon, \gamma, \gamma', a)$ is non-decreasing in $\Delta_{\min} = 2\epsilon$ and non-increasing in γ. Thus we have for $t \geq \tau(M, K, \epsilon, \gamma, \gamma', a)$

$$\begin{aligned}
&P\left(\frac{TR^i_{k,n}(t)}{t} - \frac{p_n\zeta_t}{t} < \frac{a\ln t}{t\epsilon^2} - \frac{p_n(t+1)^{(1/2)+\gamma} - 1}{t((1/2)+\gamma)}\right) \\
&\leq P\left(\frac{TR^i_{k,n}(t)}{t} - \frac{p_n\zeta_t}{t} < t^{-(1/2)+\gamma'}\right) \\
&\leq e^{-2tt^{2\gamma'-1}} = e^{-2t^{2\gamma'}} \leq e^{-2\ln t} = \frac{1}{t^2}. \qquad (7.20)
\end{aligned}$$

Let $a = 1$, and $\tau(M, K, \Delta_{\min}, \gamma) = \min_{0<\gamma'<\gamma} \tau(M, K, \epsilon, \gamma, \gamma', 1)$. Then continuing

from (7.15) by substituting (7.16) and (7.20) we have

$$E\left[\sum_{t=1}^{T} I(\omega \in H(t))\right] \leq M^2 K \left(\tau(M, K, \epsilon, \gamma, \gamma', 1) + 3\sum_{t=1}^{T} \frac{1}{t^2}\right). \tag{7.21}$$

Thus we have proved that the expected number of time steps in which there exists at least one agent that computed the socially optimal allocation incorrectly is finite. □

We next consider a second element in regret, which is the number of times an agent spends on exploration.

Lemma VII.10. *When all agents use RLOF with parameter $\gamma > 0$, the expected number of time steps by time T in which there exists at least one agent who explores is upper bounded by*

$$\frac{2M^2}{2(M+\gamma)-1}\left(1 + T^{\frac{2(M+\gamma)-1}{2M}}\right).$$

Proof. Since RLOF explores with probability $\frac{1}{t^{1/2M-\gamma/M}}$, the expected number of time steps up to time T in which at least one agent explores is

$$\begin{aligned}\sum_{t=1}^{T}\left(1 - \left(1 - \frac{1}{t^{(1/2M)-\gamma/M}}\right)^M\right) &\leq \sum_{t=1}^{T} \frac{M}{t^{1/2M-\gamma/M}} \\ &\leq \frac{2M^2}{2(M+\gamma)-1}\left(1 + T^{\frac{2(M+\gamma)-1}{2M}}\right),\end{aligned}$$

by Lemma VII.8. □

Now consider the time steps in which all agents compute the optimal allocation correctly. In these time steps each agent knows the optimal joint strategy (in terms of how many agents should use which resource), but due to lack of communication they can only try to reach this optimal allocation by randomizing their selections within the optimal resources based on the partial feedback. The next lemma states that

the expected number of time steps it takes to settle into optimal allocation given all agents have computed it correctly is finite.

Lemma VII.11. *Denote the number of resources which are selected by at least one agent in the optimal allocation by z^*. Reindex the resources in $\mathcal{O}^*$ by $\{1, 2, \ldots, z^*\}$. Let $\mathcal{N}' = \{\boldsymbol{m} : m_1 + m_2 + \ldots + m_{z^*} \leq M, m_i \geq 0, \forall i \in \{1, 2, \ldots, z^*\}\}$. When all agents use RLOF, given that all agents computed the optimal allocation correctly, the expected number of time steps before settling into the optimal allocation is upper bounded by*

$$O_B := \frac{1}{\min_{\boldsymbol{m} \in \mathcal{M}} P_{RLOF}(\boldsymbol{m})},$$

where

$$P_{RLOF}(\boldsymbol{m}) = \frac{(M-m)!}{(n_1 - m_1)! \ldots (n_{z^*} - m_{z^*})!} \left(\frac{n_1}{M}\right)^{n_1 - m_1} \ldots \left(\frac{n_{z^*}}{M}\right)^{n_{z^*} - m_{z^*}}.$$

Proof. We will refer to a sequence of exploitation steps of random length L starting at time t (i.e., $t, t+1, \ldots, t+L-1$) those in which all agents correctly calculate the optimal allocation as a *good* sequence of exploitation steps. This means that $\hat{\boldsymbol{n}}^i(t') = \boldsymbol{n}^*$ for all $i \in \mathcal{M}$, for all $t' \in \{t, t+1, \ldots, t+L-1\}$. Since agent i does not know the selections of other agents, knowing the optimal allocation is not sufficient to guarantee that the agents' joint selections are optimal. In these steps agent i remains on resource k it had chosen in the previous exploitation step if the occupancy it observed is no more than n_k^*. Otherwise, agent i selects resource k with probability n_k^*/M.

Note that $n_1 + n_2 + \ldots + n_{z^*} = M$. Consider the case where m of the agents do not randomize while the others randomize. Let $\boldsymbol{m} = (m_1, m_2, \ldots, m_{z^*})$ be the number of agents on each resource in $\mathcal{O}^*$ who do not randomize; we have $m = m_1 + m_2 + \ldots + m_{z^*}$. The probability of settling to the optimal allocation in a single step of randomization

is

$$P_{RLOF}(\boldsymbol{m}) = \frac{(M-m)!}{(n_1-m_1)!\dots(n_{z^*}-m_{z^*})!}\left(\frac{n_1}{M}\right)^{n_1-m_1}\cdots\left(\frac{n_{z^*}}{M}\right)^{n_{z^*}-m_{z^*}},$$

where $M!/(n_1^*!n_2^*!\dots n_{z^*}^*!)$ is the number of allocations that result in the unique optimal allocation in terms of number of agents using each resource, and

$$\left(\frac{n_1}{M}\right)^{n_1-m_1}\cdots\left(\frac{n_{z^*}}{M}\right)^{n_{z^*}-m_{z^*}}$$

is the probability that such an allocation happens. Then,

$$p_W := \min_{\boldsymbol{m}\in\mathcal{M}} P_{RLOF}(\boldsymbol{m}),$$

is the worst-case probability of settling to an optimal allocation in the next time step, given all agents estimated it correctly.

Let $\boldsymbol{n}_I$ be the allocation at the beginning of a good sequence of exploitation steps, in which all agents computed the optimal allocation correctly. Let $p_{\boldsymbol{n}_I}(t')$ be the probability of settling to the optimal allocation in the t'th round of randomization in this sequence. Then, the expected number of steps before settling to the optimal allocation given $L = l$ is

$$E[t_{op}|L=l] = \sum_{t'=1}^{l} t p_{\boldsymbol{n}_I}(t')\prod_{i=1}^{t'-1}(1-p_{\boldsymbol{n}_I}(i)).$$

Since $p_W \le p_{\boldsymbol{n}_I}(t)$ for all $\boldsymbol{n}_I$ and t, we have

$$E[t_{op}] \le \sum_{t'=1}^{\infty} t' p_W (1-p_W)^{t'-1} = 1/p_W.$$

□

Theorem VII.12. *When all agents use RLOF with parameter $\gamma > 0$, the regret (7.4)*

at time T is upper bounded by

$$MO_B\left(\frac{2M^2}{2(M+\gamma)-1}\left(1+T^{\frac{2(M+\gamma)-1}{2M}}\right)+M^2K(\tau+3\beta)\right),$$

where $\beta=\sum_{t=1}^{\infty}1/t^2$, $\tau=\tau(M,K,\Delta_{\min},\gamma)$ is as given in Lemma VII.9, and O_B is as given in Lemma VII.11. Asymptotically, the regret is

$$O(T^{\frac{2M-1+2\gamma}{2M}}),$$

where the parameter for RLOF, $\gamma>0$, can be chosen arbitrarily small (tradeoff between finite time and asymptotic regret).

Proof. We adopt a worst case analysis. We classify the time steps into two types: *good* time steps in which all the agents know the optimal allocation correctly and none of the agents randomize except for the randomization done for settling down to the optimal allocation, and *bad* time steps in which there exists an agent that does not know the optimal allocation correctly or there is an agent who explores. The expected number of *bad* time steps in which there exists an agent that does not know the optimal allocation correctly is upper bounded in Lemma VII.9, while the expected number of time steps in which there is an agent who explores is upper bounded in Lemma VII.10. The worst case is when each bad step is followed by a good step, resulting in repeated periods of randomization. Then from this good step, the expected number of times it takes to settle down to the optimal allocation is upper bounded by O_B which is given in Lemma VII.11. Since resource rewards are bounded within $[0,1]$, contribution of a single time step to the regret can be at most M. □

We mentioned earlier that, under a classical multi-armed bandit framework used in *Anandkumar et al.* (2011) and *Liu and Zhao* (2010), logarithmic regret ($O(\log T)$)

is achievable. The fundamental difference between these studies and the problem presented here is that we allow sharing of resources by agents. Without synchronization, the exploration rate should grow with the number of agents so that each agent i's estimate of resource-activity pair rewards are accurate enough. In the next section, we will show that logarithmic regret can be achieved if we allow synchronization between the agents.

Also note that in RLOF an agent computes the optimal allocation according to sample mean estimates of resource-activity pair rewards. This could pose significant computational requirement since integer programming is NP-hard in general. However, by exploiting the stability condition on the optimal allocation an agent may reduce the amount of computation. Since small perturbations of the expected reward of a resource-activity pair does not affect the optimal allocation, the computation of optimal allocation can be made only after sample mean reward of some resource-activity pair has significantly changed. This results in computation done at the end of geometrically increasing intervals. More is discussed in Section 7.6.

Compared to the result in Section 7.2, the feedback on the number of agents on the same resource significantly improves the performance. We not only designed an algorithm which leads to the optimal allocation asymptotically, but also proved a result on the rate of convergence of our algorithm. Although the regret is sublinear, it degrades quickly when the number of agents increase. This is because each resource-activity pair needs to be sampled sufficiently by each agent in order for all agents to estimate the optimal allocation correctly with high probability. Due to the randomized nature of the algorithm, the probability of an agent exploring a particular resource-activity pair (k, n) is small when M is large, thus the exploration probability should increase with M. We address this issue in the next section by assuming that the agents can agree on the exploration order at the beginning.

7.4 Achievable Performance with Partial Feedback and Synchronization

In this section we consider the partial feedback with initial synchronization model given in Model I.19, under both the IID and Markovian resource models given in Definitions I.1 and I.4, respectively. We propose a distributed algorithm which achieves logarithmic regret with respect to the optimal static allocation when agents are able to observe the number of simultaneous agents on a resource, and can initially synchronize their explorations.

This synchronization, which is absent in RLOF given in the previous section, is sufficient to improve the regret bound for IID resources. The algorithm is called *Distributed Learning with Ordered Exploration* (DLOE); its pseudocode is given in Figure 7.3. DLOE uses the idea of *deterministic sequencing of exploration and exploitation with initial synchronization* which was first proposed in *Liu et al.* (2011) to achieve logarithmic regret for multi-agent learning with Markovian rewards. A key difference between the problem studied here and that in *Liu et al.* (2011) is that the latter assumes that the optimal allocation of agents to resources is orthogonal, i.e., there can be at most one agent using a resource under the optimal allocation. For this reason the technical development in *Liu et al.* (2011) is not applicable to the reward model introduced in this chapter, which depends on the activity level in a resource.

DLOE operates in blocks. Each block is either an exploration block or an exploitation block. An agent running DLOE forms estimates of the resource-activity pair rewards in exploration blocks, and uses these to compute an estimated optimal allocation at the beginning of each exploitation block. One important property of DLOE is that only the observations from exploration blocks are used to form estimates of resource-activity pair rewards. While for the IID resource model, observations from exploitation blocks can also be used to update the estimates, this

will create problems in the Markovian resource model, since in order to capture the average quality accurately, a resource-activity pair should be observed in contiguous segments, which may not be always possible in exploitation blocks due to the lack of communication between the agents. Therefore, in this section $N^i_{k,n}(t)$ denotes the number of times resource k is selected and n agents are observed on resource k by agent i in exploration blocks of agent i by time t.

The length of an exploration (exploitation) block geometrically increases in the number of exploration (exploitation) blocks. The parameters that determine the length of the blocks is the same for all agents. Agent i has a fixed exploration sequence $\mathcal{N}_i$, which is a sequence of resources of length N'. In its lth exploration block agent i selects resources in $\mathcal{N}_i$ in a sequential manner and use each resource for c^{l-1} time slots before proceeding to the next resource in the sequence, where $c > 1$ is a positive integer. The zth resource in sequence $\mathcal{N}_i$ is denoted by $\mathcal{N}_i(z)$. The set of sequences $\mathcal{N}_1, \ldots, \mathcal{N}_M$ is created in a way that all agents will observe all resource-activity pairs at least once, and that the least observed resource-activity pair for each agent is observed only once in a single (parallel) run of the set of sequences by the agents. For example when $M = 2$, $K = 2$, exploration sequences $\mathcal{N}_1 = \{1, 1, 2, 2\}$ and $\mathcal{N}_2 = \{1, 2, 1, 2\}$ are sufficient for each agent to sample all resource-activity pairs once. Note that it is always possible to find such a set of sequences. With M agents and K resources, there are K^M possible assignments of agents to resources. Index each of the K^M possible assignments as $\{\boldsymbol{\alpha}(1), \boldsymbol{\alpha}(2), \ldots, \boldsymbol{\alpha}(K^M)\}$. Then using the set of sequences $\mathcal{N}_i = \{\alpha_i(1), \alpha_i(2), \ldots, \alpha_i(K^M)\}$ for $i \in \mathcal{M}$, all agents sample all resource-activity pairs at least once.

The sequence $\mathcal{N}_i$ is assumed known to agent i before the resource selection process starts. Let $l^i_O(t)$ be the number of completed exploration blocks and $l^i_I(t)$ be the number of completed exploitation blocks of agent i by time t, respectively. For agent i, the length of the lth exploration block is $N'c^{l-1}$ and the length of the lth exploitation

block is ab^{l-1}, where $a, b, c > 1$ are positive integers.

At the beginning of each block, agent i computes $N_O^i(t) := \sum_{l=1}^{l_O^i(t)} c^{l-1}$. If $N_O^i(t) \geq L \log t$, agent i starts an exploitation block at time t. Otherwise, it starts an exploration block. Here L is the exploration constant which controls the number of explorations. Clearly, the number of exploration steps up to t is non-decreasing in L. Since the estimates of mean rewards of resource-activity pairs are based on sample mean estimates of observations during exploration steps, by increasing the value of L, an agent can control the probability of deviation of estimated mean rewards from the true mean. Intuitively, L should be chosen according to $\Delta_{\min}$, since the accuracy of the estimated value of an allocation depends on the accuracy of the estimated mean rewards.

Because of the deterministic nature of the blocks and the property of the sequences $\mathcal{N}_1, \ldots, \mathcal{N}_M$ discussed above, if at time t an agent starts a new exploration (exploitation) block, then all agents start a new exploration (exploitation) block. Therefore $l_O^i(t) = l_O^j(t)$, $N_O^i(t) = N_O^j(t)$, $l_I^i(t) = l_I^j(t)$, for all $i, j \in \mathcal{M}$. Since these quantities are equal for all agents, we drop the superscripts and write them simply as $l_O(t), N_O(t), l_I(t)$. Let t_l be the time at the beginning of the lth exploitation block. At time t_l, $l = 1, 2, \ldots$, agent i computes an estimated optimal allocation $\hat{\boldsymbol{n}}^i(l) = \{\hat{n}_1^i(l), \ldots, \hat{n}_K^i(l)\}$ based on the sample mean estimates of the resource-activity pairs, in the same way it was done under the RLOF algorithm in the previous section, given by

$$\hat{\boldsymbol{n}}^i(l) = \arg\max_{\boldsymbol{n} \in \mathcal{N}} \sum_{k=1}^{K} n_k \hat{\mu}_{k,n_k}(N_{k,n_k}^i(t_l)) \ .$$

Randomization at an exploitation block of DLOE is similar to the randomization at an exploitation step of RLOF. Basically, the agent selects a resource from the set $\mathcal{O}_i(l)$, the set of resources selected by at least one agent under $\hat{\boldsymbol{n}}^i(l)$. If $\hat{\boldsymbol{n}}^i(l-1) = \hat{\boldsymbol{n}}^i(l)$ and

if in the last time step t' of exploitation block $l-1$ the number of agents on the resource selected by agent i is at most $\hat{n}^i_{\alpha_i(t')}(l)$, then agent i selects the same resource at the first time step of the lth exploitation block. Otherwise, agent i randomly chooses a resource k within $\mathcal{O}_i(l)$ with probability $\hat{n}^i_k/M$ at the first time step of the lth exploitation block. During the lth exploitation block, if the number of agents on resource $\alpha_i(t)$ that agent i selects is greater than the estimated $\hat{n}^i_{\alpha_i(t)}(l)$, then in the next time step agent i randomly chooses a resource k within $\mathcal{O}_i(l)$ with probability $\hat{n}^i_k/M$; otherwise the agent stays in the same resource in the next time step. Therefore, it is more likely for an agent to select a resource which it believes should have a large number of agents under the optimal allocation than a resource which it believes should only be used by a smaller number of agents.

7.4.1 Analysis of the regret of DLOE

We next analyze the regret of DLOE. Similar to RLOF, there are three factors contributing to the regret in both IID and the Markovian resource models. The first is the regret due to exploration blocks, the second is the regret due to incorrect computation of the optimal allocation by an agent, and the third is the regret due to the randomization before settling to the optimal allocation given that all agents computed the optimal allocation correctly. In addition to the above, another factor contributing to the regret under the Markovian resource model comes from the transient effect of a resource-activity pair not being in its stationary distribution when chosen by an agent. Finally, in this section we will also consider the impact of computational effort on the regret, by using both regret definitions 7.4 and 7.5.

In the following lemmas, we will bound parts of the regret that are common to both the IID and the Markovian models. Bounds on the other parts which depend

Distributed Learning with Ordered Exploration (DLOE) for agent i

Input: Exploration sequence $\mathcal{N}_i$, $a, b, c \in \{2, 3, \ldots\}$.
Initialize: $t = 1$, $l_O = 0$, $l_I = 0$, $\eta = 1$, $N_O^i = 0$, $F = 2$, $z = 1$, $len = 2$, $\hat{\mu}_{k,n}^i = 0$, $N_{k,n}^i = 0$, $\forall k \in \mathcal{K}, n \in \{1, 2, \ldots, M\}$, $n_{old} = 1$, $\alpha_{old} = 1$, $\hat{\boldsymbol{n}}_{old} = \mathbf{1}$.
while $t \geq 1$ **do**
  if $F = 1$ //Exploitation block **then**
    if $\eta = 1$ //beginning of an exploitation block **then**
      if $\hat{\boldsymbol{n}}^i = \hat{\boldsymbol{n}}_{old}$ and $n_{old} \leq \hat{n}^i_{\alpha_{old}}$ **then**
        $\alpha_i(t) = \alpha_{old}$
      else
        Pick $\alpha_i(t)$ randomly from $\mathcal{O}_i$ with $P(\alpha_i(t) = k) = \hat{n}_k^i / M$.
      end if
    else
      //Not the beginning of an exploitation block
      if $n^{t-1}_{\alpha_i(t-1)} > \hat{n}^i_{\alpha_i(t-1)}$ **then**
        Pick $\alpha_i(t)$ randomly from $\mathcal{O}_i$ with $P(\alpha_i(t) = k) = \hat{n}_k^i / M$.
      else
        $\alpha_i(t) = \alpha_i(t-1)$
      end if
    end if
    Observe $n^t_{\alpha_i(t)}$ and get the reward $r^i_{\alpha_i(t), n^t_{\alpha_i(t)}}(t)$.
    if $\eta = len$ **then**
      $F = 0$
      $\hat{\boldsymbol{n}}_{old} = \hat{\boldsymbol{n}}^i$, $\alpha_{old} = \alpha_i(t)$, $n_{old} = n^t_{\alpha_i(t)}$
    end if
  else if $F = 2$ //Exploration block **then**
    $\alpha_i(t) = \mathcal{N}_i'(z)$
    Observe $n^t_{\alpha_i(t)}$ and get the reward $r^i_{\alpha_i(t), n^t_{\alpha_i(t)}}(t)$.
    $++ N^i_{\alpha_i(t), n^t_{\alpha_i(t)}}$, $\hat{\mu}^i_{\alpha_i(t), n^t_{\alpha_i(t)}} = \frac{(N^i_{\alpha_i(t), n^t_{\alpha_i(t)}} - 1)\hat{\mu}^i_{\alpha_i(t), n^t_{\alpha_i(t)}} + r^i_{\alpha_i(t), n^t_{\alpha_i(t)}}(t)}{N^i_{\alpha_i(t), n^t_{\alpha_i(t)}}}$
    if $\eta = len$ **then**
      $\eta = 0$, $++ z$
    end if
    if $z = |\mathcal{N}_i| + 1$ **then**
      $N_O^i = N_O^i + len$
      $F = 0$
    end if
  end if
  if $F = 0$ **then**
    if $N_O^i \geq L \log t$ **then**
      //Start an exploitation block
      $F = 1$, $++ l_I$, $\eta = 1$, $len = a \times b^{l_I - 1}$
      //Compute the estimated optimal allocation
      $\hat{\boldsymbol{n}}^i = \arg\max_{\boldsymbol{n} \in \mathcal{N}} \sum_{k=1}^K n_k \hat{\mu}^i_{k, n_k}$
      Set $\mathcal{O}_i$ to be the set of resources in $\hat{\boldsymbol{n}}^i$ with $\hat{n}_k^i \geq 1$.
    else if $N_O^i < L \log t$ **then**
      //Start an exploration block
      $F = 2$, $++ l_O$, $\eta = 0$, $len = c^{l_O - 1}$, $z = 1$
    end if
  end if
  $++ \eta$, $++ t$
end while

Figure 7.3: pseudocode of DLOE

on the resource model are given in the next two subsections. Assume that

$$\sum_{l=1}^{l_O(t)-1} c^{l-1} \geq L \log t.$$

If this were true, then an agent would not start the $l_O(t)$th exploration block at or before time t. Thus, we should have

$$\sum_{l=1}^{l_O(t)-1} c^{l-1} < L \log t.$$

This implies that for any time t, we have

$$\begin{aligned} \sum_{l=1}^{l_O(t)-1} c^{l-1} < L \log t &\Rightarrow \frac{c^{l_O(t)-1}-1}{c-1} < L \log t \\ &\Rightarrow c^{l_O(t)-1} < (c-1) L \log t + 1 \\ &\Rightarrow l_O(t) < \log_c((c-1) L \log t + 1) + 1. \end{aligned} \tag{7.22}$$

Let $T_O(t)$ be the time spent in exploration blocks by time t. By (7.22) we have

$$\begin{aligned} T_O(t) &\leq \sum_{l=1}^{l_O(t)+1} N' c^{l-1} = N' \frac{c^{l_O(t)+1}-1}{c-1} \\ &< \frac{N'(c((c-1)L \log t + 1) - 1)}{c-1} = N'(cL \log t + 1). \end{aligned} \tag{7.23}$$

Lemma VII.13. *For any $t > 0$, regret due to explorations by time t is at most*

$$MN'(cL \log t + 1).$$

Proof. Due to the bounded rewards in $[0, 1]$, an upper bound to the worst case is when each agent loses a reward of 1 due to suboptimal decisions at each step in an exploration block. The result follows the bound (7.23) for $T_O(t)$. □

By time t at most $t - N'$ slots have been spent in exploitation blocks (because of the initial exploration the first N' slots are always in an exploration block). Therefore

$$\begin{aligned}\sum_{l=1}^{l_I(t)} ab^{l-1} &= a\frac{b^{l_I(t)}-1}{b-1} \leq t - N' \\ &\Rightarrow b^{l_I(t)} \leq \frac{b-1}{a}(t-N') + 1 \\ &\Rightarrow l_I(t) \leq \log_b\left(\frac{b-1}{a}(t-N') + 1\right). \end{aligned} \tag{7.24}$$

The next lemma bounds the computational cost of solving the NP-hard optimization problem of finding the estimated optimal allocation.

Lemma VII.14. *When agents use DLOE, the regret due to computations by time t is upper bounded by*

$$C_{cmp}M\left(\log_b\left(\frac{b-1}{a}(t-N') + 1\right) + 1\right).$$

Proof. The optimal allocation is computed at the beginning of each exploitation block. The number of completed exploitation blocks is bounded by (7.24). But time t might be in an incomplete exploitation block, hence we add the regret from that block. □

7.4.2 Regret Analysis for IID Resources

In this subsection we analyze the regret of DLOE under the IID resource model. We note that in the IID model the reward of each resource-activity pair is generated by an IID process with support in $[0, 1]$. With part of the regret bounded in Section 7.4.1, our next step is to bound the regret caused by incorrect calculation of the optimal allocation by some agent, using a Chernoff-Heoffding bound. Let $\epsilon := \Delta_{\min}/(2M)$ denote the maximum distance between the estimated resource-activity reward and the true resource-activity reward such that Lemma VII.4 holds.

Lemma VII.15. *Under the IID model, when each agent uses DLOE with constant $L \geq 1/\epsilon^2$, the regret due to incorrect calculations of the optimal allocation by time t is at most*

$$M^3K(b-1)(\log(t)+1) + M^3K(a-(b-1)N')^+\beta,$$

where $(a-(b-1)N')^+ = \max\{0,(a-(b-1)N')\}$ *and* $\beta = \sum_{t=1}^{\infty} 1/t^2$.

Proof. Similar to the proof of Lemma VII.9 for RLOF, let $H(t_l)$ be the event that at the beginning of the lth exploitation block, there exists at least one agent who computed the optimal allocation incorrectly. Let ω be a sample path of the stochastic process generated by the learning algorithm and the stochastic resource rewards. The event that agent i computes the optimal allocation incorrectly is a subset of the event

$$\{|\hat{\mu}^i_{k,n}(N^i_{k,n}(t_l)) - \mu^i_{k,n}| \geq \epsilon \text{ for some } k \in \mathcal{K}, n \in \mathcal{M}\}.$$

Therefore $H(t_l)$ is a subset of the event

$$\{|\hat{\mu}^i_{k,n}(N^i_{k,n}(t_l)) - \mu^i_{k,n}| \geq \epsilon \text{ for some } i \in \mathcal{M}, k \in \mathcal{K}, n \in \mathcal{M}\}.$$

Using a union bound, we have

$$I(\omega \in H(t_l)) \leq \sum_{i=1}^{M}\sum_{k=1}^{K}\sum_{n=1}^{M} I(|\hat{\mu}^i_{k,n}(N^i_{k,n}(t_l)) - \mu^i_{k,n}| \geq \epsilon).$$

Then taking its expected value, we get

$$P(\omega \in H(t_l)) \leq \sum_{i=1}^{M}\sum_{k=1}^{K}\sum_{n=1}^{M} P(|\hat{\mu}^i_{k,n}(N^i_{k,n}(t_l)) - \mu^i_{k,n}| \geq \epsilon). \tag{7.25}$$

Since

$$P(|a-b| \geq \epsilon) = 2P(a-b \geq \epsilon),$$

for $a, b > 0$, we have

$$P(|\hat{\mu}^i_{k,n}(N^i_{k,n}(t_l)) - \mu^i_{k,n}| \geq \epsilon) = 2P(\hat{\mu}^i_{k,n}(N^i_{k,n}(t_l)) - \mu^i_{k,n} \geq \epsilon). \tag{7.26}$$

Since l is an exploitation block we have $N^i_{k,n}(t_l) > L \log t_l$. Let $r^i_{k,n}(t) := r_k(s^t_k, n^t_k)$ and let $\tilde{t}^i_{k,n}(l)$ denote the time when agent i chooses resource k and observes n agents on it for the lth time. We have

$$\hat{\mu}^i_{k,n}(N^i_{k,n}(t_l)) = \frac{\sum_{z=1}^{N^i_{k,n}(t_l)} r^i_{k,n}(\tilde{t}^i_{k,n}(z))}{N^i_{k,n}(t_l)} \ .$$

Using a Chernoff-Hoeffding bound

$$\begin{aligned} P(\hat{\mu}^i_{k,n}(N^i_{k,n}(t_l)) - \mu^i_{k,n} \geq \epsilon) &= P\left(\sum_{z=1}^{N^i_{k,n}(t_l)} r^i_{k,n}(\tilde{t}^i_{k,n}(z)) \geq N^i_{k,n}(t_l)\mu^i_{k,n} + N^i_{k,n}(t_l)\epsilon\right) \\ &\leq e^{-2N^i_{k,n}(t_l)\epsilon^2} \leq e^{-2L\log t_l \epsilon^2} \leq \frac{1}{t_l^2}, \end{aligned} \tag{7.27}$$

where the last inequality follows from the fact that $L \geq 1/\epsilon^2$. Substituting (7.26) and (7.27) into (7.25), we have

$$P(\omega \in H(t_l)) \leq M^2 K \frac{1}{t_l^2}. \tag{7.28}$$

The regret in the lth exploitation block caused by incorrect calculation of the optimal allocation by at least one agent is upper bounded by

$$Mab^{l-1}P(\omega \in H(t_l)),$$

since there are M agents and the resource rewards are in $[0, 1]$. Since

$$\sum_{l=1}^{l_I(t)} ab^{l-1} = a\frac{b^{l_I(t)} - 1}{b-1} \leq t - N',$$

for all t, the length of the lth exploitation block is bounded by

$$ab^{l-1} \leq (b-1)(t_l - N') + a$$

Note that by time t there can be at most $l_I(t) + 1$ exploitation blocks. We do the analysis for $t_{l_I(t)+1} < t$. Otherwise, our results will still hold since we will not include the regret in the $l_I(t) + 1$ exploitation block in calculating the regret by time t. Therefore, the regret caused by incorrect calculation of the optimal allocation by at least one agent by time t is upper bounded by

$$\begin{aligned}
&\sum_{l=1}^{l_I(t)+1} Mab^{l-1}P(\omega \in H(t_l)) \leq M^3K \sum_{l=1}^{l_I(t)+1} ab^{l-1}\frac{1}{t_l^2} \\
&\leq M^3K \sum_{l=1}^{l_I(t)+1} ((b-1)(t_l - N') + a)\frac{1}{t_l^2} \\
&\leq M^3K \sum_{l=1}^{l_I(t)+1} (b-1)\frac{1}{t_l} + M^3K \sum_{l=1}^{l_I(t)+1} (a - (b-1)N')^+\frac{1}{t_l^2} \\
&\leq M^3K(b-1)\sum_{t'=1}^{t}\frac{1}{t} + M^3K(a-(b-1)N')^+\sum_{t'=1}^{\infty}\frac{1}{t'^2} \\
&\leq M^3K(b-1)(\log(t)+1) + M^3K(a-(b-1)N')^+\beta.
\end{aligned}$$

□

The following lemma bounds the expected number of exploitation blocks where some agent computes the optimal allocation incorrectly.

Lemma VII.16. *When agents use DLOE with $L \geq 1/\epsilon^2$, the expected number of exploitation blocks up to any t in which there exists at least one agent who computes*

the optimal allocation wrong is bounded by

$$E\left[\sum_{l=1}^{\infty} I(\omega \in H(t_l))\right] \leq \sum_{l=1}^{\infty} \frac{M^2 K}{t_l^2} \leq M^2 K \beta,$$

where $\beta = \sum_{t=1}^{\infty} 1/t^2$.

Proof. Proof is similar to the proof of Lemma VII.15, using the bound (7.28) for $P(\omega \in H(t_l))$. □

Finally, we bound the regret due to the randomization before settling to the optimal allocation in exploitation slots in which all agents have computed the optimal allocation correctly.

Lemma VII.17. *Denote the number of resources which are selected by at least one agent in the optimal allocation by* z^*. *Reindex the resources in* $\mathcal{O}^*$ *by* $\{1, 2, \ldots, z^*\}$. *Let* $\mathcal{M} = \{\boldsymbol{m} : m_1 + m_2 + \ldots + m_{z^*} \leq M, m_i \geq 0, \forall i \in \{1, 2, \ldots, z^*\}\}$. *In an exploitation block where each agent computed the optimal allocation correctly, the expected number of time steps in that block before settling to the optimal allocation in this block is upper bounded by*

$$O_B := \frac{1}{\min_{\boldsymbol{m} \in \mathcal{M}} P_{DLOE}(\boldsymbol{m})},$$

where

$$P_{DLOE}(\boldsymbol{m}) = \frac{(M-m)!}{(n_1 - m_1)! \ldots (n_{z^*} - m_{z^*})!} \left(\frac{n_1}{M}\right)^{n_1 - m_1} \ldots \left(\frac{n_{z^*}}{M}\right)^{n_{z^*} - m_{z^*}}.$$

Proof. The proof is similar to the proof of Lemma VII.11, thus omitted. □

Lemma VII.18. *The regret due to randomization before settling to the optimal allo-*

cation is bounded by

$$O_B M^3 K \beta,$$

where

$$\beta = \sum_{t=1}^{\infty} \frac{1}{t^2}.$$

Proof. Using similar terms as before, a *good* exploitation block is an exploitation block in which all the agents computed the optimal allocation correctly, while a *bad* exploitation block is a block in which there exists at least one agent who computed the optimal allocation incorrectly. The worst case is when each bad block is followed by a good block. The number of bad blocks is bounded by Lemma VII.16. After each such transition from a bad block to a good block, the expected loss is at most O_B, which is given in Lemma VII.17. □

Combining all the results above we have the following theorem.

Theorem VII.19. *If all agents use DLOE with $L \geq 1/\epsilon^2$, at any $t > 0$, the regret defined in (7.4) is upper bounded by,*

$$(M^3 K(b-1) + MN'cL)\log(t) + M^3 K(1 + \beta((a-(b-1)N')^+ O_B)) + MN',$$

and the regret defined in (7.5) is upper bounded by

$$\begin{aligned}&(M^3 K(b-1) + MN'cL)\log(t) + M^3 K(1 + \beta((a-(b-1)N')^+ O_B)) + MN' \\ &+ C_{cmp} M \left(\log_b \left(\frac{b-1}{a}(t - N') + 1 \right) + 1 \right),\end{aligned}$$

where O_B, given in Lemma VII.17, is the worst case expected hitting time of the optimal allocation given all agents know the optimal allocation, $(a-(b-1)N')^+ =$

$\max\{0, (a-(b-1)N')\}$ *and* $\beta = \sum_{t=1}^{\infty} 1/t^2$.

Proof. The result follows from summing the regret terms from Lemmas VII.13, VII.15, VII.18 and VII.14. □

7.4.3 Regret Analysis for Markovian Resources

In this subsection we analyze the regret of DLOE in the case of Markovian resources. The analysis in this section is quite different from that in Section 7.4.2 due to the Markovian state transition rule. Similar as before, our next step is to bound the regret caused by incorrect calculation of the optimal allocation by some agent. Although the proof of the following lemma is very similar to the proof of Lemma VII.15, due to the Markovian nature of the rewards, we need to bound the deviation probability between the estimated mean resource-activity rewards and the true mean resource-activity rewards in a different way. For simplicity of analysis, we assume that DLOE is run with parameters $a = 2$, $b = 4$, $c = 4$; similar analysis can be done for other, arbitrary parameter values. Again let $\epsilon := \Delta_{\min}/(2M)$.

The following technical assumption, which is also given in Chapter III for the analysis of the single-agent restless bandit problem, ensures sufficiently fast convergence of states to their stationary distribution.

Assumption VII.20. *Let* $(P^k)'$ *denote the* adjoint *of* P^k *on* $l_2(\pi)$ *where*

$$(p^k)'_{xy} = (\pi^k_y p^k_{yx})/\pi^k_x, \ \forall x, y \in S^k \ .$$

Let $\dot{P}^k = (P^k)'P$ *denote the* multiplicative symmetrization *of* P^k. *We assume that the* P^k*'s are such that* $\dot{P}^k$*'s are irreducible.*

To give a sense of the strength of this assumption, we note that this is a weaker condition than assuming the Markov chains to be reversible. With this bound, if an agent can estimate the mean rewards of resource-activity pairs accurately, then

the probability that the agent chooses a suboptimal resource can be made arbitrarily small. Let ξ^k be the eigenvalue gap, i.e., 1 minus the second largest eigenvalue of $\dot{P}^k$, and $\xi_{\min} = \min_{k\in\mathcal{K}} \xi^k$. Let $r_{\Sigma,\max} = \max_{k\in\mathcal{K}} \sum_{x\in S^k} r_x^k$, $r_{\Sigma,\min} = \min_{k\in\mathcal{K}} \sum_{x\in S^k} r_x^k$.

Lemma VII.21. *Under the Markovian model, when each agent uses DLOE with constant*

$$L \geq \max\{1/\epsilon^2, 50S_{\max}^2 r_{\Sigma,\max}^2/((3-2\sqrt{2})\xi_{\min})\},$$

the regret due to incorrect calculations of the optimal allocation by time t is at most

$$3M^3K\left(\frac{1}{\log 2} + \frac{\sqrt{2L}}{10r_{\Sigma,\min}}\right)\frac{S_{\max}}{\pi_{\min}}(log(t)+1).$$

Proof. Similar to the analysis for the IID rewards, let $H(t_l)$ be the event that at the beginning of the lth exploitation block, there exists at least one agent who computes the optimal allocation incorrectly, and let ω be a sample path of the stochastic process generated by the learning algorithm and the stochastic rewards. Proceeding the same way as in the proof of Lemma VII.15 by (7.25) and (7.26) we have,

$$P(\omega \in H(t_l)) \leq \sum_{i=1}^{M}\sum_{k=1}^{K}\sum_{n=1}^{M} 2P(\hat{\mu}_{k,n}^i(N_{k,n}^i(t_l)) - \mu_{k,n}^i \geq \epsilon). \tag{7.29}$$

Since t_l is the beginning of an exploitation block we have $N_{k,n}^i(t_l) \geq L\log t_l$, $\forall i \in \mathcal{M}, k \in \mathcal{K}, n \in \mathcal{M}$. This implies that $N_{k,n}^i(t_l) \geq \sqrt{N_{k,n}^i(t_l)L\log t_l}$. Hence

$$\begin{aligned}
&P(\hat{\mu}_{k,n}^i(N_{k,n}^i(t_l)) - \mu_{k,n}^i \geq \epsilon) \\
&= P(N_{k,n}^i(t_l)\hat{\mu}_{k,n}^i(N_{k,n}^i(t_l)) - N_{k,n}^i(t_l)\mu_{k,n}^i \geq \epsilon N_{k,n}^i(t_l)) \\
&\leq P\left(N_{k,n}^i(t_l)\hat{\mu}_{k,n}^i(N_{k,n}^i(t_l)) - N_{k,n}^i(t_l)\mu_{k,n}^i \geq \epsilon\sqrt{N_{k,n}^i(t_l)L\log t_l}\right) .
\end{aligned} \tag{7.30}$$

To bound (7.30), we proceed in the same way as in the proof of Theorem 1 in *Liu*

et al. (2010). The idea is to separate the total number of observations of the resource-activity pair (k, n) by agent i into multiple contiguous segments. Then, using a union bound, (7.30) is upper bounded by the sum of the deviation probabilities for each segment. By Assumption VII.20 we can use the large deviation bound given in Lemma A.1 to bound the deviation probability in each segment. Thus, for a suitable choice of the exploration constant L, the deviation probability in each segment is bounded by a negative power of t_l. Combining this with the fact that the number of such segments is logarithmic in time (due to the geometrically increasing block lengths), for block length parameters $a = 2$, $b = 4$, $c = 4$ in DLOE, and for

$$L \geq \max\{1/\epsilon^2, 50S_{\max}^2 r_{\Sigma,\max}^2/((3-2\sqrt{2})\xi_{\min})\},$$

we have,

$$\begin{aligned} &P\left(N_{k,n}^i(t_l)\hat{\mu}_{k,n}^i(N_{k,n}^i(t_l)) - N_{k,n}^i(t_l)\mu_{k,n}^i \geq \epsilon\sqrt{N_{k,n}^i(t_l)L\log t_l}\right) \\ &\leq \left(\frac{1}{\log 2} + \frac{\sqrt{2L}}{10r_{\Sigma,\min}}\right)\frac{S_{\max}}{\pi_{\min}}t_l^{-2}. \end{aligned}$$

Continuing from (7.29), we get

$$P(\omega \in H(t_l)) \leq M^2K\left(\frac{1}{\log 2} + \frac{\sqrt{2L}}{10r_{\Sigma,\min}}\right)\frac{S_{\max}}{\pi_{\min}}t_l^{-2}. \tag{7.31}$$

The result is obtained by continuing the same way as in the proof of Lemma VII.15. □

The following lemma bounds the expected number of exploitation blocks where some agent computes the optimal allocation incorrectly.

Lemma VII.22. *Under the Markovian model, when each agent uses DLOE with*

constant

$$L \geq \max\{1/\epsilon^2, 50S_{\max}^2 r_{\Sigma,\max}^2/((3-2\sqrt{2})\xi_{\min})\},$$

the expected number exploitation blocks up to any t in which there exists at least one agent who computes the optimal allocation wrong is bounded by

$$E\left[\sum_{l=1}^{\infty} I(\omega \in H(t_l))\right] \leq M^2 K \left(\frac{1}{\log 2} + \frac{\sqrt{2L}}{10 r_{\Sigma,\min}}\right) \frac{S_{\max}}{\pi_{\min}} \beta,$$

where $\beta = \sum_{t=1}^{\infty} 1/t^2$.

Proof. The proof is similar to that of Lemma VII.21, using the bound (7.31) for $P(\omega \in H(t_l))$. □

Next, we bound the regret due to the randomization before settling to the optimal allocation in exploitation blocks in which all agents have computed the optimal allocation correctly.

Lemma VII.23. *The regret due to randomization before settling to the optimal allocation is bounded by*

$$(O_B + C_{\boldsymbol{P}}) M^3 K \left(\frac{1}{\log 2} + \frac{\sqrt{2L}}{10 r_{\Sigma,\min}}\right) \frac{S_{\max}}{\pi_{\min}} \beta,$$

where O_B as given in Lemma VII.17 is the worst case expected hitting time of the optimal allocation given all agents know the optimal allocation, $\beta = \sum_{t=1}^{\infty} 1/t^2$, and $C_{\boldsymbol{P}} = \max_{k \in \mathcal{K}} C_{P^k}$ where C_P is a constant that depends on the transition probability matrix P.

Proof. Again, a *good* exploitation block refers to an exploitation block in which all agents compute the optimal allocation correctly, whereas a *bad* exploitation block is one in which there exists at least one agent who computes the optimal allocation

incorrectly. By converting the problem into a simple balls in bins problem where the balls are agents and the bins are resources, the expected number of time slots spent before settling to the optimal allocation in a good exploitation block is bounded above by O_B. The worst case is when each bad block is followed by a good block, and the number of bad blocks is bounded by Lemma VII.22. Moreover, due to the transient effect that a resource may not be at its stationary distribution when it is selected, even after settling to the optimal allocation in an exploitation block, the regret of at most $C_{\boldsymbol{P}}$ can be accrued by an agent. This is because the difference between the t-horizon expected reward of an irreducible, aperiodic Markov chain with an arbitrary initial distribution and t times the expected reward at the stationary distribution is bounded by $C_{\boldsymbol{P}}$ independent of t. Since there are M agents and resource rewards are in $[0, 1]$, the result follows. □

Combining all the results above we have the following theorem.

Theorem VII.24. *Under the Markovian model, when each agent uses DLOE with constant*

$$L \geq \max\{1/\epsilon^2, 50S_{\max}^2 r_{\Sigma,\max}^2/((3-2\sqrt{2})\xi_{\min})\},$$

then at any time $t > 0$, the regret defined in (7.4) is upper bounded by

$$\left(MN'cL + 3M^3K\left(\frac{1}{\log 2} + \frac{\sqrt{2L}}{10r_{\Sigma,\min}}\right)\frac{S_{\max}}{\pi_{\min}}\right)\log(t)$$
$$+ M^3K\left(\frac{1}{\log 2} + \frac{\sqrt{2L}}{10r_{\Sigma,\min}}\right)\frac{S_{\max}}{\pi_{\min}}(\beta(O_B + C_{\boldsymbol{P}}) + 1) + MN',$$

and the regret defined in (7.5) is upper bounded by

$$\left(MN'cL + 3M^3K\left(\frac{1}{\log 2} + \frac{\sqrt{2L}}{10r_{\Sigma,\min}}\right)\frac{S_{\max}}{\pi_{\min}}\right)\log(t_l)$$

$$+ M^3 K \left(\frac{1}{\log 2} + \frac{\sqrt{2L}}{10 r_{\Sigma,\min}} \right) \frac{S_{\max}}{\pi_{\min}} (\beta(O_B + C_{\boldsymbol{P}}) + 1) + MN'$$
$$+ C_{cmp} M \left(\log_b \left(\frac{b-1}{a}(t - N') + 1 \right) + 1 \right),$$

where O_B as given in Lemma VII.17 is the worst case expected hitting time of the optimal allocation given all agents know the optimal allocation, $\beta = \sum_{t=1}^{\infty} 1/t^2$, and $C_{\boldsymbol{P}} = \max_{k \in \mathcal{K}} C_{P^k}$ where C_P is a constant that depends on the transition probability matrix P.

Proof. The result follows from summing the regret terms from Lemmas VII.13, VII.21, VII.23 and VII.14, and the fact that $a = 2$, $b = 4$. □

To summarize this section, our results show that when initial synchronization between agents is possible, logarithmic regret, which is the optimal order of regret even in the centralized case for the IID reward model (see, e.g., *Anantharam et al.* (1987a)) can be achieved in a decentralized setting. Moreover, the proposed algorithm does not need to know whether the rewards are IID or Markovian; the logarithmic regret holds in both cases.

7.5 Achievable Performance with Costly Communication

In this section we consider the model where the resource rewards are agent-specific. Different from the previous section, where an agent can compute the optimal allocation based only on its own estimates, with agent-specific resource rewards each agent needs to know the estimated rewards of other agents in order to compute the optimal allocation. We assume that agents can communicate with each other, but this communication incurs a cost C_{com}. For example, in an opportunistic spectrum access model, agents are transmitter-receiver pairs that can communicate with each other on one of the available channels, even when no common control channel exists. In order

for communication to take place, each agent can broadcast a request for communication over all available channels. For instance, if agents are using an algorithm based on deterministic sequencing of exploration and exploitation, then at the beginning, an agent can announce the parameters that are used to determine the block lengths. This way, the agents can decide on which exploration and exploitation sequences to use, so that all of them can start an exploration block or an exploitation block at the same time. After this initial communication, just before an exploitation block, agents share their perceived resource qualities with each other, and one of the agents, which can be chosen in a round robin fashion, computes the optimal allocation and announces to each agent the resource it should select in the optimal allocation.

7.5.1 Distributed Learning with Communication

In this subsection, we propose the algorithm distributed learning with communication (DLC) for this model. Similar to DLOE, DLC (see Figure 7.4) consists of exploration and exploitation blocks with geometrically increasing lengths. The predetermined exploration order allows each agent to observe the reward from each resource-activity pair, and update its sample mean estimate. Note that in this case, since feedback about n_k^t is not needed, each agent should be given the number of agents using the same resource with it for each resource in the predetermined exploration order. Similar to the previous section, this predetermined exploration order can be seen as an input to the algorithm from the algorithm designer. On the other hand, since communication between the agents is possible, the predetermined exploration order can be determined by an agent and then communicated to the other agents, or agents may collectively reach to an agreement over a predetermined exploration order by initial communication. In both cases, the initial communication will incur a constant cost $C > 0$, which we neglect in our analysis.

Let $\mathcal{N}_i$ be the exploration sequence of agent i, which is defined the same way as in

Section 7.4, and $\mathcal{L}_i(z)$ be the number of agents using the same resource with agent i in the zth slot of an exploration block. Based on the initialization methods discussed above both $\mathcal{N}_i$ and $\mathcal{L}_i$ are known by agent i at the beginning. At the beginning of the lth exploitation block, all agents send their updated sample mean estimates of resource rewards to agent $i = (l \mod M) + 1$. Then agent i computes an optimal allocation based on the estimated rewards, and announces to each agent the resource it is going to use during the exploitation block. This round robin fashion fairly distributes the computational burden over the agents. Note that, if the agent who computed the optimal allocation announces the resources assigned to other agents as well, then rewards from the exploitation blocks can also be used to update the resource-activity pair reward estimates.

7.5.2 Analysis of the regret of DLC

In this section we bound the regret terms which are same for both IID and Markovian resource rewards.

Lemma VII.25. *For any $t > 0$, regret of DLC due to explorations by time t is at most*

$$MN'(cL \log t + 1).$$

Proof. Since DLC uses deterministic sequencing of exploration and exploitation the same way as DLOE, the proof is same as the proof of Lemma VII.13 by using the bound (7.23) for $T_O(t)$. □

Since communication takes place at the beginning of each exploitation block, it can be computed the same way as computation cost is computed for DLOE. Moreover, since resource switching is only done during exploration blocks or at the beginning of a new exploitation block, switching costs can also be computed the same way. The

Distributed Learning with Communication (DLC) for agent i

Input: Exploration sequence $\mathcal{N}_i$ and $\mathcal{L}_i$, $a, b, c \in \{2, 3, \ldots\}$.
Initialize: $t = 1$, $l_O = 0$, $l_I = 0$, $\eta = 1$, $N_O^i = 0$, $F = 2$, $z = 1$, $len = 2$, $\hat{\mu}_{k,n}^i = 0$, $N_{k,n}^i = 0$, $\forall k \in \mathcal{K}, n \in \{1, 2, \ldots, M\}$.
while $t \geq 1$ **do**
  if $F = 1$ //Exploitation block **then**
    Select resource $\alpha_i(t) = \alpha_i^*$.
    Receive reward $r_{\alpha_i(t)}^i(t)$
    if $\eta = len$ **then**
      $F = 0$
    end if
  else if $F = 2$ //Exploration block **then**
    Select resource $\alpha_i(t) = \mathcal{N}_i'(z)$
    Receive reward $r_{\alpha_i(t)}^i(t)$
    $++ N_{\alpha_i(t),\mathcal{L}_i(z)}^i$, $\hat{\mu}_{\alpha_i(t),\mathcal{L}_i(z)}^i = \frac{(N_{\alpha_i(t),\mathcal{L}_i(z)}^i - 1)\hat{\mu}_{\alpha_i(t),\mathcal{L}_i(z))}^i + r_{\alpha_i(t)}^i(t)}{N_{\alpha_i(t),\mathcal{L}_i(z)}^i}$.
    if $\eta = len$ **then**
      $\eta = 0$, $++ z$
    end if
    if $z = |\mathcal{N}_i| + 1$ **then**
      $N_O^i = N_O^i + len$
      $F = 0$
    end if
  end if
  if $F = 0$ **then**
    if $N_O^i \geq L \log t$ **then**
      //Start an exploitation epoch
      $F = 1$, $++ l_I$, $\eta = 0$, $len = a \times b^{l_I - 1}$
      //Communicate estimated resource qualities $\hat{\mu}_{k,n}^i$, $k \in \mathcal{K}$, $n \in \mathcal{M}$ with other agents.
      if $(l_I \mod M) + 1 = i$ **then**
        //Compute the estimated optimal allocation, and sent each other agent the resource it should use in the exploitation block.
        $\boldsymbol{\alpha}^* = \arg\max_{\boldsymbol{\alpha} \in \mathcal{K}^M} \sum_{i=1}^M \hat{\mu}_{\alpha_i, n_{\alpha_i}(\boldsymbol{\alpha})}^i$
      else
        //Receive the resource that will be selected, i.e., α_i^*, in the exploitation block from agent $(l_I \mod M) + 1$.
      end if
    else if $N_O^i < L \log t$ **then**
      //Start an exploration epoch
      $F = 2$, $++ l_O$, $\eta = 0$, $len = c^{l_O - 1}$, $z = 1$
    end if
  end if
  $++ \eta$, $++ t$
end while

Figure 7.4: pseudocode of DLC

following lemma bounds the communication, computation and switching cost of DLC.

Lemma VII.26. *When agents use DLC, at any time $t > 0$, the regret terms due to*

communication, computation and switching are upper bounded by

$$C_{com}M\left(\log_b\left(\frac{b-1}{a}(t-N')+1\right)+1\right)+C_i,$$
$$C_{cmp}M\left(\log_b\left(\frac{b-1}{a}(t-N')+1\right)+1\right),$$
$$C_{swc}M\left(\left(\log_b\left(\frac{b-1}{a}(t-N')+1\right)+1\right)+N'(cL\log t+1)\right),$$

respectively, where C is the cost of initial communication.

Proof. Communication is done initially and at the beginning of exploitation blocks. Computation is only performed at the beginning of exploitation blocks. Switching is only done at exploration blocks or at the beginning of exploitation blocks. Number of exploitation blocks is bounded by (7.24), and time slots in exploration blocks is bounded by (7.23). □

In the next subsections we analyze the parts of regret that are different for i.i.d. and Markovian rewards.

7.5.3 Regret Analysis for IID Resources

In this subsection we analyze the regret of DLC in the IID resource model. The analysis is similar with the agent independent, general symmetric interaction reward model given in Section 7.4.

Lemma VII.27. *For the IID resource model, when each agent uses DLC with constant $L \geq 1/\epsilon^2$, regret due to incorrect calculations of the optimal allocation by time t is at most*

$$M^3K(b-1)(\log(t)+1)+M^3K(a-(b-1)N')^+\beta,$$

where $(a-(b-1)N')^+ = \max\{0,(a-(b-1)N')\}$ and $\beta = \sum_{t=1}^{\infty} 1/t^2$.

Proof. Let $H(t_l)$ be the event that at the beginning of the lth exploitation block, the estimated optimal allocation calculated by agent $(l \mod M)+1$ is different from the true optimal allocation. Let ω be a sample path of the stochastic process generated by the learning algorithm and the stochastic arm rewards. The event that agent $(l \mod M)+1$ computes the optimal allocation incorrectly is a subset of the event

$$\{|\hat{\mu}^i_{k,n}(N^i_{k,n}(t_l)) - \mu^i_{k,n}| \geq \epsilon \text{ for some } i, n \in \mathcal{M}, k \in \mathcal{K}\}.$$

Analysis follows from using a union bound, taking the expectation, and then using a Chernoff-Hoeffding bound. Basically, it follows from (7.25) in Lemma VII.15. □

Combining all the results above we have the following theorem.

Theorem VII.28. *If all agents use DLC with $L \geq 1/\epsilon^2$, the regret by time $t > 0$ is upper bounded by*

$$\begin{aligned}
&MN'(cL\log t + 1)(1 + C_{swc}) \\
&+ (C_{com} + C_{cmp} + C_{swc})M\left(\log_b\left(\frac{b-1}{a}(t - N') + 1\right) + 1\right) + C_i \\
&+ M^3K(b-1)(\log(t) + 1) + M^3K(a - (b-1)N')^+\beta,
\end{aligned}$$

where $(a-(b-1)N')^+ = \max\{0, (a-(b-1)N')\}$ and $\beta = \sum_{t=1}^{\infty} 1/t^2$.

Proof. The result follows from combining the results of Lemmas VII.25, VII.26 and VII.27. □

7.5.4 Regret Analysis for Markovian Resources

We next analyze the regret of DLC for Markovian resources. The analysis in this section is similar to the ones in Section 7.5.4. We assume that DLC is run with parameters $a = 2$, $b = 4$, $c = 4$.

Lemma VII.29. *Under the Markovian model, when each agent uses DLC with constant*

$$L \geq \max\{1/\epsilon^2, 50S_{\max}^2 r_{\Sigma,\max}^2/((3-2\sqrt{2})\upsilon_{\min})\},$$

the regret due to incorrect calculations of the optimal allocation by time t is at most

$$3M^3K\left(\frac{1}{\log 2}+\frac{\sqrt{2L}}{10r_{\Sigma,\min}}\right)\frac{S_{\max}}{\pi_{\min}}(\log t+1).$$

Proof. The proof follows the proof of Lemma VII.21. □

Combining all the results above we have the following theorem.

Theorem VII.30. *Under the Markovian model, when each agent uses DLC with constant*

$$L \geq \max\{1/\epsilon^2, 50S_{\max}^2 r_{\Sigma,\max}^2/((3-2\sqrt{2})\upsilon_{\min})\},$$

the regret by time t is upper bounded by

$$\begin{aligned}
&MN'(4L\log t+1)(1+C_{swc})\\
&+(C_{com}+C_{cmp}+C_{swc})M\left(\log_4\left(\frac{3}{2}(t-N')+1\right)+1\right)+C_i\\
&+3M^3K\left(\frac{1}{\log 2}+\frac{\sqrt{2L}}{10r_{\Sigma,\min}}\right)\frac{S_{\max}}{\pi_{\min}}(\log t+1),
\end{aligned}$$

where $C_{\boldsymbol{P}} = \max_{k\in\mathcal{K}} C_{P^k}$ where C_P is the constant that depends on the transition probability matrix P.

Proof. The result follows from combining results of Lemmas VII.25, VII.26, and VII.29, and the fact that $a = 2$, $b = 4$, $c = 4$. Note that due to the transient effect that a resource may not be at its stationary distribution when it is selected,

even when all agents select resources according to the optimal allocation, a deviation of at most $C_{\boldsymbol{P}}$ from the expected total reward of the optimal allocation is possible. Therefore, at most $C_{\boldsymbol{P}}$ regret results from the transient effects in exploitation blocks where the optimal allocation is calculated correctly. The last term in the regret is a result of this. □

7.6 Discussion

In this section we discuss several aspects of the problems studied in this chapter, and discuss extensions and relaxations of assumptions we made.

7.6.1 Strategic Considerations

We have showed that in the case of Exp3 and a completely decentralized system, this natural learning process converges to a PNE of a congestion game. This result depicts the similarities between natural learning and better response updates in a congestion game. While both converge to a PNE, the updates of an agent under Exp3 does not explicitly depend on the actions of the other agents. In *Auer et al.* (2003) it was shown that Exp3 has regret $O(\sqrt{T})$ with respect to the best single-action strategy, under the worst-case distribution of the rewards. It is reasonable for an agent to optimize over the worst-case distribution of the rewards when it does not know the number or behavior of other agents in the system, and the distribution of the rewards of the resources. Therefore, even when the agents are strategic, if their goal is to have a high expected reward in the worst-case, they may wish to follow an algorithm with guaranteed worst-case performance (Exp3 in our model) rather than behaving strategically in the classical game-theoretic sense. This argument justifies modeling the agents as non-strategic, even though their goal is self-interest, when faced with uncertainty about the system dynamics.

When partial feedback exists, it is possible for an agent to manipulate the actions of the other agents for its own gain. As an example of strategic behavior, an agent may always choose a resource that it has learned to yield a high single-occupancy reward, to prevent other agents from learning the single-occupancy quality of that resource. This may help the agent avoid competing with the others for that resource. Such strategic interactions can yield complex behavior, and the unknown dynamics of the resources make it even harder to analyze. Therefore, for the partial feedback model we studied, we considered cooperative agents whose joint goal is to maximize the sum of the total rewards of all agents in a distributed way. Our future work involves considering the strategic version of this problem. Designing a distributed learning algorithm for strategic agents with provable performance guarantee with respect to the optimal allocation in the cooperative setting remains an open problem.

7.6.2 Multiple Optimal Allocations

Under both the partial feedback and partial feedback with synchronization models, if there are multiple optimal allocations, even if all agents correctly find an optimal allocation, they may not choose the same optimal allocation since they cannot communicate with each other. To avoid this problem, we adopted Assumption VII.3, which guarantees the uniqueness of the optimal allocation (in terms of the number of agents using each resource). We now describe a modification on DLOE so this assumption is no longer required. A similar modification will also work for RLOF.

We introduce the following subsidy scheme for DLOE. Agent i keeps a subsidy number for each allocation in $\mathcal{N}$. Let $\delta > 0$ be the amount of subsidy. An allocation $\boldsymbol{n} \in \mathcal{N}$ is subsidized by an amount δ by agent i, if agent i adds δ to the estimated value of $\boldsymbol{n}$. Similarly, $\boldsymbol{n}$ is penalized by an amount δ by agent i, if agent i subtracts δ from the estimated value of $\boldsymbol{n}$. Let $\boldsymbol{d}^i = (d^i_{\boldsymbol{n}})_{\boldsymbol{n}\in\mathcal{N}}$ be the subsidy vector of agent i, where $d^i_{\boldsymbol{n}} = 1$ means that i subsidizes $\boldsymbol{n}$, and $d^i_{\boldsymbol{n}} = -1$ means i penalizes $\boldsymbol{n}$. DLOE

is initialized such that $d^i_{\boldsymbol{n}} = 1$ for all $\boldsymbol{n} \in \mathcal{N}$. Let

$$\hat{v}^i_s(\boldsymbol{n}) = \sum_{k=1}^{K} n_k \hat{\mu}^i_{k,n_k} + \delta d^i_{\boldsymbol{n}},$$

be the estimated subsidized reward of $\boldsymbol{n}$ for agent i. At the beginning of each exploitation block, agent i computes $\hat{v}^i_s(\boldsymbol{n})$ based on its sample mean estimates of resource-activity pair rewards and the subsidy vector. Agent i keeps an ordered list $\sigma^i = (\sigma^i_1, \ldots, \sigma^i_{|\mathcal{N}|})$ of allocations in $\mathcal{N}$ such that $\hat{v}^i_s(\boldsymbol{n}_{\sigma^i_j}) \geq \hat{v}^i_s(\boldsymbol{n}_{\sigma^i_l})$ for $j < l$. The subsidized estimated optimal allocation is given by

$$\hat{v}^i_s(\boldsymbol{n}_{\sigma^i_1}) \in \arg\max_{\boldsymbol{n} \in \mathcal{N}} \hat{v}^i_s(\boldsymbol{n}). \tag{7.32}$$

If there is more than one maximizer of (7.32), then agent i randomly places one of them to the first place in its list. At each time step t in an exploitation block, agent i selects a resource which is used by at least one agent in the first allocation in its list. The subsidy vector and list of agent i dynamically changes during an exploitation block in the following way. If agent i observed that the number of agents is less than or equal to the number of agents under the subsidized estimated optimal allocation for that resource at time t, then it sets $d^i_{\boldsymbol{n}_{\sigma^i_1}} = 1$ so the list does not change (the estimated value of the first allocation in the list does not decrease). Otherwise, agent i sets $d^i_{\boldsymbol{n}_{\sigma^i_1}} = -1$ and recomputes $\hat{v}^i_s(\boldsymbol{n}_{\sigma^i_1})$. Based on the new value, it reorders the allocations in the list. Note that it keeps selecting the same resource unless the subsidy number for the allocation which is at the first place in its list changes from 1 to -1.

Let $\epsilon := \Delta_{\min}/(2M)$, where $\Delta_{\min}$ is the minimum suboptimality gap given in (7.1). Assume that each agent runs modified DLOE with subsidy value $\delta = \epsilon/3$, and exploration constant $L \geq 36/\epsilon^2$. Let $H_\epsilon(l)$ be the event that at the beginning of the lth exploitation block there exists some agent i for which $|\hat{v}^i(\boldsymbol{n}) - v(\boldsymbol{n})| \geq \epsilon/6$ for

some $\boldsymbol{n} \in \mathcal{N}$. By Lemma VII.16, the expected number of exploitation blocks in which event $H_\epsilon(l)$ happens is at most $M^2K\beta$.

We consider exploitation blocks l in which $H_\epsilon(l)$ does not happen. We call such an exploitation block a *good* exploitation block. This means that $|\hat{v}^i(\boldsymbol{n}) - v(\boldsymbol{n})| < \epsilon/6$ for all $\boldsymbol{n} \in \mathcal{N}$. In these blocks, for any agent i even if all optimal allocations in $\mathcal{N}^*$ are penalized by $\epsilon/3$, a suboptimal allocation which is subsidized by $\epsilon/3$ cannot have a larger estimated subsidized value than an optimal allocation. Therefore, in all time steps in these blocks, each agent selects a resource according to one of the optimal allocations. Next, we will show that the agents will settle to an optimal allocation in finite expected time in these blocks. Since when $H_\epsilon(l)$ does not happen, the estimated optimal allocation of an agent is always one of the allocations in $\mathcal{N}^*$, we are only interested in the components of the subsidy vector of agent i corresponding to one of these allocations. Let $\mathcal{D} = \{\boldsymbol{d} = (d^1_{\boldsymbol{n}})_{\boldsymbol{n}\in\mathcal{N}}, \ldots, (d^M_{\boldsymbol{n}})_{\boldsymbol{n}\in\mathcal{N}} : d^i_{\boldsymbol{n}} \in \{-1, 1\}\}$. We call $\boldsymbol{d}$ the *reduced subsidy vector*. Let $\mathcal{A}_s(\boldsymbol{d}) \subset \mathcal{A}$ be the set of allocations that can occur when the reduced subsidy vector is $\boldsymbol{d}$.

Consider the Markov chain whose states are $\boldsymbol{d} \times \boldsymbol{\alpha}$, where $\boldsymbol{d} \in \mathcal{D}$ and $\boldsymbol{\alpha} \in \mathcal{A}_s(\boldsymbol{d})$. Define the state transition probabilities of this Markov chain according to the randomization rule in the modified DLOE. It can be seen that this Markov chain is finite state and irreducible. Consider the set of states for which $\boldsymbol{d}$ is such that $d^i_{\boldsymbol{n}'} = 1$ for some $\boldsymbol{n}' \in \mathcal{N}^*$, and $d^i_{\boldsymbol{n}} = -1$ for $\boldsymbol{n} \in \mathcal{N}^* - \{\boldsymbol{n}'\}$, for all $i \in \mathcal{M}$. These are the set of states in which all agents subsidize the same optimal allocation $\boldsymbol{n}'$, while any other optimal allocation is penalized by all agents. Since $|\hat{v}^i(\boldsymbol{n}) - v(\boldsymbol{n})| < \epsilon/6$ for all $\boldsymbol{n} \in \mathcal{N}$, the estimated optimal subsidized allocation is $\boldsymbol{n}'$ for all agents. Therefore, there is a positive probability of settling to the optimal allocation for such a $\boldsymbol{d}$. The state $\boldsymbol{d} \times \boldsymbol{\alpha}$ in which $\boldsymbol{\alpha}$ induces the allocation $\boldsymbol{n}'$ in terms of number of agents on each resource is an absorbing state. Note that there are at least $|\mathcal{N}^*|$ absorbing states, in which the same optimal allocation is the only subsidized optimal allocation by all

agents.

Therefore, the expected time to settle to an optimal allocation in a good exploitation block is bounded above by the expected hitting time of any absorbing state of this Markov chain, which is finite. Doing a worst-case analysis similar to Lemma VII.18, we can show that the regret due to randomization before settling to an optimal allocation is bounded by a finite term which is independent of the time horizon T.

7.6.3 Unknown Suboptimality Gap

Algorithms DLOE and DLC requires that the agents know a lower bound ϵ on the difference between the estimated and true mean resource rewards for which the estimated and true optimal allocations coincide. Knowing this lower bound, these algorithms choose an *exploration constant* $L \geq 1/\epsilon^2$ so that $N'L \log t$ time steps spent in exploration is sufficient to result in resource-activity pair reward estimates that are within ϵ of the true ones with a very high probability.

However, ϵ depends on the suboptimality gap $\Delta_{\min}$ which is a function of the true mean resource rewards unknown to the agents at the beginning. This problem can be solved in the following way for both DLOE and DLC. Here we explain it only for DLOE and leave the explanation for DLC since it is similar. Instead of using the exploration constant L, DLOE uses an increasing exploration function $L(t)$ such that $L(1) = 1$ and $L(t) \to \infty$ as $t \to \infty$. In doing so, the requirement $L(t) \geq 1/\epsilon^2$ is satisfied after some finite number of time steps which we denote by T_0. In the worst case, an amount MT_0 in regret will come from these time steps where $L(t) < 1/\epsilon^2$. After T_0, only a finite amount of (time-independent) regret will result from incorrect calculations of the optimal allocation due to the inaccuracy in estimates. Since DLOE explores only if the least explored resource-activity pair is explored less than $L(t) \log t$ times, regret due to explorations will be bounded by $MN'L(t) \log t$.

Since the order of explorations with $L(t)$ is greater than with constant L, the order of exploitations is less than the case with constant L. Therefore, the order of regret due to incorrect calculations of the optimal allocation and communication at the beginning of exploitation blocks after T_0 is less than the corresponding regret terms when L is constant. Thus, instead of having $O(\log t)$ regret, without a lower bound on ϵ, the proposed modification achieves $O(L(t) \log t)$ regret.

CHAPTER VIII

An Online Contract Selection Problem as a Bandit Problem

In this chapter we study an online contract selection problem, and propose learning algorithms with sublinear regret. In an online contract selection problem there is a seller who offers a bundle of contracts to buyers arriving sequentially over time. The goal of the seller is to maximize its total expected profit up to the final time T, by learning the best bundle of contracts to offer. However, the seller does not know the best bundle of contracts beforehand because initially it does not know the preferences of the buyers.

Assuming that the buyers' preferences change stochastically over time, our goal in this chapter is to design learning algorithms for the seller to maximize its expected profit. Specifically, we assume that the preferences of a buyer depends on its *type*, and is given by a payoff function depending on the type of the buyer. The type of the buyer at time step t is drawn from a distribution not known by the seller, independently from the other time steps. Obviously, the best bundle of contracts (which maximizes the sellers expected profit) depends on the distribution of the buyers' type and the preferences of the buyers.

We assume that the seller can choose what to offer from a continuum of contracts, but it should choose a finite number of contracts to offer simultaneously. We show

that if the buyers' payoff function has a special property which we call the ordered preferences property, then there exists learning algorithms for the seller by which the seller can estimate the type distribution of the buyers by offering a set of contracts, and observing which contract is accepted by the buyer. Then, the seller can compute the expected payoff of different bundles of contracts using the estimated type distribution.

The online contract selection problem can be viewed as a combinatorial multi-armed bandit problem, where each arm is a vector (bundle) of contracts, and each component of the vector can be chosen from an interval of the real line. Two aspects that make this problem harder than the classical finite-armed bandit problem that we studied in the rest of the thesis are: (i) uncountable number of contracts; (ii) exponential number of arms in the number of contracts. We can overcome (i) by offering bundles of sufficiently *closely spaced* contracts to form an estimate of the distribution of buyer's type, and (ii) by writing the expected payoff of an arm as a function of the expected payoffs of the contracts in that arm. Different from the previous chapters of the thesis, it is not possible to achieve logarithmic regret over time, when the arm set is uncountable. However, any algorithm with sublinear regret has the nice property that the time averaged expected reward will converge to the optimal expected reward. With this motivation, in this chapter we prove sublinear regret bounds for the contract selection problem. Our bounds scale linearly with the dimension of the problem, i.e., the number of simultaneously offered contracts, which is better than most of the exponentially scaling bounds in the literature for bandit problems with uncountable number of arms. This improvement is due to the dependence of sellers payoff to the single dimensional type distribution of the buyer.

The online learning problem we consider in this chapter involves large strategy sets, combinatorial and contextual elements. Problems with continuum of arms are considered in *Agrawal* (1995b); *Kleinberg* (2004); *Cope* (2009); *Auer et al.* (2007),

where sublinear regret results are derived. Several combinatorial bandit problems are studied in *Gai et al.* (2012a,b), and problems involving stochastic linear optimization are considered in *Bartlett et al.* (2008); *Dani et al.* (2008). Another line of work *Kleinberg et al.* (2008) generalized the continuum armed bandits to bandits on metric spaces. In this setting, the difference between the expected payoffs of a pair of arms is related to the distance between the arms via a *similarity* metric. Contextual bandits, in which context information is provided to the algorithm at each round is studied in *Langford and Zhang* (2007); *Slivkins* (2009). The goal there is to learn the best arm, given the context.

The organization of the rest of this chapter is as follows. In Section 8.1, we define the online contract selection problem, the ordered preferences property, and provide two applications of the problem. We propose a contract learning algorithm with variable number of simultaneous offers at each time step in Section 8.2, and analyze its performance in Section 8.3. Then, we consider a variant of this algorithm with fixed number of offers in Section 8.4. Finally, we discuss the similarities and the differences between our work and the related work in Section 8.5.

8.1 Problem Formulation and Preliminaries

In an online contract selection problem there is a seller who offers a bundle of $m \in \{1, 2, \ldots\}$ contracts $\boldsymbol{x} \in \mathcal{X}_m$, where

$$\mathcal{X}_m := \{(x_1, x_2, \ldots, x_m), \text{ such that } x_i \in (0, 1], \forall i \in \{1, 2, \ldots, m\}, x_i \leq x_{i+1}\},$$

to buyers arriving sequentially at time steps $t = 1, 2, \ldots, T$, where T is the time horizon. Let $\boldsymbol{x}(t)$ be the bundle offered by the seller at time t. The buyer can accept a single contract $y \in \boldsymbol{x}(t)$ and pay y to the seller, or it can reject all of the offered

contracts and pay 0 to the seller. Profit of the seller by time T is

$$\sum_{t=1}^{T}(u_s(t) - c_s(t)),$$

where $u_s(t)$ represents the revenue/payoff of the seller at time t and $c_s(t)$ is any cost associated with offering the contracts at time t. We have $u_s(t) = x$ if contract x is accepted by the buyer at time t, $u_s(t) = 0$ if none of the offered contracts at time t is accepted by the buyer at time t.

The buyer who arrives at time t has type θ_t which encodes its preferences into a payoff function. At each time step, the type of the buyer present at that time step is drawn according to the probability density function $f(\theta)$ on $[0, 1]$ independently from the other time steps. We assume that buyer's type density is bounded, i.e.

$$f_{\max} := \sup_{\theta \in [0,1]} f(\theta) < \infty.$$

Neither θ_t nor $f(\theta)$ is known by the seller at any time. Therefore, in order maximize its profit, the seller should learn the best set of contracts over time. The expected profit of the seller over time horizon T is given by

$$E\left[\sum_{t=1}^{T} u_s(t) - c_s(t)\right],$$

where the expectation is taken with respect to buyer's type distribution $f(\theta)$ and the seller's offering strategy. Our goal in this chapter is to develop online learning algorithms for the seller to maximize its expected profit over time horizon T.

Let $U_B(x, \theta) : [0, 1] \times [0, 1] \to \mathbb{R}$ represent the payoff function of type θ buyer, which is a function of the contract accepted by the buyer. We assume that the seller knows U_B. For example, when the contracts represent data plans of wireless service providers, the service provider can know the worth of a 2gb contract to a buyer who

only needs 1gb a month. For instance, the amount of payment for the 2gb contract that exceeds the payment for a 1gb can represent the loss of the buyer. Similarly, a 500mb contract to a buyer who needs 1gb a month can have a cost equal to the 500mb shortage in data. Of course there should be a way to relate the monetary loss with the data loss, which can be captured by coefficients multiplying these two. These coefficients can also be known by the seller by analyzing previous consumer data.

Based on its payoff function, the buyer either selects a contract from the offered bundle or it may reject all of the contracts in the bundle. If $\boldsymbol{x} = (x_1, x_2, \ldots, x_m)$ is offered to a type θ buyer it will accept a contract randomly from the set

$$\underset{x \in \{0, x_1, \ldots, x_m\}}{\arg\max} \; U_B(x, \theta),$$

where $x = 0$ implies that the buyer does not accept any of the offered contracts. Since the seller knows the buyer's payoff function $U_B(x, \theta)$, for a given bundle of contracts $\boldsymbol{x} = (x_1, x_2, \ldots, x_m)$, it can compute which contracts will be accepted as a function of the buyer's type. For $y \in \boldsymbol{x}$, let $I_y(\boldsymbol{x})$ be the acceptance region of contract y, which is the values of θ for which contact y will be accepted from the bundle $\boldsymbol{x}$. We assume that the buyers payoff function induces *ordered preferences*, which means that for a bundle of contracts $(x_1, \ldots, x_m)$, the values of θ for which x_i is accepted only depends on x_{i-1}, x_i and x_{i+1}, and

$$I_{x_{i-1}}(\boldsymbol{x}) < I_{x_i}(\boldsymbol{x}) < I_{x_{i+1}}(\boldsymbol{x}),$$

for all $i \in \{1, 2, \ldots, m-1\}$, which means that the acceptance regions are ordered.

Assumption VIII.1. ***Ordered Preferences.*** *$U_B(x, \theta)$ induces ordered preferences which means that for any $\boldsymbol{x} \in \mathcal{X}$, $I_{x_i}(\boldsymbol{x}) = (g(x_{i-1}, x_i), g(x_i, x_{i+1})]$. The function g is such that $g(x, y) < g(y, z)$ for $x < y < z$, and g is Hölder continuous with constant L*

and exponent α, i.e.,

$$|g(x_1, x_2) - g(y_1, y_2)| \leq L\sqrt{(|x_1 - y_1|^2 + |x_2 - y_2|^2)}^{\alpha}.$$

Although the assumption on $U_B(x, \theta)$ is implicit, it is satisfied by many common payoff functions. Below we provide several examples. For notational convenience for any bundle of contracts $(x_1, x_2, \ldots, x_m)$, let $x_0 = 0$, $x_{m+1} = 1$ and $g(x_m, 1) = 1$.

Example VIII.2. Wireless Data Plan Contract. In this case payoff function for the buyers is given by

$$U_B(x, \theta) = h(a(x - \theta)^+ + b(\theta - x)^+),$$

where

$$(x - y)^+ = \max\{0, x - y\},$$

and $h : \mathbb{R}_+ \to \mathbb{R}$ is a decreasing function. For data plan contracts, $(x-\theta)^+$ corresponds to loss in accepting a contract which offers data less than the demand, while $(\theta - x)^+$ corresponds to loss in accepting a contract which offers data more than the demand but have a higher price than the price of the demanded data service. The coefficients a and b relate the buyers weighting of these losses. For this payoff function, the accepted contract from any bundle $(x_1, x_2, \ldots, x_m)$ of contracts is given as a function of the buyer's type in Figure 8.1. It is easy to check that the boundaries of the acceptance regions are

$$g(x_{i-1}, x_i) = \frac{bx_{i-1} + ax_i}{a + b}, \ \forall i = 1, 2, \ldots, m.$$

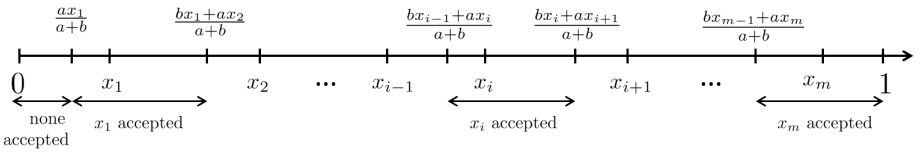

Figure 8.1: acceptance region of bundle $(x_1, \ldots, x_m)$ for $U_B(x, \theta) = h(a(x-\theta)^+ + b(\theta - x)^+)$

Since

$$\begin{aligned} |g(x_1, x_2) - g(y_1, y_2)| &= \left| \frac{b(x_1 - y_1)}{a+b} + \frac{a(x_2 - y_2)}{a+b} \right| \\ &\leq \max\{|x_1 - y_1|, |x_2 - y_2|\} \\ &\leq \sqrt{|x_1 - y_1|^2 + |x_2 - y_2|^2}, \end{aligned}$$

Assumption VIII.1 holds for this buyer payoff function with $L = 1$ and $\alpha = 1$.

Example VIII.3. Secondary Spectrum Contract. Consider a secondary spectrum market, where the service provider leases excess spectrum to secondary users. For simplicity, assume that the service provider always have a unit bandwidth available. In general, due to the primary user activity the bandwidth available for leasing at time t is $B_t \in [0, 1]$, however, all our results in this chapter will hold for dynamically changing available bandwidth, provided that the seller pays a penalty to the buyers for any bandwidth it offers but cannot guarantee to a buyer. By this way, the seller can still learn the buyer's type distribution by offering a bundle of contracts $\boldsymbol{x}$ for which there is some $x_i > B_t$. The buyer's payoff function in this case is

$$U_B(x, \theta) = -a(\theta - x)^+ - x,$$

where x is the amount of money that the buyer pays to the seller by accepting contract x and $a > 1$ is a coefficient that relates the tradeoff between the loss in data and monetary loss. For this payoff function it can be shown that the acceptance

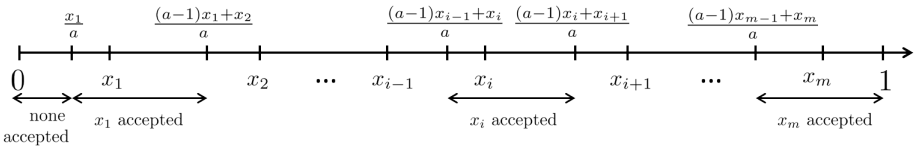

Figure 8.2: acceptance region of bundle $(x_1, \ldots, x_m)$ for $U_B(x, \theta) = -a(\theta - x)^+ - x$

region boundaries are

$$g(x_{i-1}, x_i) = \frac{(a-1)x_{i-1} + x_i}{a}, \ \forall i = 1, 2, \ldots, m,$$

and Assumption VIII.1 holds with $L = 1$, $\alpha = 1$.

By Assumption VIII.1, the expected payoff of a bundle of contracts $\boldsymbol{x} \in \mathcal{X}_m$ to the seller is

$$\begin{aligned} U_s(\boldsymbol{x}) = x_1 P(g(0, x_1) < \theta \leq g(x_1, x_2)) + x_2 P(g(x_1, x_2) < \theta \leq g(x_2, x_3)) \\ + \ldots + x_m P(g(x_{m-1}, x_m) < \theta). \end{aligned}$$

Note that the seller's problem would be solved if it knew $f(\theta)$, since it could compute the best bundle of m contracts, i.e.,

$$\arg\max_{\boldsymbol{x} \in \mathcal{X}_m} U_s(\boldsymbol{x}). \tag{8.1}$$

Remark VIII.4. We do not require that the maximizer of (8.1) is a bundle of m distinct contracts. Note that by definition of the set $\mathcal{X}_m$, the maximizer of (8.1) may be a bundle $(x_1, \ldots, x_m)$ for which $x_i = x_{i+1}$ for some $i \in \{1, 2, \ldots, m-1\}$. This is equivalent to offering $m - 1$ contracts $(x_1, \ldots, x_{i-1}, x_i, x_{i+2}, \ldots, x_m)$. Indeed, our results hold when the seller's goal is to learn the best bundle of contracts that have at most m contracts in it.

The key idea behind the learning algorithms we design for the seller in the sub-

sequent sections is to form estimates of the buyer's distribution by offering different sets of contracts. Each algorithm consists of exploration and exploitation phases. Although we stated that the seller offers m contracts at each time step, in our first algorithm m will vary over time, so we denote it by $m(t)$. In our second algorithm m will be fixed throughout the time horizon T.

We also assume that there is a cost of offering m contracts at the same time which is given by $c(m)$, which increases with m. The sellers objective of maximizing the profit over time horizon T is equivalent to minimizing the regret which is given by

$$R^{\alpha}(T) = T(U_s(\boldsymbol{x}^*) - c(m)) - E^{\alpha}\left[\sum_{t=1}^{T} r(\boldsymbol{x}(t)) - c(m(t))\right], \tag{8.2}$$

where

$$\boldsymbol{x}^* \in \arg\max_{\boldsymbol{x}\in\mathcal{X}_m} U_s(\boldsymbol{x}), \tag{8.3}$$

is the optimal set of m contracts, $r(\boldsymbol{x}(t))$ is the payoff of the seller from the bundle offered at time t, and α is the learning algorithm used by the seller. We will drop the superscript α when the algorithm used by the seller is clear from the context. Note that for any algorithm with sublinear regret, the time averaged expected profit will converge to $U_s(\boldsymbol{x}^*) - c(m)$.

8.2 A Learning Algorithm with Variable Number of Offers

In this section we present a learning algorithm which has distinct exploration and exploitation steps. The algorithm is called *type learning with variable number of offers* (TLVO), whose pseudocode is given in Figure 8.3.

Instead of searching for the best bundle of contracts in $\mathcal{X}_m$ which is uncountable,

the algorithm searches for the best bundle of contracts in the finite set

$$\mathcal{L}_{m,T} := \{\boldsymbol{x} = (x_1, \dots x_m) : x_i \leq x_{i+1} \text{ and } x_i \in \mathcal{K}_T, \forall i \in \{1, \dots, m\}\},$$

where

$$\mathcal{K}_T := \left\{\frac{1}{n_T}, \frac{2}{n_T}, \dots, \frac{n_T - 1}{n_T}\right\}.$$

Here n_T is a non-decreasing function of the time horizon T. Since the best bundle in $\mathcal{L}_{m,T}$ might have an expected reward smaller than the expected reward of the best bundle in $\mathcal{X}_m$, in order to bound the regret due to this difference sublinearly over time, n_T should be adjusted according to the time horizon.

Type Learning with Variable Number of Offers (TLVO)

Parameters: m, T, $z(t), 1 \leq t \leq T$, n_T, $\mathcal{K}_T$, $\mathcal{L}_{m,T}$.
Initialize: set $t = 1$, $N = 0$, $\mu_i = 0, N_i = 0, \forall i \in \mathcal{K}_T$.
while $t \geq 1$ **do**
  if $N < z(t)$ **then**
    EXPLORE
    Offer all contracts in $\mathcal{K}_T$ simultaneously.
    if Any contract $x \in \mathcal{K}_T$ is accepted by the buyer **then**
      Get reward x. Find $k \in \{1, 2, \dots, n_T - 1\}$ such that $k/n_T = x$.
      $++N_k$.
    end if
    $++N$.
  else
    EXPLOIT
    $\mu_i = N_i/N, \forall i \in \{1, 2, \dots, n_T - 1\}$.
    Offer bundle $\boldsymbol{x} = (x_1, \dots, x_m)$, which is a solution to (8.4) based on μ_i's.
    If some $x \in \boldsymbol{x}$ is accepted, get reward x.
  end if
  $++t$.
end while

Figure 8.3: pseudocode of TLVO

Exploration and exploitation steps are sequenced in a deterministic way. This sequencing is provided by a *control function* $z(t)$ which is a parameter of the learning algorithm. Let $N(t)$ be the number of explorations up to time t. If $N(t) < z(t)$, time t will be an exploration step. Otherwise time t will be an exploitation step. While $z(t)$ can be any sublinearly increasing function, we will optimize over $z(t)$ in our analysis.

In an exploration step, TLVO estimates the distribution of buyer's type by simultaneously offering the (possibly large) set of $n_T - 1$ uniformly spaced contracts in $\mathcal{K}_T$. Based on the accepted contract at time t, the seller learns the part of the type space that the buyers type at t lies in, and uses this to form sample mean estimates of the distribution of the buyer's type. We simply call the contract $i/n_T \in \mathcal{K}_T$ as the ith contract. Let θ be the unknown type of the buyer at some exploration step. If ith contract is accepted by the buyer, then the seller knows that

$$g\left(\frac{i-1}{n_T}, \frac{i}{n_T}\right) < \theta \leq g\left(\frac{i}{n_T}, \frac{i+1}{n_T}\right).$$

Let $N_i(t)$ be the number of times contract i is accepted in an exploration step up to t. Then the sample mean estimate of

$$P\left(g\left(\frac{i-1}{n_T}, \frac{i}{n_T}\right) < \theta \leq g\left(\frac{i}{n_T}, \frac{i+1}{n_T}\right)\right),$$

is given by

$$\mu_i(t) := \frac{N_i(t)}{N(t)}.$$

In an exploitation step, TLVO offers a bundle of m contracts chosen from $\mathcal{L}_{m,T}$, which maximizes the seller's estimated expected payoff. For constants θ_l and θ_u, let $\hat{P}_t(\theta_l < \theta \leq \theta_u)$ be the estimate of $P(\theta_l < \theta \leq \theta_u)$ at time t. TLVO computes this

estimate based on the estimates $\mu_i(t)$ in the following way:

$$\hat{P}_t(\theta_l < \theta \leq \theta_u) = \sum_{i=i_-(\theta_l)}^{i_+(\theta_u)} \mu_i(t),$$

where

$$i_-(\theta_l) = \min \left\{ i \in \{1, \ldots, n_T - 1\} \text{ such that } g\left(\frac{i-1}{n_T}, \frac{i}{n_T}\right) \geq \theta_l \right\},$$

and

$$i_+(\theta_u) = \min \left\{ i \in \{1, \ldots, n_T - 1\} \text{ such that } g\left(\frac{i}{n_T}, \frac{i+1}{n_T}\right) \geq \theta_u \right\}.$$

We can write $x_i \in \mathcal{L}_{m,T}$ as k_i/n_T for some $k_i \in \{1, 2, \ldots, n_T - 1\}$. If time t is an exploitation step, TLVO computes the estimated best bundle of contracts $\boldsymbol{x}(t)$ by solving the following optimization problem.

$$\boldsymbol{x}(t) = \arg\max_{\boldsymbol{x} \in \mathcal{L}_{m,T}} \hat{U}_t(\boldsymbol{x}), \tag{8.4}$$

where

$$\begin{aligned} \hat{U}_t(\boldsymbol{x}) := {} & x_1 \hat{P}_t \left(g(0, x_1) < \theta \leq g(x_1, x_2)\right) + x_2 \hat{P}_t \left(g(x_1, x_2) < \theta \leq g(x_2, x_3)\right) \\ & + \ldots + x_m \hat{P}_t \left(g(x_{m-1}, x_m) < \theta\right). \end{aligned}$$

Note that there might be more than one maximizer to (8.4). In such a case, TLVO arbitrarily chooses one of the maximizer bundles. Maximization in (8.4) is a combinatorial optimization problem. In general solution to such a problem is NP-hard. We assume that the solution is provided to the algorithm by an oracle. This is a common assumption in online learning literature, for example used in *Dani et al.*

(2008). Therefore, we do not consider the computational complexity of this operation. Although we do not provide a computationally efficient solution for (8.4), there exists computationally efficient methods for some special cases. We discuss more on this in Section 8.5.

We analyze the regret of TLVO in the next section.

8.3 Analysis of the Regret of TLVO

In this section we upper bound the regret of TLVO. Let

$$S = \{\boldsymbol{x} \in \mathcal{L}_{m,T} : |U_s(\boldsymbol{x}^*) - U_s(\boldsymbol{x})| < \beta n_T^{-\alpha}\},$$

be the set of near-optimal bundles of contracts where α is the Hölder exponent in Assumption VIII.1, and $\beta = 5mf_{\max}L2^{\alpha/2}$ is a constant where L is the Hölder constant in Assumption VIII.1. Denote the complement of S by S^c. Let $T_{\boldsymbol{x}}(t)$ be the number of times $\boldsymbol{x} \in \mathcal{L}_{m,T}$ is offered at exploitation steps by time t. For TLVO, regret given in (8.2) is upper bounded by

$$\begin{aligned} R(T) \leq & \sum_{\boldsymbol{x} \in S} (U_s(\boldsymbol{x}^*) - U_s(\boldsymbol{x})) E\left[T_{\boldsymbol{x}}(T)\right] \\ & + \sum_{\boldsymbol{x} \in S^c} (U_s(\boldsymbol{x}^*) - U_s(\boldsymbol{x})) E\left[T_{\boldsymbol{x}}(T)\right] \\ & + N(T)(U_s(\boldsymbol{x}^*) + c(n_T) - c(m)), \end{aligned} \tag{8.5}$$

by assuming zero worst-case payoff in exploration steps. First term in (8.5) is the contribution of selecting a *nearly* optimal bundle of contracts in exploitation steps, second term is the contribution of selecting a suboptimal bundle of contracts in the exploitation steps, and the third term is the worst-case contribution during the exploration steps to the regret.

The following theorem gives an upper bound on the regret of TLVO.

Theorem VIII.5. *The regret of the seller using TLVO with time horizon T is upper bounded by*

$$R(T) \leq 5mf_{\max}L2^{\alpha/2}n_T^{-\alpha}(T - N(T)) + N(T)(U_s(\boldsymbol{x}^*) + c(n_T) - c(m)) + 2n_T \sum_{t=1}^{T} e^{\frac{-f_{\max}^2 L^2 2^\alpha N(t)}{n_T^{2+2\alpha}}}.$$

Remark VIII.6. In this form, the regret is linear in n_T and T. The first term in the regret decreases with n_T while the second and third terms increase with n_T. Since T is known by the seller, n_T can be optimized as a function of T.

Proof. Let $\delta_{\boldsymbol{x}}^* = U_s(\boldsymbol{x}^*) - U_s(\boldsymbol{x})$. By definition of the set S, we have

$$\begin{aligned}\sum_{\boldsymbol{x}\in S} \delta_{\boldsymbol{x}}^* E[T_{\boldsymbol{x}}(T)] &\leq \max_{\boldsymbol{x}\in S} \delta_{\boldsymbol{x}}^* \sum_{\boldsymbol{x}\in S} E[T_{\boldsymbol{x}}(T)] \\ &\leq \beta n_T^{-\alpha}(T - N(T)). \end{aligned} \tag{8.6}$$

Next, we consider the term

$$\sum_{\boldsymbol{x}\in S^c} (U_s(\boldsymbol{x}^*) - U_s(\boldsymbol{x}))E\left[T_{\boldsymbol{x}}(T)\right].$$

Note that even if we bound $E\left[T_{\boldsymbol{x}}(T)\right]$ for all $\boldsymbol{x} \in S^c$, in the worst case $|S^c| = cn_T^m$, for some $c > 0$. Therefore a bound that depends on n_T^m will scale badly for large m. To overcome this difficulty, we will show that if the distribution function has sufficiently accurate sample mean estimates $\mu_i(t)$ for all

$$p_i := P\left(g\left(\frac{i-1}{n_T}, \frac{i}{n_T}\right) < \theta \leq g\left(\frac{i}{n_T}, \frac{i+1}{n_T}\right)\right),\ i \in \{1, 2, \ldots, n_T - 1\},$$

then the probability that some bundle in S^c is offered will be small. Let $T_{S^c}(t)$ be the

number of times a bundle from S^c is offered in exploitation steps by time t. We have

$$\sum_{\boldsymbol{x}\in S^c}(U_s(\boldsymbol{x}^*)-U_s(\boldsymbol{x}))E\left[T_{\boldsymbol{x}}(T)\right]\leq E\left[T_{S^c}(T)\right], \tag{8.7}$$

where

$$E\left[T_{S^c}(T)\right]=E\left[\sum_{t=1}^{T}I(\boldsymbol{x}(t)\in S^c)\right]=\sum_{t=1}^{T}P(\boldsymbol{x}(t)\in S^c). \tag{8.8}$$

For convenience let $x_0=0, x_{m+1}=1$ and $g(x_m,x_m+1)=1$. For any $x_i\in\boldsymbol{x}\in\boldsymbol{\mathcal{L}}_{m,T}$, we can write

$$\begin{aligned}P\left(g(x_{i-1},x_i)<\theta\leq g(x_i,x_{i+1})\right)&=P(g(x_{i-1},x_i)<\theta\leq i_-(g(x_{i-1},x_i)))\\&+\sum_{i=i_-(g(x_{i-1},x_i))}^{i_+(g(x_i,x_{i+1}))}p_i-P(g(x_i,x_{i+1})<\theta\leq i_+(g(x_i,x_{i+1}))).\end{aligned}$$

Let

$$\begin{aligned}err_{\boldsymbol{x}}(x_i)=|&P(g(x_{i-1},x_i)<\theta\leq i_-(g(x_{i-1},x_i)))\\&-P(g(x_i,x_{i+1})<\theta\leq i_+(g(x_i,x_{i+1})))\ |.\end{aligned}$$

Since

$$(g(x_{i-1},x_i),i_-(g(x_{i-1},x_i))]\subset(i_-(g(x_{i-1},x_i))-1,i_-(g(x_{i-1},x_i))],$$

and

$$(g(x_i,x_{i+1}),i_+(g(x_i,x_{i+1}))]\subset(i_+(g(x_i,x_{i+1}))-1,i_+(g(x_i,x_{i+1}))],$$

by Assumption VIII.1, we have for any $x_i \in \boldsymbol{x} \in \mathcal{L}_{m,T}$

$$
\begin{aligned}
&err_{\boldsymbol{x}}(x_i) \\
&\leq \max\{P(g(x_{i-1}, x_i) < \theta \leq i_-(g(x_{i-1}, x_i))), P(g(x_i, x_{i+1}) < \theta \leq i_+(g(x_i, x_{i+1})))\} \\
&\leq f_{\max} L 2^{\alpha/2} n_T^{-\alpha}. \qquad (8.9)
\end{aligned}
$$

Consider the event

$$
\xi_t = \bigcap_{i=1}^{n_T - 1} \left\{ |\mu_i(t) - p_i| \leq \frac{(\beta - m f_{\max} L 2^{\alpha/2}) n_T^{-\alpha}}{4 n_T m} \right\}.
$$

If ξ_t happens, then for any $1 \leq a < b \leq n_T - 1$

$$
\begin{aligned}
\left| \sum_{i=a}^{b} \mu_i(t) - \sum_{i=a}^{b} p_i(t) \right| &\leq (b-a) \frac{(\beta - m f_{\max} L 2^{\alpha/2}) n_T^{-\alpha}}{4 n_T m} \\
&\leq \frac{(\beta - m f_{\max} L 2^{\alpha/2}) n_T^{-\alpha}}{4m},
\end{aligned}
$$

which implies that for any $\boldsymbol{x} \in \mathcal{L}_{m,T}$

$$
\begin{aligned}
|\hat{U}_t(\boldsymbol{x}) - U_s(\boldsymbol{x})| &\leq x_1 \sum_{i=i_-(g(0,x_1))}^{i_+(g(x_1,x_2))} |\mu_i(t) - p_i| + err_{\boldsymbol{x}}(x_i) + \dots \\
&\quad + x_m \sum_{i=i_-(g(x_{m-1},x_m))}^{i_+(g(x_m,x_{m+1}))} |\mu_i(t) - p_i| + err_{\boldsymbol{x}}(x_m) \\
&\leq \sum_{i=i_-(g(0,x_1))}^{i_+(g(x_1,x_2))} |\mu_i(t) - p_i| + err_{\boldsymbol{x}}(x_i) + \dots \\
&\quad + \sum_{i=i_-(g(x_{m-1},x_m))}^{i_+(g(x_m,x_{m+1}))} |\mu_i(t) - p_i| + err_{\boldsymbol{x}}(x_m) \\
&\leq 2 m f_{\max} L 2^{\alpha/2} n_T^{-\alpha}. \qquad (8.10)
\end{aligned}
$$

Let $\boldsymbol{y}^* = \arg\max_{\boldsymbol{x}\in\mathcal{L}_{m,T}} U_s(\boldsymbol{x})$. By Assumption (VIII.1) we have

$$U_s(\boldsymbol{x}^*) - U_s(\boldsymbol{y}^*) \leq m f_{\max} L 2^{\alpha/2} n_T^{-\alpha} \ .$$

Then, using the definition of the set S^c, which denotes the set of suboptimal bundles of contracts, for any $\boldsymbol{x} \in S^c$, we have

$$U_s(\boldsymbol{y}^*) - U_s(\boldsymbol{x}) > (\beta - m f_{\max} L 2^{\alpha/2}) L n_T^{-\alpha} = 4 m f_{\max} L 2^{\alpha/2} n_T^{-\alpha}.$$

Since by (8.10) the estimated payoff of any bundle $\boldsymbol{x} \in \mathcal{L}_{m,T}$ is within $2 m f_{\max} L 2^{\alpha/2} n_T^{-\alpha}$ of its true value, the event ξ_t implies that for any $\boldsymbol{x} \in S^c$

$$\hat{U}_t(\boldsymbol{x}) \leq \hat{U}_t(\boldsymbol{y}^*),$$

which means

$$\xi_t \subset \{\hat{U}_t(\boldsymbol{x}) \leq \hat{U}_t(\boldsymbol{y}^*), \forall \boldsymbol{x} \in S^c\},$$

and

$$\{\hat{U}_t(\boldsymbol{x}) > \hat{U}_t(\boldsymbol{y}^*) \text{ for some } x \in S^c\} \subset \xi_t^c.$$

Therefore

$$\begin{aligned} P(\boldsymbol{x}(t) \in S^c) &\leq P\left(\bigcup_{i=1}^{n_T - 1} \{|\mu_i(t) - p_i| > \frac{(\beta - m f_{\max} L 2^{\alpha/2}) n_T^{-\alpha}}{4 n_T m}\} \right) \\ &\leq \sum_{i=1}^{n_T - 1} P\left(|\mu_i(t) - p_i| > \frac{(\beta - m f_{\max} L 2^{\alpha/2}) n_T^{-\alpha}}{4 n_T m} \right) \\ &\leq 2 n_T e^{\frac{-f_{\max}^2 L^2 2^\alpha N(t)}{n_T^{2+2\alpha}}}, \end{aligned}$$

by using the Chernoff-Hoeffding bound given in Lemma A.7. Using the last result in (8.7), we get

$$\sum_{\boldsymbol{x}\in S^c}(U_s(\boldsymbol{x}^*)-U_s(\boldsymbol{x}))E\left[T_{\boldsymbol{x}}(T)\right]\leq 2n_T\sum_{t=1}^{T}e^{\frac{-f_{\max}^2L^2 2^\alpha N(t)}{n_T^{2+2\alpha}}}. \tag{8.11}$$

We get the main result by substituting (8.6) and (8.11) into (8.5). □

The following corollary gives a sublinear regret result for a special case of parameters.

Corollary VIII.7. *When the cost of offering n contracts simultaneously, i.e., $c(n)\leq n^\gamma$, for all $0<n<T$, for some $\gamma>0$, the regret of the seller that runs TLVO with*

$$n_T=\left\lfloor (f_{\max}L2^{\alpha/2})^{\frac{2}{4+2\alpha}}\left(\frac{T}{\log T}\right)^{\frac{1}{4+2\alpha}}\right\rfloor,$$
$$z(t)=\left(\frac{1}{f_{\max}L2^{\alpha/2}}\right)^{\frac{2+6\alpha}{2+\alpha}}\left(\frac{T}{\log T}\right)^{\frac{2+2\alpha}{4+2\alpha}}\log t,$$

where $\lfloor y\rfloor$ is the largest integer smaller than equal to y, is upper bounded by

$$\begin{aligned}R(T)\leq\ & 5m(f_{\max}L2^{\alpha/2})^{\frac{2}{2+\alpha}}(\log T)^{\frac{\alpha}{4+2\alpha}}T^{\frac{4+\alpha}{4+2\alpha}}\\ &+\left(\frac{1}{f_{\max}L2^{\alpha/2}}\right)^{\frac{2+6\alpha}{2+\alpha}}(\log T)^{\frac{2+\gamma}{4+2\alpha}}T^{\frac{2+2\alpha+\gamma}{4+2\alpha}}\\ &+2(f_{\max}L2^{\alpha/2})^{\frac{1}{2+\alpha}}(\log T+1)(\log T)^{\frac{1}{4+2\alpha}}T^{\frac{1}{4+2\alpha}}.\end{aligned}$$

Hence

$$R(T)=O(mT^{(2+2\alpha+\gamma)/(4+2\alpha)}(\log T)^{2/(4+2\alpha)}),$$

which is sublinear in T for $\gamma<2$.

Proof. We want

$$e^{\frac{-f_{\max}^2 L^2 2^\alpha N(t)}{n_T^{2+2\alpha}}} \leq \frac{1}{t}.$$

For this,we should have

$$\frac{-f_{\max}^2 L^2 2^\alpha N(t)}{n_T^{2+2\alpha}} \leq -\log t,$$

which implies

$$N(t) \geq \frac{(n_T)^{2+2\alpha}}{f_{\max}^2 L^2 2^\alpha} \log t. \tag{8.12}$$

Note that at each time t either $N(t) \geq z(t)$ or $z(t) - 1 \leq N(t) < z(t)$ so we chose

$$z(t) = \frac{(n_T)^{2+2\alpha}}{f_{\max}^2 L^2 2^\alpha} \log t + 1.$$

Note that $z(t)$ in this form depends on n_T which we have not fixed yet. To have minimum regret, we need to balance the first and second terms of the regret given in Theorem VIII.5. Thus $T/n_T \approx N(T)n_T$. Since n_T must be an integer, substituting (8.12) into $N(T)$, we have

$$n_T = \left\lfloor (f_{\max} L 2^{\alpha/2})^{\frac{2}{4+2\alpha}} \left(\frac{T}{\log T}\right)^{\frac{1}{4+2\alpha}} \right\rfloor.$$

Proof is completed by substituting these into the result of Theorem VIII.5. □

8.4 A Learning Algorithm with Fixed Number of Offers

One drawback of TLVO is that in exploration steps it simultaneously offers $n_T - 1$ contracts, and this number increases sublinearly with T. Usually, the seller will offer

different bundles of contracts but it will include same number of contracts in each bundle. For example, a wireless service provider usually adds new data plans by removing one of the current data plans, thus the total number of data plans offered does not change significantly over time. In this section, we are interested in the case when the seller is limited to offering m contracts at every time step.

In this case, the exploration step of TLVO will not work. Because of this, we propose the algorithm *type learning with fixed number of offers* (TLFO) that always offers m contracts simultaneously. TLFO differs from TLVO only in its exploration phase. Each exploration phase of TLFO lasts multiple time steps. Instead of simultaneously offering $n_T - 1$ uniformly spaced contracts at an exploration step, TLFO has an exploration phase of $\lceil (n_T - 1)/(m-2) \rceil$ steps indexed by $l = 1, 2, \ldots, \lceil (n_T - 1)/(m-2) \rceil$. The idea behind TLFO is to estimate the buyer's type distribution from the estimates of the segments of the buyer's type distribution over different time steps of the same exploration phase. Let time t be the start of an exploration phase for TLFO. Let $l' = \lceil (n_T - 1)/(m-2) \rceil$ denote the last step of the exploration phase. Next, we define the following bundles of m contracts. The overlapping portions of these bundles are shown in Figure 8.4 for $l = 1, 2, \ldots, l'$.

$$
\begin{aligned}
\mathcal{B}_1 &= \left\{ \frac{1}{n_T}, \frac{2}{n_T}, \ldots, \frac{m}{n_T} \right\}, \\
\tilde{\mathcal{B}}_1 &= \left\{ \frac{1}{n_T}, \frac{2}{n_T}, \ldots, \frac{m-1}{n_T} \right\}, \\
\mathcal{B}_{l'} &= \left\{ \frac{n_T - m}{n_T}, \frac{n_T - m + 1}{n_T}, \ldots, \frac{n_T - 1}{n_T} \right\}, \\
\tilde{\mathcal{B}}_{l'} &= \left\{ \frac{(l'-1)m - 2(l'-1) + 2}{n_T}, \frac{(l'-1)m - 2(l'-1) + 3}{n_T}, \ldots, \frac{n_T - 1}{n_T} \right\},
\end{aligned}
$$

and for $l \in \{2, \ldots, l' - 1\}$

$$
\mathcal{B}_l = \left\{ \frac{(l-1)m - 2(l-1) + 1}{n_T}, \frac{(l-1)m - 2(l-1) + 2}{n_T}, \ldots, \frac{lm - 2(l-1)}{n_T} \right\},
$$

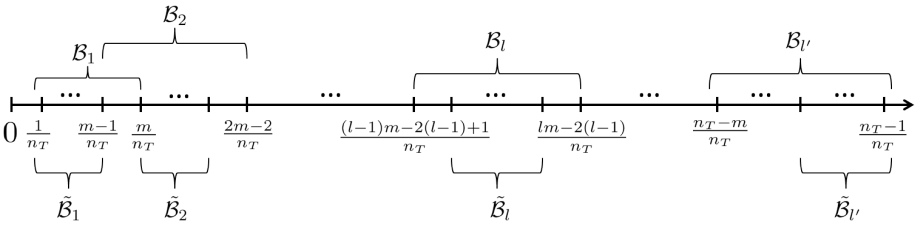

Figure 8.4: bundles of m contracts offered in exploration steps $l = 1, 2, \ldots, l'$ in an exploration phase

$$\tilde{\mathcal{B}}_l = \left\{ \frac{(l-1)m - 2(l-1) + 2}{n_T}, \frac{(l-1)m - 2(l-1) + 3}{n_T}, \ldots, \frac{lm - 2(l-1) - 1}{n_T} \right\}.$$

Similar to TLVO let N and N_k, $k \in \{1, 2, \ldots, n_T - 1\}$ be the counters that are used to form type distribution estimates which are set to zero initially. Basically, at an exploitation step the estimates $\mu_k = N_k/N$ are formed based on the current values of N_k and N. Different from the analysis of TLVO, $N(t)$ which is the value of counter N at time t represents the number of completed exploration phases by time t, not the number of exploration steps by time t. The condition $N(t) < z(t)$ is checked at the end of each exploration phase or exploitation step, and if the condition is true, a new exploration phase starts. In the first exploration step of the exploration phase, TLFO offers the bundle $\mathcal{B}_1$. If a contract $k/n_T \in \tilde{\mathcal{B}}_1$ is accepted, N_k is incremented by one. In the lth exploration step, $l \in \{2, \ldots, l' - 1\}$, it offers the bundle $\mathcal{B}_l$. If a contract $k/n_T \in \tilde{\mathcal{B}}_l$ is accepted, N_k is incremented by one. In the last exploration step l', it offers $\mathcal{B}_{l'}$. If a contract $k/n_T \in \tilde{\mathcal{B}}_{l'}$ is accepted, N_k is incremented by one. At the time t' when all the exploration steps in the exploration phase are completed, N is incremented by one. Pseudocode of the exploration phase for TLFO is given in Figure 8.5.

Exploration phase of TLFO.

1: **for** $l = 1, 2, \ldots, \lceil (n_T - 1)/(m-2) \rceil$ **do**
2: Offer bundle $\mathcal{B}_l$.
3: Let $k/n_T \in \mathcal{B}_l$ be the accepted contract. Get reward k/n_T.
4: **if** $k/n_T \in \tilde{\mathcal{B}}_l$ **then**
5: $++N_k$
6: **end if**
7: $++t$
8: **end for**
9: $++N$

Figure 8.5: pseudocode of the exploration phase of TLFO

Note that regret of the seller in this case is upper bounded by

$$\begin{aligned} R(T) \leq & \sum_{\boldsymbol{x} \in S} (U_s(\boldsymbol{x}^*) - U_s(\boldsymbol{x})) E\left[T_{\boldsymbol{x}}(T)\right] \\ & + \sum_{\boldsymbol{x} \in S^c} (U_s(\boldsymbol{x}^*) - U_s(\boldsymbol{x})) E\left[T_{\boldsymbol{x}}(T)\right] \\ & + N(T) \lceil (n_T - 1)/(m-2) \rceil (U(\boldsymbol{x}^*)). \end{aligned} \tag{8.13}$$

By the exploration phase of TLFO, the accuracy of the estimates $\mu_i(t)$ at the beginning of each exploitation block is the same as TLVO. Moreover, the the regret due to near-optimal exploitations can be upper bounded by the same term as in TLVO. Only the regret due to explorations changes. The number of exploration steps of TLFO is about $(n_T - 1)/(m-2)$ times the number of exploration steps of TLVO, but there is no cost of offering more than m (possibly a large number of) contracts in TLFO. The following theorem and corollary gives an upper bound on the regret of TLFO, by using an approach similar to the proofs of Theorem VIII.5 and Corollary VIII.7.

Theorem VIII.8. *The regret of seller using TLFO with time horizon T is upper bounded by*

$$R(T) \leq 5m f_{\max} L 2^{\alpha/2} n_T^{-\alpha} (T - N(T)) + N(T) \left(\frac{n_T - 1}{m-2} + 1 \right) U(\boldsymbol{x}^*)$$

$$+ 2n_T \sum_{t=1}^{T} e^{\frac{-f_{\max}^2 L^2 2^\alpha N(t)}{n_T^{2+2\alpha}}}.$$

Since TLFO simultaneously offers m contracts both in explorations and exploitations, its regret does not depend on the cost function $c(.)$ of offering multiple contracts simultaneously. Therefore our sublinear regret bound always holds independent of $c(.)$.

Corollary VIII.9. *When the seller runs TLFO with time horizon T and*

$$n_T = \left\lfloor (f_{\max} L 2^{\alpha/2})^{\frac{2}{4+2\alpha}} \left(\frac{T}{\log T}\right)^{\frac{1}{4+2\alpha}} \right\rfloor,$$
$$z(t) = \left(\frac{1}{f_{\max} L 2^{\alpha/2}}\right)^{\frac{2+6\alpha}{2+\alpha}} \left(\frac{T}{\log T}\right)^{\frac{2+2\alpha}{4+2\alpha}} \log t,$$

we have

$$R(T) = C_m + mT^{(3+2\alpha)/(4+2\alpha)} (\log T)^{2/(4+2\alpha)},$$

uniformly over T for some constant $C_m > 0$. Hence,

$$R(T) = O(mT^{(3+2\alpha)/(4+2\alpha)} (\log T)^{2/(4+2\alpha)}).$$

8.5 Discussion

A contract design problem for a secondary spectrum market is studied in *Sheng and Liu* (2012). In this work the authors assume that the type distribution $f(\theta)$ is known by the seller, and they characterize the optimal set of contracts. They show that when the channel condition is common to all types, i.e., probability that the channel is idle is the same for all types of users, a computationally efficient procedure exists for choosing the best bundle of m contracts out of $\mathcal{L}_{m,T}$. This procedure can

be used by the seller to efficiently solve (8.4).

In the fixed number of offers case, we assume that at each time step the seller offers a bundle $(x_1, x_2, \ldots, x_m) \subset \mathcal{X}_m \subset [0,1]^m$. Therefore, the strategy set is a subset of the m-dimensional unit cube. Because of this relation, we can compare the performance of our contract learning algorithms with bandit algorithms for high dimensional strategy sets. For example, if the reward from any bundle $\boldsymbol{x}$ were of linear form, i.e., $U_s(\boldsymbol{x}) = \boldsymbol{C} \cdot \boldsymbol{x}$ for some $\boldsymbol{C} \in \mathbb{R}^m$, then the online stochastic linear optimization algorithm in *Dani et al.* (2008) would give regret $O((m \log T)^{3/2}\sqrt{T})$. However, in our problem $U_s(\boldsymbol{x})$ is not a linear function, thus this approach will not work. One can also show that in general $U_s(\boldsymbol{x})$ is neither convex or nor concave, therefore any bandit algorithm exploiting these properties will not work in our setting.

Another work, *Bubeck et al.* (2008), considers online linear optimization in a general topological space. For an m-dimensional strategy space, they prove a lower bound of $\tilde{O}(T^{(m+1)/(m+2)})$. Therefore, our bound is better than their lower bound for $m > 2 + 2\alpha$. This is not a contradiction since in our problem it is the type θ that is drawn independently at each time step, not the rewards of the individual contracts, and we focus on estimating the expected rewards of arms (bundles of contracts) from the type distribution. In the same paper, a $\tilde{O}(\sqrt{T})$ regret upper bound is also proved, under the assumption that the mean reward function is locally equivalent to a bi-Hölder function near any maxima, i.e., $\exists c_1, c_2, \epsilon_0 > 0$ such that for $||\boldsymbol{x} - \boldsymbol{x'}|| \leq \epsilon_0$

$$c_1||\boldsymbol{x} - \boldsymbol{x}'||^\alpha \leq |U_s(\boldsymbol{x}) - U_s(\boldsymbol{x}')| \leq c_2||\boldsymbol{x} - \boldsymbol{x}'||^\alpha.$$

However, in this chapter, we only require a Hölder condition for the boundaries of the acceptance regions (see Assumption VIII.1), which implies that

$$|U_s(\boldsymbol{x}) - U_s(\boldsymbol{x}')| \leq c_3||\boldsymbol{x} - \boldsymbol{x}'||^\alpha,$$

for some $c_3 > 0$ and $\forall \boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}_m$.

CHAPTER IX

Conclusions and Future Work

In this thesis we consider bandit problems in an agent-centric setting and develop online learning algorithms with provable performance guarantees by considering computation, decentralization and communication aspects of the problems. For single-agent bandits, our results depict the tradeoff between computational complexity and performance of the learning algorithms. Specifically, we show that in the general (restless) Markovian model, learning the optimal solution with logarithmic regret is possible, however the learning algorithm is computationally intractable. When we restrict our attention to a special case of the general Markovian model, we can design polynomial complexity algorithms that are approximately optimal in terms of the average reward. When we change our performance objective from the optimal solution to the best static solution, we prove that linear complexity algorithms with logarithmic weak regret exist.

For multi-agent bandits, we conclude that the achievable performance depends on the communication and feedback structure of the agents. Due to the complexities rising from informational decentralization, limited computational power and memory requirements for the agents, in most of the multi-agent applications such as cognitive radio networks, we only consider weak regret as the performance measure. Specifically, we show that as the degree of feedback and communication between the agents

increases, weak regret of the agents decreases from sublinear to logarithmic rates.

Apart from the work on single-agent and multi-agent bandits, we also consider a novel application of the bandit problem, i.e., the online contract selection problem. In this thesis, this problem is modeled as a combinatorial bandit with an uncountable number of arms. The formulation and analysis of this problem is significantly different then the single-agent and multi-agent models with finite number of independent arms. We propose a learning algorithm for the contract seller so that it can achieve sublinear regret with respect to the optimal set of contracts. Different from the related work in bandit problems with large strategy sets, our regret bound only depends linearly on the dimension of the problem.

In this thesis, the agents are collaborative and they simply follow the algorithms that are given to them. However, there are multi-agent applications in which the agents selfishly try to maximize their own payoff. For example, in a wireless power control problem, an agent's payoff an a channel depends on the power levels of the other agents on the same channel. Payoff of the agent usually increases with its own transmit power and decreases with the other agent's transmit power. When the agents selfishly aim to maximize their own payoffs, they suffer from *the tragedy of the commons* (see *Hardin* (2009)), meaning that all agents get low payoffs. This example illustrates that the selfish objectives of the agents may not coincide with the objective of maximizing the system utility, which, for example, can be the sum of the payoffs of all the agents. Several approaches based on game theory and mechanism design are proposed to overcome this problem. For example, in *Kakhbod and Teneketzis* (2010), authors consider the wireless power control problem, and propose a mechanism in which a central entity enforces payments on the agents based on their reported types. This mechanism is shown to achieve the desired outcome as an equilibrium of the game induced by the mechanism. A different approach is adopted in *Xiao and van der Schaar* (2012), where authors consider the wireless power control problem

as a repeated game between the agents, characterize the conditions under which the desired outcome is an equilibrium point, and propose a deviation proof policy that achieves this equilibrium. In both of the approaches mentioned above, agents know their payoff function, but because of the informational decentralization they do not perfectly know the payoffs or actions of other agents. However, in the multi-agent learning settings we consider, agents do not know their (expected) payoffs but learn them over time. For example, when an agent is a recently deployed wireless device, it may need to explore the channels to learn which ones offer higher data rates on average. As another example, when an agent is a business venture it needs to experiment different investment strategies to find out which one offers higher payoffs.

An open question is under what conditions a desired objective can be achieved when strategic agents dynamically learn their payoffs from the actions. Clearly, this problem is harder to solve than a learning problem involving collaborative agents because of the strategic behavior based on incomplete knowledge which may result in a large set of outcomes. As a future work, we plan to develop novel techniques that combine learning with game theory and mechanism design to achieve various performance objectives when the agents are strategic.

APPENDICES

APPENDIX A

Results from the Theory of Large Deviations

In this appendix we list results from large deviations theory which are used through our proofs. The following two lemmas bound the probability of a large deviation from the stationary distribution of a Markov chain.

Lemma A.1. *[Theorem 3.3 from Lezaud (1998)] Consider a finite-state, irreducible Markov chain* $\{X_t\}_{t\geq 1}$ *with state space* S*, matrix of transition probabilities* P*, an initial distribution* $\mathbf{q}$ *and stationary distribution* π*. Let* $V_{\boldsymbol{q}} = \left\| (\frac{q_x}{\pi_x}, x \in S) \right\|_2$*. Let* $\dot{P} = P'P$ *be the multiplicative symmetrization of* P *where* P' *is the adjoint of* P *on* $l_2(\pi)$*. Let* $\epsilon = 1 - \lambda_2$*, where* λ_2 *is the second largest eigenvalue of the matrix* $\dot{P}$*.* ϵ *will be referred to as the eigenvalue gap of* $\dot{P}$*. Let* $f : S \to \mathbb{R}$ *be such that* $\sum_{y\in S} \pi_y f(y) = 0$*,* $\|f\|_\infty \leq 1$ *and* $0 < \|f\|_2^2 \leq 1$*. If* $\dot{P}$ *is irreducible, then for any positive integer* T *and all* $0 < \gamma \leq 1$

$$P\left(\frac{\sum_{t=1}^{T} f(X_t)}{T} \geq \gamma \right) \leq V_{\boldsymbol{q}} \exp\left[-\frac{T\gamma^2 \epsilon}{28} \right] .$$

Lemma A.2. *[Theorem 2.1 from Gillman (1998)] Consider a finite-state, irreducible, aperiodic and reversible Markov chain with state space* S*, matrix of transition probabilities* P*, and an initial distribution* $\boldsymbol{q}$*. Let* $V_{\boldsymbol{q}} = \left\| (\frac{q_x}{\pi_x}, x \in S) \right\|_2$*. Let* $\tilde{\epsilon} = 1 - \lambda_2$*,*

where λ_2 is the second largest eigenvalue of the matrix P. $\tilde{\epsilon}$ will be referred to as the eigenvalue gap of the transition probability matrix. Let $A \subset S$. Let $N_A(T)$ be the number of times that states in the set A are visited up to time T. Then for any $\gamma \geq 0$, we have

$$P(N_A(T) - T\pi_A \geq \gamma) \leq (1 + \frac{\gamma\epsilon}{10T})V_{\boldsymbol{q}}e^{-\gamma^2\epsilon/20T},$$

where

$$\pi_A = \sum_{x \in A} \pi_x.$$

The lemma below relates the number of observations of a particular state of a Markov chain with its stationary probability.

Lemma A.3. *[Lemma 2.1 from Anantharam et al. (1987b)] Let Y be an irreducible aperiodic Markov chain with a state space S, transition probability matrix P, an initial distribution that is non-zero in all states, and a stationary distribution $\{\pi_x\}, \forall x \in S$. Let F_t be the σ-field generated by random variables $X_1, X_2, ..., X_t$ where X_t corresponds to the state of the chain at time t. Let G be a σ-field independent of $F = \vee_{t \geq 1} F_t$, the smallest σ-field containing $F_1, F_2, ...$ Let τ be a stopping time with respect to the increasing family of σ-fields $\{G \vee F_t, t \geq 1\}$. Define $U(x, \tau)$ such that*

$$U(x, \tau) = \sum_{t=1}^{\tau} I(X_t = x).$$

Then $\forall \tau$ such that $E[\tau] < \infty$, we have

$$|E[U(x, \tau)] - \pi_x E[\tau]| \leq C_P,$$

where C_P is a constant that depends on P.

The next lemma says that for a positive recurrent Markov chain, the conditional expected number of visits to a state before a stopping time is equal to the conditional expectation of the stopping time multiplied by the stationary probability of that state.

Lemma A.4. *If $\{X_n\}_{n\geq 0}$ is a positive recurrent homogeneous Markov chain with state space S, stationary distribution π, and τ is a stopping time that is finite almost surely for which $X_\tau = x$ then for all $y \in S$*

$$E\left[\sum_{t=0}^{\tau-1} I(X_t = y)|X_0 = x\right] = E[\tau|X_0 = x]\pi_y \ .$$

Next, we define a uniformly ergodic Markov chain, and give a large deviation bound for a perturbation of that uniformly ergodic chain. The norm used in the definition and lemma below is the total variation norm. For finite and countable vectors this corresponds to l_1 norm, and the induced matrix norm corresponds to maximum absolute row sum norm.

Definition A.5. *Mitrophanov* (2005) A Markov chain $X = \{X_t, t \in \{1, 2, \ldots\}\}$ on a measurable space $(\mathcal{S}, \mathcal{B})$, with transition kernel $P(x, \mathcal{G})$, $x \in \mathcal{S}$, $\mathcal{G} \in \mathcal{B}$ is uniformly ergodic if there exists constants $\rho < 1, C < \infty$ such that for all $x \in \mathcal{S}$,

$$\left\|e_x P^t - \pi\right\| \leq C\rho^t, \ t \in \{1, 2, \ldots\} \ . \tag{A.1}$$

Clearly, for a finite state Markov chain uniform ergodicity is equivalent to ergodicity.

Lemma A.6. *(Mitrophanov (2005) Theorem 3.1.) Let $X = \{X_t, t \in \{1, 2, \ldots\}\}$ be a uniformly ergodic Markov chain for which (A.1) holds. Let $\hat{X} = \{\hat{X}_t, t \in \{1, 2, \ldots\}\}$ be the perturbed chain with transition kernel $\hat{P}$. Given the two chains have the same*

initial distribution, let $\psi_t, \hat{\psi}_t$ *be the distribution of* $X, \hat{X}$ *at time* t *respectively. Then,*

$$\left\|\psi_t - \hat{\psi}_t\right\| \leq C_1(P,t) \left\|\hat{P} - P\right\|, \tag{A.2}$$

where $C_1(P,t) = \left(\hat{t} + C\frac{\rho^{\hat{t}} - \rho^t}{1-\rho}\right)$ *and* $\hat{t} = \left\lceil \log_\rho C^{-1} \right\rceil$.

Another large deviation bound is the Chernoff-Hoeffding bound which bounds the difference between the sample mean and expected reward for distributions with bounded support.

Lemma A.7. *(Chernoff-Hoeffding Bound) Let* $X_1, \ldots, X_T$ *be random variables with common range [0,1], such that* $E[X_t | X_{t-1}, \ldots, X_1] = \mu$. *Let* $S_T = X_1 + \ldots + X_T$. *Then for all* $\epsilon \geq 0$

$$P(|S_T - T\mu| \geq \epsilon) \leq 2e^{\frac{-2\epsilon^2}{T}}.$$

Finally, we state a large deviation bound for independent Bernoulli random variables.

Lemma A.8. *Let* $X_i, i = 1, 2, \ldots$ *be a sequence of independent Bernoulli random variables such that* X_i *has mean* q_i *with* $0 \leq q_i \leq 1$. *Let* $\bar{X}_t = \frac{1}{t}\sum_{i=1}^t X_i$, $\bar{q}_t = \frac{1}{t}\sum_{i=1}^t q_i$. *Then for any constant* $\epsilon \geq 0$ *and any integer* $T \geq 0$,

$$P\left(\bar{X}_T - \bar{q}_T \leq -\epsilon\right) \leq e^{-2T\epsilon^2}.$$

Proof. The result follows from symmetry and *D.W. Turner* (1995). □

APPENDIX B

Proof of Lemma II.2

We first state and prove the following lemma which will be used to prove Lemma II.2.

Lemma B.1. *Let* $g_{t,s}^k = \bar{r}^k(s) + c_{t,s}$, $c_{t,s} = \sqrt{L \ln t/s}$. *Under UCB with constant* $L \geq 112 S_{\max}^2 r_{\max}^2 \hat{\pi}_{\max}^2 / \epsilon_{\min}$, *for any suboptimal arm* k *we have*

$$E\left[\sum_{t=1}^{T}\sum_{w=1}^{t-1}\sum_{w_k=l}^{t-1} I(g_{t,w}^1 \leq g_{t,w_k}^k)\right] \leq \frac{|S^k| + |S^1|}{\pi_{\min}}\beta, \tag{B.1}$$

where $l = \left\lceil \frac{4L \ln T}{(\mu^1 - \mu^k)^2} \right\rceil$ *and* $\beta = \sum_{t=1}^{\infty} t^{-2}$.

Proof. First, we show that for any suboptimal arm k we have that $g_{t,w}^1 \leq g_{t,w_k}^k$ implies at least one of the following holds:

$$\bar{r}^1(w) \leq \mu^1 - c_{t,w} \tag{B.2}$$

$$\bar{r}^k(w_k) \geq \mu^k + c_{t,w_k} \tag{B.3}$$

$$\mu^1 < \mu^k + 2c_{t,w_k}. \tag{B.4}$$

This is because if none of the above holds, then we must have

$$g^1_{t,w} = \bar{r}^1(w) + c_{t,w} > \mu^1 \geq \mu^k + 2c_{t,w_k} > \bar{r}^k(w_k) + c_{t,w_k} = g^k_{t,w_k},$$

which contradicts $g^1_{t,w} \leq g^k_{t,w_k}$.

If we choose $w_k \geq 4L \ln T/(\mu^1 - \mu^k)^2$, then

$$2c_{t,w_k} = 2\sqrt{\frac{L \ln t}{w_k}} \leq 2\sqrt{\frac{L \ln t(\mu^1 - \mu^k)^2}{4L \ln T}} \leq \mu^1 - \mu^i$$

for $t \leq T$, which means (B.4) is false, and therefore at least one of (B.2) and (B.3) is true with this choice of w_k. Let $l = \left\lceil \frac{4L \ln T}{(\mu^1 - \mu^k)^2} \right\rceil$. Then we have,

$$\begin{aligned}
&E\left[\sum_{t=1}^{T}\sum_{w=1}^{t-1}\sum_{w_k=l}^{t-1} I(g^1_{t,w} \leq g^k_{t,w_k})\right] \\
&\leq \sum_{t=1}^{T}\sum_{w=1}^{t-1}\sum_{w_k=\left\lceil \frac{4L \ln T}{(\mu^1-\mu^k)^2} \right\rceil}^{t-1} \left(P(\bar{r}^1(w) \leq \mu^1 - c_{t,w}) + P(\bar{r}^k(w_k) \geq \mu^k + c_{t,w_k})\right) \\
&\leq \sum_{t=1}^{\infty}\sum_{w=1}^{t-1}\sum_{w_k=\left\lceil \frac{4L \ln T}{(\mu^1-\mu^k)^2} \right\rceil}^{t-1} \left(P(\bar{r}^1(w) \leq \mu^1 - c_{t,w}) + P(\bar{r}^k(w_k) \geq \mu^k + c_{t,w_k})\right) \ .
\end{aligned}$$

Consider an initial distribution $\mathbf{q}^k$ for the kth arm. We have:

$$V_{\mathbf{q}^k} = \left\| \left(\frac{q^k_y}{\pi^k_y}, y \in S^k \right) \right\|_2 \leq \sum_{y \in S^k} \left\| \frac{q^k_y}{\pi^k_y} \right\|_2 \leq \frac{1}{\pi_{\min}},$$

where the first inequality follows from the Minkowski inequality. Let $n^k_y(t)$ denote the number of times state y of arm k is observed up to and including the tth play of arm k. For simplicity of presentation assume that the state rewards are positive, i.e., $r^k_x > 0$ for all $x \in S^k$, $k \in \mathcal{K}$. All of the analysis for the rested bandits will also hold when we have $r^k_x = 0$ for some $x \in S^k$. In that case we define $S^k_+ \subset S^k$ as the set of

states of arm k with positive rewards, and perform the same analysis as below.

$$
\begin{aligned}
P(\bar{r}^k(w_k) \geq \mu^k + c_{t,w_k}) &= P\left(\sum_{y \in S^k} r_y^k n_y^k(w_k) \geq w_k \sum_{y \in S^k} r_y^k \pi_y^k + w_k c_{t,w_k}\right) \\
&= P\left(\sum_{y \in S^k} (r_y^k n_y^k(w_k) - w_k r_y^k \pi_y^k) \geq w_k c_{t,w_k}\right) \\
&= P\left(\sum_{y \in S^k} (-r_y^k n_y^k(w_k) + w_k r_y^k \pi_y^k) \leq -w_k c_{t,w_k}\right) \qquad \text{(B.5)}
\end{aligned}
$$

Consider a sample path ω and the events

$$
\begin{aligned}
A &= \left\{\omega : \sum_{y \in S^k} (-r_y^k n_y^k(w_k)(\omega) + w_k r_y^k \pi_y^k) \leq -w_k c_{t,w_k}\right\}, \\
B &= \bigcup_{y \in S^k} \left\{\omega : -r_y^k n_y^k(w_k)(\omega) + w_k r_y^k \pi_y^k \leq -\frac{w_k c_{t,w_k}}{|S^k|}\right\} .
\end{aligned}
$$

If $\omega \notin B$, then

$$
\begin{aligned}
& - r_y^k n_y^k(w_k)(\omega) + w_k r_y^k \pi_y^k > -\frac{w_k c_{t,w_k}}{|S^k|}, \ \forall y \in S^k \\
\Rightarrow & \sum_{y \in S^k} (-r_y^k n_y^k(w_k)(\omega) + w_k r_y^k \pi_y^k) > -w_k c_{t,w_k} .
\end{aligned}
$$

Thus $\omega \notin A$, therefore $P(A) \leq P(B)$. Then continuing from (B.5):

$$
\begin{aligned}
P(\bar{r}^k(w_k) \geq \mu^k + c_{t,w_k}) &\leq \sum_{y \in S^k} P\left(-r_y^k n_y^k(w_k) + w_k r_y^k \pi_y^k \leq -\frac{w_k c_{t,w_k}}{|S^k|}\right) \\
&= \sum_{y \in S^k} P\left(r_y^k n_y^k(w_k) - w_k r_y^k \pi_y^k \geq \frac{w_k c_{t,w_k}}{|S^k|}\right) \\
&= \sum_{y \in S^k} P\left(n_y^k(w_k) - w_k \pi_y^k \geq \frac{w_k c_{t,w_k}}{|S^k| r_y^k}\right) \\
&= \sum_{y \in S^k} P\left(\frac{\sum_{t=1}^{w_k} I(y^k(t) = y) - w_k \pi_y^k}{\hat{\pi}_y^k w_k} \geq \frac{c_{t,w_k}}{|S^k| r_y^k \hat{\pi}_y^k}\right)
\end{aligned}
$$

$$\leq \sum_{y \in S^k} V_{q^k} t^{-\frac{L\epsilon^k}{28(|S^k| r_y^k \hat{\pi}_y^k)^2}} \tag{B.6}$$

$$\leq \frac{|S^k|}{\pi_{\min}} t^{-\frac{L\epsilon_{\min}}{28 S_{\max}^2 r_{\max}^2 \hat{\pi}_{\max}^2}}, \tag{B.7}$$

where (B.6) follows from Lemma A.1 by letting

$$\gamma = \frac{c_{t,w_k}}{|S^k| r_y^k \hat{\pi}_y^k}, \; f(y^k(t)) = \frac{I(y^k(t) = y) - \pi_y^k}{\hat{\pi}_y^k} \; ,$$

and recalling $\hat{\pi}_y^k = \max\{\pi_y^k, 1 - \pi_y^k\}$ (note $\dot{P}^k$ is irreducible). Similarly, we have

$$\begin{aligned}
P\left(\bar{r}^1(w) \leq \mu^1 - c_{t,w}\right) &= P\left(\sum_{y \in S^1} r_y^1 (n_y^1(w) - w\pi_y^1) \leq -w c_{t,w}\right) \\
&\leq \sum_{y \in S^1} P\left(r_y^1 n_y^1(w) - w r_y^1 \pi_y^1 \leq -\frac{w c_{t,w}}{|S^1|}\right) \\
&= \sum_{y \in S^1} P\left(r_y^1 (w - \sum_{x \neq y} n_x^1(w)) - w r_y^1 (1 - \sum_{x \neq y} \pi_x^1) \leq -\frac{w c_{t,w}}{|S^1|}\right) \\
&= \sum_{y \in S^1} P\left(r_y^1 \sum_{x \neq y} n_x^1(w) - w r_y^1 \sum_{x \neq y} \pi_x^1 \geq \frac{w c_{t,w}}{|S^1|}\right) \\
&\leq \sum_{y \in S^1} N_{q^1} t^{-\frac{L\epsilon^1}{28(|S^1| r_y^1 \hat{\pi}_y^1)^2}} \qquad \text{(B.8)} \\
&\leq \frac{|S^1|}{\pi_{\min}} t^{-\frac{L\epsilon_{\min}}{28 S_{\max}^2 r_{\max}^2 \hat{\pi}_{\max}^2}} \qquad \text{(B.9)}
\end{aligned}$$

where (B.8) again follows from Lemma A.1. The result then follows from combining (B.7) and (B.9):

$$\begin{aligned}
E\left[\sum_{t=1}^{T} \sum_{w=1}^{t-1} \sum_{w_k=l}^{t-1} I(g_{t,w}^1 \leq g_{t,w_k}^k)\right] &\leq \frac{|S^k| + |S^1|}{\pi_{\min}} \sum_{t=1}^{\infty} \sum_{w=1}^{t-1} \sum_{w_k=1}^{t-1} t^{-\frac{L\epsilon_{\min}}{28 S_{\max}^2 r_{\max}^2 \hat{\pi}_{\max}^2}} \\
&= \frac{|S^k| + |S^1|}{\pi_{\min}} \sum_{t=1}^{\infty} t^{-\frac{L\epsilon_{\min} - 56 S_{\max}^2 r_{\max}^2 \hat{\pi}_{\max}^2}{28 S_{\max}^2 r_{\max}^2 \hat{\pi}_{\max}^2}} \\
&\leq \frac{|S^k| + |S^1|}{\pi_{\min}} \sum_{t=1}^{\infty} t^{-2}.
\end{aligned}$$

□

Let l be any positive integer and consider a suboptimal arm k. Then,

$$N^k(T) = 1 + \sum_{t=K+1}^{T} I(\alpha(t) = k) \leq l + \sum_{t=K+1}^{T} I(\alpha(t) = k, N^k(t-1) \geq l) \ . \quad \text{(B.10)}$$

Consider the event

$$E = \left\{ g^1_{t,N^1(t)} \leq g^k_{t,N^k(t)} \right\},$$

For a sample path $\omega \in E^C$ we have $\alpha(t) \neq k$. Therefore $\{\omega : \alpha(t) = k\} \subset E$ and

$$\begin{aligned} I(\alpha(t) = k,\ N^k(t-1) \geq l) &\leq I(\omega \in E,\ N^k(t-1) \geq l) \\ &= I(g^1_{t,N^1(t)} \leq g^k_{t,N^k(t)},\ N^k(t-1) \geq l). \end{aligned}$$

Therefore continuing from (B.10),

$$\begin{aligned} N^k(T) &\leq l + \sum_{t=K+1}^{T} I(g^1_{t,N^1(t)} \leq g^k_{t,N^k(t)}, N^k(t-1) \geq l) \\ &\leq l + \sum_{t=K+1}^{T} I\left(\min_{1 \leq w < t} g^1_{t,w} \leq \max_{l \leq w_k < t} g^k_{t,w_k} \right) \\ &\leq l + \sum_{t=K+1}^{T} \sum_{w=1}^{t-1} \sum_{w_k=l}^{t-1} I(g^1_{t,w} \leq g^k_{t,w_k}) \\ &\leq l + \sum_{t=1}^{T} \sum_{w=1}^{t-1} \sum_{w_k=l}^{t-1} I(g^1_{t,w} \leq g^k_{t,w_k}). \end{aligned}$$

Using Lemma B.1 with $l = \left\lceil \frac{4L \ln T}{(\mu^1 - \mu^k)^2} \right\rceil$, we have for any suboptimal arm

$$E[N^k(T)] \leq 1 + \frac{4L \ln T}{(\mu^1 - \mu^k)^2} + \frac{(|S^k| + |S^1|)\beta}{\pi_{\min}}.$$

APPENDIX C

Proof of Lemma III.1

We first state and prove the following lemma which will be used to prove Lemma III.1.

Lemma C.1. *Assume all arms are restless Markovian with irreducible multiplicative symmetrizations. Let* $g^k_{t,w} = \bar{r}^k(w) + c_{t,w}$, $c_{t,w} = \sqrt{L \ln t/w}$. *Under RCA with constant* $L \geq 112 S^2_{\max} r^2_{\max} \hat{\pi}^2_{\max}/\epsilon_{\min}$, *for any suboptimal arm* k *we have*

$$E\left[\sum_{t=1}^{t_2(b)} \sum_{w=1}^{t-1} \sum_{w_i=l}^{t-1} I(g^1_{t,w} \leq g^k_{t,w_k})\right] \leq \frac{|S^k| + |S^1|}{\pi_{\min}} \beta, \tag{C.1}$$

where $l = \left\lceil \frac{4L \ln T}{(\mu^1 - \mu^k)^2} \right\rceil$ *and,* $\beta = \sum_{t=1}^{\infty} t^{-2}$.

Proof. Note that all the quantities in computing the indices in (C.1) comes from the intervals $X^k_2(1), X^k_2(2), \cdots \forall k \in \mathcal{K}$. Since these intervals begin with state γ^k and end with a return to γ^k (but excluding the return visit to γ^k), by the strong Markov property the process at these stopping times have the same distribution as the original process. Moreover by connecting these intervals together we form a continuous sample path which can be viewed as a sample path generated by a Markov chain with a transition matrix identical to the original arm. Therefore we can proceed in exactly

the same way as the proof of Lemma II.2. If we choose $s_k \geq 4L\ln(T)/(\mu^1 - \mu^k)^2$, then for $t \leq t_2(b) = T' \leq T$, and for any suboptimal arm k,

$$2c_{t,s_k} = 2\sqrt{\frac{L\ln(t)}{s_k}} \leq 2\sqrt{\frac{L\ln(t)(\mu^1-\mu^k)^2}{4L\ln(T)}} \leq \mu^1 - \mu^k.$$

The result follows from letting $l = \left\lceil \frac{4L\ln T}{(\mu^1-\mu^k)^2} \right\rceil$ as in the proof of Lemma II.2. □

Let $c_{t,s} = \sqrt{L\ln t/s}$, and let l be any positive integer. Then,

$$\begin{aligned}
B^k(b) &= 1 + \sum_{m=K+1}^{b} I(\tilde{\alpha}(m) = k) \\
&\leq l + \sum_{m=K+1}^{b} I(\tilde{\alpha}(m) = k, B^k(m-1) \geq l) \\
&\leq l + \sum_{m=K+1}^{b} I\left(g^1_{t_2(m-1),N_2^1(t_2(m-1))} \leq g^k_{t_2(m-1),N_2^k(t_2(m-1))}, B^k(m-1) \geq l\right) \\
&\leq l + \sum_{m=K+1}^{b} I\left(\min_{1\leq w\leq t_2(m-1)} g^1_{t_2(m-1),w} \leq \max_{t_2(l)\leq w_k\leq t_2(m-1)} g^k_{t_2(m-1),w_k}\right) \\
&\leq l + \sum_{m=K+1}^{b} \sum_{w=1}^{t_2(m-1)} \sum_{w_k=t_2(l)}^{t_2(m-1)} I(g^1_{t_2(m),w} \leq g^k_{t_2(m),w_k}) \qquad \text{(C.2)} \\
&\leq l + \sum_{t=1}^{t_2(b)} \sum_{w=1}^{t-1} \sum_{w_k=l}^{t-1} I(g^1_{t,w} \leq g^k_{t,w_k}) \qquad \text{(C.3)}
\end{aligned}$$

where as given in (3.1), $g^k_{t,w} = \bar{r}^k(w) + c_{t,w}$. The inequality in (C.3) follows from the fact that the outer sum in (C.3) is over time while the outer sum in (C.2) is over blocks and each block lasts at least two time slots.

From this point on we use Lemma C.1 to get

$$E[B^i(b(T))|b(T) = b] \leq \left\lceil \frac{4L\ln t_2(b)}{(\mu^1-\mu^k)^2} \right\rceil + \frac{(|S^k| + |S^1|)\beta}{\pi_{\min}},$$

for all suboptimal arms. Therefore,

$$E[B^k(b(T))] \leq \frac{4L \ln T}{(\mu^1 - \mu^k)^2} + C_{k,1}, \tag{C.4}$$

since $T \geq t_2(b(T))$ almost surely.

The total number of plays of arm k at the end of block $b(T)$ is equal to the total number of plays of arm k during the regenerative cycles of visiting state γ^k plus the total number of plays before entering the regenerative cycles plus one more play resulting from the last play of the block which is state γ^k. This gives:

$$E[N^k(N(T))] \leq \left(\frac{1}{\pi^k_{\min}} + \Omega^k_{\max} + 1 \right) E[B^k(b(T))] \ .$$

APPENDIX D

Proof of Theorem III.2

Assume that the states which determine the regenerative sample paths are given *a priori* by $\gamma = [\gamma^1, \cdots, \gamma^K]$. We denote the expectations with respect to RCA given γ as E_γ. First we rewrite the regret in the following form:

$$\begin{aligned} R_\gamma(T) &= \mu^1 E_\gamma[N(T)] - E_\gamma\left[\sum_{t=1}^{N(T)} r^{\alpha(t)}(t)\right] \\ &\quad + \mu^1 E_\gamma[T - N(T)] - E_\gamma\left[\sum_{t=N(T)+1}^{T} r^{\alpha(t)}(t)\right] \\ &= \left\{\mu^1 E_\gamma[N(T)] - \sum_{k=1}^{K} \mu^k E_\gamma\left[N^k(N(T))\right]\right\} - Z_\gamma(T) \\ &\quad + \mu^1 E_\gamma[T - N(T)] - E_\gamma\left[\sum_{t=N(T)+1}^{T} r^{\alpha(t)}(t)\right], \end{aligned} \tag{D.1}$$

where for notational convenience, we have used

$$Z_\gamma(T) = E_\gamma\left[\sum_{t=1}^{N(T)} r^{\alpha(t)}(t)\right] - \sum_{k=1}^{K} \mu^k E_\gamma\left[N^k(N(T))\right].$$

We can bound the first difference in (D.1) logarithmically using Lemma III.1, so it

remains to bound $Z_\gamma(T)$ and the last difference. We have

$$\begin{aligned} Z_\gamma(T) \geq & \sum_{y \in S^1} r_y^1 E_\gamma \left[\sum_{j=1}^{B^1(b(T))} \sum_{r^1(t) \in X^1(j)} I(r^1(t) = y) \right] \\ & + \sum_{k:\mu^k < \mu^1} \sum_{y \in S^k} r_y^k E_\gamma \left[\sum_{j=1}^{B^k(b(T))} \sum_{r^k(t) \in X_2^k(j)} I(r^k(t) = y) \right] \\ & - \mu^1 E_\gamma \left[N^1(N(T)) \right] - \sum_{k>1} \mu^k \left(\frac{1}{\pi_{\gamma^k}^k} + \Omega_{\max}^k + 1 \right) E_\gamma \left[B^k(b(T)) \right] \ , \end{aligned} \tag{D.2}$$

where the inequality comes from counting only the rewards obtained during the SB2s for all suboptimal arms. Applying Lemma A.4 to (D.2) we get

$$E_\gamma \left[\sum_{j=1}^{B^k(b(T))} \sum_{r^k(t) \in X_2^k(j)} I(r^k(t) = y) \right] = \frac{\pi_y^k}{\pi_{\gamma^k}^k} E_\gamma \left[B^k(b(T)) \right] \ .$$

Rearrange terms and noting $\mu^1 = \sum_y r_y^1 \pi_y^1$,

$$Z_\gamma(T) \geq R^1(T) - \sum_{k:\mu^k < \mu^1} \mu^k (\Omega_{\max}^k + 1) E_\gamma \left[B^k(b(T)) \right] \tag{D.3}$$

where

$$R^1(T) = \sum_{y \in S^1} r_y^1 E_\gamma \left[\sum_{j=1}^{B^1(b(T))} \sum_{r^1(t) \in X^1(j)} I(r^1(t) = y) \right] - \sum_{y \in S^1} r_y^1 \pi_y^1 E_\gamma \left[N^1(N(T)) \right].$$

Consider now $R^1(T)$. Since all suboptimal arms are played at most logarithmically, the number of time steps in which the best arm is not played is at most logarithmic. It follows that the number of discontinuities between plays of the best arm is at most logarithmic. Suppose we combine successive blocks in which the best arm is played, and denote by $\bar{X}^1(j)$ the jth combined block. Let $\bar{b}^1$ denote the total number of combined blocks up to block b. Each $\bar{X}^1$ thus consists of two sub-blocks: $\bar{X}_1^1$ that contains the states visited from beginning of $\bar{X}^1$ (empty if the first state is γ^1) to

the state right before hitting γ^1, and sub-block $\bar{X}_2^1$ that contains the rest of $\bar{X}^1$ (a random number of regenerative cycles).

Since a block $\bar{X}^1$ starts after discontinuity in playing the best arm, $\bar{b}^1(T)$ is less than or equal to total number of completed blocks in which the best arm is not played up to time T. Thus

$$E_\gamma[\bar{b}^1(T)] \leq \sum_{k>1} E_\gamma[B^k(b(T))]. \tag{D.4}$$

We rewrite $R^1(T)$ in the following from:

$$\begin{aligned}
R^1(T) = &\sum_{y\in S^1} r_y^1 E_\gamma \left[\sum_{j=1}^{\bar{b}^1(T)} \sum_{r^1(t)\in \bar{X}_2^1(j)} I(r^1(t)=y) \right] && \text{(D.5)}\\
&- \sum_{y\in S^1} r_y^1 \pi_y^1 E_\gamma \left[\sum_{j=1}^{\bar{b}^1(T)} |\bar{X}_2^1(j)| \right] && \text{(D.6)}\\
&+ \sum_{y\in S^1} r_y^1 E_\gamma \left[\sum_{j=1}^{\bar{b}^1(T)} \sum_{r^1(t)\in \bar{X}_1^1(j)} I(r^1(t)=y) \right] && \text{(D.7)}\\
&- \sum_{y\in S^1} r_y^1 \pi_y^1 E_\gamma \left[\sum_{j=1}^{\bar{b}^1(T)} |\bar{X}_1^1(j)| \right] && \text{(D.8)}\\
&> 0 - \mu^1 \Omega_{\max}^1 \sum_{k>1} E_\gamma[B^k(b(T))],
\end{aligned}$$

where the last inequality is obtained by noting the difference between (D.5) and (D.6) is zero by Lemma A.4, using non-negativity of the rewards to lower bound (D.7) by 0, and (D.4) to upper bound (D.8). Combine this with (C.4) and (D.3) we can thus obtain a logarithmic upper bound on $-Z_\gamma(T)$. Finally, we have

$$\mu^1 E_\gamma[T - N(T)] - E_\gamma \left[\sum_{t=N(T)+1}^{T} r^{\alpha(t)}(t) \right] \leq \mu^1 \left(\frac{1}{\pi_{\min}} + \max_{k\in\mathcal{K}} \Omega_{\max}^k + 1 \right) .$$

Therefore we have obtained the stated logarithmic bound for (D.1). Note that this

bound does not depend on γ, and therefore is also an upper bound for $R(T)$, completing the proof.

APPENDIX E

Proof of Theorem III.3

Recall that our description of multiple-plays is in the equivalent form of multiple coordinated agents each with a single play. A list of notations used in the proof (in addition to the ones in Tables 2.1 and 3.2) is given below:

- $N^{k,i}(t)$: the total number of times (slots) arm k is played by agent i up to the last completed block of arm k up to time t.
- $O(b)$: the set of arms that are *free* to be selected by some agent i upon its completion of the bth block; these are arms that are currently not being played by other agents (during time slot $t(b)$), and the arms whose blocks are completed at time $t(b)$.

Before proving Theorem III.3, we state the following lemmas which will be used in the proof.

Lemma E.1. *Let* $g_{t,w}^k = \bar{r}^k(w) + c_{t,w}$, $c_{t,w} = \sqrt{L \ln t/w}$. *Under RCA-M with constant* $L \geq 112 S_{\max}^2 r_{\max}^2 \hat{\pi}_{\max}^2 / \epsilon_{\min}$, *for any suboptimal arm* k *and optimal arm* j *we have*

$$E\left[\sum_{t=1}^{t_2(b)} \sum_{w=1}^{t-1} \sum_{w_k=l}^{t-1} I(g_{t,w}^j \leq g_{t,w_k}^k)\right] \leq \frac{|S^k| + |S^j|}{\pi_{\min}} \beta,$$

where $l = \left\lceil \frac{4L \ln T}{(\mu^M - \mu^k)^2} \right\rceil$ *and,* $\beta = \sum_{t=1}^{\infty} t^{-2}$.

Proof. Result is obtained by following steps similar to the proof of Lemma C.1. □

Lemma E.2. *For RCA-M run with a constant* $L \geq 112 S_{\max}^2 r_{\max}^2 \hat{\pi}_{\max}^2 / \epsilon_{\min}$*, we have*

$$\sum_{k>M} (\mu^1 - \mu^k) E[N^k(T)] \leq 4L \sum_{k>M} \frac{(\mu^1 - \mu^k) D_k \ln T}{(\mu^M - \mu^k)^2} + \sum_{k>M} (\mu^1 - \mu^k) D_k \left(1 + M \sum_{j=1}^{M} C_{k,j}\right),$$

where

$$C_{k,j} = \frac{(|S^k| + |S^j|)\beta}{\pi_{\min}},\ \beta = \sum_{t=1}^{\infty} t^{-2}, D_k = \left(\frac{1}{\pi_{\min}^k} + \Omega_{\max}^k + 1\right).$$

Proof. Let $c_{t,w} = \sqrt{L \ln t / w}$, and let l be any positive integer. Then,

$$B^k(b) = 1 + \sum_{m=K+1}^{b} I(\tilde{\alpha}(m) = k) \leq l + \sum_{m=K+1}^{b} I(\tilde{\alpha}(m) = k, B^k(m-1) \geq l) \quad \text{(E.1)}$$

Consider any sample path ω and the following sets

$$E = \bigcup_{j=1}^{M} \left\{\omega : g^j_{t_2(m-1), N_2^j(t_2(m-1))}(\omega) \leq g^k_{t_2(m-1), N_2^k(t_2(m-1))}(\omega)\right\},$$

and

$$E^C = \bigcap_{j=1}^{M} \left\{\omega : g^j_{t_2(m-1), N_2^j(t_2(m-1))}(\omega) > g^k_{t_2(m-1), N_2^k(t_2(m-1))}(\omega)\right\}.$$

If $\omega \in E^C$ then $\tilde{\alpha}(m) \neq k$. Therefore $\{\omega : \tilde{\alpha}(m)(\omega) = k\} \subset E$ and

$$\begin{aligned} & I(\tilde{\alpha}(m) = k, B^k(m-1) \geq l) \\ & \leq I(\omega \in E, B^k(m-1) \geq l) \end{aligned}$$

$$\leq \sum_{j=1}^{M} I\left(g^j_{t_2(m-1),N_2^j(t_2(m-1))} \leq g^k_{t_2(m-1),N_2^k(t_2(m-1))}, B^k(m-1) \geq l\right) \ .$$

Therefore continuing from (E.1),

$$\begin{aligned}
B^k(b) &\leq l + \sum_{j=1}^{M} \sum_{m=K+1}^{b} I\left(g^j_{t_2(m-1),N_2^j(t_2(m-1))} \leq g^k_{t_2(m-1),N_2^k(t_2(m-1))}, B^k(m-1) \geq l\right) \\
&\leq l + \sum_{j=1}^{M} \sum_{m=K+1}^{b} I\left(\min_{1\leq w\leq t_2(m-1)} g^j_{t_2(m-1),w} \leq \max_{t_2(l)\leq w_k\leq t_2(m-1)} g^k_{t_2(m-1),w_k}\right) \\
&\leq l + \sum_{j=1}^{M} \sum_{m=K+1}^{b} \sum_{w=1}^{t_2(m-1)} \sum_{w_k=t_2(l)}^{t_2(m-1)} I(g^j_{t_2(m),w} \leq g^k_{t_2(m),w_k}) \qquad \text{(E.2)} \\
&\leq l + M \sum_{j=1}^{M} \sum_{t=1}^{t_2(b)} \sum_{w=1}^{t-1} \sum_{w_k=l}^{t-1} I(g^j_{t,w} \leq g^k_{t,w_k}) \ , \qquad \text{(E.3)}
\end{aligned}$$

where $g^k_{t,w} = \bar{r}^k(w) + c_{t,w}$, and we have assumed that the index value of an arm remains the same between two updates. The inequality in (E.3) follows from the facts that the second outer sum in (E.3) is over time while the second outer sum in (E.2) is over blocks, each block lasts at least two time slots and at most M blocks can be completed in each time step. From this point on we use Lemma E.1 to get

$$E[B^k(b(T))|b(T)=b] \leq \left\lceil \frac{4L\ln t_2(b)}{(\mu^M - \mu^k)^2} \right\rceil + M\sum_{j=1}^{M} \frac{(|S^k| + |S^j|)\beta}{\pi_{\min}},$$

for all suboptimal arms. Therefore,

$$E[B^k(b(T))] \leq \frac{4L\ln T}{(\mu^M - \mu^k)^2} + 1 + M\sum_{j=1}^{M} C_{k,j}, \qquad \text{(E.4)}$$

since $T \geq t_2(b(T))$ almost surely.

The total number of plays of arm k at the end of block $b(T)$ is equal to the total number of plays of arm k during the regenerative cycles of visiting state γ^k plus the total number of plays before entering the regenerative cycles plus one more play

resulting from the last play of the block which is state γ^k. This gives:

$$E[N^k(N(T))] \leq \left(\frac{1}{\pi^k_{\min}} + \Omega^k_{\max} + 1\right) E[B^k(b(T))] \ .$$

Thus,

$$\begin{aligned}
&\sum_{k>M} (\mu^1 - \mu^k) E[N^k(N(T))] \\
&\leq 4L \sum_{k>M} \frac{(\mu^1 - \mu^k) D_k \ln T}{(\mu^M - \mu^k)^2} + \sum_{k>M} (\mu^1 - \mu^k) D_k \left(1 + M \sum_{j=1}^{M} C_{k,j}\right). \qquad \text{(E.5)}
\end{aligned}$$

□

Now we give the proof of Theorem III.3. Assume that the states which determine the regenerative sample paths are given *a priori* by $\gamma = [\gamma^1, \cdots, \gamma^K]$. This is to simplify the analysis by skipping the initialization stage of the algorithm and we will show that this choice does not affect the regret bound. We denote the expectations with respect to RCA-M given γ as E_γ. First we rewrite the regret in the following form:

$$\begin{aligned}
R_\gamma(T) &= \sum_{j=1}^{M} \mu^j E_\gamma[N(T)] - E_\gamma \left[\sum_{t=1}^{N(T)} \sum_{k \in \boldsymbol{\alpha(t)}} r^k(t)\right] \\
&\quad + \sum_{j=1}^{M} \mu^j E_\gamma[T - N(T)] - E_\gamma \left[\sum_{t=N(T)+1}^{T} \sum_{k \in \boldsymbol{\alpha}(t)} r^k(t)\right] \\
&= \left\{\sum_{j=1}^{M} \mu^j E_\gamma[N(T)] - \sum_{k=1}^{K} \mu^k E_\gamma\left[N^k(N(T))\right]\right\} - Z_\gamma(T) \qquad \text{(E.6)} \\
&\quad + \sum_{j=1}^{M} \mu^j E_\gamma[T - N(T)] - E_\gamma \left[\sum_{t=N(T)+1}^{T} \sum_{k \in \boldsymbol{\alpha}(t)} r^k(t)\right] \qquad \text{(E.7)}
\end{aligned}$$

where for notational convenience, we have used

$$Z_\gamma(T) = E_\gamma \left[\sum_{t=1}^{N(T)} \sum_{k \in \boldsymbol{\alpha}(t)} r^k(t) \right] - \sum_{k=1}^{K} \mu^k E_\gamma \left[N^k(N(T)) \right].$$

We have

$$\begin{aligned}
&\sum_{j=1}^{M} \mu^j E_\gamma[N(T)] - \sum_{k=1}^{K} \mu^k E_\gamma \left[N^k(N(T)) \right] \\
&= \sum_{j=1}^{M} \sum_{k=1}^{K} \mu^j E_\gamma[N^{k,j}(N(T))] - \sum_{j=1}^{M} \sum_{k=1}^{K} \mu^k E_\gamma[N^{k,j}(N(T))] \\
&= \sum_{j=1}^{M} \sum_{k>M} (\mu^j - \mu^k) E_\gamma[N^{k,j}(N(T))] \\
&\leq \sum_{k>M} (\mu^1 - \mu^k) E_\gamma[N^k(N(T))]. \qquad \text{(E.8)}
\end{aligned}$$

Since we can bound (E.8), i.e. the difference in the brackets in (E.6) logarithmically using Lemma E.2, it remains to bound $Z_\gamma(T)$ and the difference in (E.7). We have

$$\begin{aligned}
Z_\gamma(T) \geq &\sum_{k=1}^{M} \sum_{y \in S^k} r_y^k E_\gamma \left[\sum_{b=1}^{B^k(b(T))} \sum_{r^k(t) \in X^k(b)} I(r^k(t) = y) \right] \\
&+ \sum_{k>M} \sum_{y \in S^k} r_y^k E_\gamma \left[\sum_{b=1}^{B^k(b(T))} \sum_{r^k(t) \in X_2^k(b)} I(r^k(t) = y) \right] \qquad \text{(E.9)} \\
&- \sum_{k=1}^{M} \mu^k E_\gamma \left[N^k(N(T)) \right] - \sum_{k>M} \mu^k \left(\frac{1}{\pi_{\gamma^k}^k} + \Omega_{\max}^k + 1 \right) E_\gamma \left[B^k(b(T)) \right] ,
\end{aligned}$$

where the inequality comes from counting only the rewards obtained during the SB2's for all suboptimal arms and the last part of the proof of Lemma E.2. Applying Lemma

A.4 to (E.9) we get

$$E_\gamma\left[\sum_{b=1}^{B^k(b(T))}\sum_{r^k(t)\in X_2^k(b)} I(r^k(t)=y)\right]=\frac{\pi_y^k}{\pi_{\gamma^k}^k}E_\gamma\left[B^k(b(T))\right]\ .$$

Rearranging terms we get

$$Z_\gamma(T)\geq R^*(T)-\sum_{k>M}\mu^k(\Omega_{\max}^k+1)E_\gamma\left[B^k(b(T))\right], \tag{E.10}$$

where

$$\begin{aligned}R^*(T)=&\sum_{k=1}^{M}\sum_{y\in S^k} r_y^k E_\gamma\left[\sum_{b=1}^{B^k(b(T))}\sum_{r^k(t)\in X^k(b)} I(r^k(t)=y)\right]\\&-\sum_{k=1}^{M}\sum_{y\in S^k} r_y^k\pi_y^k E_\gamma\left[N^k(N(T))\right].\end{aligned}$$

Consider now $R^*(T)$. Since all suboptimal arms are played at most logarithmically, the total number of time slots in which an optimal arm is not played is at most logarithmic. It follows that the number of discontinuities between plays of any single optimal arm is at most logarithmic. For any optimal arm $k\in\{1,\cdots,M\}$ we combine *consecutive* blocks in which arm k is played into a single *combined* block, and denote by $\bar{X}^k(j)$ the jth combined block of arm k. Let $\bar{b}^k$ denote the total number of combined blocks for arm k up to block b. Each $\bar{X}^k$ thus consists of two sub-blocks: $\bar{X}_1^k$ that contains the states visited from the beginning of $\bar{X}^k$ (empty if the first state is γ^k) to the state right before hitting γ^k, and sub-block $\bar{X}_2^k$ that contains the rest of $\bar{X}^k$ (a random number of regenerative cycles).

Since a combined block $\bar{X}^k$ necessarily starts after certain discontinuity in playing the kth best arm, $\bar{b}^k(T)$ is less than or equal to the total number of discontinuities of play of the kth best arm up to time T. At the same time, the total number of discontinuities of play of the kth best arm up to time T is less than or equal to the

total number of blocks in which suboptimal arms are played up to time T. Thus

$$E_\gamma[\bar{b}^k(T)] \leq \sum_{j>M} E_\gamma[B^j(b(T))]. \tag{E.11}$$

We now rewrite $R^*(T)$ in the following from:

$$R^*(T) = \sum_{k=1}^{M} \sum_{y \in S^k} r_y^k E_\gamma \left[\sum_{b=1}^{\bar{b}^k(T)} \sum_{r^k(t) \in \bar{X}_2^k(b)} I(r^k(t) = y) \right] \tag{E.12}$$

$$- \sum_{k=1}^{M} \sum_{y \in S^k} r_y^k \pi_y^k E_\gamma \left[\sum_{b=1}^{\bar{b}^k(T)} |\bar{X}_2^k(b)| \right] \tag{E.13}$$

$$+ \sum_{k=1}^{M} \sum_{y \in S^k} r_y^k E_\gamma \left[\sum_{b=1}^{\bar{b}^k(T)} \sum_{r^k(t) \in \bar{X}_1^k(b)} I(r^k(t) = y) \right] \tag{E.14}$$

$$- \sum_{k=1}^{M} \sum_{y \in S^k} r_y^k \pi_y^k E_\gamma \left[\sum_{b=1}^{\bar{b}^k(T)} |\bar{X}_1^k(b)| \right] \tag{E.15}$$

$$> 0 - \sum_{k=1}^{M} \mu^k \Omega_{\max}^k \sum_{j>M} E_\gamma[B^j(b(T))], \tag{E.16}$$

where the last inequality is obtained by noting the difference between (E.12) and (E.13) is zero by Lemma A.4, using non-negativity of the rewards to lower bound (E.14) by 0, and (E.11) to upper bound (E.15). Combining this with (E.4) and (E.10) we can obtain a logarithmic upper bound on $-Z_\gamma(T)$ by the following steps:

$$\begin{aligned}
&- Z_\gamma(T) \leq -R^*(T) + \sum_{k>M} \mu^k (\Omega_{\max}^k + 1) E_\gamma \left[B^k(b(T)) \right] \\
&\leq \sum_{k=1}^{M} \mu^k \Omega_{\max}^k \sum_{j>M} \left(\frac{4L \ln T}{(\mu^M - \mu^j)^2} + 1 + M \sum_{l=1}^{M} C_{j,l} \beta \right) \\
&+ \sum_{k>M} \mu^k (\Omega_{\max}^k + 1) \left(\frac{4L \ln T}{(\mu^M - \mu^k)^2} + 1 + M \sum_{j=1}^{M} C_{j,k} \beta \right).
\end{aligned}$$

We also have,

$$\sum_{j=1}^{M} \mu^j E_\gamma[T - N(T)] - E_\gamma \left[\sum_{t=N(T)+1}^{T} \sum_{k \in \boldsymbol{\alpha}(t)} r^k(t) \right]$$
$$\leq \sum_{j=1}^{M} \mu^j E_\gamma[T - N(T)] = \sum_{j=1}^{M} \mu^j \left(\frac{1}{\pi_{\min}} + \max_{k \in \mathcal{K}} \Omega^k_{\max} + 1 \right). \tag{E.17}$$

Finally, combining the above results as well as Lemma E.2 we get

$$\begin{aligned}
R_\gamma(T) &= \left\{ \sum_{j=1}^{M} \mu^j E_\gamma[N(T)] - \sum_{k=1}^{K} \mu^k E_\gamma \left[N^k(N(T)) \right] \right\} - Z_\gamma(T) \\
&\quad + \sum_{j=1}^{M} \mu^j E_\gamma[T - N(T)] - E_\gamma \left[\sum_{t=N(T)+1}^{T} \sum_{k \in \boldsymbol{\alpha}(t)} r^k(t) \right] \\
&\leq \sum_{k>M} (\mu^1 - \mu^k) E_\gamma[N^k(N(T))] \\
&\quad + \sum_{k=1}^{M} \mu^k \Omega^k_{\max} \sum_{l>M} \left(\frac{4L \ln T}{(\mu^M - \mu^l)^2} + 1 + M \sum_{j=1}^{M} C_{l,j} \beta \right) \\
&\quad + \sum_{k>M} \mu^k (\Omega^k_{\max} + 1) \left(\frac{4L \ln T}{(\mu^M - \mu^k)^2} + 1 + M \sum_{j=1}^{M} C_{j,k} \beta \right) \\
&\quad + \sum_{j=1}^{M} \mu^j \left(\frac{1}{\pi_{\min}} + \max_{k \in \mathcal{K}} \Omega^k_{\max} + 1 \right) \\
&= 4L \ln T \sum_{k>M} \frac{1}{(\mu^M - \mu^k)^2} \left((\mu^1 - \mu^k) D_k + E_k \right) \\
&\quad + \sum_{k>M} \left((\mu^1 - \mu^k) D_k + E_k \right) \left(1 + M \sum_{j=1}^{M} C_{k,j} \right) + F.
\end{aligned}$$

Therefore we have obtained the stated logarithmic bound for (E.6). Note that this bound does not depend on γ, and thus is also an upper bound for $R(T)$, completing the proof.

APPENDIX F

Proof of Lemma IV.13

From symmetry we have

$$P\left(|\hat{p}_{ij,t}^k - p_{ij}^k| > \epsilon, \mathcal{E}_t\right) = P\left(\hat{p}_{ij,t}^k - p_{ij}^k > \epsilon, \mathcal{E}_t\right) + P\left(\hat{p}_{ij,t}^k - p_{ij}^k < -\epsilon, \mathcal{E}_t\right)$$
$$= 2P\left(\hat{p}_{ij,t}^k - p_{ij}^k > \epsilon, \mathcal{E}_t\right). \tag{F.1}$$

Then

$$\begin{aligned} P\left(\hat{p}_{ij,t}^k - p_{ij}^k > \epsilon, \mathcal{E}_t\right) &= P\left(\frac{\bar{p}_{ij}^k}{\sum_{l\in S^k}\bar{p}_{il}^k} - p_{ij}^k > \epsilon, \mathcal{E}_t\right) \\ &= P\left(\frac{\bar{p}_{ij}^k}{\sum_{l\in S^k}\bar{p}_{il}^k} - p_{ij}^k > \epsilon, \left|\sum_{l\in S^k}\bar{p}_{il}^k - 1\right| < \delta, \mathcal{E}_t\right) \\ &+ P\left(\frac{\bar{p}_{ij}^k}{\sum_{l\in S^k}\bar{p}_{il}^k} - p_{ij}^k > \epsilon, \left|\sum_{l\in S^k}\bar{p}_{il}^k - 1\right| \geq \delta, \mathcal{E}_t\right) \\ &\leq P\left(\frac{\bar{p}_{ij}^k}{1-\delta} - p_{ij}^k > \epsilon, \mathcal{E}_t\right) + P\left(\left|\sum_{l\in S^k}\bar{p}_{il}^k - 1\right| \geq \delta, \mathcal{E}_t\right). \end{aligned} \tag{F.2}$$

We have

$$P\left(\frac{\bar{p}_{ij}^k}{1-\delta} - p_{ij}^k > \epsilon, \mathcal{E}_t\right) \leq P\left(\bar{p}_{ij}^k - p_{ij}^k > \epsilon(1-\delta) - \delta, \mathcal{E}_t\right). \tag{F.3}$$

Note that $\epsilon(1-\delta)-\delta$ is decreasing in δ. We can choose a δ small enough such that $\epsilon(1-\delta)-\delta > \epsilon/2$. Then

$$\begin{aligned} P\left(\bar{p}_{ij}^k - p_{ij}^k > \epsilon(1-\delta)-\delta, \mathcal{E}_t\right) &\leq P\left(\bar{p}_{ij}^k - p_{ij}^k > \frac{\epsilon}{2}, \mathcal{E}_t\right) \\ &\leq \frac{2}{t^2}, \end{aligned} \tag{F.4}$$

for $L \geq 6/(\epsilon^2)$. We also have

$$P\left(\left|\sum_{l\in S^k} \bar{p}_{il}^k - 1\right| \geq \delta, \mathcal{E}_t\right) \leq P\left(\sum_{l\in S^k} |\bar{p}_{il}^k - p_{il}^k| \geq \delta, \mathcal{E}_t\right).$$

Consider the events

$$\begin{aligned} \mathcal{A}_{k,i} &= \left\{|\bar{p}_{il}^k - p_{il}^k| < \delta/|S^k|, \forall k \in \mathcal{K}\right\}, \\ \mathcal{B}_{k,i} &= \left\{\sum_{l\in S^k} |\bar{p}_{il}^k - p_{il}^k| < \delta\right\}. \end{aligned}$$

If $\omega \in \mathcal{A}_{k,i}$, then $\omega \in \mathcal{B}_{k,i}$. Thus, $\mathcal{A}_{k,i} \subset \mathcal{B}_{k,i}$, $\mathcal{B}_{k,i}^c \subset \mathcal{A}_{k,i}^c$. Then

$$\begin{aligned} P\left(\sum_{l\in S^k} |\bar{p}_{il}^k - p_{il}^k| \geq \delta, \mathcal{E}_t\right) &= P(\mathcal{B}_{k,i}^c, \mathcal{E}_t) \\ &\leq P(\mathcal{A}_{k,i}^c, \mathcal{E}_t) \\ &= P\left(\bigcup_{l\in S^k} \left\{|\bar{p}_{il}^k - p_{il}^k| \geq \delta/|S^k|\right\}, \mathcal{E}_t\right) \\ &\leq \sum_{l\in S^k} P\left(|\bar{p}_{il}^k - p_{il}^k| \geq \delta/S_{\max},\ \mathcal{E}_t\right) \\ &\leq \frac{2S_{\max}}{t^2}, \end{aligned} \tag{F.5}$$

for $L \geq S_{\max}^2/(2\delta^2)$. Combining (F.1), (F.4) and (F.5) we get

$$P\left(|\hat{p}_{ij,t}^k - p_{ij}^k| > \epsilon, \mathcal{E}_t\right) \leq \frac{2S_{\max}+2}{t^2},$$

for $L \geq \max\{6/(\epsilon^2), S_{\max}^2/(2\delta^2)\} = C_{\boldsymbol{P}}(\epsilon)$.

APPENDIX G

Proof of Lemma IV.17

When the estimated belief is in J_l, for any suboptimal action u, we have

$$\mathcal{L}^*(\psi_t, \boldsymbol{P}) - \mathcal{L}(\psi_t, u, h_{\boldsymbol{P}}, \boldsymbol{P}) \geq \underline{\Delta}. \tag{G.1}$$

Let $\epsilon < \underline{\Delta}/4$. When $\mathcal{F}_t(\epsilon)$ happens, we have

$$\left|\mathcal{I}_t(\hat{\psi}_t, u) - \mathcal{L}(\hat{\psi}_t, u, h_{\boldsymbol{P}}, \hat{\boldsymbol{P}}_t)\right| \leq \epsilon, \tag{G.2}$$

for all $u \in U$. Since $T_{\boldsymbol{P}}(\psi, y, u)$ is continuous in $\boldsymbol{P}$, and $h_{\boldsymbol{P}}(\psi)$ is continuous in ψ, there exists $\delta_e > 0$ such that $||\hat{\boldsymbol{P}}_t - \boldsymbol{P}||_1 < \delta_e$ implies that

$$\left|\mathcal{L}(\hat{\psi}_t, u, h_{\boldsymbol{P}}, \boldsymbol{P}) - \mathcal{L}(\hat{\psi}_t, u, h_{\boldsymbol{P}}, \hat{\boldsymbol{P}}_t)\right| \leq \underline{\Delta}/4, \tag{G.3}$$

for all $u \in U$. Let $u^* \in O(J_l; \boldsymbol{P})$. Using (G.1), (G.2) and (G.3), we have

$$\begin{aligned}
\mathcal{I}(\hat{\psi}_t, u^*) &\geq \mathcal{L}(\hat{\psi}_t, u^*, h_{\boldsymbol{P}}, \hat{\boldsymbol{P}}_t) - \epsilon \\
&\geq \mathcal{L}(\hat{\psi}_t, u^*, h_{\boldsymbol{P}}, \boldsymbol{P}) - \epsilon - \underline{\Delta}/4 \\
&= \mathcal{L}^*(\hat{\psi}_t, \boldsymbol{P}) - \epsilon - \underline{\Delta}/4
\end{aligned}$$

$$
\begin{aligned}
&\geq \mathcal{L}(\hat{\psi}_t, u, h_{\boldsymbol{P}}, \boldsymbol{P}) + 3\underline{\Delta}/4 - \epsilon \\
&\geq \mathcal{L}(\hat{\psi}_t, u, h_{\boldsymbol{P}}, \hat{\boldsymbol{P}}_t) + \underline{\Delta}/2 - \epsilon \\
&\geq \mathcal{I}(\hat{\psi}_t, u) + \underline{\Delta}/2 - 2\epsilon \\
&> \mathcal{I}(\hat{\psi}_t, u) \ .
\end{aligned}
$$

Therefore, we have

$$
\left\{\hat{\psi}_t \in J_l, U_t = u, ||\hat{\boldsymbol{P}}_t - \boldsymbol{P}||_1 < \delta_e, \mathcal{E}_t, \mathcal{F}_t\right\} = \emptyset \ . \tag{G.4}
$$

Recall that for any $u \notin O(J_l; \boldsymbol{P})$,

$$
\begin{aligned}
E^{\boldsymbol{P}}_{\psi_0,\alpha}[D_{1,1}(T, \epsilon, J_l, u)] &= \sum_{t=1}^{T} P\left(\hat{\psi}_t \in J_l, U_t = u, \mathcal{E}_t, \mathcal{F}_t\right) \\
&= \sum_{t=1}^{T} P\left(\hat{\psi}_t \in J_l, U_t = u, ||\hat{\boldsymbol{P}}_t - \boldsymbol{P}||_1 < \delta_e, \mathcal{E}_t, \mathcal{F}_t\right) \\
&+ \sum_{t=1}^{T} P\left(\hat{\psi}_t \in J_l, U_t = u, ||\hat{\boldsymbol{P}}_t - \boldsymbol{P}||_1 \geq \delta_e, \mathcal{E}_t, \mathcal{F}_t\right) \\
&\leq \sum_{t=1}^{T} P\left(||\hat{\boldsymbol{P}}_t - \boldsymbol{P}||_1 \geq \delta_e, \mathcal{E}_t\right), \qquad \text{(G.5)}
\end{aligned}
$$

where (G.5) follows from (G.4). Therefore for any $u \notin O(J_l; \boldsymbol{P})$,

$$
\begin{aligned}
&E^{\boldsymbol{P}}_{\psi_0,\alpha}\left[D_{1,1}(T, \epsilon, J_l, u)\right] \\
&\leq \sum_{t=1}^{T} P\left(\left\|\hat{\boldsymbol{P}}_t - \boldsymbol{P}\right\|_1 \geq \delta_e, \mathcal{E}_t\right) \\
&\leq \sum_{t=1}^{T} P\left(\left\{|\hat{p}^k_{ij,t} - p^k_{ij}| \geq \frac{\delta_e}{K S^2_{\max}}, \text{ for some } k \in \mathcal{K},\ i, j \in S^k\right\}, \mathcal{E}_t\right) \\
&\leq \sum_{t=1}^{T} \sum_{k=1}^{K} \sum_{(i,j) \in S^k \times S^k} P\left(|\hat{p}^k_{ij,t} - p^k_{ij}| \geq \frac{\delta_e}{K S^2_{\max}}, \mathcal{E}_t\right) \\
&\leq 2K S^2_{\max}(S_{\max} + 1)\beta,
\end{aligned}
$$

for $L \geq C_{\boldsymbol{P}}(\delta_e/(KS_{\max}^2))$, where the last equation follows from Lemma IV.13.

APPENDIX H

Proof of Lemma IV.18

Since $h_{\tilde{\boldsymbol{P}}}$ is continuous in ψ by Lemma IV.2 for any $\tilde{\boldsymbol{P}}$ such that Assumption IV.1 holds, and since $\bar{r}(\psi), V_{\tilde{\boldsymbol{P}}}, T_{\tilde{\boldsymbol{P}}}$ are continuous in $\tilde{\boldsymbol{P}}$, we have for any $\psi \in \boldsymbol{\Psi}$:

$$g_{\hat{\boldsymbol{P}}} + h_{\hat{\boldsymbol{P}}}(\psi) = \arg\max_{u \in U} \left\{ \bar{r}(\psi, u) + \sum_{y \in S^u} V_{\hat{\boldsymbol{P}}}(\psi, y, u) h_{\hat{\boldsymbol{P}}}(T_{\hat{\boldsymbol{P}}}(\psi, y, u)) \right\}$$

$$= \arg\max_{u \in U} \left\{ \bar{r}(\psi, u) + \sum_{y \in S^u} V_{\boldsymbol{P}}(\psi, y, u) h_{\hat{\boldsymbol{P}}}(T_{\boldsymbol{P}}(\psi, y, u)) + q(\boldsymbol{P}, \hat{\boldsymbol{P}}, \psi, u) \right\}, \quad \text{(H.1)}$$

for some function q such that

$$\lim_{\hat{\boldsymbol{P}} \to \boldsymbol{P}} q(\boldsymbol{P}, \hat{\boldsymbol{P}}, \psi, u) = 0, \ \forall \psi \in \boldsymbol{\Psi}, u \in U \ .$$

Let $\bar{r}(\boldsymbol{P}, \hat{\boldsymbol{P}}, \psi, u) = \bar{r}(\psi, u) + q(\boldsymbol{P}, \hat{\boldsymbol{P}}, \psi, u)$. We can write (H.1) as

$$g_{\hat{\boldsymbol{P}}} + h_{\hat{\boldsymbol{P}}}(\psi) = \arg\max_{u \in U} \left\{ \bar{r}(\boldsymbol{P}, \hat{\boldsymbol{P}}, \psi, u) + \sum_{y \in S^u} V_{\boldsymbol{P}}(\psi, y, u) h_{\hat{\boldsymbol{P}}}(T_{\boldsymbol{P}}(\psi, y, u)) \right\} \ . \quad \text{(H.2)}$$

Note that (H.2) is the average reward optimality equation for a system with set of

transition probability matrices $\boldsymbol{P}$, and perturbed rewards $\bar{r}(\boldsymbol{P}, \hat{\boldsymbol{P}}, \psi, u)$. Since

$$\lim_{\hat{\boldsymbol{P}} \to \boldsymbol{P}} r(\boldsymbol{P}, \hat{\boldsymbol{P}}, \psi, u) = \bar{r}(\psi, u), \ \forall \psi \in \boldsymbol{\Psi}, u \in U,$$

we expect $h_{\hat{\boldsymbol{P}}}$ to converge to $h_{\boldsymbol{P}}$. Next, we prove that this is true. Let $F_{\hat{P}}$ denote the dynamic programming operator defined in (4.3), with transition probabilities $\boldsymbol{P}$ and rewards $r(\boldsymbol{P}, \hat{\boldsymbol{P}}, \psi, u)$. Then, by S-1 of Lemma (IV.2), there exists a sequence of functions $v_{0,\hat{\boldsymbol{P}}}, v_{1,\hat{\boldsymbol{P}}}, v_{2,\hat{\boldsymbol{P}}}, \ldots$ such that $v_{0,\hat{\boldsymbol{P}}} = 0$, $v_{l,\hat{\boldsymbol{P}}} = F_{\hat{\boldsymbol{P}}} v_{l-1,\hat{\boldsymbol{P}}}$ and another sequence of functions $v_{0,\boldsymbol{P}}, v_{1,\boldsymbol{P}}, v_{2,\boldsymbol{P}}, \ldots$ such that $v_{0,\boldsymbol{P}} = 0$, $v_{l,\boldsymbol{P}} = F_{\boldsymbol{P}} v_{l-1,\boldsymbol{P}}$, for which

$$\lim_{l \to \infty} v_{l,\hat{\boldsymbol{P}}} = h_{\hat{\boldsymbol{P}}}, \tag{H.3}$$

$$\lim_{l \to \infty} v_{l,\boldsymbol{P}} = h_{\boldsymbol{P}}, \tag{H.4}$$

uniformly in ψ.

Next, we prove that for any $l \in \{1, 2, \ldots\}$, $\lim_{\hat{\boldsymbol{P}} \to \boldsymbol{P}} v_{l,\hat{\boldsymbol{P}}} = v_{l,\boldsymbol{P}}$ uniformly in ψ. Let

$$q_{\max}(\boldsymbol{P}, \hat{\boldsymbol{P}}) = \sup_{u \in U, \psi \in \boldsymbol{\Psi}} \left| q(\boldsymbol{P}, \hat{\boldsymbol{P}}, \psi, u) \right| .$$

By Equation 2.27 in *Platzman* (1980), we have

$$\begin{aligned} \sup_{\psi \in \boldsymbol{\Psi}} \left\{ \left| v_{l,\hat{\boldsymbol{P}}}(\psi) - v_{l,\boldsymbol{P}}(\psi) \right| \right\} &= \sup_{\psi \in \boldsymbol{\Psi}} \left\{ \left| F^{l-1} v_{1,\hat{\boldsymbol{P}}}(\psi) - F^{l-1} v_{l,\boldsymbol{P}}(\psi) \right| \right\} \\ &\leq \sup_{\psi \in \boldsymbol{\Psi}} \left\{ \left| v_{1,\hat{\boldsymbol{P}}}(\psi) - v_{1,\boldsymbol{P}}(\psi) \right| \right\} \\ &\leq 2 q_{\max}(\boldsymbol{P}, \hat{\boldsymbol{P}}), \end{aligned} \tag{H.5}$$

where the last inequality follows form $v_{0,\boldsymbol{P}} = 0$, $v_{0,\hat{\boldsymbol{P}}} = 0$, and

$$v_{1,\boldsymbol{P}}(\psi) = \max_{u \in U} \{ \bar{r}(\psi, u) \}$$

$$v_{1,\hat{\boldsymbol{P}}}(\psi) = \max_{u \in U} \left\{ \bar{r}(\psi, u) + q(\boldsymbol{P}, \hat{\boldsymbol{P}}, \psi, u) \right\}.$$

Consider a sequence $\{\hat{\boldsymbol{P}}_n\}_{n=1}^{\infty}$ which converges to $\boldsymbol{P}$. Since $\lim_{n\to\infty} q_{\max}(\boldsymbol{P}, \hat{\boldsymbol{P}}_n) = 0$, for any $\epsilon > 0$, there exists N_0 such that for all $n > N_0$ we have $q_{\max}(\boldsymbol{P}, \hat{\boldsymbol{P}}_n) < \epsilon/2$, which implies by (H.5) that

$$\sup_{\psi \in \boldsymbol{\Psi}} \left\{ |v_{l,\hat{\boldsymbol{P}}_n}(\psi) - v_{l,\boldsymbol{P}}(\psi)| \right\} < \epsilon,$$

for all $\psi \in \boldsymbol{\Psi}$. Therefore, for any $l \in \{1, 2, \ldots\}$, we have

$$\lim_{\hat{\boldsymbol{P}} \to \boldsymbol{P}} v_{l,\hat{\boldsymbol{P}}} = v_{l,\boldsymbol{P}}, \tag{H.6}$$

uniformly in ψ. Using (H.3) and (H.4), for any $\epsilon > 0$ and any $n \in \{1, 2, \ldots\}$, there exists $N_1(n)$ such that for any $l > N_1(n)$ and $\psi \in \boldsymbol{\Psi}$, we have

$$|v_{l,\hat{\boldsymbol{P}}_n}(\psi) - h_{\hat{\boldsymbol{P}}_n}(\psi)| < \epsilon/3,$$
$$|v_{l,\boldsymbol{P}}(\psi) - h_{\boldsymbol{P}}(\psi)| < \epsilon/3.$$

Similarly using (H.6), for any $\epsilon > 0$, there exists N_0 such that for all $n > N_0$ and $\psi \in \boldsymbol{\Psi}$, we have

$$|v_{l,\hat{\boldsymbol{P}}_n}(\psi) - v_{l,\boldsymbol{P}}(\psi)| \leq \epsilon/3. \tag{H.7}$$

These imply that for any $\epsilon > 0$, there exists $N_2 \geq N_0$ and such that for all $n > N_2$, such that for all $\psi \in \boldsymbol{\Psi}$, we have

(H.8)

$$|h_{\boldsymbol{P}}(\psi) - h_{\hat{\boldsymbol{P}}_n}(\psi)| \leq |h_{\boldsymbol{P}}(\psi) - v_{l,\boldsymbol{P}}(\psi)| + |v_{l,\hat{\boldsymbol{P}}_n}(\psi) - v_{l,\boldsymbol{P}}(\psi)| + |v_{l,\hat{\boldsymbol{P}}_n}(\psi) - h_{\hat{\boldsymbol{P}}_n}(\psi)|$$

$$< \epsilon,$$

since there exists some $l > N_1(n)$ such that (H.7) holds. Therefore, for any $\epsilon > 0$ there exists some $\eta > 0$ such that $|\boldsymbol{P} - \hat{\boldsymbol{P}}| < \eta$ implies $|h_{\boldsymbol{P}}(\psi) - h_{\hat{\boldsymbol{P}}}(\psi)|_\infty \leq \epsilon$.

APPENDIX I

Proof of Lemma IV.19

We have by Lemma IV.18,

$$\left\{\left\|P^k - \hat{P}_t^k\right\|_1 < \varsigma, \forall k \in \mathcal{K}\right\} \subset \left\{\|h_{\boldsymbol{P}} - h_t\|_\infty < \epsilon\right\} \ ,$$

which implies

$$\left\{\left\|P^k - \hat{P}_t^k\right\|_1 \geq \varsigma, \text{ for some } k \in \mathcal{K}\right\} \supset \left\{\|h_{\boldsymbol{P}} - h_t\|_\infty \geq \epsilon\right\} \ .$$

Then

$$\begin{aligned}
E_{\psi_0,\alpha}^{\boldsymbol{P}}[D_{1,2}(T,\epsilon)] &= E_{\psi_0,\alpha}^{\boldsymbol{P}}\left[\sum_{t=1}^{T} I(\mathcal{E}_t, \mathcal{F}_t^c(\epsilon))\right] \\
&\leq \sum_{t=1}^{T} P\left(\left\|P^k - \hat{P}_t^k\right\|_1 \geq \varsigma, \text{ for some } k \in \mathcal{K}, \mathcal{E}_t\right) \\
&\leq \sum_{k=1}^{K} \sum_{(i,j)\in S^k \times S^k} \sum_{t=1}^{T} P\left(|p_{ij}^k - \hat{p}_{ij,t}^k| > \frac{\varsigma}{S_{\max}^2}, \mathcal{E}_t\right) \\
&\leq 2KS_{\max}^2(S_{\max}+1)\beta \ ,
\end{aligned}$$

where the last equation follows from Lemma IV.13.

APPENDIX J

Proof of Lemma IV.20

Consider $t > 0$

$$
\begin{aligned}
|(\hat{\psi}_t)_{\boldsymbol{x}} - (\psi_t)_{\boldsymbol{x}}| &= \left| \prod_{k=1}^{K} \left((\hat{P}_t^k)^{\tau^k} e_{s^k}^k \right)_{x^k} - \prod_{k=1}^{K} \left((P^k)^{\tau_k} e_{s^k}^k \right)_{x^k} \right| \\
&\leq \sum_{k=1}^{K} \left| \left((\hat{P}_t^k)^{\tau^k} e_{s^k}^k \right)_{x^k} - \left((P^k)^{\tau_k} e_{s^k}^k \right)_{x^k} \right| \\
&\leq \sum_{k=1}^{K} \left\| (\hat{P}_t^k)^{\tau^k} e_{s^k}^k - (P^k)^{\tau^k} e_{s^k}^k \right\|_1 \\
&\leq C_1(\boldsymbol{P}) \sum_{k=1}^{K} \left\| \hat{P}_t^k - P^k \right\|_1, \qquad \text{(J.1)}
\end{aligned}
$$

where last inequality follows from Lemma A.6. By (J.1)

$$
\left\| \hat{\psi}_t - \psi_t \right\|_1 \leq |S^1| \dots |S^K| C_1(\boldsymbol{P}) \sum_{k=1}^{K} \left\| \hat{P}_t^k - P^k \right\|_1.
$$

Thus we have

$$
\begin{aligned}
&P\left(\left\| \hat{\psi}_t - \psi_t \right\|_1 > \epsilon, \mathcal{E}_t \right) \\
&\leq P\left(\sum_{k=1}^{K} \left\| \hat{P}_t^k - P^k \right\|_1 > \epsilon / (|S^1| \dots |S^K| C_1(\boldsymbol{P})), \mathcal{E}_t \right)
\end{aligned}
$$

$$
\begin{aligned}
&\leq \sum_{k=1}^{K} P\left(\left\|\hat{P}_t^k - P^k\right\|_1 > \epsilon/(K|S^1|\dots|S^K|C_1(\boldsymbol{P})), \mathcal{E}_t\right) \\
&\leq \sum_{k=1}^{K} \sum_{(i,j)\in S^k\times S^k} P\left(|\hat{p}_{ij,t}^k - p_{ij}^k| > \frac{\epsilon}{(KS_{\max}^2|S^1|\dots|S^K|C_1(\boldsymbol{P}))}, \mathcal{E}_t\right) \\
&\leq 2KS_{\max}^2 \frac{S_{\max}+1}{t^2},
\end{aligned}
$$

where last inequality follows from Lemma IV.13 since

$$
L \geq C_{\boldsymbol{P}}(\epsilon/(KS_{\max}^2|S^1|\dots|S^K|C_1(\boldsymbol{P}))) \ .
$$

Then,

$$
E_{\psi_0,\alpha}^{\boldsymbol{P}}[D_{2,1}(T,\epsilon)] = \sum_{t=1}^{T} P_{\psi_0,\alpha}\left(\left\|\psi_t - \hat{\psi}_t\right\|_1 > \epsilon, \mathcal{E}_t\right) \leq 2KS_{\max}^2(S_{\max}+1)\beta.
$$

APPENDIX K

Proof of Lemma VI.1

Let $\tilde{\alpha}_j(b)$ be the arm selected by agent j in its bth block. Assume that agent j has completed the b'th block.

$$\begin{aligned}
B^{i,j}(b') &= 1 + \sum_{b=K+1}^{b'} I(\tilde{\alpha}_j(b) = i) \\
&\leq l + \sum_{b=K+1}^{b'} I(\tilde{\alpha}_j(b) = i, B^{i,j}(b-1) \geq l) \\
&\leq l + \sum_{b=K+1}^{b'} \sum_{k=1}^{M} I\left(g^{k,j}_{b-1,B^{k,j}(b-1)} \leq g^{i,j}_{b-1,B^{i,j}(b-1)}, B^{i,j}(b-1) \geq l\right) \\
&\leq l + \sum_{k=1}^{M} \sum_{b=K+1}^{b'} I\left(\min_{0<s_k<b} g^{k,j}_{b-1,s_k} \leq \max_{l\leq s_i<b} g^{i,j}_{b-1,s_i}\right) \\
&\leq l + \sum_{k=1}^{M} \sum_{b=1}^{b'} \sum_{s_k=1}^{b-1} \sum_{s_i=l}^{b-1} I\left(g^{k,j}_{b-1,s_k} \leq g^{i,j}_{b-1,s_i}\right) \ . \qquad \text{(K.1)}
\end{aligned}$$

Then, proceeding from (K.1) the same way as in the proof of Lemma E.2, but using the Chernoff-Hoeffding bound given in Lemma A.7 for the IID reward process instead of the large deviation bound for a Markov chain, for $l = \left\lceil \frac{8 \ln b'}{(\mu^M - \mu^i)^2} \right\rceil$, we have

$$E[B^{i,j}(b_j(T))|b_j(T) = b'] \leq \frac{8 \ln b'}{(\mu^M - \mu^i)^2} + 1 + M\beta \ ,$$

which implies

$$E[B^{i,j}(b_j(T))] \leq \frac{8 \ln T}{(\mu^M - \mu^i)^2} + 1 + M\beta \ .$$

APPENDIX L

Proof of Lemma VI.3

The event that the index of any one of the optimal arms calculated by agent j is in wrong order at v_jth block of agent j is included in the event

$$E_j(v_j) := \bigcup_{a=1}^{M} \bigcup_{c=a+1}^{K} \{g^{a,j}_{v_j,B^{a,j}(v_j)} \leq g^{c,j}_{v_j,B^{c,j}(v_j)}\}.$$

Let $\mathcal{B}_{i,j}(b)$ denote the set of blocks that agent i is in, during the bth block of agent j. The event that the index of any one of the optimal arms calculated by agent $i \neq j$ is in wrong order during any interval at v_jth block of agent j is included in the event

$$E_i(v_j) := \bigcup_{v_i \in \mathcal{B}_{i,j}(v_j)} \bigcup_{a=1}^{M} \bigcup_{c=a+1}^{K} \{g^{a,i}_{v_i,B^{a,i}(v_i)} \leq g^{c,i}_{v_i,B^{c,i}(v_i)}\}.$$

The event that the index of any one of the optimal arms calculated by any agent is in wrong order during any interval at v_jth block of agent j is included in the event

$$\bigcup_{i=1}^{M} E_i(v_j).$$

Let $\tilde{B}^j(b_j(T))$ be the number of completed blocks of agent j up to time T in which there is at least one agent who has a wrong order for an index of some optimal arm

during some part of a block of agent j. Then

$$\tilde{B}^j(b_j(T)) = \sum_{v_j=1}^{b_j(T)} I\left(\bigcup_{i=1}^{M} E_i(v_j)\right) \leq \sum_{i=1}^{M}\sum_{v_j=1}^{b_j(T)} I(E_i(v_j)).$$

Using union bound we have,

$$\sum_{v_j=1}^{b_j(T)} I(E_j(v_j)) \leq \sum_{v_j=1}^{b_j(T)}\sum_{a=1}^{M}\sum_{c=a+1}^{K} I(g^{a,j}_{v_j,B^{a,j}(v_j)} \leq g^{c,j}_{v_j,B^{c,j}(v_j)}), \tag{L.1}$$

and

$$\sum_{v_j=1}^{b_j(T)} I(E_i(v_j)) \leq \sum_{v_j=1}^{b_j(T)}\sum_{v_i\in\mathcal{B}_{i,j}(v_j)}\sum_{a=1}^{M}\sum_{c=a+1}^{K} I(g^{a,i}_{v_i,B^{a,i}(v_i)} \leq g^{c,i}_{v_i,B^{c,i}(v_i)}). \tag{L.2}$$

Proceeding from (L.1) the same way as in the proof of Lemma VI.1, we have

$$E\left[\sum_{v_j=1}^{b_j(T)} I(E_j(v_j))\right] \leq \sum_{a=1}^{M}\sum_{c=a+1}^{K}\left(\frac{8\ln T}{(\mu^a-\mu^c)^2}+1+\beta\right). \tag{L.3}$$

In (L.2) for each block of agent j, the second sum counts the number of blocks of agent i which intersects with that block of agent j. This is less than or equal to counting the number of blocks of agent j which intersects with a block of agent i for blocks $1,\ldots,b_i(T)+1$ of i. We consider block $b_i(T)+1$ of i because it may intersect with completed blocks of agent j up to $b_j(T)$. Thus we have

$$\sum_{v_j=1}^{b_j(T)} I(E_i(v_j)) \leq \sum_{v_i=1}^{b_i(T)+1}\sum_{v_j\in\mathcal{B}_{j,i}(v_i)}\sum_{a=1}^{M}\sum_{c=a+1}^{K} I(g^{a,i}_{v_i,B^{a,i}(v_i)} \leq g^{c,i}_{v_i,B^{c,i}(v_i)})$$

with probability 1. Taking the conditional expectation we get

$$E\left[\sum_{v_j=1}^{b_j(T)} I(E_i(v_j))\middle| |\mathcal{B}_{j,i}(1)| = n_1,\ldots,|\mathcal{B}_{j,i}(b_i(T)+1)| = n_{b_i(T)+1}\right]$$

$$= E\left[\sum_{v_i=1}^{b_i(T)+1}\sum_{a=1}^{M}\sum_{c=a+1}^{K} n_{v_i} I(g^{a,i}_{v_i,B^{a,i}(v_i)} \leq g^{c,i}_{v_i,B^{c,i}(v_i)})\right]$$

$$\leq \max_{v_i=1:b_i(T)+1} E\left[\sum_{v_i=1,a=1,c=a+1}^{b_i(T)+1,M,K} I(g^{a,i}_{v_i,B^{a,i}(v_i)} \leq g^{c,i}_{v_i,B^{c,i}(v_i)})\right].$$

Using the above result and following the same approach as in (L.3), we have

$$E\left[\sum_{v_j=1}^{b_j(T)} I(E_i(v_j))\right]$$

$$\leq E\left[\max_{v_i=1:b_i(T)+1} |\mathcal{B}_{j,i}(v_i)|\right]\left(\sum_{a=1}^{M}\sum_{c=a+1}^{K}\left(\frac{8\ln T}{(\mu^a-\mu^c)^2}+1+\beta\right)\right). \tag{L.4}$$

The next step is to bound $E\left[\max_{v_i=1:b_i(T)+1} |\mathcal{B}_{j,i}(v_i)|\right]$. Let $l_i(v_i)$ be the length of the v_ith block of agent i. Clearly we have $|\mathcal{B}_{j,i}(v_i)| \leq l_i(v_i)$ with probability 1. Therefore $E\left[\max_{v_i=1:b_i(T)+1} |\mathcal{B}_{j,i}(v_i)|\right] \leq E\left[\max_{v_i=1:b_i(T)+1} l_i(v_i)\right]$. Note that the random variables $l_i(v_i), v_i = 1 : b_i(T)+1$ are independent due to Markov property but not necessarily identically distributed since agent i might play different arms at different blocks.

Let $p^k_{xy}(t)$ denote the t step transition probability from state x to y of arm k. Since all arms are ergodic there exists $N > 0$ such that $p^k_{xy}(N) > 0$, for all $k \in \mathcal{K}, x, y \in S^k$. Let $p^* = \min_{k\in\mathcal{K},x,y\in S^k} p^k_{xy}(N)$. We define a geometric random variable $l_{\max}$ with distribution

$$P(l_{\max} = 2Nz) = (1-p^*)^{z-1}p^*, z = 1, 2, \ldots$$

It is easy to see that

$$P(l_i(v_i) \leq z) \geq P(l_{\max} \leq z), z = 1, 2, \ldots$$

Consider an IID set of random variables $\{l_{\max}(1), \ldots, l_{\max}(b_i(T)+1)\}$ where each $l_{\max}(v)$, $v = 1, 2, \ldots, b_i(T)+1$, have the same distribution as $l_{\max}$. Since $l_i(.)$ and $l_{\max}(.)$ are non-negative random variables we have

$$\begin{aligned} E\left[\max_{v_i=1:b_i(T)+1} l_i(v_i)\middle| b_i(T)=b\right] &= \sum_{z=0}^{\infty} P\left(\max_{v_i=1:b+1} l_i(v_i) > z\right) \\ &= \sum_{z=0}^{\infty}\left(1-\prod_{v_i=1}^{b+1} P(l_i(v_i)\le z)\right) \\ &\le \sum_{z=0}^{\infty}\left(1-\prod_{v_i=1}^{b+1} P(l_{\max}(v_i)\le z)\right) \\ &= E\left[\max_{v_i=1:b_i(T)+1} l_{\max}(v_i)\middle| b_i(T)=b\right]. \end{aligned}$$

Finally,

$$\begin{aligned} E\left[\max_{v_i=1:b_i(T)+1} l_{\max}(v_i)\middle| b_i(T)=b\right] &= \sum_{z=0}^{\infty}\left(1-P(l_{\max}\le z)^{b+1}\right) \\ &= 2N\sum_{z=0}^{\infty}\left(1-P(l_{\max}\le 2Nz)^{b+1}\right) \\ &< 2N\left(1+\frac{1}{\lambda}\sum_{l=1}^{b+1}\frac{1}{l}\right) \qquad \text{(L.5)} \\ &\le 2N\left(1+\frac{1}{\lambda}(\ln T+1)\right), \qquad \text{(L.6)} \end{aligned}$$

where $\lambda = \ln\left(\frac{1}{1-p^*}\right)$, (L.5) follows from Equation 4 in *Eisenberg* (2008) and (L.6) follows from $b_i(T)+1 \le \log T$ with probability 1. Using the above results on (L.4) we get

$$\begin{aligned} &E\left[\sum_{v_j=1}^{b_j(T)} I(E_i(v_j))\right] \\ &< 2N\left(1+\frac{1}{\lambda}(\ln T+1)\right)\left(\sum_{a=1}^{M}\sum_{c=a+1}^{K}\left(\frac{8\ln T}{(\mu^a-\mu^c)^2}+1+\beta\right)\right). \qquad \text{(L.7)} \end{aligned}$$

Using (L.3) and (L.7), we have

$$E[\tilde{B}^j(b_j(T))] \leq E\left[\sum_{i=1}^{M}\sum_{v_j=1}^{b_j(T)} I(E_i(v_j))\right]$$
$$< \left[2N(M-1)\left(1+\frac{1}{\lambda}(\ln T+1)\right)+1\right]\left(\sum_{a=1}^{M}\sum_{c=a+1}^{K}\left(\frac{8\ln T}{(\mu^a-\mu^c)^2}+1+\beta\right)\right).$$

Thus we have

$$E[B'(T)] < M\left[2N(M-1)\left(1+\frac{1}{\lambda}(\ln T+1)\right)+1\right]$$
$$\times\sum_{a=1}^{M}\sum_{c=a+1}^{K}\left(\frac{8\ln T}{(\mu^a-\mu^c)^2}+1+\beta\right). \tag{L.8}$$

APPENDIX M

Proof of Lemma VI.4

Let b be a block in which all agents know the correct order of the M-best channels and $b-1$ be a block in which there exists at least one agent whose order of indices for M-best channels are different then the order of the mean rewards. We call such an event a transition from a bad state to a good state. Then by Lemma VI.1 the expected number of blocks needed to settle to an orthogonal configuration after block b is bounded by O_B. Since the expected number of such transitions is $E[B'(T)]$, we have $E[H(T)] \leq O_B E[B'(T)]$.

BIBLIOGRAPHY

BIBLIOGRAPHY

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

Agrawal, R. (1995b), The continuum-armed bandit problem, *SIAM journal on control and optimization*, *33*, 1926.

Agrawal, R., D. Teneketzis, and V. Anantharam (1989), Asymptotically efficient adaptive allocation schemes for controlled Markov chains: Finite parameter space, *IEEE Trans. Automat. Control*, pp. 258–267.

Ahmad, S., M. Liu, T. Javidi, Q. Zhao, and B. Krishnamachari (2009), Optimality of myopic sensing in multichannel opportunistic access, *Information Theory, IEEE Transactions on*, *55*(9), 4040–4050.

Anandkumar, A., N. Michael, A. Tang, and A. Swami (2011), Distributed algorithms for learning and cognitive medium access with logarithmic regret, *Selected Areas in Communications, IEEE Journal on*, *29*(4), 731–745.

Anantharam, V., P. Varaiya, and J. Walrand (1987a), Asymptotically efficient allocation rules for the multiarmed bandit problem with multiple plays-Part I: IID rewards, *IEEE Trans. Automat. Contr.*, pp. 968–975.

Anantharam, V., P. Varaiya, and J. Walrand (1987b), Asymptotically efficient allocation rules for the multiarmed bandit problem with multiple plays-Part II: Markovian rewards, *IEEE Trans. Automat. Contr.*, pp. 977–982.

Arora, R., O. Dekel, and A. Tewari (2012), Online bandit learning against an adaptive adversary: from regret to policy regret, *arXiv preprint arXiv:1206.6400*.

Audibert, J., R. Munos, and C. Szepesvári (2009), Exploration-exploitation tradeoff using variance estimates in multi-armed bandits, *Theoretical Computer Science*, *410*(19), 1876–1902.

Auer, P., and R. Ortner (2010), UCB revisited: Improved regret bounds for the stochastic multi-armed bandit problem, *Periodica Mathematica Hungarica*, *61*(1), 55–65.

Auer, P., N. Cesa-Bianchi, and P. Fischer (2002), Finite-time analysis of the multi-armed bandit problem, *Machine Learning*, *47*, 235–256.

Auer, P., N. Cesa-Bianchi, Y. Freund, and R. Schapire (2003), The nonstochastic multiarmed bandit problem, *SIAM Journal on Computing*, *32*(1), 48–77.

Auer, P., R. Ortner, and C. Szepesvári (2007), Improved rates for the stochastic continuum-armed bandit problem, *Learning Theory*, pp. 454–468.

Auer, P., T. Jaksch, and R. Ortner (2009), Near-optimal regret bounds for reinforcement learning.

Bartlett, P., V. Dani, T. Hayes, S. Kakade, A. Rakhlin, and A. Tewari (2008), High-probability regret bounds for bandit online linear optimization.

Bergemann, D., and J. Valimaki (2006), Bandit problems, *Cowles Foundation discussion paper*, *1551*.

Bertsimas, D., and J. Niño-Mora (1996), Conservation laws, extended polymatroids and multiarmed bandit problems; a polyhedral approach to indexable systems, *Mathematics of Operations Research*, pp. 257–306.

Bianchi, C., and G. Lugosi (2009), Combinatorial bandits, in *COLT 2009: proceedings of the 22nd Annual Conference on Learning Theory, Montréal, Canada*, Omnipress.

Brezzi, M., and T. Lai (2000), Incomplete learning from endogenous data in dynamic allocation, *Econometrica*, *68*(6), 1511–1516.

Bubeck, S., R. Munos, G. Stoltz, and C. Szepesvari (2008), Online optimization in X-armed bandits, in *Twenty-Second Annual Conference on Neural Information Processing Systems*, Vancouver, Canada.

Burnetas, A., and M. Katehakis (1997), Optimal adaptive policies for Markov decision processes, *Mathematics of Operations Research*, pp. 222–255.

Chlebus, E. (2009), An approximate formula for a partial sum of the divergent p-series, *Applied Mathematics Letters*, *22*, 732–737.

Cope, E. (2009), Regret and convergence bounds for a class of continuum-armed bandit problems, *Automatic Control, IEEE Transactions on*, *54*(6), 1243–1253.

Dai, W., Y. Gai, B. Krishnamachari, and Q. Zhao (2011), The non-bayesian restless multi-armed bandit: A case of near-logarithmic regret, in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pp. 2940–2943, IEEE.

Dani, V., T. Hayes, and S. Kakade (2008), Stochastic linear optimization under bandit feedback, in *Proceedings of the 21st Annual Conference on Learning Theory (COLT)*.

D.W. Turner, J. S., D.M. Young (1995), A Kolmogorov inequality for the sum of independent Bernoulli random variables with unequal means, *Statistics and Probability Letters*, *23*, 243–245.

Eisenberg, B. (2008), On the expectation of the maximum of IID geometric random variables, *Statistics and Probability Letters*, *78*(2), 135–143.

Even-Dar, E., S. Mannor, and Y. Mansour (2002), PAC bounds for multi-armed bandit and Markov decision processes, in *Computational Learning Theory*, pp. 193–209, Springer.

Frostig, E., and G. Weiss (1999), Four proofs of Gittins multiarmed bandit theorem, *Applied Probability Trust*, pp. 1–20.

Gai, Y., B. Krishnamachari, and M. Liu (2011), On the combinatorial multi-armed bandit problem with Markovian rewards, in *IEEE Global Communications Conference (GLOBECOM)*.

Gai, Y., B. Krishnamachari, and R. Jain (2012a), Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations, *to appear in IEEE/ACM Trans. Netw.*

Gai, Y., B. Krishnamachari, and M. Liu (2012b), Online learning for combinatorial network optimization with restless Markovian rewards, *to appear in the 9th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*.

Garivier, A., and O. Cappé (2011), The KL-UCB algorithm for bounded stochastic bandits and beyond, *Arxiv preprint arXiv:1102.2490*.

Garivier, A., and E. Moulines (2008), On upper-confidence bound policies for non-stationary bandit problems, *arXiv preprint arXiv:0805.3415*.

Gillman, D. (1998), A Chernoff bound for random walks on expander graphs, *SIAM Journal on Computing*, *27*, 1203.

Gittins, J., and D. Jones (1972), A dynamic allocation index for sequential design of experiments, *Progress in Statistics, Euro. Meet. Statis.*, *1*, 241–266.

Gittins, J., R. Weber, and K. Glazebrook (1989), *Multi-armed bandit allocation indices*, vol. 25, Wiley Online Library.

Guha, S., K. Munagala, and P. Shi (2010), Approximation algorithms for restless bandit problems, *Journal of the ACM (JACM)*, *58*(1), 3.

Hardin, G. (2009), The tragedy of the commons*, *Journal of Natural Resources Policy Research*, *1*(3), 243–253.

Hazan, E., and S. Kale (2009), Better algorithms for benign bandits, in *Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 38–47, Society for Industrial and Applied Mathematics.

Jiang, C., and R. Srikant (2011), Parametrized stochastic multi-armed bandits with binary rewards, in *American Control Conference (ACC), 2011*, pp. 119–124, IEEE.

Kakhbod, A., and D. Teneketzis (2010), Power allocation and spectrum sharing in multi-user, multi-channel systems with strategic users, in *Decision and Control (CDC), 2010 49th IEEE Conference on*, pp. 1088–1095, IEEE.

Kale, S., L. Reyzin, and R. Schapire (2010), Non-stochastic bandit slate problems, *Advances in Neural Information Processing Systems*, *23*, 1054–1062.

Kleinberg, R. (2004), Nearly tight bounds for the continuum-armed bandit problem, *Advances in Neural Information Processing Systems*, *17*, 697–704.

Kleinberg, R., A. Slivkins, and E. Upfal (2008), Multi-armed bandits in metric spaces, in *Proceedings of the 40th annual ACM symposium on Theory of computing*, pp. 681–690, ACM.

Kleinberg, R., G. Piliouras, and E. Tardos (2009), Multiplicative updates outperform generic no-regret learning in congestion games, in *Annual ACM Symposium on Theory of Computing (STOC)*.

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

Langford, J., and T. Zhang (2007), The epoch-greedy algorithm for contextual multi-armed bandits, *Advances in Neural Information Processing Systems*, *20*.

Lezaud, P. (1998), Chernoff-type bound for finite Markov chains, *Annals of Applied Probability*, pp. 849–867.

Liu, H., K. Liu, and Q. Zhao (2010), Learning in a changing world: Non-bayesian restless multi-armed bandit, *Techinal Report, UC Davis*.

Liu, H., K. Liu, and Q. Zhao (2011), Learning and sharing in a changing world: Non-bayesian restless bandit with multiple players, in *Information Theory and Applications Workshop (ITA), 2011*.

Liu, K., and Q. Zhao (2010), Distributed learning in multi-armed bandit with multiple players, *Signal Processing, IEEE Transactions on*, *58*(11), 5667–5681.

Liu, K., and Q. Zhao (2011), Multi-armed bandit problems with heavy-tailed reward distributions, in *Communication, Control, and Computing (Allerton), 2011 49th Annual Allerton Conference on*, pp. 485–492, IEEE.

López-Benítez, M., and F. Casadevall (2011), Discrete-time spectrum occupancy model based on Markov chain and duty cycle models, in *New Frontiers in Dynamic Spectrum Access Networks (DySPAN), 2011 IEEE Symposium on*, pp. 90–99, IEEE.

Mahajan, A., and D. Teneketzis (2008), Multi-armed bandit problems, *Foundations and Applications of Sensor Management*, pp. 121–151.

Mannor, S., and J. Tsitsiklis (2004), The sample complexity of exploration in the multi-armed bandit problem, *The Journal of Machine Learning Research*, *5*, 623–648.

McMahan, H., and A. Blum (2004), Online geometric optimization in the bandit setting against an adaptive adversary, *Learning theory*, pp. 109–123.

Mersereau, A., P. Rusmevichientong, and J. Tsitsiklis (2009), A structured multi-armed bandit problem and the greedy policy, *Automatic Control, IEEE Transactions on*, *54*(12), 2787–2802.

Mitrophanov, A. Y. (2005), Senstivity and convergence of uniformly ergodic Markov chains, *J. Appl. Prob.*, *42*, 1003–1014.

Monderer, D., and L. S. Shapley (1996), Potential games, *Games and Economic Behavior*, *14*(1), 124–143.

Nino-Mora, J. (2001), Restless bandits, partial conservation laws and indexability, *Advances in Applied Probability*, *33*(1), 76–98.

Ortner, P. (2007), Logarithmic online regret bounds for undiscounted reinforcement learning, in *Advances in Neural Information Processing Systems 19: Proceedings of the 2006 Conference*, vol. 19, p. 49, The MIT Press.

Ortner, R. (2008), Online regret bounds for Markov decision processes with deterministic transitions, in *Algorithmic Learning Theory*, pp. 123–137, Springer.

Pandey, S., D. Chakrabarti, and D. Agarwal (2007), Multi-armed bandit problems with dependent arms, in *Proceedings of the 24th international conference on Machine learning*, pp. 721–728, ACM.

Papadimitriou, C., and J. Tsitsiklis (1999), The complexity of optimal queuing network control, *Mathematics of Operations Research*, *24*(2), 293–305.

Platzman, L. K. (1980), Optimal infinite-horizon undiscounted control of finite probabilistic systems, *SIAM J. Control Optim.*, *18*, 362–380.

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

Rosenthal, R. (1973), A class of games possessing pure-strategy Nash equilibria, *International Journal of Game Theory*, *2*, 65–67.

Rosin, C. (2011), Multi-armed bandits with episode context, *Annals of Mathematics and Artificial Intelligence*, pp. 1–28.

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

Sandholm, W. (2011), *Population games and evolutionary dynamics*, MIT press.

Sheng, S. P., and M. Liu (2012), Optimal contract design for an efficient secondary spectrum market, in *3rd International Conference on Game Theory for Networks (GAMENETS)*.

Slivkins, A. (2009), Contextual bandits with similarity information, *Arxiv preprint arXiv:0907.3986*.

Slivkins, A., and E. Upfal (2008), Adapting to a changing environment: The Brownian restless bandits, in *Proc. 21st Annual Conference on Learning Theory*, pp. 343–354.

Smith, J. M. (1982), Evolution and the theory of games, *Cambridge University Press*.

Tekin, C., and M. Liu (2010), Online algorithms for the multi-armed bandit problem with Markovian rewards, in *Proc. of the 48th Annual Allerton Conference on Communication, Control, and Computing*, pp. 1675–1682.

Tekin, C., and M. Liu (2011a), Adaptive learning of uncontrolled restless bandits with logarithmic regret, in *Proc. of the 49th Annual Allerton Conference on Communication, Control, and Computing*, pp. 983–990.

Tekin, C., and M. Liu (2011b), Online learning in opportunistic spectrum access: A restless bandit approach, in *Proc. of the 30th Annual IEEE International Conference on Computer Communications (INFOCOM)*, pp. 2462 –2470.

Tekin, C., and M. Liu (2011c), Performance and convergence of multi-user online learning, in *Proc. of the 2nd International Conference on Game Theory for Networks (GAMENETS)*.

Tekin, C., and M. Liu (2012a), Approximately optimal adaptive learning in opportunustic spectrum access, in *Proc. of the 31st Annual IEEE International Conference on Computer Communications (INFOCOM)*.

Tekin, C., and M. Liu (2012b), Performance and convergence of multi-user online learning, in *Mechanisms and Games for Dynamic Spectrum Allocation*, edited by T. Alpcan, H. Boche, M. Honig, and H. V. Poor, Cambridge University Press.

Tekin, C., and M. Liu (2012c), Online contract design with ordered preferences, in *Proc. of the 50th Annual Allerton Conference on Communication, Control, and Computing*.

Tekin, C., and M. Liu (2012d), Online learning of rested and restless bandits, *Information Theory, IEEE Transactions on*, *58*(8), 5588–5611.

Tekin, C., M. Liu, R. Southwell, J. Huang, and S. Ahmad (2012), Atomic congestion games on graphs and their applications in networking, *Networking, IEEE/ACM Transactions on*, *20*(5), 1541 –1552.

Tewari, A., and P. Bartlett (2008), Optimistic linear programming gives logarithmic regret for irreducible MDPs, *Advances in Neural Information Processing Systems*, *20*, 1505–1512.

Thompson, W. (1933), On the likelihood that one unknown probability exceeds another in view of the evidence of two samples, *Biometrika*, pp. 285–294.

Tsitsiklis, J. (1994), A short proof of the Gittins index theorem, *The Annals of Applied Probability*, pp. 194–199.

Varaiya, P., J. Walrand, and C. Buyukkoc (1985), Extensions of the multiarmed bandit problem: the discounted case, *Automatic Control, IEEE Transactions on*, *30*(5), 426–439.

Wang, C., S. Kulkarni, and H. Poor (2005), Bandit problems with side observations, *Automatic Control, IEEE Transactions on*, *50*(3), 338–355.

Weber, R. (1992), On the Gittins index for multiarmed bandits, *The Annals of Applied Probability*, pp. 1024–1033.

Whittle, P. (1980), Multi-armed bandits and the Gittins index, *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 143–149.

Whittle, P. (1981), Arm-acquiring bandits, *The Annals of Probability*, pp. 284–292.

Whittle, P. (1988), Restless bandits: Activity allocation in a changing world, *Journal of Applied Probability*, pp. 287–298.

Xiao, Y., and M. van der Schaar (2012), Spectrum sharing and resource allocation-dynamic spectrum sharing among repeatedly interacting selfish users with imperfect monitoring, *IEEE Journal on Selected Areas in Communications*, *30*(10), 1890.