Sequential Decision Problems in Online Education

Ciara Pike-Burke, B.Sc.(Hons.), M.Res

Submitted for the Degree of Doctor of Philosophy at Lancaster University.

April 2019

STOR-i
excellence with impact

Abstract

This thesis is concerned with the study of sequential decision problems motivated by the challenge of selecting questions to give to students in an online educational environment. In online education there is the potential to develop personalized and adaptive learning environments, where students can receive individualized sequences of questions which update as the student is observed to be struggling or flourishing. In order to achieve this personalization, we must learn about how good each question is, while simultaneously giving students good questions. Multi-armed bandits are a popular technique for sequential decision making under uncertainty. Due to their online nature and their ability to balance the trade-off between exploitation and exploration, multi-armed bandits lend themselves naturally to this problem of adaptively selecting questions in education software. However, due to the complexity of the educational problem, standard approaches to multi-armed bandits cannot be applied directly. In this thesis variants of the multi-armed bandit problem specifically motivated by the issues arising in the educational domain are considered.

The first contribution is to consider the problem of selecting questions to give to a student in a homework task, where the homework task has a fixed length. Both the time it takes the student to answer each question and the benefit they gain from doing so are stochastic, and so we wish to develop an algorithm which adapts to the amount of time remaining in the homework task. This is an instance of the stochastic knapsack problem and so we develop a new approach for this problem when a generative model

of item sizes and rewards is available. This algorithm is an anytime algorithm based on the optimistic planning principle. We prove that with high probability our algorithm returns a near optimal policy and bound the number of samples necessary for this.

A further problem in education is that when a student answers a question, the benefit to their learning from doing so may not be evident immediately. Instead, the benefit may be delayed and, when we observe an improvement in their performance, it is often unclear exactly what the contribution of each individual question was to this improvement. Hence, in an educational domain the feedback from answering questions may be delayed, but also aggregated and anonymous. The second contribution of this thesis is the study of a variant of the stochastic multi-armed bandit problem with this form of delayed, aggregated anonymous feedback. For this problem, a rarely switching algorithm is presented which is able to learn from this kind of feedback and achieve almost the same performance as a state of the art algorithm for the simpler delayed feedback bandit problem, where observations are delayed but there is no anonymity.

One factor that will have a clear effect on the student's ability to answer a question correctly is the length of time since they have seen similar (or the same) questions. Consider, for example, the challenge of teaching students times tables in an app. In this case, the student's ability to recall the solution to one question will depend on how long it has been since they were last asked that question. We assume that the 'reward' to the student of answering a question is given by some function of the length of time it has been since they were last asked it, and we assume that this function is smooth enough to be modeled by a Gaussian process. We study a bandit problem where the expected reward of each arm is given by this unknown function of the time since the arm has been played. For this problem, we develop an algorithm which performs well experimentally, learning to play each arm when its reward is highest. Under the additional assumption that the noise is Gaussian, we also provide theoretical guarantees for the performance of this algorithm.

Acknowledgements

I have been fortunate to work with some great people during my PhD. In particular, I would like to thank my supervisors for all their help and advice. Especially, thank you to Dr Steffen Grünewälder for his patience, rigor and optimism. I am grateful to have had the chance to work with someone as knowledgeable and with as much enthusiasm for research as Steffen, and I have learned a lot from him. Thanks also Dr Anton Altmann for always giving me a different perspective on my problem during our Skype calls and for being able to solve whatever problem I was having with R. I would also like to thank Professor Jonathan Tawn and Professor David Leslie for their genuine support and encouragement (especially while I was writing up) and for the many chats about research, careers and life. Thanks also to Professor Csaba Szepesvári and Dr Shipra Agrawal for their help with the material in Chapter 5. Thank you to my examiners, Professor Nicolò Cesa-Bianchi and Dr Azadeh Khaleghi, for their helpful comments and suggestions for improving the final version of this thesis.

I am grateful that my PhD has been funded thanks to the support of EPSRC and Sparx. I would particularly like to thank Dr Mark Dixon for his role in creating the project and everyone at Sparx (and Oxygen House) for being so welcoming and making the time I spent in Exeter so rewarding.

Thanks to all the staff and students at STOR-i for making it such a fantastic and enjoyable place to work. Special thanks must go to the rest of my year, Aaron,

Chrissy, Dan, Emma, Emma, Jack, James, Lucy, Ollie, Rebecca and Stephen for their friendship, support and willingness to watch *Love Actually* every Christmas. Massive thanks also to Paul and Liz for filling so many evenings with laughter, advice and good food.

Then, I got to thinking, what would I do without the girls? Thanks to Alex, Alice, Charlie, Megan, Miriam and Sarah, and Emily, Natalia and Ellie for the numerous weekends away filled with laughter, love and wine. Thank you to Adam for your kindness, humor, and support, and for always lifting me up. Thank you also to Ronan for your love and understanding.

Finally, I would like to thank my parents, Noirin and Nigel, without whose love and support I wouldn't be where I am today. Thank you for making me believe that I could be anything I wanted to be, and achieve anything I set out to achieve.

Declaration

I declare that the work in this thesis has been done by myself and has not been submitted elsewhere for the award of any other degree.

Chapter 4 has been published as Pike-Burke, C. and Grünewälder, S. (2017). Optimistic Planning for the Stochastic Knapsack Problem. In *International Conference on Artificial Intelligence and Statistics*.

Chapter 5 has been published as Pike-Burke, C., Agrawal, S., Szepesvári, C. and Grünewälder, S. (2018). Bandits with Delayed, Aggregated Anonymous Feedback. In *International Conference on Machine Leaning*.

Chapter 6 has been submitted for publication as Pike-Burke, C. and Grünewälder, S. (2018). Recovering Bandits. An early version was presented at the European Workshop on Reinforcement Leaning (2018).

The word count for this thesis is 62,073.

Ciara Pike-Burke

Contents

Abstract I

Acknowledgements III

Declaration V

Contents VI

List of Figures XII

List of Tables XIV

1 Introduction 1

1.1 Contributions 4

2 Multi-Armed Bandits 7

2.1 Regret 8

2.2 Popular Algorithms 12

2.2.1 UCB 13

2.2.2 Thompson Sampling 17

2.2.3 Gittins Indicies 20

2.3 Extensions 22

2.3.1 Non-Stochastic Bandits 22

2.3.2 Linear Bandits . . . 25
2.3.3 Gaussian Process Bandits . . . 29
2.3.4 Delayed Feedback Bandits . . . 34
2.3.5 Non-Stationary Bandits . . . 38
2.3.6 Bandits with Knapsacks . . . 45
2.3.7 Optimistic Planning . . . 46

3 Motivating Problems from Education 51

3.1 Previous Work on Using Multi-Armed Bandits in Education . . . 52
3.2 Defining the Reward . . . 56
3.3 Fixed Limit on Homework Time . . . 59
3.4 Delay in the Effect of Answering Questions . . . 61
3.5 Allowing Time between Repetitions of a Question . . . 62

4 Optimistic Planning for the Stochastic Knapsack Problem 65

4.1 Introduction . . . 65
4.1.1 Related Work . . . 68
4.1.2 Our Contribution . . . 70
4.2 Problem Formulation . . . 70
4.2.1 Planning Trees and Policies . . . 71
4.3 High Confidence Bounds . . . 72
4.4 Algorithms . . . 74
4.4.1 Stochastic Optimistic Planning for Knapsacks . . . 74
4.4.2 Optimistic Stochastic Knapsacks . . . 75
4.5 ϵ-Critical Policies . . . 78
4.6 Analysis . . . 80
4.7 Experimental Results . . . 83
4.8 Conclusion . . . 85

4.A Supplementary Material . . . 85
4.A.1 Illustration of Policies . . . 85
4.A.2 Illustration of Bounds . . . 86
4.A.3 Algorithms . . . 86
4.B Proofs of Theoretical Results . . . 87
4.B.1 Bounding the Value of a Policy . . . 87
4.B.2 Theoretical Results for Optimistic Stochastic Knapsacks (`OpStoK`) 94

5 Bandits with Delayed, Aggregated Anonymous Feedback 102
5.1 Introduction . . . 102
5.1.1 Our Techniques and Results . . . 105
5.1.2 Related Work . . . 107
5.1.3 Organization . . . 107
5.2 Problem Definition . . . 108
5.3 Our Algorithm . . . 110
5.4 Regret Analysis . . . 114
5.4.1 Known and Bounded Expected Delay . . . 114
5.4.2 Delay with Bounded Support . . . 116
5.4.3 Delay with Bounded Variance . . . 119
5.5 Experimental Results . . . 120
5.6 Conclusion . . . 122
5.A Supplementary Material . . . 123
5.A.1 Table of Notation . . . 123
5.A.2 Beginning and End of Phases . . . 124
5.A.3 Useful Results . . . 125
5.B Results for Known and Bounded Expected Delay . . . 126
5.B.1 High Probability Bounds . . . 126
5.B.2 Regret Bounds . . . 141

5.C Results for Delays with Bounded Support . . . 148
5.C.1 High Probability Bounds . . . 148
5.C.2 Regret Bounds . . . 158
5.D Results for Delay with Known and Bounded Variance and Expectation 163
5.D.1 High Probability Bounds . . . 163
5.D.2 Regret Bounds . . . 177
5.E Additional Experimental Results . . . 181
5.E.1 Increasing the Expected Delay . . . 181
5.E.2 Comparison with Vernade et al. (2017) . . . 182
5.F Naive Approach for Bounded Delays . . . 184

6 Recovering Bandits 186

6.1 Introduction . . . 186
6.2 Related Work . . . 188
6.3 Problem Definition . . . 190
6.4 Defining the Regret . . . 192
6.4.1 Full Horizon Regret . . . 192
6.4.2 Instantaneous Regret . . . 193
6.4.3 d-step Lookahead Regret . . . 193
6.5 Baseline Approach . . . 195
6.6 Gaussian Process Recovery . . . 196
6.6.1 Single Play Lookahead Regret . . . 199
6.6.2 Multiple Play Lookahead Regret . . . 200
6.6.3 Instantaneous Algorithm . . . 201
6.6.4 Bounds on the Information Gain . . . 201
6.7 Improving Computational Efficiency via Optimistic Planning . . . 202
6.8 Experimental Results . . . 205
6.9 Conclusion . . . 207

6.A Preliminaries . . . 208
6.B Theoretical Results for *d*RGP-UCB . . . 214
6.B.1 Non-Repeating . . . 217
6.B.2 Repeating . . . 218
6.C Theoretical Results for *d*RGP-TS . . . 220
6.C.1 Non-Repeating . . . 220
6.C.2 Repeating . . . 221
6.D Regret Bounds for Non-Parametric Approach . . . 221
6.E Theoretical Guarantees on Optimistic Planning Procedure . . . 223
6.F Further Experimental Results . . . 227
6.F.1 Posterior Distributions and Covariates . . . 227
6.F.2 Values of Theta in Parametric Experiments . . . 234
6.F.3 Results for Different Lengthscales . . . 235

7 Conclusions **238**

7.1 Further Work . . . 241
7.1.1 Optimistic Planning for the Stochastic Knapsack Problem . . . 242
7.1.2 Bandits with Delayed, Aggregated Anonymous Feedback . . . 243
7.1.3 Recovering Bandits . . . 245
7.1.4 Bandit Problems in Online Education . . . 247

A Useful Results and Definitions **250**

A.1 Definitions . . . 250
A.2 Inequalities . . . 251
A.3 Markov Decision Processes . . . 252
A.4 Gaussian Processes and RKHS's . . . 253
A.4.1 Regression and Classification . . . 253
A.4.2 RKHS . . . 254

A.4.3 Covariance Functions 255

Bibliography **257**

List of Figures

3.1 An example of a forgetting curve. 63

4.1 The three possible cases of $E\Psi(B_\Pi)$. 82
4.2 Item sizes and rewards. 83
4.3 Number of policies vs value. 84
4.4 Examples of policies in the simple 3 item, 2 sizes stochastic knapsack problem. 85
4.5 Example of where just looking at the optimistic policy might fail. . . . 86

5.1 The relative difficulties of the different delayed feedback problems. . . . 103
5.2 An example of phase i of our algorithm. 113
5.3 The ratios of regret of variants of our algorithm to that of QPM-D for different delay distributions. 121
5.4 A detailed example of phase i of our algorithm. 125
5.5 The relative increase in regret of the different algorithms. 182
5.6 The ratios of regret of variants of our algorithm to that of DUCB for different delay distributions. 182

6.1 Examples of the recovery functions. 189
6.2 An example of a d-step lookahead tree. 196
6.3 The posterior recovery curves of all arms with observations indicated by crosses. 204

6.4 The total reward and final depth of the lookahead tree, d_N, as the policy budget, N, increases. 205

6.5 Cumulative instantaneous regret for parametric setup 207

6.6 dRGP-UCB with squared exponential kernel with $l = 0.5$. 228

6.7 dRGP-UCB with squared exponential kernel with $l = 2$. 229

6.8 dRGP-UCB with squared exponential kernel with $l = 5$. 230

6.9 dRGP-TS for squared exponential kernel with $l = 0.5$. 231

6.10 dRGP-TS for squared exponential kernel with $l = 2$. 232

6.11 dRGP-TS wit squared exponential kernel with $l = 5$. 233

6.12 Cumulative instantaneous regret for parametric setup with $l = 2.5$ . . . 237

6.13 Cumulative instantaneous regret for parametric setup with $l = 7.5$ . . . 237

List of Tables

6.1 Total reward at $T = 1000$ for single step experiments with parametric functions 205

6.2 θ values used in experiments with logistic recovery functions 234

6.3 θ values used in experiments with gamma recovery functions 235

6.4 Total reward at $T = 1000$ for single step experiments with parametric functions and $l = 2.5$ 236

6.5 Total reward at $T = 1000$ for single step experiments with parametric functions and $l = 7.5$ 236

Chapter 1

Introduction

The world of education is changing. With the development of the internet and smartphones, people across the world are increasingly able to access encyclopedias worth of knowledge from their pockets. This has dramatically changed how people learn and develop new skills. One example of this is the development of Massive Open Online Courses (MOOCs), e.g. EdX www.edx.org and coursera www.coursera.org, and other large online courses which mean that anyone can sign up for courses offered by top institutions and follow these courses online, using online quizzes to test their knowledge. On a smaller scale, there are now a multitude of educational games and apps available online where students can learn or consolidate skills while having fun. Even in a traditional education environment, teaching is now being aided by the use of online quizzes and tests, which allow teachers to track the performance of their students in realtime. All this contributes to a new, more online, way of learning.

One of the most exciting aspects of online education is the potential for personalizing learning. This means that each student can be given individual tasks specifically tailored to their strengths and weaknesses. The benefits of this would be enormous, struggling students would have the time to revise key concepts and learn at their own pace, whereas students who are excelling can be pushed further and their knowledge

deepened. Moreover, these online education systems also have the potential to adapt to how the student is getting on in a specific task, noticing right away if a student is struggling and taking direct action to help them. The challenge is how to achieve this. How do we decide which questions to give to the student when we do not know a priori how beneficial each one is? And how do we use the limited data we observe about the students to improve our future decisions?

Sequential decision models are a way of mathematically formalizing the concept of making a decision and using feedback on the outcome of that decision to inform future decisions. Within this area, algorithms for the multi-armed bandit problem will be particularly useful. The multi-armed bandit problem gets its name from the classical casino analogy of choosing which one armed bandit (slot machine) to play in order to maximize the payout, when the payout of each slot machine is stochastic with unknown expectation. In order to maximize their total winnings, a player must decide whether to explore their options, gathering more information about the slot machines, or exploit their current knowledge to select one which currently looks good. In recent years algorithms for multi-armed bandits have been developed and applied to settings such as online advertising, website optimization and recommendation systems to great success. One reason for this success is their ability to expertly and accurately balance the trade-off between wanting to explore and learn about the effectiveness of different actions and wanting to exploit the current knowledge and take the best action. This is similar to the challenge faced when trying to decide which questions to give to a student in an online education setting. However, the complexities of the educational domain mean that standard algorithms for the multi-armed bandit problem cannot directly be applied. The aim of this thesis is to investigate multi-armed bandit models inspired by the problem of selecting personalized questions in online education.

The particular problem this thesis is motivated by is how to select an adaptive sequence of questions to present to students in an online educational environment.

The work of this thesis has been carried out in collaboration with Sparx, an education research company. Since their foundation, Sparx have been gathering data on student performance in a series of online exercises accessed via their app. The app is incorporated into a traditional teaching environment and is designed to aid the teachers as well as the students. Once a teacher introduces a topic, the students will work through some exercises on the app, both in class and at home. As they do so, data will be obtained tracking their progress. The data consists of logging student interaction with the online platform and, as such, may be a lot more detailed than that gathered in a traditional schooling environment. Sparx's long term aim is to be able to use this extra information to improve students' experience and attainment. The aim of this particular work is to develop sequential decision making algorithms that are able to learn from this detailed feedback and suggest good questions to give to the students.

In this thesis, the focus will be on the statistical and mathematical foundations of multi-armed bandit algorithms motivated by this problem of suggesting questions to students in an online education environment. There are many challenges in the educational domain which make applying the standard algorithms for multi-armed bandits difficult. The three main challenges that have motivated the methodological work in this thesis are the following. Firstly, when we are setting homework tasks, there is a limit on the amount of time each homework can take, so we need to develop approaches that can handle this short horizon and adapt to the time remaining in the homework. Furthermore, when a student answers a question, the benefit they gain from doing so is not immediate, but instead is only observed as an aggregate sometime after answering the question. Lastly, the benefit to a student of answering a question will not be constant over time, it will most likely depend on how long it has been since they answered similar questions.

.

1.1 Contributions

This thesis studies sequential decision problems which are motivated by the challenge of selecting questions to give to students in an online education environment. In Chapter 2 we will introduce the multi-armed bandit problem and give an overview of some related work on algorithms and extensions to the classical problem. In Chapter 3 we will discuss existing work on using multi-armed bandits in online education domains and give further details of the specific problems in online education which have motivated the work in this thesis. The main contributions of this thesis are methodological developments in the field of multi-armed bandits. These will be presented in Chapters 4–Chapter 6. The work in each of these chapters has been submitted for publication as a standalone paper, and as such there may be some repetition of material. We outline below the main technical contributions of each of these chapters.

Chapter 4: Optimistic Planning for the Stochastic Knapsack Problem

The stochastic knapsack problem is a stochastic resource allocation problem that arises frequently and yet is exceptionally hard to solve. We derive and study an optimistic planning algorithm specifically designed for the stochastic knapsack problem. Unlike other optimistic planning algorithms for Markov Decision Processes (MDPs), our algorithm, `OpStoK`, avoids the use of discounting and is adaptive to the amount of resources available. We achieve this behavior by means of a concentration inequality that simultaneously applies to capacity and reward estimates. Crucially, we are able to guarantee that the aforementioned confidence regions hold collectively over all time steps by an application of Doob's inequality. We demonstrate that the method returns an ϵ-optimal solution to the stochastic knapsack problem with high probability. To the best of our knowledge, our algorithm is the first which provides such

guarantees for the stochastic knapsack problem. Furthermore, our algorithm is an anytime algorithm and will return a good solution even if stopped prematurely. This is particularly important given the difficulty of the problem. We also provide theoretical conditions to guarantee `OpStoK` does not expand all policies and demonstrate favorable performance in a simple experimental setting.

The work in this chapter appeared as: Pike-Burke, C. and Grünewälder, S. (2017). Optimistic Planning for the Stochastic Knapsack Problem. In *International Conference on Artificial Intelligence and Statistics*.

Chapter 5: Bandits with Delayed, Aggregated Anonymous Feedback

We study a variant of the stochastic K-armed bandit problem, which we call "bandits with delayed, aggregated anonymous feedback". In this problem, when the player pulls an arm, a reward is generated, however it is not immediately observed. Instead, at the end of each round the player observes only the sum of a number of previously generated rewards which happen to arrive in the given round. The rewards are stochastically delayed and due to the aggregated nature of the observations, the information of which arm led to a particular reward is lost. The question is what is the cost of the information loss due to this delayed, aggregated anonymous feedback? Previous works have studied bandits with stochastic, non-anonymous delays and found that the regret increases only by an additive factor relating to the expected delay. In Chapter 5, we show that this additive regret increase can be maintained in the harder delayed, aggregated anonymous feedback setting when the expected delay (or a bound on it) is known. We provide an algorithm that matches the worst case regret of the non-anonymous problem exactly when the delays are bounded, and up to logarithmic factors or an additive variance term for unbounded delays.

The work in this chapter appeared as: Pike-Burke, C., Agrawal, S., Szepesvári,

C. and Grünewälder, S. (2018). Bandits with Delayed, Aggregated Anonymous Feedback. In *International Conference on Machine Leaning*.

Chapter 6: Recovering Bandits

The recovering bandits problem is a variant of the non-stationary stochastic mutli-armed bandit problem designed to capture the effect of the time between plays on the reward of a given arm. In many scenarios such as product recommendation, the benefit of suggesting a product will depend on how long it has been since it was last suggested. This is captured in recovering bandits where, the expected reward of each arm changes depending on the time since the arm was last played according to some unknown *recovery function*. Under the assumption that the recovery functions are sampled from a Gaussian process, we present and analyze two algorithms for the recovering bandits problem. Furthermore, we show how their performance can be improved by allowing them to lookahead and select good sequences of actions. Finally, we demonstrate the experimental performance of our algorithms and present an approximation based on optimistic planning to improve computational efficiency at little cost to accuracy.

The work in this chapter is in submission. An early version was presented at the European Workshop on Reinforcement Learning (2018).

Chapter 2

Multi-Armed Bandits

The multi-armed bandit problem is a classical sequential decision problem that has been studied for many years (for example by Thompson (1933); Lai and Robbins (1985); Auer et al. (2002a); Gittins et al. (2011); Bubeck and Cesa-Bianchi (2012); Lattimore and Szepesvári (2018)). It gets its name from the fact that in its simplest form it can be expressed using an analogy to slot machines (or one-armed bandits) in a casino. Assume that a player is faced with a row of slot machines, or arms, and that each slot machine has a different probability distribution governing the payoff it generates when it is played. We call the payoff the player receives the *reward* from playing an arm. All the reward distributions are unknown to the player, whose aim is to play the slot machines that will maximize the total reward. The player's task is then to choose between playing arms that they already know produce a high reward, or trying alternative arms about which they have less information, but whose reward could be high. The player must therefore decide how to trade-off between *exploiting* good arms to maximize their immediate reward and *exploring* other arms to gather information about their performance in order to potentially improve future rewards.

Formally, we define the stochastic K-armed bandit as follows. We assume that there are K arms (or actions, the two terms will be used interchangeably in this the-

sis) in a set $\mathcal{A}$, and associated with every arm $1 \leq j \leq K$ is an underlying reward distribution ν_j. Whenever an arm j is played a stochastic reward is generated independently from the underlying distribution ν_j and presented to the player. The multi-armed bandit problem proceeds in rounds and, in each round, the player selects an arm and then receives a reward from the underlying reward distribution of that arm. The player can then use the previously observed rewards to inform future decisions about which arms to play. We define the horizon, T, as the total number of plays of the bandit game. The game can be summarized in the following sequence. Beginning with an empty history, $\mathcal{H}_0 = \emptyset$, at each time step $t = 1, \ldots, T$, the player,

1. Selects an arm $J_t \in \{1, \ldots, K\}$, possibly using the history $\mathcal{H}_{t-1}$
2. Receives an observation $X_{t,J_t} \sim \nu_{J_t}$
3. Adds the pair $\{J_t, X_{t,J_t}\}$ to the history, $\mathcal{H}_t = \mathcal{H}_{t-1} \cup \{J_t, X_{t,J_t}\}$.

The player's aim is to select the actions that will maximize their total reward over T time steps.

It is typically necessary to make some assumptions about the reward distributions, ν_j, in order to construct a tractable algorithm for the multi-armed bandit problem. Generally, it is assumed that the rewards of all arms are independent and that all arms $j = 1, \ldots, K$ have a finite expectations $\mu_1, \ldots, \mu_K$ (so $\mathbb{E}[X_{1,j}] = \mu_j$ for $X_{1,j} \sim \nu_j$). In some cases, it is assumed that the reward distribution is from a particular family of distributions. Otherwise, it is assumed that the reward distributions are (λ-)*sub-Gaussian* (see Appendix A.1) or bounded, often in $[0, 1]$.

2.1 Regret

In the multi-armed bandit problem, the aim is usually to select arms to play such that the cumulative reward over the T rounds is maximized. Traditionally, a discount factor

was used and the aim was to maximize the expected discounted reward. However, recently an alternative performance measure has been considered. In particular, the performance of an algorithm for the multi-armed bandit problem is typically measured in terms of its *(cumulative) regret*. The cumulative regret up to horizon T, $\mathfrak{R}_T$, is the total difference in the reward that could have been obtained by repeatedly playing the optimal arm and the reward that was actually obtained by the arms played. We will mostly be interested in the *expected* regret of an algorithm, where the expectation is taken over the actions taken (note that the actions may be random variables since they can depend on the past observations). Specifically, let $\mu^* = \max_{1 \leq j \leq K} \mu_j$ be the maximum expected reward of any arm. Clearly the best possible algorithm will constantly play this arm for all T rounds. We define the regret as the difference in expected cumulative reward between this oracle and the arms $J_1, \ldots, J_T$ chosen by the algorithm. In particular, we define the expected cumulative regret up to horizon T as,

$$\mathbb{E}[\mathfrak{R}_T] = \sum_{t=1}^{T} \mathbb{E}[\mu^* - \mu_{J_t}].$$

The aim of a bandit algorithm is to select arms J_t in order to minimize this expected regret. Note that this is essentially equivalent to maximizing the expected cumulative reward of the algorithm.

It can be difficult to calculate the regret of a bandit algorithm exactly. It can be estimated through simulations, but it is often useful to have theoretical guarantees on the performance of an algorithm. In some problem instances, it is possible to obtain lower bounds on the regret that must be incurred by any bandit algorithm in that setting. It is also commonplace to provide an upper bound on the expected regret of a particular algorithm. If the upper regret bound of an algorithm matches the lower bound, we say that the algorithm is optimal for this particular setting. When considering these theoretical regret bounds, there are two main types of regret that it is useful to look at, the *problem dependent* regret and the *problem independent* regret.

The *problem dependent* regret of a bandit algorithm depends on the specifics of the problem instance we are considering. For a particular set $\mathcal{A}$ of actions numbered 1 to K, the problem dependent regret will typically depend on the means, μ_j, of the actions. For any arm $j \neq j^*$, let $\Delta_j = \mu^* - \mu_j$ be the *sub-optimality gap* of arm j and for any arm j, and let $N_j(T)$ be the random number of times arm j has been played up to horizon T. Then the expected regret can be expressed as,

$$\mathbb{E}[\mathfrak{R}_T] = \sum_{j=1;j\neq j^*}^{K} \Delta_j \mathbb{E}[N_j(T)]$$

Using this in their seminal paper, Lai and Robbins (1985) proved the following lower bound on the problem dependent regret of any bandit algorithm. Specifically, under some mild assumptions on the reward distributions (see (Lai and Robbins, 1985) for details), it was shown that the regret of any algorithm for the multi-armed bandit problem must satisfy,

$$\liminf_{T\to\infty} \frac{\mathbb{E}[\mathfrak{R}_T]}{\log T} \geq \sum_{j=1;j\neq j^*}^{K} \frac{\Delta_j}{KL(\nu_j, \nu_{j^*})} \tag{2.1}$$

where $KL(\nu_j, \nu_{j^*})$ represents the Kullback-Leibler divergence between the reward distribution of arm j and that of the optimal arm (see Appendix A.1). This means that in this problem setting, no algorithm can achieve a smaller problem dependent rate of regret. Hence, the aim is often to construct algorithms that can achieve problem dependent regret of this order.

Sometimes it is not desirable to bound the regret of a bandit algorithm for a particular problem instance. In practice, the expected rewards, μ_j's, of each arm are not known before we start playing the bandit game, and so we may wish to have regret bounds which hold regardless of the specific problem setup and arm distributions. In this case, it is useful to consider the *problem independent* or *worst case regret* which holds for any problem instance. Auer et al. (2002b) provide the following lower bound

on the problem independent regret of any bandit algorithm. Particularly, for any algorithm, there exists a problem instance where

$$\mathbb{E}[\mathfrak{R}_T] \geq \frac{1}{20}\min\{\sqrt{KT}, T\}. \tag{2.2}$$

This is a bound on the regret of the algorithm in the worst possible case (in fact, it is derived for the adversarial bandit problem, see Section 2.3.1) and so it is natural that it is larger than the problem dependent regret bound.

The above definitions of regret have all assumed a *frequentist* representation of the problem. In some cases it is desirable to take a Bayesian approach (see e.g. Gelman et al. (2013) for an introduction to Bayesian reasoning). Here, any parameters of the reward distribution are assumed to be random variables and a prior distribution is placed on them. This induces a prior distribution over the μ_j's for all arms $1 \leq j \leq K$. In this case, the regret definition changes and we consider the *Bayesian regret*. In the Bayesian regret, the expectation is taken over this prior distribution as well as the arms selected. Hence, we define the cumulative Bayesian regret up to horizon T as,

$$\mathbb{E}[\mathfrak{R}_T^B] = \sum_{t=1}^{T} \mathbb{E}[\mu^* - \mu_{J_t}] = \sum_{t=1}^{T} \mathbb{E}[\mathbb{E}[\mu^* - \mu_{J_t} | \mu_1 \dots \mu_K]].$$

Bubeck and Liu (2013) state that the lower bound of (2.2) also holds for Bayesian regret when the rewards are in $[0, 1]$. This means that in the Bayesian setting we we can always find a prior distribution such that the Bayesian regret satisfies $\mathbb{E}[\mathfrak{R}_T^B] = \Omega(\sqrt{KT})$. In a slightly different setting where the rewards are discounted, it is possible to design algorithms which asymptotically match the expected cumulative discounted reward (where the expectation is taken with respect to the prior as well) of the optimal strategy (see Section 2.2.3 for details).

Given the above lower bounds on the regret, we have some idea of how well an algorithm for the stochastic K-armed bandit problem can be expected to perform.

The challenge is therefore to develop algorithms for the multi-armed bandit problem that achieve these rates of regret. Furthermore, it is desirable to develop algorithms which exhibit strong *finite time* regret as well as asymptotically having low regret. Finite time regret is the regret up to a fixed horizon T and is more informative about the real life performance of the algorithm. In the following section (Section 2.2), we detail some of the key algorithmic developments that have lead to (near) optimal algorithms for the stochastic multi-armed bandit problem.

2.2 Popular Algorithms

The multi-armed bandit problem has been formally studied at least since the seminal paper of Thompson (1933). Since then, research into the problem has expanded in various directions. One of the most popular lines of work is into Upper Confidence Bound (or UCB) style algorithms, inspired by Lai and Robbins (1985). These have received a resurgence of interest in recent years following the work of Auer et al. (2002a) where finite time theoretical regret bounds and experimental results were given. During this resurgence, interest also returned to the original algorithm of Thompson (1933), now known as Thompson Sampling, and regret bounds and strong experimental results have also been demonstrated for this approach. A different line of work follows that by Gittins (1979) in defining optimal index policies for the discounted Markovian problem. In this section, we will review some of the key developments in these three lines of work. Note that in this thesis, focus has been on developing UCB and Thompson Sampling style algorithms for variants of the multi-armed bandit problem, so emphasis will be placed on these approaches.

2.2.1 UCB

The use of optimistic estimates or Upper Confidence Bounds (UCBs) for the multi-armed bandit problem stems from the seminal work of Lai and Robbins (1985). Intuitively, the idea behind the upper confidence bound approach is to use an optimistic estimate of the expected reward of each arm given the information available. Then, playing the arm with the largest optimistic estimate will lead to selecting arms which either have high reward or are poorly estimated, in which case they are worth exploring more. The confidence bounds presented in (Lai and Robbins, 1985) relied on the entire history of rewards of all arms and as such were difficult to compute. A simpler algorithm was proposed by Agrawal (1995) who provided an asymptotic analysis. This was later adapted by Auer et al. (2002a), who provided a finite time analysis of the regret of this algorithm, and several other related algorithms.

The UCB1 algorithm from (Auer et al., 2002a) constructs upper confidence bounds around the sample mean of the reward of each arm in a way that guarantees that the true mean of the arm is less than the upper confidence bound with high probability. These are constructed using Hoeffding's inequality (see Appendix A.2) and hold for any reward distribution bounded in $[0, 1]$. For any arm j which has been played n_j times, let $\bar{X}_j$ be the sample mean of these n_j observations, then, with probability greater that $1 - \delta$,

$$\mu_j \leq \bar{X}_j + \sqrt{\frac{\log(1/\delta)}{2n_j}}.$$

This is used in the construction of the upper confidence bounds of the UCB1 algorithm. In particular, the UCB1 algorithm proceeds by playing each arm once to guarantee that the initial sample means exist, and then at each time step $t = K+1, \ldots, T$, it selects arm,

$$J_t = \underset{1 \leq j \leq K}{\arg\max} \left\{ \bar{X}_{j,t} + \sqrt{\frac{\log(t)}{2N_j(t)}} \right\}$$

where $N_j(t)$ is the number of times arm j has been played in t rounds of the bandit

game and $\bar{X}_{j,t} = \frac{1}{N_j(t)} \sum_{s=1}^{t} X_{s,J_s} \mathbb{I}\{J_s = j\}$ is the sample mean of these observations. Note that the only knowledge the UCB1 algorithm has about the reward distribution is that the arms are independent and the rewards are bounded in $[0, 1]$.

Auer et al. (2002a) showed that the problem dependent regret of UCB1 satisfies,

$$\mathbb{E}[\mathfrak{R}_T] \leq 8 \sum_{j=1}^{K} \frac{\log(T)}{\Delta_j} + (1 + \pi^2/3) \sum_{j=1}^{K} \Delta_j.$$

This has the same $\log(T)$ dependence on the horizon T as seen in the lower bound (2.1). Moreover, for Normal distributions with means μ^* and μ_j and unit variances, the KL divergence simplifies to Δ_j^2 in which case the lower bound in (2.1) is matched exactly by the dominant term of this regret bound. However, for alternative reward distributions, $KL(\nu_j, \nu_j^*) \neq \Delta_j^2$ and so this upper bound does not match the lower bound in (2.1) exactly. The proof of this regret bound relies on bounding $\mathbb{E}[N_j(T)]$, the expected number of times any sub-optimal arm is played by the algorithm. It can be shown that if $N_j(T) > 8\frac{\log(T)}{\Delta_j^2}$, then the confidence term for arm j is smaller than $\Delta_j/2$, and so the only way arm j can be played again is if the confidence bounds on arm j or the optimal arm fail. By Hoeffding's inequality, this happens with low probability. Hence the main contribution of arm j to the regret is from these first plays when the algorithm is learning about the arm and this is bounded by $\Delta_j 8\frac{\log(T)}{\Delta_j^2}$. This gives the result. From this problem dependent regret bound, it is easy to show that the problem independent regret of UCB1 satisfies,

$$\mathbb{E}[\mathfrak{R}_T] \leq O(\sqrt{KT \log(T)}). \tag{2.3}$$

This matches the order of the lower bound in (2.2) up to a $\sqrt{\log(T)}$ term. This problem independent regret bound was obtained from the problem dependent regret bound via a standard worst case analysis. This consists of separating the arms into those with $\Delta_j < \Delta$ and those with $\Delta_j \geq \Delta$ for some fixed $\Delta > 0$ and optimizing

the problem dependent regret to find the worst case value of Δ (see for example, (Lattimore and Szepesvári, 2018)). This gives the problem independent regret bound. This value of Δ represents the sub-optimality gap that is hardest for the algorithm to deal with.

While UCB1 is a simple and intuitive algorithm that enjoys theoretical guarantees on the regret that almost match the lower bounds in (2.1) and (2.2), considerable effort has been invested in constructing UCB approaches for the multi-armed bandit problem which have tighter regret bounds. One of the most important of these, the KL-UCB algorithm of Cappé et al. (2013), aims to recover the KL divergence term in the denominator of (2.1) and thus focuses on problem dependent regret. For a one parameter exponential family reward distribution which can be parameterized by the mean, their approach is to to construct a pseudo upper confidence bound for each arm by selecting the parameter that will maximize the expected reward while still being close to the sample mean in KL distance. Specifically, let $d(\mu, \mu')$ be the KL-divergence between the particular exponential family distribution of interest when the mean parameters are $\mu \in \Theta$ and $\mu' \in \Theta$ (and Θ is the parameter set). In the KL-UCB algorithm, each arm is played once to begin with, then at time $t = K+1, \ldots, T$, we play arm

$$J_t = \underset{1 \leq j \leq K}{\arg\max} \sup \left\{ \mu \in \Theta; d(\bar{X}_{j,t}, \mu) \leq \frac{\log(t) + 3\log(\log(t))}{N_j(t)} \right\}.$$

Cappé et al. (2013) show that the regret of this algorithm satisfies,

$$\mathbb{E}[\mathfrak{R}_T] \leq \sum_{j=1}^{K} \Delta_j \frac{\log(T)}{d(\mu_j, \mu^*)}(1 + o(1)).$$

Hence, the problem dependent regret of KL-UCB matches the lower bound in (2.1) up to lower order terms. Note that using Pinsker's inequality to bound $d(\mu_j, \mu^*) \geq \frac{1}{2}(\mu_j - \mu^*)^2 = \frac{1}{2}\Delta_j^2$ and using the standard worst case analysis results in a problem

independent regret bound of $O(\sqrt{KT\log(T)})$ for KL-UCB, the same as UCB1. Cappé et al. (2013) also provide a version of the algorithm for distributions which are not one parameter exponential family. However, note that, in all cases, in order to calculate the selection criteria, it is necessary to be able to calculate the KL-divergence and this requires knowledge of the reward distributions, which is not required for UCB1.

The problem independent regret of UCB1 in (2.3) suffers from an additional $\sqrt{\log(T)}$ term compared to the lower bound in (2.2). There have been several approaches designed to remove this term. The first, the so-called Improved-UCB algorithm of Auer and Ortner (2010), is an example of a rarely switching algorithm. It runs in phases and in each phase it plays each active arm consecutively and then, at the end of a phase, an active arm is eliminated if it is clearly suboptimal. Specifically, in every phase i, the algorithm maintains a tolerance gap $\tilde{\Delta}_i$ and plays each active arm until the total number of times it has been played is $n_i = \left\lceil \frac{2\log(T\tilde{\Delta}_i^2)}{\tilde{\Delta}_i^2} \right\rceil$. Then an arm is eliminated if its estimated mean reward is further than $\tilde{\Delta}_i$ from the best estimated mean reward, and the tolerance gap is reduced, $\tilde{\Delta}_{i+1} = \tilde{\Delta}_i/2$. The regret analysis of this algorithm again uses Hoeffding's inequality (but this time to get confidence bounds that hold with probability greater than $1 - \frac{1}{T\tilde{\Delta}_i^2}$ in each phase i) to bound the probability of erroneously eliminating arms. This leads to problem dependent regret $\mathbb{E}[\mathfrak{R}_T] \leq \sum_{j=1}^{K} \frac{C\log(T\Delta_j^2)}{\Delta_j}$ for some constant $C > 1$. This corresponds to problem independent regret of

$$\mathbb{E}[\mathfrak{R}_T] \leq O(\sqrt{KT\log(K)}).$$

This is an improvement over the worst case regret of UCB1 by replacing $\sqrt{\log(T)}$ with $\sqrt{\log(K)}$. However, it is still loose by $\sqrt{\log(K)}$.

Another approach that improves on the problem independent regret of UCB1, is the MOSS algorithm of Audibert and Bubeck (2009). This algorithm is more similar to the UCB approach since it plays each arm once then at time $t = K+1, \ldots, T$, it

plays arm

$$J_t = \underset{1 \leq j \leq K}{\arg\max} \left\{ \bar{X}_{j,t} + \sqrt{\frac{\max\{\log(\frac{T}{KN_j(t)}), 0\}}{N_j(t)}} \right\}.$$

This leads to problem dependent regret of $\mathbb{E}[\mathfrak{R}_T] \leq CK \sum_{j=1}^{K} \frac{\max\{\log(T\Delta_j^2/K), 1\}}{\Delta_j}$ which corresponds to a problem independent regret bound of

$$\mathbb{E}[\mathfrak{R}_T] \leq O(\sqrt{KT}),$$

matching the optimal rate indicated by (2.2).

2.2.2 Thompson Sampling

One of the earliest instances of the multi-armed bandit problem appeared in (Thompson, 1933) in the context of clinical trials. The proposed algorithm, now known as Thompson sampling, is very popular due to its intuitiveness, ease of implementation and strong experimental performance. It is a Bayesian approach and so begins with placing a prior distribution over the parameters of the reward distribution of each arm. Let θ_j be the parameters of the reward distribution over arm j and let $r(\theta)$ give the expected reward as a function the parameters θ (which is common across all arms). Let $\pi(\theta_j)$ be the prior placed on the parameters of arm j. Then, under the assumption that the family of reward distributions is known, the posterior distribution of the parameters can be obtained by using the likelihood of the observations of arm j (see (Gelman et al., 2013) for further details). For any arm j and time t, let $\pi_t(\theta_j)$ denote the posterior distribution of θ_j at time t, using the $N_j(t)$ previously observed samples from the reward distribution of arm j. Then the Thompson sampling algorithm proceeds as follows. At time t,

1. For all arms $1 \leq j \leq K$, sample $\tilde{\theta}_j \sim \pi_t(\theta_j)$
2. Play arm $J_t = \arg\max_{1 \leq j \leq K} r(\tilde{\theta}_j)$.

Note that since we have a prior distribution over each θ_j, we do not need an initialization step as in the UCB approaches. It can be shown that at time t, the above procedure is equivalent to playing each arm with the posterior probability that it is optimal. If the reward distributions admit a conjugate prior (e.g. exponential family distributions), the posterior distributions of the reward parameters can be easily computed. If not, alternative methods such as MCMC (see e.g. (Gilks et al., 1995)) may need to be used in order to obtain samples from the posterior.

The strong empirical performance of Thompson sampling was demonstrated in (May and Leslie, 2011; Chapelle and Li, 2011) where it was shown to outperform the UCB approach in various experiments. May et al. (2012) proved asymptotic results on the performance of Thompson Sampling for general reward distributions. Theoretical regret bounds for Thompson sampling were given in (Agrawal and Goyal, 2012) and (Kaufmann et al., 2012b). These results consider the Beta-Bernoulli bandit problem, where the prior on the expected reward of each arm is a Beta distribution and the observations of each arm are Bernoulli, leading to a conjugate Beta posterior. However, as discussed in (Agrawal and Goyal, 2012), this can be extended to other reward distributions taking values in $[0, 1]$ by first observing the random reward $X_{j,t}$ and then performing a Bernoulli trial with success probability $X_{j,t}$ and updating the posterior distribution of a pseudo parameter θ_j for these Bernoulli trials for arm j.

Kaufmann et al. (2012b) mainly considered the problem dependent regret and proved that for any $\epsilon > 0$, the expected regret of Thompson Sampling satisfies,

$$\mathbb{E}[\mathfrak{R}_T] \leq (1+\epsilon) \sum_{j=1}^{K} \frac{\Delta_j (\log(T) \log(\log(T)))}{KL(\mu_j, \mu^*)} + C(\epsilon, \boldsymbol{\mu}).$$

where C is a constant depending on ϵ and $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_K)$. This almost matches the lower bound in (2.1). Their proof technique is similar to that of UCB1 and KL-UCB and draws on properties of the Bernoulli distribution. Agrawal and Goyal (2013a)

prove a slightly different problem dependent regret bound and also show that the problem independent regret of Thompson Sampling satisfies,

$$\mathbb{E}[\mathfrak{R}_T] \leq O(\sqrt{KT\log(T)})$$

for the Beta-Bernoulli bandit problem. This is the same problem independent regret rate as UCB1 (Auer et al., 2002a) and has an additional $\sqrt{\log(T)}$ term compared to the lower bound in (2.2). These results for Beta-Bernoulli bandits were extended in (Korda et al., 2013) to cover reward distributions in the one-parameter exponential family and in (Agrawal and Goyal, 2013a) to consider Gaussian distributions, and similar results were shown.

All the above theoretical results focused on the frequentist regret (where the mean rewards are fixed and the expectation is only taken over the arms chosen). However, since Thompson sampling is a Bayesian procedure, is also makes sense to consider the Bayesian regret. Theoretical regret guarantees on the Bayesian regret of Thompson Sampling were obtained by Russo and Van Roy (2014). Here, they were able to relate the Bayesian regret of Thompson sampling to that of a UCB approach, and using results on the performance of various UCB strategies, they obtained Bayesian regret bounds for a wide variety of different settings. For finitely many arms, they show that the Bayesian regret is $O(\sqrt{KT\log(T)})$. This was improved in (Bubeck and Liu, 2013) where it was shown that the Bayesian regret of Thompson sampling is $O(\sqrt{KT})$.

There have also been variations of the Thompson Sampling algorithm considered in the literature. The Optimistic Bayesian Sampling algorithm of May et al. (2012) aims to combine Thompson sampling with the optimistic principle underpinning the UCB strategies. Here, at each time step, t, after sampling $\tilde{\theta}_j$ from the posterior of each arm, if the sampled value is less that the posterior mean, then the sampled value is replaced by the posterior mean, and this is used to select which arm to play. Instead of considering the regret of this algorithm, they show that, asymptotically, the ratio

of the sum of rewards from their algorithm to the sum of optimal reward will tend to 1. Another approach aimed at combining Bayesian and optimistic strategies is the Bayes UCB algorithm of Kaufmann et al. (2012a). This approach uses the quantiles of the posterior distribution as upper confidence bounds and then proceeds as a standard UCB algorithm. They show that this approach achieves problem dependent frequentist regret for Bernoulli rewards of

$$\mathbb{E}[\mathfrak{R}_T] \leq \sum_{j=1}^{K} \left(\frac{(1+\epsilon)\Delta_j}{KL(\mu_j, \mu_j^*)} \log(T) + c_\epsilon(\log(T)) \right)$$

for $\epsilon > 0$ where c_ϵ is some constant depending on ϵ. They also show that, for Bernoulli rewards, the index they use is equivalent to the index used in the KL-UCB algorithm.

2.2.3 Gittins Indicies

A different Bayesian approach to the multi-armed bandit problem is the Gittins Index approach (Gittins et al., 2011; Gittins, 1979). In this framework, the multi-armed bandit problem is represented as a (semi) Markov decision process (see Appendix A.3) where the state is the current posterior distribution over the reward parameters of each arm and the actions are the set of arms in the bandit problem. It is assumed that the state of each arm evolves according to an independent Markov chain with transition density D. At each time t, the player observes the states of each arm j, $S_j(t)$, and selects an action. If the action chosen at time t is arm J_t, the player receives a reward $r(S_{J_t}(t))$ and then the states are updated so that $S_j(t+1) = S_j(t)$ for all $j \neq J_t$ and $S_{J_t}(t+1) = D(S_{J_t}(t))$.

In this setting, the aim is normally to maximize the expected total discounted reward, $\mathbb{E}\left[\sum_{t=1}^{T} \gamma^t r(S_{J_t}(t))\right]$, where $\gamma \in (0,1)$ is the discount factor and the expectation is taken over the prior distribution of each arm as well. This is the same as in a Markov Decision Process (MDP) (see Appendix A.3). In a MDP, it has been shown

that dynamic programming will find the optimal policy (Bellman, 1956). However, in a MDP representation of the bandit problem, the state space is very large and so such a dynamic programming solution would not be computationally tractable. Despite this, Gittins (1979) (see (Gittins et al., 2011) for alternative proofs) showed that there exists an optimal index style policy that defines an index for each arm independently. Playing arms with the largest such index at each time step maximizes the total expected discounted reward of a policy. This is the so-called *Gittins index* policy.

The Gittins index policy consists of playing the arm from a given state with highest Gittins index. The Gittins index of arm j from initial state s is defined as,

$$G(j,s) = \sup_{\tau>0} \frac{\mathbb{E}\left[\sum_{t=0}^{\tau-1} \gamma^t r(S_j(t)) \middle| S_j(0) = s\right]}{\mathbb{E}\left[\sum_{t=0}^{\tau-1} \gamma^t \middle| S_j(0) = s\right]},$$

where $S_j(t) = D(S_j(t-1))$. Note that this can be intuitively interpreted for each arm as the largest cost the player is willing to pay to receive at least one more reward from that process (Lattimore and Szepesvári, 2018). It is then shown that playing arm $J_t = \arg\max_{1\leq j\leq K} G(j, S_j(t))$ at time t maximizes the cumulative discounted expected reward, $\mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^t r(S_{J_t}(t))\right]$. Finite time regret guarantees of the Gittins index policy were provided by Lattimore (2016).

The Gittins indicies policy is an index approach and only requires using the information about one arm at a time to compute its index, thus reducing the computational complexity compared to dynamic programming. However, for most reasonably sized problems, and particularly those involving extensions of the standard multi-armed bandit problem as will be considered in this thesis, it is still computationally infeasible to compute the Gittins index in reasonable time. It is also not clear how to adapt the MDP representation and Gittins index strategy to these more complex

problems. Furthermore, for most of the problems studied here, using Gittins indicies would involve including an artificial discount factor which may not be appropriate.

2.3 Extensions

The multi-armed bandit problem as presented above is an interesting problem for which several algorithms have been developed. However, it can be argued that in its present form it is too simple for many applications. Consequently, there have been numerous extensions to the standard stochastic K-armed bandit problem to make it more appropriate in many practical applications. We detail here some of these extensions which are most relevant to the work in this thesis.

2.3.1 Non-Stochastic Bandits

Another version of the bandit problem that has been studied in the literature is the *non-stochastic* or *adversarial* bandit problem. This problem was first introduced by Auer et al. (1995) and removes several of the assumptions underlying the stochastic multi-armed bandit problem. Specifically, in the adversarial bandits problem, it is no longer assumed that the rewards are sampled from an underlying reward distribution, nor that they are even random variables. Instead, they are assumed to lie in $[0, 1]$ and to be generated by an 'adversary'. At each time t, the adversary selects a vector $\mathbf{x}_t = (x_{1,t}, \ldots, x_{K,t})^T$ of rewards in $[0, 1]^K$. Then, if the player plays arm J_t at time t, they will receive reward $x_{J_t,t}$. An important point is that the adversary may have knowledge of the players strategy and could select the reward vectors accordingly. This means that if the player plays according to some deterministic strategy, the adversary will be able to make them suffer a lot by adapting to their strategy. However, if the player's strategy involves an element of randomness, there is less that the adversary can do to harm them (Lattimore and Szepesvári, 2018).

The performance of an algorithm for the adversarial bandits problem is again measured in terms of its regret. However, the definition of regret here differs slightly from in the stochastic bandit problem. Since the adversary choses a vector of rewards at each time step, the best arm is constantly changing, however, the regret is defined with respect to the best *constant* arm, that is $\arg\max_{1\leq j\leq K}\sum_{t=1}^{T} x_{j,t}$. Note that by Arora et al. (2012) this is more appropriate since sub-linear regret with respect to the best sequence of actions is not possible. We are also often interested in the worst case regret which is taken over all possible action choices of the adversary. Hence, in the adversarial K-armed bandit problem, the regret is defined as,

$$\mathbb{E}[\mathfrak{R}_T^a] = \max_{\mathbf{x}_1,\ldots,\mathbf{x}_T}\left\{\max_{1\leq j\leq K}\sum_{t=1}^{T} x_{j,t} - \mathbb{E}\left[\sum_{t=1}^{T} x_{J_t,t}\right]\right\},$$

where the expectation is taken over the potential randomness of the choice of actions. The worst case lower bound on the regret in (2.2) was proved for the adversarial bandits problem, and so also holds in this case.

As mentioned previously, randomized strategies will generally outperform deterministic ones in the adversarial bandits problem. Interestingly, this means that often algorithms for the stochastic bandits problem perform poorly in the adversarial problem. Hence specific strategies have been developed for the adversarial bandits problem. The first such algorithm to be proposed was the EXP3 algorithm of Auer et al. (1995), which is based on the Hedge algorithm of Freund and Schapire (1997). At each time step t, EXP3 plays arm j with probability $P_{j,t}$. These probabilities are calculated using exponential weighting of importance sampling estimators of each arm's reward. Specifically, let $\hat{S}_{j,t} = \sum_{n=1}^{t} \frac{\mathbb{I}\{J_n=j\}x_{j,n}}{P_{j,n}}$ for all arms j, then at time t, arm j is played with probability,

$$P_{j,t} = (1-\eta)\frac{\exp\{\eta\hat{S}_{j,t}/K\}}{\sum_{j=1}^{K}\exp\{\eta\hat{S}_{j,t}/K\}} + \frac{\eta}{K},$$

where η is a tuning parameter used to control how quickly we want to stop exploring. When $\eta = \sqrt{\frac{K\log(K)}{(e-1)T}}$, the regret of EXP3 is bounded by,

$$\mathbb{E}[\mathfrak{R}_T^a] \leq 2\sqrt{(e-1)KT\log(K)}.$$

This matches the lower bound in (2.2) up to logarithmic factors.

There have also been numerous variants of the EXP3 algorithm and different approaches proposed in the literature. These include algorithms designed to remove the logarithmic terms (Audibert and Bubeck, 2009), algorithms with high probability regret guarantees (Neu, 2015), or general algorithms for both stochastic and adversarial bandits (Auer and Chiang, 2016; Seldin and Lugosi, 2017) among others. Of particular relevance to us is the 'bandits with expert advice' problem and the EXP4 algorithm of Auer et al. (1995). In the bandits with expert advice problem, at each time t, the player is presented with N probability vectors over the arms, each representing a different expert's opinion of which arm to play. The player can then use this expert advice to influence their choice of which action to take. Here, the regret is defined with respect to the expected reward of the best expert. Formally, if the beliefs of experts $i = 1, \ldots, N$ at time $t = 1, \ldots, T$ are represented by the probability vectors $\boldsymbol{\xi}_1(t), \ldots, \boldsymbol{\xi}_N(t)$ and $\mathbf{x}_t$ is the vector of the rewards of each arm, then the regret is defined as

$$\mathbb{E}[\mathfrak{R}_T^e] = \max_{1\leq i\leq N} \sum_{t=1}^{T} \boldsymbol{\xi}_i(t)^T \mathbf{x}_t - \mathbb{E}\left[\sum_{t=1}^{T} x_{J_t,t}\right]\Bigg\},$$

when the player plays actions $J_1, \ldots, J_T$. The EXP4 algorithm of Auer et al. (1995) is a modification of EXP3 to this setting. For some tuning parameter $\gamma \in (0, 1]$, at each time step $t = 1, \ldots, T$, the learner recieves the expert advice vectors and then plays arm j with probability

$$P_{j,t} = (1-\gamma)\frac{\sum_{i=1}^{N} w_i(t)\xi_i^{(j)}(t)}{\sum_{i=1}^{N} w_i(t)} + \frac{\gamma}{K},$$

where $w_i(t)$ is the weight given to expert i at time t and is defined iteratively by $w_i(1) = 1$, $w_i(t+1) = w_i(t)\exp(\frac{\gamma \boldsymbol{\xi}_i(t)^T \mathbf{y_i}(t)}{K})$ where $y_i^{(j)} = \mathbb{I}\{J_t = j\}\frac{x_{j,t}}{P_{j,t}}$ and superscript (j) is used to denote the jth element of a vector. Under the assumption that the family of experts contains the uniform expert, Auer et al. (1995) proved that the regret of the EXP4 algorithm in the bandits with experts problem is bounded by $\mathbb{E}[\mathfrak{R}_T^e] \leq (e-1)\gamma \max_{1\leq i\leq N} \sum_{t=1}^T \boldsymbol{\xi}_i(t)^T \mathbf{x}_t + \frac{K\log(N)}{\gamma}$.

Since the adversarial bandit problem removes many assumptions about the reward generating process, it can often be used as a baseline in variants of the stochastic bandit problem which change the assumptions on the reward generating process (although sometimes the regret definition will be different). For example, when the rewards are stochastic but the distributions can change over time, adversarial bandit algorithms can be used as a baseline for comparison. It is mainly for this purpose that adversarial bandits will be considered in this thesis.

2.3.2 Linear Bandits

In all of the bandit models so far described, it has been assumed that there are only finitely many arms and the regret bounds presented scale with the number of arms. However, often we are in settings where we have a very large, or possibly infinite, number of arms. In this case it is desirable to develop algorithms that scale better. Clearly, this will be impossible if all the arms are still assumed to be independent and there is no information shared between them. Therefore, it is necessary to make some assumptions on the structure and correlation between the arms. The simplest such assumption is that each action can be represented as a d dimensional feature vector, and that the expected reward is the inner product of this feature vector with some unknown d dimensional parameter vector, θ^*, common to all actions. This setting is formalized in the linear bandits problem.

In the (stochastic) linear bandits problem, at each time step t, the player selects

an action $X_t \in \mathcal{X}_t \subset \mathbb{R}^d$ from a possibly changing set of d dimensional feature vectors $\mathcal{X}_t$. The player then receives reward

$$Y_t = \langle X_t, \theta^* \rangle + \epsilon_t,$$

where ϵ_t is conditionally R-sub-Gaussian noise (see Appendix A.1) and $\theta^* \in \Theta \subset \mathbb{R}^d$. Note that it is assumed that the player has knowledge of the feature vectors of all actions $X \in \mathcal{X}_t$ and that it is the parameter θ^* which is unknown (although Θ will typically be known). The performance of an algorithm for the linear bandits problem is again typically measured in terms of it's regret. In this case, the regret up to horizon T is,

$$\mathfrak{R}_T = \sum_{t=1}^{T} \max_{X \in \mathcal{X}_t} \langle X, \theta^* \rangle - \sum_{t=1}^{T} \langle X_t, \theta^* \rangle.$$

Note that this is not the expected regret and many approaches for linear bandits will give high probability regret bounds. As in the stochastic K-armed bandit problem, there have been algorithms developed for linear bandits based on both the upper confidence bound and Thompson sampling approaches. Before discussing these, it is worth considering lower bounds on the regret.

Multiple lower bounds on the regret in the linear bandits problem have been presented under different assumptions about $\mathcal{X}_t$. Firstly, if $\mathcal{X}_t = \mathcal{X} = \{(x_1, \ldots, x_d) : x_1^2 + x_2^2 = x_3^2 + x_4^2 = \cdots = x_{d-1}^2 + x_d^2 = 1\}$ is the Cartesian product of d/2 circles, with θ^* restricted so that rewards lie in $\{-1, 1\}$, then Dani et al. (2008) showed that the regret must be $\Omega(d\sqrt{T})$. If the action set is a hypercube, that is $\mathcal{X}_t = \mathcal{X} = [-1, 1]^d$, and $\Theta = \{-1/\sqrt{T}, 1/\sqrt{T}\}^d$, the regret must also satisfy $\mathbb{E}[\mathfrak{R}_T] = \Omega(d\sqrt{T})$ (Lattimore and Szepesvári, 2018). Additionally, when $d \leq 2T$ and the action set is a sphere $\mathcal{X}_t = \mathcal{X} = \{x \in \mathbb{R}^d : \|x\|_2 = 1\}$, then there exists a $\theta \in \mathbb{R}^d$ with $\|\theta\|_2 = d/\sqrt{T}$ such that $\mathbb{E}[\mathfrak{R}_T] = \Omega(d\sqrt{T})$ (Rusmevichientong and Tsitsiklis, 2010). Hence, in most settings, the non-asymptotic lower bound for the linear bandits problem is $\Omega(d\sqrt{T})$.

Asymptotic lower bounds for linear bandits with finite action spaces were proven in (Lattimore and Szepesvari, 2017).

In the upper confidence bound approaches for linear bandits, the general idea is to construct high probability bounds on θ^*. In linear bandits, these upper confidence bounds on θ^* are constructed by estimating θ^* (often by regularized least squares) using all past observations $(X_1, Y_1), \ldots, (X_{t-1}, Y_{t-1})$ and then building confidence ellipsoids around this estimate which contain θ^* with high probability. Then, at each time step t, the action X_t which maximizes the inner product with some θ in the confidence set C_{t-1}, is selected, i.e.,

$$\text{select} \quad (\tilde{X}_t, \tilde{\theta}_t) = \underset{(x,\theta) \in \mathcal{X}_t \times C_{t-1}}{\arg\max} \langle x, \theta \rangle \qquad \text{then play,} \qquad X_t = \tilde{X}_t.$$

The key issue when defining confidence sets for this problem is to deal with the dependencies between the covariates. The first optimistic approach to the linear bandits problem was in (Auer, 2002) where the dependencies were dealt with theoretically by using a wrapper algorithm to divide observations into sets of almost independent observations. A similar approach was taken in (Chu et al., 2011) and (Li et al., 2010). Dani et al. (2008) use a more sophisticated martingale argument to deal with the dependencies, through which they are able to obtain a tighter regret bound for the LinRel algorithm of Auer (2002) of $O(d \log(T) \sqrt{T \log(T/\delta)})$ with probability $1-\delta$. A different approach was taken in (Rusmevichientong and Tsitsiklis, 2010) for the case where $\|\theta^*\|_2 \leq S$ for some constant $S > 0$. Here a regret bound of $O(d\sqrt{T} \log^{3/2}(T))$ was shown, which matches their lower bound up to polylogarithmic factors. Abbasi-Yadkori et al. (2011) improved on these results by estimating θ^* using regularized least squares and developing strong self-normalized bounds for vector martingales.

They define the confidence sets C_t at time t as,

$$C_t = \left\{ \theta \in \mathbb{R}^d : \|\hat{\theta}_t - \theta\|_{V_t^{-1}} \leq R\sqrt{2\log\left(\frac{\det(V_t)^{1/2}\det(\lambda I_d)^{-1/2}}{\delta}\right)} + \sqrt{\lambda}S \right\}$$

where $V_t = I_d\lambda + \sum_{n=1}^{t} X_n X_n^T$, for a regularization parameter of the least squares procedure, λ, and $\|x\|_A^2 = x^T Ax$ for a $x \in \mathbb{R}^d, A \in \mathbb{R}^{d\times d}$. This leads to the algorithm OFUL whose regret is shown to be $O(d\log(T)\sqrt{T} + \sqrt{dT\log(T/\delta)})$ with probability at least $1-\delta$ by Abbasi-Yadkori et al. (2011). Thus the OFUL algorithm matches the lower bound of $\Omega(d\sqrt{T})$ up to logarithmic factors.

As in the stochastic K-armed bandit problem, an alternative to the upper confidence bound approach is to take a Bayesian approach and use a Thompson Sampling style algorithm. When using a Thompson sampling approach for the linear bandits problem, it is useful to observe that in linear regression when the noise is Gaussian with known variance σ^2, if a Gaussian prior is placed on θ^* then the posterior is conjugate and consequently is also Gaussian. This means that Thompson sampling can be easily applied to the linear bandits problem. This was demonstrated experimentally by Scott (2010); Chapelle and Li (2011); May et al. (2012). The theoretical regret guarantees, however, were more difficult to obtain. In particular, for frequentist regret guarantees, it has been necessary to inflate the posterior variance. In (Agrawal and Goyal, 2013b), regret bounds of $O^*(d^{2/3}\sqrt{T})$[1] are achieved with probability $1-\delta$ by using a Gaussian prior with variance of $\sigma\sqrt{9d\log(T/\delta)}$ for some $\delta \in (0,1)$. This is equivalent to inflating the variance of the posterior at each time step by $\sqrt{9d\log(T/\delta)}$. A further regret bound of $O^*(d^{2/3}\sqrt{T})$ was provided in (Abeille and Lazaric, 2017), where an interesting connection between optimistic approaches and Thompson sampling was given. Particularly, they showed that Thompson sampling is equivalent to playing the arm with largest upper confidence bound when the confidence term has been multiplied by a random sample from an appropriate distribution. The Bayesian

[1] O^* is used to suppress logarithmic factors

regret of Thompson sampling for linear bandits was also considered in (Russo and Van Roy, 2014, 2016) where it is shown to be $O(d\log(T)\sqrt{T})$.

Note that, as shown in (Lattimore and Szepesvari, 2017), neither Thompson sampling nor any of the optimistic approaches described are asymptotically optimal. Lattimore and Szepesvari (2017) propose an 'Explore-Then-Commit' style algorithm in which the horizon is split into an exploratory phase where all arms are played, and an exploitative phase, where only the best arm is played. This algorithm achieves asymptotically optimal rate but has weaker finite time regret guarantees. The linear bandits problem has also been studied in the adversarial setting (see Part VI of (Lattimore and Szepesvári, 2018) and references therein for an overview of the work done in this area). Furthermore, there have been several extensions of the stochastic linear bandits problem to consider the case where the reward can be modeled using a generalized linear model (Filippi et al., 2010; Li et al., 2017; Jun et al., 2017). The stochastic contextual bandits problem generalizes this further by allowing the expected reward to be any function of the action and context (Langford and Zhang, 2008; Rigollet and Zeevi, 2010; Lattimore and Szepesvári, 2018).

2.3.3 Gaussian Process Bandits

Although the linear bandits problem discussed above allows us to deal with the case where we have a large number of arms, the assumption that the reward is a linear combination of the feature vectors and the unknown parameter is somewhat limiting. Therefore, various settings where the linearity assumption can be relaxed have also been studied. One such setting is the Gaussian process bandits problem. In this problem, the actions are covariates in $\mathcal{X} = [0,1]^d$ and at each time step t, a covariate is selected and a reward of the form $Y_t = f(X_t) + \epsilon_t$ is received, where f is some function and ϵ_t is i.i.d $\mathcal{N}(0, \sigma^2)$ noise. The aim is to minimize the regret with respect to the maximum of this function. In the Gaussian process bandits problem, a Bayesian

approach is taken and it is assumed that f is a sample from a Gaussian process (GP). A brief introduction to Gaussian processes is given in Appendix A.4 and more details can be found in (Rasmussen and Williams, 2006).

In the Gaussian process bandits problem, at time $t = 0$ we assume f is sampled from a GP with mean 0 and known kernel $k(x, x')$. Typically in Gaussian process bandits, the kernel will be assumed to be known exactly, and this is almost equivalent to an assumption on the smoothness of the functions. The aim is then to select the sequence of covariates to maximize $f(x)$, or equivalently to minimize the regret. For covariates $X_1, \ldots, X_T$ chosen by the algorithm, the regret is defined as

$$\mathfrak{R}_T = \sum_{t=1}^{T} (f^* - f(X_t)).$$

Note that this definition of $\mathfrak{R}_T$ is random due to both the potential randomness in the choice of the X_t and the random function f.

One of the most popular algorithms for the Gaussian process bandits problem is the GP-UCB algorithm of Srinivas et al. (2010). This is an intuitive algorithm which uses the normality of the posterior at each covariate, x, to define an upper confidence bound on $f(x)$, and then selects the covariate with highest upper confidence bound. Let $\mu_{t-1}(x)$ denote the posterior mean of the GP at time t and covariate $x \in \mathcal{X}$ and let $k_{t-1}(x, x')$ denote the posterior covariance function. Then, at time t, the GP-UCB algorithm selects $X_t = \arg\max_{x \in \mathcal{X}} \{\mu_{t-1}(x) + \sqrt{\beta_t k_{t-1}(x, x)}\}$ where β_t is a confidence parameter defined in relation to the assumptions on f and $\mathcal{X}$, but which is usually logarithmic in t. The performance of this approach will depend on how much information can be shared between covariates. Hence, the performance of GP-UCB will depend on the kernel of the GP. This is manifested in the regret bound by an information theoretic term, γ_T, defined as the maximal information gain. Intuitively, this is the maximum amount of information we can gain about f after observing T samples. Srinivas et al. (2010) bound this for common Gaussian process kernels

(see Appendix A.4.3 for definitions of some common kernels). Specifically, for the squared exponential kernel with any lengthscale, $\gamma_T = O((\log(T)^{d+1})$, for Matérn kernels again with any lengthscale and parameter ν, $\gamma_T = O(T^{\frac{d(d+1)}{2\nu+d(d+1)}})$, and for linear kernels, $\gamma_T = d\log(T)$. The regret of GP-UCB is then shown to be $O(\sqrt{T\beta_T\gamma_T})$ with probability $1-\delta$ (where this probability is over f). Srinivas et al. (2010) also provide a high probability regret bound of $O(\sqrt{T}(B\sqrt{\gamma_T} + \gamma_T))$ for the case where f is a fixed function in the Reproducing Kernel Hilbert Space (RKHS) corresponding to the kernel $k(x, x')$ and has RKHS norm bounded by B (so in this case, the probability is over the noise). See Appendix A.4.2 or (Rasmussen and Williams, 2006) for details of the RKHS associated with a Gaussian process. The GP-UCB algorithm has also been shown to work well in practice (Srinivas et al., 2010).

There have been various extensions of the GP-UCB algorithm and other methods proposed for the Gaussian process bandits problem. Furthermore, the Gaussian process bandits problem is similar to the Bayesian optimization problem (Frazier, 2018) which has also been studied extensively. However, in Bayesian optimization, the aim is to output a good $X_T \in \mathcal{X}$ after T plays rather than minimizing the regret. Here, we will focus on methods that come with theoretical regret guarantees on the cumulative regret, as this is more relevant to our setting.

Wang et al. (2014) considered the case where the hyperparameters of the GP kernel (e.g. lengthscale) were unknown. They showed that it is possible to both tune the hyperparameters and minimize the regret simultaneously, proposing an algorithm that has regret $O^*(\gamma_{T-1}\sqrt{T\gamma_T})$[2] with high probability for γ_T defined as in (Srinivas et al., 2010). Krause and Ong (2011) extend the GP-UCB algorithm to consider a contextual version. Here, in each round t, the environment presents the player with a m-dimensional context c_t and the player must select an $x \in \mathcal{X}$ to minimize $f(c_t, x)$. For this problem, the regret is defined as $\mathfrak{R}_T = \sum_{t=1}^{T}(f(c_t, X_t^*) - f(c_t, X_t))$. Using

[2]We use the notation O^* to suppress logarithmic factors.

a composite kernel, Krause and Ong (2011) develop an upper confidence bound approach which has regret $O^*(\sqrt{(d+m)T\gamma_T})$ where here γ_T is defined as in Srinivas et al. (2010) but for the $d' = d+m$ dimensional case. Bogunovic et al. (2016) consider the case where the aim is not to find the maximum of a single GP f, but rather a sequence of Gaussian processes which evolve according to the dynamics $f_{t+1}(x) = \sqrt{1-\epsilon}f_t(x) + \sqrt{\epsilon}g_{t+1}(x)$ where $\{g_t\}$ are a sequence of $\mathcal{GP}(0,k)$ random functions, $f_1 = g_1$, and $\epsilon \in [0,1]$. The regret here is defined as $\mathfrak{R}_T = \sum_{t=1}^{T}(f_t(X_t^*) - f_t(X_t))$. Bogunovic et al. (2016) present two modifications of GP-UCB to this setting which either use a sliding window or discount factor to forget old observations. They provide a lower bound for this problem of $\Omega(T\epsilon)$ and then show that, with high probability, their approaches achieve regret $O^*(\max\{\sqrt{T}, T\epsilon^\alpha\})$ for squared exponential kernels and some known $\alpha \in [0,1]$ depending on the algorithm, and $O^*(\max\{\sqrt{T^{\frac{d(d+1)}{2\nu+d(d+1)}}}, T\epsilon^\alpha\})$ for Matérn kernels.

Since Gaussian processes are typically interpreted using Bayesian inference, it is natural to use a Bayesian algorithm such as Thompson sampling in this setting. Russo and Van Roy (2014) show that it is possible to use a standard Thompson sampling algorithm (where at each time t a function is sampled from the posterior and then the covariate maximizing this sampled function is played) to achieve Bayesian regret of $O(\sqrt{T\gamma_T \log(T)})$ where γ_T is the maximal information gain of Srinivas et al. (2010). This gives an almost identical regret bound as that of the GP-UCB algorithm (Srinivas et al., 2010). There have also been various different algorithms proposed which are not based on upper confidence bounds or Thompson sampling but that have theoretical guarantees (see e.g., Bull (2011); Wang et al. (2016); Contal and Vayatis (2016); Wang et al. (2014); Shekhar et al. (2018)).

The problem of finding lower bounds for the Gaussian process bandits problem has been studied by Grünewälder et al. (2010); Scarlett et al. (2017); Scarlett (2018). Grünewälder et al. (2010) provide a lower bound on the maximal Bayesian regret, that

is $\mathbb{E}[\max_{x\in\mathcal{X}} f(x) - \max_{1\leq t\leq T} f(X_t)]$ with expectation taken over f as well, and provide an algorithm with nearly matching upper bound. Lower bounds on the Bayesian cumulative regret (the regret defined at the beginning of this section) for the one dimensional case where $\mathcal{X} = [0, 1]$ were provided in (Scarlett, 2018). Here it was shown that the Bayesian cumulative regret must satisfy $\mathbb{E}[\mathfrak{R}_T] \geq \Omega(\sqrt{T})$ for any kernel satisfying some assumptions on the smoothness (these assumptions hold for the squared exponential kernel and for Matérn kernels with $\nu > 2$). This means that the celebrated approach in Srinivas et al. (2010), and any extensions that have regret bounds involving γ_T, are sub-optimal, particularly for the Matérn kernel. Scarlett (2018) then provide an algorithm based on successively eliminating sub-optimal regions (similar to the Improved UCB algorithm (Auer and Ortner, 2010) for K-armed bandits) which achieves Bayesian regret $O(\sqrt{T\log(T)})$ for the one-dimensional problem. Scarlett et al. (2017) provide lower bounds on the frequentist cumulative regret for specific kernels. In this case, there is a fixed function f_0 being maximized which has bounded RKHS norm. They show that for the squared exponential kernel, the frequentist regret must be $\Omega(\sqrt{T}\log(T)^{d/4})$, while for the Matérn kernel with parameter v, it must be $\Omega(T^{\frac{v+d}{2v+d}})$. These bounds show that the frequentist version of Srinivas et al. (2010), and consequently, any frequentist regret bounds that involve γ_T, are sub-optimal, although these are not as sub-optimal as in the Bayesian case. It is interesting to observe that for the one-dimensional Matérn kernel, there is a significant difference between the Bayesian regret of Scarlett (2018) and the frequentist regret of Scarlett et al. (2017).

In Chapter 6, we will use Gaussian processes within a stochastic bandit problem to model the dependence of the reward of an arm on the time since it was last played. Although the recovering bandits problem we consider in Chapter 6 is different to the Gaussian process bandits problem discussed here, some of the techniques and results we use will come from the literature on Gaussian process bandits.

2.3.4 Delayed Feedback Bandits

One extension of the multi-armed bandit problem that arises naturally in many applications, such as advertising and clinical trials, is that of delayed feedback. Typically, in these applications, after an arm is played (i.e. an advert is shown or a drug is given to a patient) the reward from that play is not received immediately, but instead it is delayed. This problem is also relevant to education since the benefit to a student of answering a question will be delayed. Furthermore, in education the individual effects of the questions will often be confounded, so we only observe the cumulative effect of a series of questions. In Chapter 5, we study an extension of the delayed feedback bandits problem to the setting where we only receive an aggregated reward after some delay and we do not know which arms contributed to it. However, in most of the related work on delayed feedback bandits discussed here, it is assumed that after some delay, the player receives an observation along with knowledge of which arm generated it.

In the delayed feedback bandits problem, it is necessary to make some assumptions about the delays. In the simplest case, it can be assumed that the delay is a fixed, known constant. Dudik et al. (2011) study the contextual bandit problem with constant delays. Here, at each time t, the learner observes a context presented by the environment and then selects an action. The reward from this context-action pair is then observed $d \geq 0$ steps later. The algorithm presented in Dudik et al. (2011) is a policy elimination algorithm which at each time step, uses only the received observations to eliminate sub-optimal policies. With probability greater than $1 - \delta$, they show that the worst case regret of this algorithm is bounded by $O(\sqrt{KT \log(TN/\delta)} + d\sqrt{K \log(TN/\delta)})$ where N is the number of policies in the initial policy class. This corresponds to an additive regret penalty of $O(d\sqrt{K \log(TN/\delta)})$ compared to the non-delayed version of the problem.

Delays have also been studied in the adversarial bandits problem. In (Neu et al.,

2010) Markov decision processes with adversarial bandit feedback are studied in the setting where any policy will achieve reward close to its average reward in $O(\rho)$ steps for some known ρ. In order to deal with this, they propose to use observations up to time $t-d$ where $d \approx \rho \log(T)$ to construct estimates of the rewards. From introducing this 'delay' into their approach, they are penalized in the regret multiplicatively. In their follow-up work, Neu et al. (2014) consider the same setting and give an algorithm whose regret is also penalized multiplicatively, but obtains the improved rate $O(\sqrt{(d+1)KT})$. Cesa-Bianchi et al. (2016) consider a different adversarial delayed setting where agents interact and only receive information about the other agents after some $\tau \in \{0, \ldots, d\}$ steps. They provide an algorithm which consists of running the EXP3 algorithm (Auer et al., 2002b) using only the received observations. This leads to regret $O(\sqrt{(d+1+K)T\log(K)}$.

An alternative, more realistic assumption about the delay is to assume that it is stochastic. In this case, when an arm is played at time t, a delay τ_t is sampled from the delay distribution of that arm and the observation from that play is received at the (random) time $t + \tau_t$. In many cases, it is assumed that the delay distribution is the same across all arms and that the delays are sampled independently of the rewards. Joulani et al. (2013) considered the general partial monitoring setting under this assumption about the delay, and, as an example, also considered the stochastic and adversarial K-armed bandit problem. In the stochastic multi-armed bandit setting, they showed that compared to the standard (non-delayed) stochastic bandit problem, the regret increases by an additive factor relating to the number of missing observations. For delay distributions with a finite expected delay, $\mathbb{E}[\tau]$, this additive regret penalty is the expected delay itself.

Joulani et al. (2013) provide two algorithms for the stochastic delayed multi-armed bandit problem, both of which achieve worst case regret that scales with $O(\sqrt{KT\log T} + K\mathbb{E}[\tau])$. In the first algorithm, they directly modify the UCB1 algo-

rithm of Auer et al. (2002a) by constructing modified confidence bounds based only on the observations received by each time point. The second approach, QPM-D, is a black box algorithm which allows for any algorithm for the multi-armed bandit problem to be used in the presence of delays. It works by creating queues of rewards for each arm. Every time the algorithm receives an observation from a given arm, it is placed in the queue. At each time step, the base bandit algorithm will suggest an arm to play (using only the received observations). If there are rewards in the queue for that arm, the first one will be taken, otherwise the suggested arm will be played until a reward arrives. This black box approach can also be used in the adversarial setting where the base algorithm is one for non-stochastic bandits. In this case, the regret scales with $O((1+\mathbb{E}[\tau])\sqrt{KT})$.

The queue based idea also underpins the approach of Mandel et al. (2015). Here they consider stochastic bandits with stochastic delays that are bounded by some constant $d \geq 0$. They take a Bayesian approach and prove the same regret bound as the QPM-D algorithm of Joulani et al. (2013) and show improved empirical performance. Note that neither Joulani et al. (2013) nor Mandel et al. (2015) assume any knowledge of the delay distribution. Vernade et al. (2017) also consider stochastic delays in a stochastic bandit problem but take a different approach, which assumes that the entire delay distribution is known. Motivated by the problem of selecting adverts in online advertising, they only consider Bernoulli reward distributions but also consider the case where the observations may be 'censored', that is, no observation for a play may ever be received if the delay exceeds some threshold. In the censored setting, with threshold m, they show a lower bound on the regret of $\Omega(\sum_{j\neq j^*} \frac{\mathbb{P}(\tau\leq m)(\mu^*-\mu_j)}{KL^B(\mathbb{P}(\tau\leq m)\mu_j, \mathbb{P}(\tau\leq m)\mu^*)})$ where $KL^B(\theta,\theta')$ is the KL-divergence between two Bernoulli random variables with success probabilities θ and θ'. In this censored settings, they provide modifications of the UCB and KL-UCB algorithms which nearly match this lower bound. For the non-censored case, they show that the Lai-Robbins lower bound of (2.1) also holds,

and for delay distributions with light tails, they obtain similar upper bounds to the censored case.

Another setting where delays are encountered in sequential decision problems is when parallelizing or performing batch updates. Perchet et al. (2016) considered the batched problem in the stochastic K-armed bandit setting. They propose Explore-Then-Commit policies which play each arm an equal number of times in a batch and then, at the end of the batch, they test if one arm is significantly better than the others. If there is found to be a statistically significant better arm, this arm is played until the horizon is reached. In this setting, we can only make decisions at the end of a batch, which is equivalent to having delays on the observations (although here all observations from arms played in the batch are received at the end of the batch). In the two-armed case, if there are a relatively small number of batches (approximately less than $\log(T)$), Perchet et al. (2016) show that the problem dependent regret of their algorithm will be almost minimax optimal.

Desautels et al. (2014) consider both the delayed and batch version of the Gaussian process bandits problem. Their approach consists of modifying the exploration term of the GP-UCB algorithm of Srinivas et al. (2010) by a multiplicative factor of $\exp(2C)$, where C is such that $\frac{\tilde{\sigma}_{t-1}(x)}{\sigma_{t-1}(x)} \leq \exp(C)$ for $\sigma_{t-1}(x)$, the posterior standard deviation using knowledge of all played arms up to time t (for a fixed, known delay, this can be calculated exactly without needing the observations), and $\tilde{\sigma}_{t-1}(x)$, the equivalent using only the observations received up to time t. They show that the regret of their approach increases by a multiplicative term of $O(\exp(C))$ compared to the non-delayed case. Chapelle and Li (2011) consider the practical performance of several K-armed bandit algorithms under the presence of delay, and specifically in the batch setting. They show that empirically Thompson sampling style algorithms are more robust to the effects of delay.

Since an initial version of our work on delayed, aggregated anonymous feedback

(Chapter 5) was made available online, there have been several new works looking at delayed feedback in a bandit setting. Cesa-Bianchi et al. (2018) consider the adversarial version of the problem we consider in Chapter 5 with the additional difficulty that the rewards can be divided between future time steps. Here, when an arm is played, an adversary will decide how to split the reward up over the next d time steps, and at each time step, the player only observes the sum of the portions of the past d rewards that are received in that round. They do not learn which arms contributed to each observation, nor do they ever learn the complete (or partial) reward of a play. They consider rarely switching strategies and play each arm consecutively, using geometrically distributed phase lengths. Their algorithm achieves regret $O(\sqrt{KTd})$ and they also provide a lower bound of the same order, demonstrating that their algorithm is rate optimal in the adversarial setting.

Vernade et al. (2018) extend their previous work on delayed (non-anonymous) feedback to the contextual linear bandit setting. They provide a lower bound of $\Omega(\sqrt{\frac{TD}{\mathbb{P}(\tau \leq m)}})$ where m is the censoring threshold and D the dimension of the feature vector and also provide an algorithm whose regret almost matches this.

2.3.5 Non-Stationary Bandits

The non-stationary bandit problem is an extension of the multi-armed bandit problem to allow the reward distributions of each arm to change over time. In Chapter 6, we consider a particular non-stationary bandit problem where the reward of an arm changes depending on how long it has been since the arm was played. There are typically two types of non-stationary bandit problems which have been studied in the literature, namely the *restless* bandit problem, where the reward distribution of each arm can change at any time, and the *rested* bandits problem, where the reward distribution of an arm only changes when the arm has been played. Here we will discuss some of the key works in both these domains, since the recovering bandits

problem studied in Chapter 6 is related to both problems. Note that in the non-stationary problem, and particularly the rested bandits problem, the regret can be difficult to define, and as such, many authors consider their own definition of regret unique to the problem studied.

The restless bandits problem was first introduced by Whittle (1988). In this initial work, the bandit problem was represented as a Markov decision process (as in the work on Gittins indicies, see Section 2.2.3), but Whittle (1988) also assumed that the states of the unplayed arms can change, and specifically, that the rewards of the unplayed arms also evolve according to a Markov chain, which is not necessarily the same as the one governing the evolution of the arms when they are played. This problem has been shown to be PSPACE-hard (Papadimitriou and Tsitsiklis, 1999), and so finding optimal solutions to this problem is not feasible. Instead, Whittle (1988) presented a heuristic index policy, now known as 'Whittle index', which works well for many of the problems motivating his work. However, it has been shown in (Ortner et al., 2012) that such index policies can be sub-optimal in terms of regret in the restless bandits problem.

Slivkins and Upfal (2008) consider the restless bandit problem where the 'expected' reward (or state) of each arm j, $\mu_j(t)$, evolves according to a Brownian motion with volatility $\sigma_j \leq \sigma$ taking values on a bounded interval. At time t when arm J_t is played, a stochastic reward in $[0,1]$ with expected value $\mu_{J_t}(t)$ is observed. They consider the 'steady-state' regret, $\mathbb{E}[\mathfrak{R}^{ss}] = \limsup_t \sup_{t_0} \mathbb{E}[\frac{1}{t}\sum_{s=t_0+1}^{t_0+t}(\mu^*(t) - \mu_{J_t}(t))]$ where $\mu^*(t)$ is the optimal expected reward available at time t. They consider two variants of the problem, the 'state-informed' case where the player knows the current state (expected rewards) before selecting an arm (as in Whittle (1988)) and the 'state-oblivious' case where after playing an arm, the player only receives knowledge of the reward generated, and not the state. In the state-informed case, Slivkins and Upfal (2008) provide a lower bound on the steady-state regret of $\Omega(K\sigma^2)$ and an intuitive

algorithm which plays the arm with the largest expected reward unless the uncertainty of another arm is sufficiently large. This algorithm matches the lower bound up to logarithmic factors. An alternative algorithm is provided for the state-oblivious case which achieves steady state regret $O^*\left(\sqrt{K\sum_{i=1}^{K}\sigma_i^2}\right)$ up to logarithmic factors.

A more general restless bandit problem was studied in (Ortner et al., 2012). Here, the only assumption on the Markov chain governing the evolution of the reward of each arm is that it is irreducible, meaning that it is possible to get from any state to any other state. In their definition of regret, the performance of their algorithm is compared to an oracle policy which knows the rewards and transition probabilities and selects the best sequence of T actions using this information. Note that since in the restless bandits problem, the state evolves independently of the actions chosen, this is equivalent to selecting the best action at each time step. They present an algorithm for this problem based on a modification of the popular UCRL2 algorithm (Jaksch et al., 2010) for reinforcement learning, and bound its regret in terms of the mixing times of the Markov chains and the maximal length of time it takes to get from one state to another.

Another variant of the restless bandit problem is the setting where the reward distribution changes abruptly at certain points. Finding the points where a time series changes abruptly has been studied as the changepoint problem in statistics (see e.g. Page (1955); Hinkley and Hinkley (1970)) and many works studying bandits in an abruptly changing environment employ these methods to detect the changes in an arm's reward distribution. Hartland et al. (2006) present two algorithms for the problem which, after detecting a change using a statistical procedure, either discounts the data from the time before the change, or employs a second bandit algorithm to determine whether to restart the bandit problem. Both of these algorithms use a modification of UCB, and are shown to perform well experimentally, although no theoretical guarantees are given. Mellor and Shapiro (2013) present an empirical study

of an algorithm that incorporates Bayesian changepoint detection into Thompson sampling.

Garivier and Moulines (2011) also consider restless bandits in an abruptly changing environment but take a different approach. Instead of using changepoint techniques, they consider two modified UCB approaches. The first, Discounted UCB, uses a discount factor $\gamma \in (0, 1)$ to downweigh past observations and correspondingly adjust the confidence bounds. The second, Sliding-Window UCB, only considers the past $\tau > 0$ timesteps. Garivier and Moulines (2011) provide a surprising lower bound on the worst case regret of any algorithm used in the abruptly changing bandit problem in terms of its regret in the standard stochastic bandit problem. They define the non-stationary regret as $\mathbb{E}[\mathfrak{R}_T^{\text{NS}}] = \mathbb{E}[\sum_{t=1}^{T}(\mu^*(t) - \mu_{J_t}(t))]$ where $\mu_j(t)$ is the expected reward of arm j at time t, and $\mu^*(t)$ is the expected reward of the optimal arm at time t. Then Garivier and Moulines (2011) show that for the problem with two changepoints, $\mathbb{E}[\mathfrak{R}_T^{\text{NS}}] \geq \frac{cT}{\mathbb{E}[\mathfrak{R}_T]}$ for some universal constant $c > 0$ where the same policy is used to define the standard regret, $\mathbb{E}[\mathfrak{R}_T]$, and the non-stationary regret. This motivates the need to develop strategies specifically tailored to the non-stationary bandit problem. If the number of changepoints in T steps, C_T is known, the non-stationary regret of Discounted UCB and Sliding-Window UCB are both $O^*(\sqrt{KTC_T})$. Raj and Kalyani (2017) consider a discounted version of Thompson sampling and optimistic Bayesian sampling and show empirically that these perform better than several of the UCB approaches in many rested and restless bandit environments, although they present no theoretical guaranties on the performance of the Bayesian approaches.

Besbes et al. (2014) consider a restless bandit problem where there is a known 'variation budget', V_T, quantifying the total possible change in the reward distributions of the arms in T plays. Specifically, V_T is defined such that $\sum_{t=1}^{T-1} \sup_{1 \leq j \leq K} |\mu_j(t) - \mu_j(t+1)| \leq V_T$. They consider the same definition of non-stationary regret as Garivier and Moulines (2011) and prove a lower bound of $\Omega((KV_T)^{1/3}T^{2/3})$. They then propose an

algorithm related to the EXP3 algorithm of Auer et al. (2002b) which uses prior knowledge of V_T, and show that this algorithm achieves regret $O((K\log(K)V_T)^{1/3}T^{2/3})$, thus almost matching the lower bound.

In the rested bandits problem, the expected reward of each arm only changes when the arm is played. This type of problem has received interest in recent years due to its applicability in the online retail setting where many bandit algorithms have been applied. However, in rested bandits, it is often difficult to decide how to define the regret since the expected reward of each arm at a given time step will depend on the past actions taken, and this will be different for the algorithm of interest and the oracle. Hence, the per step non-stationary regret, as used in the restless case, may not be appropriate here. Conversely, considering the policy regret, which compares the total expected reward of a sequence of plays to that of an oracle, may not be appropriate since computing the oracle may be too difficult. Furthermore, this may penalize an algorithm that makes a mistake early on in the learning process, which is similar to the notion of Arora et al. (2012), that sub-linear policy regret is not achievable in an adversarial bandits problem. Due to this difficulty, several simplified rested bandits problems, with alternative regret definitions, have been studied in the literature.

Bouneffouf and Feraud (2016) assume that the reward of each arm varies according to some known trend function of the times that each arm has been played. For this problem, they consider the policy regret and show that an adaptation of the UCB algorithm achieves policy regret similar to the regret of UCB in the standard bandit setting. In the rotting bandits problem of Levine et al. (2017), the expected reward of each arm decays according to some unknown monotonically decreasing function of the number of times it has previously been played. They consider the policy regret, however, they are able to show that in this setting the optimal policy (when the rewards are known) is to greedily choose the arm with highest expected reward at

any time step. In the case where the decay of the reward is not governed by any function with known functional form, Levine et al. (2017) present a sliding window algorithm that achieves policy regret $O((K\log(T))^{1/3}T^{2/3})$. However, if the decay is known to have some specific functional form with unknown parameters, these can be estimated, and an alternative algorithm is presented that achieves problem dependent regret $O(\sum_{j=1}^{K}\frac{\log(T)}{\Delta_j})$, as in the standard K-armed bandit problem.

In (Bouneffouf and Feraud, 2016) and (Levine et al., 2017), they were able to consider the policy regret since the problem specification was such that the expected reward depended on the sequence of past plays only through the number of times each arm had previously been played. This meant that their analysis could be done by bounding the number of times an arm was played when it was sub-optimal, as is commonly the case in standard multi-armed bandits. Cortes et al. (2017) consider a setting where this is not possible. In this setting, it is assumed that the reward process is selected by an adversary and the performance of an algorithm is measured in terms of its per step regret. Cortes et al. (2017) propose a UCB algorithm, DISC-UCB, that incorporates a notion of 'weighted discrepancy' into the confidence bounds. For arm j at time t, the weighted discrepancy measures how different the future observations are likely to be from the past ones, and is defined as $D_{j,t}(\mathbf{w}) = \mathbb{E}[X_{j,t+1}|\mathbf{X}_t^j] - \sum_{s=1}^{t} w_s\mathbb{E}[X_{j,s}|\mathbf{X}_j^{s-1}]$. If the discrepancy is known or bounded for all arms, the problem dependent regret can be bounded by $O(\sum_{j=1}^{K}\max_{1\leq t\leq T}\Delta_{j,t}\log(T)/\min_{1\leq t\leq T}\Delta_{j,t}^2)$ where $\Delta_{j,t}$ is the per step sub-optimality gap of arm j, $\Delta_{j,t} = \mathbb{E}[X_{j_t^*,t}|\mathbf{X}_{j_t^*}^{t-1}] - \mathbb{E}[X_{j,t}|\mathbf{X}_j^{t-1}]$. Note that if the reward of an arm gets arbitrarily close to the optimal at any time point, this regret bound will increase to infinity. Hence it may make more sense to consider the problem independent regret in this setting. The corresponding problem independent regret bound of this approach would be $O^*(T^{2/3}K^{1/3})$ (up to logarithmic factors).

In (Mintz et al., 2017), a problem somewhere between the restless and rested

bandits problem was studied. In this problem, which they refer to as the rogue bandits problem, the expected reward of arm j at time t depends on some underlying state $x_{j,t}$ via a parametric function with unknown parameters. Here, the states evolve according to known non-linear dynamics depending on the previous state and whether or not the arm was played at the previous time step. Thus, the evolution can be different when the arm is played or when it is not played. This is similar to the recovering bandits problem studied in Chapter 6. Using knowledge of the parametric form of the reward function and the complete noise model, they estimate the parameters of the reward function for each arm using maximum likelihood. They then use these maximum likelihood estimates to develop a KL-UCB style algorithm. However, since the state dynamics evolve depending on the previous plays, it is possible to select a sequence of plays such that the maximum likelihood estimates do not converge to the true parameter values (i.e. if the observed states do not span the state space sufficiently, the maximum likelihood estimates will be biased).

Even though the choice of actions by the algorithm will affect the next state, Mintz et al. (2017) only consider the per-step regret, that is the cumulative difference in reward from the optimal action and the action taken when the state is generated by the algorithm of interest. They show that their algorithm achieves problem dependent per step regret of $O(\sum_j {}^{\log(T)}/_{\delta_j^2})$ where δ_j depends on the (random) number of plays of each arm and the minimum distance between the rewards of any arms at any time. As in (Cortes et al., 2017), δ_j can be arbitrarily small leading to almost infinite regret. The problem independent regret bound of this approach is $O^*(T^{2/3}K^{1/3})$ (up to logarithmic factors). Furthermore, the constants in this regret bound are quite large and in practice, the authors found that an algorithm based on asymptotics performs far better, although this algorithm comes with no theoretical guarantees.

2.3.6 Bandits with Knapsacks

In the (non-stochastic) knapsack problem, a player must decide which of a set of K items to place into a knapsack of fixed capacity where each item has a fixed size and reward, and the aim is to maximize the total reward of the items placed in the knapsack. In the stochastic knapsack problem, the knapsack still has a fixed size but the item sizes and rewards are stochastic. In Chapter 4, we consider using bandit techniques within the stochastic knapsack problem. The bandits with knapsacks problem, introduced by Badanidiyuru et al. (2013), is an alternative bandit problem related to the stochastic knapsack problem. In this problem, as in the standard multi-armed bandit problem, playing an arm generates a stochastic reward, but here playing each arm also generates a sample from some cost distribution. Badanidiyuru et al. (2013) assume that there is some fixed budget and the aim is to select items sequentially that maximize the total reward while ensuring that the total cost is less than the budget. They propose two algorithms for the problem, one which is a phase-based elimination algorithm and the other which uses optimistic estimates of the reward-cost ratio, and present theoretical regret bounds for both.

The bandits with knapsacks problem was extended by Agrawal and Devanur (2014) to consider the case where the knapsack constraints were no longer a linear function of the costs, but some arbitrary convex function, and the reward is also some concave function of the reward of each play. Agrawal and Devanur (2016) considered the linear contextual version of the bandits with knapsack problem, where at each time step the player receives a set of contexts $x_t(1), \dots x_t(K)$ and then selects an action J_t. The expected reward of taking action j at time t is given by $\theta^{*T} x_t(j)$ and the expected size of the item is $\lambda^{*T} x_t(j)$ for unknown parameters $\theta^*, \lambda^* \in \mathbb{R}^d$, and the aim is to maximize the cumulative reward subject to the knapsack constraint, $\sum_{t=1}^{T} \lambda^{*T} x_t(j) \leq B$. Agrawal et al. (2016) also considered a more general version of the problem, and provided a computationally efficient algorithm with strong regret

guarantees. It is important to observe that all these approaches work with knapsack sizes that effectively tend to infinity. Hence they are not directly applicable to the knapsack problem we study in Chapter 4 where the knapsack size is relatively small. Burnetas et al. (2015) considered bandits with knapsack problems with deterministic item sizes and capacities that are regularly renewed, and developed asymptotically optimal strategies for this problem. This problem is again different to the one studied in Chapter 4.

2.3.7 Optimistic Planning

In Chapter 4, we use optimistic planning techniques to find near-optimal solutions to the stochastic knapsack problem. Optimistic planning refers to a planning problem which has been tackled using optimistic (UCB) approaches from the multi-armed bandit literature. In planning problems, the aim is to return the optimal next action to take, starting from any given state in a Markov Decision Process (MDP). A complete definition of a MDP is given in Appendix A.3. A policy Π for a MDP is a mapping from state to actions dictating which action to take from any given state. We define the discounted value of a policy Π up to horizon T as $V(\Pi) = \mathbb{E}[\sum_{t=0}^{T} r_t \gamma^t | A_t = \Pi(S_t)]$, where r_t is the reward recieved at time t by taking action $A_t = \Pi(S_t)$ from state S_t, and $\gamma \in (0, 1)$ is a discount factor. The aim is often to find an optimal first action to take starting from a given state (Sutton and Barto, 1998). Optimistic planning has been shown to be able to do achieve this is various settings, while only needing to evaluate a small number of policies.

When the transition distribution is discrete, MDPs can often be represented as a tree (Szörényi et al., 2014). Here the root node is some initial state s_0 and from there, the branches represent taking each possible action to arrive at an 'action node', and then the next set of branches are the transitions to the next states leading to 'state nodes'. This repeats so that the nodes on odd levels are state nodes with branches

for each action, and the nodes on even levels are action nodes with branches for each state transition. Each policy is a subtree of this tree. Clearly for most problems, this complete tree will be huge and so performing a search of the entire tree to find the optimal policy or first action is computationally infeasible. Optimistic planning aims to use bandit techniques together with a synthetic model of the environment, which knows the reward and transition probabilities, or has access to a generative model of them, to facilitate this tree search by only searching policies (or subtrees) that have the potential to be optimal.

The aim of optimistic planning is to to find the best action to take from the initial state s_0. In some cases it is possible to bound the difference between the best possible reward that can be achieved after starting from the optimal initial action and the initial action the algorithm outputs. To this end, we define the simple regret as the difference between the maximal discounted value of a policy starting with the optimal action, and that of a policy starting with the action chosen by the algorithm. Bounds on the simple regret often involve properties of the tree and the MDP, such as similarity between leaf nodes, and as such may be difficult to interpret. In practice, when optimistic planning algorithms are deployed in real systems, the algorithm will be run using the synthetic model from the initial state s_0 to return a (near) optimal initial action a_0. This action will be taken in the real environment and the algorithm will be re-run from the resulting state.

Hren and Munos (2008) developed an optimistic planning algorithm for the case where the rewards and transitions were deterministic and the agent had a fixed budget of computational time in order to return a (near) optimal initial action. In this case the decision tree just consists of the action nodes since the transitions are deterministic. Their approach starts with an initial tree, consisting of just a root node s_0, and selects nodes to expand. A node is expanded when some computational time is used to consider all the next states from this node (i.e. all the states reachable by taking

one action from the current state) and add these to the tree. If $\mathcal{S}_t$ represents the nodes in the tree that have not been expanded (i.e. that do not have branches coming from them), the node i_t to be expanded is chosen such that $\forall j \in \mathcal{S}_t, u_{i_t} + \frac{\gamma^{d_{i_t}}}{1-\gamma} \geq u_j + \frac{\gamma^{d_j}}{1-\gamma}$ where u_i is the sum of discounted rewards along the path to node i from root s_0, and d_i is the depth of node i. For each node i, $u_i + \frac{\gamma^{d_i}}{1-\gamma}$ is an upper bound on the discounted reward of any policy passing through i. At horizon T, the action leading to the node in $\mathcal{S}_T$ with the highest u_i is selected and returned by the algorithm. Hren and Munos (2008) show that this initial action chosen by their algorithm after using T units of computational resource will have near optimal value. In particular, if $\beta \in [0, \frac{\log(K)}{\log(1/\gamma)}]$ is such that the proportion of ϵ-optimal nodes (nodes whose value is within ϵ of the optimal value) at depth d is less than ϵ^β, the simple regret is bounded by $O(T^{-\frac{\log(1/\gamma)}{\log(K\gamma^\beta)}})$.

Optimistic planning with stochastic rewards and transitions was first considered in (Bubeck and Munos, 2010) where the reward and transition probabilities were assumed to be known. Here, they considered open-loop planning, where the action taken only depends on its position in a sequence of actions and not the state the MDP arrives in after taking the previous actions in the sequence. For this problem, Bubeck and Munos (2010) provided lower bounds on the simple regret and an optimistic planning algorithm that almost matches this lower bound.

Busoniu and Munos (2012) also considered the stochastic setting with known transition probabilities and deterministic rewards in $[0, 1]$ but developed an optimistic planning algorithm for the closed-loop problem. This approach also starts with an initial tree of root node s_0 and optimistically selects nodes to expand. In this case expanding a node involves adding branches for each possible action from that state and from each of these actions adding branches to the possible next states (note that in this case the leaves of the subtree constructed by the algorithm at any stage will always be states). The decision of which nodes to expand at any time t is made using

optimistic estimates of the expected discounted reward of a continuation of a node. These optimistic estimates involve the known transition density and a bound on the discounted future reward. Busoniu and Munos (2012) provide a bound on the simple regret of this algorithm in terms of some characteristics of the tree.

Szörényi et al. (2014) presented an algorithm for optimistic planning in the general MDP framework, where the rewards and transitions are both stochastic but the rewards are bounded in $[0, 1]$. Furthermore, they only assumed access to a generative model of both the rewards and transitions, rather than knowledge of the distributions. In this case, since the reward and transition densities are unknown, the upper bounds on value of a policy need to take into account uncertainty of any estimates. Hence, they will usually consist of a term relating to this uncertainty along with bounds on the future rewards like those seen in (Busoniu and Munos, 2012). The StOP algorithm of Szörényi et al. (2014) works by maintaining a set of active policies and computing upper confidence bounds on the value of a continuation of each active policy, in order to select which one to expand. For a policy Π of depth d, these upper confidence bounds are obtained by playing according to the policy in a virtual environment m times, that is using the generative models to obtain samples of the rewards and transitions which can be combined to get m samples of the value of the policy. From these samples, they get an estimate, $\overline{V(\Pi)}$, of the value of the policy up to depth d, and then construct the upper confidence bounds on the value of a continuation of a policy, as $UCB(\Pi) = \overline{V(\Pi)} + \frac{\gamma^d}{1-\gamma} + \frac{1-\gamma^d}{1-\gamma}\sqrt{\frac{\log(1/\delta)}{2m}}$. At time t, the active policies with the two best upper confidence bounds are selected and the one with smallest depth is expanded. This arm selection criteria relies on the pure exploration multi-armed bandits algorithm of Gabillon et al. (2012)[3]. In (Szörényi et al., 2014), expanding a policy is equivalent to expanding the leaf nodes of the corresponding tree and then

[3]In the pure exploration or best arm identification version of the multi-armed bandit problem, rather than minimizing the regret over the whole horizon, the aim is to return a single best arm either (i) after a certain number of plays or (ii) with fixed confidence. See e.g. (Audibert and Bubeck, 2010; Even-Dar et al., 2006)

sampling the value of all policies generated from it a given number of times. The original policy is then replaced by its newly expanded descendants in the active set. A termination criteria is used to ensure that the algorithm outputs an initial action and that the expected maximal value of a policy starting with this action is within ϵ of the optimal. Szörényi et al. (2014) bound the number of samples necessary to do this.

Another approach using bandits to facilitate tree search has been developed in (Kocsis and Szepesvári, 2006). Here an upper confidence bound is constructed on each node of the tree (rather than on a policy) and samples are obtained by sequentially selecting nodes according the optimistic principle and observing transitions in the virtual environment. This algorithm, UCT, is very popular and has been shown to work in practice (see Browne et al. (2012) for examples). However, it may be too optimistic (Coquelin and Munos, 2007). In this thesis, we will mainly consider optimistic planning approaches rather than UCT.

Chapter 3

Motivating Problems from Education

In this chapter, we start to consider how multi-armed bandits can be used to select questions in education software. We begin by discussing some of the other progress made in applying multi-armed bandits to online education, before considering the fundamental issue of how to define the reward in such an educational environment. Finally, we detail the specific challenges arising from the task of selecting questions in education software that have motivated the multi-armed bandit problems studied in this thesis. These are, having a fixed limit of homework time, the delay in the effect of answering a question, and the importance of allowing enough time before repeating a question. In our work, we focus on the teaching of mathematics to UK secondary school students (aged 11-16), where students can answer each question correctly or incorrectly. We assume that a student will receive some benefit from answering the question regardless of whether they got it correct. We will generally also focus on the task of selecting questions for one student individually as this not only reduces the complexity of the problem, but can also be shown to improve student performance (Lee and Brunskill, 2012).

3.1 Previous Work on Using Multi-Armed Bandits in Education

One of the most notable works on using multi-armed bandit techniques in education software appears in (Clement et al., 2014, 2015). Here, two algorithms (based on different pedagogical assumptions) are presented and evaluated in a real life environment where students interacted with an online education platform that was selecting tasks based on their algorithms. Both these algorithms are adaptations of the EXP4 algorithm (Auer et al., 2002b) and, as such, consider bandits with expert advice. In this setting, the arms are the different questions and the experts are used at each time step to present the algorithm with a subset of potential arms (questions) which are appropriate at the given time. This set is determined using the so called *zone of proximal development* (Luckin, 2001). Intuitively, questions in the zone of proximal development are questions that will slightly challenge the student, but that are not too challenging.

In the first algorithm of Clement et al. (2015), minimal assumptions about a learning model are made, and the reward is defined for some parameter $d > 0$, as the difference in the proportion of the most recent $d/2$ questions that were answered correctly, and the proportion of the $d/2$ questions before that that were correct. The expert then gives a set of feasible next questions according to the theory of the zone of proximal development and the algorithm selects questions from this set to give to the student using EXP4. In the second algorithm, Clement et al. (2015) assume that each student has an estimated competency level in a skill and each activity has a corresponding difficulty level. The expert then provides minimal competency levels for each question. This can then be translated to give a set of questions in the zone of proximal development. The reward in this case is the difference in the difficulty level of the question and the student's competency level. They evaluate their approaches

in both a simulated and real life environment where the aim is to teach children to decompose numbers by considering money, and each question can be defined by a finite set of parameters. Their experimental results show that both these approaches perform well in practice, and can lead to improved student learning.

Mu et al. (2017) extend the work of Clement et al. (2015) to consider the case where the zone of proximal development needs to be estimated. They show that in the first algorithm of Clement et al. (2015), a model of student knowledge can be used to give an estimated zone of proximal development. Mu et al. (2018) then extend this model to capture how students forget material and incorporate a trace-based procedure for modeling student progression through tasks (see (Andersen et al., 2013) for details). Segal et al. (2018) also use an EXP4 style algorithm and borrow ideas from the theory of zone of proximal development. Instead of updating the weights of the EXP4 algorithm using the standard procedure, they instead update them differently depending on whether or not the student got the question correct. Their initial weights are based on offline estimates of the difficulty of each question for a particular student. Then when the student is given a question, if they get it correct, the weights of any harder questions (based on these initial estimates) are increased, whereas if they get it wrong, the weights of the harder questions are decreased. This approach shows good performance on data simulated from a model and also on a large experiment with real students.

Xu et al. (2016) take a contextual bandit approach to recommending entire sequences of courses (in this work they are thinking of courses at university level, so consider sequences of modules taken across the whole degree), taking into account any prerequisites of the courses and other external features. To do this, they first find all feasible sequences of courses and use these as arms in a contextual bandit problem. The contexts are features of the student, such as educational background, and the reward they aim to maximize is the student's Grade Point Average (GPA)

when graduating after having taken this sequence of courses. Their algorithm clusters students into groups based on their contexts and explores the course sequences for each cluster until there are enough students in the cluster, at which point the cluster is uniformly split into two. They provide theoretical regret bounds for their approach and experiments using historical data.

Lan and Baraniuk (2016) also take a contextual bandit approach, but their aim is to suggest activities for students (these activities could be a series of questions or videos) in between assessments. Here, the reward is the score in the assessment following the suggestion. They assume that estimates of the student's latent competencies in a range of different skills are available to the algorithm and use these to define contexts along with other features of the student. To reduce the dimensionality of their context space they use sparse factor analysis. They provide three algorithms for this problem; the first is a UCB logistic regression approach which has theoretical guarantees, the second is an alternative UCB approach based on asymptotics with good experimental performance (on historical data) but no theoretical guarantees, and the final one is a Thompson sampling logistic regression algorithm which again has good empirical performance but no regret guarantees.

Liu et al. (2014); Erraqabi et al. (2017) take an alternative approach. Here they are motivated by working directly with educational app and games designers. This means that as well as suggesting questions which help learning, they also want to be able to provide the designers with feedback on the effectiveness of each activity. Consequently, the trade-off between suggesting good questions and reducing uncertainty about the reward of the questions is made more explicit here. They develop algorithms which aim to play arms which maximize a weighted combination of the reward and uncertainty where the weights are determined by the user. Liu et al. (2014) show that this approach works well on data simulated from a model of student learning, when the student's reward is defined as whether they get the next random question

they are given correct. Erraqabi et al. (2017) consider an alternative definition of reward, the number of additional questions the student answers after answering the question of interest (they are considering an educational game environment where the student can give up and leave at any time), and again demonstrate good performance on simulated data along with theoretical regret bounds.

Lindsey et al. (2013) consider the problem of selecting an optimal instructional policy from a set of policies where each is represented by a set of parameters. This setting includes problems such as teaching people to distinguish between cancerous and clear xrays. A policy for teaching in this case is the sequence of positive and negative examples the student is shown. In particular, this sort of policy can be represented by a parameter, p, which gives the probability of showing a positive or negative sample given the last example was from the same category. For learning this sort of educational policy, Lindsey et al. (2013) propose to use a Gaussian process bandits algorithm, and specifically the GP-UCB algorithm of Srinivas et al. (2010). In a real life experiment, the tth participant is given a sequence of images to learn from according to the tth parameter value chosen by GP-UCB and then all participants are given the same test. The authors observe that later participants perform better on the test and that the algorithm finds a near optimal instructional policy.

Matiisen et al. (2017) present an interesting approach to teaching machine learning algorithms tasks which could also be applicable to teaching students in the educational context. They assume that each task they are trying to teach has a learning curve which governs how well the 'student' will learn the task at a given time point. They aim to give the student tasks at times when the learning curve of that task is steepest (so when the gradient is largest). For this, they apply a bandit approach where the reward is the gradient of the learning curve. This is not known explicitly and so the algorithm must estimate it. They use a Boltzmann exploration strategy (Sutton and Barto, 1998) and estimate the gradient of the learning curves using linear regression

on the last K plays. The empirical performance of this approach is demonstrated by training machine learning algorithms.

There have also been other approaches looking at applying bandits to different challenges in education systems. For example, Lomas et al. (2016) use bandits to select between different layouts of an educational app and Williams et al. (2016) present students with explanations for incorrect answers by crowdsourcing the explanations and using a bandit algorithm to pick between them. An additional line of work comes from modeling the problem of selecting questions in education software as a partially observable MDP (POMDP). Here the states are often the student's 'knowledge state' given by a procedure such as Bayesian knowledge tracing (Corbett and Anderson, 1994) and the student transitions between states by answering questions. Examples of such POMDP approaches can be found in (Rafferty et al., 2011; Theocharous et al., 2009; Antonova et al., 2016).

3.2 Defining the Reward

One of the most fundamental challenges when applying multi-armed bandit techniques to the problem of selecting questions in education software is how to define the 'reward' of a question. Intuitively, this reward should measure the amount of learning the question provided, or how much the student benefited from answering the question. A bandit algorithm will learn to suggest questions with high reward so it is important to make sure that this is appropriately defined in order to ensure that the algorithm is behaving in the desired way. Consider, for example, defining the reward as whether the student got a question correct. We may believe that it is desirable for students to get questions correct, however, using the correctness as the reward in a bandit problem will lead to the algorithm suggesting questions which are too easy for the student, as these will have the highest chance of being correctly answered. Instead,

there are several different approaches that can be taken. We discuss here some of these.

In most online education systems, the type of data that will be collected when a student answers a question will include whether they got it correct, and how long it took them to answer it. Hence, one option is to define the reward in terms of this data. For example, if the student took a long time to answer a question and then eventually got it correct, this would suggest that they thought about it a lot and then managed to figure it out. This is potentially the sort of question we want to be giving to the student. Hence, one could define the reward as $r_t = \mathbb{I}\{\text{correct}\}s_t$ where s_t is the time it took them to answer the tth question. This would stop the system suggesting really easy questions that can be answered very fast. One possible drawback of this approach is that it treats all incorrect answers the same. There are different degrees of incorrect answers which could be used to inform rewards (e.g. in a multiple choice scenario one wrong answer may be closer to the correct answer than another). There have also been several similar data-based definitions of reward in the literature. For example, Clement et al. (2015) define the reward as the difference in the proportion of the last d questions answered correctly, and Rafferty et al. (2011) use the negative of the time taken to answer the question as the reward.

An alternative approach is to use an educational model and define the reward in terms of this. For example if the model consists of various parameters representing the student's understanding in different topics, where large values indicate a high understanding of the topic, one approach could be to define the reward as the difference in the parameters after and before the question has been answered and the model has been updated with the new data. One drawback of this approach is that you will only ever be as good as your model, so if the model is wrong, the questions chosen may not be optimal. Model based approaches have been considered in the literature, for example, Clement et al. (2015) measure the reward as the difference in the knowl-

edge required to answer a question and the current knowledge of the student (both calculated by a model).

In some cases, there may be something observable that we directly want to maximize. For example, if we know that student progress is monitored through a sequence of questions at the end of every homework (a mini-test or equivalent), then it is clear that we wish to give them questions which will maximize the score in these tests. Alternatively, if participation is optional, we could define the reward as the number of future questions answered. This definition of reward is very much dependent on the specifics of the online educational system, as not all of them will have the capacity (or desire) to test students regularly or measure engagement. Using alternative observable features to define the reward has been considered by Liu et al. (2014); Erraqabi et al. (2017); Lindsey et al. (2013); Lan and Baraniuk (2016). In particular, Liu et al. (2014) use whether the next (randomly generated) question is answered correctly as a proxy for reward, whereas Erraqabi et al. (2017) use the number of additional questions the student answers. Lindsey et al. (2013) look at the score on a test after giving the student a sequence of questions, and Lan and Baraniuk (2016) consider an environment where a test is given after every activity selected by the algorithm.

From the above discussion, it is clear that defining the reward for a bandit algorithm used in education software is not straightforward. There have been many approaches proposed, each of which has advantages and disadvantages. Furthermore, not all of these definitions will be appropriate in all online education systems. Interestingly, in the studies that involve using multi-armed bandits in a live educational environment with real students, there has been no consensus made about which definition of reward to use. However, it is pleasing that in most cases the bandit algorithm still performed well in practice. Hence, the challenge of defining the reward when using a multi-armed bandit algorithm largely comes down to the setup of the system and which particular features the educator/designer wants to optimize. In what fol-

lows, and for the remainder of the thesis, we will always assume that the reward has been defined and that it is an appropriate measure of the learning process. We now discuss the specific problems in education that have motivated the work in this thesis.

3.3 Fixed Limit on Homework Time

It has long been acknowledged that, at the secondary school level, setting students homework can lead to improved academic performance (Cooper et al., 2006). As such, it has become an integral part of the learning experience. However, recently, it has been shown that setting students too much homework can lead to increased stress and anxiety (Galloway et al., 2013) and may even cause students to burn out, hindering academic performance. Therefore, it is desirable to set only a limited amount of homework. It is also beneficial to quantify the amount of homework in terms of the time that students spend on it, rather than the number of questions they are set, since there can be high variability in the amount of time it takes students to answer questions (Jarušek et al., 2013). We are therefore interested in selecting questions for students to answer in a fixed time limit. Given this limited time frame, we want to give the students questions that will most improve their learning early on, to make sure that they have enough time to complete them, before moving on to additional extension questions. For simplicity of the mathematical model, we do not assume that the order that students are given questions in the homework has an effect on the benefit from answering each individual question, although this is an interesting area for future work.

Online education software has the potential for adaptive learning strategies to be easily incorporated (Alshammari et al., 2014). These adaptive strategies are particularly useful when dealing with the problem of setting homework tasks with a fixed time limit, since we can develop strategies which are adaptive to the amount of time

remaining in the homework. This means that if the system suggests a question to a student that ends up taking a long time, the rest of the homework can be modified so that the student still achieves the optimal amount of learning given the time they have remaining to complete the homework.

When designing such a strategy, we must take into account the fact that both the amount of time it takes students to answer questions and the benefit they gain from answering each question are stochastic and will only be observed once we have asked the questions (although we can assume we have access to a generative model). Hence, we wish to select a sequence of questions that maximize the expected cumulative benefit to the student while satisfying the time constraint. This is mathematically equivalent to an instance of the stochastic knapsack problem. Here, the items are questions, with sizes corresponding to the amount of time it takes a student to answer the questions, and the rewards of each item is the benefit to the student of answering the question. The knapsack constraint is then the time limit of the homework task.

In Chapter 4, we present an algorithm for the stochastic knapsack problem built on the optimistic planning principle (see Section 2.3.7 for background of optimistic planning). This algorithm could be used to provide an adaptive sequence of questions for the student. We assume that this algorithm has access to a generative model of item sizes and rewards. In the educational setting this is a reasonable assumption since there has been much work in the literature on developing models of student performance (Corbett and Anderson, 1994; Hambleton and Swaminathan, 2013; Shahiri et al., 2015) and the time taken for students to answer questions (Jarušek and Pelánek, 2012; Jarušek et al., 2013; Ma et al., 2016). For our algorithm, a further assumption that is necessary to make is that the item size distributions are discrete, and that there are only a finite number of possible item sizes. In the educational setting, this is equivalent to having discrete response times and so an additional discretization step may be necessary in order to apply our algorithm. Our algorithm models the prob-

lem as a decision tree and returns a near-optimal *policy* which tells us which item to play (question to give) based on how long the past questions have taken. This policy is constructed offline so the idea is that once we have run the algorithm to get a near-optimal policy, this can be incorporated into education software to determine the adaptive sequence of homework questions to give to a student. Here we consider one student individually, however, many of the modeling approaches used to obtain the generative models combine information from different students.

3.4 Delay in the Effect of Answering Questions

Typically in an online education environment, students will answer many questions consecutively in a short period of time. The benefit to a student of answering each question is not normally immediate. Instead, it takes time for the information in a question to be consolidated into knowledge (Dudai et al., 2015; Cowan, 2008), and for us to observe that the student has learnt something. This means that when we observe an improvement (or decline) in their performance, it is often difficult to determine when the learning took place and exactly which of the past questions caused this effect. Particularly, this makes it difficult to assign credit to each question and determine the effectiveness of each question individually. However, it is reasonable to assume that when we see a change in the student's performance, that this is the aggregated effect of several past questions, the individual effects of which are delayed and only visible in this aggregate.

We consider a variant of the multi-armed bandit problem where the individual rewards are delayed and only visible as an aggregate in Chapter 5. This can then be related back to the education problem described above by setting each question as an arm and assuming that at time t we observe the summed reward of some number of questions asked previously. However, we do not learn which questions contributed to

this aggregated reward, nor the individual reward of each question. This setting lends itself naturally to a definition of reward that is directly observable. Particularly, it is common for students to see various questions on a topic and then be assessed on their knowledge of it through an end-of-module test or equivalent. In this case, it would be possible to define the aggregated reward from all the questions the student has seen as their score on this end-of-module test. We can assume that the reward from each individual question is delayed and only observed in this aggregate end-of-module test reward. With this definition of reward, the aim is to give the students questions that maximize their score in the end-of-module test, and this is also a reasonable aim pedagogically.

3.5 Allowing Time between Repetitions of a Question

Consider now the task of teaching students times tables via an app or other online environment. In this case, the ability of the student to recall the solution to a particular question (e.g. 6×3) will often depend on how long it has been since they last saw that question. Intuitively, if a student has just answered a question and are asked it again immediately, they will not learn as much as if we wait some time before asking it again. This phenomenon has been studied extensively in educational research (e.g. (Bahrick and Phelphs, 1987; Dempster, 1989)). A common approach is to assume that the rate at which the student forgets information is a function of how long it has been since they last saw the information, and this function is known as the *forgetting curve* (Ebbinghaus, 2013; Averell and Heathcote, 2011). An example of a forgetting curve is given in Figure 3.1. In the times tables context, each question may have an individual forgetting curve. There have been various theories developed about where on the forgetting curve it is best to revise each question (see (Cepeda et al., 2006)

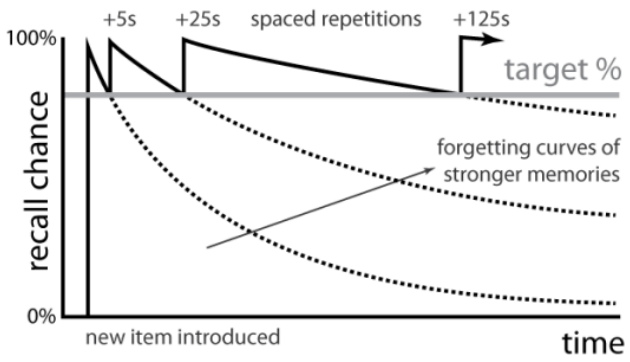


Figure 3.1: An example of a forgetting curve. Image taken from (Edge et al., 2012).

for a review of some of these). However, it is generally agreed that allowing time between repetitions of the same question leads to improved long-term retention of the information (Dempster, 1989). Hence, we wish to develop regimes that consider this forgetting curve and ask students questions at appropriately spaced intervals, corresponding to the points where they are likely to learn the most.

This problem has been studied in the educational literature and various approaches proposed. The most well known of these is spaced repetition (Wozniak and Gorzelanczyk, 1994; Cepeda et al., 2006; Edge et al., 2012; Reddy et al., 2016). Spaced repetition is a very general framework for devising a schedule which considers the spacing between repetitions of the same question. One particularly popular form of spaced repetition is the Leitner system (Leitner, 1995; Reddy et al., 2016). Here, there are several 'boxes' into which questions can be placed depending on how well they are assumed to be known. All questions start in the first box. When a question is selected, if it is answered correctly, it is moved into the next box along, while if it is answered incorrectly it is returned to the first box. Boxes are chosen according to some schedule which prioritizes the initial boxes, since these are the unknown questions. This technique, while simple, has proved immensely popular with many e-learning sites employing it (see references in (Reddy et al., 2016)). There have been many regimes proposed to learn the optimal frequency at which to take a question

from each box. For example, Pavlik and Anderson (2008) use a predictive model of performance, Reddy et al. (2016) use results from queuing theory, and several algorithms based on the SuperMemo algorithms (e.g. Biedka et al. (1998)) use neural networks.

In the literature, it is often assumed that the forgetting curve takes the form of an exponential or power law curve (Averell and Heathcote, 2011) and most spaced repetition, and alternative approaches, are built under this assumption. However, we will take a more general approach and directly model the expected reward of the question as an unknown function of the time since the question was last asked. We only assume that this function is smooth enough to be modeled by a Gaussian process. Note that this allows for the expected reward to increase with the time since it was last asked, but also to decrease if it has been too long since the question was asked. When we select a question, we assume that the reward we observe is given by this reward curve with additive Gaussian noise. In Chapter 6, we consider a stochastic K-armed bandit problem where the expected reward of each arm is modeled as a function of the time since each arm was last played, called the *recovery function*. We assume that these functions are sampled from a Gaussian process and present Thompson sampling and UCB algorithms for this problem. Our algorithms learn to play each arm when its expected reward is high, without needing knowledge of the functional form of the recovery curve. This corresponds to waiting an appropriate amount of time between plays of the same arm. We can apply this approach to the education problem by assuming that each question is an arm. This would ensure that questions are asked when their reward is high, corresponding to asking questions at appropriately spaced intervals.

Chapter 4

Optimistic Planning for the Stochastic Knapsack Problem

4.1 Introduction

The stochastic knapsack problem (Dantzig, 1957), is a classic resource allocation problem that consists of selecting a subset of items to place into a knapsack of given capacity. Placing each item in the knapsack consumes a random amount of the capacity and provides a stochastic reward. Many real world scheduling, investment, portfolio selection, and planning problems can be formulated as the stochastic knapsack problem. Consider, for instance, a fitness app that suggests a one hour workout to a user. Each exercise (item) will take a random amount of time (size) and burn a random amount of calories (reward). To make optimal use of the available time the app needs to track the progress of the user and adjust accordingly. Once an item is placed in the knapsack, we assume we observe its realized size and can use this to make future decisions. This enables us to consider adaptive or closed loop strategies, which will generally perform better (Dean et al., 2008) than open loop strategies in which the items chosen are invariant of the remaining budget. We assume that we do

not know the reward and size distributions of the items but are able to sample these from a generative model.

Finding exact solutions to the simpler deterministic knapsack problem, in which item weights and rewards are deterministic, is known to be NP-hard and it has been stated that the stochastic knapsack problem is PSPACE-hard (Dean et al., 2008). Due to the difficulty of the problem, there are currently no algorithms that are guaranteed to find satisfactory approximations in acceptable computation time. While ultimately one aims to have algorithms that can approach large scale problems, the current state-of-the-art makes it apparent that the small scale stochastic knapsack problem must be tackled first. The emphasis in this chapter is therefore on this small scale stochastic knapsack setting.

The current state-of-the-art approaches to the stochastic knapsack problem where the reward and size distributions are known, were introduced in (Dean et al., 2008). Their algorithm splits the items into small and large items and fills the knapsack exclusively with items of one of the two groups, ignoring potentially good items in the other group. This returns a solution that comes within a factor of $1/(3+\kappa)$ of the optimal, where $\kappa > 0$ is used to set a threshold for the small items. The strategy for small items is non-adaptive and places items in the knapsack according to their reward - consumption ratio. For the large items, a decision tree is built to some predefined depth and an exhaustive search for the best solution in that decision tree is performed. For most non-trivial problems, this tree can be exceptionally large. The notion of small items is also underlying recent work in machine learning where the reward and consumption distributions are assumed to be unknown (Badanidiyuru et al., 2013). The approach in (Badanidiyuru et al., 2013) works with a knapsack size that converges (in a suitable way) to infinity, rendering all items small. In (Burnetas et al., 2015) adaptive strategies are considered for deterministic item sizes and renewable capacities. The stochastic knapsack problem is also a generalization of the pure

exploration combinatorial bandit problem (Chen et al., 2014; Gabillon et al., 2016).

It is desirable to have methods for the stochastic knapsack problem that can make use of all available resources and adapt to the remaining capacity. For this, the tree structure from (Dean et al., 2008) can be useful. We propose using ideas from optimistic planning (Busoniu and Munos, 2012; Szörényi et al., 2014) to significantly accelerate the tree search approach and find adaptive strategies. Most optimistic planning algorithms were developed for discounted MDPs and as such rely on discount factors to limit future rewards, effectively reducing the search tree to a tree with small depth. However, these discount factors are not present in the stochastic knapsack problem. Furthermore, in our problem, the random variables representing state transitions (item sizes) also provide us with information on the remaining capacity which relates to possible future rewards. To avoid the use of discount factors and use this transition information, we work with confidence bounds that incorporate estimates of the remaining capacity. We also use these estimates to determine how many samples we need from the generative model of the reward/size of an item.

For this, we need techniques that can deal with weak dependencies and give confidence regions that hold simultaneously for multiple sample sizes. We therefore combine Doob's martingale inequality (Doob, 1953) with Azuma-Hoeffding bounds (Azuma, 1967) to create our high probability bounds. Following the optimistic planning approach, we use these bounds to develop an algorithm that adapts to the complexity of the problem instance. In contrast to the current state-of-the-art, it is guaranteed to find an ϵ-good approximation for all problem instances and, if the problem instance is easy to solve, it expands only a moderate sized tree. Our algorithm, `OpStoK`, is also an 'anytime' algorithm in the sense that it improves rapidly to begin with and, even if stopped prematurely, it will still return a good solution. For `OpStoK`, we only require access to a generative model of item sizes and rewards, and no further knowledge of the distributions.

A solution to the stochastic knapsack problem will take the form of a policy. A policy can be thought of as a sub-tree or a set of rules telling us which item to play next depending on previous item sizes (see Section 4.A.1 for examples). We define the value of policy to be its expected cumulative reward and seek to find policies whose value is within ϵ of the optimal value. The performance of our algorithm is measured in terms of the number of policies it expands in order to find such an ϵ-optimal policy, since this quantity relates to the run-time and complexity. In practice, the number of policies explored by our algorithm `OpStoK` is small and compares favorably to that of Dean et al. (2008).

4.1.1 Related Work

Due to the difficulty of the stochastic knapsack problem, the main approximation algorithms focus on the variant of the problem with deterministic sizes and stochastic rewards (eg. Steinberg and Parks (1979); Morton and Wood (1998)), or stochastic sizes and deterministic rewards (eg. Dean et al. (2008); Bhalgat et al. (2011)), where the relevant distributions are known. Of these, the most relevant works to our are (Dean et al., 2008) and (Bhalgat et al., 2011) where decision trees are used to obtain approximate adaptive solutions. To limit the size of the decision tree, Dean et al. (2008) use a greedy strategy for 'small' items while Bhalgat et al. (2011) group items together. Morton and Wood (1998) use a Monte-Carlo sampling strategy to generate a non-adaptive solution in the case with stochastic rewards and deterministic sizes.

The UCT style of bandit based tree search algorithms (Kocsis and Szepesvári, 2006) uses upper confidence bounds at each node of the tree to select the best action. UCT has been shown to work in practice, however, it may be too optimistic (Coquelin and Munos, 2007).

Optimistic planning was developed for tree search in large deterministic (Hren and Munos, 2008) and stochastic systems, both open (Bubeck and Munos, 2010)

and closed loop (Busoniu and Munos, 2012). The general idea is to use the upper confidence principle of the UCB algorithm for multi-armed bandits (see Chapter 2 for an introduction to the multi-armed bandit problem) to expand a tree. This is achieved by expanding nodes that have the potential to lead to good solutions, using bounds that take into account both the reward received in getting to a node and the reward that could be obtained after moving on from that node. An overview of optimistic planning and a more detailed discussion of the related work is given in Section 2.3.7.

The closest work to ours is that of Szörényi et al. (2014) who use optimistic planning in discounted MDPs, requiring only a generative model of the rewards and transitions. Instead of the UCB algorithm, like ours, their work relies on the best arm identification algorithm of Gabillon et al. (2012). However, there are several key differences between our problem and the MDPs optimistic planning algorithms are typically designed for. Generally, in optimistic planning it is assumed that the state transitions do not provide any information about future reward. However, in the stochastic knapsack problem this information is relevant and should be taken into account when defining the high confidence bounds. Furthermore, optimistic planning algorithms are typically used to approximate complex systems at just one point and so only return a near optimal first action. In our case, the decision tree is a good approximation to the entire problem, so we output a near-optimal policy. Furthermore, to the best of our knowledge, our algorithm is the first optimistic planning algorithm to iteratively build confidence bounds which are used to determine whether it is necessary to sample more. One would imagine that the `StOP` algorithm from (Szörényi et al., 2014) could be easily adapted to the stochastic knapsack problem. However, as discussed in Section 4.4.1, the assumptions required for this algorithm to terminate are too strong for it to be considered feasible for this problem.

4.1.2 Our Contribution

Our main contributions are the anytime algorithm `OpStoK` (Algorithm 4.1) and subroutine `BoundValueShare` (Algorithm 4.2). These are supported by the confidence bounds in Proposition 4.2 that allow us to simultaneously estimate remaining capacity and value with guarantees that hold uniformly over multiple sample sizes. Proposition 4.4 shows how we can avoid discount based arguments and use adaptive capacity estimates in our algorithm, and still return an adaptive policy whose value comes within ϵ of the optimal policy with high probability. Theorem 4.5 and Corollary 4.6 provide bounds on the number of samples our algorithm uses in terms of how many policies are ϵ-close to the best policy. The empirical performance of `OpStoK` is considered in Section 4.7.

4.2 Problem Formulation

We consider the problem of selecting a subset of items from a set, I, of K items, to place into a knapsack of capacity (or budget) B where each item can be played at most once. For each item $i \in I$, let C_i and R_i be non-negative, bounded random variables defined on a joint probability space $(\Omega, \mathcal{A}, P)$ which represent its size and reward. It is assumed that we can simulate from the generative model of (R_i, C_i) for all $i \in I$ and we will use lower case c_i and r_i, to denote realizations of these random variables. We assume that the random variables (R_i, C_i) are independent of (R_j, C_j) for all $i, j \in I$, $i \neq j$. Further, it is believed that item sizes and rewards do not change depending on the other items in the knapsack. We assume the problem is non-trivial, in the sense that it is not possible to fit all items in the knapsack at once. If we place an item i in the knapsack and the consumption c_i is strictly greater than the remaining capacity then we gain no reward for that item. Our final important assumption is that there exists a known, non-decreasing function $\Psi(\cdot)$, satisfying $\lim_{b \to 0} \Psi(b) = 0$

and $\Psi(B) < \infty$, such that the total reward that can be achieved with budget b is upper bounded by $\Psi(b)$. It will always be possible to define such a Ψ, however, the choice of Ψ will impact the performance of the algorithm, so we will choose it to be as tight as possible.

Representing the stochastic knapsack problem as a tree requires that all item sizes take discrete values. While in this work, it will generally be assumed that this is the case, in some problem instances, continuous item sizes need to be discretized. In this case, let ξ^* be the discretization error of the optimal policy. Then $\Psi(\xi^*)$ is an upper bound on the extra reward that could be gained from the space lost due to discretization. For discrete sizes, we assume there are s possible values the random variable C_i can take and that there exists $\theta > 0$ such that $C_i \geq \theta$ for all $i \in I$.

4.2.1 Planning Trees and Policies

The stochastic knapsack problem can be thought of as a planning tree with the initial empty state as the root at level 0. The branches from the root represent playing an item. Similarly, each node on an even level is an *action* node and its branches represent placing an item in the knapsack. The nodes on odd levels are *transition* nodes with branches representing item sizes. We define a *policy* Π as a finite subtree where each action node has at most one branch from it and each transition node has s branches (see Section 4.A.1 for examples). The *depth* of a policy Π, $d(\Pi)$, is the number of transition nodes in any realization of the policy (where each transition node links to one branch), or equivalently, the number of items. Let $d^* = \lfloor B/\theta \rfloor$ be the maximal depth of any policy. For any $1 \leq d \leq d^*$, the number of policies of depth d is,

$$N_d = \prod_{i=0}^{d-1} (K - i)^{s^i} \tag{4.1}$$

where $K = |I|$ is the number of items, and s the number of discrete sizes.

We define a *child* policy, Π', of a policy Π as a policy that follows Π up to depth $d(\Pi)$ then plays additional items and has depth $d(\Pi') = d(\Pi) + 1$. We say Π is the *parent* policy of Π'. A policy Π' is a *descendant* policy of Π if Π' follows Π up to depth $d(\Pi)$ but is then continued to depth $d(\Pi') \geq d(\Pi)+1$. Correspondingly, we say Π is an *ancestor* of Π'. A policy is said to be *incomplete* if the remaining capacity allows for another item to be inserted into the knapsack (see Section 4.4.2 for a formal definition). Note that the policy an algorithm outputs may be incomplete, as it could be that any continuation of it is optimal.

The *(expected) value* of a policy Π is defined as the cumulative expected reward obtained by playing items according to Π, $V_\Pi = \sum_{d=1}^{d(\Pi)} E[R_{i(d)}]$ where $i(d)$ is the d-th item chosen by Π. Let $\mathcal{P}$ be the set of all policies, then define the *optimal policy* as $\Pi^* = \arg\max_{\Pi\in\mathcal{P}} V_\Pi$, and corresponding *optimal value* as $v^* = \max_{\Pi\in\mathcal{P}} V_\Pi$. Our algorithm returns an ϵ-*optimal* policy with value $v^* - \epsilon$. For any policy Π, we define a *sample* of Π as follows. The first item of any policy is fixed so we take a sample of the reward and size from the generative model of that item. We then use Π and the observed size of the previous item to tell us which item to sample next and sample the reward and size of that item. This continues until the policy finishes or the cumulative sampled sizes of the selected items exceeds B.

4.3 High Confidence Bounds

In order to select policies to expand, we require confidence bounds for the value of a continuation of a policy. A policy Π may not consume all available budget, and our algorithm will work by constructing iteratively longer policies, starting from the shortest policies of playing a single item. Consequently, we are interested in R_Π^+, the expected maximal extra reward that can be obtained after playing according to

policy Π until all the budget is consumed. Let B_Π be a random variable representing the remaining budget after playing policy Π. Our assumptions guarantee that there exists a function Ψ such that $R_\Pi^+ \leq E\Psi(B_\Pi)$. We then define V_Π^+ to be the maximal expected value of any continuation of policy Π, so $V_\Pi^+ = V_\Pi + R_\Pi^+ \leq V_\Pi + E\Psi(B_\Pi)$.

From m_1 samples of the value of policy Π, we estimate the true value of Π as $\overline{V_\Pi}_{m_1} = \frac{1}{m_1}\sum_{j=1}^{m_1}\sum_{d=1}^{d(\Pi)} r_{i(d)}^{(j)}$, where $r_{i(d)}^{(j)}$ is the reward of item $i(d)$ chosen at depth d of sample j. However, we wish to identify the policy with greatest value when continued until the budget is exhausted, so our real interest is in the value of V_Π^+. From Hoeffding's inequality, $P\left(|\overline{V_\Pi}_{m_1} - V_\Pi^+| > E\Psi(B_\Pi) + \sqrt{\frac{\Psi(B)^2\log(2/\delta)}{2m_1}}\right) \leq \delta$. This bound depends on the quantity $E\Psi(B_\Pi)$ which is typically not known. Lemma 4.1 shows how this bound can be significantly improved by independently sampling B_Π m_2 times to get samples $\psi_1, \cdots, \psi_{m_2}$ of $\Psi(B_\Pi)$ and estimating $\overline{\Psi(B_\Pi)}_{m_2} = \frac{1}{m_2}\sum_{j=1}^{m_2}\psi_j$.

Lemma 4.1. *Let $(\Omega, \mathcal{A}, P)$ be the probability space from Section 4.2, then for m_1+m_2 independent samples of policy Π and $\delta_1, \delta_2 > 0$, with probability $1-\delta_1-\delta_2$,*

$$\overline{V_\Pi}_{m_1} - k_1 \leq V_\Pi^+ \leq \overline{V_\Pi}_{m_1} + \overline{\Psi(B_\Pi)}_{m_2} + k_1 + k_2.$$

Where, $k_1 := \sqrt{\frac{\Psi(B)^2\log(2/\delta_1)}{2m_1}}$, $k_2 := \sqrt{\frac{\Psi(B)^2\log(1/\delta_2)}{2m_2}}$.

We will not use the bound in this form since our algorithm will sample $\Psi(B_\Pi)$ until we are sufficiently confident that it is small or large. This introduces weak dependencies into the sampling process so we need guarantees to hold simultaneously for multiple sample sizes, m_2. For this, we work with martingales and use Azuma-Hoeffding like bounds (Azuma, 1967), similar to the technique used in (Perchet et al., 2016). Specifically, in Lemma 4.8 (Section 4.B), we use Doob's maximal inequality (Doob, 1953) and a peeling argument to get bounds on the maximal deviation of $\overline{\Psi(B_\Pi)}_{m_2}$ from its expectation. Assuming we sample the value of a policy m_1 times and the remaining budget m_2 times, the following key result holds.

Proposition 4.2. *The Algorithm* `BoundValueShare` *(Algorithm 4.2) returns confidence bounds,*

$$L(V_\Pi^+) = \overline{V_\Pi}_{m_1} - c_1$$
$$U(V_\Pi^+) = \overline{V_\Pi}_{m_1} + \overline{\Psi(B_\Pi)}_{m_2} + c_1 + c_2$$

with $c_1 = \sqrt{\frac{\Psi(B)^2 \log(2/\delta_1)}{2m_1}}, c_2 = 2\Psi(B)\sqrt{\frac{1}{m_2}\log\left(\frac{8n}{\delta_2 m_2}\right)}$ *which hold with probability* $1 - \delta_1 - \delta_2$.

This upper bound depends on n, the maximum number of samples of $\Psi(B_\Pi)$. For any policy Π, the minimum width a confidence interval of $\Psi(B_\Pi)$ will ever need to be is $\epsilon/4$. Hence, taking

$$n = \left\lceil \frac{16^2 \Psi(B)^2 \log(8/\delta)}{\epsilon^2} \right\rceil, \tag{4.2}$$

ensures that for all policies, $2c_2 \leq \epsilon/4$ when $m_2 = n$. This is a necessary condition for the termination of our algorithm, `OpStoK`, as will be discussed in Section 4.4.2

4.4 Algorithms

Before presenting our algorithm for optimistic planning for the stochastic knapsack problem, we first discuss a simple adaptation of the algorithm `StOP` from Szörényi et al. (2014).

4.4.1 Stochastic Optimistic Planning for Knapsacks

One naive approach to optimistic planning in the stochastic knapsack problem is to adapt the algorithm `StOP` from (Szörényi et al., 2014). We call this adaptation `StOP-K` and replace the $\frac{\gamma^d}{1-\gamma}$ discounting term used to control future rewards with $\Psi(B - d\theta)$.

This is the best upper bound on the future reward that can be achieved without using samples of item sizes. The upper bound on V_{Π}^{+} is then $\overline{V_{\Pi}}_m + \Psi(B - d\theta) + c$, for m samples and confidence bound c. With this, most of the results from (Szörényi et al., 2014) follow fairly naturally. Although `StOP-K` appears to be an intuitive extension of `StOP` to the stochastic knapsack setting, it can be shown that for a finite number of samples, unless $\Psi(B - \theta d^*) \leq \frac{\epsilon}{2}$, the algorithm will not terminate. As such, unless this restrictive assumption is satisfied `StOP-K` will not converge.

4.4.2 Optimistic Stochastic Knapsacks

In `OpStoK` we aim to be more efficient by only exploring promising policies and making better use of all information. In the stochastic knapsack problem, in order to sample the value of a policy, we must sample item sizes to decide which item to play next. We propose to also use the item size samples to calculate $U(\Psi(B_{\Pi}))$, and then incorporate this into $U(V_{\Pi}^{+})$. We also pool samples of the reward and size of items across policies, thus reducing the number of calls to the generative model. `OpStoK` benefits from an adaptive sampling scheme that reduces sample complexity and ensures that an entire ϵ-optimal policy is returned when the algorithm stops. The performance of this sampling strategy is guaranteed by Proposition 4.2.

In the main algorithm, `OpStoK` (Algorithm 4.1) is very similar to `StOP-K` (Szörényi et al., 2014) with the key differences appearing in the sampling and construction of confidence bounds which are defined in `BoundValueShare` (Algorithm 4.2). The general intuition is that only promising policies are explored. `OpStoK` maintains a set of 'active' policies. As in (Szörényi et al., 2014) and (Gabillon et al., 2012), at each time step t, a policy, Π_t to expand is chosen by comparing the upper confidence bounds of the two best active policies. We select the policy with most uncertainty in the bounds since we want our estimates of the near-optimal policies to be such that we can confidently conclude that the policy we output is better (see Figure 4.5,

Section 4.A.2). Once we have selected a policy, Π_t, if the stopping criteria in Line 10 is not met, we replace Π_t in the set of active policies with all its children. We refer to this as *expanding* a policy. For each child policy, Π', we bound its value using `BoundValueShare` with parameters

$$\delta_{d(\Pi'),1} = \frac{\delta_{0,1}}{d^*} N^{-1}_{d(\Pi')} \text{ and, } \delta_{d(\Pi'),2} = \frac{\delta_{0,2}}{d^*} N^{-1}_{d(\Pi')} \tag{4.3}$$

where N_d is the number of policies of depth d as given in (4.1). This ensures that all our bounds hold simultaneously with probability greater than $1-\delta_{0,1}-\delta_{0,2}$ (as shown in Lemma 4.12, Section 4.B). The algorithm stops in Line 10 and returns a policy Π^* if $L(V^+_{\Pi^*}) + \epsilon \geq \max_{\Pi \in \text{Active} \setminus \{\Pi^*\}} U(V^+_\Pi)$ and we can be confident Π^* is within ϵ of optimal. `OpStoK` relies on `BoundValueShare` (Algorithm 4.2) and subroutines, `EstimateValue` and `SampleBudget` (Algorithms 4.4 and 4.3, Section 4.A.3), which sample the value and budget of policies.

In `BoundValueShare`, we use samples of both item size and reward to bound the value of a policy. We define upper and lower bounds on the value of any extension of a policy Π as,

$$\begin{aligned} U(V^+_\Pi) &= \overline{V_\Pi}_{m_1} + \overline{\Psi(B_\Pi)}_{m_2} + c_1 + c_2, \\ L(V^+_\Pi) &= \overline{V_\Pi}_{m_1} - c_1, \end{aligned}$$

with c_1 and c_2 as in Proposition 4.2. It is also possible to define upper and lower bounds on $\Psi(B_\Pi)$ with m_2 samples and confidence δ_2. From this, we can formally define a *complete* policy as a policy Π with $U(\Psi(B_\Pi)) = \overline{\Psi(B_\Pi)}_{m_2} + c_2 \leq \frac{\epsilon}{2}$. For complete policies, since there is very little capacity left, it is more important to get tight confidence bounds on the value of the policy. Hence, in `BoundValueShare`, we sample the remaining budget of a policy as much as is necessary to conclude whether the policy is complete or not. As soon as we realize we have a complete policy

($U(\Psi(B_\Pi)) \leq \epsilon/2$), we sample the value of that policy sufficiently to get a confidence interval on V_Π^+ of width less than ϵ. Then, when it comes to choosing an optimal policy to return, the confidence intervals of all complete policies will be narrow enough for this to happen. This is appropriate since pre-specifying the number of samples may not lead to confidence bounds tight enough to select an ϵ-optimal policy. Furthermore, we focus sampling efforts only on promising policies that are near completion.

If a complete policy is chosen as $\Pi_t^{(1)}$ in `OpStoK`, for some t, the algorithm will stop and this policy will be returned. For this to happen, we check the stopping criterion before selecting a policy to expand. Note that in `BoundValueShare`, the value and remaining budget of a policy must be sampled separately as we are considering closed-loop planning so the item chosen may depend on the size of the previous item, and hence the value will depend on the instantiated item sizes. For an incomplete policy, the number of samples of the value, m_1, is defined to ensure that the uncertainty in the estimate of V_Π is less than $u(\Psi(B_\Pi)) = \min\{U(\Psi(B_\Pi)), \Psi(B)\}$, since a maximal upper bound for the value of Π is $\Psi(B)$.

Since at each time step `OpStoK` expands the policy with best or second best upper confidence bound, the policy it expands will always have the potential to be optimal. Therefore, if the algorithm is stopped before the termination criteria is met and the active policy with best estimated value is selected, this policy will be the best of those with the potential to be optimal that have already been explored. Hence, it will be a good policy (or beginning of policy). `OpStoK` considerably reduces the number of calls to the generative model by creating sets $\mathcal{S}_i^*$ of samples of the reward and size of each item $i \in I$. When it is necessary to sample the reward and size of an item, i, for the evaluation of a policy, we sample without replacement from $\mathcal{S}_i^*$ until $|\mathcal{S}_i^*|$ samples have been taken. At this point new calls to the generative model are made and the new samples added to the sets for use by future policies. This is illustrated in `EstimateValue` and `SampleBudget` (Algorithms 4.4 and 4.3, Section 4.A.3). We

Algorithm 4.1: `OpStoK` $(I, \delta_{0,1}, \delta_{0,2}, \epsilon)$

Initialization: $\text{ACTIVE} = \emptyset$.
for all $i \in I$ **do**
  Π_i = policy consisting of just playing item i;
  $d(\Pi_i) = 1$, $\delta_{1,1} = \frac{\delta_{0,1}}{d^*} N_1^{-1}$, $\delta_{1,2} = \frac{\delta_{0,2}}{d^*} N_1^{-1}$;
  $(L(V^+_{\Pi_i}), U(V^+_{\Pi_i})) =$ `BoundValueShare`$(\Pi_i, \delta_{1,1}, \delta_{1,2}, \mathcal{S}^*, \epsilon)$;
  $\text{ACTIVE} = \text{ACTIVE} \cup \{\Pi_i\}$;
end
for $t = 1, 2, \ldots$ **do**
  $\Pi_t^{(1)} = \arg\max_{\Pi \in \text{Active}} U(V^+_\Pi)$, $\Pi_t^{(2)} = \arg\max_{\Pi \in \text{Active} \setminus \{\Pi_t^{(1)}\}} U(V^+_\Pi)$;
  if $L(V^+_{\Pi_t^{(1)}}) + \epsilon \geq U(V^+_{\Pi_t^{(2)}})$ **then**
    Stop: $\Pi^* = \Pi_t^{(1)}$;
  $a^* = \arg\max_{a \in \{1,2\}} U(\Psi(B_{\Pi_t^{(a)}}))$;
  $\Pi_t = \Pi_t^{(a^*)}$;
  $\text{ACTIVE} = \text{ACTIVE} \setminus \{\Pi_t\}$
  for all *children* Π' *of* Π_t **do**
    $d(\Pi') = d(\Pi_t) + 1$;
    $\delta_{d(\Pi'),1} = \frac{\delta_{0,1}}{d^*} N^{-1}_{d(\Pi')}, \delta_{d(\Pi'),2} = \frac{\delta_{0,2}}{d^*} N^{-1}_{d(\Pi')}$
    $(L(V^+_{\Pi'}), U(V^+_{\Pi'})) =$ `BoundValueShare`$(\Pi', \delta_{d(\Pi'),1}, \delta_{d(\Pi'),2}, \mathcal{S}^*, \epsilon)$;
    $\text{ACTIVE} = \text{ACTIVE} \cup \{\Pi'\}$;
  end
end

denote by $\mathcal{S}^*$ the collection of all sets $\mathcal{S}_i^*$.

4.5 ϵ-Critical Policies

The set of ϵ-critical policies associated with an algorithm is the set of all policies the algorithm may potentially expand in order to obtain an ϵ-optimal solution. Hence, the number of ϵ-critical policies represents a bound on the number of policies an algorithm may explore in order to obtain this ϵ-optimal solution.

Algorithm 4.2: `BoundValueShare`$(\Pi, \delta_1, \delta_2, S^*, \epsilon)$

Initialization: For all $i \in I$, $\mathcal{S}_i = \mathcal{S}_i^*$.

Set $m_2 = 1$ and $(\psi_1, \mathcal{S}) =$ `SampleBudget`$(\Pi, \mathcal{S})$;

`/* sample the remaining budget */`

$\overline{\Psi(B_\Pi)}_{m_2} = \frac{1}{m_2}\sum_{j=1}^{m_2}\psi_j$;

$U(\Psi(B_\Pi)) = \overline{\Psi(B_\Pi)}_{m_2} + 2\Psi(B)\sqrt{\frac{1}{m_2}\log\left(\frac{8n}{\delta m_2}\right)}$,

$L(\Psi(B_\Pi)) = \overline{\Psi(B_\Pi)}_{m_2} - 2\Psi(B)\sqrt{\frac{1}{m_2}\log\left(\frac{8n}{\delta m_2}\right)}$;

`/* calculate bounds on` $\Psi(B_\Pi)$ `*/`

if $U(\Psi(B_\Pi)) \leq \frac{\epsilon}{2}$ **then** $m_1 = \left\lceil \frac{8\Psi(B)^2\log(2/\delta_1)}{\epsilon^2} \right\rceil$;

else if $L(\Psi(B_\Pi)) \geq \frac{\epsilon}{4}$ **then**

  $m_1 = \left\lceil \frac{1}{2}\frac{\Psi(B)^2\log(2/\delta_1)}{u(\Psi(B_\Pi))^2} \right\rceil$;

else

  Set $m_2 = m_2 + 1$;

  $(\psi_{m_2}, \mathcal{S}) =$ `SampleBudget`$(\Pi, \mathcal{S})$ and go to **2**

$\overline{V_\Pi}_{m_1} =$ `EstimateValue`(Π, m_1);

$L(V_\Pi^+) = \overline{V_\Pi}_{m_1} - \sqrt{\frac{\Psi(B)^2\log(2/\delta_1)}{2m_1}}$;

$U(V_\Pi^+) = \overline{V_\Pi}_{m_1} + \overline{\Psi(B_\Pi)}_{m_2} + \sqrt{\frac{\Psi(B)^2\log(2/\delta_1)}{2m_1}} + 2\Psi(B)\sqrt{\frac{1}{m_2}\log\left(\frac{8n}{\delta m_2}\right)}$;

return $(L(V_\Pi^+), U(V_\Pi^+))$

To define the set of ϵ-critical policies associated with `OpStoK`, let

$$\mathcal{Q}_{IC}^\epsilon = \{\Pi; V_\Pi + 9E\Psi(B_\Pi) - 3\epsilon/4 \geq v^* - 9E\Psi(B_\Pi) + 3\epsilon/4 + \epsilon\}$$

$$\text{and } \mathcal{Q}_C^\epsilon = \{\Pi; V_\Pi + \epsilon \geq v^*\},$$

represent the set of potentially optimal incomplete and complete policies. The set of all ϵ-critical policies is then $\mathcal{Q}^\epsilon = \mathcal{Q}_{IC}^\epsilon \bigcup \mathcal{Q}_C^\epsilon$. The following lemma shows that all policies expanded by `OpStoK` are in $\mathcal{Q}^\epsilon$.

Lemma 4.3. *Assume that $L(V_\Pi^+) \leq V_\Pi \leq U(V_\Pi^+)$ holds simultaneously for all policies $\Pi \in$ Active with $U(V_\Pi^+)$ and $L(V_\Pi^+)$ as defined in Proposition 4.2. Then, $\Pi_t \in \mathcal{Q}^\epsilon$ for every policy, Π_t, selected by `OpStoK` at every time point t, except for possibly the last one.*

We now turn to demonstrating that under certain conditions, `OpStoK` will not expand all policies (although in practice this claim should hold even when some of the assumptions are violated). From considering the definition of $\mathcal{Q}^{\epsilon}_{IC}$ above, it can be shown that if there exists a subset I' of items and $\lambda > 0$ satisfying,

$$\begin{aligned} \sum_{i \in I'} E[R_i] < v^* - \epsilon, \quad \text{and,} \\ E\left[\Psi\left(B - \sum_{i \in I'} C_i\right)\right] < \frac{5\epsilon}{36} - \frac{\lambda}{18} \end{aligned} \tag{4.4}$$

then $\mathcal{Q}^{\epsilon}_{IC}$ is a proper subset of all incomplete policies and as such, not all incomplete policies will need to be evaluated by `OpStoK`. Furthermore, since any policy of depth $d > 1$ will only be evaluated by `OpStoK` if a descendant of it has previously been evaluated, it follows that a complete policy in $\mathcal{Q}^{\epsilon}_{C}$ must have an incomplete descendant in $\mathcal{Q}^{\epsilon}_{IC}$. Therefore, since $\mathcal{Q}^{\epsilon}_{IC}$ is not equal to the set of all incomplete policies, $\mathcal{Q}^{\epsilon}_{C}$ will also be a proper subset of all complete policies and so $\mathcal{Q}^{\epsilon} \subsetneq \mathcal{P}$. Note that the bounds used to obtain these conditions are worst case as they involve assuming the true value of $\Psi(B_{\Pi})$ lies at one extreme of the confidence interval. Hence, even if the conditions in (4.4) are not satisfied, it is unlikely that `OpStoK` will evaluate all policies. However, the conditions in (4.4) are easily satisfied. Consider, for example, the problem instance where $\epsilon = 0.25, \Psi(b) = b \quad \forall 0 \leq b \leq B, v^* = 1$ and $B = 1$. Assume there are 3 items $i_1, i_2, i_3 \in I$ with $E[R_i] < 1/4$ and $E[C_i] = 12/37$. Then if $I' = \{i_1, i_2, i_3\}$ and $\lambda = 1/8$, the conditions of (4.4) are satisfied and `OpStoK` will not evaluate all policies.

4.6 Analysis

In this section we give theoretical guarantees on the performance of `OpStoK`, with the proofs of all results in Section 4.B. We begin with the consistency result:

Proposition 4.4. *For $\epsilon > 0$, with probability at least $(1 - \delta_{0,1} - \delta_{0,2})$, the algorithm* `OpStoK` *returns a policy with value at least $v^* - \epsilon$.*

To obtain a bound on the sample complexity of `OpStoK`, we return to the definition of ϵ-critical policies from Section 4.5. The set of ϵ-critical policies, $\mathcal{Q}^\epsilon$, can be represented as the union of three disjoint sets, $\mathcal{Q}^\epsilon = \mathcal{A}^\epsilon \cup \mathcal{B}^\epsilon \cup \mathcal{C}^\epsilon$, as illustrated in Figure 4.1 where $\mathcal{A}^\epsilon = \{\Pi \in \mathcal{Q}^\epsilon | E\Psi(B_\Pi) \leq \epsilon/4\}, \mathcal{B}^\epsilon = \{\Pi \in \mathcal{Q}^\epsilon | E\Psi(B_\Pi) \geq \epsilon/2\}$ and $\mathcal{C}^\epsilon = \{\Pi \in \mathcal{Q}^\epsilon | \epsilon/4 < E\Psi(B_\Pi) < \epsilon/2\}$. Using this, in Theorem 4.5 the total number of samples of item size or reward required by `OpStoK` can be bounded as follows.

Theorem 4.5. *With probability greater than $1 - \delta_{0,2}$, the total number of samples required by* `OpStoK` *is bounded from above by,*

$$\sum_{\Pi \in \mathcal{Q}^\epsilon} (m_1(\Pi) + m_2(\Pi))\, d(\Pi).$$

$$\textit{Where, for } \Pi \in \mathcal{A}^\epsilon, m_1(\Pi) = \left\lceil 8\Psi(B)^2 \log(\tfrac{2}{\delta_{d(\Pi),1}}) / \epsilon^2 \right\rceil,$$

$$\textit{for } \Pi \in \mathcal{B}^\epsilon, m_1(\Pi) \leq \left\lceil \Psi(B)^2 \log(\tfrac{2}{\delta_{d(\Pi),1}}) / 2E\Psi(B_\Pi)^2 \right\rceil,$$

$$\textit{and for } \Pi \in \mathcal{C}^\epsilon, m_1(\Pi) \leq \max\left\{ \left\lceil 8\Psi(B)^2 \log(\tfrac{2}{\delta_{d(\Pi),1}}) / \epsilon^2 \right\rceil, \left\lceil 2\Psi(B)^2 \log(\tfrac{2}{\delta_{d,1}}) / E\Psi(B_\Pi)^2 \right\rceil \right\}.$$

And $m_2(\Pi) = m^*$, where m^* is the smallest integer satisfying,

$$32\Psi(B)^2/(E\Psi(B_\Pi) - \epsilon/2)^2 \leq m/\log(4n/m\delta_2) \textit{ for } \Pi \in \mathcal{A}^\epsilon,$$

$$32\Psi(B)^2/(E\Psi(B_\Pi) - \epsilon/4)^2 \leq m/\log(4n/m\delta_2) \textit{ for } \Pi \in \mathcal{B}^\epsilon,$$

$$32\Psi(B)^2/(\epsilon/4)^2 \leq m/\log(4n/m\delta_2) \textit{ for } \Pi \in \mathcal{C}^\epsilon.$$

We now bound the number of calls to the generative model required by `OpStoK`. We consider the expected number of times item i needs to be sampled by a policy Π. Let $i_1, \ldots, i_q$ denote the q nodes in policy Π where item i is played. Then for each

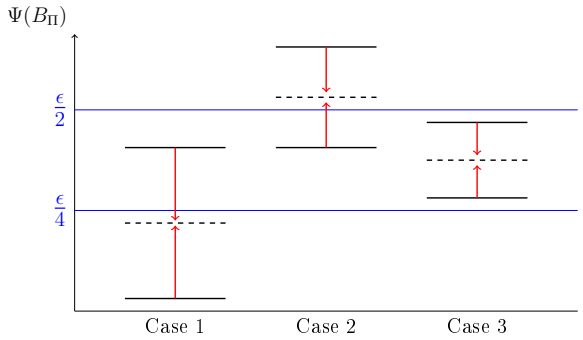


Figure 4.1: The three possible cases of $E\Psi(B_\Pi)$. In the first case, $E\Psi(B_\Pi) \leq \frac{\epsilon}{4}$ so $\Pi \in \mathcal{A}^\epsilon$, in the second case $E\Psi(B_\Pi) \geq \frac{\epsilon}{2}$ so $\Pi \in \mathcal{B}^\epsilon$, and in the final case $\frac{\epsilon}{4} < E\Psi(B_\Pi) < \frac{\epsilon}{2}$ so $\Pi \in \mathcal{C}^\epsilon$.

node i_k $(1 \leq k \leq q)$, denote by ζ_{i_k} the unique route to node i_k. Define $d(\zeta_{i_k})$ to be the depth of node i_k, or the number of items played along route ζ_{i_k}. Then the probability of reaching node i_k (or taking route ζ_{i_k}) is $P(\zeta_{i_k}) = \prod_{\ell=1}^{d(\zeta_{i_k})} p_{\ell,\Pi}(i_{k,\ell})$, where $i_{k,\ell}$ denotes the ℓth item on the route to node i_k and $p_{l,\Pi}(i)$ is the probability of playing item i at depth l of policy Π for given size distributions. Denote the probability of playing item i in policy Π by $P_\Pi(i)$, then $P_\Pi(i) = \sum_{k=1}^{q} P(\zeta_{i_k})$. Using this, the expected number of samples of the reward and size of item i required by policy Π are less than $m_1(\Pi)P_\Pi(i)$ and $m_2(\Pi)P_\Pi(i)$, respectively. Since samples are shared between policies, the expected number of calls to the generative model of item i is as given below and used in Corollary 4.6,

$$M(i) \leq \max_{\Pi \in \mathcal{Q}^\epsilon} \Big\{ \max\{m_1(\Pi)P_\Pi(i), m_2(\Pi)P_\Pi(i)\} \Big\}.$$

Corollary 4.6. *The expected total number of calls to the generative model by* `OpStoK` *for a stochastic knapsack problem of K items is less than or equal to $\sum_{i=1}^{K} M(i)$.*

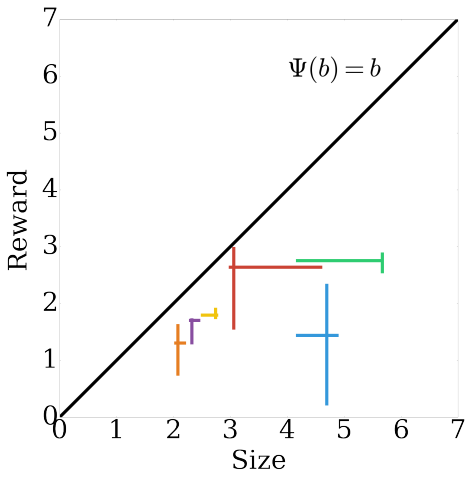


Figure 4.2: Item sizes and rewards. Each color is an item with horizontal lines between the two sizes and vertical lines between minimum and maximum reward. The lines cross at the point (mean size, mean reward).

4.7 Experimental Results

We demonstrate the performance of `OpStoK` on a simple experimental setup with 6 items. Each item i can take two sizes and is larger with probability x_i. The rewards come from scaled and shifted Beta distributions. The budget is 7 meaning that a maximum of 3 items can be placed in the knapsack. We take $\Psi(b) = b$ and set the parameters of the algorithm to $\delta_{0,1} = \delta_{0,2} = 0.1$ and $\epsilon = 0.5$. Figure 4.2 illustrates the problem.

We compare the performance of `OpStoK` in this setting to the algorithm in (Dean et al., 2008) run with various values of κ, the parameter used to define the small items threshold. We chose κ to ensure that we consider all cases from 0 small items to 6 small items. Note that the algorithm in (Dean et al., 2008) is designed for deterministic rewards so we sampled the rewards for each item at the start to get estimates of the true rewards. When sampling item sizes for (Dean et al., 2008), we used the `OpStoK` sampling strategy. For both algorithms, when evaluating the value of a policy, we re-sampled the value of the chosen policies as discussed in Section 4.2.1. The results of this experiment are shown in Figure 4.3. From this, the anytime

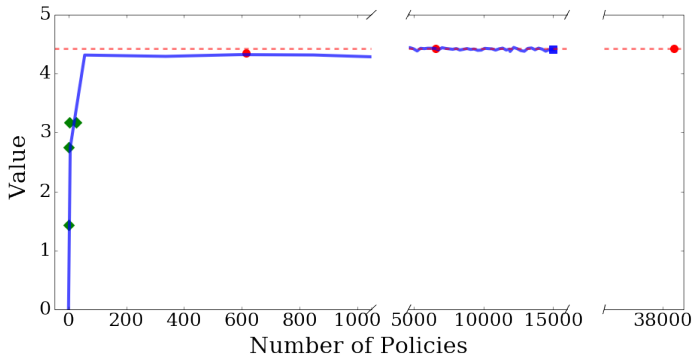


Figure 4.3: Number of policies vs value. The blue line is the estimated value of the best policy so far found by `OpStoK` which terminates at the square. The green diamonds are the best value for (Dean et al., 2008) when small items are chosen, and red circles when it chooses large items. The estimated value of the best solution from (Dean et al., 2008) is given by the red dashed line.

property of our algorithm can be seen; it is able to find a good policy early on (after less than 100 policies) so if it was stopped early, it would still return a policy with a high expected value. Furthermore, at termination, the algorithm has almost reached the best solution from Dean et al. (2008) which required more than twice as many policies to be evaluated. Thus this experiment has shown that our algorithm not only returns a policy with near optimal value, but it does this after evaluating significantly fewer policies and, even if stopped prematurely, it will return a good policy.

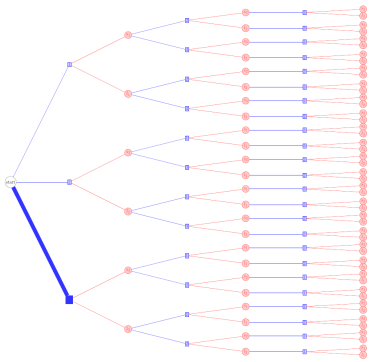
These experimental results were obtained using the `OpStoK` algorithm as stated in Algorithm 4.1. This algorithm incorporates the sharing of samples between policies and preferential sampling of complete policies to improve performance. For large problems, the computational performance of `OpStoK` can be further improved by parallelization. In particular, the expansion of a policy can be done in parallel with each leaf of the policy being expanded on a different core and then recombined. It is also possible to sample the value and remaining budget of a policy in parallel.

4.8 Conclusion

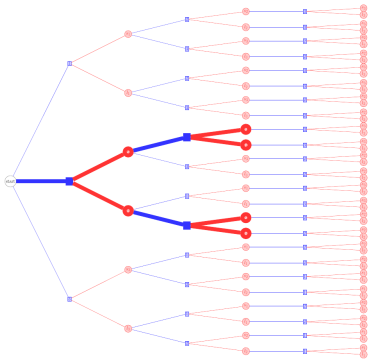
In this chapter we have presented `OpStoK`, an anytime optimistic planning algorithm specifically tailored to the stochastic knapsack problem. For this algorithm, we have provided confidence intervals, consistency results, bounds on the sample size and shown that it needn't evaluate all policies to find an ϵ-optimal solution; making it the first such algorithm for the stochastic knapsack problem. By using estimates of the remaining budget and value, `OpStoK` is adaptive and also benefits from a unique streamlined sampling scheme. While `OpStoK` was developed for the stochastic knapsack problem, it is hoped that it is just the first step towards using optimistic planning to tackle many frequently occurring resource allocation problems.

4.A Supplementary Material

4.A.1 Illustration of Policies



(a) A policy of just playing item 3. This policy has depth 1.



(b) A policy that plays item 2 first. If it is small, it plays item 1 whereas if it is large it plays item 3. After this, the final item is determined due to the fact that there are only 3 items in the problem. This policy has depth 2.

Figure 4.4: Examples of policies in the simple 3 item, 2 sizes stochastic knapsack problem. Each blue line represents choosing an item and the red lines represent the sizes of the previous items.

4.A.2 Illustration of Bounds

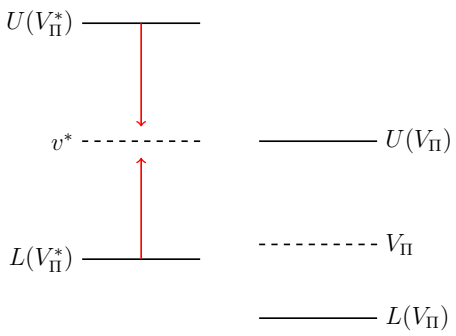


Figure 4.5: Example of where just looking at the optimistic policy might fail: If we always play the optimistic policy then, since $U(V^+_{\Pi^*}) \geq U(V^+_\Pi)$, we will always play Π^* and so the confidence bounds on Π will not shrink. This means that $L(V^+_{\Pi^*})$ will never be (epsilon) greater than the best alternative upper bound so it will not be possible to conclude we have found the best policy with high confidence.

4.A.3 Algorithms

In these algorithms $\texttt{Generate}(i)$ samples a reward and item size pair from the generative model of item i, whereas $\texttt{sample}(A, k)$ samples from a set A with replacement to get k samples. The notation $i(d) = \Pi(d, b)$ indicates that item $i(d)$ was chosen by policy Π at depth d when the remaining capacity was b.

Algorithm 4.3: $\texttt{SampleBudget}(\Pi, \mathcal{S})$

Initialization: $B_0 = B$ and for all $i \in I$, $\mathcal{S}_i = \mathcal{S}^*_i$

for $d = 1, \ldots, d(\Pi)$ **do**
  $i(d) = \Pi(d, B_{d-1})$;
  if $|\mathcal{S}_{i(d)}| \leq 0$ **then** $(r_{i(d)}, c_{i(d)}) = \texttt{Generate}(i(d))$, $\mathcal{S}^*_i = \mathcal{S}^*_i \cup \{r_{i(d)}, c_{i(d)})\}$;
    else $(r_{i(d)}, c_{i(d)}) = \texttt{sample}(\mathcal{S}_i, 1)$, and $\mathcal{S}_i = \mathcal{S}_i \setminus \{(r_{i(d)}, c_{i(d)})\}$;
  $B_d = B_{d-1} - c_{i(d)}$;
end
$\overline{\Psi(B_\Pi)}^{(j)} = \Psi(\max\{B - \sum_{d=1}^{d(\Pi)} c_{i(d)}, 0\})$;
return $\left(\overline{\Psi(B_\Pi)}^{(j)}, \mathcal{S}^*\right)$

Algorithm 4.4: `EstimateValue`(Π, m)

Initialization: For all $i \in I$, $\mathcal{S}_i = \mathcal{S}_i^*$
for $j = 1, \dots, m$ **do**
  $B_0 = B$;
  for $d = 1, \dots, d(\Pi)$ **do**
    $i(d) = \Pi(d, B_{d-1})$;
    if $|\mathcal{S}_{i(d)}| \leq 0$ **then** $(r_{i(d)}, c_{i(d)}) =$ `Generate`$(i(d))$, $\mathcal{S}_i^* = \mathcal{S}_i^* \cup \{r_{i(d)}, c_{i(d)})\}$;
      else $(r_{i(d)}, c_{i(d)}) =$ `sample`$(\mathcal{S}_i, 1)$, and $\mathcal{S}_i = \mathcal{S}_i \setminus \{(r_{i(d)}, c_{i(d)})\}$;
    $B_d = B_{d-1} - c_{i(d)}$;
    if $B_d < 0$ **then** $r_{i(d)} = 0$;
  end
  $\overline{V_\Pi}^{(j)} = \sum_{d=1}^{d(\Pi)} r_{i(d)}$;
end
return $(\overline{V_\Pi}_m = \frac{1}{m} \sum_{j=1}^{m} \overline{V_\Pi}^{(j)}, \mathcal{S}^*)$

4.B Proofs of Theoretical Results

For convenience we restate any results before proving them.

4.B.1 Bounding the Value of a Policy

Lemma 4.7. (Lemma 4.1 in main text) *Let $(\Omega, \mathcal{A}, P)$ be the probability space from Section 4.2, then for $m_1 + m_2$ independent samples of policy Π, and $\delta_1, \delta_2 > 0$, with probability $1 - \delta_1 - \delta_2$,*

$$\overline{V_\Pi}_{m_1} - c_1 \leq V_\Pi^+ \leq \overline{V_\Pi}_{m_1} + \overline{\Psi(B_\Pi)}_{m_2} + c_1 + c_2.$$

Where $c_1 := \sqrt{\frac{\Psi(B)^2 \log(2/\delta_1)}{2m_1}}$ and $c_2 := \sqrt{\frac{\Psi(B)^2 \log(1/\delta_2)}{2m_2}}$.

Proof. Consider the average value of policy Π over m_1 many trials. By Hoeffding's Inequality, $P\left(|\overline{V_\Pi}_{m_1} - V_\Pi| > c_1\right) \leq \delta_1$ and, $P\left(|\overline{\Psi(B_\Pi)}_{m_2} - E[\Psi(B_\Pi)]| > c_2\right) \leq \delta_2$.

We are interested in the probability,

$$\begin{aligned} P(|\overline{V_\Pi}_{m_1} - V_\Pi^+| > t) &\leq P(|\overline{V_\Pi}_{m_1} - V_\Pi| + |V_\Pi - V_\Pi^+| > t) \\ &\leq P(|\overline{V_\Pi}_{m_1} - V_\Pi| + E[\Psi(B_\Pi)] > t). \end{aligned}$$

where the first line follows from the triangle inequality and the second from the definition of $\Psi(B_\Pi)$. From the Hoeffding bounds and defining $t = \overline{\Psi(B_\Pi)}_{m_2} + c_1 + c_2$, we consider $P\left(|\overline{V_\Pi}_{m_1} - V_\Pi| + E[\Psi(B_\Pi)] > \overline{\Psi(B_\Pi)}_{m_2} + c_1 + c_2\right)$. Define the events

$$A_1 = \{|\overline{V_\Pi}_{m_1} - V_\Pi| + E[\Psi(B_\Pi)] \leq E[\Psi(B_\Pi)] + c_1\}, A_2 = \left\{|\overline{\Psi(B_\Pi)}_{m_2} - E[\Psi(B_\Pi)]| \leq c_2\right\}.$$

Then,

$$\begin{aligned} P\left(|\overline{V_\Pi}_{m_1} - V_\Pi| + E[\Psi(B_\Pi)] > \overline{\Psi(B_\Pi)}_{m_2} + c_1 + c_2\right) &\leq P(\Omega \backslash (A_1 \cap A_2)) \\ &\leq P(\Omega \backslash A_1) + P(\Omega \backslash A_2) \\ &\leq \delta_1 + \delta_2. \end{aligned}$$

Hence,

$$P\left(\overline{V_\Pi}_{m_1} - V_\Pi^+ > c_1\right) \leq P\left(\overline{V_\Pi}_{m_1} - V_\Pi > c_1\right) \leq \delta_1 < \delta_1 + \delta_2$$

which gives the left hand side of the result. For the right hand side,

$$\begin{aligned} P\left(\overline{V_\Pi}_{m_1} - V_\Pi^+ < -\overline{\Psi(B_\Pi)}_{m_2} - c_1 - c_2\right) & \\ &\leq P\left(\overline{V_\Pi}_{m_1} - V_\Pi - E[\Psi(B_\Pi)] < -\overline{\Psi(B_\Pi)}_{m_2} - c_1 - c_2\right) \\ &\leq \delta_1 + \delta_2. \end{aligned}$$

□

Lemma 4.8. *Let $\{Z_m\}_{m=1}^\infty$ be a martingale with Z_m defined on the filtration $\mathcal{F}_m$,*

$E[Z_m] = 0$ and $|Z_m - Z_{m-1}| \leq d$ for all m where $Z_0 = 0$. Then,

$$P\left(\exists m \leq n; \frac{Z_m}{m} \geq 2d^2\sqrt{\frac{2}{m}\log\left(\frac{n}{m}\frac{4}{\delta}\right)}\right) \leq \delta$$

Proof. The proof is similar to that of Lemma B.1 in (Perchet et al., 2016) and will make use of the following standard results:

Theorem 4.9. Doob's maximal inequality: *Let Z be a non-negative submartingale. Then for $c > 0$,*

$$P\left(\sup_{k\leq n} Z_k \geq c\right) \leq \frac{E[Z_n]}{c}.$$

Proof. See, for example, (Williams, 1991), Theorem 14.6, page 137. □

Lemma 4.10. *Let Z_n be a martingale such that $|Z_i - Z_{i-1}| \leq d_i$ for all i with probability 1. Then, for $\lambda > 0$,*

$$E[e^{\lambda Z_n}] \leq e^{\frac{\lambda^2 D^2}{2}},$$

where $D^2 = \sum_{i=1}^n d_i^2$.

Proof. See the proof of the Azuma-Hoeffding inequality in (Azuma, 1967). □

Then, for the proof of Lemma 4.8, we first notice that since $\{Z_m\}_{m=1}^{\infty}$ is a martingale, by Jensen's inequality for conditional expectations, it follows that for any $\lambda > 0$,

$$E[e^{\lambda Z_m}|\mathcal{F}_{m-1}] \geq e^{\lambda E[Z_m|\mathcal{F}_{m-1}]} = e^{\lambda Z_{m-1}}.$$

Hence, for any $\lambda > 0$, $\{e^{\lambda Z_m}\}_{m=1}^{\infty}$ is a positive sub-martingale so we can apply Doob's maximal inequality (Theorem 4.9) to get

$$P\left(\sup_{m\leq n} Z_m \geq c\right) = P\left(\sup_{m\leq n} e^{\lambda Z_m} \geq e^{\lambda c}\right) \leq \frac{E[e^{\lambda Z_n}]}{e^{\lambda c}}.$$

Then, by Lemma 4.10, since $|Z_i - Z_{i-1}| \leq d$ for all i, it follows that

$$P\left(\sup_{m\leq n} Z_m \geq c\right) \leq \frac{E[e^{\lambda Z_n}]}{e^{\lambda c}} \leq \frac{e^{\lambda^2 D^2/2}}{e^{\lambda c}} = \exp\left\{\frac{\lambda^2 D^2}{2} - \lambda c\right\}. \tag{4.5}$$

Minimizing the right hand side with respect to λ gives $\hat{\lambda} = \frac{c}{D^2}$ and substituting this back into (4.5) gives,

$$P\left(\sup_{m\leq n} Z_m \geq c\right) \leq \exp\left\{-\frac{c^2}{2D^2}\right\}.$$

Then, since we are considering the case where $d_i = d$ for all i, $D^2 = nd^2$ and so,

$$P\left(\sup_{m\leq n} Z_m \geq c\right) \leq \exp\left\{-\frac{c^2}{2nd^2}\right\}.$$

Further, if we are interested in $P(\sup_{k\leq m\leq n} Z_m \geq c)$, we can redefine the indices to get

$$P\left(\sup_{k\leq m\leq n} Z_m \geq c\right) = P\left(\sup_{m'\leq n-k+1} Z_m \geq c\right) \leq \exp\left\{-\frac{c^2}{2(n-k+1)d^2}\right\}. \tag{4.6}$$

We then define $\varepsilon_m = 2d\sqrt{\frac{1}{m}\log\left(\frac{n}{m}\frac{8}{\delta}\right)}$ and use a peeling argument similar to that in

Lemma B.1 of (Perchet et al., 2016) to get

$$
\begin{aligned}
&P\left(\exists m \leq n; \frac{Z_m}{m} \geq \varepsilon_m\right) \\
&\quad \leq \sum_{t=0}^{\lfloor \log_2(n) \rfloor + 1} P\left(\bigcup_{m=2^t}^{2^{t+1}-1} \left\{\frac{Z_m}{m} \geq \varepsilon_m\right\}\right) && \text{(by union bound)} \\
&\quad \leq \sum_{t=0}^{\lfloor \log_2(n) \rfloor + 1} P\left(\bigcup_{m=2^t}^{2^{t+1}-1} \left\{\frac{Z_m}{m} \geq \varepsilon_{2^{t+1}}\right\}\right) && \text{(since } \varepsilon_m \text{ decreasing in } m) \\
&\quad \leq \sum_{t=0}^{\lfloor \log_2(n) \rfloor + 1} P\left(\bigcup_{m=2^t}^{2^{t+1}-1} \{Z_m \geq 2^t \varepsilon_{2^{t+1}}\}\right) && \text{(as } m \geq 2^t) \\
&\quad \leq \sum_{t=0}^{\lfloor \log_2(n) \rfloor + 1} \exp\left\{-\frac{(2^t \varepsilon_{2^{t+1}})^2}{2^{t+1} d^2}\right\} && \text{(from (4.6))} \\
&\quad \leq \sum_{t=0}^{\lfloor \log_2(n) \rfloor + 1} \frac{2^{t+1}\delta}{8n} && \text{(substituting } \varepsilon_{2^{t+1}}) \\
&\quad \leq \frac{2^{\log_2(n)+3}\delta}{8n} = \delta. && \text{(since } \sum_{i=1}^{k} 2^i = 2^{k+1} - 1)
\end{aligned}
$$

□

Proposition 4.11. (Proposition 4.2 in main text) *The Algorithm* `BoundValueShare` *(Algorithm 4.2) returns confidence bounds,*

$$
\begin{aligned}
L(V_\Pi^+) &= \overline{V_\Pi}_{m_1} - \sqrt{\frac{\Psi(B)^2 \log(2/\delta_1)}{2m_1}} \\
U(V_\Pi^+) &= \overline{V_\Pi}_{m_1} + \overline{\Psi(B_\Pi)}_{m_2} + \sqrt{\frac{\Psi(B)^2 \log(2/\delta_1)}{2m_1}} + 2\Psi(B)\sqrt{\frac{1}{m_2}\log\left(\frac{8n}{\delta_2 m_2}\right)}
\end{aligned}
$$

which hold with probability $1 - \delta_1 - \delta_2$.

Proof. We begin by noting that our samples of item size are dependent since in each iteration we construct a bound based on past samples and we use this bound to decide if we need to continue sampling or if we can stop. To model this dependence let us introduce a stopping time τ such that $\tau(\omega) = n$ if our algorithm exits the loop at

time n. Consider the sequence

$$\overline{\Psi(B_\Pi)}_{1\wedge\tau}, \overline{\Psi(B_\Pi)}_{2\wedge\tau}, \dots$$

and define for $m \geq 1$

$$M_m = (m \wedge \tau)(\overline{\Psi(B_\Pi)}_{m\wedge\tau} - E[\Psi(B_\Pi)]) \quad \text{with} \quad M_0 = 0.$$

Furthermore, define the filtration $\mathcal{F}_m = \sigma(B_{\Pi,1}, \dots, B_{\Pi,m})$ then for $m \geq 1$

$$E[M_m|\mathcal{F}_{m-1}] = E[M_m|\mathcal{F}_{m-1}, \tau \leq m-1] + E[M_m|\mathcal{F}_{m-1}, \tau > m-1].$$

Now

$$E[M_m|\mathcal{F}_{m-1}, \tau \leq m-1] = E[M_{m-1}|\tau \leq m-1].$$

and due to independence of the samples $B_{\Pi,1}, \dots, B_{\Pi,m}$

$$\begin{aligned}
&E[M_m|\mathcal{F}_{m-1}, \tau > m-1] \\
&\quad = E[m(\overline{\Psi(B_\Pi)}_m - E[\Psi(B_\Pi)])|\mathcal{F}_{m-1}, \tau > m-1] \\
&\quad = E\left[\sum_{j=1}^{m-1} \Psi(B_{\Pi,j}) + \Psi(B_{\Pi,m}) - mE[\Psi(B_\Pi)]\middle|\mathcal{F}_{m-1}, \tau > m-1\right] \\
&\quad = (m-1)E[\overline{\Psi(B_\Pi)}_{m-1} - E[\Psi(B_\Pi)]|\mathcal{F}_{m-1}, \tau > m-1] \\
&\qquad\qquad + E[\Psi(B_{\Pi,m}) - E[\Psi(B_\Pi)]|\mathcal{F}_{m-1}, \tau > m-1] \\
&\quad = E[M_{m-1}|\tau > m-1] + E[\Psi(B_{\Pi,m})] - E[\Psi(B_\Pi)] = E[M_{m-1}|\tau > m-1].
\end{aligned}$$

Hence, $E[M_m|\mathcal{F}_{m-1}] = M_{m-1}$ and M_m is a martingale with increments $|M_m - M_{m-1}| \leq |\Psi(B_{\Pi,m}) - E[\Psi(B_\Pi)]| \leq \Psi(B)$. We could apply the Azuma-Hoeffding inequality to

gain guarantees for individual m-values. Alternatively, we can use Lemma 4.8 to get,

$$P\left(\sup_{m\leq n}\frac{M_m}{m}\geq 2\Psi(B)\sqrt{\frac{1}{m}\log\left(\frac{8n}{\delta m}\right)}\right)\leq\delta_2.$$

Combining this with the argument in Lemma 4.1 gives

$$\overline{V_\Pi}_{m_1}-c_1\leq V_\Pi^+\leq\overline{V_\Pi}_{m_1}+\overline{\Psi(B_\Pi)}_{m_2}+c_1+c_2,$$

where $c_1:=\sqrt{\frac{\Psi(B)^2\log(2/\delta_1)}{2m_1}}$ and $c_2:=2\Psi(B)\sqrt{\frac{1}{m_2}\log\left(\frac{8n}{\delta_2 m_2}\right)}$ and these bounds hold with probability $1-\delta_1-\delta_2$. □

Lemma 4.12. *With probability* $1-\delta_{0,1}-\delta_{0,2}$*, the bounds generated by* `BoundValueShare` *with parameters* $\delta_{1,d}=\frac{\delta_{0,1}}{d^*}N_d^{-1}$ *and* $\delta_{2,d}=\frac{\delta_{0,2}}{d^*}N_d^{-1}$ *hold for all policies* Π *of depth* $d=d(\Pi)\leq d^*$ *simultaneously.*

Proof. The probability that all bounds hold simultaneously is $P(\bigcap_{\Pi\in\mathcal{P}}\{L(V_\Pi^+)\leq V_\Pi\leq U(V_\Pi^+)\})$ where $\mathcal{P}$ is the set of all policies. From Proposition 4.2, for any policy Π of depth $d=d(\Pi)$, $P(L(V_\Pi^+)\leq V_\Pi\leq U(V_\Pi^+))\geq 1-\delta_{d,1}-\delta_{d,2}$. Then,

$$\begin{aligned}
P\left(\bigcap_{\Pi\in\mathcal{P}}\{L(V_\Pi^+)\leq V_\Pi\leq U(V_\Pi^+)\}\right)&=1-P\left(\bigcup_{\Pi\in\mathcal{P}}\{L(V_\Pi^+)\leq V_\Pi\leq U(V_\Pi^+)\}^c\right)\\
&\geq 1-\sum_{\Pi\in\mathcal{P}}P(\{L(V_\Pi^+)\leq V_\Pi\leq U(V_\Pi^+)\}^c)\\
&\geq 1-\sum_{\Pi\in\mathcal{P}}(\delta_{d(\Pi),1}+\delta_{d(\Pi),2})\\
&=1-\sum_{d=1}^{d^*}N_d(\delta_{d,1}+\delta_{d,2})\\
&\geq 1-\sum_{d=1}^{d^*}N_d\left(\frac{\delta_{0,1}}{d^*}N_d^{-1}+\frac{\delta_{0,2}}{d^*}N_{d(\Pi_t)}^{-1}\right)\\
&=1-\sum_{d=1}^{d^*}\frac{1}{d^*}(\delta_{0,1}+\delta_{0,2})=1-\delta_{0,1}-\delta_{0,2}
\end{aligned}$$

□

4.B.2 Theoretical Results for Optimistic Stochastic Knapsacks (OpStoK)

Proposition 4.13. (Proposition 4.4 in main text) *With probability at least* $(1-\delta_{0,1}-\delta_{0,2})$*, the algorithm* `OpStoK` *returns a policy with value at least* $v^* - \epsilon$*.*

Proof. The proof follows from the following lemma.

Lemma 4.14. *For every round of the algorithm and incomplete policy* Π*, let* $D(\Pi)$ *be the set of all descendants of* Π*. Define the event* $A = \bigcap_{\Pi' \in D(\Pi)} \{V_{\Pi'} \in [L(V_\Pi^+), U(V_\Pi^+)]\}$*. Then* $P(A) \geq 1 - \delta_{0,1} - \delta_{0,2}$*.*

Proof. When `BoundValueShare` is called for a policy Π with $d(\Pi) = d$, it is done so with parameters $\delta_{d,1} = \frac{\delta_{0,1}}{d^*} N_d^{-1}$ and $\delta_{d,2} = \frac{\delta_{0,2}}{d^*} N_d^{-1}$, where $\delta_{d,1}$ and $\delta_{d,2}$ are used to control the accuracy of the estimated value of V_Π and $E\Psi(B_\Pi)$ respectively. It follows from Proposition 4.2, that for any active policy Π, the probability that the interval $\left[\overline{V_\Pi}_{m_1} - c_1, \overline{V_\Pi}_{m_1} + \overline{\Psi(B_\Pi)}_{m_2} + c_1 + c_2\right]$ generated by `BoundValueShare` does not contain V_Π^+ is less than $\delta_{d,1} + \delta_{d,2}$. Furthermore, from standard Hoeffding bounds, the probability that V_Π is outside the interval $[V_\Pi - c_1, V_\Pi + c_1]$ is less than $\delta_{d,1}$. Since any descendant policy Π' of Π consists of adding at least one item to the knapsack and item rewards are all ≥ 0, it follows that $V_\Pi \leq V_{\Pi'} \leq V_\Pi^+$. Hence, the probability of the value of a descendant policy being outside the interval $[L(V_\Pi^+), U(V_\Pi^+)]$ is less than $\delta_{d,1} + \delta_{d,2}$. By the same argument as in Lemma 4.12, it can be shown that $P(A) > 1 - \sum_{d=1}^{d^*} (\delta_{d,1} + \delta_{d,2}) N_d = 1 - \delta_{0,1} - \delta_{0,2}$. □

The result of the proposition follows by noting that the true optimal policy Π^{OPT} will be a descendant of Π_i for some $i \in I$. Let Π^* be the policy outputted by the algorithm. By the stopping criterion, $L(V_{\Pi^*}^+) + \epsilon \geq \max_{\Pi \in \text{ACTIVE} \setminus \{\Pi^*\}} \geq U(V_\Pi^+)$ for any $\Pi \in$ ACTIVE. From the expansion rule of `OpStoK`, it follows that either $\Pi^{OPT} \in$ ACTIVE or there exists some ancestor policy Π' of Π^{OPT} in ACTIVE. In the

first case, $V_{\Pi^{OPT}} = v^* \leq U(V^+_{\Pi^{OPT}})$ whereas in the latter $V_{\Pi^{OPT}} = v^* \leq U(V^+_{\Pi'})$ with high probability from Lemma 4.14. In either case, it follows that $L(V^+_{\Pi^*}) + \epsilon \geq v^*$ and so $V_{\Pi^*} + \epsilon \geq v^*$. □

Lemma 4.15. *When the confidence bounds hold, if* Π *is a complete policy then,* $U(V^+_\Pi) - L(V^+_\Pi) \leq \epsilon$*, otherwise* $U(V^+_\Pi) - L(V^+_\Pi) \leq 9E\Psi(B_\Pi) - \frac{3}{4}\epsilon$.

Proof. By the bounds in Proposition 4.2, $U(V^+_\Pi) - L(V^+_\Pi) \leq \overline{\Psi(B_\Pi)}_{m_2} + c_2 + 2c_1 = U(\Psi(B_\Pi)) + 2c_1$. For a complete policy, $U(\Psi(B_\Pi)) \leq \frac{\epsilon}{2}$ and according to `BoundValueShare`, m_1 is chosen such that $2c_1 \leq \frac{\epsilon}{2}$ which implies $U(V^+_\Pi) - L(V^+_\Pi) \leq \epsilon$.
If Π is not complete, by the sampling strategy in `BoundValueShare`, we continue sampling the remaining budget until $L(\Psi(B_\Pi)) \geq \frac{\epsilon}{4}$. In this setting, the maximal width of the confidence interval of $E\Psi(B_\Pi)$ will satisfy

$$c_2 \leq E\Psi(B_\Pi) - \frac{\epsilon}{4}. \tag{4.7}$$

since, if $c_2 > E\Psi(B_\Pi) - \frac{\epsilon}{4}$, then for $E\Psi(B_\Pi) > \overline{\Psi(B_\Pi)}_{m_2}$, $\frac{\epsilon}{4} > E\Psi(B_\Pi) - c_2 > \overline{\Psi(B_\Pi)}_{m_2} - 2c_2 = L(\Psi(B_\Pi))$, and for $E\Psi(B_\Pi) \leq \overline{\Psi(B_\Pi)}_{m_2}$, $\overline{\Psi(B_\Pi)}_{m_2} - E\Psi(B_\Pi) \leq c_2$ so $\frac{\epsilon}{4} > E\Psi(B_\Pi) - c_2 > \overline{\Psi(B_\Pi)}_{m_2} - 2c_2 = L(\Psi(B_\Pi))$. In both cases this contradicts the assumption that $L(\Psi(B_\Pi)) > \frac{\epsilon}{4}$ by definition of the algorithm. Hence,

$$\begin{aligned}
U(V^+_\Pi) - L(V^+_\Pi) &\leq U(\Psi(B_\Pi)) + 2c_1 \\
&\leq 3U(\Psi(B_\Pi)) && (4.8)\\
&\leq 3(E\Psi(B_\Pi) + 2c_2) \\
&\leq 3\left(E\Psi(B_\Pi) + 2E\Psi(B_\Pi) - 2\frac{\epsilon}{4}\right) && (4.9)\\
&\leq 9E\Psi(B_\Pi) - \frac{3}{4}\epsilon.
\end{aligned}$$

Where (4.8) follows since, when $L(\Psi(B_\Pi)) \geq \frac{\epsilon}{4}$, we sample the value of policy Π until $c_1 \leq U(\Psi(B_\Pi))$, and (4.9) by substituting in (4.7). □

Lemma 4.16. (Lemma 4.3 in main text) *Assume that* $L(V_\Pi^+) \le V_\Pi \le U(V_\Pi^+)$ *holds simultaneously for all policies* $\Pi \in$ ACTIVE *with* $U(V_\Pi^+)$ *and* $L(V_\Pi^+)$ *as defined in Proposition 4.2. Then,* $\Pi_t \in \mathcal{Q}^\epsilon$ *for every policy selected by* `OpStoK` *at every time point* t*, except for possibly the last one.*

Proof. Since, when we expand a policy, we replace it in ACTIVE by all its child policies, at any time point $t \ge 1$ there will be one ancestor of Π^* in the active set, denote this policy by Π_t^*. If $\Pi_t = \Pi_t^*$, then by Lemma 4.14, $V_{\Pi^*} \in [L(V_{\Pi_t}^+), U(V_{\Pi_t}^+)]$. Hence,

$$V_\Pi + 9E\Psi(B_\Pi) - \frac{3}{4}\epsilon \ge U(V_\Pi^+) \ge v^* \ge v^* - 9E\Psi(B_\Pi) + \frac{3}{4}\epsilon + \epsilon.$$

Where the last inequality will hold for any incomplete policy (since for an incomplete policy $L(\Psi(B_\Pi)) \ge \frac{\epsilon}{4}$) and so, $\Pi_t \in \mathcal{Q}^\epsilon$. For $\Pi_t = \Pi^*$, $V_\Pi + \epsilon \ge v^*$ so $\Pi_t \in \mathcal{Q}^\epsilon$.

Assume $\Pi_t \ne \Pi_t^*$. If Π_t is a complete policy, $U(V_{\Pi_t}^+) - L(V_{\Pi_t}^+) \le \epsilon$. For a complete policy Π to be selected, it must have the largest $U(V_\Pi^+)$, since most alternative policies will have larger $U(\Psi(B_\Pi))$. Hence $\Pi_t^{(1)} = \Pi_t$ and

$$L(V_{\Pi_t^{(1)}}^+) + \epsilon \ge U(V_{\Pi_t^{(1)}}^+) \ge \max_{\Pi \in \text{ACTIVE} \setminus \{\Pi_t^{(1)}\}} U(V_\Pi^+),$$

so the algorithm stops.

Assume $\Pi_t = \Pi_t^{(1)} \ne \Pi_t^*$ is an incomplete policy. By Lemma 4.15, for an incomplete policy,

$$U(V_\Pi^+) - L(V_\Pi^+) \le 3U(\Psi(B_\Pi)) \le 9E\Psi(B_\Pi) - \frac{3}{4}\epsilon. \tag{4.10}$$

Then, if the termination criteria is not met,

$$\begin{aligned} V_{\Pi_t} \geq L(V^+_{\Pi_t}) \quad \Longrightarrow \quad V_{\Pi_t} + 9E\Psi(B_\Pi) - \frac{3}{4}\epsilon - \epsilon &\geq L(V^+_{\Pi_t}) + 9E\Psi(B_\Pi) - \frac{3}{4}\epsilon - \epsilon \\ &\geq U(V^+_{\Pi_t}) - \epsilon \\ &\geq \max_{\Pi \in \text{ACTIVE} \setminus \{\Pi_t\}} U(V^+_\Pi) - \epsilon \\ &\geq L(V^+_{\Pi_t}) \\ &\geq U(V^+_{\Pi_t}) - 9E\Psi(B_\Pi) + \frac{3}{4}\epsilon \\ &\geq U(V^+_{\Pi^*_t}) - 9E\Psi(B_\Pi) + \frac{3}{4}\epsilon \\ &\geq v^* - 9E\Psi(B_\Pi) + \frac{3}{4}\epsilon \end{aligned}$$

which follows since $\Pi^{(1)}_t$ is the policy with largest upper bound. Therefore, $\Pi_t \in \mathcal{Q}^\epsilon$.

By the stopping criteria of `OpStoK`, if the algorithm does not stop and select $\Pi^{(1)}_t$ as the optimal policy, then $\Pi_t = \Pi^{(2)}_t$ and

$$L(V^+_{\Pi^{(1)}_t}) + \epsilon < \max_{\Pi \in \text{ACTIVE} \setminus \{\Pi^{(1)}_t\}} U(V^+_\Pi) = U(V^+_{\Pi^{(2)}_t}).$$

By equation (4.10),

$$L(V^+_{\Pi^{(1)}_t}) + 9E\Psi(B_\Pi) - \frac{3}{4}\epsilon \geq U(V^+_{\Pi^{(1)}_t}).$$

and by the selection criterion $U(\Psi(B_{\Pi_t^{(2)}})) \geq U(\Psi(B_{\Pi_t^{(1)}}))$. Hence, for $\Pi_t = \Pi_t^{(2)} \neq \Pi_t^*$,

$$\begin{aligned}
V_{\Pi_t} + 18E\Psi(B_\Pi) - \frac{6}{4}\epsilon - \epsilon &\geq L(V_{\Pi_t}^+) + 9E\Psi(B_{\Pi_t}) - \frac{3}{4}\epsilon + 9E\Psi(B_{\Pi_t}) - \frac{3}{4}\epsilon - \epsilon \\
&\geq U(V_{\Pi_t}^+) + 9E\Psi(B_{\Pi_t}) - \frac{3}{4}\epsilon - \epsilon \qquad \text{(by (4.7))} \\
&\geq U(V_{\Pi_t}^+) + 3U(\Psi(B_{\Pi_t})) - \epsilon \\
&\geq U(V_{\Pi_t}^+) + 3U(\Psi(B_{\Pi_t^{(1)}})) - \epsilon \\
&\geq L(V_{\Pi_t^{(1)}}^+) + 3U(\Psi(B_{\Pi_t^{(1)}})) \\
&\geq U(V_{\Pi_t^{(1)}}^+) \\
&\geq U(V_{\Pi_t^*}^+) \\
&\geq v^*.
\end{aligned}$$

Therefore, $\Pi_t \in \mathcal{Q}^\epsilon$. □

Theorem 4.17. (Theorem 4.5 in main text) *The total number of samples required by* `OpStoK` *is bounded from above by,*

$$\sum_{\Pi \in \mathcal{Q}^\epsilon} (m_1(\Pi) + m_2(\Pi))\, d(\Pi),$$

with probability $1 - \delta_{0,2}$.

Proof. The result follows from the following three lemmas.

Lemma 4.18. *For* $\Pi \in \mathcal{A}^\epsilon$ *of depth* $d = d(\Pi)$*, then, with probability* $1 - \delta_{d,2}$*, the minimum number of samples of the value and remaining budget of the policy* Π *are bounded by*

$$m_1(\Pi) = \left\lceil \frac{8\Psi(B)^2 \log(\frac{2}{\delta_{d,1}})}{\epsilon^2} \right\rceil \quad \text{and} \quad m_2(\Pi) = m^*,$$

where m^* *is the smallest integer satisfying* $\frac{16\Psi(B)^2}{(E\Psi(B_\Pi) - \epsilon/2)^2} \leq \frac{m}{\log(8n/m\delta_2)}$ *with* n *defined as in* (4.2).

Proof. When $E\Psi(B_\Pi) \leq \frac{\epsilon}{4}$, the event $\{U(\Psi(B_\Pi)) \leq \frac{\epsilon}{2}\}$ will eventually occur with enough samples of the remaining budget of the policy. With probability greater than $1-\delta_{d,2}$, this will happen when $2c_2 \leq \frac{\epsilon}{2} - E\Psi(B_\Pi)$, since by Proposition 4.2 we know $\overline{\Psi(B_\Pi)}_{m_2} \in [E\Psi(B_\Pi) - c_2, E\Psi(B_\Pi) + c_2]$ where c_2 is as defined in Proposition 4.2. From this, it follows that $U(\Psi(B_\Pi)) \in [E\Psi(B_\Pi), E\Psi(B_\Pi) + 2c_2]$. We want to make sure that $U(\Psi(B_\Pi)) \leq \frac{\epsilon}{2}$ will eventually happen so we need to construct a confidence interval such that c_2 satisfies $E\Psi(B_\Pi) + 2c_2 \leq \frac{\epsilon}{2}$. Therefore we select m_2 such that,

$$\begin{aligned} 2c_2 &\leq \frac{\epsilon}{2} - E\Psi(B_\Pi) \\ \implies 4\Psi(B)\sqrt{\frac{2\log(\frac{8n}{m_2\delta_{d,2}})}{m_2}} &\leq \frac{\epsilon}{2} - E\Psi(B_\Pi) \\ \implies \frac{16\Psi(B)^2}{(E\Psi(B_\Pi) - \epsilon/2)^2} &\leq \frac{m_2}{\log(4n/m_2\delta_2)}. \end{aligned}$$

Defining, $m_2(\Pi) = m^*$, where m^* is the smallest integer satisfying the above, is therefore an upper bound on the minimum number of samples necessary to ensure that $U(\Psi(B_\Pi)) \leq \frac{\epsilon}{2}$ with probability greater than $1-\delta_{d,2}$. When $U(\Psi(B_\Pi)) \leq \frac{\epsilon}{2}$, `BoundValueShare` requires $m_1(\Pi) = \left\lceil \frac{2\Psi(B)^2\log(\frac{2}{\delta_{d,1}})}{\epsilon^2} \right\rceil$ samples of the value of the policy to ensure $2c_1 \leq \frac{\epsilon}{2}$. □

Lemma 4.19. *For $\Pi \in \mathcal{B}^\epsilon$ of depth $d = d(\Pi)$, then, with probability $1-\delta_{d,2}$, the minimum number of samples of the value and remaining budget of the policy Π are bounded by*

$$m_1(\Pi) \leq \left\lceil \frac{\Psi(B)^2\log(\frac{2}{\delta_{d,1}})}{2E\Psi(B_\Pi)^2} \right\rceil \quad \text{and} \quad m_2(\Pi) = m^*,$$

where m^* is the smallest integer satisfying $\frac{16\Psi(B)^2}{(E\Psi(B_\Pi)-\epsilon/4)^2} \leq \frac{m}{\log(8n/m\delta_2)}$ with n defined as in (4.2).

Proof. When $E\Psi(B_\Pi) \geq \frac{\epsilon}{2}$, by noting that the event $\{L(\Psi(B_\Pi)) \geq \frac{\epsilon}{4}\}$ will eventually happen and using a very similar argument to Lemma 4.18, it follows that $m_2(\Pi)$ is

the smallest integer solution to

$$\frac{16\Psi(B)^2}{(E\Psi(B_\Pi) - \epsilon/4)^2} \leq \frac{m}{\log(8n/m\delta_2)},$$

with probability greater than $1 - \delta_{d,2}$. Whenever $L(\Psi(B_\Pi)) \geq \frac{\epsilon}{4}$, `BoundValueShare` requires $m_1(\Pi) = \left\lceil \frac{2\Psi(B)^2 \log(\frac{2}{\delta_{d,1}})}{(U(\Psi(B_\Pi))^2} \right\rceil$ samples of the value of policy Π. Since $U(\Psi(B_\Pi)) \in [E\Psi(B_\Pi), E\Psi(B_\Pi) + 2c_2]$ with probability $1 - \delta_{0,2}$, $U(\Psi(B_\Pi)) \geq E\Psi(B_\Pi)$, and so,

$$m_1(\Pi) = \left\lceil \frac{2\Psi(B)^2 \log(\frac{2}{\delta_{d,1}})}{(U(\Psi(B_\Pi))^2} \right\rceil \leq \left\lceil \frac{2\Psi(B)^2 \log(\frac{2}{\delta_{d,1}})}{E\Psi(B_\Pi)^2} \right\rceil$$

and the result holds. □

Lemma 4.20. *For $\Pi \in \mathcal{C}^\epsilon$ of depth $d = d(\Pi)$, then, with probability $1 - \delta_{d,2}$, the minimum number of samples of the value and remaining budget of the policy Π are bounded by*

$$m_1(\Pi) \leq \max\left\{ \left\lceil \frac{8\Psi(B)^2 \log(\frac{2}{\delta_{d,1}})}{\epsilon^2} \right\rceil, \left\lceil \frac{\Psi(B)^2 \log(\frac{2}{\delta_{d,1}})}{2E\Psi(B_\Pi)^2} \right\rceil \right\}$$

and $m_2(\Pi) = m^*$, where m^* is the smallest integer satisfying $\frac{16\Psi(B)^2}{(\epsilon/4)^2} \leq \frac{m}{\log(8n/m\delta_2)}$ with n defined as in (4.2).

Proof. When $\frac{\epsilon}{4} < E\Psi(B_\Pi) < \frac{\epsilon}{2}$, then the minimum width we will need a confidence interval to be is $\epsilon/4$. By an argument similar to Lemma 4.18, we can deduce that $m_2(\Pi)$ will be the smallest integer satisfying $\frac{16\Psi(B)^2}{(\epsilon/4)^2} \leq \frac{m}{\log(8n/m\delta_2)}$.

To determine the number of samples of the value required by `BoundValueShare`, we need to know which of $\{U(\Psi(B_\Pi)) \leq \frac{\epsilon}{2}\}$ or $\{L(\Psi(B_\Pi)) \geq \frac{\epsilon}{4}\}$ occurs first. However, when $\Pi \in \mathcal{C}^\epsilon$, we do not know this so the best we can do is bound $m_1(\Pi)$ by the

maximum of the two alternatives,

$$m_1(\Pi) \leq \max\left\{\left\lceil \frac{2\Psi(B)^2 \log(\frac{2}{\delta_{d,1}})}{\epsilon^2} \right\rceil, \left\lceil \frac{2\Psi(B)^2 \log(\frac{2}{\delta_{d,1}})}{E\Psi(B_\Pi)^2} \right\rceil\right\}.$$

□

The result of the theorem then follows by noting that for any policy Π of depth $d(\Pi)$, it will be necessary to have $m_1(\Pi)$ samples of the value of the policy and $m_2(\Pi)$ samples of the value of the policy. This requires $m_1(\Pi)d(\Pi)$ samples of item rewards, $m_1(\Pi)d(\Pi)$ samples of item sizes (to calculate the rewards) and $m_2(\Pi)d(\Pi)$ samples of item sizes (to calculate remaining budget), thus a total of $(m_1(\Pi) + m_2(\Pi))d(\Pi)$ calls to the generative model. From Lemma 4.3, any policy expanded by `OpStoK` will be in $\mathcal{Q}^\epsilon$ so it suffices to sum over all policies in $\mathcal{Q}^\epsilon$. This result assumes that all confidence bounds hold, whereas we know that for any policy Π of depth $d(\Pi)$, the probability of the confidence bound holding is greater than $1 - \delta_{d,2}$. By an argument similar to Lemma 4.12, the probability that all bounds hold is greater than $1 - \delta_{0,2}$. Note that, since $|\mathcal{Q}^\epsilon| \leq |\mathcal{P}|$, the probability should be considerably greater than $1 - \delta_{0,2}$. □

Chapter 5

Bandits with Delayed, Aggregated Anonymous Feedback

5.1 Introduction

The stochastic multi-armed bandit (MAB) problem is a prominent framework for capturing the exploration-exploitation tradeoff in online decision making and experiment design. An introduction to the MAB problem is given in Chapter 2. In the classic stochastic MAB setting, when the player pulls an arm, they immediately observes feedback in the form of a stochastic reward which can be used to improve the decisions in subsequent rounds. One of the main application areas of MABs is in online advertising. Here, the arms correspond to adverts, and the feedback would correspond to *conversions*, that is users buying a product after seeing an advert. However, in practice, these conversions may not necessarily happen immediately after the advert is shown, and it may not always be possible to assign the credit of a sale to a particular showing of an advert. A similar challenge is encountered in many other applications, e.g., in personalized treatment planning, where the effect of a treatment on a patient's health may be delayed, and it may be difficult to determine which out

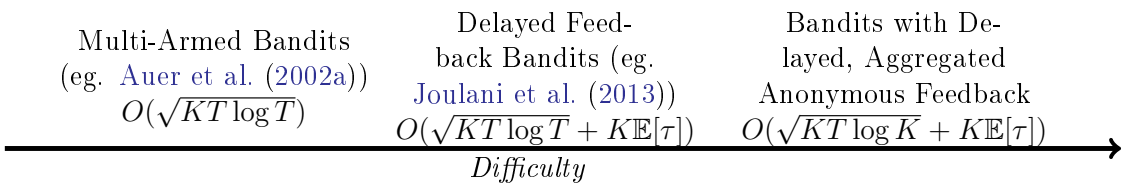


Figure 5.1: The relative difficulties and problem independent regret bounds of the different problems. For MABDAAF, our algorithm uses knowledge of $\mathbb{E}[\tau]$ and a mild assumption of a delay bound, which is not required by Joulani et al. (2013).

of several past treatments caused the change in the patient's health; or, in content design applications, where the effects of multiple changes in the website design on website traffic and footfall may be delayed and difficult to distinguish.

In this chapter, we propose a new bandit model to handle online problems with such 'delayed, aggregated and anonymous' feedback. In our model, a player interacts with an environment of K actions (or arms) in a sequential fashion. At each time step the player selects an action which leads to a reward generated at random from the underlying reward distribution. At the same time, a nonnegative random integer-valued delay is also generated i.i.d. from an underlying delay distribution. Denoting this delay by $\tau \geq 0$ and the index of the current round by t, the reward generated in round t will arrive at the end of the $(t+\tau)$th round. At the end of each round, the player observes only the *sum* of all the rewards that arrive in that round. Crucially, the player does not know which of the past plays have contributed to this aggregated reward. We call this problem *multi-armed bandits with delayed, aggregated anonymous feedback* (MABDAAF). As in the standard MAB problem, in MABDAAF, the goal is to maximize the cumulative reward from T plays of the bandit, or equivalently to minimize the regret.

If the delays are all zero, the MABDAAF problem reduces to the standard (stochastic) MAB problem, which has been studied considerably (see Chapter 2 for details). Compared to the MAB problem, the job of the player in our problem appears to be significantly more difficult since the player has to deal with *(i)* that some feedback

from the previous pulls may be *missing* due to the delays, and *(ii)* that the feedback takes the form of the sum of an *unknown number* of rewards of *unknown origin*.

An easier problem is when the observations are delayed, but they are *non-aggregated* and *non-anonymous*: that is, the player has to only deal with challenge (i) and not (ii). Here, the player receives delayed feedback in the shape of action-reward pairs that inform the player of both the individual reward and which action generated it. This problem, which we shall call the *(non-anonymous) delayed feedback bandit problem*, has been studied by Joulani et al. (2013), and later followed up by Mandel et al. (2015) for bounded delays. Remarkably, they show that compared to the standard (non-delayed) stochastic MAB setting, the regret will only increase additively by a factor that scales with the expected delay. For delay distributions with a finite expected delay, $\mathbb{E}[\tau]$, the worst case regret scales with $O(\sqrt{KT\log T} + K\mathbb{E}[\tau])$. Hence, the price to pay for the delay in receiving the observations is negligible. The QPM-D algorithm from (Joulani et al., 2013) and the SBD algorithm from (Mandel et al., 2015) place received rewards into queues for each arm, taking one whenever a base bandit algorithm suggests playing the arm. Throughout, we take UCB1 (Auer et al., 2002a) as the base algorithm in QPM-D. Joulani et al. (2013) also present a direct modification of the UCB1 algorithm. All of these algorithms achieve the stated regret. None of them require *any* knowledge of the delay distributions, but they all rely heavily upon the non-anonymous nature of the observations.

While these results are encouraging, the assumption that the rewards are observed individually in a non-anonymous fashion is limiting for most practical applications with delays (e.g., recall the applications discussed earlier). How big is the price to be paid for receiving only aggregated anonymous feedback? Our main result is to prove that essentially there is no extra price to be paid provided that the value of the expected delay (or a bound on it) is available. In particular, this means that detailed knowledge of which action led to a particular delayed reward can be replaced

by the much weaker requirement that the expected delay, or a bound on it, is known. Figure 5.1 summarizes the relationship between the non-delayed, the delayed and the new problem by showing the leading terms of the regret. In all cases, the dominant term is $\sqrt{KT}$. Hence, asymptotically, the delayed, aggregated anonymous feedback problem is no more difficult than the standard multi-armed bandit problem.

5.1.1 Our Techniques and Results

We now consider what sort of algorithm will be able to achieve the aforementioned results for the MABDAAF problem. Since the player only observes delayed, aggregated anonymous rewards, the first problem we face is how to even estimate the mean reward of individual actions. Due to the delays and anonymity, it appears that to be able to estimate the mean reward of an action, the player wants to have played it consecutively for long stretches. Indeed, if the stretches are sufficiently long compared to the mean delay, the observations received during the stretch will mostly consist of rewards of the action played in that stretch. This naturally leads to considering algorithms that *switch actions rarely* and this is indeed the basis of our approach.

Several popular MAB algorithms are based on choosing the action with the largest upper confidence bound (UCB) in each round (see Section 2.2.1). UCB-style algorithms tend to switch arms frequently and will only play the optimal arm for long stretches if a unique optimal arm exists. Therefore, for MABDAAF, we will consider alternative algorithms where arm switching is more tightly controlled. The design of such algorithms goes back at least to the work of Agrawal et al. (1988) where the problem of bandits with switching costs was studied. The general idea of these rarely switching algorithms is to gradually eliminate suboptimal arms by playing arms in phases and comparing each arm's upper confidence bound to the lower confidence bound of a leading arm at the end of each phase. Generally, this sort of rarely switching algorithm switches arms only $O(\log T)$ times. We base our approach on one such

algorithm, the so-called Improved UCB[1] algorithm of Auer and Ortner (2010).

Using a rarely switching algorithm alone will not be sufficient for MABDAAF. The remaining problem, and where the bulk of our contribution lies, is to construct appropriate confidence bounds and adjust the length of the periods of playing each arm to account for the delayed, aggregated anonymous feedback. In particular, in the confidence bounds attention must be paid to fine details: it turns out that unless the variance of the observations is dealt with, there is a blow-up by a multiplicative factor of K. We avoid this by an improved analysis involving Freedman's inequality (Freedman, 1975). Further, to handle the dependencies between the number of plays of each arm and the past rewards, we combine Doob's optimal skipping theorem (Doob, 1953) and Azuma-Hoeffding inequalities. Using a rarely switching algorithm for MABDAAF means we must also consider the dependencies between the elimination of arms in one phase and the corruption of observations in the next phase (i.e. past plays can influence both whether an arm is still active and the corruption of the next plays). We deal with this through careful algorithmic design.

Using the above, we provide an algorithm that achieves worst case regret of $O(\sqrt{KT \log K} + K\mathbb{E}[\tau] \log T)$ using only knowledge of the expected delay, $\mathbb{E}[\tau]$. We then show that this regret can be improved by using a more careful martingale argument that exploits the fact that our algorithm is designed to remove most of the dependence between the corruption of future observations and the elimination of arms. Particularly, if the delays are bounded with known bound, $0 \leq d \leq \sqrt{T/K}$, we can recover worst case regret of $O(\sqrt{KT \log K} + K\mathbb{E}[\tau])$, matching that in (Joulani et al., 2013). If the delays are unbounded but have known variance $\mathbb{V}(\tau)$, we show that the problem independent regret can be reduced to $O(\sqrt{KT \log K} + K\mathbb{E}[\tau] + K\mathbb{V}(\tau))$.

[1] The adjective "Improved" indicates that the algorithm improves upon the regret bounds achieved by UCB1. The improvement replaces $\log(T)/\Delta_j$ by $\log(T\Delta_j^2)/\Delta_j$ in the regret bound.

5.1.2 Related Work

We have already discussed several of the most relevant works to our own. However, there has also been other work looking at different flavors of the bandit problem with delayed (non-anonymous) feedback. A detailed review of this work is given in Section 2.3.4. Neu et al. (2010) and Cesa-Bianchi et al. (2016) consider non-stochastic bandits with fixed constant delays; Dudik et al. (2011) look at stochastic contextual bandits with a constant delay and Desautels et al. (2014) consider Gaussian Process bandits with a bounded stochastic delay. The general observation that delay causes an additive regret penalty in stochastic bandits and a multiplicative one in adversarial bandits is made in (Joulani et al., 2013). The empirical performance of K-armed stochastic bandit algorithms in delayed settings was investigated in (Chapelle and Li, 2011). A further related problem is the 'batched bandit' problem studied by Perchet et al. (2016). Here the player must fix a set of time points at which to collect feedback on all plays leading up to that point. Vernade et al. (2017) consider delayed Bernoulli bandits where some observations could also be censored (e.g., no conversion is ever actually observed if the delay exceeds some threshold) but require complete knowledge of the delay distribution. Crucially, here and in all the aforementioned works, the feedback is always assumed to take the form of arm-reward pairs and knowledge of the assignment of rewards to arms underpins the suggested algorithms, rendering them unsuitable for MABDAAF. To the best of our knowledge, ours is the first work to develop algorithms to deal with delayed, aggregated anonymous feedback in the bandit setting.

5.1.3 Organization

The reminder of this chapter is organized as follows: In the next section (Section 5.2) we give the formal problem definition. We present our algorithm in Section 5.3. In Section 5.4, we discuss the performance of our algorithm under various delay assump-

tions; known expectation, bounded support with known bound and expectation, and known variance and expectation. This is followed by a numerical illustration of our results in Section 5.5. We conclude in Section 5.6.

5.2 Problem Definition

There are $K > 1$ actions or arms in the set $\mathcal{A}$. Each action $j \in \mathcal{A}$ is associated with a reward distribution, ζ_j, and a delay distribution, δ_j. The reward distribution is supported in $[0, 1]$ and the delay distribution is supported on $\mathbb{N} \doteq \{0, 1, \dots\}$. We denote by μ_j the mean of ζ_j, $\mu^* = \mu_{j^*} = \max_j \mu_j$ and define $\Delta_j = \mu^* - \mu_j$ to be the *reward gap*, that is the expected loss of reward each time action j is chosen instead of an optimal action. Let $(R_{l,j}, \tau_{l,j})_{l \in \mathbb{N}, j \in \mathcal{A}}$ be an infinite array of random variables defined on the probability space (Ω, Σ, P) which are mutually independent. Further, $R_{l,j}$ follows the distribution ζ_j and $\tau_{l,j}$ follows the distribution δ_j. The meaning of these random variables is that if the player plays action j at time l, a payoff of $R_{l,j}$ will be added to the aggregated feedback that the player receives at the end of the $(l + \tau_{l,j})$th play. Formally, if $J_l \in \mathcal{A}$ denotes the action chosen by the player at time $l = 1, 2, \dots$, then the observation received at the end of the tth play is

$$X_t = \sum_{l=1}^{t} \sum_{j=1}^{K} R_{l,j} \times \mathbb{I}\{l + \tau_{l,j} = t, J_l = j\}.$$

For the remainder, we will consider i.i.d. delays across arms. We also assume discrete delay distributions, although most results hold for continuous delays by redefining the event $\{\tau_{l,j} = t - l\}$ as $\{t - l - 1 < \tau_{l,j} \leq t - l\}$ in X_t. In our analysis, we will sum over stochastic index sets. For a stochastic index set I and random variables $\{Z_n\}_{n \in \mathbb{N}}$ we denote such sums as $\sum_{t \in I} Z_t \doteq \sum_{t \in \mathbb{N}} \mathbb{I}\{t \in I\} \times Z_t$.

Regret definition In most bandit problems, the regret is the cumulative loss due to not playing an optimal action. In the case of delayed feedback, there are several possible ways to define the regret. One option is to consider only the loss of the rewards *received* before horizon T (as in (Vernade et al., 2017)). However, we will not use this definition. Instead, as in (Joulani et al., 2013), we consider the loss of all *generated* rewards and define the (pseudo-)regret by

$$\mathfrak{R}_T = \sum_{t=1}^{T}(\mu^* - \mu_{J_t}) = T\mu^* - \sum_{t=1}^{T}\mu_{J_t}.$$

This includes the rewards received after the horizon T and does not penalize large delays as long as an optimal action is taken. This definition is natural since, in practice, the player should eventually receive all outstanding reward.

Lai and Robbins (1985) showed that the regret of any algorithm for the standard MAB problem must satisfy,

$$\liminf_{T\to\infty} \frac{\mathbb{E}[\mathfrak{R}_T]}{\log(T)} \geq \sum_{j:\Delta_j>0} \frac{\Delta_j}{KL(\zeta_j, \zeta^*)}, \tag{5.1}$$

where $KL(\zeta_j, \zeta^*)$ is the KL-divergence between the reward distributions of arm j and an optimal arm. Theorem 4 of Vernade et al. (2017) shows that the lower bound in (5.1) also holds for delayed feedback bandits with no censoring and their alternative definition of regret. We therefore suspect (5.1) should hold for MABDAAF. However, due to the specific problem structure, finding a lower bound for MABDAAF is non-trivial and remains an open problem.

Assumptions on delay distribution For our algorithm for MABDAAF, we need some assumptions on the delay distribution. We assume that the expected delay, $\mathbb{E}[\tau]$, is bounded and known. This quantity is used in the algorithm.

Assumption 1. *The expected delay, $\mathbb{E}[\tau]$, is bounded and known to the algorithm.*

We then show that under some further mild assumptions on the delay, we can obtain better algorithms with even more efficient regret guarantees. We consider two settings: delay distributions with bounded support, and bounded variance.

Assumption 2 (Bounded support)**.** *There exists some constant $d > 0$ known to the algorithm such that the support of the delay distribution is bounded by d.*

Assumption 3 (Bounded variance)**.** *The variance, $\mathbb{V}(\tau)$, of the delay is bounded and known to the algorithm.*

In fact the known expected value and known variance assumption can be replaced by a 'known upper bound' on the expected value and variance respectively. However, for simplicity, in the remaining we use $\mathbb{E}[\tau]$ and $\mathbb{V}(\tau)$ directly. The next sections provide algorithms and regret analyses for different combinations of the above assumptions.

5.3 Our Algorithm

Our algorithm is a phase-based elimination algorithm based on the Improved UCB algorithm by Auer and Ortner (2010). The general structure is as follows. In each phase, each arm is played multiple times consecutively. At the end of the phase, the observations received are used to update mean estimates, and any arm with an estimated mean below the best estimated mean by a gap larger than a 'separation gap tolerance' is eliminated. This separation tolerance is decreased exponentially over phases, so that it is very small in later phases, eliminating all but the best arm(s) with high probability. An alternative formulation of the algorithm is that at the end of a phase, any arm with an upper confidence bound lower than the best lower confidence bound is eliminated. These confidence bounds are computed so that with high probability they are more (less) than the true mean, but within the separation gap tolerance. The phase lengths are then carefully chosen to ensure that the confidence

Algorithm 5.1 Optimism for Delayed, Aggregated Anonymous Feedback (ODAAF)

Require: A set of arms, $\mathcal{A}$; a horizon, T; choice of n_m for each phase $m = 1, 2, \ldots$.

Initialization: Set $\tilde{\Delta}_1 = 1/2$ (tolerance), the set of active arms $\mathcal{A}_1 = \mathcal{A}$. Let $T_i(1) = \emptyset, i \in A$, $m = 1$ (phase index), $t = 1$ (round index)

while $t \leq T$ **do**

Step 1: Play arms.

for $j \in \mathcal{A}_m$ **do**

Let $T_j(m) = T_j(m-1)$

while $|T_j(m)| \leq n_m$ **and** $t \leq T$ **do**

Play arm j, receive X_t. Add t to $T_j(m)$. Increment t by 1.

end while

end for

Step 2: Eliminate sub-optimal arms.

For every arm in $j \in \mathcal{A}_m$, compute $\bar{X}_{m,j}$ as the average of observations at time steps $t \in T_j(m)$. That is,

$$\bar{X}_{m,j} = \frac{1}{|T_j(m)|} \sum_{t \in T_j(m)} X_t \,.$$

Construct $\mathcal{A}_{m+1}$ by eliminating actions $j \in \mathcal{A}_m$ with

$$\bar{X}_{m,j} + \tilde{\Delta}_m < \max_{j' \in \mathcal{A}_m} \bar{X}_{m,j'} \,.$$

Step 3: Decrease Tolerance.

Set $\tilde{\Delta}_{m+1} = \frac{\tilde{\Delta}_m}{2}$.

Step 4: Bridge period.

Pick an arm $j \in \mathcal{A}_{m+1}$ and play it $\nu_m = n_m - n_{m-1}$ times while incrementing $t \leq T$. Discard all observations from this period. Do not add t to $T_j(m)$.

Increment phase index m.

end while

bounds hold. Here we assume that the horizon T is known, but we expect that this can be relaxed as in (Auer and Ortner, 2010).

Algorithm overview Our algorithm, ODAAF, is given in Algorithm 5.1. It operates in phases $m = 1, 2, \ldots$. Define $\mathcal{A}_m$ to be the set of active arms in phase m. The algorithm takes parameter n_m which defines the number of samples of each active arm required by the end of phase m.

In Step 1 of phase m of the algorithm, each active arm j is played repeatedly for

$n_m - n_{m-1}$ steps. We record all timesteps where arm j was played in the first m phases (excluding bridge periods) in the set $T_j(m)$. The active arms are played in any arbitrary but fixed order. In Step 2, the n_m observations from timesteps in $T_j(m)$ are averaged to obtain a new estimate $\bar{X}_{m,j}$ of μ_j. Arm j is eliminated if $\bar{X}_{m,j}$ is further than $\tilde{\Delta}_m$ from $\max_{j' \in \mathcal{A}_m} \bar{X}_{m,j'}$.

A further nuance in the algorithm structure is the '*bridge period*' (see Figure 5.2). The algorithm picks an active arm $j \in \mathcal{A}_{m+1}$ to play in this bridge period for $n_m - n_{m-1}$ steps. The observations received during the bridge period are discarded, and not used for computing confidence intervals. The significance of the bridge period is that it breaks the dependence between confidence intervals calculated in phase m and the delayed payoffs seeping into phase $m+1$. Without the bridge period this dependence would impair the validity of our confidence intervals. However, we suspect that, in practice, it may be possible to remove it.

Choice of $\mathbf{n_m}$ A key element of our algorithm design is the careful choice of n_m. Since n_m determines the number of times each active (possibly suboptimal) arm is played, it clearly has an impact on the regret. Furthermore, n_m needs to be chosen so that the confidence bounds on the estimation error hold with given probability. The main challenge is developing these confidence bounds from delayed, aggregated anonymous feedback. Handling this form of feedback involves a credit assignment problem of deciding which samples can be used for a given arm's mean estimation, since each sample is an aggregate of rewards from multiple previously played arms. This credit assignment problem would be hopeless in a passive learning setting without further information on how the samples were generated. Our algorithm utilizes the power of active learning to design the phases in such a way that the feedback can be effectively 'decensored' without losing too many samples.

A naive approach to defining the confidence bounds for delays bounded by a

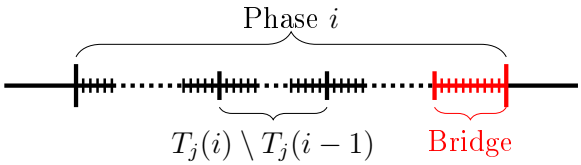


Figure 5.2: An example of phase i of our algorithm.

constant $d \geq 0$ would be to observe that,

$$\left| \sum_{t \in T_j(m) \setminus T_j(m-1)} X_t - \sum_{t \in T_j(m) \setminus T_j(m-1)} R_{t,j} \right| \leq d,$$

since all rewards are in $[0, 1]$. Then we could use Hoeffding's inequality to bound R_{t,J_t} (see Section 5.F) and select

$$n_m = \frac{C_1 \log(T\tilde{\Delta}_m^2)}{\tilde{\Delta}_m^2} + \frac{C_2 m d}{\tilde{\Delta}_m}$$

for some constants C_1, C_2. This corresponds to worst case regret of $O(\sqrt{KT \log K} + K \log(T) d)$. For $d \gg \mathbb{E}[\tau]$ and large T, this is significantly worse than that of Joulani et al. (2013). In Section 5.4, we show that, surprisingly, it is possible to recover the same rate of regret as Joulani et al. (2013), but this requires a significantly more nuanced argument to get tighter confidence bounds and smaller n_m. In the next section, we describe this improved choice of n_m for every phase $m \in \mathbb{N}$ and its implications on the regret, for each of the three cases mentioned previously: (i) Known and bounded expected delay (Assumption 1), (ii) Bounded delay with known bound and expected value (Assumptions 1 and 2), (iii) Delay with known and bounded variance and expectation (Assumptions 1 and 3).

5.4 Regret Analysis

In this section, we specify the choice of parameters n_m and provide regret guarantees for Algorithm 5.1 for each of the three previously mentioned cases.

5.4.1 Known and Bounded Expected Delay

First, we consider the setting with the weakest assumption on the delay distribution: we only assume that the expected delay, $\mathbb{E}[\tau]$, is bounded and known. No assumption on the support or variance of the delay distribution is made. The regret analysis for this setting will not use the bridge period, so Step 4 of the algorithm could be omitted in this case.

Choice of $\mathbf{n_m}$ Here, we use Algorithm 5.1 with

$$n_m = \frac{C_1 \log(T\tilde{\Delta}_m^2)}{\tilde{\Delta}_m^2} + \frac{C_2 m \mathbb{E}[\tau]}{\tilde{\Delta}_m} \tag{5.2}$$

for some large enough constants C_1, C_2. The exact value of n_m is given in Equation (5.14) in Section 5.B.

Estimation of error bounds We bound the error between $\bar{X}_{m,j}$ and μ_j by $\tilde{\Delta}_m/2$. In order to do this we first bound the corruption of the observations received during timesteps $T_j(m)$ due to delays.

Fix a phase m and arm $j \in \mathcal{A}_m$. Then the observations X_t in the period $t \in T_j(m) \setminus T_j(m-1)$ are composed of two types of rewards: a subset of rewards from plays of arm j in this period, and delayed rewards from some of the plays before this period. The expected value of observations from this period would be $(n_m - n_{m-1})\mu_j$ but for the rewards entering and leaving this period due to the delays. Since the reward is bounded by 1, a simple observation is that the expected discrepancy between

the sum of observations in this period and the quantity $(n_m - n_{m-1})\mu_j$ is bounded by the expected delay $\mathbb{E}[\tau]$,

$$\mathbb{E}\left[\sum_{t\in T_j(m)\setminus T_j(m-1)} (X_t - \mu_j)\right] \leq \mathbb{E}[\tau]. \tag{5.3}$$

Summing this over phases $\ell = 1, \dots m$ gives a bound

$$|\mathbb{E}[\bar{X}_{m,j}] - \mu_j| \leq \frac{m\mathbb{E}[\tau]}{|T_j(m)|} = \frac{m\mathbb{E}[\tau]}{n_m}. \tag{5.4}$$

Note that given the choice of n_m in (5.2), the above is smaller than $\tilde{\Delta}_m/2$, when large enough constants are used. Using this, along with concentration inequalities and the choice of n_m from (5.2), we can obtain the following high probability bound. A detailed proof is provided in Section 5.B.1.

Lemma 5.1. *Under Assumption 1 and the choice of n_m given by (5.2), the estimates $\bar{X}_{m,j}$ constructed by Algorithm 5.1 satisfy the following: For every fixed arm j and phase m, with probability $1 - \frac{3}{T\tilde{\Delta}_m^2}$, either $j \notin \mathcal{A}_m$, or:*

$$\bar{X}_{m,j} - \mu_j \leq \tilde{\Delta}_m/2\,.$$

Regret bounds Using Lemma 5.1, we derive the following regret bounds in the current setting.

Theorem 5.2. *Under Assumption 1, the expected regret of Algorithm 5.1 is upper bounded as*

$$\mathbb{E}[\mathfrak{R}_T] \leq \sum_{\substack{j=1\\ j\neq j^*}}^{K} O\left(\frac{\log(T\Delta_j^2)}{\Delta_j} + \log(1/\Delta_j)\mathbb{E}[\tau]\right). \tag{5.5}$$

Proof. Given Lemma 5.1, the proof of Theorem 5.2 closely follows the analysis of the

Improved UCB algorithm of Auer and Ortner (2010). Lemma 5.1 and the elimination condition in Algorithm 5.1 ensure that, with high probability, any suboptimal arm j will be eliminated by phase $m_j = \log(1/\Delta_j)$, thus incurring regret at most $n_{m_j}\Delta_j$ We then substitute in n_{m_j} from (5.2), and sum over all suboptimal arms. A detailed proof is in Section 5.B.2. As in (Auer and Ortner, 2010), we avoid a union bound over all arms (which would result in an extra $\log K$) by *(i)* reasoning about the regret of each arm individually, and *(ii)* bounding the regret resulting from erroneously eliminating the optimal arm by carefully controlling the probability it is eliminated in each phase. □

Considering the worst-case values of Δ_j (roughly $\sqrt{K/T}$), we obtain the following problem independent bound.

Corollary 5.3. *For any problem instance satisfying Assumption 1, the expected regret of Algorithm 5.1 satisfies*

$$\mathbb{E}[\mathfrak{R}_T] \leq O(\sqrt{KT\log(K)} + K\mathbb{E}[\tau]\log(T)).$$

5.4.2 Delay with Bounded Support

If the delay is bounded by some constant $d \geq 0$ and a single arm is played repeatedly for long enough, we can restrict the number of arms corrupting the observation X_t at a given time t. In fact, if each arm j is played consecutively for more than d rounds, then at any time $t \in T_j(m)$, the observation X_t will be composed of the rewards from at most two arms: the current arm j, and the previous arm j'. Further, from the elimination condition, with high probability, arm j' will have been eliminated if it is clearly suboptimal. We can then recursively use the confidence bounds for arms j and j' from the previous phase to bound $|\mu_j - \mu_{j'}|$. Below, we formalize this intuition to obtain a tighter bound on $|\bar{X}_{m,j} - \mu_j|$ for every arm j and phase m, when each active arm is played a specified number of times per phase.

Choice of $\mathbf{n_m}$ Here, we define,

$$n_m = \frac{C_1 \log(T\tilde{\Delta}_m^2)}{\tilde{\Delta}_m^2} + \frac{C_2 \mathbb{E}[\tau]}{\tilde{\Delta}_m} + \min\left\{md, \frac{C_3 \log(T\tilde{\Delta}_m^2)}{\tilde{\Delta}_m^2} + \frac{C_4 m \mathbb{E}[\tau]}{\tilde{\Delta}_m}\right\} \tag{5.6}$$

for some large enough constants C_1, C_2, C_3, C_4 (see Section 5.C, Equation (5.18) for the exact values). This choice of n_m means that for large d, we essentially revert back to the choice of n_m from (5.2) for the unbounded case, and we gain nothing by using the bound on the delay. However, if d is not large, the choice of n_m in (5.6) is smaller than (5.2) since the second term now scales with $\mathbb{E}[\tau]$ rather than $m\mathbb{E}[\tau]$.

Estimation of error bounds In this setting, by the elimination condition and bounded delays, the expectation of each reward entering $T_j(m)$ will be within $\tilde{\Delta}_{m-1}$ of μ_j, with high probability. Then, using knowledge of the upper bound of the support of τ, we can obtain a tighter bound and get an error bound similar to Lemma 5.1 with the smaller value of n_m in (5.6). We prove the following proposition. Since $\tilde{\Delta}_m = 2^{-m}$, this is considerably tighter than (5.3).

Proposition 5.4. *Assume $n_i - n_{i-1} \geq d$ for phases $i = 1, \ldots, m$. Define $\mathcal{E}_{m-1}$ as the event that all arms $j \in \mathcal{A}_m$ satisfy error bounds $|\bar{X}_{m-1,j} - \mu_j| \leq \tilde{\Delta}_{m-1}/2$. Then, for every arm $j \in \mathcal{A}_m$,*

$$\mathbb{E}\left[\sum_{t \in T_j(m) \setminus T_j(m-1)} (X_t - \mu_j) \middle| \mathcal{E}_{m-1}\right] \leq \tilde{\Delta}_{m-1} \mathbb{E}[\tau].$$

Proof. (Sketch). Consider a fixed arm $j \in \mathcal{A}_m$. The expected value of the sum of observations X_t for $t \in T_j(m) \setminus T_j(m-1)$ would be $(n_m - n_{m-1})\mu_j$ were it not for some rewards entering and leaving this period due to the delays. Because of the i.i.d. assumption on the delay, in expectation, the number of rewards leaving the period is roughly the same as the number of rewards entering this period, i.e., $\mathbb{E}[\tau]$ (conditioning on $\mathcal{E}_{m-1}$ does not effect this due to the bridge period). Since $n_m - n_{m-1} \geq d$, the

reward coming into the period $T_j(m) \setminus T_j(m-1)$ can only be from the previous arm j'. All rewards leaving the period are from arm j. Therefore the expected difference between rewards entering and leaving the period is $(\mu_j - \mu_{j'})\mathbb{E}[\tau]$. Then, if μ_j is close to $\mu_{j'}$, the total reward leaving the period is compensated by total reward entering. Due to the bridge period, even when j is the first arm played in phase m, $j' \in \mathcal{A}_m$, so it was not eliminated in phase $m-1$. By the elimination condition in Algorithm 5.1, if the error bounds $|\bar{X}_{m-1,j} - \mu_j| \le \tilde{\Delta}_{m-1}/2$ are satisfied for all arms in $\mathcal{A}_m$, then $|\mu_j - \mu_{j'}| \le \tilde{\Delta}_{m-1}$. This gives the result. □

Repeatedly using Proposition 5.4 we get,

$$\sum_{i=1}^{m} \mathbb{E}\left[\sum_{t \in T_j(i) \setminus T_j(i-1)} (X_t - \mu_j) \middle| \mathcal{E}_{i-1}\right] \le 2\mathbb{E}[\tau]$$

since $\sum_{i=1}^{m} \tilde{\Delta}_{i-1} = \sum_{i=0}^{m-1} 2^{-i} \le 2$. Then, observe that $\mathbb{P}(\mathcal{E}_i^C)$ is small. This bound is an improvement of a factor of m compared to (5.4). For the regret analysis, we derive a high probability version of the above result. Using this, and the choice of $n_m \ge \Omega\left(\frac{\log(T\tilde{\Delta}_m^2)}{\tilde{\Delta}_m^2} + \frac{\mathbb{E}[\tau]}{\tilde{\Delta}_m}\right)$ from (5.6), for large enough constants, we derive the following lemma. A detailed proof is given in Section 5.C.1.

Lemma 5.5. *Under Assumptions 1 of known expected delay and 2 of bounded delays, and choice of n_m given in (5.6), the estimates $\bar{X}_{m,j}$ obtained by Algorithm 5.1 satisfy the following: For any arm j and phase m, with probability at least $1 - \frac{12}{T\tilde{\Delta}_m^2}$, either $j \notin \mathcal{A}_m$ or*

$$\bar{X}_{m,j} - \mu_j \le \tilde{\Delta}_m/2.$$

Regret bounds We now give regret bounds for this case.

Theorem 5.6. *Under Assumption 1 and bounded delay Assumption 2, the expected*

regret of Algorithm 5.1 satisfies

$$\mathbb{E}[\mathfrak{R}_T] \leq \sum_{j=1;j\neq j^*}^{K} O\left(\frac{\log(T\Delta_j^2)}{\Delta_j} + \mathbb{E}[\tau] + \min\left\{d, \frac{\log(T\Delta_j^2)}{\Delta_j} + \log(\frac{1}{\Delta_j})\mathbb{E}[\tau]\right\}\right).$$

Proof. (Sketch). Given Lemma 5.5, the proof is similar to that of Theorem 5.2. The full proof is in Section 5.C.2. □

Then, if $d \leq \sqrt{\frac{T\log K}{K}} + \mathbb{E}[\tau]$, we get the following problem independent regret bound which matches that of Joulani et al. (2013).

Corollary 5.7. *For any problem instance satisfying Assumptions 1 and 2 with* $d \leq \sqrt{\frac{T\log K}{K}} + \mathbb{E}[\tau]$*, the expected regret of Algorithm 5.1 satisfies*

$$\mathbb{E}[\mathfrak{R}_T] \leq O(\sqrt{KT\log(K)} + K\mathbb{E}[\tau]).$$

5.4.3 Delay with Bounded Variance

If the delay is unbounded but well behaved in the sense that we know (a bound on) the variance, then we can obtain similar regret bounds to the bounded delay case. Intuitively, delays from the previous phase will only corrupt observations in the current phase if their delays exceed the length of the bridge period. We control this by using the bound on the variance in Chebychev's inequality to bound the tails of the delay distributions.

Choice of $\mathbf{n_m}$ Let $\mathbb{V}(\tau)$ be the known variance (or bound on the variance) of the delay, as in Assumption 3. Then, we use Algorithm 5.1 with the following value of n_m,

$$n_m = C_1\frac{\log(T\tilde{\Delta}_m^2)}{\tilde{\Delta}_m^2} + C_2\frac{\mathbb{E}[\tau] + \mathbb{V}(\tau)}{\tilde{\Delta}_m} \tag{5.7}$$

for some large enough constants C_1, C_2. The exact value of n_m is given in Section 5.D, Equation (5.25).

Regret bounds We get the following instance specific and problem independent regret bound in this case.

Theorem 5.8. *Under Assumption 1 and Assumption 3 of known (bound on) the expectation and variance of the delay, and choice of n_m from (5.7), the expected regret of Algorithm 5.1 can be upper bounded by,*

$$\mathbb{E}[\mathfrak{R}_T] \leq \sum_{j=1:\mu_j \neq \mu^*}^{K} O\left(\frac{\log(T\Delta_j^2)}{\Delta_j} + \mathbb{E}[\tau] + \mathbb{V}(\tau)\right).$$

Proof. (Sketch). See Section 5.D.2. We use Chebychev's inequality to get a result similar to Lemma 5.5 and then use a similar argument to the bounded delay case. □

Corollary 5.9. *For any problem instance satisfying Assumptions 1 and 3, the expected regret of Algorithm 5.1 satisfies*

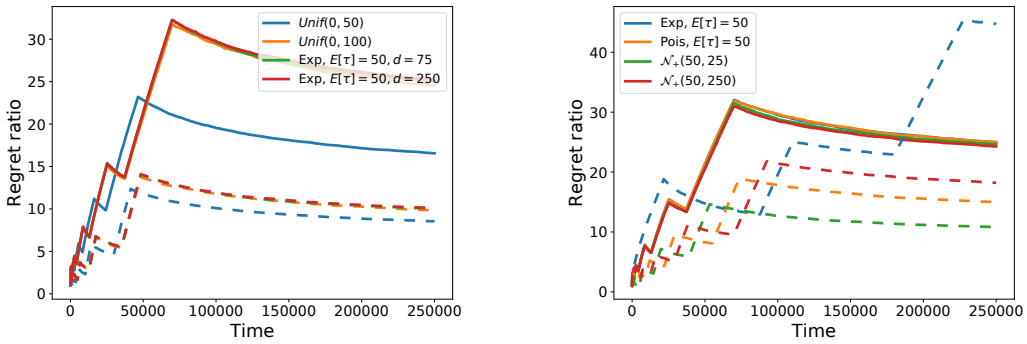
$$\mathbb{E}[\mathfrak{R}_T] \leq O(\sqrt{KT\log(K)} + K\mathbb{E}[\tau] + K\mathbb{V}(\tau)).$$

Remark If $\mathbb{E}[\tau] \geq 1$, then the delay penalty can be reduced to $O(K\mathbb{E}[\tau]+K\mathbb{V}(\tau)/\mathbb{E}[\tau])$ (see Section 5.D).

Thus, it is sufficient to know a bound on the variance to obtain regret bounds similar to those in the bounded delay case. Note that this approach is not possible just using knowledge of the expected delay since we cannot guarantee that with high probability, most of the reward entering phase i is from an arm active in phase $i-1$.

5.5 Experimental Results

We compared the performance of our algorithm (under different assumptions) to QPM-D (Joulani et al., 2013) in various experimental settings. In these experiments,



(a) Bounded delays. Ratios of regret of ODAAF (solid lines) and ODAAF-B (dotted lines) to that of QPM-D.

(b) Unbounded delays. Ratios of regret of ODAAF (solid lines) and ODAAF-V (dotted lines) to that of QPM-D.

Figure 5.3: The ratios of regret of variants of our algorithm to that of QPM-D for different delay distributions.

our aim was to investigate the effect of the delay on the performance of the algorithms. In order to focus on this, we used a simple setup of two arms with Bernoulli rewards and $\boldsymbol{\mu} = (0.5, 0.6)$. In every experiment, we ran each algorithm to horizon $T = 250000$ and used UCB1 (Auer et al., 2002a) as the base algorithm in QPM-D. The regret was averaged over 200 replications. For ease of reading, we define ODAAF to be our algorithm using only knowledge of the expected delay, with n_m defined as in (5.2) and run without a bridge period, and ODAAF-B and ODAAF-V to be the versions of Algorithm 5.1 that use a bridge period and information on the bounded support or the finite variance of the delay to define n_m as in (5.6) and (5.7) respectively.

We tested the algorithms with different delay distributions. In the first case, we considered bounded delay distributions whereas in the second case, the delays were unbounded. In Figure 5.3a, we plotted the ratios of the regret of ODAAF and ODAAF-B (with knowledge of d, the delay bound) to the regret of QPM-D for bounded delay distributions. We see that in all cases the ratios converge to a constant. This shows that the regret of our algorithm is essentially of the same order as that of QPM-D. Our algorithm predetermines the number of times to play each active arm per phase (the randomness appears in whether an arm is active), so the jumps in the regret are

it changing arm. This occurs at the same points in all replications.

Figure 5.3b shows a similar story for unbounded delays with mean $\mathbb{E}[\tau] = 50$ (where $\mathcal{N}_+$ denotes the the half normal distribution). The ratios of the regret of ODAAF and ODAAF-V (with knowledge of the delay variance) to the regret of QPM-D again converge to constants. Note that in this case, these constants, and the location of the jumps, vary with the delay distribution and $\mathbb{V}(\tau)$. When the variance of the delay is small, it can be seen that using the variance information leads to improved performance. However, for exponential delays where $\mathbb{V}(\tau) = \mathbb{E}[\tau]^2$, the large variance causes n_m to be large and so the suboptimal arm is played more, increasing the regret. In this case ODAAF-V had only just eliminated the suboptimal arm at time T.

It can also be illustrated experimentally that the regret of our algorithms and that of QPM-D all increase linearly in $\mathbb{E}[\tau]$. This is shown in Section 5.E. We also provide an experimental comparison to Vernade et al. (2017) in Section 5.E.

5.6 Conclusion

We have studied an extension of the multi-armed bandit problem to bandits with delayed, aggregated anonymous feedback. Here, a sum of observations is received after some stochastic delay and we do not learn which arms contributed to each observation. In this more difficult setting, we have proven that, surprisingly, it is possible to develop an algorithm that performs comparably to those for the simpler delayed feedback bandits problem, where the assignment of rewards to plays is known. Particularly, using only knowledge of the expected delay, our algorithm matches the worst case regret of Joulani et al. (2013) up to a logarithmic factor. This logarithmic factor can be removed using an improved analysis and slightly more information about the delay; if the delay is bounded, we achieve the same worst case regret as Joulani et al. (2013), and for unbounded delays with known finite variance, we have an extra

additive $\mathbb{V}(\tau)$ term. We supported these claims experimentally. Note that while our algorithm matches the order of regret of QPM-D, the constants are worse. Hence, it is an open problem to find algorithms with better constants.

5.A Supplementary Material

5.A.1 Table of Notation

For ease of reading, we define here key notation that will be used in this section.

T	:	The horizon.
Δ_j	:	The gap between the mean of the optimal arm and the mean of arm j, $\Delta_j = \mu^* - \mu_j$.
$\tilde{\Delta}_m$	:	The approximation to Δ_j at round m of the ODAAF algorithm, $\tilde{\Delta}_m = \frac{1}{2^m}$.
n_m	:	The number of samples of an active arm j ODAAF needs by the end of round m.
ν_m	:	The number of times each arm is played in phase m, $\nu_m = n_m - n_{m-1}$.
d	:	The bound on the delay in the case of bounded delay.
m_j	:	The first round of the ODAAF algorithm where $\tilde{\Delta}_m < \Delta_j/2$.
M_j	:	The random variable representing the round arm j is eliminated in.
$T_j(m)$	:	The set of all time point where arm j is played up to (and including) round m.
X_t	:	The reward received at time t (from any possible past plays).
$R_{t,j}$	:	The reward generated by playing arm j at time t.
$\tau_{t,j}$	:	The delay associated with playing arm j at time t.
$\mathbb{E}[\tau]$	:	The expected delay (assuming i.i.d. delays).

$\mathbb{V}(\tau)$: The variance of the delay (assuming i.i.d. delays).

$\bar{X}_{m,j}$: The estimated reward of arm j in phase m. See Algorithm 5.1 for the definition.

S_m : The start point of the mth phase. See Section 5.A.2 for more details.

U_m : The end point of the mth phase. See Section 5.A.2 for more details.

$S_{m,j}$: The start point of phase m of playing arm j. See Section 5.A.2 for more details.

$U_{m,j}$: The end point of phase m of playing arm j. See Section 5.A.2 for more details.

$\mathcal{A}_m$: The set of active arms in round m of the ODAAF algorithm.

$A_{i,t}, B_{i,t}, C_{i,t}$: The contribution of the reward generated at time t in certain intervals relating to phase i to the corruption. See (5.11) for the exact definitions.

$\mathcal{G}_t$: The smallest σ-algebra containing all information up to time t, see (5.8) for a definition.

5.A.2 Beginning and End of Phases

We formalize here some notation that will be used throughout the analysis to denote the start and end points of each phase. Define the random variables S_i and U_i for each phase $i = 1, \ldots, m$ to be the start and end points of the phase. Then let $S_{i,j}$, $U_{i,j}$ denote the start and end points of playing arm j in phase i. See Figure 5.4 for details. By convention, let $S_{i,j} = U_{i,j} = \infty$ if arm j is not active in phase i, $S_i = U_i = \infty$ if the algorithm never reaches phase i, and let $S_{0,j} = U_{0,j} = S_0 = U_0 = 0$ for all j. It is important to point out that n_m are deterministic so at the end of any phase $m-1$, once we have eliminated sub-optimal arms, we also know which arms are in $\mathcal{A}_m$ and consequently the start and end points of phase m. Furthermore, since we play arms

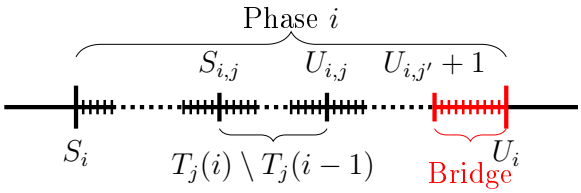


Figure 5.4: An example of phase i of our algorithm. Here j' is the last active arm played in phase i.

in a given order, we also know the specific rounds when we start and finish playing each active arm in phase m. Hence, at any time step t in phase m, S_m, U_m, S_{m+1} and $U_{m,j}, S_{m,j}$ for all active arms $j \in \mathcal{A}_m$ will be known. More formally, define the filtration $\{\mathcal{G}_t\}_{t=0}^{\infty}$ where

$$\mathcal{G}_t = \sigma(X_1, \ldots, X_t, \tau_{1,J_1}, \ldots, \tau_{t,J_t}, R_{1,J_1}, \ldots, R_{t,J_t}, J_1, \ldots, J_t) \tag{5.8}$$

and $\mathcal{G}_0 = \{\emptyset, \Omega\}$. This means the joint events like $\{S_i \leq t\} \cap \{S_{i,j} = s'\} \in \mathcal{G}_t$ for all $s' \in \mathbb{N}$, $j \in \mathcal{A}$.

5.A.3 Useful Results

For our analysis, we will need Freedman's version of Bernstein's inequality for the right-tail of martingales with bounded increments:

Theorem 5.10 (Freedman's version of Bernstein's inequality; Theorem 1.6 of (Freedman, 1975)). *Let $\{Y_k\}_{k=0}^{\infty}$ be a real-valued martingale with respect to the filtration $\{\mathcal{F}_k\}_{k=0}^{\infty}$ with increments $\{Z_k\}_{k=1}^{\infty}$: $\mathbb{E}[Z_k|\mathcal{F}_{k-1}] = 0$ and $Z_k = Y_k - Y_{k-1}$, for $k = 1, 2, \ldots$. Assume that the difference sequence is uniformly bounded on the right: $Z_k \leq b$ almost surely for $k = 1, 2, \ldots$. Define the predictable variation process $W_k = \sum_{j=1}^{k} \mathbb{E}[Z_j^2|\mathcal{F}_{j-1}]$ for $k = 1, 2, \ldots$. Then, for all $t \geq 0$, $\sigma^2 > 0$,*

$$\mathbb{P}\left(\exists k \geq 0 : Y_k \geq t \text{ and } W_k \leq \sigma^2\right) \leq \exp\left\{-\frac{t^2/2}{\sigma^2 + bt/3}\right\}.$$

This result implies that if for some deterministic constant, σ^2, $W_k \leq \sigma^2$ holds almost surely, then $\mathbb{P}(Y_k \geq t) \leq \exp\left\{-\frac{t^2/2}{\sigma^2+bt/3}\right\}$ holds for any $t \geq 0$.

We will also make use of the following technical lemma which combines the Hoeffding-Azuma inequality and Doob's optional skipping theorem (Theorem 2.3 in Chapter VII of (Doob, 1953))):

Lemma 5.11. *Fix the positive integers m, n and let $a, c \in \mathbb{R}$. Let $\mathcal{F} = \{\mathcal{F}_t\}_{t=0}^n$ be a filtration, $(\epsilon_t, Z_t)_{t=1,2,\ldots,n}$ be a sequence of $\{0,1\} \times \mathbb{R}$-valued random variables such that for $t \in \{1, 2, \ldots, n\}$, ϵ_t is $\mathcal{F}_{t-1}$-measurable, Z_t is $\mathcal{F}_t$-measurable, $\mathbb{E}[Z_t|\mathcal{F}_{t-1}] = 0$ and $Z_t \in [a, a+c]$. Further, assume that $\sum_{s=1}^n \epsilon_s \leq m$ with probability one. Then, for any $\lambda > 0$,*

$$\mathbb{P}\left(\sum_{t=1}^n \epsilon_t Z_t \geq \lambda\right) \leq \exp\left\{-\frac{2\lambda^2}{c^2 m}\right\}. \tag{5.9}$$

Proof. This lemma appeared in a slightly more general form (where $n = \infty$ is allowed) as Lemma A.1 in the paper by Szita and Szepesvári (2011) so we refer the reader to the proof there. □

5.B Results for Known and Bounded Expected Delay

5.B.1 High Probability Bounds

Lemma 5.1. *Under Assumption 1 and the choice of n_m given by (5.2), the estimates $\bar{X}_{m,j}$ constructed by Algorithm 5.1 satisfy the following: For every fixed arm j and phase m, with probability $1 - \frac{3}{T\tilde{\Delta}_m^2}$, either $j \notin \mathcal{A}_m$, or:*

$$\bar{X}_{m,j} - \mu_j \leq \tilde{\Delta}_m/2\,.$$

Proof. Let

$$w_m = \frac{4\log(T\tilde{\Delta}_m^2)}{3n_m} + \sqrt{\frac{2\log(T\tilde{\Delta}_m^2)}{n_m}} + \frac{3m\mathbb{E}[\tau]}{n_m}. \tag{5.10}$$

We first show that with probability greater than $1-\frac{3}{T\tilde{\Delta}_m^2}$, $j \notin \mathcal{A}_m$ or $\frac{1}{n_m}\sum_{t\in T_j(m)}(X_t - \mu_j) \leq w_m$.

For arm j and phase m, assume $j \in \mathcal{A}_m$. For notational simplicity we will use in the following $\mathbb{I}_i\{H\} := \mathbb{I}\{H \cap \{j \in \mathcal{A}_i\}\} \leq \mathbb{I}\{H\}$ for any event H. If $j \in \mathcal{A}_m$ for a particular experiment ω then $\mathbb{I}_i(H)(\omega) = \mathbb{I}(H)(\omega)$. Then for any phase $i \leq m$ and time t, define,

$$A_{i,t} = R_{t,J_t}\mathbb{I}\{\tau_{t,J_t} + t \geq S_i\},\ B_{i,t} = R_{t,J_t}\mathbb{I}\{\tau_{t,J_t} + t \geq S_{i,j}\},\ C_{i,t} = R_{t,J_t}\mathbb{I}\{\tau_{t,J_t} + t > U_{i,j}\}, \tag{5.11}$$

and note that since $S_{i,j} = U_{i,j} = \infty$ if arm j is not active in phase i, we have the equalities $\mathbb{I}_i\{\tau_{t,J_t} + t \geq S_{i,j}\} = \mathbb{I}\{\tau_{t,J_t} + t \geq S_{i,j}\}$ and $\mathbb{I}_i\{\tau_{t,J_t} + t > U_{i,j}\} = \mathbb{I}\{\tau_{t,J_t} + t > U_{i,j}\}$. Define the filtration $\{\mathcal{G}_s\}_{s=0}^{\infty}$ by $\mathcal{G}_0 = \{\Omega, \emptyset\}$ and

$$\mathcal{G}_t = \sigma(X_1, \ldots, X_t, J_1, \ldots, J_t, \tau_{1,J_1}, \ldots, \tau_{t,J_t}, R_{1,J_1}, \ldots R_{t,J_t}). \tag{5.12}$$

Then, we use the decomposition,

$$\begin{aligned}
\sum_{i=1}^{m}\sum_{t=S_{i,j}}^{U_{i,j}}(X_t-\mu_j) &\le \sum_{i=1}^{m}\Bigg(\sum_{t=S_{i-1,j}}^{S_{i,j}-1} R_{t,J_t}\mathbb{I}_i\{\tau_{t,J_t}+t\ge S_{i,j}\}+\sum_{t=S_{i,j}}^{U_{i,j}}(R_{t,J_t}-\mu_j)\\
&\qquad -\sum_{t=S_{i,j}}^{U_{i,j}} R_{t,J_t}\mathbb{I}_i\{\tau_{t,J_t}+t>U_{i,j}\}\Bigg)\\
&\le \sum_{i=1}^{m}\Bigg(\sum_{t=S_{i-1,j}}^{S_i-1} R_{t,J_t}\mathbb{I}\{\tau_{t,J_t}+t\ge S_i\}+\sum_{t=S_i}^{S_{i,j}-1} R_{t,J_t}\mathbb{I}\{\tau_{t,J_t}+t\ge S_{i,j}\}\\
&\qquad +\sum_{t=S_{i,j}}^{U_{i,j}}(R_{t,J_t}-\mu_j)-\sum_{t=S_{i,j}}^{U_{i,j}} R_{t,J_t}\mathbb{I}\{\tau_{t,J_t}+t>U_{i,j}\}\Bigg)\\
&= \sum_{i=1}^{m}\Bigg(\sum_{t=S_{i-1,j}}^{S_i-1} A_{i,t}+\sum_{t=S_i}^{S_{i,j}-1} B_{i,t}+\sum_{t=S_{i,j}}^{U_{i,j}}(R_{t,J_t}-\mu_j)-\sum_{t=S_{i,j}}^{U_{i,j}} C_{i,t}\Bigg)\\
&= \sum_{i=1}^{m}\sum_{t=S_{i,j}}^{U_{i,j}}(R_{t,J_t}-\mu_j)+\sum_{t=1}^{S_{m,j}} Q_t-\sum_{t=1}^{U_{m,j}} P_t\\
&= \underbrace{\sum_{i=1}^{m}\sum_{t=S_{i,j}}^{U_{i,j}}(R_{t,J_t}-\mu_j)}_{\text{Term I.}}+\underbrace{\sum_{t=1}^{S_{m,j}}(Q_t-\mathbb{E}[Q_t|\mathcal{G}_{t-1}])}_{\text{Term II.}}+\underbrace{\sum_{t=1}^{U_{m,j}}(\mathbb{E}[P_t|\mathcal{G}_{t-1}]-P_t)}_{\text{Term III.}}\\
&\qquad +\underbrace{\Bigg(\sum_{t=1}^{S_{m,j}}\mathbb{E}[Q_t|\mathcal{G}_{t-1}]-\sum_{t=1}^{U_{m,j}}\mathbb{E}[P_t|\mathcal{G}_{t-1}]\Bigg)}_{\text{Term IV.}},
\end{aligned} \tag{5.13}$$

where,

$$\begin{aligned}
Q_t &= \sum_{i=1}^{m}(A_{i,t}\mathbb{I}\{S_{i-1,j}\le t\le S_i-1\}+B_{i,t}\mathbb{I}\{S_i\le t\le S_{i,j}-1\})\\
P_t &= \sum_{i=1}^{m} C_{i,t}\mathbb{I}\{S_{i,j}\le t\le U_{i,j}\}.
\end{aligned}$$

Recall that the filtration $\{\mathcal{G}_s\}_{s=0}^{\infty}$ is defined by

$$\mathcal{G}_0=\{\Omega,\emptyset\},\quad \mathcal{G}_t=\sigma(X_1,\dots,X_t,J_1,\dots,J_t,\tau_{1,J_1},\dots,\tau_{t,J_t},R_{1,J_1},\dots R_{t,J_t})$$

and we have defined $S_{i,j} = \infty$ if arm j is eliminated before phase i and $S_i = \infty$ if the algorithm stops before reaching phase i.

Outline of proof We will bound each term of the above decomposition in (5.13) in turn, however first we need to prove several intermediary results. For term II., we will use Freedman's inequality so we first need Lemma 5.12 to show that $Z_t = Q_t - \mathbb{E}[Q_t|\mathcal{G}_{t-1}]$ is a martingale difference and Lemma 5.13 to bound the variance of the sum of the Z_t's. Similarly, for term III., in Lemma 5.14, we show that $Z'_t = \mathbb{E}[P_t|\mathcal{G}_{t-1}] - P_t$ is a martingale difference and bound its variance in Lemma 5.15. In Lemma 5.16, we consider term IV. and bound the conditional expectations of $A_{i,t}, B_{i,t}, C_{i,t}$. Finally, in Lemma 5.17, we bound term I. using Lemma 5.11. We then combine the bounds on all terms together to conclude the proof.

Lemma 5.12. *Let* $Y_s = \sum_{t=1}^{s}(Q_t - \mathbb{E}[Q_t|\mathcal{G}_{t-1}])$ *for all* $s \geq 1$, $Y_0 = 0$. *Then* $\{Y_s\}_{s=0}^{\infty}$ *is a martingale with respect to the filtration* $\{\mathcal{G}_s\}_{s=0}^{\infty}$ *with increments* $Z_s = Y_s - Y_{s-1} = Q_s - \mathbb{E}[Q_s|\mathcal{G}_{s-1}]$ *satisfying* $\mathbb{E}[Z_s|\mathcal{G}_{s-1}] = 0, Z_s \leq 1$ *for all* $s \geq 1$.

Proof. To show $\{Y_s\}_{s=0}^{\infty}$ is a martingale with respect to $\{\mathcal{G}_s\}_{s=0}^{\infty}$, we need to show that Y_s is G_s measurable for all s and $\mathbb{E}[Y_s|\mathcal{G}_{s-1}] = Y_{s-1}$.

Measurability: First note that by definition of $\mathcal{G}_s$, τ_{t,J_t}, R_{t,J_t} are all $\mathcal{G}_s$-measurable for $t \leq s$. Then, for each i, either t is in a phase later than i so $S_{i-1,j}$ and S_i are $\mathcal{G}_t$-measurable, or $S_{i-1,j}$ and S_i are not $\mathcal{G}_t$-measurable, but $\mathbb{I}\{t \geq S_{i,j}\} = 0$ so $\mathbb{I}\{t \geq S_{i,j}\}$ is $\mathcal{G}_t$-measurable. In the first case, since $S_{i-1,j}$ and S_i are $\mathcal{G}_t$-measurable $A_{i,t}\mathbb{I}\{S_{i-1,j} \leq t \leq S_i - \nu_i\}$ is $\mathcal{G}_t$-measurable. In the second case, $A_{i,t}\mathbb{I}\{S_{i-1,j} \leq t \leq S_i - 1\} = A_{i,t}\mathbb{I}\{\{S_{i-1,j} \leq t\}\mathbb{I}\{t \leq S_i - 1\} = 0$ so it is also $\mathcal{G}_t$-measurable. Similarly, if t is after S_i , S_i and $S_{i,j}$ will be $\mathcal{G}$-measurable or $\mathbb{I}\{S_i \leq t \leq S_{i,j} - 1\} = 0$. In both cases, $B_{i,t}\mathbb{I}\{S_i \leq t \leq S_{i,j} - 1\}$ is $\mathcal{G}_t$-measurable. Hence, Q_t is $\mathcal{G}_t$-measurable, and also Q_t is $\mathcal{G}_s$ measurable for any $s \geq t$. It then follows that Y_s is $\mathcal{G}_s$-measurable for all s.

Expectation: Since Q_t is $\mathcal{G}_s$ measurable for all $t \leq s$,

$$\begin{aligned}
\mathbb{E}[Y_s|\mathcal{G}_{s-1}] &= \mathbb{E}\bigg[\sum_{t=1}^{s}(Q_t - \mathbb{E}[Q_t|\mathcal{G}_{t-1}])|\mathcal{G}_{s-1}\bigg] \\
&= \mathbb{E}\bigg[\sum_{t=1}^{s-1}(Q_t - \mathbb{E}[Q_t|\mathcal{G}_{t-1}])|\mathcal{G}_{s-1}\bigg] + \mathbb{E}[(Q_s - \mathbb{E}[Q_s|\mathcal{G}_{s-1}])|\mathcal{G}_{s-1}] \\
&= \sum_{t=1}^{s-1}(Q_t - \mathbb{E}[Q_t|\mathcal{G}_{t-1}]) + \mathbb{E}[Q_s|\mathcal{G}_{s-1}] - \mathbb{E}[Q_s|\mathcal{G}_{s-1}] \\
&= \sum_{t=1}^{s-1}(Q_t - \mathbb{E}[Q_t|\mathcal{G}_{t-1}]) = Y_{s-1}
\end{aligned}$$

Hence, $\{Y_s\}_{s=0}^{\infty}$ is a martingale with respect to the filtration $\{\mathcal{G}_s\}_{s=0}^{\infty}$.

Increments: For any $s = 1, \ldots,$ we have that

$$Z_s = Y_s - Y_{s-1} = \sum_{t=1}^{s}(Q_t - \mathbb{E}[Q_t|\mathcal{G}_{t-1}]) - \sum_{t=1}^{s-1}(Q_t - \mathbb{E}[Q_t|\mathcal{G}_{t-1}]) = Q_s - \mathbb{E}[Q_s|\mathcal{G}_{s-1}].$$

Then,

$$\mathbb{E}[Z_s|\mathcal{G}_{s-1}] = \mathbb{E}[Q_s - \mathbb{E}[Q_s|\mathcal{G}_{s-1}]|\mathcal{G}_{s-1}] = \mathbb{E}[Q_s|\mathcal{G}_{s-1}] - \mathbb{E}[Q_s|\mathcal{G}_{s-1}] = 0.$$

Lastly, since for any t, there is only one i where one of $\mathbb{I}\{S_{i-1,j} \leq t \leq S_i - 1\} = 1$ or $\mathbb{I}\{S_i \leq t \leq S_{i,j} - 1\} = 1$ (and they cannot both be one), and since $R_{t,J_t} \in [0,1]$, $A_{i,t}, B_{i,t} \leq 1$, so it follows that $Z_s = Q_s - \mathbb{E}[Q_s|\mathcal{G}_{s-1}] \leq 1$ for all s. □

Lemma 5.13. *For any t, let $Z_t = Q_t - \mathbb{E}[Q_t|\mathcal{G}_{t-1}]$, then, for any $s < S_{m,j}$,*

$$\sum_{t=1}^{s}\mathbb{E}[Z_t^2|\mathcal{G}_{t-1}] \leq 2m\mathbb{E}[\tau].$$

Proof. First note that

$$\sum_{t=1}^{s} \mathbb{E}[Z_t^2|\mathcal{G}_{t-1}] = \sum_{t=1}^{s} \mathbb{V}(Q_t|\mathcal{G}_{t-1}) \leq \sum_{t=1}^{s} \mathbb{E}[Q_t^2|\mathcal{G}_{t-1}]$$
$$= \sum_{t=1}^{s} \mathbb{E}\left[\left(\sum_{i=1}^{m}(A_{i,t}\mathbb{I}\{S_{i-1,j} \leq t \leq S_i - 1\} + B_{i,t}\mathbb{I}\{S_i \leq t \leq S_{i,j} - 1\})\right)^2 \middle| \mathcal{G}_{t-1}\right].$$

Then, given $\mathcal{G}_{t-1}$, all indicator terms $\mathbb{I}\{S_{i-1,j} \leq t \leq S_i - 1\}$ and $\mathbb{I}\{S_i \leq S_{i,j} - 1\}$ for all $i = 1, \ldots, m$ are measurable and only one can be non zero. Hence, all interaction terms in the expansion of the quadratic are 0 and so we are left with

$$\begin{aligned}
&\sum_{t=1}^{s} \mathbb{E}[Z_t^2|\mathcal{G}_{t-1}] \\
&\leq \sum_{t=1}^{s} \mathbb{E}\left[\left(\sum_{i=1}^{m}(A_{i,t}\mathbb{I}\{S_{i-1,j} \leq t \leq S_i - 1\} + B_{i,t}\mathbb{I}\{S_i \leq t \leq S_{i,j} - 1\})\right)^2 \middle| \mathcal{G}_{t-1}\right] \\
&= \sum_{t=1}^{s} \mathbb{E}\left[\sum_{i=1}^{m}(A_{i,t}^2\mathbb{I}\{S_{i-1,j} \leq t \leq S_i - 1\}^2 + B_{i,t}^2\mathbb{I}\{S_i \leq t \leq S_{i,j} - 1\}^2) \middle| \mathcal{G}_{t-1}\right] \\
&= \sum_{i=1}^{m}\sum_{t=1}^{s} \mathbb{E}[A_{i,t}^2\mathbb{I}\{S_{i-1,j} \leq t \leq S_i - 1\}|\mathcal{G}_{t-1}] \\
&\qquad + \sum_{i=1}^{m}\sum_{t=1}^{s} \mathbb{E}[B_{i,t}^2\mathbb{I}\{S_i \leq t \leq S_{i,j} - 1\}|\mathcal{G}_{t-1}] \\
&\leq \sum_{i=1}^{m}\sum_{t=S_{i-1,j}}^{S_i-1} \mathbb{E}[A_{i,t}^2|\mathcal{G}_{t-1}] + \sum_{i=1}^{m}\sum_{t=S_i}^{S_{i,j}-1} \mathbb{E}[B_{i,t}^2|\mathcal{G}_{t-1}].
\end{aligned}$$

Then, for any $i \geq 1$,

$$\begin{aligned}
\sum_{t=S_{i-1,j}}^{S_i-1} \mathbb{E}[A_{i,t}^2|\mathcal{G}_{t-1}] &= \sum_{t=S_{i-1,j}}^{S_i-1} \mathbb{E}[R_{t,J_t}^2 \mathbb{I}\{\tau_{t,J_t} + t \geq S_i\}|\mathcal{G}_{t-1}] \\
&\leq \sum_{t=S_{i-1,j}}^{S_i-1} \mathbb{E}[\mathbb{I}\{\tau_{t,J_t} + t \geq S_i\}|\mathcal{G}_{t-1}] \\
&= \sum_{s=0}^{\infty}\sum_{s'=s}^{\infty} \mathbb{I}\{S_{i-1,j} = s, S_i = s'\} \sum_{t=s}^{s'-1} \mathbb{E}[\mathbb{I}\{\tau_{t,J_t} + t \geq S_i\}|\mathcal{G}_{t-1}] \\
&= \sum_{s=0}^{\infty}\sum_{s'=s}^{\infty}\sum_{t=s}^{s'-1} \mathbb{E}[\mathbb{I}\{S_{i-1,j} = s, S_i = s', \tau_{t,J_t} + t \geq S_i\}|\mathcal{G}_{t-1}] \\
&\qquad\qquad \text{(Since } \{t \geq S_{i-1,j}, S_i = s'\} \in \mathcal{G}_{t-1}\text{)} \\
&= \sum_{s=0}^{\infty}\sum_{s'=s}^{\infty}\sum_{t=s}^{s'-1} \mathbb{E}[\mathbb{I}\{S_{i-1,j} = s, S_i = s', \tau_{t,J_t} + t \geq s'\}|\mathcal{G}_{t-1}] \\
&= \sum_{s=0}^{\infty}\sum_{s'=s}^{\infty} \mathbb{I}\{S_{i-1,j} = s, S_i = s'\} \sum_{t=s}^{s'-1} \mathbb{P}(\tau_{t,J_t} + t \geq s') \\
&\qquad\qquad \text{(Since } \{t \geq S_{i-1,j}, S_i = s'\} \in \mathcal{G}_{t-1}\text{)} \\
&\leq \sum_{s=0}^{\infty}\sum_{s'=s}^{\infty} \mathbb{I}\{S_{i-1,j} = s, S_i = s'\} \sum_{l=0}^{\infty} \mathbb{P}(\tau > l) \\
&\leq \mathbb{E}[\tau].
\end{aligned}$$

Likewise, for any $i \geq 1$,

$$\begin{aligned}
\sum_{t=S_i}^{S_{i,j}-1} \mathbb{E}[B_{i,t}^2|\mathcal{G}_{t-1}] &= \sum_{t=S_i}^{S_{i,j}-1} \mathbb{E}[R_{t,J_t}^2 \mathbb{I}\{\tau_{t,J_t} + t \geq S_{i,j}\}|\mathcal{G}_{t-1}] \\
&\leq \sum_{t=S_i}^{S_{i,j}-1} \mathbb{E}[\mathbb{I}\{\tau_{t,J_t} + t \geq S_{i,j}\}|\mathcal{G}_{t-1}] \\
&= \sum_{s=0}^{\infty}\sum_{s'=s}^{\infty} \mathbb{I}\{S_i = s, S_{i,j} = s'\} \sum_{t=s}^{s'-1} \mathbb{E}[\mathbb{I}\{\tau_{t,J_t} + t \geq S_{i,j}\}|\mathcal{G}_{t-1}] \\
&= \sum_{s=0}^{\infty}\sum_{s'=s}^{\infty}\sum_{t=s}^{s'-1} \mathbb{E}[\mathbb{I}\{S_i = s, S_{i,j} = s', \tau_{t,J_t} + t \geq S_{i,j}\}|\mathcal{G}_{t-1}] \\
&\qquad\qquad \text{(Since } \{t \geq S_i, S_{i,j} = s'\} \in \mathcal{G}_{t-1}\text{)} \\
&= \sum_{s=0}^{\infty}\sum_{s'=s}^{\infty}\sum_{t=s}^{s'-1} \mathbb{E}[\mathbb{I}\{S_i = s, S_{i,j} = s', \tau_{t,J_t} + t \geq s'\}|\mathcal{G}_{t-1}] \\
&= \sum_{s=0}^{\infty}\sum_{s'=s}^{\infty} \mathbb{I}\{S_i = s, S_{i,j} = s'\} \sum_{t=s}^{s'-1} \mathbb{P}(\tau_{t,J_t} + t \geq s') \\
&\qquad\qquad \text{(Since } \{t \geq S_i, S_{i,j} = s'\} \in \mathcal{G}_{t-1}\text{)} \\
&\leq \sum_{s=0}^{\infty}\sum_{s'=s}^{\infty} \mathbb{I}\{S_i = s, S_{i,j} = s'\} \sum_{l=0}^{\infty} \mathbb{P}(\tau \geq l) \\
&\leq \mathbb{E}[\tau].
\end{aligned}$$

Hence, combining both terms and summing over the phases m gives the result. □

Lemma 5.14. *Let $Y'_s = \sum_{t=1}^{s}(\mathbb{E}[P_s|\mathcal{G}_{s-1}] - P_s)$ for all $s \geq 1$, $Y'_0 = 0$. Then $\{Y'_s\}_{s=0}^{\infty}$ is a martingale with respect to the filtration $\{\mathcal{G}_s\}_{s=0}^{\infty}$ with increments $Z'_s = Y'_s - Y'_{s-1} = \mathbb{E}[P_s|\mathcal{G}_{s-1}] - P_s$ satisfying $\mathbb{E}[Z'_s|\mathcal{G}_{s-1}] = 0, Z'_s \leq 1$ for all $s \geq 1$.*

Proof. The proof is similar to that of Lemma 5.12. To show $\{Y'_s\}_{s=0}^{\infty}$ is a martingale with respect to $\{\mathcal{G}_s\}_{s=0}^{\infty}$, we need to show that Y'_s is G_s measurable for all s and $\mathbb{E}[Y'_s|\mathcal{G}_{s-1}] = Y'_{s-1}$.

Measurability: As before, by definition of $\mathcal{G}_s$, τ_{t,J_t}, R_{t,J_t} are all $\mathcal{G}_s$-measurable for $t \leq s$. Also, we can reduce measurability again to measurability of $\mathbb{I}\{\tau_{s,J_s} + s \geq U_{i,j}, S_{i,j} \leq s \leq U_{i,j}\}$. But, $\{U_{i,j} = s'\} \cap \{S_{i,j} \leq s\} \in \mathcal{G}_s$ for all $s' \in \mathbb{N}$ and Y'_s is adapted to $\mathcal{G}_s$.

Increments: For any $s \geq 1$, we have that

$$Z'_s = Y'_s - Y'_{s-1} = \sum_{t=1}^{s}(\mathbb{E}[P_t|\mathcal{G}_{t-1}] - P_t) - \sum_{t=1}^{s-1}(\mathbb{E}[P_t|\mathcal{G}_{t-1}] - P_t) = \mathbb{E}[P_s|\mathcal{G}_{s-1}] - P_s.$$

Then,

$$\mathbb{E}[Z'_s|\mathcal{G}_{s-1}] = \mathbb{E}[\mathbb{E}[P_s|\mathcal{G}_{s-1}] - P_s|\mathcal{G}_{s-1}] = \mathbb{E}[P_s|\mathcal{G}_{s-1}] - \mathbb{E}[P_s|\mathcal{G}_{s-1}] = 0.$$

Lastly, since for any t and $\omega \in \Omega$, there is at most one i for which $\mathbb{I}\{S_{i,j} \leq t \leq U_{i,j}\} = 1$, and by definition of R_{t,J_t}, $C_{i,t} \leq 1$, so it follows that $Z'_s = \mathbb{E}[P_s|\mathcal{G}_{s-1}] - P_s \leq 1$ for all s. □

Lemma 5.15. *For any t, let $Z'_t = \mathbb{E}[P_t|\mathcal{G}_{t-1}] - P_t$, then*

$$\sum_{t=1}^{U_{m,j}} \mathbb{E}[{Z'_t}^2|\mathcal{G}_{t-1}] \leq m\mathbb{E}[\tau].$$

Proof. The proof is similar to that of Lemma 5.13. First note that

$$\begin{aligned}\sum_{t=1}^{U_{m,j}} \mathbb{E}[{Z'_t}^2|\mathcal{G}_{t-1}] &= \sum_{t=1}^{U_{m,j}} \mathbb{V}(P_t|\mathcal{G}_{t-1}) \leq \sum_{t=1}^{U_{m,j}} \mathbb{E}[P_t^2|\mathcal{G}_{t-1}] \\ &= \sum_{t=1}^{U_{m,j}} \mathbb{E}\left[\left(\sum_{i=1}^{m}(C_{i,t}\mathbb{I}\{S_{i,j} \leq t \leq U_{i,j}\}\right)^2 |\mathcal{G}_{t-1}\right].\end{aligned}$$

Then, given $\mathcal{G}_{t-1}$, all indicator terms $\mathbb{I}\{S_{i,j} \leq t \leq U_{i,j}\}$ for $i = 1, \ldots, m$ are measurable and at most one can be non zero. Hence, all interaction terms are 0 and so we are

left with

$$\begin{aligned}
\sum_{t=1}^{U_{m,j}} \mathbb{E}[Z_t'^2|\mathcal{G}_{t-1}] &\leq \sum_{t=1}^{U_{m,j}} \mathbb{E}\left[\left(\sum_{i=1}^{m}(C_{i,t}\mathbb{I}\{S_{i,j} \leq t \leq U_{i,j}\}\right)^2 |\mathcal{G}_{t-1}\right] \\
&= \sum_{i=1}^{m}\sum_{t=1}^{U_{m,j}} \mathbb{E}[C_{i,t}^2\mathbb{I}\{S_{i,j} \leq t \leq U_{i,j}\}|\mathcal{G}_{t-1}] \\
&\leq \sum_{i=1}^{m}\sum_{t=S_{i,j}}^{U_{i,j}} \mathbb{E}[C_{i,t}^2|\mathcal{G}_{t-1}] \qquad \text{(since the indicator is } \mathcal{G}_{t-1}\text{-measurable)} \\
&= \sum_{i=1}^{m}\sum_{t=S_{i,j}}^{U_{i,j}} \mathbb{E}[R_{t,J_t}^2\mathbb{I}\{\tau_{t,J_t} + t > U_{i,j}\}|\mathcal{G}_{t-1}] \\
&\leq \sum_{i=1}^{m}\sum_{t=S_{i,j}}^{U_{i,j}} \mathbb{E}[\mathbb{I}\{\tau_{t,J_t} + t > U_{i,j}\}|\mathcal{G}_{t-1}] \\
&= \sum_{i=1}^{m}\sum_{s=0}^{\infty}\sum_{s'=s}^{\infty}\mathbb{I}\{S_{i,j} = s, U_{i,j} = s'\}\sum_{t=s}^{s'}\mathbb{E}[\mathbb{I}\{\tau_{t,J_t} + t > U_{i,j}\}|\mathcal{G}_{t-1}] \\
&= \sum_{i=1}^{m}\sum_{s=0}^{\infty}\sum_{s'=s}^{\infty}\sum_{t=s}^{s'}\mathbb{E}[\mathbb{I}\{S_{i,j} = s, U_{i,j} = s', \tau_{t,J_t} + t > U_{i,j}\}|\mathcal{G}_{t-1}] \\
&= \sum_{i=1}^{m}\sum_{s=0}^{\infty}\sum_{s'=s}^{\infty}\sum_{t=s}^{s'}\mathbb{E}[\mathbb{I}\{S_{i,j} = s, U_{i,j} = s', \tau_{t,J_t} + t > s'\}|\mathcal{G}_{t-1}] \\
&= \sum_{i=1}^{m}\sum_{s=0}^{\infty}\sum_{s'=s}^{\infty}\mathbb{I}\{S_{i,j} = s, U_{i,j} = s'\}\sum_{t=s}^{s'}\mathbb{P}(\tau_{t,J_t} + t > s') \\
&\leq \sum_{i=1}^{m}\sum_{s=0}^{\infty}\sum_{s'=s}^{\infty}\mathbb{I}\{S_{i,j} = s, U_{i,j} = s'\}\sum_{l=0}^{\infty}\mathbb{P}(\tau > l) \\
&\leq \sum_{i=1}^{m}\mathbb{E}[\tau] = m\mathbb{E}[\tau].
\end{aligned}$$

□

Lemma 5.16. *For $A_{i,t}, B_{i,t}$ and $C_{i,t}$ defined as in (5.11), let $\nu_i = n_i - n_{i-1}$ be the number of times each arm is played in phase i and j_i' be the arm played directly before arm j in phase i. Then, it holds that, for any arm j and phase $i \geq 1$,*

(i) $\displaystyle\sum_{t=S_{i-1,j}}^{S_i-1} \mathbb{E}[A_{i,t}|\mathcal{G}_{t-1}] \leq \mathbb{E}[\tau]$

(ii) $\displaystyle\sum_{t=S_i}^{S_{i,j}-1} \mathbb{E}[B_{i,t}|\mathcal{G}_{t-1}] \leq \mathbb{E}[\tau] + \mu_{j_i'} \sum_{l=0}^{\nu_i} \mathbb{P}(\tau > l)$

(iii) $\displaystyle\sum_{t=S_{i,j}}^{U_{i,j}} \mathbb{E}[C_{i,t}|\mathcal{G}_{t-1}] = \mu_j \sum_{l=0}^{\nu_i} \mathbb{P}(\tau > l)$

Proof. We prove each statement individually. Several of the proofs are similar to those appearing in Lemmas 5.13 and 5.15.

Statement (i):

$$\begin{aligned}
\sum_{t=S_{i-1,j}}^{S_i-1} \mathbb{E}[A_{i,t}|\mathcal{G}_{t-1}] &\leq \sum_{t=S_{i-1,j}}^{S_i-1} \mathbb{E}[\mathbb{I}\{\tau_{t,J_t} + t \geq S_i\}|\mathcal{G}_{t-1}] \\
&= \sum_{s=0}^{\infty}\sum_{s'=s}^{\infty} \mathbb{I}\{S_{i-1,j} = s, S_i = s'\} \sum_{t=s}^{s'-1} \mathbb{E}[\mathbb{I}\{\tau_{t,J_t} + t \geq S_i\}|\mathcal{G}_{t-1}] \\
&= \sum_{s=0}^{\infty}\sum_{s'=s}^{\infty}\sum_{t=s}^{s'-1} \mathbb{E}[\mathbb{I}\{S_{i-1,j} = s, S_i = s', \tau_{t,J_t} + t \geq S_i\}|\mathcal{G}_{t-1}] \\
&\qquad\qquad \text{(Since } \{t \geq S_{i-1,j}, S_i = s'\} \in \mathcal{G}_{t-1}\text{)} \\
&= \sum_{s=0}^{\infty}\sum_{s'=s}^{\infty}\sum_{t=s}^{s'-1} \mathbb{E}[\mathbb{I}\{S_{i-1,j} = s, S_i = s', \tau_{t,J_t} + t \geq s'\}|\mathcal{G}_{t-1}] \\
&= \sum_{s=0}^{\infty}\sum_{s'=s}^{\infty} \mathbb{I}\{S_{i-1,j} = s, S_i = s'\} \sum_{t=s}^{s'-1} \mathbb{P}(\tau_{t,J_t} + t \geq s') \\
&\qquad\qquad \text{(Since } \{t \geq S_{i-1,j}, S_i = s'\} \in \mathcal{G}_{t-1}\text{)} \\
&\leq \sum_{s=0}^{\infty}\sum_{s'=s}^{\infty} \mathbb{I}\{S_{i-1,j} = s, S_i = s'\} \sum_{l=0}^{\infty} \mathbb{P}(\tau > l) \\
&= \sum_{l=0}^{\infty} \mathbb{P}(\tau > l) = \mathbb{E}[\tau].
\end{aligned}$$

Statement (iii):

$$\begin{aligned}
\sum_{t=S_{i,j}}^{U_{i,j}} \mathbb{E}[C_{i,t}|\mathcal{G}_{t-1}] &= \sum_{t=S_{i,j}}^{U_{i,j}} \mathbb{E}[R_{t,J_t}\mathbb{I}\{\tau_{t,J_t}+t > U_{i,j}\}|\mathcal{G}_{t-1}] \\
&= \sum_{s=0}^{\infty}\sum_{s'=s}^{\infty}\mathbb{I}\{S_{i,j}=s, U_{i,j}=s'\}\sum_{t=s}^{s'}\mathbb{E}[R_{t,J_t}\mathbb{I}\{\tau_{t,J_t}+t > U_{i,j}\}|\mathcal{G}_{t-1}] \\
&= \sum_{s=0}^{\infty}\sum_{s'=s}^{\infty}\sum_{t=s}^{s'}\mathbb{E}[R_{t,J_t}\mathbb{I}\{S_{i,j}=s, U_{i,j}=s', \tau_{t,J_t}+t > U_{i,j}\}|\mathcal{G}_{t-1}] \\
&\qquad\qquad \text{(Since } \{S_{i,j}=s, U_{i,j}=s'\} \in \mathcal{G}_{t-1} \text{ for } s \le t) \\
&= \sum_{s=0}^{\infty}\sum_{s'=s}^{\infty}\sum_{t=s}^{s'}\mathbb{E}[R_{t,J_t}\mathbb{I}\{S_{i,j}=s, U_{i,j}=s', \tau_{t,J_t}+t > s'\}|\mathcal{G}_{t-1}] \\
&= \sum_{s=0}^{\infty}\sum_{s'=s}^{\infty}\mathbb{I}\{S_{i,j}=s, U_{i,j}=s'\}\sum_{t=s}^{s'}\mu_j\mathbb{P}(\tau_{t,J_t}+t > s') \\
&\text{(Since } \{S_{i,j}=s, U_{i,j}=s'\} \in \mathcal{G}_{t-1} \text{ and given } \mathcal{G}_{t-1}, R_{t,J_t} \text{ and } \tau_{t,J_t} \text{ are independent)} \\
&= \sum_{s=0}^{\infty}\sum_{s'=s}^{\infty}\mathbb{I}\{S_{i,j}=s, U_{i,j}=s'\}\mu_j\sum_{l=0}^{\nu_i}\mathbb{P}(\tau > l) \\
&= \mu_j\sum_{l=0}^{\nu_i}\mathbb{P}(\tau > l)
\end{aligned}$$

Statement (ii): For statement (ii), we have that for $(i,j) \neq (1,1)$,

$$\sum_{t=S_i}^{S_{i,j}-1}\mathbb{E}[B_{i,t}|\mathcal{G}_{t-1}] = \sum_{t=S_i}^{S_{i,j}-\nu_{i-1}-2}\mathbb{E}[B_{i,t}|\mathcal{G}_{t-1}] + \sum_{t=S_{i,j}-\nu_{i-1}-1}^{S_{i,j}-1}\mathbb{E}[B_{i,t}|\mathcal{G}_{t-1}].$$

Then, $S_{i,j}$ is $\mathcal{G}_{t-1}$ measurable for $t \ge S_i$, so we can use the same technique as for statement (i) to bound the first term. For the second term, since we will only be playing arm j_i' for $S_{i,j}-\nu_{i-1}-1, \ldots, S_{i,j}-1$, we can use the same technique as for statement (iii). Hence,

$$\sum_{t=S_i}^{S_{i,j}-1}\mathbb{E}[B_{i,t}|\mathcal{G}_{t-1}] \le \sum_{l=\nu_{i-1}+1}^{\infty}\mathbb{P}(\tau > l) + \mu_{j_i'}\sum_{l=0}^{\nu_{i-1}}\mathbb{P}(\tau > l) \le \mathbb{E}[\tau] + \mu_{j_i'}\sum_{l=0}^{\nu_i}\mathbb{P}(\tau > l).$$

Note that, for $(i,j) = (1,1)$, the amount seeping in will be 0, so using $\nu_0 = 0, \mu'_{1_1} = 0$, the result trivially holds. Hence the result holds for all $i, j \geq 1$. □

Lemma 5.17. *For any arm $j \in \{1, \ldots, K\}$ and phase m, it holds that for any $\lambda > 0$,*

$$\mathbb{P}\Big(\sum_{t \in T_j(m)} (R_{t,j} - \mu_j) \geq \lambda \Big) \leq \exp\Big\{ - \frac{2\lambda^2}{n_m} \Big\}.$$

Proof. The result follows from Lemma 5.11. When applying this lemma, we use $n = T$, $m = n_m$, for $t = 0, 1, \ldots, T$ set $\mathcal{F}_t = \sigma(X_1, \ldots, X_t, R_{1,j}, \ldots, R_{t,j})$ and for $t = 1, 2, \ldots, T$ define $Z_t = R_{t,j} - \mu_j$ and $\epsilon_t = \mathbb{I}\{J_t = j, t \leq U_{m,j}\}$. Note that $T_j(m) = \{t \in \{1, \ldots, T\} : \epsilon_t = 1\}$ and hence $\sum_{t \in T_j(m)} (R_{t,j} - \mu_j) = \sum_{t=1}^{T} \epsilon_t (R_{t,j} - \mu_j)$. Further, $\sum_{t=1}^{T} \epsilon_t = |T_j(m)| \leq n_m$ with probability one.

Fix $1 \leq t \leq T$. We now argue that ϵ_t is $\mathcal{F}_{t-1}$-measurable. First, notice that by the definition of ODAAF, the index M of the phase that t belongs to can be calculated based on the observations $X_1, \ldots, X_{t-1}$ up to time $t-1$. Since $t \leq U_{m,j}$ is equivalent to whether for this phase index M, the inequality $M \leq m$ holds, it follows that $\{t \leq U_{m,j}\}$ is $\mathcal{F}_{t-1}$-measurable. The same holds for $\{J_t = j\}$ for the same reason. Hence, it follows that ϵ_t is indeed $\mathcal{F}_{t-1}$-measurable.

Now, Z_t is $\mathcal{F}_t$-measurable as $R_{t,j}$ is clearly $\mathcal{F}_t$-measurable. Furthermore, by our assumptions on $(R_{t,j})_{t,j}$ and $(X_t)_t$, $\mathbb{E}[R_{t,j} | \mathcal{F}_{t-1}] = \mu_j$ also holds, implying that Z_t also satisfies the conditions of the lemma with $a = -\mu_j$ and $c = 1$. Thus, the result follows by applying Lemma 5.11. □

We now bound each term of the decomposition in (5.13) in turn.

Bounding Term I.: For Term I., we use Lemma 5.17 to get that with probability greater than $1 - \frac{1}{T\tilde{\Delta}_m^2}$,

$$\sum_{i=1}^{m} \sum_{t=S_{i,j}}^{U_{i,j}} (R_{t,J_t} - \mu_j) \leq \sqrt{\frac{n_m \log(T\tilde{\Delta}_m^2)}{2}}.$$

Bounding Term II.: For Term II., we will use Freedmans inequality (Theorem 5.10). From Lemma 5.12, $\{Y_s\}_{s=0}^{\infty}$ with $Y_s = \sum_{t=1}^{s}(Q_t - \mathbb{E}[Q_t|\mathcal{G}_{t-1}])$ is a martingale with respect to $\{\mathcal{G}_s\}_{s=0}^{\infty}$ with increments $\{Z_s\}_{s=0}^{\infty}$ satisfying $\mathbb{E}[Z_s|\mathcal{G}_{s-1}] = 0$ and $Z_s \leq 1$ for all s. Further, by Lemma 5.13, $\sum_{t=1}^{s} \mathbb{E}[Z_t^2|\mathcal{G}_{t-1}] \leq 2m\mathbb{E}[\tau] \leq \frac{6m \times 2^m \mathbb{E}[\tau]}{12} \leq n_m/12$ with probability 1. Hence we can apply Freedman's inequality to get that with probability greater than $1 - \frac{1}{T\tilde{\Delta}_m^2}$,

$$\sum_{t=1}^{S_{m,j}} (Q_t - \mathbb{E}[Q_t|\mathcal{G}_{t-1}]) \leq \frac{2}{3}\log(T\tilde{\Delta}_m^2) + \sqrt{\frac{1}{12} n_m \log(T\tilde{\Delta}_m^2)}.$$

Bounding Term III.: For Term III., we again use Freedman's inequality (Theorem 5.10) but using Lemma 5.14 to show that $\{Y'_s\}_{s=0}^{\infty}$ with $Y'_s = \sum_{t=1}^{s}(\mathbb{E}[P_t|\mathcal{G}_{t-1}] - P_t)$ is a martingale with respect to $\{\mathcal{G}_s\}_{s=0}^{\infty}$ with increments $\{Z'_s\}_{s=0}^{\infty}$ satisfying $\mathbb{E}[Z'_s|\mathcal{G}_{s-1}] = 0$ and $Z'_s \leq 1$ for all s. Further, by Lemma 5.15, $\sum_{t=1}^{s} \mathbb{E}[Z_t^2|\mathcal{G}_{t-1}] \leq m\mathbb{E}[\tau] \leq n_m/12$ with probability 1. Hence, with probability greater than $1 - \frac{1}{T\tilde{\Delta}_m^2}$,

$$\sum_{t=1}^{U_{m,j}} (\mathbb{E}[P_t|\mathcal{G}_{t-1}] - P_t) \leq \frac{2}{3}\log(T\tilde{\Delta}_m) + \sqrt{\frac{1}{12} n_m \log(T\tilde{\Delta}_m^2)}.$$

Bounding Term IV.: We bound term IV. using Lemma 5.16,

$$\begin{aligned}
&\sum_{t=1}^{S_{m,j}} \mathbb{E}[Q_t|\mathcal{G}_{t-1}] - \sum_{t=1}^{U_{m,j}} \mathbb{E}[P_t|\mathcal{G}_{t-1}] \\
&= \sum_{t=1}^{S_{m,j}} \mathbb{E}\Bigg[\sum_{i=1}^{m} (A_{i,t}\mathbb{I}\{S_{i-1,j} \le t \le S_i - 1\} + B_{i,t}\mathbb{I}\{S_i \le t \le S_{i,j} - 1\}) \Bigg| \mathcal{G}_{t-1} \Bigg] \\
&\qquad - \sum_{t=1}^{U_{m,j}} \mathbb{E}\Bigg[\sum_{i=1}^{m} C_{i,t}\mathbb{I}\{S_{i,j} \le t \le U_{i,j}\} \Bigg| \mathcal{G}_{t-1} \Bigg] \\
&= \sum_{i=1}^{m} \sum_{t=1}^{S_{m,j}} \mathbb{E}[A_{i,t}\mathbb{I}\{S_{i-1,j} \le t \le S_i - 1\}|\mathcal{G}_{t-1}] + \sum_{i=1}^{m} \sum_{t=1}^{S_{m,j}} \mathbb{E}[B_{i,t}\mathbb{I}\{S_i \le t \le S_{i,j} - 1\}|\mathcal{G}_{t-1}] \\
&\qquad - \sum_{i=1}^{m} \sum_{t=1}^{U_{m,j}} \mathbb{E}[C_{i,t}\mathbb{I}\{S_{i,j} \le t \le U_{i,j}\}|\mathcal{G}_{t-1}] \\
&= \sum_{i=1}^{m} \Bigg(\sum_{t=S_{i-1,j}}^{S_i - 1} \mathbb{E}[A_{i,t}|\mathcal{G}_{t-1}] + \sum_{t=S_i}^{S_{i,j}-1} \mathbb{E}[B_{i,t}|\mathcal{G}_{t-1}] - \sum_{t=S_{i,j}}^{U_{i,j}} \mathbb{E}[C_{i,t}|\mathcal{G}_{t-1}] \Bigg) \\
&\le \sum_{i=1}^{m} \Bigg(2\mathbb{E}[\tau] + \mu_{j_i'} \sum_{l=0}^{\nu_i} \mathbb{P}(\tau > l) - \mu_j \sum_{l=0}^{\nu_i} \mathbb{P}(\tau > l) \Bigg) \le 3m\mathbb{E}[\tau].
\end{aligned}$$

since $R_{t,j} \in [0,1]$.

Combining all terms: To get the final high probability bound, we sum the bounds for each term I.-IV.. Then, with probability greater than $1 - \frac{3}{T\tilde{\Delta}_m^2}$, either $j \notin \mathcal{A}_m$ or arm j is played n_m times by the end of phase m and

$$\begin{aligned}
\frac{1}{n_m} \sum_{t \in T_j(m)} (X_t - \mu_j) &\le \frac{4\log(T\tilde{\Delta}_m^2)}{3n_m} + \left(\frac{2}{\sqrt{12}} + \frac{1}{\sqrt{2}} \right) \sqrt{\frac{\log(T\tilde{\Delta}_m^2)}{n_m}} + \frac{3m\mathbb{E}[\tau]}{n_m} \\
&\le \frac{4\log(T\tilde{\Delta}_m^2)}{3n_m} + \sqrt{\frac{2\log(T\tilde{\Delta}_m^2)}{n_m}} + \frac{3m\mathbb{E}[\tau]}{n_m} = w_m.
\end{aligned}$$

Defining $\mathbf{n_m}$: Setting

$$n_m = \left\lceil \frac{1}{\tilde{\Delta}_m^2} \left(\sqrt{2\log(T\tilde{\Delta}_m^2)} + \sqrt{2\log(T\tilde{\Delta}_m^2) + \frac{8}{3}\tilde{\Delta}_m \log(T\tilde{\Delta}_m^2) + 6\tilde{\Delta}_m m\mathbb{E}[\tau]} \right)^2 \right\rceil. \tag{5.14}$$

ensures that $w_m \leq \frac{\tilde{\Delta}_m}{2}$ which concludes the proof. □

5.B.2 Regret Bounds

Here we prove the regret bound in Theorem 5.2 under Assumption 1 and the choice of n_m given by (5.14). Under Assumption 1, the bridge period is not necessary so the results here hold for the version of Algorithm 5.1 with the bridge period omitted. Note that if we were to include the bridge period, we would be playing each arm at most $2n_m$ times by the end of phase m so our regret would simply increase by a factor of 2.

Theorem 5.2. *Under Assumption 1, the expected regret of Algorithm 5.1 is upper bounded as*

$$\mathbb{E}[\mathfrak{R}_T] \leq \sum_{\substack{j=1 \\ j \neq j^*}}^{K} O\left(\frac{\log(T\Delta_j^2)}{\Delta_j} + \log(1/\Delta_j)\mathbb{E}[\tau] \right). \tag{5.5}$$

Proof. Our proof is a restructuring of the proof of Theorem 3.1 in (Auer and Ortner, 2010). For any arm j, define M_j to be the random variable representing the phase when arm j is eliminated in. We set $M_j = \infty$ if the arm did not get eliminated before time step T. Note that if M_j is finite, $j \in \mathcal{A}_{M_j}$ (this also means that $\mathcal{A}_{M_j}$ is well-defined) and if $\mathcal{A}_{M_j+1}$ is also defined (M_j is not the last phase) then $j \notin \mathcal{A}_{M_j+1}$. We also let m_j denote the phase arm j *should* be eliminated in, that is $m_j = \min\{\, m \geq$

$1 : \tilde{\Delta}_m < \frac{\Delta_j}{2}\}$. From the definition of $\tilde{\Delta}_m$ in our algorithm, we get the relations

$$2^{m_j} = \frac{1}{\tilde{\Delta}_{m_j}} \leq \frac{4}{\Delta_j} < \frac{1}{\tilde{\Delta}_{m_j+1}} \quad \text{and} \quad \frac{\Delta_j}{4} \leq \tilde{\Delta}_{m_j} \leq \frac{\Delta_j}{2}\,. \tag{5.15}$$

Define $N_j = \sum_{t=1}^{T} \mathbb{I}\{J_t = j\}$ be the number of times arm j is used and let $\mathfrak{R}_T^{(j)} = N_j \Delta_j$ be the "pseudo"-regret contribution from each arm $1 \leq j \leq K$ so that $\mathbb{E}[\mathfrak{R}_T] = \mathbb{E}\left[\sum_{j=1}^{K} \mathfrak{R}_T^{(j)}\right]$. Let M^* be the round when the optimal arm j^* is eliminated. Hence,

$$\mathbb{E}[\mathfrak{R}_T] = \mathbb{E}\left[\sum_{j=1}^{K} \mathfrak{R}_T^{(j)}\right] = \underbrace{\mathbb{E}\left[\sum_{j=1}^{K} \mathfrak{R}_T^{(j)} \mathbb{I}\left\{M^* \geq m_j\right\}\right]}_{\text{Term I.}} + \underbrace{\mathbb{E}\left[\sum_{j=1}^{K} \mathfrak{R}_T^{(j)} \mathbb{I}\left\{M^* < m_j\right\}\right]}_{\text{Term II.}}\,.$$

We will bound the regret in each of these cases in turn. To do so, we need the following results which consider the probabilities of confidence bounds failing and arms being eliminated in the incorrect rounds.

Lemma 5.18. *For any suboptimal arm j,*

$$\mathbb{P}(M_j > m_j \text{ and } M^* \geq m_j) \leq \frac{6}{T\tilde{\Delta}_{m_j}^2}\,.$$

Proof. Define

$$E = \{\bar{X}_{m_j,j} \leq \mu_j + w_{m_j}\} \quad \text{and} \quad H = \{\bar{X}_{m_j,j^*} > \mu^* - w_{m_j}\}\,.$$

If both E and H occur, it follows that,

$$\begin{aligned}
\bar{X}_{m_j,j} &\leq \mu_j + w_{m_j} \\
&= \mu_j^* - \Delta_j + w_{m_j} && \text{(since } \Delta_j = \mu_{j^*} - \mu_j\text{)} \\
&\leq \bar{X}_{m_j,j^*} + w_{m_j} - \Delta_j + w_{m_j} \\
&< \bar{X}_{m_j,j^*} - 2\tilde{\Delta}_{m_j} + 2w_{m_j} && \text{(by (5.15))} \\
&\leq \bar{X}_{m_j,j^*} - \tilde{\Delta}_{m_j} && \text{(since } n_m \text{ is such that } w_m \leq \tilde{\Delta}_m/2\text{)}
\end{aligned}$$

and arm j would be eliminated. Hence, on the event $M^* \geq m_j$, $M_j \leq m_j$. Thus, $M^* \geq m_j$ and $M_j > m_j$ imply that either E or H does not occur and so $\mathbb{P}(M_j > m_j \text{ and } M^* \geq m_j) \leq \mathbb{P}(\{E^c \cup H^c\} \cap \{j, j^* \in \mathcal{A}_{m_j}\}) \leq \mathbb{P}(E^c \cap j \in \mathcal{A}_{m_j}) + \mathbb{P}(H^c \cap j^* \in \mathcal{A}_{m_j})$. Using Lemma 5.1, we then get that,

$$\mathbb{P}(M_j \geq m_j \text{ and } M^* \geq m_j) \leq \frac{6}{T\tilde{\Delta}_{m_j}^2}.$$

□

Note that the random set $\mathcal{A}_m$ may not be defined for certain $\omega \in \Omega$. That is, $\mathcal{A}_m$ is a partially defined random element. For convenience, we modify the definition of $\mathcal{A}_m$ so that it is an emptyset for any ω when it is not defined by the previous definition. Define the event $F_j(m) = \{\bar{X}_{m,j^*} < \bar{X}_{m,j} - \tilde{\Delta}_m\} \cap \{j, j^* \in \mathcal{A}_m\}$ to be the event that arm j^* is eliminated by arm j in phase m (given our note on $\mathcal{A}_m$, this is well-defined). The probability of this occurring is bounded in the following lemma.

Lemma 5.19. *The probability that the optimal arm j^* is eliminated in round $m < \infty$ by the suboptimal arm j is bounded by*

$$\mathbb{P}(F_j(m)) \leq \frac{6}{T\tilde{\Delta}_m^2}.$$

Proof. First note that for a suboptimal arm j to eliminate arm j^* in round m, both j and j^* must be active in round m and $\bar{X}_{m,j} - w_m > \bar{X}_{m,j^*} + w_m$. Hence,

$$\mathbb{P}(F_j(m)) = \mathbb{P}(j, j^* \in \mathcal{A}_m \text{ and } \bar{X}_{m,j} - w_m > \bar{X}_{m,j^*} + w_m)$$

Then, observe that if

$$E = \{\bar{X}_{m,j} \leq \mu_j + w_m\} \quad \text{and} \quad H = \{\bar{X}_{m,j^*} > \mu^* - w_m\}$$

both hold in round m, it follows that,

$$\bar{X}_{m,j} - \tilde{\Delta}_m \leq \mu_j + w_m - \tilde{\Delta}_m \leq \mu_j - \frac{\tilde{\Delta}_m}{2} \leq \mu_{j^*} - \frac{\tilde{\Delta}_m}{2} \leq \bar{X}_{m,j^*} + w_m - \frac{\tilde{\Delta}_m}{2} \leq \bar{X}_{m,j^*}$$

so arm j^* will not be eliminated by arm j in round m. Hence, for arm j^* to be eliminated by arm j in round m, one of E or H must not occur and the probability of this is bounded by Lemma 5.1 as,

$$\begin{aligned}\mathbb{P}(F_j(m)) &\leq \mathbb{P}((E^C \cup H^C) \cap (j, j^* \in \mathcal{A}_m)) \leq \mathbb{P}(E^C \cap (j \in \mathcal{A}_m)) + \mathbb{P}(H^C \cap (j^* \in \mathcal{A}_m)) \\ &\leq \frac{6}{T\tilde{\Delta}_m^2}.\end{aligned}$$

□

We now return to bounding the expected regret in each of the two cases.

Bounding Term I. To bound the first term, we consider the cases where arm j is eliminated in or before the correct round ($M_j \leq m_j$) and where arm j is eliminated

late ($M_j > m_j$). Then, by Lemma 5.18,

$$\begin{aligned}
&\mathbb{E}\Bigg[\sum_{j=1}^{K} \mathfrak{R}_T^{(j)} \mathbb{I}\{M^* \geq m_j\}\Bigg] \\
&\quad = \mathbb{E}\Bigg[\sum_{j=1}^{K} \mathfrak{R}_T^{(j)} \mathbb{I}\{M^* \geq m_j\}\mathbb{I}\{M_j \leq m_j\}\Bigg] + \mathbb{E}\Bigg[\sum_{j=1}^{K} \mathfrak{R}_T^{(j)} \mathbb{I}\{M^* \geq m_j\}\mathbb{I}\{M_j > m_j\}\Bigg] \\
&\quad \leq \sum_{j=1}^{K} \mathbb{E}[\mathfrak{R}_T^{(j)} \mathbb{I}\{M_j \leq m_j\}] + \sum_{j=1}^{K} \mathbb{E}[T\Delta_j \mathbb{I}\{M^* \geq m_j, M_j > m_j\}] \\
&\quad \leq \sum_{j=1}^{K} \Delta_j n_{m_j} + \sum_{j=1}^{K} T\Delta_j \mathbb{P}(M_j > m_j \text{ and } M^* \geq m_j) \\
&\quad \leq \sum_{j=1}^{K} \Delta_j n_{m_j} + \sum_{j=1}^{K} T\Delta_j \frac{6}{T\tilde{\Delta}_{m_j}^2} \\
&\quad \leq \sum_{j=1}^{K} \left(\Delta_j n_{m_j} + \frac{24}{\tilde{\Delta}_{m_j}}\right) \leq \sum_{j=1}^{K} \left(\frac{96}{\Delta_j} + \Delta_j n_{m_j}\right).
\end{aligned}$$

Bounding Term II For the second term, let $m_{\max} = \max_{j \neq j^*} m_j$. and recall that N_j is the total number of times arm j is played. Then,

$$\begin{aligned}
\mathbb{E}\Bigg[\sum_{j=1}^{K} \mathfrak{R}_T^{(j)} \mathbb{I}\left\{M^* < m_j\right\}\Bigg] &= \mathbb{E}\Bigg[\sum_{m=1}^{m_{\max}} \sum_{j:m<m_j} \mathfrak{R}_T^{(j)} \mathbb{I}\{M^* = m\}\Bigg] \\
&= \sum_{m=1}^{m_{\max}} \mathbb{E}\Bigg[\mathbb{I}\{M^* = m\} \sum_{j:m_j>m} \mathfrak{R}_T^{(j)}\Bigg] \\
&= \sum_{m=1}^{m_{\max}} \mathbb{E}\Bigg[\mathbb{I}\{M^* = m\} \sum_{j:m_j>m} N_j\Delta_j\Bigg] \\
&\leq \sum_{m=1}^{m_{\max}} \mathbb{E}\Bigg[\mathbb{I}\{M^* = m\} T \max_{j:m_j>m} \Delta_j\Bigg] \\
&\leq \sum_{m=1}^{m_{\max}} 4\mathbb{P}(M^* = m) T\tilde{\Delta}_m\,.
\end{aligned}$$

Now consider the probability that arm j^* is eliminated in round m. This includes the probability that it is eliminated by any suboptimal arm. For arm j^* to be eliminated in round m by a suboptimal arm with $m_j < m$, arm j must be active ($M_j > m_j$)

and the optimal arm must also have been active in round m_j ($M^* \geq m_j$). Using this, it follows that

$$\begin{aligned}\mathbb{P}(M^* = m) &\leq \sum_{j=1}^{K} \mathbb{P}(F_j(m)) = \sum_{j:m_j<m} \mathbb{P}(F_j(m)) + \sum_{j:m_j \geq m} \mathbb{P}(F_j(m)) \\ &\leq \sum_{j:m_j<m} \mathbb{P}(M_j > m_j \text{ and } M^* \geq m_j) + \sum_{j:m_j \geq m} \mathbb{P}(F_j(m)).\end{aligned}$$

Then, using Lemmas 5.18 and 5.19 and summing over all $m \leq M$ gives,

$$\begin{aligned}&\sum_{m=1}^{m_{\max}} \left(\sum_{j:m_j<m} 4\mathbb{P}(M_j > m_j \text{ and } M^* \geq m_j) T\tilde{\Delta}_m + \sum_{j:m_j \geq m} 4\mathbb{P}(F_j(m)) T\tilde{\Delta}_m \right) \\ &\leq \sum_{m=1}^{m_{\max}} \left(\sum_{j:m_j<m} 4\frac{6}{T\tilde{\Delta}_{m_j}^2} T\frac{\tilde{\Delta}_{m_j}}{2^{m-m_j}} + \sum_{j:m_j \geq m} \frac{24}{T\tilde{\Delta}_m^2} T\tilde{\Delta}_m \right) \\ &\leq \sum_{j=1}^{K} \frac{24}{\tilde{\Delta}_{m_j}} \sum_{m=m_j}^{m_{\max}} 2^{-(m-m_j)} + \sum_{j=1}^{K} \sum_{m=1}^{m_j} \frac{24}{2^{-m}} \\ &\leq \sum_{j=1}^{K} \frac{96 \cdot 2}{\Delta_j} + \sum_{j=1}^{K} 24 \cdot 2^{m_j+1} \\ &\leq \sum_{j=1}^{K} \frac{192}{\Delta_j} + \sum_{j=1}^{K} 48 \cdot \frac{4}{\Delta_j} = \sum_{j=1}^{K} \frac{384}{\Delta_j}.\end{aligned}$$

Combining the regret from terms I and II gives,

$$\mathbb{E}[\mathfrak{R}_T] \leq \sum_{j=1}^{K} \left(\frac{480}{\Delta_j} + \Delta_j n_{m_j} \right).$$

Hence, all that remains is to bound n_m in terms of Δ_j, T and d,

$$\begin{aligned}
n_{m_j} &= \left\lceil \frac{1}{\tilde{\Delta}^2_{m_j}} \left(\sqrt{2\log(T\tilde{\Delta}^2_{m_j})} + \sqrt{2\log(T\tilde{\Delta}^2_{m_j}) + \frac{8}{3}\tilde{\Delta}_{m_j}\log(T\tilde{\Delta}^2_{m_j}) + 6\tilde{\Delta}_{m_j} m_j \mathbb{E}[\tau]} \right)^2 \right\rceil \\
&\leq \left\lceil \frac{1}{\tilde{\Delta}^2_{m_j}} \left(8\log(T\tilde{\Delta}^2_{m_j}) + \frac{16}{3}\tilde{\Delta}_{m_j}\log(T\tilde{\Delta}^2_{m_j}) + 12\tilde{\Delta}_{m_j} m_j \mathbb{E}[\tau] \right) \right\rceil \\
&\leq 1 + \frac{8\log(T\Delta_j^2/4)}{\tilde{\Delta}^2_{m_j}} + \frac{16\log(T\Delta_j^2/4)}{3\tilde{\Delta}_{m_j}} + \frac{12\log_2(4/\Delta_j)\mathbb{E}[\tau]}{\tilde{\Delta}_{m_j}} \\
&\leq 1 + \frac{128\log(T\Delta_j^2)}{\Delta_j^2} + \frac{32\log(T\Delta_j^2)}{3\Delta_j} + \frac{96\log(4/\Delta_j)\mathbb{E}[\tau]}{\Delta_j},
\end{aligned}$$

where we have used $(a+b)^2 \leq 2(a^2+b^2)$ for $a, b \geq 0$ and $\log_2(x) \leq 2\log(x)$ for $x > 0$.

Hence, the total expected regret from ODAAF with bounded delays can be bounded by,

$$\mathbb{E}[\mathfrak{R}_t] \leq \sum_{j=1:j\neq j^*}^{K} \left(\frac{128\log(T\Delta_j^2)}{\Delta_j} + \frac{32}{3}\log(T\Delta_j^2) + 96\log(4/\Delta_j)\mathbb{E}[\tau] + \frac{480}{\Delta_j} + \Delta_j \right). \tag{5.16}$$

□

We now prove the problem independent regret bound,

Corollary 5.3. *For any problem instance satisfying Assumption 1, the expected regret of Algorithm 5.1 satisfies*

$$\mathbb{E}[\mathfrak{R}_T] \leq O(\sqrt{KT\log(K)} + K\mathbb{E}[\tau]\log(T)).$$

Proof. Let

$$\lambda = \sqrt{\frac{K\log(K)e^2}{T}}$$

and note that for $\Delta > \lambda$, $\log(T\Delta^2)/\Delta$ is a decreasing function of Δ. Then, for some

constants C_1, C_2, and using the previous theorem, we can bound the regret by,

$$\mathbb{E}[\mathfrak{R}_T] \leq \sum_{j:\Delta_j \leq \lambda} \mathbb{E}[\mathfrak{R}_t^{(j)}] + \sum_{j:\Delta_j > \lambda} \mathbb{E}[\mathfrak{R}_T^{(j)}] \leq \frac{KC_1 \log(T\lambda^2)}{\lambda} + KdC_2 \log(1/\lambda) + T\lambda.$$

Then, substituting the above value of λ gives a worst case regret bound that scales with $O(\sqrt{KT\log(K)} + K\mathbb{E}[\tau]\log(T))$. □

5.C Results for Delays with Bounded Support

5.C.1 High Probability Bounds

Lemma 5.5. *Under Assumptions 1 of known expected delay and 2 of bounded delays, and choice of n_m given in (5.6), the estimates $\bar{X}_{m,j}$ obtained by Algorithm 5.1 satisfy the following: For any arm j and phase m, with probability at least $1 - \frac{12}{T\tilde{\Delta}_m^2}$, either $j \notin \mathcal{A}_m$ or*

$$\bar{X}_{m,j} - \mu_j \leq \tilde{\Delta}_m/2.$$

Proof. Let

$$w_m = \frac{4\log(T\tilde{\Delta}_m^2)}{3n_m} + \sqrt{\frac{2\log(T\tilde{\Delta}_m^2)}{n_m}} + \frac{2\mathbb{E}[\tau]}{n_m}. \tag{5.17}$$

We show that with probability greater than $1 - \frac{12}{T\tilde{\Delta}_m^2}$, either $j \notin \mathcal{A}_m$ or $\frac{1}{n_m}\sum_{t\in T_j(m)}(X_t - \mu_j) \leq w_m$. For now, assume that $n_m \geq md$.

For arm j and phase m, assume $j \in \mathcal{A}_m$ and define p_i to be the probability of the confidence bounds on arm j failing at the end of each phase $i \leq m$, ie. $p_i \doteq \mathbb{P}(\sum_{t\in T_j(i)}(X_t - \mu_j) \geq n_i w_i)$ with $p_0 = 0$. Again, let $B_{i,t} = R_t \mathbb{I}\{\tau_{t,J_t} + t \geq S_{i,j}\}$ and $C_{i,t} = R_t \mathbb{I}\{\tau_{t,J_t} + t > U_{i,j}\}$ (note that we don't need to consider $A_{i,t}$ since $\nu_i = n_i - n_{i-1} \geq d$ so all reward entering $[S_{i,j}, U_{i,j}]$ will be from the last $\nu_i \geq d$ plays) and for any event H, let $\mathbb{I}_i\{H\} := \mathbb{I}\{H \cap \{j \in \mathcal{A}_i\}\}$. Recall the filtration $\{\mathcal{G}_t\}_{t=0}^{\infty}$ from (5.12)

where $\mathcal{G}_t = \sigma(X_1, \ldots, X_t, J_1, \ldots, J_t, \tau_{1,J_1}, \ldots, \tau_{t,J_t}, R_{1,J_1}, \ldots, R_{t,J_t})$ and $\mathcal{G}_0 = \{\emptyset, \Omega\}$. Now, defining,

$$Q_t = \sum_{i=1}^{m} B_{i,t} \mathbb{I}\{S_{i,j} - d - 1 \leq t \leq S_{i,j} - 1\}),$$
$$P_t = \sum_{i=1}^{m} C_{i,t} \mathbb{I}\{S_{i,j} \leq t \leq U_{i,j}\},$$

we use the decomposition

$$\begin{aligned}
\sum_{t \in T_j(m)} (X_t - \mu_j) &= \sum_{i=1}^{m} \sum_{t=S_{i,j}}^{U_{i,j}} (X_t - \mu_j) \\
&\leq \sum_{i=1}^{m} \Bigg(\sum_{t=S_{i-1,j}}^{S_{i,j}-1} R_{t,J_t} \mathbb{I}_i\{\tau_{t,J_t} + t \geq S_{i,j}\} + \sum_{t=S_{i,j}}^{U_{i,j}} (R_{t,J_t} - \mu_j) \\
&\qquad - \sum_{t=S_{i,j}}^{U_{i,j}} R_{t,J_t} \mathbb{I}_i\{\tau_{t,J_t} + t > U_{i,j}\} \Bigg) \\
&\leq \sum_{i=1}^{m} \Bigg(\sum_{t=S_{i,j}-d}^{S_{i,j}-1} B_{i,t} + \sum_{t=S_{i,j}}^{U_{i,j}} (R_t - \mu_j) - \sum_{t=S_{i,j}}^{U_{i,j}} C_{i,t} \Bigg) \\
&= \sum_{i=1}^{m} \sum_{t=S_{i,j}}^{U_{i,j}} (R_{t,J_t} - \mu_j) + \sum_{t=1}^{S_{m,j}} Q_t - \sum_{t=1}^{U_{m,j}} P_t \\
&= \underbrace{\sum_{i=1}^{m} \sum_{t=S_{i,j}}^{U_{i,j}} (R_{t,J_t} - \mu_j)}_{\text{Term I.}} + \underbrace{\sum_{t=1}^{S_{m,j}} (Q_t - \mathbb{E}[Q_t|\mathcal{G}_{t-1}])}_{\text{Term II.}} + \underbrace{\sum_{t=1}^{U_{m,j}} (\mathbb{E}[P_t|\mathcal{G}_{t-1}] - P_t)}_{\text{Term III.}} \\
&\qquad + \underbrace{\sum_{t=1}^{S_{m,j}} \mathbb{E}[Q_t|\mathcal{G}_{t-1}] - \sum_{t=1}^{U_{m,j}} \mathbb{E}[P_t|\mathcal{G}_{t-1}]}_{\text{Term IV.}}.
\end{aligned}$$

Outline of proof Again, the proof continues by bounding each term of this decomposition in turn. Note that we do not have the $A_{i,t}$ terms in this decomposition since there will be no reward from phase $i-1$ (before the bridge period) received in $[S_{i,j}, U_{i,j}]$. We bound each of these terms with high probability. For terms I. and III., this is the same as in the general case (see the proof of Lemma 5.1, Section 5.B).

For term II. we need the following results to show that $Z_t = Q_t - \mathbb{E}[Q_s|\mathcal{G}_{t-1}]$ is a martingale difference (Lemma 5.20) and to bound its variance (Lemma 5.21) before we can apply Freedman's inequality. The bound for term IV. is also different due to the bridge period and boundedness of the delay. After bounding each term, we collect them together and recursively calculate the probability with which the bounds hold.

Lemma 5.20. *Let* $Y_s = \sum_{t=1}^{s}(Q_t - \mathbb{E}[Q_t|\mathcal{G}_{t-1}])$ *for all* $s \geq 1$, *and* $Y_0 = 0$. *Then* $\{Y_s\}_{s=0}^{\infty}$ *is a martingale with respect to the filtration* $\{\mathcal{G}_s\}_{s=0}^{\infty}$ *with increments* $Z_s = Y_s - Y_{s-1} = Q_s - \mathbb{E}[Q_s|\mathcal{G}_{s-1}]$ *satisfying* $\mathbb{E}[Z_s|\mathcal{G}_{s-1}] = 0, |Z_s| \leq 1$ *for all* $s \geq 1$.

Proof. To show $\{Y_s\}_{s=0}^{\infty}$ is a martingale we need to show that Y_s is $\mathcal{G}_s$-measurable for all s and $\mathbb{E}[Y_s|\mathcal{G}_{s-1}] = Y_{s-1}$.

Measurability: We show that $B_{i,s}\mathbb{I}\{S_{i,j} - d - 1 \leq s \leq S_{i,j} - 1\}$ is $\mathcal{G}_s$-measurable. This then suffices to show that Y_s is $\mathcal{G}_s$-measurable since the filtration $\mathcal{G}_s$ is non-decreasing in s.

First note that by definition of $\mathcal{G}_s$, τ_{t,J_t}, R_{t,J_t} are all $\mathcal{G}_s$-measurable for $t \leq s$. Hence, it is sufficient to show that $\mathbb{I}\{\tau_{s,J_s} + s \geq S_{i,j}, S_{i,j} - d - 1 \leq s \leq S_{i,j} - 1\}$ is $\mathcal{G}_s$-measurable since the product of measurable functions is measurable. For any $s' \in \mathbb{N} \cup \{\infty\}$, $\{S_{i,j} = s', s' - d - 1 \leq s\} \in \mathcal{G}_s$ for $s \geq S_i - \nu_{i-1}$ and so the union $\bigcup_{s' \in \mathbb{N}\cup\{\infty\}}\{\tau_{s,J_s}+s \geq s', s'-d-1 \leq s \leq s'-1, S_{i,j} = s'\} = \{\tau_{s,J_s}+s \geq S_{i,j}, S_{i,j}-d-1 \leq s \leq S_{i,j} - 1\}$ is an element of $\mathcal{G}_s$.

Increments: Hence, $\{Y_s\}_{s=0}^{\infty}$ is a martingale with respect to the filtration $\{\mathcal{G}_s\}_{s=0}^{\infty}$ if the increments conditional on the past are zero. For any $s \geq 1$, we have that

$$Z_s = Y_s - Y_{s-1} = \sum_{t=1}^{s}(Q_t - \mathbb{E}[Q_t|\mathcal{G}_{t-1}]) - \sum_{t=1}^{s-1}(Q_t - \mathbb{E}[Q_t|\mathcal{G}_{t-1}]) = Q_s - \mathbb{E}[Q_s|\mathcal{G}_{s-1}].$$

Then,

$$\mathbb{E}[Z_s|\mathcal{G}_{s-1}] = \mathbb{E}[Q_s - \mathbb{E}[Q_s|\mathcal{G}_{s-1}]|\mathcal{G}_{s-1}] = \mathbb{E}[Q_s|\mathcal{G}_{s-1}] - \mathbb{E}[Q_s|\mathcal{G}_{s-1}] = 0$$

and so $\{Y_s\}_{s=0}^{\infty}$ is a martingale.

Lastly, since for any t and $\omega \in \Omega$, there is at most one i where $\mathbb{I}\{S_{i,j} - d \leq t \leq S_{i,j} - 1\}(\omega) = 1$, and by definition of R_{t,J_t}, $B_{i,t} \leq 1$, it follows that $|Z_s| = |Q_s - \mathbb{E}[Q_s|\mathcal{G}_{s-1}]| \leq 1$ for all s. □

Lemma 5.21. *For any $t \geq 1$, let $Z_t = Q_t - \mathbb{E}[Q_t|\mathcal{G}_{t-1}]$, then*

$$\sum_{t=1}^{S_{m,j}-1} \mathbb{E}[Z_t^2|\mathcal{G}_{t-1}] \leq m\mathbb{E}[\tau].$$

Proof. Let us denote $S' \doteq S_{m,j} - 1$. Observe that

$$\begin{aligned}\sum_{t=1}^{S'} \mathbb{E}[Z_t^2|\mathcal{G}_{t-1}] &= \sum_{t=1}^{S'} \mathbb{V}(Q_t|\mathcal{G}_{t-1}) \leq \sum_{t=1}^{S'} \mathbb{E}[Q_t^2|\mathcal{G}_{t-1}] \\ &= \sum_{t=1}^{S'} \mathbb{E}\Big[\Big(\sum_{i=1}^{m}(B_{i,t}\mathbb{I}\{S_{i,j} - d \leq t \leq S_{i,j} - 1\})\Big)^2\Big|\mathcal{G}_{t-1}\Big].\end{aligned}$$

Then for all $i = 1, \ldots, m$, all indicator terms $\mathbb{I}\{S_{i,j} - d \leq t \leq S_{i,j} - 1\}$ are $\mathcal{G}_{t-1}$-measurable and only one can be non zero for any $\omega \in \Omega$. Hence, for any $i, i' \leq m$, $i \neq i'$,

$$B_{i,t} \times \mathbb{I}\{S_{i,j} - d - 1 \leq t \leq S_{i,j} - 1\} \times B_{i',t} \times \mathbb{I}\{S_{i',j} - d - 1 \leq t \leq S_{i',j} - 1\} = 0,$$

Using the above we see that

$$\begin{aligned}
\sum_{t=1}^{S'} \mathbb{E}[Z_t^2|\mathcal{G}_{t-1}] &\leq \sum_{t=1}^{S'} \mathbb{E}\left[\left(B_{i,t}\mathbb{I}\{S_{i,j}-d-1 \leq t \leq S_{i,j}-1\}\right)^2 \Big| \mathcal{G}_{t-1}\right] \\
&= \sum_{t=1}^{S'} \mathbb{E}\left[\sum_{i=1}^{m} B_{i,t}^2 \mathbb{I}\{S_{i,j}-d-1 \leq t \leq S_{i,j}-1\}^2 \Big| \mathcal{G}_{t-1}\right] \\
&= \sum_{i=1}^{m}\sum_{t=1}^{S'} \mathbb{E}[B_{i,t}^2 \mathbb{I}\{S_{i,j}-d-1 \leq t \leq S_{i,j}-1\}|\mathcal{G}_{t-1}] \\
&\qquad\qquad \text{(using that the indicator is } \mathcal{G}_{t-1}\text{-measurable)} \\
&\leq \sum_{i=1}^{m} \sum_{t=S_{i,j}-d-1}^{S_{i,j}-1} \mathbb{E}[B_{i,t}^2|\mathcal{G}_{t-1}].
\end{aligned}$$

Then, for any $i \geq 1$,

$$\begin{aligned}
\sum_{t=S_{i,j}-d-1}^{S_{i,j}-1} \mathbb{E}[B_{i,t}^2|\mathcal{G}_{t-1}] &= \sum_{t=S_{i,j}-d-1}^{S_{i,j}-1} \mathbb{E}[R_{t,J_t}^2 \mathbb{I}\{\tau_{t,J_t}+t \geq S_{i,j}\}|\mathcal{G}_{t-1}] \\
&\leq \sum_{t=S_{i,j}-d-1}^{S_{i,j}-1} \mathbb{E}[\mathbb{I}\{\tau_{t,J_t}+t \geq S_{i,j}\}|\mathcal{G}_{t-1}] \\
&= \sum_{s=0}^{\infty} \mathbb{I}\{S_{i,j}=s\} \sum_{t=s-d-1}^{s-1} \mathbb{E}[\mathbb{I}\{\tau_{t,J_t}+t \geq S_{i,j}\}|\mathcal{G}_{t-1}] \\
&= \sum_{s=0}^{\infty} \sum_{t=s-d-1}^{s-1} \mathbb{E}[\mathbb{I}\{S_{i,j}=s, \tau_{t,J_t}+t \geq S_{i,j}\}|\mathcal{G}_{t-1}]
\end{aligned}$$

(Since $S_{i,j} \geq S_i$ and so, due to the bridge period, $\{S_{i,j}=s\} \in \mathcal{G}_{t-1}$ for any $t \geq s-d$)

$$\begin{aligned}
&= \sum_{s=0}^{\infty} \sum_{t=s-d-1}^{s-1} \mathbb{E}[\mathbb{I}\{S_{i,j}=s, \tau_{t,J_t}+t \geq s\}|\mathcal{G}_{t-1}] \\
&= \sum_{s=0}^{\infty} \mathbb{I}\{S_{i,j}=s\} \sum_{t=s-d-1}^{s-1} \mathbb{P}(\tau_{t,J_t}+t \geq s) \\
&\qquad\qquad \text{(Since } \{S_{i,j}=s\} \in \mathcal{G}_{t-1} \text{ for any } t \geq s-d\text{)} \\
&\leq \sum_{s=0}^{\infty} \mathbb{I}\{S_{i,j}=s\} \sum_{l=0}^{\infty} \mathbb{P}(\tau > l) \\
&\leq \mathbb{E}[\tau].
\end{aligned}$$

Combining all terms gives the result. □

We now return to bounding each term of the decomposition

Bounding Term I.: For term II., as in Lemma 5.1, we can use Lemma 5.17 to get that with probability greater than $1 - \frac{1}{T\tilde{\Delta}_m^2}$,

$$\sum_{i=1}^{m} \sum_{t=S_{i,j}}^{U_{i,j}} (R_{t,J_t} - \mu_j) \leq \sqrt{\frac{n_m \log(T\tilde{\Delta}_m^2)}{2}}.$$

Bounding Term II.: For Term II., we will use Freedmans inequality (Theorem 5.10). From Lemma 5.20, $\{Y_s\}_{s=0}^{\infty}$ with $Y_s = \sum_{t=1}^{s}(Q_t - \mathbb{E}[Q_t|\mathcal{G}_{t-1}])$ is a martingale with respect to $\{\mathcal{G}_s\}_{s=0}^{\infty}$ with increments $\{Z_s\}_{s=0}^{\infty}$ satisfying $\mathbb{E}[Z_s|\mathcal{G}_{s-1}] = 0$ and $Z_s \leq 1$ for all s. Further, by Lemma 5.21, $\sum_{t=1}^{s}\mathbb{E}[Z_t^2|\mathcal{G}_{t-1}] \leq m\mathbb{E}[\tau] \leq \frac{4\times 2^m\mathbb{E}[\tau]}{8} \leq n_m/8$ with probability 1. Hence we can apply Freedman's inequality to get that with probability greater than $1 - \frac{1}{T\tilde{\Delta}_m^2}$,

$$\sum_{t=1}^{S_{m,j}}(Q_t - \mathbb{E}[Q_t|\mathcal{G}_{t-1}]) \leq \frac{2}{3}\log(T\tilde{\Delta}_m^2) + \sqrt{\frac{1}{8}n_m\log(T\tilde{\Delta}_m^2)}.$$

Bounding Term III.: For Term III., we again use Freedman's inequality (Theorem 5.10). As in Lemma 5.1, we use Lemma 5.14 to show that $\{Y'_s\}_{s=0}^{\infty}$ with $Y'_s = \sum_{t=1}^{s}(\mathbb{E}[P_t|\mathcal{G}_{t-1}] - P_t)$ is a martingale with respect to $\{\mathcal{G}_s\}_{s=0}^{\infty}$ with increments $\{Z'_s\}_{s=0}^{\infty}$ satisfying $\mathbb{E}[Z'_s|\mathcal{G}_{s-1}] = 0$ and $Z'_s \leq 1$ for all s. Further, by Lemma 5.15, $\sum_{t=1}^{s}\mathbb{E}[Z_t^2|\mathcal{G}_{t-1}] \leq m\mathbb{E}[\tau] \leq n_m/8$ with probability 1. Hence, with probability greater than $1 - \frac{1}{T\tilde{\Delta}_m^2}$,

$$\sum_{t=1}^{U_{m,j}}(\mathbb{E}[P_t|\mathcal{G}_{t-1}] - P_t) \leq \frac{2}{3}\log(T\tilde{\Delta}_m^2) + \sqrt{\frac{1}{8}n_m\log(T\tilde{\Delta}_m^2)}.$$

Bounding Term IV.: For term IV., we consider the expected difference at each round $1 \leq i \leq m$ and exploit the independence of τ_{t,J_t} and R_{t,J_t}. Consider first $i \geq 2$ and let j'_i be the arm played just before arm j is played in the ith phase (allowing for j'_i to be the last arm played in phase $i-1$). Then, much in the same way as

Lemma 5.21,

$$\begin{aligned}
\sum_{t=S_{i,j}-d-1}^{S_{i,j}-1} \mathbb{E}[B_{i,t}|\mathcal{G}_{t-1}] &= \sum_{t=S_{i,j}-d-1}^{S_{i,j}-1} \mathbb{E}[R_{t,J_t}\mathbb{I}\{\tau_{t,J_t}+t \geq S_{i,j}\}|\mathcal{G}_{t-1}] \\
&= \sum_{s'=d+1}^{\infty}\sum_{s=s'}^{\infty}\mathbb{I}\{S_i=s', S_{i,j}=s\}\sum_{t=s-d}^{s-1}\mathbb{E}[R_{t,J_t}\mathbb{I}\{\tau_{t,J_t}+t \geq S_{i,j}\}|\mathcal{G}_{t-1}] \\
&= \sum_{s'=d+1}^{\infty}\sum_{s=s'}^{\infty}\sum_{t=s-d}^{s-1}\sum_{k=1}^{K}\mathbb{E}[R_{t,J_t}\mathbb{I}\{S_i=s', S_{i,j}=s, \tau_{t,J_t}+t \geq S_{i,j}, J_t=k\}|\mathcal{G}_{t-1}] \\
&\qquad \text{(Due to the bridge period } \{S_i=s', S_{i,j}=s\} \in \mathcal{G}_{t-1} \text{ for } t \geq s-d \geq s'-d) \\
&= \sum_{s'=d+1}^{\infty}\sum_{s=s'}^{\infty}\sum_{t=s-d}^{s-1}\sum_{k=1}^{K}\mathbb{I}\{S_i=s', S_{i,j}=s, J_t=k\}\mathbb{E}[R_{t,k}\mathbb{I}\{\tau_{t,k}+t \geq s\}|\mathcal{G}_{t-1}] \\
&= \sum_{s'=d+1}^{\infty}\sum_{s=s'}^{\infty}\sum_{t=s-d}^{s-1}\sum_{k=1}^{K}\mu_k\mathbb{I}\{S_i=s', S_{i,j}=s, J_t=k\}\mathbb{P}(\tau \geq s-t) \\
&= \mu_{j_i'}\sum_{l=0}^{d-1}\mathbb{P}(\tau > l).
\end{aligned}$$

A similar argument works for $i=1, j>1$ with the simplification that $S_{i,j}$ is not a random quantity but known . Finally, for $i=1, j=1$ the sum is 0. Furthermore, using a similar argument, for all i, j,

$$\begin{aligned}
\sum_{t=S_{i,j}}^{U_{i,j}} \mathbb{E}[C_{i,t}|\mathcal{G}_{t-1}] &= \sum_{t=U_{i,j}-d+1}^{U_{i,j}} \mathbb{E}[C_{i,t}|\mathcal{G}_{t-1}] \\
&= \sum_{s'=d+1}^{\infty}\sum_{s=s'}^{\infty}\sum_{t=s-d}^{s}\mathbb{E}[R_{t,j}\mathbb{I}\{\tau_{t,j}+t > s\}\mathbb{I}\{U_{i,j}=s, S_i=s'\}|\mathcal{G}_{t-1}] \\
&= \mu_j\sum_{s=d+1}^{\infty}\mathbb{I}\{U_{i,j}=s, S_i=s'\}\sum_{t=s-d}^{s}\mathbb{P}(\tau+t > s) \\
&= \mu_j\sum_{l=0}^{d-1}\mathbb{P}(\tau > l).
\end{aligned}$$

Combining these we get the following bound for term IV for all $(i,j) \neq (1,1)$,

$$\sum_{t=S_{i,j}-d-1}^{S_{i,j}-1} \mathbb{E}[B_{i,t}|\mathcal{G}_{t-1}] - \sum_{t=S_{i,j}}^{U_{i,j}} \mathbb{E}[C_{i,t}|\mathcal{G}_{t-1}] \leq \mu_{j_i'} \sum_{l=0}^{d-1} \mathbb{P}(\tau > l) - \mu_j \sum_{l=0}^{d-1} \mathbb{P}(\tau > l)$$
$$\leq |\mu_{j_i'} - \mu_j| \mathbb{E}[\tau].$$

If $(i,j) = (1,1)$ then we have the upper bounded by $\mu_1 \mathbb{E}[\tau] \leq \mathbb{E}[\tau] = \tilde{\Delta}_0 \mathbb{E}[\tau]$ since no pay-off seeps in and we define $\tilde{\Delta}_0 = 1$.

Let p_i be the probability that the confidence bounds for one arm fail in phase i and $p_0 = 0$. Then, the probability that either arm j_i' or j is active in phase i when it should have been eliminated in or before phase $i-1$ is less than $2p_{i-1}$. If neither arm should have been eliminated by phase i, this means that their mean rewards are within $\tilde{\Delta}_{i-1}$ of each other. This follows since if the confidence bounds on arms j and j' both hold and both arms are active in phase i, then $|\mu_j - \mu_{j'}| < \Delta_{i-1}$. Hence, with probability greater than $1 - 2p_{i-1}$,

$$\sum_{t=S_{i,j}-d-1}^{S_{i,j}-1} \mathbb{E}[B_{i,t}|\mathcal{G}_{t-1}] - \sum_{t=S_{i,j}}^{U_{i,j}} \mathbb{E}[C_{i,t}|\mathcal{G}_{t-1}] \leq \tilde{\Delta}_{i-1} \mathbb{E}[\tau].$$

Then, summing over all phases gives that with probability greater than $1 - 2\sum_{i=0}^{m-1} p_i$,

$$\sum_{i=1}^{m} \left(\sum_{t=S_{i,j}-d-1}^{S_{i,j}-1} \mathbb{E}[B_{i,t}|\mathcal{G}_{t-1}] - \sum_{t=S_{i,j}}^{U_{i,j}} \mathbb{E}[C_{i,t}|\mathcal{G}_{t-1}] \right) \leq \mathbb{E}[\tau] \sum_{i=1}^{m} \tilde{\Delta}_{i-1} = \mathbb{E}[\tau] \sum_{i=0}^{m-1} \frac{1}{2^i} \leq 2\mathbb{E}[\tau].$$

Combining all Terms: To get the final high probability bound, we sum the bounds for each term I.-IV.. Then, with probability greater than $1 - (\frac{3}{T\tilde{\Delta}_m^2} + 2\sum_{i=1}^{m-1} p_i)$ either

$j \notin \mathcal{A}_m$ or arm j is played n_m times by the end of phase m and

$$\begin{aligned}\frac{1}{n_m}\sum_{t\in T_j(m)}(X_t-\mu_j) &\le \frac{4\log(T\tilde{\Delta}_m^2)}{3n_m}+\left(\frac{2}{\sqrt{8}}+\frac{1}{\sqrt{2}}\right)\sqrt{\frac{\log(T\tilde{\Delta}_m^2)}{n_m}}+\frac{2\mathbb{E}[\tau]}{n_m}\\ &\le \frac{4\log(T\tilde{\Delta}_m^2)}{3n_m}+\sqrt{\frac{2\log(T\tilde{\Delta}_m^2)}{n_m}}+\frac{2\mathbb{E}[\tau]}{n_m}=w_m.\end{aligned}$$

Using the fact that $p_0 = 0$ and substituting the other p_i's using the recursive relationship $p_i = \frac{3}{T\tilde{\Delta}_i^2} + 2\sum_{l=1}^{i-1} p_l$ gives,

$$\begin{aligned}\frac{3}{T\tilde{\Delta}_m^2}+2\sum_{i=0}^{m-1}p_i &= \frac{3}{T\tilde{\Delta}_m^2}+2(\frac{3}{T\tilde{\Delta}_{m-1}^2}+2(p_{m-2}+\cdots+p_1)+p_{m-2}+\cdots+p_1)\\ &= \frac{3}{T\tilde{\Delta}_m^2}+2(\frac{3}{T\tilde{\Delta}_{m-1}^2}+3(p_{m-2}+\cdots+p_1))\\ &= \frac{3}{T\tilde{\Delta}_m^2}+2(\frac{3}{T\tilde{\Delta}_{m-1}^2}+3(\frac{3}{T\tilde{\Delta}_{m-2}^2}+3(p_{m-3}+\cdots+p_1))\\ &\le \sum_{i=1}^{m}3^{m-i}\frac{3}{T\tilde{\Delta}_i^2}\\ &= \frac{3}{T}\sum_{i=1}^{m}3^{m-i}2^{2i}\\ &= \frac{3}{T}\sum_{i=1}^{m}3^{m-i}4i\\ &= \frac{3}{T}\sum_{i=1}^{m}(\frac{3}{4})^{m-i}4^{m-i}4^i\\ &= \frac{3\times 4^m}{T}\sum_{i=1}^{m}(\frac{3}{4})^{m-i}\\ &\le \frac{12}{T\tilde{\Delta}_m^2}.\end{aligned}$$

Hence, with probability greater than $1-\frac{12}{T\tilde{\Delta}_m^2}$, either $j \notin \mathcal{A}_m$ or $\frac{1}{n_m}\sum_{t\in T_j(m)}(X_t-\mu_j) \le w_m$.

Defining $\mathbf{n_m}$: The above results rely on the assumption that $n_m \geq md$, so that only the previous arm can corrupt our observations. In practice, if d is too large then we will not want to play each active arm d times per phase because we will end up playing sub-optimal arms too many times. In this case, it is better to ignore the bound on the delay and use the results from Lemma 5.1 to set n_m as in (5.14). Formalizing this gives

$$n_m = \max\Bigg\{m\tilde{d}_m, \left\lceil \frac{1}{\tilde{\Delta}_m^2}\left(\sqrt{2\log(T\tilde{\Delta}_m^2)} + \sqrt{2\log(T\tilde{\Delta}_m^2) + \frac{8}{3}\tilde{\Delta}_m\log(T\tilde{\Delta}_m^2) + 4\tilde{\Delta}_m\mathbb{E}[\tau]}\right)^2\right\rceil\Bigg\} \tag{5.18}$$

where $\tilde{d}_m = \min\{d, \frac{(5.14)}{m}\}$. This ensures that if d is small, we play each active arm enough times to ensure that $w_m \leq \frac{\tilde{\Delta}_m}{2}$ for w_m in (5.17). Similarly, for large d, by Lemma 5.1, we know that n_m is suffiently large to guarantee $w_m \leq \frac{\tilde{\Delta}_m}{2}$ for w_m from (5.10). □

5.C.2 Regret Bounds

We now prove the regret bound given in Theorem 5.6. Note that for these results, it is necessary to use the bridge period of the algorithm.

Theorem 5.6. *Under Assumption 1 and bounded delay Assumption 2, the expected regret of Algorithm 5.1 satisfies*

$$\mathbb{E}[\mathfrak{R}_T] \leq \sum_{j=1;j\neq j^*}^{K} O\left(\frac{\log(T\Delta_j^2)}{\Delta_j} + \mathbb{E}[\tau] + \min\left\{d, \frac{\log(T\Delta_j^2)}{\Delta_j} + \log(\frac{1}{\Delta_j})\mathbb{E}[\tau]\right\}\right).$$

Proof. For any sub-optimal arm j, define M_j to be the random variable representing the phase arm j is eliminated in and note that if M_j is finite, $j \in \mathcal{A}_{M_j}$ but $j \notin \mathcal{A}_{M_j+1}$. Then let m_j be the phase arm j *should* be eliminated in, that is $m_j = \min\{m|\tilde{\Delta}_m <$

$\frac{\Delta_j}{2}\}$ and note that, from the definition of $\tilde{\Delta}_m$ in our algorithm, we get the relations

$$2^m = \frac{1}{\tilde{\Delta}_m}, \quad 2\tilde{\Delta}_{m_j} = \tilde{\Delta}_{m_j-1} \geq \frac{\Delta_j}{2} \quad \text{and so,} \quad \frac{\Delta_j}{4} \leq \tilde{\Delta}_{m_j} \leq \frac{\Delta_j}{2}. \tag{5.19}$$

Define $\mathfrak{R}_T^{(j)}$ to be the regret contribution from each arm $1 \leq j \leq K$ and let M^* be the round where the optimal arm $j^*$is eliminated. Hence,

$$\begin{aligned}
\mathbb{E}[\mathfrak{R}_T] &= \mathbb{E}\Big[\sum_{j=1}^{K} \mathfrak{R}_T^{(j)}\Big] = \mathbb{E}\Big[\sum_{m=0}^{\infty}\sum_{j=1}^{K} \mathfrak{R}_T^{(j)}\mathbb{I}\{M^* = m\}\Big] \\
&= \mathbb{E}\Big[\sum_{m=0}^{\infty}\sum_{j:m_j<m} \mathfrak{R}_T^{(j)}\mathbb{I}\{M^* = m\} + \sum_{j:m_j\geq m} \mathfrak{R}_T^{(j)}\mathbb{I}\{M^* = m\}\Big] \\
&= \underbrace{\mathbb{E}\Big[\sum_{m=0}^{\infty}\sum_{j:m_j<m} \mathfrak{R}_T^{(j)}\mathbb{I}\{M^* = m\}\Big]}_{\text{I.}} + \underbrace{\mathbb{E}\Big[\sum_{m=0}^{\infty}\sum_{j:m_j\geq m} \mathfrak{R}_T^{(j)}\mathbb{I}\{M^* = m\}\Big]}_{\text{II.}}
\end{aligned}$$

We will bound the regret in each of these cases in turn. First, however, we need the following results.

Lemma 5.22. *For any suboptimal arm j, if $j^* \in \mathcal{A}_{m_j}$, then the probability arm j is* not *eliminated by round m_j is,*

$$\mathbb{P}(M_j > m_j \text{ and } M^* \geq m_j) \leq \frac{24}{T\tilde{\Delta}_{m_j}^2}$$

Proof. The proof is exactly that of Lemma 5.18 but using Lemma 5.5 to bound the probability of the confidence bounds on either arm j or j^* failing. □

Define the event $F_j(m) = \{\bar{X}_{m,j^*} < \bar{X}_{m,j} - \tilde{\Delta}_m\} \cap \{j, j^* \in \mathcal{A}_m\}$ to be the event that arm j^* is eliminated by arm j in phase m. The probability of this occurring is bounded in the following lemma.

Lemma 5.23. *The probability that the optimal arm j^* is eliminated in round $m < \infty$*

by the suboptimal arm j is bounded by

$$\mathbb{P}(F_j(m)) \leq \frac{24}{T\tilde{\Delta}_m^2}$$

Proof. Again, the proof follows from Lemma 5.19 but using Lemma 5.5 to bound the probability of the confidence bounds failing. □

We now return to bounding the expected regret in each of the two cases.

Bounding Term I. To bound the first term, we consider the cases where arm j is eliminated in or before the correct round ($M_j \leq m_j$) and where arm j is eliminated late ($M_j > m_j$). Then,

$$\begin{aligned}
\mathbb{E}\Big[\sum_{m=0}^{\infty}\sum_{j:m_j<m}\mathfrak{R}_T^{(j)}\mathbb{I}\{M^*=m\}\Big] &= \mathbb{E}\Big[\sum_{j=1}^{K}\mathfrak{R}_T^{(j)}\mathbb{I}\{M^*\geq m_j\}\Big] \\
&= \mathbb{E}\Big[\sum_{j=1}^{K}\mathfrak{R}_T^{(j)}\mathbb{I}\{M^*\geq m_j\}\mathbb{I}\{M_j\leq m_j\}\Big] \\
&\qquad + \mathbb{E}\Big[\sum_{j=1}^{K}\mathfrak{R}_T^{(j)}\mathbb{I}\{M^*\geq m_j\}\mathbb{I}\{M_j> m_j\}\Big] \\
&\leq \sum_{j=1}^{K}\mathbb{E}[\mathfrak{R}_T^{(j)}\mathbb{I}\{M_j\leq m_j\}] + \sum_{j=1}^{K}\mathbb{E}[T\Delta_j\mathbb{I}\{M^*\geq m_j, M_j> m_j\}] \\
&\leq \sum_{j=1}^{K}2\Delta_j n_{m_j,j} + \sum_{j=1}^{K}T\Delta_j\mathbb{P}(M_j>m_j \text{ and } M^*\geq m_j) \\
&\leq \sum_{j=1}^{K}2\Delta_j n_{m_j,j} + \sum_{j=1}^{K}T\Delta_j\frac{24}{T\tilde{\Delta}_{m_j}^2} \\
&\leq \sum_{j=1}^{K}\left(2\Delta_j n_{m_j,j} + \frac{384}{\Delta_j}\right),
\end{aligned}$$

where the extra factor of 2 comes from the fact that each arm will be played n_m times by the end of phase m to get the data for the estimated mean, then in the worst case, arm j is chosen as the arm to be played in the bridge period of each phase that it is

active, and thus is played another n_m times.

Bounding Term II For the second term, we use the results from Theorem 5.2, but using Lemma 5.22 to bound the probability a suboptimal arm is eliminated in a later round and Lemma 5.23 to bound the probability j^* is eliminated by a suboptimal arm. Hence,

$$\mathbb{E}\Big[\sum_{m=0}^{\infty}\sum_{j:m_j\geq m}\mathfrak{R}_T^{(j)}\mathbb{I}\{M^*=m\}\Big]\leq\sum_{j=1}^{K}\frac{1536}{\Delta_j}.$$

Combining the regret from terms I and II gives,

$$\mathbb{E}[\mathfrak{R}_T]\leq\sum_{j=1}^{K}\left(\frac{1920}{\Delta_j}+2\Delta_j n_{m_j,j}\right)$$

Hence, all that remains is to bound n_m in terms of Δ_j, T and d. Using $L_{m,T}=\log(T\tilde{\Delta}_m^2)$, we have that,

$$\begin{aligned}
n_{m_j,j} &= \max\Bigg\{m_j\tilde{d}_{m_j}, \\
&\qquad\left\lceil\frac{1}{\tilde{\Delta}_m^2}\left(\sqrt{2\log(T\tilde{\Delta}_m)}+\sqrt{2\log(T\tilde{\Delta}_m)+\frac{8}{3}\tilde{\Delta}_m\log(T\tilde{\Delta}_m)+4\tilde{\Delta}_m\mathbb{E}[\tau]}\right)^2\right\rceil\Bigg\} \\
&\leq \max\left\{m_j\tilde{d}_{m_j},\left\lceil\frac{1}{\tilde{\Delta}_{m_j}^2}\left(8L_{m_j,T}+\frac{16}{3}\tilde{\Delta}_{m_j}L_{m_j,T}+8\tilde{\Delta}_{m_j}\mathbb{E}[\tau]\right)\right\rceil\right\} \\
&\leq \max\left\{m_j\tilde{d}_{m_j},1+\frac{8L_{m_j,T}}{\tilde{\Delta}_{m_j}^2}+\frac{8L_{m_j,T}}{3\tilde{\Delta}_{m_j}}+\frac{8\mathbb{E}[\tau]}{\tilde{\Delta}_{m_j}}\right\} \\
&\leq \max\left\{m_j\tilde{d}_{m_j},1+\frac{128L_{m_j,T}}{\Delta_j^2}+\frac{32L_{m_j,T}}{\Delta_j}+\frac{32\mathbb{E}[\tau]}{\Delta_j}.\right\}
\end{aligned}$$

where we have used $(a+b)^2\leq 2(a^2+b^2)$ for $a,b\geq 0$.

Hence, using the definition of $\tilde{d}_m=\min\{d,\frac{(5.14)}{m}\}$ and the results from Theorem 5.2, the total expected regret from ODAAF with bounded delays can be bounded

by,

$$\mathbb{E}[\mathfrak{R}_t] \leq \sum_{j=1;j\neq j^*}^{K} \max\Bigg\{ \min\{d, (5.16)\}, \left(\frac{256\log(T\Delta_j^2)}{\Delta_j} + 64\mathbb{E}[\tau] + \frac{1920}{\Delta_j} + 64\log(T\Delta_j^2) + 2\Delta_j\right)\Bigg\}. \quad (5.20)$$

$$\leq \sum_{j=1;j\neq j^*}^{K} \left(\frac{256\log(T\Delta_j^2)}{\Delta_j} + 64\mathbb{E}[\tau] + \frac{1920}{\Delta_j} + 64\log(T\Delta_j^2) + 2\Delta_j + \min\left\{d, \frac{128\log(T\Delta_j^2)}{\Delta_j} + 96\log(4/\Delta_j)\mathbb{E}[\tau]\right\}\right)$$

□

Note that the constants in these regret bounds can be improved by only requiring the confidence bounds in phase m to hold with probability $\frac{1}{T\tilde{\Delta}_m}$ rather than $\frac{1}{T\tilde{\Delta}_m^2}$. This comes at a cost of increasing the logarithmic term to $\log(T\Delta_j)$. We now prove the problem independent regret bound,

Corollary 5.7. *For any problem instance satisfying Assumptions 1 and 2 with* $d \leq \sqrt{\frac{T\log K}{K}} + \mathbb{E}[\tau]$*, the expected regret of Algorithm 5.1 satisfies*

$$\mathbb{E}[\mathfrak{R}_T] \leq O(\sqrt{KT\log(K)} + K\mathbb{E}[\tau]).$$

Proof. We consider the maximal value each part of the regret in (5.20) can take. From Corollary 5.3, the first term is bounded by

$$O(\min\{Kd, \sqrt{KT\log K} + K\log(T)\mathbb{E}[\tau]\}).$$

For the first term, we again set $\lambda = \sqrt{\frac{K\log(K)e^2}{T}}$. Then, as in corollary Corollary 5.3, for constants $C_1, C_2 > 0$, we bound the regret contribution by

$$\sum_{j:\Delta_j\leq\lambda} \mathbb{E}[\mathfrak{R}_t^{(j)}] + \sum_{j:\Delta_j>\lambda} \mathbb{E}[\mathfrak{R}_T^{(j)}] \leq \frac{KC_1\log(T\lambda^2)}{\lambda} + C_2K\mathbb{E}[\tau] + T\lambda.$$

Then, substituting in for λ implies that the second term of (5.20) is $O(\sqrt{KT\log K} + K\mathbb{E}[\tau])$.

For $d \leq \sqrt{\frac{T\log K}{K}} + \mathbb{E}[\tau]$, $\min\{Kd, \sqrt{KT\log K} + K\log T\mathbb{E}[\tau]\} \leq \sqrt{KT\log K} + K\mathbb{E}[\tau]$. Hence the bound in (5.20) gives

$$\mathbb{E}[\mathfrak{R}_T] \leq O(\sqrt{KT\log K} + K\mathbb{E}[\tau] + \sqrt{KT\log K} + K\mathbb{E}[\tau]) = O(\sqrt{KT\log K} + K\mathbb{E}[\tau]).$$

□

5.D Results for Delay with Known and Bounded Variance and Expectation

5.D.1 High Probability Bounds

Lemma 5.24. *Under Assumption 1 of known expected value and 3 of known (bound on) the expectation and variance of the delay, and choice of n_m given in (5.7), the estimates $\bar{X}_{m,j}$ obtained by Algorithm 5.1 satisfy the following: For any arm j and phase m, with probability at least $1 - \frac{12}{T\tilde{\Delta}_m^2}$, either $j \notin \mathcal{A}_m$ or*

$$\bar{X}_{m,j} - \mu_j \leq \tilde{\Delta}_m/2.$$

Proof. Let

$$w_m = \frac{4\log(T\tilde{\Delta}_m^2)}{3n_m} + \sqrt{\frac{2\log(T\tilde{\Delta}_m^2)}{n_m} + \frac{2\mathbb{E}[\tau] + 4\mathbb{V}(\tau)}{n_m}}. \tag{5.21}$$

We show that with probability greater than $1 - \frac{12}{T\tilde{\Delta}_m^2}$, $j \notin \mathcal{A}_m$ or $\frac{1}{n_m}\sum_{t\in T_j(m)}(X_t - \mu_j) \leq w_m$.

For any arm j, phase i and time t, define,

$$A_{i,t} = R_{t,J_t}\mathbb{I}\{\tau_{t,J_t} + t \geq S_i\}, \qquad B_{i,t} = R_{t,J_t}\mathbb{I}\{\tau_{t,J_t} + t \geq S_{i,j}\}, \qquad (5.22)$$
$$C_{i,t} = R_{t,J_t}\mathbb{I}\{\tau_{t,J_t} + t > U_{i,j}\}$$

as in (5.11) and

$$Q_t = \sum_{i=1}^{m}(A_{i,t}\mathbb{I}\{S_{i-1,j} \leq t \leq S_i - \nu_{i-1} - 1\} + B_{i,t}\mathbb{I}\{S_i - \nu_{i-1} \leq t \leq S_{i,j} - 1\}),$$
$$P_t = \sum_{i=1}^{m} C_{i,t}\mathbb{I}\{S_{i,j} \leq t \leq U_{i,j}\},$$

where $\nu_i = n_i - n_{i-1}$ is the number of times each active arm is played in phase $i \geq 1$ (assume $n_0 = 0$). Recall from the proof of Theorem 5.2, $\mathbb{I}_i\{H\} := \mathbb{I}\{H \cap \{j \in \mathcal{A}_i\}\} \leq \mathbb{I}\{H\}$ and for all arms j and phases i, $\mathbb{I}_i\{\tau_{t,J_t} + t \geq S_{i,j}\} = \mathbb{I}\{\tau_{t,J_t} + t \geq S_{i,j}\}$ and $\mathbb{I}_i\{\tau_{t,J_t} + t > U_{i,j}\} = \mathbb{I}\{\tau_{t,J_t} + t > U_{i,j}\}$.

Then, using the convention $S_0 = S_{0,j} = 0$ for all arms j, we use the decomposition,

$$\begin{aligned}
\sum_{i=1}^{m}\sum_{t=S_{i,j}}^{U_{i,j}}(X_t-\mu_j) &\le \sum_{i=1}^{m}\Bigg(\sum_{t=S_{i-1,j}}^{S_{i,j}-1} R_{t,J_t}\mathbb{I}_i\{\tau_{t,J_t}+t\ge S_{i,j}\}+\sum_{t=S_{i,j}}^{U_{i,j}}(R_{t,J_t}-\mu_j) \\
&\qquad -\sum_{t=S_{i,j}}^{U_{i,j}} R_{t,J_t}\mathbb{I}_i\{\tau_{t,J_t}+t>U_{i,j}\}\Bigg) \\
&\le \sum_{i=1}^{m}\Bigg(\sum_{t=S_{i-1,j}}^{S_i-\nu_{i-1}-1} R_{t,J_t}\mathbb{I}\{\tau_{t,J_t}+t\ge S_i\}+\sum_{t=S_i-\nu_{i-1}}^{S_{i,j}-1} R_{t,J_t}\mathbb{I}\{\tau_{t,J_t}+t\ge S_{i,j}\} \\
&\qquad +\sum_{t=S_{i,j}}^{U_{i,j}}(R_{t,J_t}-\mu_j)-\sum_{t=S_{i,j}}^{U_{i,j}} R_{t,J_t}\mathbb{I}\{\tau_{t,J_t}+t>U_{i,j}\}\Bigg) \\
&= \sum_{i=1}^{m}\Bigg(\sum_{t=S_{i-1,j}}^{S_i-\nu_{i-1}-1} A_{i,t}+\sum_{t=S_i-\nu_{i-1}}^{S_{i,j}-1} B_{i,t}+\sum_{t=S_{i,j}}^{U_{i,j}}(R_{t,J_t}-\mu_j)-\sum_{t=S_{i,j}}^{U_{i,j}} C_{i,t}\Bigg) \\
&= \sum_{i=1}^{m}\sum_{t=S_{i,j}}^{U_{i,j}}(R_{t,J_t}-\mu_j)+\sum_{t=1}^{S_{m,j}} Q_t-\sum_{t=1}^{U_{m,j}} P_t \\
&= \underbrace{\sum_{i=1}^{m}\sum_{t=S_{i,j}}^{U_{i,j}}(R_{t,J_t}-\mu_j)}_{\text{Term I.}}+\underbrace{\sum_{t=1}^{S_{m,j}}(Q_t-\mathbb{E}[Q_t|\mathcal{G}_{t-1}])}_{\text{Term II.}}+\underbrace{\sum_{t=1}^{U_{m,j}}(\mathbb{E}[P_t|\mathcal{G}_{t-1}]-P_t)}_{\text{Term III.}} \\
&\qquad +\underbrace{\sum_{t=1}^{S_{m,j}}\mathbb{E}[Q_t|\mathcal{G}_{t-1}]-\sum_{t=1}^{U_{m,j}}\mathbb{E}[P_t|\mathcal{G}_{t-1}]}_{\text{Term IV.}},
\end{aligned} \tag{5.23}$$

Recall that the filtration $\{\mathcal{G}_s\}_{s=0}^{\infty}$ is defined by $\mathcal{G}_0 = \{\Omega, \emptyset\}$ and

$$\mathcal{G}_t = \sigma(X_1, \ldots, X_t, J_1, \ldots, J_t, \tau_{1,J_1}, \ldots, \tau_{t,J_t}, R_{1,J_1}, \ldots R_{t,J_t}).$$

Furthermore, we have defined $S_{i,j} = \infty$ if arm j is eliminated before phase i and $S_i = \infty$ if the algorithm stops before reaching phase i.

Outline of proof: We will bound each term of the above decomposition in turn. We first show in Lemma 5.25 how the bounded second moment information can be

incorporated using Chebychev's inequality. In Lemma 5.26, we show that $Z_t = Q_t - \mathbb{E}[Q_t|\mathcal{G}_{t-1}]$ is a martingale difference sequence and bound its variance in Lemma 5.27 before using Freedman's inequality. Then in Lemma 5.28, we provide alternative (tighter) bounds on $A_{i,t}, B_{i,t}, C_{i,t}$ which are used to bound term IV.. All these results are then combined to give a high probability bound on the entire decomposition.

Lemma 5.25. *For any $a > \lfloor \mathbb{E}[\tau] \rfloor + 1$, $a \in \mathbb{N}$,*

$$\sum_{l=a}^{\infty} \mathbb{P}(\tau \geq l) \leq \frac{\mathbb{V}(\tau)}{a - \lfloor \mathbb{E}[\tau] \rfloor - 1}.$$

Proof. For any $b > a$, $b \in \mathbb{N}$, and by denoting $\xi \doteq \lfloor \mathbb{E}(\tau) \rfloor$,

$$\begin{aligned}
\sum_{l=a}^{b} \mathbb{P}(\tau \geq l) = \sum_{l=a}^{b} \mathbb{P}(\tau - \xi \geq l - \xi) &= \sum_{l=a-\xi}^{b-\xi} \mathbb{P}(\tau - \xi \geq l) \\
&\leq \sum_{l=a-\xi}^{b-\xi} \frac{\mathbb{V}(\tau)}{l^2} \\
&\qquad \text{(by Chebychev's inequality since } l + \xi > \mathbb{E}[\tau] \text{ for } l \geq a - \xi) \\
&\leq \mathbb{V}(\tau) \sum_{l=a-\xi-1}^{b-\xi-1} \frac{1}{l(l+1)} \\
&= \mathbb{V}(\tau) \sum_{l=a-\xi-1}^{b-\xi-1} \left(\frac{1}{l} - \frac{1}{l+1} \right) \\
&= \mathbb{V}(\tau) \left(\frac{1}{a-\xi-1} - \frac{1}{b-\xi} \right).
\end{aligned}$$

Hence, taking $b \to \infty$ gives

$$\sum_{l=a}^{\infty} \mathbb{P}(\tau \geq l) \leq \mathbb{V}(\tau) \frac{1}{a-\xi-1}.$$

□

Lemma 5.26. *Let $Y_s = \sum_{t=1}^{s} (Q_t - \mathbb{E}[Q_t|\mathcal{G}_{t-1}])$ for all $s \geq 1$, and $Y_0 = 0$. Then $\{Y_s\}_{s=0}^{\infty}$ is a martingale with respect to the filtration $\{\mathcal{G}_s\}_{s=0}^{\infty}$ with increments $Z_s =$*

$Y_s - Y_{s-1} = Q_s - \mathbb{E}[Q_s|\mathcal{G}_{s-1}]$ *satisfying* $\mathbb{E}[Z_s|\mathcal{G}_{s-1}] = 0, |Z_s| \leq 1$ *for all* $s \geq 1$.

Proof. To show $\{Y_s\}_{s=0}^{\infty}$ is a martingale we need to show that Y_s is $\mathcal{G}_s$-measurable for all s and $\mathbb{E}[Y_s|\mathcal{G}_{s-1}] = Y_{s-1}$.

Measurability: We show that $A_{i,s}\mathbb{I}\{S_{i-1,j} \leq s \leq S_i - \nu_{i-1}\} + B_{i,s}\mathbb{I}\{S_i - \nu_{i-1} + 1 \leq s \leq S_{i,j} - 1\}$ is $\mathcal{G}_s$-measurable for every $i \leq m$. This then suffices to show that Y_s is $\mathcal{G}_s$-measurable since each Q_t is a sum of such terms and the filtration $\mathcal{G}_s$ is non-decreasing in s.

First note that by definition of $\mathcal{G}_s$, τ_{t,J_t}, R_{t,J_t} are all $\mathcal{G}_s$-measurable for $t \leq s$. It is sufficient to show that $\mathbb{I}\{\tau_{s,J_s} + s \geq S_i, S_{i-1,j} \leq s \leq S_i - \nu_i\} + \mathbb{I}\{\tau_{s,J_s} + s \geq S_{i,j}, S_i - \nu_{i-1} + 1 \leq s \leq S_{i,j} - 1\}$ is $\mathcal{G}_s$-measurable since the product of measurable functions is measurable. The first summand is $\mathcal{G}_s$ measurable since $\{S_{i-1,j} \leq s\} \in \mathcal{G}_s$ and $\{S_i = s', S_{i-1,j} \leq s\} \in \mathcal{G}_s$ for all $s' \in \mathbb{N} \cup \{\infty\}$. So the union $\bigcup_{s' \in \mathbb{N}\cup\{\infty\}} \{\tau_{s,J_s} + s \geq s', S_{i-1,j} \leq s \leq s' - \nu_i, S_i = s'\} = \{\tau_{s,J_s} + s \geq S_i, S_{i-1,j} \leq s \leq S_i - \nu_{i-1}\}$ is an element of $\mathcal{G}_s$. The same argument works for the second summand since $\{S_{ij} = s', S_i - \nu_{i-1} \leq s\} \in \mathcal{G}_s$ for all $s' \in \mathbb{N} \cup \{\infty\}$

Increments: Hence, to show that $\{Y_s\}_{s=0}^{\infty}$ is a martingale with respect to the filtration $\{\mathcal{G}_s\}_{s=0}^{\infty}$ it just remains to show that the increments conditional on the past are zero. For any $s \geq 1$, we have that

$$Z_s = Y_s - Y_{s-1} = \sum_{t=1}^{s}(Q_t - \mathbb{E}[Q_t|\mathcal{G}_{t-1}]) - \sum_{t=1}^{s-1}(Q_t - \mathbb{E}[Q_t|\mathcal{G}_{t-1}]) = Q_s - \mathbb{E}[Q_s|\mathcal{G}_{s-1}].$$

Then,

$$\mathbb{E}[Z_s|\mathcal{G}_{s-1}] = \mathbb{E}[Q_s - \mathbb{E}[Q_s|\mathcal{G}_{s-1}]|\mathcal{G}_{s-1}] = \mathbb{E}[Q_s|\mathcal{G}_{s-1}] - \mathbb{E}[Q_s|\mathcal{G}_{s-1}] = 0$$

and so $\{Y_s\}_{s=0}^{\infty}$ is a martingale.

Lastly, since for any t and $\omega \in \Omega$, there is only one i where one of $\mathbb{I}\{S_{i-1,j} \leq t \leq S_i - \nu_{i-1}\}$ or $\mathbb{I}\{S_i - \nu_{i-1} + 1 \leq t \leq S_{i,j} - 1\}$ is equal to one (they cannot both be one),

and by definition of R_{t,J_t}, $A_{i,t}, B_{i,t} \leq 1$, it follows that $|Z_s| = |Q_s - \mathbb{E}[Q_s|\mathcal{G}_{s-1}]| \leq 1$ for all s. □

Lemma 5.27. *For any $t \geq 1$, let $Z_t = Q_t - \mathbb{E}[Q_t|\mathcal{G}_{t-1}]$, then*

$$\sum_{t=1}^{S_{m,j}-1} \mathbb{E}[Z_t^2|\mathcal{G}_{t-1}] \leq m\mathbb{E}[\tau] + m\mathbb{V}(\tau).$$

Proof. Let us denote $S' \doteq S_{m,j} - 1$. Observe that

$$\begin{aligned}
&\sum_{t=1}^{S'} \mathbb{E}[Z_t^2|\mathcal{G}_{t-1}] = \sum_{t=1}^{S'} \mathbb{V}(Q_t|\mathcal{G}_{t-1}) \leq \sum_{t=1}^{S'} \mathbb{E}[Q_t^2|\mathcal{G}_{t-1}] \\
&= \sum_{t=1}^{S'} \mathbb{E}\Bigg[\Bigg(\sum_{i=1}^{m}(A_{i,t}\mathbb{I}\{S_{i-1,j} \leq t \leq S_i - \nu_{i-1} - 1\} \\
&\qquad\qquad + B_{i,t}\mathbb{I}\{S_i - \nu_{i-1} \leq t \leq S_{i,j} - 1\})\Bigg)^2 \Bigg| \mathcal{G}_{t-1}\Bigg].
\end{aligned}$$

Then all indicator terms $\mathbb{I}\{S_{i-1,j} \leq t \leq S_i - \nu_{i-1} - 1\}$ and $\mathbb{I}\{S_i - \nu_{i-1} \leq t \leq S_{i,j} - 1\}$ for all $i = 1, \ldots, m$ are $\mathcal{G}_{t-1}$-measurable and only one can be non zero for any $\omega \in \Omega$. Hence, for any $\omega \in \Omega$, their product must be 0. Furthermore, for any $i, i' \leq m$, $i \neq i'$,

$$\begin{aligned}
A_{i,t}\mathbb{I}\{S_{i-1,j} \leq t \leq S_i - \nu_{i-1} - 1\} \times A_{i',t}\mathbb{I}\{S_{i'-1,j} \leq t \leq S_{i'} - \nu_{i'-1} - 1\} = 0, \\
B_{i,t}\mathbb{I}\{S_i - \nu_{i-1} \leq t \leq S_{i,j} - 1\} \times B_{i',t}\mathbb{I}\{S_{i'} - \nu_{i'-1} \leq t \leq S_{i',j} - 1\} = 0, \\
A_{i,t}\mathbb{I}\{S_{i-1,j} \leq t \leq S_i - \nu_{i-1} - 1\} \times B_{i',t}\mathbb{I}\{S_{i'} - \nu_{i'-1} \leq t \leq S_{i',j} - 1\} = 0, \\
A_{i',t}\mathbb{I}\{S_{i'-1,j} \leq t \leq S_{i'} - \nu_{i'-1} - 1\} \times B_{i,t} \times \mathbb{I}\{S_i - \nu_{i-1} \leq t \leq S_{i,j} - 1\} = 0.
\end{aligned}$$

Using the above we see that,

$$
\begin{aligned}
&\sum_{t=1}^{S'} \mathbb{E}[Z_t^2|\mathcal{G}_{t-1}] \\
&\leq \sum_{t=1}^{S'} \mathbb{E}\Bigg[\Bigg(\sum_{i=1}^{m}(A_{i,t}\mathbb{I}\{S_{i-1,j} \leq t \leq S_i - \nu_{i-1} - 1\} \\
&\qquad\qquad + B_{i,t}\mathbb{I}\{S_i - \nu_{i-1} \leq t \leq S_{i,j} - 1\})\Bigg)^2 \Bigg| \mathcal{G}_{t-1}\Bigg] \\
&= \sum_{t=1}^{S'} \mathbb{E}\Bigg[\sum_{i=1}^{m}(A_{i,t}^2\mathbb{I}\{S_{i-1,j} \leq t \leq S_i - \nu_{i-1} - 1\}^2 \\
&\qquad\qquad + B_{i,t}^2\mathbb{I}\{S_i - \nu_{i-1} \leq t \leq S_{i,j} - 1\}^2) \Big| \mathcal{G}_{t-1}\Bigg] \\
&= \sum_{i=2}^{m}\sum_{t=1}^{S'} \mathbb{E}[A_{i,t}^2\mathbb{I}\{S_{i-1,j} \leq t \leq S_i - \nu_{i-1} - 1\}|\mathcal{G}_{t-1}] \\
&\qquad + \sum_{i=1}^{m}\sum_{t=1}^{S'} \mathbb{E}[B_{i,t}^2\mathbb{I}\{S_i - \nu_i \leq t \leq S_{i,j} - 1\}|\mathcal{G}_{t-1}] \\
&\qquad\qquad\qquad\qquad \text{(using that both indicators are } \mathcal{G}_{t-1}\text{-measurable)} \\
&\leq \sum_{i=2}^{m}\sum_{t=S_{i-1,j}}^{S_i-\nu_{i-1}-1} \mathbb{E}[A_{i,t}^2|\mathcal{G}_{t-1}] + \sum_{i=1}^{m}\sum_{t=S_i-\nu_{i-1}}^{S_{i,j}-1} \mathbb{E}[B_{i,t}^2|\mathcal{G}_{t-1}].
\end{aligned}
$$

Then, for any $i \geq 2$,

$$\begin{aligned}
\sum_{t=S_{i-1,j}}^{S_i-\nu_{i-1}-1} \mathbb{E}[A_{i,t}^2|\mathcal{G}_{t-1}] &= \sum_{t=S_{i-1,j}}^{S_i-\nu_{i-1}-1} \mathbb{E}[R_{t,J_t}^2 \mathbb{I}\{\tau_{t,J_t}+t \geq S_i\}|\mathcal{G}_{t-1}] \\
&\leq \sum_{t=S_{i-1,j}}^{S_i-\nu_{i-1}-1} \mathbb{E}[\mathbb{I}\{\tau_{t,J_t}+t \geq S_i\}|\mathcal{G}_{t-1}] \\
&= \sum_{s=0}^{\infty}\sum_{s'=s}^{\infty} \mathbb{I}\{S_{i-1,j}=s, S_i=s'\} \sum_{t=s}^{s'-\nu_{i-1}-1} \mathbb{E}[\mathbb{I}\{\tau_{t,J_t}+t \geq S_i\}|\mathcal{G}_{t-1}] \\
&= \sum_{s=0}^{\infty}\sum_{s'=s}^{\infty} \sum_{t=s}^{s'-\nu_{i-1}-1} \mathbb{E}[\mathbb{I}\{S_{i-1,j}=s, S_i=s', \tau_{t,J_t}+t \geq S_i\}|\mathcal{G}_{t-1}] \\
&\qquad\qquad \text{(Since } \{S_i=s', S_{i-1,j}=s\} \in \mathcal{G}_{t-1} \text{ for } t \geq s) \\
&= \sum_{s=0}^{\infty}\sum_{s'=s}^{\infty} \sum_{t=s}^{s'-\nu_{i-1}-1} \mathbb{E}[\mathbb{I}\{S_{i-1,j}=s, S_i=s', \tau_{t,J_t}+t \geq s'\}|\mathcal{G}_{t-1}] \\
&= \sum_{s=0}^{\infty}\sum_{s'=s}^{\infty} \mathbb{I}\{S_{i-1,j}=s, S_i=s'\} \sum_{t=s}^{s'-\nu_{i-1}-1} \mathbb{P}(\tau_{t,J_t}+t \geq s') \\
&\qquad\qquad \text{(Since } \{S_i=s', S_{i-1,j}=s\} \in \mathcal{G}_{t-1} \text{ for } t \geq s) \\
&\leq \sum_{s=0}^{\infty}\sum_{s'=s}^{\infty} \mathbb{I}\{S_{i-1,j}=s, S_i=s'\} \sum_{l=\nu_{i-1}+1}^{\infty} \mathbb{P}(\tau > l) \\
&\leq \mathbb{V}[\tau],
\end{aligned}$$

by Lemma 5.25 since $\nu_i \geq \lfloor \mathbb{E}[\tau] \rfloor + 2$ for all i. Likewise, for any $i \geq 2$,

$$\begin{aligned}
\sum_{t=S_i-\nu_{i-1}}^{S_{i,j}-1} \mathbb{E}[B_{i,t}^2|\mathcal{G}_{t-1}] &= \sum_{t=S_i-\nu_{i-1}}^{S_{i,j}-1} \mathbb{E}[R_{t,J_t}^2 \mathbb{I}\{\tau_{t,J_t} + t \geq S_{i,j}\}|\mathcal{G}_{t-1}] \\
&\leq \sum_{t=S_i-\nu_{i-1}}^{S_{i,j}-1} \mathbb{E}[\mathbb{I}\{\tau_{t,J_t} + t \geq S_{i,j}\}|\mathcal{G}_{t-1}] \\
&= \sum_{s=\nu_{i-1}+1}^{\infty} \sum_{s'=s}^{\infty} \mathbb{I}\{S_i = s, S_{i,j} = s'\} \sum_{t=s-\nu_{i-1}}^{s'-1} \mathbb{E}[\mathbb{I}\{\tau_{t,J_t} + t \geq S_{i,j}\}|\mathcal{G}_{t-1}] \\
&= \sum_{s=\nu_{i-1}+1}^{\infty} \sum_{s'=s}^{\infty} \sum_{t=s-\nu_{i-1}}^{s'-1} \mathbb{E}[\mathbb{I}\{S_i = s, S_{i,j} = s', \tau_{t,J_t} + t \geq s'\}|\mathcal{G}_{t-1}] \\
&\qquad\qquad \text{(Since } \{S_{i,j} = s', S_i = s\} \in \mathcal{G}_{t-1} \text{ for } t \geq s - \nu_i - 1) \\
&= \sum_{s=\nu_{i-1}+1}^{\infty} \sum_{s'=s}^{\infty} \mathbb{I}\{S_i = s, S_{i,j} = s'\} \sum_{t=s-\nu_{i-1}}^{s'-1} \mathbb{P}(\tau_{t,J_t} + t \geq s') \\
&\leq \sum_{s=\nu_{i-1}+1}^{\infty} \sum_{s'=s}^{\infty} \mathbb{I}\{S_i = s, S_{i,j} = s'\} \sum_{l=0}^{\infty} \mathbb{P}(\tau > l) \\
&\leq \mathbb{E}[\tau]
\end{aligned}$$

and for $i = 1$ the derivation simplifies since we need to some over 1 to $S_{1,j} - 1$ only. Combining all terms gives the result. □

Lemma 5.28. *For $A_{i,t}, B_{i,t}$ and $C_{i,t}$ defined as in (5.22), let $\nu_i = n_i - n_{i-1}$ be the number of times each arm is played in phase i and j_i' be the arm played directly before arm j in phase i. Then, it holds that, for any arm j and phase $i \geq 1$,*

(i) $$\sum_{t=S_{i-1,j}}^{S_i-\nu_{i-1}-1} \mathbb{E}[A_{i,t}|\mathcal{G}_{t-1}] \leq \sum_{l=\nu_{i-1}+1}^{\infty} \mathbb{P}(\tau \geq l).$$

(ii) $$\sum_{t=S_i-\nu_{i-1}}^{S_{i,j}-1} \mathbb{E}[B_{i,t}|\mathcal{G}_{t-1}] \leq \sum_{l=\nu_{i-1}+1}^{\infty} \mathbb{P}(\tau \geq l) + \mu_{j_i'} \sum_{l=0}^{\nu_{i-1}} \mathbb{P}(\tau > l).$$

(iii) $$\sum_{t=S_{i,j}}^{U_{i,j}} \mathbb{E}[C_{i,t}|\mathcal{G}_{t-1}] = \mu_j \sum_{l=0}^{\nu_{i-1}} \mathbb{P}(\tau > l).$$

Proof. The proof is very similar to that of Lemma 5.27. We prove each statement individually.

Statement (i): This is similar to the proof of Lemma 5.27,

$$\begin{aligned}
\sum_{t=S_{i-1,j}}^{S_i-\nu_{i-1}-1} \mathbb{E}[A_{i,t}|\mathcal{G}_{t-1}] &\leq \sum_{t=S_{i-1,j}}^{S_i-\nu_{i-1}-1} \mathbb{E}[\mathbb{I}\{\tau_{t,J_t}+t \geq S_i\}|\mathcal{G}_{t-1}] \\
&= \sum_{s=0}^{\infty}\sum_{s'=s}^{\infty} \mathbb{I}\{S_{i-1,j}=s, S_i=s'\} \sum_{t=s}^{s'-\nu_{i-1}-1} \mathbb{E}[\mathbb{I}\{\tau_{t,J_t}+t \geq S_i\}|\mathcal{G}_{t-1}] \\
&= \sum_{s=0}^{\infty}\sum_{s'=s}^{\infty} \sum_{t=s}^{s'-\nu_{i-1}-1} \mathbb{E}[\mathbb{I}\{S_{i-1,j}=s, S_i=s', \tau_{t,J_t}+t \geq s'\}|\mathcal{G}_{t-1}] \\
&\qquad\qquad \text{(Since } \{S_i=s', S_{i-1,j}=s\} \in \mathcal{G}_{t-1} \text{ for } t \geq s) \\
&= \sum_{s=0}^{\infty}\sum_{s'=s}^{\infty} \mathbb{I}\{S_{i-1,j}=s, S_i=s'\} \sum_{t=s}^{s'-\nu_{i-1}-1} \mathbb{P}(\tau_{t,J_t}+t \geq s') \\
&\leq \sum_{s=0}^{\infty}\sum_{s'=s}^{\infty} \mathbb{I}\{S_{i-1,j}=s, S_i=s'\} \sum_{l=\nu_{i-1}+1}^{\infty} \mathbb{P}(\tau > l) \\
&= \sum_{l=\nu_{i-1}+1}^{\infty} \mathbb{P}(\tau > l).
\end{aligned}$$

Statement (ii): For statement (ii), we have that for $(i,j) \neq (1,1)$,

$$\sum_{t=S_i-\nu_{i-1}}^{S_{i,j}-1} \mathbb{E}[B_{i,t}|\mathcal{G}_{t-1}] = \sum_{t=S_i-\nu_{i-1}}^{S_{i,j}-\nu_{i-1}-2} \mathbb{E}[B_{i,t}|\mathcal{G}_{t-1}] + \sum_{t=S_{i,j}-\nu_{i-1}-1}^{S_{i,j}-1} \mathbb{E}[B_{i,t}|\mathcal{G}_{t-1}].$$

Then, since$\{S_{i,j}=s'\} \cap \{S_i-\nu_{i-1} \leq t\} \in \mathcal{G}_{t-1}$ so we can use the same technique as for statement (i) to bound the first term. For the second term, since we will be

playing only arm j_i' for $S_{i,j} - \nu_{i-1} - 1, \ldots, S_{i,j} - 1$, so,

$$\begin{aligned}
\sum_{t=S_{i,j}-\nu_{i-1}-1}^{S_{i,j}-1} \mathbb{E}[B_{i,t}|\mathcal{G}_{t-1}] &= \sum_{t=S_{i,j}-\nu_{i-1}-1}^{S_{i,j}-1} \mathbb{E}[R_{t,J_t}\mathbb{I}\{\tau_{t,J_t}+t \geq S_{i,j}\}|\mathcal{G}_{t-1}] \\
&= \sum_{s=0}^{\infty} \mathbb{I}\{S_{i,j}=s\} \sum_{t=s-\nu_{i-1}-1}^{s-1} \mathbb{E}[R_{t,J_t}\mathbb{I}\{\tau_{t,J_t}+t \geq S_{i,j}\}|\mathcal{G}_{t-1}] \\
&= \sum_{s=0}^{\infty} \sum_{t=s-\nu_{i-1}-1}^{s-1} \mathbb{E}[R_{t,J_t}\mathbb{I}\{S_{i,j}=s, \tau_{t,J_t}+t \geq S_{i,j}\}|\mathcal{G}_{t-1}] \\
&\qquad\qquad \text{(Since } \{S_{i,j}=s', S_{i,j}-\nu_{i-1} \leq t\} \in \mathcal{G}_{t-1} \text{)} \\
&= \sum_{s=0}^{\infty} \sum_{t=s-\nu_{i-1}-1}^{s-1} \mathbb{E}[R_{t,J_t}\mathbb{I}\{S_{i,j}=s, \tau_{t,J_t}+t \geq s\}|\mathcal{G}_{t-1}] \\
&= \sum_{s=0}^{\infty} \mathbb{I}\{S_{i,j}=s\} \sum_{t=s-\nu_{i-1}-1}^{s-1} \mu_{j_i'}\mathbb{P}(\tau_{t,J_t}+t \geq s)
\end{aligned}$$

(Since $\{S_{i,j}=s\} \in \mathcal{G}_{t-1}$ for $t \geq s-\nu_{i-1}-1$ and given $\mathcal{G}_{t-1}$, R_{t,J_t} and τ_{t,J_t} are independent)

$$\begin{aligned}
&= \sum_{s=0}^{\infty} \mathbb{I}\{S_{i,j}=s\}\mu_{j_i'} \sum_{l=0}^{\nu_{i-1}} \mathbb{P}(\tau > l) \\
&= \mu_{j_i'} \sum_{l=0}^{\nu_{i-1}} \mathbb{P}(\tau > l).
\end{aligned}$$

Then, for $(i,j) = (1,1)$, the amount seeping in will be 0, so using $\nu_0 = 0, \mu_{1_1}' = 0$, the result trivially holds. Hence,

$$\sum_{t=S_i-\nu_{i-1}}^{S_{i,j}-1} \mathbb{E}[B_{i,t}|\mathcal{G}_{t-1}] \leq \sum_{l=\nu_{i-1}+1}^{\infty} \mathbb{P}(\tau \geq l) + \mu_{j_i'} \sum_{l=0}^{\nu_{i-1}} \mathbb{P}(\tau > l).$$

Statement (iii): This is the same as in Lemma 5.16. □

We now bound each term of the decomposition in (5.23).

Bounding Term I.: For Term I., we can again use Lemma 5.17 as in the proof of Lemma 5.1 to get that with probability greater than $1 - \frac{1}{T\tilde{\Delta}_m^2}$,

$$\sum_{i=1}^{m} \sum_{t=S_{i,j}}^{U_{i,j}} (R_{t,J_t} - \mu_j) \leq \sqrt{\frac{n_m \log(T\tilde{\Delta}_m^2)}{2}}.$$

Bounding Term II.: For Term II., we will use Freedmans inequality (Theorem 5.10). From Lemma 5.26, $\{Y_s\}_{s=0}^{\infty}$ with $Y_s = \sum_{t=1}^{s}(Q_t - \mathbb{E}[Q_t|\mathcal{G}_{t-1}])$ is a martingale with respect to $\{\mathcal{G}_s\}_{s=0}^{\infty}$ with increments $\{Z_s\}_{s=0}^{\infty}$ satisfying $\mathbb{E}[Z_s|\mathcal{G}_{s-1}] = 0$ and $Z_s \leq 1$ for all s. Further, by Lemma 5.27, $\sum_{t=1}^{s} \mathbb{E}[Z_t^2|\mathcal{G}_{t-1}] \leq m\mathbb{E}[\tau] + m\mathbb{V}(\tau) \leq \frac{4\times 2^m}{8}(\mathbb{E}[\tau] + \mathbb{V}(\tau)) \leq n_m/8$ with probability 1. Hence we can apply Freedman's inequality to get that with probability greater than $1 - \frac{1}{T\tilde{\Delta}_m^2}$,

$$\sum_{t=1}^{S_{m,j}}(Q_t - \mathbb{E}[Q_t|\mathcal{G}_{t-1}]) = \sum_{s=1}^{\infty} \mathbb{I}\{S_{m,j} = s\} \times Y_s \leq \frac{2}{3}\log(T\tilde{\Delta}_m^2) + \sqrt{\frac{1}{8}n_m \log(T\tilde{\Delta}_m^2)},$$

using that Freedman's inequality applies simultaneously to all $s \geq 1$.

Bounding Term III.: For Term III., we again use Freedman's inequality (Theorem 5.10), using Lemma 5.14 to show that $\{Y'_s\}_{s=0}^{\infty}$ with $Y'_s = \sum_{t=1}^{s}(\mathbb{E}[P_t|\mathcal{G}_{t-1}] - P_t)$ is a martingale with respect to $\{\mathcal{G}_s\}_{s=0}^{\infty}$ with increments $\{Z'_s\}_{s=0}^{\infty}$ satisfying $\mathbb{E}[Z'_s|\mathcal{G}_{s-1}] = 0$ and $Z'_s \leq 1$ for all s. Further, by Lemma 5.15, $\sum_{t=1}^{s} \mathbb{E}[Z_t^2|\mathcal{G}_{t-1}] \leq m\mathbb{E}[\tau] \leq n_m/8$ with probability 1. Hence, with probability greater than $1 - \frac{1}{T\tilde{\Delta}_m^2}$,

$$\sum_{t=1}^{U_{m,j}}(\mathbb{E}[P_t|\mathcal{G}_{t-1}] - P_t) = \sum_{s=1}^{\infty} \mathbb{I}\{U_{m,j} = s\} \times Y'_s \leq \frac{2}{3}\log(T\tilde{\Delta}_m^2) + \sqrt{\frac{1}{8}n_m \log(T\tilde{\Delta}_m^2)}.$$

Bounding Term IV.: To begin with, observe that,

$$
\begin{aligned}
&\sum_{t=1}^{S_{m,j}} \mathbb{E}[Q_t|\mathcal{G}_{t-1}] - \sum_{t=1}^{U_{m,j}} \mathbb{E}[P_t|\mathcal{G}_{t-1}] \\
&= \sum_{t=1}^{S_{m,j}} \mathbb{E}\Bigg[\sum_{i=1}^{m} (A_{i,t}\mathbb{I}\{S_{i-1,j} \le t \le S_i - \nu_{i-1} - 1\} + B_{i,t}\mathbb{I}\{S_i - \nu_{i-1} \le t \le S_{i,j} - 1\}) \Bigg| \mathcal{G}_{t-1} \Bigg] \\
&\qquad - \sum_{t=1}^{U_{m,j}} \mathbb{E}\Bigg[\sum_{i=1}^{m} C_{i,t}\mathbb{I}\{S_{i,j} \le t \le U_{i,j}\} \Bigg| \mathcal{G}_{t-1} \Bigg] \\
&= \sum_{i=1}^{m} \sum_{t=1}^{S_{m,j}} \mathbb{E}[A_{i,t}\mathbb{I}\{S_{i-1,j} \le t \le S_i - \nu_{i-1} - 1\}|\mathcal{G}_{t-1}] \\
&\qquad + \sum_{i=1}^{m} \sum_{t=1}^{S_{m,j}} \mathbb{E}[B_{i,t}\mathbb{I}\{S_i - \nu_{i-1} \le t \le S_{i,j} - 1\}|\mathcal{G}_{t-1}] \\
&\qquad - \sum_{i=1}^{m} \sum_{t=1}^{U_{m,j}} \mathbb{E}[C_{i,t}\mathbb{I}\{S_{i,j} \le t \le U_{i,j}\}|\mathcal{G}_{t-1}] \\
&= \sum_{i=1}^{m} \Bigg(\sum_{t=S_{i-1,j}}^{S_i - \nu_{i-1} - 1} \mathbb{E}[A_{i,t}|\mathcal{G}_{t-1}] + \sum_{t=S_i - \nu_{i-1}}^{S_{i,j}-1} \mathbb{E}[B_{i,t}|\mathcal{G}_{t-1}] - \sum_{t=S_{i,j}}^{U_{i,j}} \mathbb{E}[C_{i,t}|\mathcal{G}_{t-1}] \Bigg) \\
&\qquad\qquad \text{(using that the indicators are } \mathcal{G}_{t-1}\text{-measurable)} \\
&\le \sum_{i=1}^{m} \Bigg(\sum_{l=\nu_{i-1}+1}^{\infty} \mathbb{P}(\tau \ge l) + \mu_{j_i'} \sum_{l=0}^{\nu_{i-1}} \mathbb{P}(\tau > l) - \mu_j \sum_{l=0}^{\nu_i} \mathbb{P}(\tau > l) \Bigg), \\
&\le \sum_{i=1}^{m} \Bigg(\frac{2\mathbb{V}(\tau)}{\nu_{i-1} - \mathbb{E}[\tau]} + (\mu_{j_i'} - \mu_j) \sum_{l=0}^{\nu_i} \mathbb{P}(\tau > l) \Bigg), \\
&\le \sum_{i=1}^{m} \Bigg(\frac{2\mathbb{V}(\tau)}{2^{i-1}} + (\mu_{j_i'} - \mu_j) \sum_{l=0}^{\nu_i} \mathbb{P}(\tau > l) \Bigg), \qquad (5.24)
\end{aligned}
$$

by Lemma 5.28 and Lemma 5.25 where we have used the fact that since $n_m \le T$, the maximal number of rounds of the algorithm is $\frac{1}{2}\log_2(T/4)$ and for $m \le \frac{1}{2}\log_2(T/4)$, $\frac{\log(T\tilde{\Delta}_m^2)}{\tilde{\Delta}_m^2} \ge \frac{2\log(T\tilde{\Delta}_{m-1}^2)}{\tilde{\Delta}_{m-1}^2}$ so $n_m \ge 2n_{m-1}$ and $\nu_m \ge n_{m-1}$. Then for $\mathbb{E}[\tau] \ge 1$, $\nu_{i-1} - \mathbb{E}[\tau] \ge 2/\tilde{\Delta}_{i-1}\mathbb{E}[\tau] - \mathbb{E}[\tau] \ge (2 \times 2^{i-1} - 1)\mathbb{E}[\tau] \ge 2^{i-1}\mathbb{E}[\tau] \ge 2^{i-1}$ and for $\mathbb{E}[\tau] \le 1$, $\nu_{i-1} - \mathbb{E}[\tau] \ge \nu_{i-1} - 1 \ge 2\log(4)/\tilde{\Delta}_{i-1} - 1 \ge 2^{i-1}$ so $\nu_{i-1} - \mathbb{E}[\tau] \ge 2^{i-1}$. Then, the probability that either arm j_i' or j is active in phase i when it should have been eliminated in or before phase $i-1$ is less than $2p_{i-1}$, where p_i is the probability that

the confidence bounds for one arm holds in phase i and $p_0 = 0$. If neither arm should have been eliminated by phase i, this means that their mean rewards are within $\tilde{\Delta}_{i-1}$ of each other. Hence, with probability greater than $1 - 2p_{i-1}$,

$$\mu_{j_i'} \sum_{l=0}^{\nu_i} \mathbb{P}(\tau > l) - \mu_j \sum_{l=0}^{\nu_i} \mathbb{P}(\tau > l) \leq \tilde{\Delta}_{i-1} \sum_{l=0}^{\nu_i} \mathbb{P}(\tau > l) \leq \tilde{\Delta}_{i-1} \mathbb{E}[\tau].$$

Then, summing over all phases gives that with probability greater than $1 - 2\sum_{i=0}^{m-1} p_i$,

$$\begin{aligned}\sum_{t=1}^{S_{m,j}} \mathbb{E}[Q_t | \mathcal{G}_{t-1}] - \sum_{t=1}^{U_{m,j}} \mathbb{E}[P_t | \mathcal{G}_{t-1}] &\leq 2\mathbb{V}(\tau) \sum_{i=1}^{m} \frac{1}{2^{i-1}} + \mathbb{E}[\tau] \sum_{i=1}^{m} \tilde{\Delta}_{i-1} \\ &= (2\mathbb{V}(\tau) + \mathbb{E}[\tau]) \sum_{i=0}^{m-1} \frac{1}{2^i} \leq 4\mathbb{V}(\tau) + 2\mathbb{E}[\tau].\end{aligned}$$

Combining all terms: To get the final high probability bound, we sum the bounds for each term I.-IV.. Then, with probability greater than $1 - (\frac{3}{T\tilde{\Delta}_m^2} + 2\sum_{i=1}^{m-1} p_i)$, either $j \notin \mathcal{A}_m$ or arm j is played n_m times by the end of phase m and

$$\begin{aligned}\frac{1}{n_m} \sum_{t \in T_j(m)} (X_t - \mu_j) &\leq \frac{4\log(T\tilde{\Delta}_m^2)}{3n_m} + \left(\frac{2}{\sqrt{8}} + \frac{1}{\sqrt{2}}\right)\sqrt{\frac{\log(T\tilde{\Delta}_m^2)}{n_m}} + \frac{2\mathbb{E}[\tau] + 4\mathbb{V}(\tau)}{n_m} \\ &\leq \frac{4\log(T\tilde{\Delta}_m^2)}{3n_m} + \sqrt{\frac{2\log(T\tilde{\Delta}_m^2)}{n_m}} + \frac{2\mathbb{E}[\tau] + 4\mathbb{V}(\tau)}{n_m} = w_m.\end{aligned}$$

Using the fact that $p_0 = 0$ and substituting the other p_i's using the same recursive relationship $p_i = \frac{3}{T\tilde{\Delta}_i^2} + 2\sum_{l=1}^{i-1} p_l$ as in the case for bounded delays (see the proof of Lemma 5.5) gives, $p_m = \frac{12}{T\tilde{\Delta}_m^2}$ so the above bound holds with probability greater than $1 - \frac{12}{T\tilde{\Delta}_m^2}$.

Defining $\mathbf{n_m}$: Setting

$$n_m = \left\lceil \frac{1}{\tilde{\Delta}_m^2} \left(\sqrt{2\log(T\tilde{\Delta}_m^2)} + \sqrt{2\log(T\tilde{\Delta}_m^2) + \frac{8}{3}\tilde{\Delta}_m \log(T\tilde{\Delta}_m^2) + 4\tilde{\Delta}_m(\mathbb{E}[\tau] + 2\mathbb{V}(\tau))} \right)^2 \right\rceil. \quad (5.25)$$

ensures that $w_m \leq \frac{\tilde{\Delta}_m}{2}$ which concludes the proof. □

Remark: Note that if $\mathbb{E}[\tau] \geq 1$, then the confidence bounds can be tightened by replacing (5.24) with

$$\sum_{i=1}^{m} \left(\frac{2\mathbb{V}(\tau)}{2^{i-1}\mathbb{E}[\tau]} + (\mu_{j_i'} - \mu_j) \sum_{l=0}^{\nu_i} \mathbb{P}(\tau > l) \right)$$

This is obtained by noting that for $\mathbb{E}[\tau] \geq 1$. $\nu_{i-1} - \mathbb{E}[\tau] \geq 2/\tilde{\Delta}_{i-1}\mathbb{E}[\tau] - \mathbb{E}[\tau] \geq (2 \times 2^{i-1} - 1)\mathbb{E}[\tau] \geq 2^{i-1}\mathbb{E}[\tau]$. This leads to replacing the $\mathbb{V}(\tau)$ term in the definition of n_m by $\mathbb{V}(\tau)/\mathbb{E}[\tau]$.

5.D.2 Regret Bounds

Theorem 5.8. *Under Assumption 1 and Assumption 3 of known (bound on) the expectation and variance of the delay, and choice of n_m from (5.7), the expected regret of Algorithm 5.1 can be upper bounded by,*

$$\mathbb{E}[\mathfrak{R}_T] \leq \sum_{j=1:\mu_j \neq \mu^*}^{K} O\left(\frac{\log(T\Delta_j^2)}{\Delta_j} + \mathbb{E}[\tau] + \mathbb{V}(\tau) \right).$$

Proof. The proof is very similar to that of Theorem 5.2, however, for clarity, we repeat the main arguments here. For any sub-optimal arm j, define M_j to be the random variable representing the phase arm j is eliminated in and note that if M_j is finite, $j \in \mathcal{A}_{M_j}$ but $j \notin \mathcal{A}_{M_j+1}$. Then let m_j be the phase arm j *should* be eliminated in, that is $m_j = \min\{m | \tilde{\Delta}_m < \frac{\Delta_j}{2}\}$ and note that, from the new definition of $\tilde{\Delta}_m$ in our

algorithm, we get the relations

$$2^m = \frac{1}{\tilde{\Delta}_m}, \quad 2\tilde{\Delta}_{m_j} = \tilde{\Delta}_{m_j-1} \geq \frac{\Delta_j}{2} \quad \text{and so,} \quad \frac{\Delta_j}{4} \leq \tilde{\Delta}_{m_j} \leq \frac{\Delta_j}{2}. \tag{5.26}$$

Define $\mathfrak{R}_T^{(j)}$ to be the regret contribution from each arm $1 \leq j \leq K$ and let M^* be the round where the optimal arm $j^*$is eliminated. Hence,

$$\begin{aligned}
\mathbb{E}[\mathfrak{R}_T] &= \mathbb{E}\Big[\sum_{j=1}^{K} \mathfrak{R}_T^{(j)}\Big] = \mathbb{E}\Big[\sum_{m=0}^{\infty}\sum_{j=1}^{K} \mathfrak{R}_T^{(j)}\mathbb{I}\{M^* = m\}\Big] \\
&= \mathbb{E}\Big[\sum_{m=0}^{\infty}\sum_{j:m_j<m} \mathfrak{R}_T^{(j)}\mathbb{I}\{M^* = m\} + \sum_{j:m_j\geq m} \mathfrak{R}_T^{(j)}\mathbb{I}\{M^* = m\}\Big] \\
&= \underbrace{\mathbb{E}\Big[\sum_{m=0}^{\infty}\sum_{j:m_j<m} \mathfrak{R}_T^{(j)}\mathbb{I}\{M^* = m\}\Big]}_{\text{I.}} + \underbrace{\mathbb{E}\Big[\sum_{m=0}^{\infty}\sum_{j:m_j\geq m} \mathfrak{R}_T^{(j)}\mathbb{I}\{M^* = m\}\Big]}_{\text{II.}}
\end{aligned}$$

We will bound the regret in each of these cases in turn. First, however, we need the following results.

Lemma 5.29. *For any suboptimal arm j, if $j^* \in \mathcal{A}_{m_j}$, then the probability arm j is* not *eliminated by round m_j is,*

$$\mathbb{P}(M_j > m_j \text{ and } M^* \geq m_j) \leq \frac{24}{T\tilde{\Delta}_{m_j}^2}$$

Proof. The proof is exactly that of Lemma 5.18 but using Lemma 5.24 to bound the probability of the confidence bounds on either arm j or j^* failing. □

Define the event $F_j(m) = \{\bar{X}_{m,j^*} < \bar{X}_{m,j} - \tilde{\Delta}_m\} \cap \{j, j^* \in \mathcal{A}_m\}$ to be the event that arm j^* is eliminated by arm j in phase m. The probability of this event is bounded in the following lemma.

Lemma 5.30. *The probability that the optimal arm j^* is eliminated in round $m < \infty$*

by the suboptimal arm j is bounded by

$$\mathbb{P}(F_j(m)) \leq \frac{24}{T\tilde{\Delta}_m^2}$$

Proof. Again, the proof follows from Lemma 5.19 but using Lemma 5.24 to bound the probability of the confidence bounds failing. □

We now return to bounding the expected regret in each of the two cases.

Bounding Term I. As in the proof of Theorem 5.2, to bound the first term, we consider the cases where arm j is eliminated in or before the correct round ($M_j \leq m_j$) and where arm j is eliminated late ($M_j > m_j$). Then, using Lemma 5.29,

$$\mathbb{E}\Big[\sum_{m=0}^{\infty}\sum_{j:m_j<m}\mathfrak{R}_T^{(j)}\mathbb{I}\{M^*=m\}\Big] \leq \sum_{j=1}^{K}\left(2\Delta_j n_{m_j,j} + \frac{384}{\Delta_j}\right)$$

Bounding Term II For the second term, we again use the results from Theorem 5.2, but using Lemma 5.29 to bound the probability a suboptimal arm is eliminated in a later round and Lemma 5.30 to bound the probability j^* is eliminated by a suboptimal arm. Hence,

$$\mathbb{E}\Big[\sum_{m=0}^{\infty}\sum_{j:m_j\geq m}\mathfrak{R}_T^{(j)}\mathbb{I}\{M^*=m\}\Big] \leq \sum_{j=1}^{K}\frac{1920}{\Delta_j}.$$

Combining the regret from terms I and II gives,

$$\mathbb{E}[\mathfrak{R}_T] \leq \sum_{j=1}^{K}\left(\frac{1920}{\Delta_j} + 2\Delta_j n_{m_j,j}\right)$$

Hence, all that remains is to bound n_m in terms of Δ_j, T and $\mathbb{E}[\tau], \mathbb{V}(\tau)$. Using

$L_{m,T} = \log(T\tilde{\Delta}_m^2)$, we have that,

$$
\begin{aligned}
&n_{m_j,j} \\
&= \left\lceil \frac{1}{\tilde{\Delta}_m^2}\left(\sqrt{2\log(T\tilde{\Delta}_m^2)} + \sqrt{2\log(T\tilde{\Delta}_m^2) + \frac{8}{3}\tilde{\Delta}_m \log(T\tilde{\Delta}_m) + 4\tilde{\Delta}_m(\mathbb{E}[\tau] + 2\mathbb{V}(\tau))}\right)^2 \right\rceil \\
&\leq \left\lceil \frac{1}{\tilde{\Delta}_{m_j}^2}\left(8L_{m_j,T} + \frac{16}{3}\tilde{\Delta}_{m_j} L_{m_j,T} + 8\tilde{\Delta}_{m_j}\mathbb{E}[\tau] + 16\tilde{\Delta}_{m_j}\mathbb{V}(\tau)\right)\right\rceil \\
&\leq 1 + \frac{8L_{m_j,T}}{\tilde{\Delta}_{m_j}^2} + \frac{16L_{m_j,T}}{3\tilde{\Delta}_{m_j}} + \frac{8\mathbb{E}[\tau]}{\tilde{\Delta}_{m_j}} + \frac{16\mathbb{V}(\tau)}{\tilde{\Delta}_{m_j}} \\
&\leq 1 + \frac{128L_{m_j,T}}{\Delta_j^2} + \frac{32L_{m_j,T}}{\Delta_j} + \frac{32\mathbb{E}[\tau]}{\Delta_j} + \frac{64\mathbb{V}(\tau)}{\Delta_j}.
\end{aligned}
$$

where we have used $(a+b)^2 \leq 2(a^2+b^2)$ for $a, b \geq 0$.

Hence, the total expected regret from ODAAF with bounded delays can be bounded by,

$$\mathbb{E}[\mathfrak{R}_t] \leq \sum_{j=1}^{K}\left(\frac{256\log(T\Delta_j^2)}{\Delta_j} + 64\mathbb{E}[\tau] + 128\mathbb{V}(\tau) + \frac{1920}{\Delta_j} + 64\log(T) + 2\Delta_j\right).$$

□

Note that again, these constants can be improved at a cost of increasing $\log(T\Delta_j^2)$ to $\log(T\Delta_j)$. We now prove the problem independent regret bound.

Corollary 5.9. *For any problem instance satisfying Assumptions 1 and 3, the expected regret of Algorithm 5.1 satisfies*

$$\mathbb{E}[\mathfrak{R}_T] \leq O(\sqrt{KT\log(K)} + K\mathbb{E}[\tau] + K\mathbb{V}(\tau)).$$

Proof. Let $\lambda = \sqrt{\frac{K\log(K)e^2}{T}}$ and note that for $\Delta > \lambda$, $\log(T\Delta^2)/\Delta$ is decreasing in Δ.

Then, for constants $C_1, C_2 > 0$ we can bound the regret in the previous theorem by

$$\mathbb{E}[\mathfrak{R}_T] \leq \sum_{j:\Delta_j \leq \lambda} \mathbb{E}[\mathfrak{R}_t^{(j)}] + \sum_{j:\Delta_j > \lambda} \mathbb{E}[\mathfrak{R}_T^{(j)}] \leq \frac{KC_1 \log(T\lambda^2)}{\lambda} + KC_2(\mathbb{E}[\tau] + \mathbb{V}(\tau)) + T\lambda.$$

substituting in the above value of λ gives a worst case regret bound that scales with $O(\sqrt{KT\log(K)} + K(\mathbb{E}[\tau] + \mathbb{V}(\tau)))$. □

Remark: If $\mathbb{E}[\tau] \geq 1$, we can replace the $\mathbb{V}(\tau)$ terms in the regret bounds with $\mathbb{V}(\tau)/\mathbb{E}[\tau]$. This follows by using the alternative definition of n_m suggested in the remark at the end of Section 5.D.1.

5.E Additional Experimental Results

5.E.1 Increasing the Expected Delay

Here we investigate the effect of increasing the mean delay on both our algorithm and QPM-D (Joulani et al., 2013) and demonstrate that the regret of both algorithms increases linearly with $\mathbb{E}[\tau]$, as indicated by our theoretical results. We use the same experimental set up as described in Section 5.5. In Figure 5.5, we are interested in the impact of the mean delay on the regret so we kept the delay distribution family the same, using a $\mathcal{N}_+(\mu, 100)$ (Normal distribution with mean μ, variance 100, truncated at 0) as the delay distribution. We then ran the algorithms for increasing mean delays and plotted the ratio of the regret at T to the regret of the same algorithm when the delay distribution was $\mathcal{N}_+(0, 100)$. In this case, the regret was averaged over 1000 replications for ODAAF and ODAAF-V, and 5000 for QPM-D (this was necessary since the variance of the regret of QPM-D was significant). Here, it can be seen that increasing the mean delay causes the regret of all three algorithms to increase linearly. This is in accordance with the regret bounds which all include a linear factor

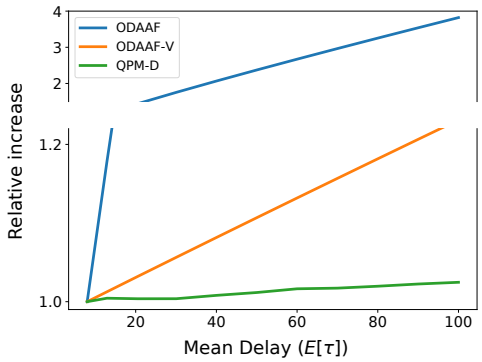
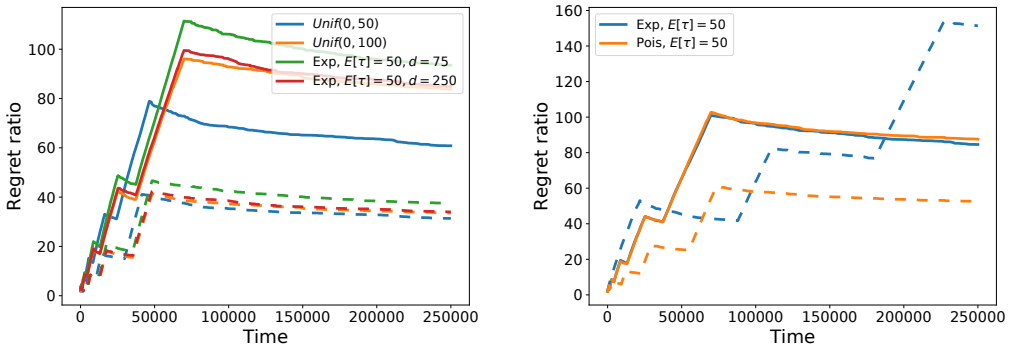


Figure 5.5: The relative increase in regret at horizon $T = 250000$ for increasing mean delay when the delay is $\mathcal{N}_+$ with variance 100.



(a) Bounded delays. Ratios of regret of ODAAF (solid lines) and ODAAF-B (dotted lines) to that of DUCB.

(b) Unbounded delays. Ratios of regret of ODAAF (solid lines) and ODAAF-V (dotted lines) to that of DUCB.

Figure 5.6: The ratios of regret of variants of our algorithm to that of DUCB for different delay distributions.

of $\mathbb{E}[\tau]$ (since here $\log(T)$ is kept constant). It can also be seen that ODAAF-V scales better with $\mathbb{E}[\tau]$ than ODAAF (for constant variance). Particularly, at $\mathbb{E}[\tau] = 100$, the relative increase in ODAAF-V is only 1.2 whereas that of ODAAF is 4 (QPM-D has the best relative increase of 1.05).

5.E.2 Comparison with Vernade et al. (2017)

Here we compare our algorithms, ODAAF, ODAAF-B and ODAAF-V, to the (non-censored) DUCB algorithm of Vernade et al. (2017). We use the same experimental setup as described in Section 5.5. As in the comparison to QPM-D, in Figure 5.6 we

plot the ratios of the cumulative regret of our algorithms to that of DUCB for different delay distributions. In Figure 5.6a, we consider bounded delay distributions and in Figure 5.6b, we consider unbounded delay distributions. From these plots, we observe that, as in the comparison to QPM-D in Figure 5.3, the regret ratios all converge to a constant. Thus we can conclude that the order of regret of our algorithms match that of DUCB, even though the DUCB algorithm of Vernade et al. (2017) has considerably more information about the delay distribution. In particular, along with knowledge on the individual rewards of each play (non-anonymous observations), DUCB also uses complete knowledge of the cdf of the delay distribution to re-weigh the average reward for each arm. Thus, our algorithms are able to match the rate of regret of Vernade et al. (2017) and QPM-D of Joulani et al. (2013) while just receiving aggregated, anonymous observations and using only knowledge of the expected delay rather than the entire cdf.

We ran the DUCB algorithm with parameter $\epsilon = 0$. As pointed out in (Vernade et al., 2017), the computational bottleneck in the DUCB algorithm is evaluating the cdf at all past plays of the arms in every round. For bounded delay distributions, this can be avoided using the fact that the cdf will be 1 for plays more than d steps ago. In the case of unbounded distributions, in order to make our experiments computationally feasible, we used the approximation $\mathbb{P}(\tau \leq d) = 1$ for $d \geq 200$. Another nuance of the DUCB algorithm is that in the early stages, the upper confidence bounds are dominated by the uncertainty terms, which themselves involve dividing by the cdf of the delay distributions. The arm that is played last in the initialization period will have the highest cdf and so it's confidence bound will be largest and DUCB will play this arm at time $K + 1$ (and possibly in subsequent rounds unless the cdf increases quickly enough). In order to overcome this, we randomize the order that we play the arms in during the initialization period in each replication of the experiment. Note that we did not run DUCB with half normal delays as DUCB divides by the cdf of

the delay distribution and in this case the cdf would be 0 at some points.

5.F Naive Approach for Bounded Delays

In this section we describe a naive approach to defining the confidence intervals when the delay is bounded by some $d \geq 0$ and show that this leads to sub-optimal regret. Let

$$w_m = \sqrt{\frac{\log(T\tilde{\Delta}_m^2)}{2n_m}} + \frac{md}{n_m}\,.$$

denote the width of the confidence intervals used in phase m for any arm j. We start by showing that the confidence bounds hold with high probability:

Lemma 5.31. *For any phase m and arm, j,*

$$\mathbb{P}(|\bar{X}_{m,j} - \mu_j| > w_m) \leq \frac{2}{T\tilde{\Delta}_m^2}\,.$$

Proof. First note that since the delay is bounded by d, at most d rewards from other arms can seep into phase i of playing arm j and at most d rewards from arm j can be lost. Defining $S_{i,j}$ and $U_{i,j}$ as the start and end points of playing arm j in phase i, respectively, we have

$$\left|\sum_{t=S_{i,j}}^{U_{i,j}} R_{j,t} - \sum_{t=S_{i,j}}^{U_{i,j}} X_t\right| \leq d\,, \tag{5.27}$$

because we can pair up some of the missing and extra rewards, and in each pair the difference is at most one. Then, by definition of $T_j(m)$ and using (5.27) we get

$$\frac{1}{n_m}\left|\sum_{t\in T_j(m)} R_{j,t} - \sum_{t\in T_j(m)} X_t\right| \leq \frac{md}{n_m}\,.$$

Define $\bar{R}_{m,j} = \frac{1}{|T_j(m)|}\sum_{t\in T_j(m)} R_{j,t}$ and recall that $\bar{X}_{m,j} = \frac{1}{|T_j(m)|}\sum_{t\in T_j(m)} X_t$. For any

$a > \frac{md}{n_m}$,

$$\mathbb{P}\left(|\bar{X}_{m,j} - \mu_j| > a\right) \leq \mathbb{P}\left(|\bar{X}_{m,j} - \bar{R}_{m,j}| + |\bar{R}_{m,j} - \mu_j| > a\right) \leq \mathbb{P}\left(|\bar{R}_{m,j} - \mu_j| > a - \frac{md}{n_m}\right)$$
$$\leq 2\exp\left\{-2n_m\left(a - \frac{md}{n_m}\right)^2\right\},$$

where the first inequality is from the triangle inequality and the last from Hoeffding's inequality since $R_{j,t} \in [0,1]$ are independent samples from ν_j, the reward distribution of arm j. In particular, taking $a = \sqrt{\frac{\log(T\tilde{\Delta}_m^2)}{2n_m}} + \frac{md}{n_m}$ guarantees that $\mathbb{P}\left(|\bar{X}_j - \mu_j| > a\right) \leq \frac{2}{T\tilde{\Delta}_m^2}$, finishing the proof. □

Observe that setting

$$n_m = \left\lceil \frac{1}{2\tilde{\Delta}_m^2}\left(\sqrt{\log(T\tilde{\Delta}_m^2)} + \sqrt{\log(T\tilde{\Delta}_m^2) + 4\tilde{\Delta}_m md}\right)^2 \right\rceil. \tag{5.28}$$

ensures that $w_m \leq \frac{\tilde{\Delta}_m}{2}$. We can substitute this value of n_m into Improved UCB and use the analysis from (Auer and Ortner, 2010) to get the following regret bound.

Theorem 5.32. *Assume there exists a bound $d \geq 0$ on the delay. Then for all $\lambda > 0$, the expected regret of the Improved UCB algorithm run with n_m defined as in (5.28) can be upper bounded by*

$$\sum_{\substack{j \in A \\ \Delta_j > \lambda}} \left(\Delta_j + \frac{64\log(T\Delta_j^2)}{\Delta_j} + 64\log(2/\Delta_j)d + \frac{96}{\Delta_j}\right) + \sum_{\substack{j \in A \\ 0 < \Delta_j < \lambda}} \frac{64}{\lambda} + T\max_{\substack{j \in A \\ \Delta_j \leq \lambda}} \Delta_j$$

Proof. The result follows from the proof of Theorem 3.1 in (Auer and Ortner, 2010) using the above definition of n_m. □

In particular, optimizing with respect to λ gives worst case regret of $O(\sqrt{KT\log K} + Kd\log T)$. This is a suboptimal dependence on the delay, particularly when $d >> \mathbb{E}[\tau]$.

Chapter 6

Recovering Bandits

6.1 Introduction

The multi-armed bandit problem has been introduced in Chapter 2. In its standard form, it consists of T rounds where, in each round $1 \leq t \leq T$, we play an arm J_t and receive a reward Y_t generated from the underlying reward distribution of the arm. The aim is to maximize the total reward over T rounds. Bandit algorithms have become ubiquitous in many settings such as web advertising and product recommendation. Consider, for example, suggesting items to a user on an internet shopping platform. This is typically modeled as a bandit problem where each product (or group of products) is an arm. Over time, the bandit algorithm will learn to suggest only good products to the user. In particular, once the algorithm learns that a product (eg. a television) has good reward, it will continue to suggest it to the user. However, if the user buys the television, the benefit of continuing to show them televisions is immediately diminished (but may increase again as the purchased television reaches the end of its lifetime). To improve customer experience (and profit), it would be beneficial for the recommendation algorithm to learn not to recommend the same product again immediately, but to wait an appropriate amount of time until the re-

ward from that product has 'recovered'. This sort of reward dynamic also occurs in other scenarios such as film and TV recommendation where a user may wish to wait before re-watching their favorite film, or conversely, may wish to continue watching a series but will lose interest in it if they haven't seen it recently. The recovering bandits framework presented here provides a natural extension of the stochastic bandit problem to capture these phenomena.

In the recovering bandits problem, we assume that the expected reward of each arm can be modeled as an (unknown) function of the number of rounds since it was last played. In particular, we assume that for each arm j, there is a function $f_j(z)$ that specifies the expected reward from playing arm j when it has not been played for z steps, and that this function is smooth enough to be modeled by a Gaussian process (GP) (see Figure 6.1). We take a Bayesian approach and further assume that the f_j's are sampled from a GP. For any time t, let $Z_{j,t}$ be the time since arm j was last played. At every time step, this changes for both the played arm (it resets to 0) and also for the unplayed arms (it increases by 1). Hence, the expected reward of every arm changes at every time step, and the magnitude of this change depends on which arm was played. This problem is therefore related to both the restless and rested bandits problems (Whittle, 1988).

A key feature of the recovering bandits problem is that the reward of each arm depends on the entire sequence of past actions we have taken. This means that, even when the reward functions are known, selecting the best sequence of T arms is intractable (since, in particular the state space of a MDP representation would be unacceptably large). One tractable alternative is to select the action that maximizes the *instantaneous* reward, without considering future decisions. This still poses quite a challenge compared to the standard K-armed bandit problem as instead of just learning the reward of each arm, we must learn an entire recovery function. In many cases, maximizing the instantaneous reward may not be optimal. Recall the earlier

internet shopping example. If a user has recently purchased a television, the expected reward of suggesting another one may be low, but it could still be higher than that of other products. Maximizing the instantaneous reward would mean suggesting the television. However, the total reward from showing the other products and waiting until the reward of the television recovers is greater. Thus, although it is infeasible to select a sequence of T arms, it is natural to consider selecting a sequence of $d \geq 1$ arms to maximize the reward in the next d plays.

In this chapter, we present and analyse two algorithms for the recovering bandits problem, one based on the Upper Confidence Bound (UCB) approach (Auer et al., 2002a), and one based on Thompson Sampling (Thompson, 1933). Both of these lookahead to select a good sequence of actions and achieve good regret guarantees and experimental performance. The chapter continues as follows. In Section 6.2 we discuss related work then formally define our problem in Section 6.3. In Section 6.4 we define our regret with respect to a d-step lookahead oracle. In Section 6.5, we briefly introduce a baseline algorithm. Then, in Section 6.6, we present our algorithms for recovering bandits and bound their regret. We discuss an optimistic planning approximation to improve computational complexity in Section 6.7, then demonstrate the empirical performance in Section 6.8 before concluding.

6.2 Related Work

In the restless bandits problem, the reward distribution of any arm can change at any time, regardless of whether it has been played. This problem has been studied by Whittle (1988); Slivkins and Upfal (2008); Garivier and Moulines (2011); Raj and Kalyani (2017); Besbes et al. (2014) and others (see Section 2.3.5 for more details). In the rested bandits problem, the reward distribution of an arm only changes when it is played. Recently, this has been applied to the problem of user fatigue in recommen-

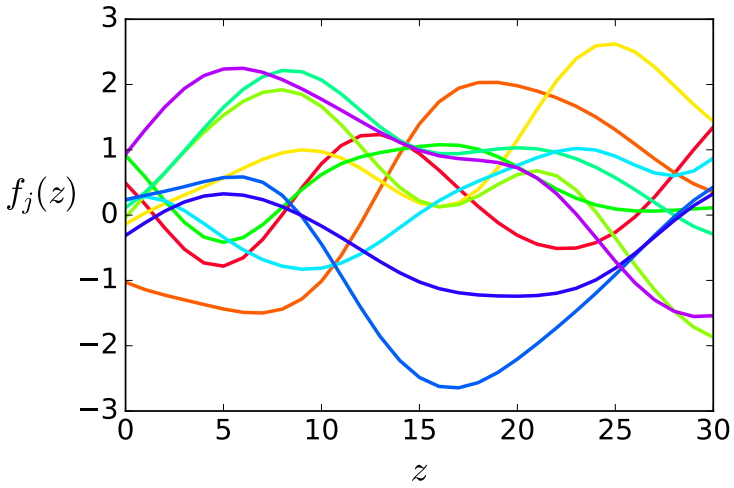


Figure 6.1: Examples of the recovery functions.

dation scenarios. Levine et al. (2017); Cortes et al. (2017); Bouneffouf and Feraud (2016); Heidari et al. (2016) study rested bandits problems with rewards that vary predominantly with the number of plays of an arm. More details on these approaches are given in Section 2.3.5.

In recovering bandits, the reward distributions change in every round and this change depends on whether the arm was played. Yi et al. (2017) incorporate inter-purchase times into recommendation systems by considering recovery functions that are known step functions. In the rogue bandits problem of Mintz et al. (2017), the expected reward of each arm depends on an underlying state (which could be the time since the arm was last played) via some parametric function. This is related to the recovering bandits problem. They use maximum likelihood estimation (although there are no guarantees the estimates will converge) and adapt the KL-UCB algorithm (Cappé et al., 2013) to this problem. The expected frequentist regret of their algorithm is bounded by $O(\sum_j {}^{\log(T)}/_{\delta_j^2})$ where δ_j depends on the random number of plays of each arm and the minimum distance between the rewards of any arms at any time. These δ_j's can get arbitrarily small so these bounds can be very poor. By the standard worst case analysis, the frequentist problem independent regret is $O^*(T^{2/3}K^{1/3})$, where we

use the notation O^* to suppress log factors. In comparison, our algorithms achieve $O^*(\sqrt{KT})$ Bayesian regret while requiring less knowledge of the recovery functions. Mintz et al. (2017) also provide an algorithm based on asymptotics which has no theoretical guarantees but improved experimental performance. In Section 6.8, we show that our algorithms outperform this algorithm experimentally.

The Gasussian process bandits problem was introduced in Section 2.3.3. In this problem, there is a single function, f, sampled from a GP and the aim is to minimize the (Bayesian) regret of the actions taken with respect to the maximum of f. The celebrated GP-UCB algorithm of Srinivas et al. (2010) has Bayesian regret $O^*(\sqrt{T\gamma_T})$ where γ_T is the 'maximal information gain' (see Section 6.6.4). Russo and Van Roy (2014) showed that a Thompson sampling algorithm for the GP bandits problem achieves the same Bayesian regret as GP-UCB. Bogunovic et al. (2016) considered the GP bandit problem with a slowly drifting reward function and Krause and Ong (2011) studied the contextual GP bandit problem. In both these problems, the contexts or drifts do not depend on the previous actions taken.

It is important to note that all of the above approaches only look at instantaneous regret whereas in recovering bandits, it is more appropriate to consider lookahead regret (see Section 6.4). We will also consider Bayesian regret.

6.3 Problem Definition

We have K independent arms and play the bandit game over T rounds (T is not necessarily known). For each arm $j \in A = \{1, \dots, K\}$ and round $t \in \{1, \dots, T\}$, denote by $Z_{j,t}$ the number of rounds since arm j was last played, where $Z_{j,t} \in \mathcal{Z} = \{0, \dots, z_{\max}\}$ for a finite $z_{\max} \in \mathbb{N}$ and $T \geq K|\mathcal{Z}|$. Note that $Z_{j,t}$ are random variables

since they depend on our past actions. If we play arm J_t at time t, then, at time $t+1$,

$$Z_{j,t+1} = \begin{cases} 0 & \text{if } J_t = j, \\ \min\{z_{\max}, Z_{j,t} + 1\} & \text{if } J_t \neq j. \end{cases} \tag{6.1}$$

Hence, if arm j has not been played for more than $z_{\max}$ steps, $Z_{j,t}$ will stay at $z_{\max}$. Only one arm is played at time t, so there is always one arm with $Z_{j,t} = 0$, and if $Z_{j,t} \neq z_{\max}$ then $Z_{j,t} \neq Z_{i,t}$ for $i \neq j$.

The expected reward for arm j is modeled by an (unknown) recovery function, f_j. We assume that the f_j's are sampled independently from a Gaussian processes with mean 0 and known kernel. Let $\mathbf{Z}_t = (Z_{1,t}, \ldots, Z_{K,t})$ be the vector of covariates for each arm at time t. At round t, we observe $\mathbf{Z}_t$ and use this and past observations to select an arm J_t to play. We then receive a noisy observation $Y_{J_t,t} = f_{J_t}(Z_{J_t,t}) + \epsilon_t$ where ϵ_t are iid $\mathcal{N}(0, \sigma^2)$ random variables and the standard deviation, σ, is known.

Gaussian Processes A brief introduction to Gaussian Processes (GP) is given in Appendix A.4 and more details can be found in (Rasmussen and Williams, 2006). A Gaussian process gives a distribution over functions, when for every finite set $z_1, \ldots, z_N$ of covariates, the distribution of $f(z_1), \ldots, f(z_N)$ is multivariate Gaussian. A GP is defined by its mean function, $\mu(z) = \mathbb{E}[f(z)]$, and kernel function, $k(z, z') = \mathbb{E}[(f(z) - \mu(z))(f(z') - \mu(z'))]$, which specifies the smoothness. If we observe $\mathbf{Y}_N = (Y_1, \ldots, Y_N)^T$ at covariates $\mathbf{z}_N = (z_1, \ldots, z_N)^T$ where $Y_n = f(z_n) + \epsilon_n$ and ϵ_n are iid $\mathcal{N}(0, \sigma^2)$ noise variables, then the posterior distribution after N observations is conjugate, and so is $\mathcal{GP}(\mu(z; N), k(z, z'; N))$. Where for $\mathbf{k}_N(z) = (k(z_1, z), \ldots, k(z_N, z))^T$ and positive semi-definite kernel matrix $\mathbf{K}_N = [k(z_i, z_j)]_{i,j=1}^N$, the posterior mean and

covariance are given by,

$$\mu(z; N) = \mathbf{k}_N(z)^T(\mathbf{K}_N + \sigma^2\mathbf{I})^{-1}\mathbf{y}_N,$$
$$k(z, z; N) = k(z, z') - \mathbf{k}_N(z)^T(\mathbf{K}_N + \sigma^2\mathbf{I})^{-1}\mathbf{k}_N(z'),$$

so $\sigma^2(z; N) = k(z, z; N)$. For any $z \in \mathcal{Z}$, the posterior distribution of $f(z)$ is then $\mathcal{N}(\mu(z; N), \sigma^2(z; N))$. We consider the posterior distribution of f_j for each arm at every round, when it has been played some (random) number of times. For each arm j, denote the posterior mean and variance of f_j at z after n plays of the arm by $\mu_j(z; n)$ and $\sigma_j^2(z; n)$. Let $N_j(t)$ be the (random) number of times arm j has been played up to time t. It will be convenient to denote the posterior mean and variance of arm j at round t of the algorithm by,

$$\mu_t(j) = \mu_j(Z_{j,t}; N_j(t-1)), \quad \text{and,} \quad \sigma_t^2(j) = \sigma_j^2(Z_{j,t}; N_j(t-1)).$$

6.4 Defining the Regret

We will measure the performance of our algorithm for the recovering bandits problem in terms of its Bayesian regret. The regret is typically defined as the cumulative difference in the expected reward of an algorithm and an oracle. In the Bayesian regret, the expectation is taken over the recovery curves as well as the actions. In recovering bandits, there are various choices for the oracle. We discuss some of these here before defining the d-step lookahead regret which will be used in the remainder of this chapter.

6.4.1 Full Horizon Regret

A natural candidate for the oracle is one which uses knowledge of the recovery functions to select the best sequence of T actions up to horizon T. However, computing

this policy is computationally infeasible even when the f_j's are known. Specifically, if we were to model this as a Markov decision process, the state space would have size $K^{|\mathcal{Z}|}$. This would make solution techniques such as dynamic programming impossible to apply to any realistically sized problem, especially since there are no discount factors. Furthermore, it would require that the horizon T is known, whereas we are interested in anytime algorithms which do not know T. For these reasons, we will not consider the full horizon regret.

6.4.2 Instantaneous Regret

Another candidate for the oracle is the policy which greedily plays the action corresponding to the highest immediate reward given the $\mathbf{Z}_t$ available at each time step t. These $\mathbf{Z}_t$ would depend on the actions previously taken by the oracle. Consider an alternative policy which plays this oracle up to time $s-1$, and then selects a different action at time s, and continues to play greedily. The cumulative reward of this alternative policy could be vastly different to that of the oracle since they may end up with very different $\mathbf{Z}$ values. Therefore, defining regret in relation to this oracle could penalize us severely for early mistakes. This is similar to the notion of Arora et al. (2012) that sub-linear policy regret (with respect to a sequence of actions) may not be achievable in adversarial bandits. Instead, one can define the regret of an algorithm π with respect to an oracle which selects the best action *at the* $\mathbf{Z}_t$*'s generated by* π. We will call this the *instantaneous regret*. This is the definition of regret in most non-stationary bandit problems and in (Mintz et al., 2017).

6.4.3 d-step Lookahead Regret

A policy achieving low instantaneous regret could be missing out on additional reward by not considering the impact of its actions on the future $\mathbf{Z}_t$'s. In particular, looking ahead and using knowledge of how the $Z_{j,t}$'s evolve can lead to choosing a good

sequence of arms which are collectively better than the individual greedy arms. For example, if there are two arms j_1, j_2 with similar $f_j(Z_{j,t})$ but if we don't play j_1 then its reward doubles, whereas the reward of j_2 stays the same, it is better to play j_2 first and wait for the reward of j_1 to increase. This leads us to consider oracles which take the current $\mathbf{Z}_t$ generated by our algorithm and select the best sequence of d actions for $d \geq 1$. We call the regret with respect to this oracle the *d-step lookahead regret*.

In order to formally define this regret, we model the problem of selecting a sequence of d actions as a decision tree. Here nodes correspond to $\mathbf{Z}$ values and edges represent playing arms and updating $\mathbf{Z}$ (see Figure 6.2). Each sequence of d actions is a leaf of this tree. Let $\mathcal{L}_d(\mathbf{Z})$ be the set of leaves of a d-step lookahead tree with root $\mathbf{Z}$. For any leaf $i \in \mathcal{L}_d(\mathbf{Z})$, denote by $M_i(\mathbf{Z})$ the expected reward at that leaf, that is the sum of the f_j's along the path to i at the corresponding Z_j values (see Section 6.6 for a full definition). The d-step lookahead regret is defined with respect to an oracle which knows the f_j's and, when given a root node $\mathbf{Z}_t$, selects the leaf with highest $M_i(\mathbf{Z}_t)$, denote this value by $M^*(\mathbf{Z}_t)$. This corresponds to selecting the best sequence of d arms from $\mathbf{Z}_t$. Let I_t be the leaf we select at time t. We play the arms to I_t for the next d steps so select a sequence of arms every d steps. The d-step lookahead regret is then,

$$\mathbb{E}[\mathfrak{R}_T^{(d)}] = \sum_{h=0}^{\lfloor T/d \rfloor} \mathbb{E}\Big[M^*(\mathbf{Z}_{hd+1}) - M_{I_{hd+1}}(\mathbf{Z}_{hd+1})\Big],$$

where the expectation is over both I_{hd+1} and f_j. The full horizon and instantaneous regret can be recovered from this by setting $d = T$ and $d = 1$, respectively. We consider two variants of this regret. In the single play regret, $\mathbb{E}[\mathfrak{R}_T^{(d,s)}]$, each arm can only be played a single time in the d-step lookahead (this can occur if there is a constraint on how often an arm can be played). In the multiple play regret, $\mathbb{E}[\mathfrak{R}_T^{(d,m)}]$, arms can be played multiple times in a lookahead.

Selecting d

For large d the optimal d-step lookahead policy will behave similarly to the (infeasible) full horizon oracle. Intuitively, if we look far enough ahead that we consider each arm at its maximal value, then the d-step lookahead oracle will be able to use knowledge of the peaks of the recovery curve of each arm to select a sequence of arms to play (this could be playing each arm at its maximal value or an alternative which gives higher reward). The challenge is how to select d to guarantee this occurs. However, observe that if $d \geq |\mathcal{Z}|$, looking d-steps ahead will guarantee we consider each arm at its maximum (since in the worst case each arm arrives at its optimal state after $|\mathcal{Z}| - 1$ steps).

In some cases, it may not be feasible to look $d \geq |\mathcal{Z}|$ steps ahead. In these cases, we can use the assumptions on the recovery functions to select d according to how often we expect to see near-optimal values of the recovery functions. For example, if the recovery functions are sampled from a GP whose kernel has lengthscale l (many kernels such as squared exponential and Matérn kernels satisfy this), then, on average, we will see a local maximum of each function every $2l$ steps (Murray, 2016; Rasmussen and Williams, 2006). Hence, looking $2l$ steps ahead means that, on average, we will consider a local maximum of each f_j.

6.5 Baseline Approach

We use an algorithm which has no information about the recovery structure as a baseline. For this, we model each (arm, z) pair as an arm. This reduces the problem to a standard multi-armed bandit problem with $K|\mathcal{Z}|$ arms, where only some arms are available each round. Using the UCB1 algorithm (Auer et al., 2002a) gives the following regret.

Theorem 6.1. *The instantaneous regret up to time T of the UCB1 algorithm with*

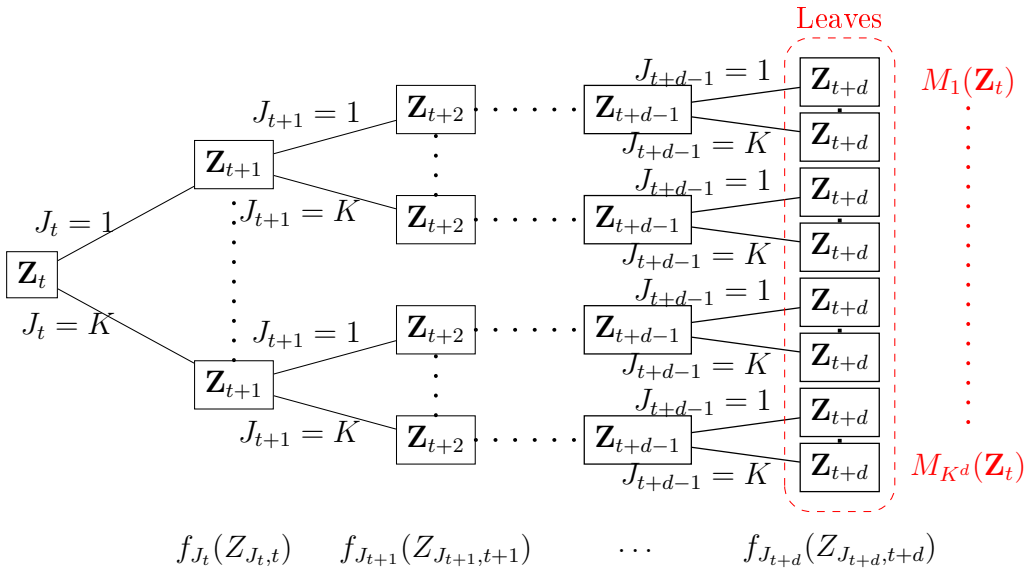


Figure 6.2: An example of a d-step lookahead tree.

$K|\mathcal{Z}|$ arms can be bounded by

$$\mathbb{E}[\mathfrak{R}_T^{(1)}] \leq O(\sqrt{K|\mathcal{Z}|T\log(T)} + K|\mathcal{Z}|^2)$$

See Section 6.D for details. This is as to be expected as there are now essentially $K|\mathcal{Z}|$ arms. The additional $K|\mathcal{Z}|^2$ term comes from having to wait for each arm to recover so it can be played at each $z \in \mathcal{Z}$ during initialization. A common baseline in non-stationary bandits is to use an algorithm for adversarial bandits on K arms. This would lead to poor results here since the aim in the adversarial bandits problem is to minimize the regret with respect to the best *constant* arm whereas in recovering bandits the regret is with respect to an optimal switching strategy.

6.6 Gaussian Process Recovery

In Algorithm 6.1 we present a UCB (dRGP-UCB) and Thompson Sampling (dRGP-TS) algorithm for the d-step lookahead recovering bandits problem. We present the algorithms here for both the single and multiple play case.

Our algorithms (described in Algorithm 6.1) proceed as follows. For each arm j, we place a prior GP distribution on f_j and initialize $Z_{j,1}$ (often this initial value is known, otherwise we set it to 0). Every d steps we construct the d-step lookahead tree as in Figure 6.2. At time t, we select a sequence of arms by choosing a leaf I_t of the tree with root node $\mathbf{Z}_t$. Each leaf represents a unique sequence of d arms at z values which have been updated using (6.1). For any leaf $i \in \mathcal{L}_d(\mathbf{Z}_t)$, define the total reward at i as $M_i(\mathbf{Z}_t)$,

$$M_i(\mathbf{Z}_t) = \sum_{\ell=0}^{d-1} f_{J_{t+\ell}}(Z_{J_{t+\ell},t+\ell})$$

where $\{J_{t+\ell}\}_{\ell=0}^{d-1}$ and$\{Z_{J_{t+\ell},t+\ell}\}_{\ell=0}^{d-1}$ are the sequences of arms and z's on the path to leaf i. Since the posterior distribution of each $f_j(z)$ at time t is Gaussian, for any node $i \in \mathcal{L}_d(\mathbf{Z}_t)$, $M_i(\mathbf{Z}_t) \sim \mathcal{N}(\eta_t(i), \varsigma_t^2(i))$ where,

$$\eta_t(i) = \sum_{\ell=0}^{d-1} \mu_t(J_{t+\ell}) \tag{6.2}$$

$$\text{and } \varsigma_t^2(i) = \sum_{\ell,q=0}^{d-1} \text{cov}_t(f_{J_{t+\ell}}(Z_{J_{t+\ell},t+\ell}), f_{J_{t+q}}(Z_{J_{t+q},t+q}))$$

for $\text{cov}_t(f_{J_{t+\ell}}(Z_{J_{t+\ell},t+\ell}), f_{J_{t+q}}(Z_{J_{t+q},t+q})) = \mathbb{I}\{J_{t+\ell} = J_{t+q}\}k_{J_{t+\ell}}(Z_{J_{t+\ell},t+\ell}, Z_{J_{t+q},t+q}; N_{J_{t+\ell}}(t))$.

For dRGP-UCB, we construct upper confidence bounds on each $M_i(\mathbf{Z}_t)$ using Gaussianity. We then select the leaf I_t with largest upper confidence bound at time t. That is,

$$I_t = \underset{1 \le i \le K^d}{\arg\max} \{\eta_t(i) + \alpha_t \varsigma_t(i)\}$$

$$\text{where } \alpha_t = \sqrt{2\log((K|\mathcal{Z}|)^d(t+d-1)^2)}. \tag{6.3}$$

In dRGP-TS, we select a sequence of d arms by sampling the recovery function of each arm j at $\mathbf{Z}_{j,t}^{(d)} = (Z_{j,t}, \ldots, Z_{j,t} + d - 1, 0, \ldots, d-1)^T$ and then calculating the sampled reward of each node using these sampled values. Denote the sampled reward

Algorithm 6.1 d-step lookahead UCB and Thompson Sampling

Input: α_t from (6.3) (for UCB).
Initialization: Define $\mathcal{T}_d = \{1, d+1, 2d+1, \dots\}$. For all arms $j \in A$, set $Z_{j,1} = 0$.

for $t \in \mathcal{T}_d$ **do**
Construct the d-step lookahead tree. Then,

$$\text{If UCB: } I_t = \underset{i \in \mathcal{L}_d(\mathbf{Z}_t)}{\arg\max} \left\{ \eta_t(i) + \alpha_t \varsigma_t(i) \right\}.$$

$$\text{If TS: } \begin{array}{l} \text{(i) } \forall j \in A, \text{ sample } \tilde{f}_j \text{ from the posterior at } \mathbf{Z}_{j,t}^{(d)}, \\ \text{(ii) } \forall i \in \mathcal{L}_d(\mathbf{Z}_t), \tilde{\eta}_t(i) = \sum_{l=0}^{d-1} \tilde{f}_{J_{t+\ell}}(Z_{J_{t+\ell},t+\ell}), \\ \text{(iii) } I_t = \arg\max_{i \in \mathcal{L}_d(\mathbf{Z}_t)} \{\tilde{\eta}_t(i)\}. \end{array}$$

for $\ell = 0, \dots, d-1$ **do**
Play ℓth arm to I_t, J_ℓ, and get reward $Y_{J_\ell,t+\ell}$.
Set $Z_{J_\ell,t+\ell+1} = 0$. For all $j \neq J_\ell$, set $Z_{j,t+\ell+1} = \min\{Z_{j,t+\ell} + 1, z_{\max}\}$.
end for
Update the posterior distributions of the played arms.
end for

of node i by $\tilde{\eta}_t(i)$. We choose the leaf I_t with highest $\tilde{\eta}_t(i)$.

In both dRGP-UCB and dRGP-TS, we play the sequence of d arms indicated by I_t over the next d time steps. We then update the posteriors and repeat this process.

We analyze the regret in the single and multiple play cases separately since in the multiple play case, we may loose information from not updating the posterior between plays of the same arm. The regret of our algorithms will depend on the kernel of the GP through the maximal information gain, as in (Srinivas et al., 2010). For a set $\mathcal{S}$ of covariates and observations $Y_\mathcal{S} = [f(z) + \epsilon_z]_{z \in \mathcal{S}}$, we define the *information gain*, $\mathcal{I}(Y_\mathcal{S}; f) = H(Y_\mathcal{S}) - H(Y_\mathcal{S}|f)$ where $H(\cdot)$ is the entropy. Intuitively, this is the increase in information about f after observing data $Y_\mathcal{S}$. As in (Srinivas et al., 2010), we express the information gain in terms of the posterior variances and bound it by

the *maximal information gain* from N samples, γ_N. If $z_t \in \mathcal{S}$ is played at time t,

$$\mathcal{I}(Y_{\mathcal{S}}, f) = \frac{1}{2}\sum_{t=1}^{|\mathcal{S}|} \log(1 + \sigma^{-2}\sigma^2(z_t; t-1)), \quad \text{and,} \quad \gamma_N = \max_{\mathcal{S}\subset\mathcal{Z}^N:|\mathcal{S}|=N} \mathcal{I}(Y_{\mathcal{S}}; f). \tag{6.4}$$

6.6.1 Single Play Lookahead Regret

In the single play case, each am can only be played once in the d-step lookahead. This simplifies the variance of the M_i's in (6.2) since the arms are independent. In this case, for any leaf i corresponding to playing arms $J_t, \ldots, J_{t+d-1}$ (at the corresponding z values), $\varsigma_t^2(i) = \sum_{\ell=0}^{d-1} \sigma_t^2(J_{t+\ell})$. This involves the posterior variances at time t. However, as we cannot repeat arms, if we play arm j at time $t+\ell$ for $0 \le \ell \le d-1$, it cannot have been played since time t so its posterior distribution is the same. By (6.4), we then relate the variances of $M_{I_t}(\mathbf{Z}_t)$ to the posterior variance of each arm when it was played, and hence to the information gain about the f_j's. We get the following regret bounds.

Theorem 6.2. *The d-step single play lookahead regret of dRGP-UCB satisfies,*

$$\mathbb{E}[\mathfrak{R}_T^{(d,s)}] \le O(\sqrt{KT\gamma_T \log(TK|\mathcal{Z}|)}).$$

Proof. The full proof is in Section 6.B.1. By normality, the confidence bounds fail with low probability. If the confidence bounds hold, our regret bound involves $\sum_{t\in\mathcal{T}_d} \varsigma_t(I_t) = \sum_{t=1}^{T} \sigma_t^2(J_t)$. We then relate this to the information gain about the f_j's. Dependence on d is avoided since we only use these confidence bounds every d steps. □

Theorem 6.3. *The d-step non-repeating lookahead regret of dRGP-TS satisfies,*

$$\mathbb{E}[\mathfrak{R}_T^{(d,s)}] \le O(\sqrt{KT\gamma_T \log(TK|\mathcal{Z}|)}).$$

Proof. See Section 6.C.1. The result follows by (Russo and Van Roy, 2014) and Theorem 6.2. □

6.6.2 Multiple Play Lookahead Regret

When arms can be played multiple times in the d-step lookahead, it is not as straightforward to relate $\varsigma_t^2(I_t)$ to the information gain about each f_j. In particular, $\varsigma_t^2(I_t)$ contains covariance terms and is defined using the posteriors at time t. On the other hand, γ_T is defined in terms of the posterior variances when each arm is played (which may be different to the posterior variance at time t if an arm is played multiple times in the lookahead). However, using the fact that the posterior covariance matrix of any arm is positive semi-definite, $2k_j(z_1, z_2; n) \leq \sigma_j^2(z_1; n) + \sigma_j^2(z_2; n)$, so we can bound $\varsigma_t^2(I_t) \leq 3\sum_{\ell=0}^{d-1} \sigma_t^2(J_{t+\ell})$. Then, the change in the posterior variance of a repeated arm can be bounded using the following lemma (whose proof is in Section 6.A).

Lemma 6.4. *For any $z \in \mathcal{Z}$, arm j and $n \in \mathbb{N}, n \geq 1$, let $Z_j^{(n)}$ be the z value at the nth play of arm j. Then, $\sigma_j^2(z; n-1) - \sigma_j^2(z; n) \leq \sigma^{-2}\sigma_j^2(Z_j^{(n)}; n-1)$.*

This leads to the following regret bounds for dRGP-UCB and dRGP-TS. Due to not updating the posterior between repeated plays of an arm, they both increase by a factor of $\sqrt{d}$ compared to the single play case.

Theorem 6.5. *The d-step multiple play lookahead regret of dRGP-UCB satisfies,*

$$\mathbb{E}[\mathfrak{R}_T^{(d,m)}] \leq O\left(\sqrt{KT\gamma_T \log((K|\mathcal{Z}|)^d T)}\right).$$

Proof. See Section 6.B.2. The regret is again bounded in terms of $\sum_{t\in\mathcal{T}_d} \varsigma_t(I_t)$. Using Lemma 6.4 we bound $\sum_{t\in\mathcal{T}_d} \varsigma_t^2(I_t)$ by $d\sum_{t=1}^{T} \sigma_t^2(J_t)$ and relate this to γ_T. □

Theorem 6.6. *The d-step multiple play lookahead regret of dRGP-TS satisfies,*

$$\mathbb{E}[\mathfrak{R}_T^{(d,m)}] \leq O\left(\sqrt{KT\gamma_T \log((K|\mathcal{Z}|)^d T)}\right).$$

Proof. See Section 6.C.2. We again use Theorem 6.5 and (Russo and Van Roy, 2014). □

6.6.3 Instantaneous Algorithm

If we set $d = 1$ in Algorithm 6.1, we obtain algorithms for minimizing the instantaneous regret. In this case, $\mathcal{T} = \{1, \ldots, T\}$ and there are K leaves of the 1-step lookahead tree, so each $M_i(\mathbf{Z}_t)$ corresponds to one arm. Hence, one arm is selected and played at each time step and $\eta_t(i) = \mu_t(j)$, $\varsigma_t^2(i) = \sigma_t^2(j)$ for some arm j. For the UCB approach, we define α_t as in (6.3) with $d = 1$. We bound the regret of 1RGP-TS and 1RGP-UCB in the following corollary,

Corollary 6.7. *The instantaneous regret of the* 1*RGP-UCB and* 1*RGP-TS algorithms up to horizon T satisfy*

$$\mathbb{E}[\mathfrak{R}_T^{(1)}] \leq O(\sqrt{KT\gamma_T \log(TK|\mathcal{Z}|)}).$$

Hence, the instantaneous regret of both algorithms is $O^*(\sqrt{KT\gamma_T})$ and by exploiting the GP structure, we have reduced the dependency on $|\mathcal{Z}|$ from $\sqrt{|\mathcal{Z}|}$ to $\sqrt{\log|\mathcal{Z}|}$ compared to the naive algorithm in Section 6.5.

6.6.4 Bounds on the Information Gain

Our regret bounds depend on the kernel of the recovery functions through the maximal information gain, γ_T. Theorem 5 of Srinivas et al. (2010) gives bounds on γ_T for some popular kernels. For linear and squared exponential kernels (with any length-

scale), $\gamma_T = O(\log(T))$ and for Matérn kernels with smoothness parameter ν and any lengthscale, $\gamma_T = O(T^{2/(2\nu+2)} \log(T))$. These can be placed into our regret bounds for recovering bandits.

6.7 Improving Computational Efficiency via Optimistic Planning

For large values of K and d, the proposed algorithm (Algorithm 6.1) may not be computationally efficient since it searches over K^d leaves. However, we can use ideas from optimistic planning (Hren and Munos, 2008; Munos et al., 2014) to improve the computational complexity of this tree search. This works particularly well for Thompson sampling and so we will focus on this case. We adapt the Thompson sampling procedure as follows. At time t, for all arms j, we sample $\tilde{f}_j(z)$ from the posterior distribution of f_j at $\mathbf{Z}_{j,t}^{(d)} = (Z_{j,t}, \ldots, Z_{j,t} + d, 0, \ldots d)^T$. Instead of searching the complete tree to find the sequence of arms with largest total $\tilde{f}_j(z)$'s (as in Algorithm 6.1), we iteratively build the tree, starting with the most promising sequences. It is known that in many settings this approach returns a good sequence even if the algorithm is stopped after only a limited number of evaluations (Hren and Munos, 2008; Munos et al., 2014).

We base our approach on optimistic planning for deterministic systems. The original approach in (Hren and Munos, 2008) uses discount factors and rewards bounded in $[0, 1]$. We adapt this to consider undiscounted rewards that are in the range $[\min_{j,z} \tilde{f}_j(z), \max_{j,z} \tilde{f}_j(z)]$. We start from an initial tree of just one node, $i_0 = \mathbf{Z}_t$. At step n of the optimistic planning procedure, let $\mathcal{T}_n$ be the expanded tree and let $\mathcal{S}_n$ be the set of nodes not in $\mathcal{T}_n$ but whose parents are in $\mathcal{T}_n$. We select a node in $\mathcal{S}_n$ to expand, and move it from $\mathcal{S}_n$ to $\mathcal{T}_n$, adding its children to $\mathcal{S}_n$. If we select a node i_n of depth d to expand, we stop the algorithm and output node i_n. Otherwise we

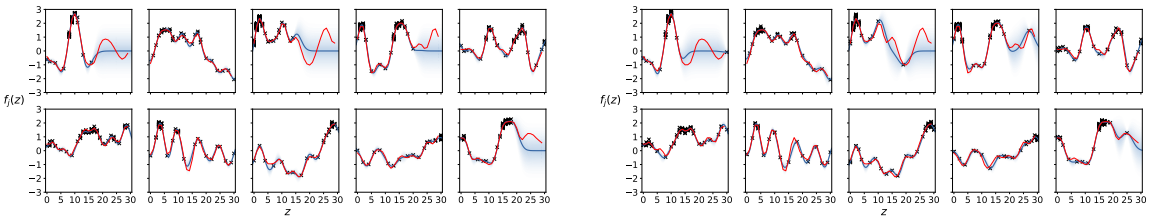
run the algorithm until we reach the computational limit (i.e. until $n = N$ for some predefined N). Let d_N be the maximal depth of any node in $\mathcal{T}_N$. We then output the node at depth d_N with the largest upper bound on the value of its continuation (i.e. with largest $b_N(i)$ defined in (6.5)).

The choice of which node to expand is made using upper bounds on the total value of a continuation of a sequence passing through each node. For node $i \in \mathcal{S}_n \cup \mathcal{T}_n$, let $u(i)$ denote the summed reward on the path to i (i.e. the sum of the corresponding $\tilde{f}_j(z)$'s) and define the value, $v(i)$, as the maximal reward of any continuation of the path to node i to depth d. Then, we define upper bounds on $v(i)$ as,

$$b_n(i) = u(i) + \Psi(\mathbf{z}(i), d - l(i)) \text{ for } i \in \mathcal{S}_n \tag{6.5}$$

where $l(i)$ is the depth of node i and, with some abuse of notation, $\mathbf{z}(i)$ is the vector of z_j's at node i. The function $\Psi(\mathbf{z}(i), d-l(i))$ provides an upper bound on the maximal reward of a sub-path from node i to a leaf. In the multiple play case, for every arm $j \in A$, $z \in \mathbf{Z}_{j,t}^{(d)}$, and $1 \leq l \leq d$, let $g_j(z, l) = \max\{\tilde{f}_j(z), \ldots, \tilde{f}_j(z+l), \tilde{f}_j(0), \ldots, \tilde{f}_j(l)\}$ be the maximal reward that can be gained from playing arm j in the next l steps. Then, $\Psi(\mathbf{z}(i), d-l(i)) = (d-l(i)) \max_{1 \leq j \leq K} g_j(z_j(i), d-l(i))$. In the single play case, we can get a tighter bound. Define $\Psi(\mathbf{z}(i), d-l(i)) = \max_{B \subseteq \mathcal{J}_i, |B| = d-l(i)} \sum_{j \in B} g_j(z_j(i), d-l(i))$ where $\mathcal{J}_i$ is the set of arms that have not been played on the path to node i. Note that in both cases, $\Psi(\mathbf{z}(i), 0) = 0$ for any $\mathbf{z}(i)$.

In some cases it is possible to bound the error resulting from this procedure. Let $v^* = \max_{i \in \mathcal{L}_d(\mathbf{Z})} v(i)$ be the value of the maximal node. The performance of the procedure depends on the number of near-optimal nodes. Let $p_l(\epsilon)$ be the proportion of ϵ-optimal nodes at depth l of the lookahead tree, where i is ϵ-optimal if $v^* - v(i) \leq \epsilon$. Also define $\Psi^*(l) = \max_{\mathbf{z} \in \mathcal{Z}} \Psi(\mathbf{z}, l)$ for any $l = 0, 1, \ldots, d$ and let $\Delta = \max_{j,z} \tilde{f}_j(z) - \min\{\min_{j,x} \tilde{f}_j(z), 0\}$. Then,



(a) Instantaneous: $d = 1$ (b) Lookahead: $d = 3$

Figure 6.3: The posterior mean (blue) of RGP-UCB with density given by the blue region for a squared exponential kernel with $l = 2$. The red curve is the true recovery curve and the crosses are our observed samples.

Proposition 6.8. *In the multiple play case, for the optimistic planning procedure with a budget of N samples, if the procedure is stopped at step $n < N$ because we selected a node i_n of depth d to expand, then $v^* - v(i_n) = 0$. Otherwise, if there exists some $\lambda \in (\frac{1}{K}, 1]$ and $d_0 \in \{1, \dots, d\}$ such that $\forall l \geq d_0$, $p_l((d-l)\Delta) \leq \lambda^l$, then for $N > n_0 = \frac{K^{d_0+1}-1}{K-1}$,*

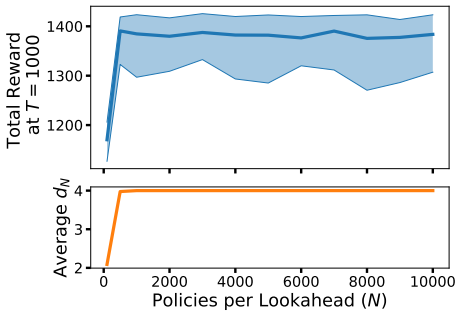
$$v^* - v(i_N) \leq \left(d - \frac{\log(N - n_0)}{\log(\lambda K)} - \frac{\log(\lambda K - 1)}{\log(\lambda K)} + 1 \right) \Delta. \tag{6.6}$$

Proof. See Section 6.E. □

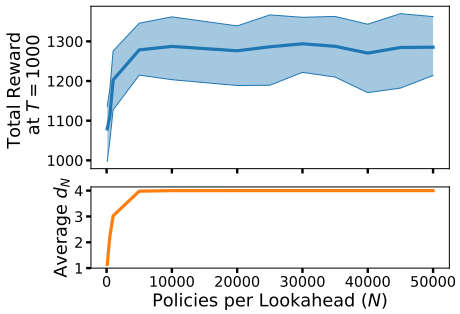
Hence, if we stop the procedure at $n < N$, the node i_n of depth d we return will be optimal. In many cases, especially for small λ (where there are not many near optimal policies), this will occur. Note that, in such cases, for very small λ, the bound in (6.6) can be weak. Otherwise, by (6.6), when we do not stop the procedure early, the sub-optimality of the returned node will depend on the proportion of other near-optimal nodes, λ, and the budget, N. Furthermore, by (6.6), for $N \approx (\lambda K)^d$, we can conclude that the returned node should be optimal.

Table 6.1: Total reward at $T = 1000$ for single step experiments with parametric functions

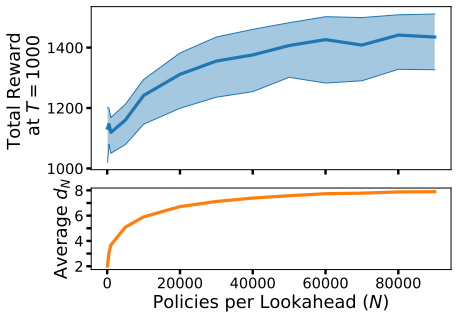
Setting	1RGP-UCB ($l = 5$)	1RGP-TS ($l = 5$)	RogueUCB-Tuned	UCB-Z
Logistic	461.7	462.6	446.2	242.6
	(454.3,468.9)	(455.7,469.3)	(438.2,453.5)	(229.6,256.0)
Gamma	145.6	156.5	132.7	116.8
	(139.6, 151.7)	(149.6,163.0)	(111.0,144.5)	(108.4,125.5)



(a) Lookahead: $d = 4$, arms: $K = 10$



(b) Lookahead: $d = 4$, arms: $K = 30$



(c) Lookahead: $d = 8$, arms: $K = 10$

Figure 6.4: The total reward and final depth of the lookahead tree, d_N, as the policy budget, N, increases.

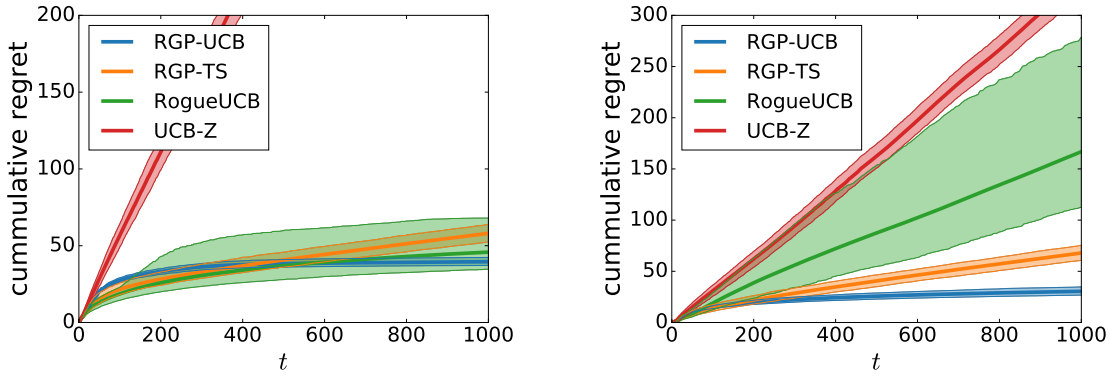
6.8 Experimental Results

We tested our algorithms in various experimental settings with $z_{\max} = 30$, noise standard deviation $\sigma = 0.1$, and horizon $T = 1000$. We used the GPy package (GPy, 2012) to fit the GPs. The first experiment aimed to check that our algorithms were playing arms at good z values (i.e. play arm j when $f_j(z)$ is high). For this, we

set $K = 10$ and sampled the recovery functions from a GP with squared exponential kernel and ran the algorithms once. Figure 6.3 illustrates that, for lengthscale $l = 2$, 1RGP-UCB and 3RGP-UCB both accurately estimate the recovery functions and learn to play each arm in the regions of $\mathcal{Z}$ where the reward is high. Although, as expected, 3RGP-UCB has more samples at the top of the peaks, it is reassuring that the instantaneous algorithm also plays in good regions. The same is true for dRGP-TS and different values of d and l (see Section 6.F.1).

In the second experiment, we tested the performance of the optimistic planning procedure within dRGP-TS. We averaged all results over 100 replications and used a squared exponential kernel with $l = 4$. In the first setting, $K = 10$ and $d = 4$, so the lookahead tree was relatively small and direct tree search would have been possible. Figure 6.4a shows that, when the bound on the number of policies the optimistic planning procedure can evaluate per lookahead, N, increases above 500, the total reward plateaus, and the average depth of the returned policy, d_N, is approximately 4. By Proposition 6.8, this means that we have found the same leaf of the lookahead tree as dRGP-TS, while evaluating significantly fewer policies. Next, we increased the number of arms to $K = 30$. Here, searching the whole lookahead tree would be computationally inefficient. However, Figure 6.4b shows that we found the optimal policy after searching about 20,000 policies (since here $d_N = d$), which is less than 0.1% of the total number of policies. In Figure 6.4c, we increased d, the depth of the lookahead policy. In this case, we needed to search more policies to find optimal leaves. However this was still less than 0.1% of the total number of policies. From Figure 6.4c, we also see that even when $d_N < d$, increasing N leads to higher reward.

Lastly, we compared our algorithms to RogueUCB-Tuned (Mintz et al., 2017) and the baseline from Section 6.5 (denoted UCB-Z) in two settings with parametric recovery functions. As in (Mintz et al., 2017), we only considered the instantaneous case ($d = 1$). We used squared exponential kernels in 1RGP-UCB and 1RGP-TS,



(a) Logistic setup

(b) Gamma setup

Figure 6.5: Cumulative instantaneous regret for parametric setup

with lengthscale $l = 5$ (results for other lengthscales are in Section 6.F.2). In the first experiment, the recovery function was a 3 parameter logistic function, $f(z) = \theta_0(1 + \exp\{-\theta_1(z - \theta_2)\})^{-1}$ which increases in z. In the second case, we used a modified gamma, $f(z) = \theta_0 C \exp\{-\theta_1 z\} z^{\theta_2}$ where C is a normalizer. This increases until a point and then decreases. The values of θ were sampled uniformly and are given in Section 6.F.2. We averaged the results over 500 replications. The cumulative regret (and confidence regions) in these experiments is shown in Figure 6.5 and the cumulative reward (and confidence bounds) in Table 6.1. Our algorithms achieve lower regret and higher reward than RogueUCB-Tuned. UCB-Z does badly here since the time required to play each (arm,z) combination once is greater than the horizon.

6.9 Conclusion

In recovering bandits, the expected reward of each arm is a function of the time since it was last played. Modeling this recovery curve as a Gaussian process, we presented UCB and Thompson sampling algorithms for this problem. These algorithms lookahead to find good sequences of arms. They achieve d-step lookahead Bayesian regret of $O^*(\sqrt{KdT})$ for linear and squared exponential kernels, and perform well experimentally. We also improved the computational efficiency using optimistic planning.

Future work would include extending this optimistic planning approximation to the UCB case (this is challenging since the UCBs cannot be decomposed by arm) and obtaining frequentist regret bounds for our algorithms.

6.A Preliminaries

Define the filtration $\{\mathcal{F}_t\}_{t=0}^{\infty}$ as $\mathcal{F}_0 = \emptyset$ and

$$\mathcal{F}_t = \sigma(J_1, \ldots, J_t, Y_1, \ldots, Y_t, \mathbf{Z}_1, \ldots, \mathbf{Z}_t) \tag{6.7}$$

where $\mathbf{Z}_t = [Z_{1,t}, \ldots, Z_{K,t}]$. It is important to note that $\mu_t(j), \sigma_t(j), J_t$ and $\mathbf{Z}_t$ are $\mathcal{F}_{t-1}$ measurable.

Recall that in both dRGP-UCB and dRGP-TS, we select a sequence of arms to play at time t by building a d-step lookahead tree with root $\mathbf{Z}_t$ and selecting the leaf node i with highest upper confidence bound on M_i, the cumulative reward from playing all arms in that policy,

$$M_i(\mathbf{Z}_t) = \sum_{\ell=0}^{d-1} f_{J_{t+\ell}}(Z_{J_{t+\ell},t+\ell})$$

where $\{J_{t+\ell}\}_{\ell=0}^{d-1}$ are the sequence of arms played on the path to leaf i and $\{Z_{J_{t+\ell},t+\ell}\}_{\ell=0}^{d-1}$ the corresponding z values. Denote the posterior mean and variance of $M_i(\mathbf{Z}_t)$ at time t as $\eta_t(i)$ and $\varsigma_t(i)$, then, conditional on the history $\mathcal{F}_{t-1}$, $M_i(\mathbf{Z}_t) \sim \mathcal{N}(\eta_t(i), \varsigma_t^2(i))$. When each arm can be played multiple times, there are interaction terms in the variance of the $M_i(\mathbf{Z}_t)$'s and thus we suffer some additional cost for not updating after every play. For each leaf node i, we can calculate

$$\varsigma_t^2(i) = \sum_{\ell=0}^{d-1} \sigma_t^2(J_{t+\ell}) + \sum_{\ell \neq q; \ell, q=0}^{d-1} \operatorname{cov}_t(f_{J_{t+\ell}}(Z_{J_{t+\ell},t+\ell}), f_{J_{t+q}}(Z_{J_{t+q},t+q}))$$

where $\text{cov}_t(f_{J_{t+\ell}}(Z_{J_{t+\ell},t+\ell}), f_{J_{t+q}}(Z_{J_{t+q},t+q}))$ is 0 if $J_{t+\ell} \neq J_{t+q}$ and $k_{J_{t+\ell}}(Z_{J_{t+\ell},t+\ell}, Z_{J_{t+q}t+q}; N_{J_{t+\ell}}(t-1))$ for $J_{t+\ell} = J_{t+q}$. Note that throughout, we assume that the variances and covariances are calculated at the $Z_{j,t}$'s where the arms are played, ie. $\sigma_t^2(J_{t+\ell}) = \sigma^2_{J_{t+\ell}}(Z_{J_{t+\ell},t+\ell}; N_{J_{t+\ell}}(t-1))$.

Before providing the proofs of the regret bounds, we need the following lemmas,

Lemma 6.9.

$$\sum_{t=1}^{T}\sum_{j=1}^{K} \sigma_t^2(J_t)\mathbb{I}\{J_t = j\} \leq C_1 K \gamma_T.$$

where $C_1 = 1/\log(1+\sigma^{-2})$.

Proof. Using the results of Lemma 5.4 of Srinivas et al. (2010) and the fact that the maximal information gain is increasing in the number of data points, it follows that

$$\begin{aligned}\sum_{t=1}^{T}\sum_{j=1}^{K} \sigma_t^2(J_t)\mathbb{I}\{J_t = j\} &= \sum_{j=1}^{K}\sum_{n=1}^{N_j(T)} \sigma_j^2(Z_j^{(n)}; n-1) \\ &\leq T\sum_{j=1}^{K} C_1 I(\mathbf{y}_{j,N_j(T)}; \mathbf{f}_{j,N_j(T)}) \leq C_1 \sum_{j=1}^{K} \gamma_{N_j(T)} \leq C_1 K \gamma_T.\end{aligned}$$

□

The following lemmas bound the amount of information we loose by only updating the posterior every d steps in the case where we can play each arm multiple times in a d-step lookahead. The first result proves Lemma 6.4 in the main text.

Lemma 6.10. *For any $z \in \mathcal{Z}$ arm j and $n \in \mathbb{N}, n \geq 1$, let $Z^{(n)}$ be the z value when arm j is played for the nth time. Then,*

$$\sigma_j^2(z; n-1) - \sigma_j^2(z; n) = \frac{k_j^2(Z_j^{(n)}, z; n-1)}{\sigma_j^2(Z_j^{(n)}; n-1) + \sigma^2} \leq \frac{\sigma_j^2(Z_j^{(n)}; n-1)}{\sigma^2}$$

Proof. For convenience, we drop the j notation and let $\mathbf{k}_n(z) = [k(Z^{(1)}, z), \ldots, k(Z^{(n)}, z)]^T$

and $\mathbf{K}_n = [k(Z^{(i)}, Z^{(j)})]_{i,j=1}^n$. Then,

$$\begin{aligned}
&\sigma^2(z; n-1) - \sigma^2(z; n) \\
&\quad = k(z,z) - \mathbf{k}_{n-1}(z)^T(\mathbf{K}_{n-1} + \sigma^2\mathbf{I})^{-1}\mathbf{k}_{n-1}(z) - k(z,z) + \mathbf{k}_n(z)^T(\mathbf{K}_n + \sigma^2\mathbf{I})^{-1}\mathbf{k}_n(z) \\
&\quad = \mathbf{k}_n(z)^T(\mathbf{K}_n + \sigma^2\mathbf{I})^{-1}\mathbf{k}_n(z) - \mathbf{k}_{n-1}(z)^T(\mathbf{K}_{n-1} + \sigma^2\mathbf{I})^{-1}\mathbf{k}_{n-1}(z) \qquad (6.8)
\end{aligned}$$

We write,

$$\mathbf{k}_n(z) = \begin{bmatrix} \mathbf{k}_{n-1}(z) \\ k(Z^{(n)}, z) \end{bmatrix} \quad \mathbf{K}_n + \sigma^2\mathbf{I} = \begin{pmatrix} \mathbf{K}_{n-1} + \sigma^2\mathbf{I} & \mathbf{k}_{n-1}(z) \\ \mathbf{k}_{n-1}(z)^T & k(Z^{(n)}, Z^{(n)}) + \sigma^2 \end{pmatrix} = \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & C \end{pmatrix}.$$

Then, by the block matrix inversion formula,

$$(\mathbf{K}_n + \sigma^2\mathbf{I})^{-1} = \begin{pmatrix} \mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{B}(C - \mathbf{B}^T\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{B}^T\mathbf{A}^{-1} & -\mathbf{A}^{-1}\mathbf{B}(C - \mathbf{B}^T\mathbf{A}^{-1}\mathbf{B})^{-1} \\ -(C - \mathbf{B}^T\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{B}^T\mathbf{A}^{-1} & (C - \mathbf{B}^T\mathbf{A}^{-1}\mathbf{B})^{-1} \end{pmatrix}$$

Hence,

$$\mathbf{k}_n(z)^T(\mathbf{K}_n+\sigma^2\mathbf{I})^{-1}\mathbf{k}_n(z) = [\mathbf{k}_{n-1}(z)^T, k(Z^{(n)},z)](\mathbf{K}_n+\sigma^2\mathbf{I})^{-1}\begin{bmatrix}\mathbf{k}_{n-1}(z)\\ k(Z^{(n)},z)\end{bmatrix}$$

$$\begin{aligned}
&= \mathbf{k}_{n-1}(z)^T(\mathbf{A}^{-1}+\mathbf{A}^{-1}\mathbf{B}(C-\mathbf{B}^T\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{B}^T\mathbf{A}^{-1})\mathbf{k}_{n-1}(z)\\
&\quad - k(Z^{(n)},z)(C-\mathbf{B}^T\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{B}^T\mathbf{A}^{-1}\mathbf{k}_{n-1}(z)\\
&\quad - \mathbf{k}_{n-1}(z)^T\mathbf{A}^{-1}\mathbf{B}(C-\mathbf{B}^T\mathbf{A}^{-1}\mathbf{B})^{-1}k(Z^{(n)},z)\\
&\quad + k(Z^{(n)},z)(C-\mathbf{B}^T\mathbf{A}^{-1}\mathbf{B})^{-1}k(Z^{(n)},z)\\
&= \mathbf{k}_{n-1}(z)^T\mathbf{A}^{-1}\mathbf{k}_{n-1}(z)\\
&\quad + \mathbf{k}_{n-1}(z)^T(\mathbf{A}^{-1}\mathbf{B}(C-\mathbf{B}^T\mathbf{A}^{-1}\mathbf{B})^{-1}(\mathbf{B}^T\mathbf{A}^{-1}\mathbf{k}_{n-1}(z)-k(Z^{(n)},z))\\
&\quad + (k(Z^{(n)},z)-\mathbf{k}_{n-1}(z)^T\mathbf{A}^{-1}\mathbf{B})(C-\mathbf{B}^T\mathbf{A}^{-1}\mathbf{B})^{-1}k(Z^{(n)},z)\\
&= \mathbf{k}_{n-1}(z)^T\mathbf{A}^{-1}\mathbf{k}_{n-1}(z)\\
&\quad + (k(Z^{(n)},z)-\mathbf{k}_{n-1}(z)^T\mathbf{A}^{-1}\mathbf{B})(C-\mathbf{B}^T\mathbf{A}^{-1}\mathbf{B})^{-1}(k(Z^{(n)},z)-(\mathbf{k}_{n-1}(z)^T\mathbf{A}^{-1}\mathbf{B})^T)
\end{aligned}$$

Then, substituting back $\mathbf{A}=\mathbf{K}_{n-1}+\sigma^2\mathbf{I}, \mathbf{B}=\mathbf{k}_{n-1}(z), C=k(Z^{(n)},z_{(n)})+\sigma^2$ gives,

$$\begin{aligned}
\mathbf{k}_n(z)^T(\mathbf{K}_n+\sigma^2\mathbf{I})^{-1}\mathbf{k}_n(z) =& \mathbf{k}_{n-1}(z)^T(\mathbf{K}_{n-1}+\sigma^2\mathbf{I})^{-1}\mathbf{k}_{n-1}(z)\\
&+ (k(Z^{(n)},z)-\mathbf{k}_{n-1}(z)^T(\mathbf{K}_{n-1}+\sigma^2\mathbf{I})^{-1}\mathbf{k}_{n-1}(z))\\
&\quad (k(Z^{(n)},Z^{(n)})-\mathbf{k}_{n-1}(z_n)^T(\mathbf{K}_{n-1}+\sigma^2\mathbf{I})^{-1}\mathbf{k}_{n-1}(z)+\sigma^2)^{-1}\\
&\quad (k(Z^{(n)},z)-(\mathbf{k}_{n-1}(z)^T(\mathbf{K}_{n-1}+\sigma^2\mathbf{I})^{-1}\mathbf{k}_{n-1}(z))^T)\\
=& \mathbf{k}_{n-1}(z)^T(\mathbf{K}_{n-1}+\sigma^2\mathbf{I})^{-1}\mathbf{k}_{n-1}(z)+\frac{k^2(Z^{(n)},z;n-1)}{\sigma^2(Z^{(n)};n-1)+\sigma^2}
\end{aligned}$$

Hence, substituting into (6.8) gives,

$$\sigma^2(z;n-1)-\sigma^2(z;n)=\frac{k^2(Z^{(n)},z;n-1)}{\sigma^2(Z^{(n)};n-1)+\sigma^2}.$$

Then, since the covariance matrix is positive semi-definite, for any z, z' and $m\in\mathbb{N}$,

$k(z, z'; m) \leq \sqrt{\sigma^2(z; m)\sigma^2(z'; m)}$ and so

$$\sigma^2(z; n-1) - \sigma^2(z; n) \leq \frac{\sigma^2(Z^{(n)}; n-1)\sigma^2(z; n-1)}{\sigma^2(Z^{(n)}; n-1) + \sigma^2} \leq \frac{\sigma^2(Z^{(n)}; n-1)}{\sigma^2}$$

since for any $z \in \mathcal{Z}$ and $m \in \mathbb{N}$, $0 \leq \sigma^2(z; m) \leq 1$. This concludes the proof. □

We then use this result in the following lemma,

Lemma 6.11. *For any leaf node i of the d-step look ahead tree constructed at time t,*

$$\varsigma_t^2(i) \leq 3 \sum_{j=1}^{K} \left(\sum_{m=N_j(t)+1}^{N_j(t+d)} \frac{N_j(t+d) - m + 1}{\sigma^2} \sigma_j^2(z^{(m)}; m-1) \right) = \zeta_t^2$$

and ζ_t is $\mathcal{F}_{t-1}$ measurable.

Proof. First note that since the posterior covariance matrix of f_j is positive semi-definite, for any z_1, z_2 and number of samples, $n-1$, $k_j(z_1, z_2; n-1) \leq 1/2(\sigma_j^2(z_1; n-1) + \sigma_j^2(z_2; n-1))$. Hence,

$$\varsigma_t(i) \leq 3 \sum_{\ell=0}^{d-1} \sigma_t^2(J_{t+\ell}).$$

Now consider arm j and assume it appears $s \leq d$ times in the d-step look ahead policy selected at time t. Then, the contribution of arm j (which for ease of notation we assume has been played $n-1$ times previously) to $\varsigma_t^2(i)$ is given below where we use the notation $\sigma_j^2(z^{(i)}; n-1)$ to denote the posterior variance at the ith z of arm j given

$n-1$ observations of arm j.

$$\begin{aligned}
&\sum_{m=n}^{n+s-1} \sigma_j^2(Z_j^{(m)}; n-1) = \sigma_j^2(z^{(n)}; n-1) + \cdots + \sigma_j^2(z^{(n+s-1)}; n-1) \\
&= \sigma_j^2(z^{(n)}; n-1) + \sigma_j^2(z^{(n+1)}; n) + \left(\sigma_j^2(z^{(n+1)}; n-1) - \sigma_j^2(z^{(n+1)}; n)\right) + \dots \\
&\quad + \sigma_j^2(z^{(n+s-1)}; n+s-2) + \left(\sigma_j^2(z^{(n+s-1)}; n+s-3) - \sigma_j^2(z^{(n+s-1)}; n+s-2)\right) \\
&\qquad + \cdots + \left(\sigma_j^2(z^{(n+s-1)}; n-1) - \sigma_j^2(z^{(n+s-1)}; n)\right) \\
&\leq \sigma_j^2(z^{(n)}; n-1) + \sigma_j^2(z^{(n+1)}; n) + \frac{\sigma_j^2(z^{(n)}; n-1)}{\sigma^2} + \dots \\
&\quad + \sigma_j^2(z^{(n+s-1)}; n+s-2) + \cdots + \frac{\sigma_j^2(z^{(n+1)}; n)}{\sigma^2} + \frac{\sigma_j^2(z^{(n)}; n-1)}{\sigma^2} \\
&= \sum_{q=0}^{s-1} (1 + \frac{s-q-1}{\sigma^2}) \sigma_j^2(z^{(n+q)}; n+q-1) \\
&\leq \sum_{q=0}^{s-1} \frac{s-q}{\sigma^2} \sigma_j^2(z^{(n+q)}; n+q-1)
\end{aligned}$$

which follows by recursively applying Lemma 6.4. Then, summing over all arms j gives,

$$\begin{aligned}
\varsigma_t^2(i) &\leq 3 \sum_{j=1}^{K} \left(\sum_{m=N_j(t)+1}^{N_j(t+d)} \sigma_j^2(z^{(m)}; N_j(t)) \right) \\
&\leq 3 \sum_{j=1}^{K} \left(\sum_{m=N_j(t)+1}^{N_j(t+d)} \frac{N_j(t+d) - m + 1}{\sigma^2} \sigma_j^2(z^{(m)}; m-1) \right)
\end{aligned}$$

Then, we note that ζ_t is $\mathcal{F}_{t-1}$ measurable since for a given leaf node i of the tree constructed at time t, the sequence of arms played to get to node i is known so $N_j(t+d)$ will be known and also the sequence of $Z_j^{(m)}$'s where arm j is played will also be known. Since the posterior variance of arm j after m plays depends only on the number of plays and the covariates (not the observed rewards), $\sigma_j^2(z^{(m)}; m-1)$ is $\mathcal{F}_{t-1}$ measurable for $m = N_j(t)+1, \dots, N_j(t+d)$. □

Lemma 6.12. *Let $X_1, \dots X_n$ be Gaussian random variables such that* $\max_{1 \leq i \leq n} \mathbb{V}(X_i) \leq$

ζ^2. *Then,*

$$\mathbb{E}\left[\max_{1\leq i\leq n} X_i\right] \leq \zeta\sqrt{2\log(n)}.$$

Proof. See for example, Lemma 2.2 in (Devroye and Lugosi, 2001). □

6.B Theoretical Results for dRGP-UCB

We first prove the following lemma.

Lemma 6.13. *For any leaf node i, initial node z and constant $a > 0$,*

$$\int_a^\infty \mathbb{P}(M_i(z) - \eta_t(i) \geq x|\mathcal{F}_{t-1})dx \leq \sqrt{2\pi}\varsigma_t(i)\exp\left\{-\frac{a^2}{2\varsigma_t^2(i)}\right\}.$$

Proof. The proof follows using the normality of the posterior of $M_i(z)$ (so at time t, $M_i(\mathbf{Z}_t) \sim \mathcal{N}(\eta_t(i), \varsigma_t(i)^2)$.

$$\begin{aligned}\int_a^\infty \mathbb{P}(M_i(z) - \eta_t(i) \geq x|\mathcal{F}_{t-1})dx &\leq \int_a^\infty \exp\left\{-\frac{x^2}{2\varsigma_t^2(i)}\right\}dx \\ &= \sqrt{2\pi}\varsigma_t(i)\int_a^\infty \frac{1}{\sqrt{2\pi}\varsigma_t(i)}\exp\left\{-\frac{x^2}{2\varsigma_t^2(i)}\right\}dx \\ &\leq \sqrt{2\pi}\varsigma_t(i)\exp\left\{-\frac{a^2}{2\varsigma_t^2(i)}\right\}.\end{aligned}$$

Where we have used that if $X \sim \mathcal{N}(\mu, \sigma^2)$, $\mathbb{P}(X - \mu \geq b) \leq \exp\{-\frac{b^2}{2\sigma^2 2}\}$ for any $b > 0$, and the last inequality follows through integration of the pmf of a $\mathcal{N}(0, \varsigma_t(i))$ random variable. □

Then, define $M_{I_t^*}(\mathbf{Z}_t)$ to be the sum of the $f_j(z)$'s to leaf $I_t{}^*$ of the optimal d step look ahead policy from time t chosen using the unknown $f_j(z)$'s. Let r_t be the per step regret at time t. We now bound the expected regret from time steps $t, t+1, \ldots, t+d-1$ where we have played arms according to the choice of I_t by our algorithm. Let r_s be

the contribution to the regret at time s, that is $r_s = f_{J_t^*}(Z_{J_t^*,t}) - f_{J_t}(Z_{J_t,t})$. Then, let

$$\alpha_t = \sqrt{2\log((K|\mathcal{Z}|)^d(t+d-1)^2)}.$$

We will use the following lemma,

Lemma 6.14. *Assume we start a d-step look ahead policy at time t, selecting leaf node I_t, then*

$$\sum_{s=t}^{t+d-1} \mathbb{E}[r_s|\mathcal{F}_{t-1}] \leq \frac{\sqrt{2d\pi}}{(t+d-1)^2} + \alpha_t \varsigma_t(I_t).$$

Proof. From (6.3), the upper confidence bound of node i at time t is given by,

$$\eta_t(i) + \alpha_t \varsigma_t(i),$$

and since we play node I_t, this has the highest upper confidence bound. Then, we use the following decomposition of the regret,

$$\begin{aligned}
\sum_{s=t}^{t+d-1} \mathbb{E}[r_s|\mathcal{F}_{t-1}] &= \mathbb{E}[M_{I_t^*}(\mathbf{Z}_t) - M_{I_t}(\mathbf{Z}_t)|\mathcal{F}_{t-1}] \\
&= \mathbb{E}[M_{I_t^*}(\mathbf{Z}_t) - (\eta_t(I_t^*) + \alpha_t\varsigma_t(I_t^*)) + (\eta_t(I_t^*) + \alpha_t\varsigma_t(I_t^*)) - M_{I_t}(\mathbf{Z}_t)|\mathcal{F}_{t-1}] \\
&\leq \mathbb{E}[M_{I_t^*}(\mathbf{Z}_t) - (\eta_t(I_t^*) + \alpha_t\varsigma_t(I_t^*)) + (\eta_t(I_t) + \alpha_t\varsigma_t(I_t)) - M_{I_t}(\mathbf{Z}_t)|\mathcal{F}_{t-1}] \\
&= \mathbb{E}[M_{I_t^*}(\mathbf{Z}_t) - \eta_t(I_t^*) - \alpha_t\varsigma_t(I_t^*)|\mathcal{F}_{t-1}] + \mathbb{E}[\eta_t(I_t) + \alpha_t\varsigma_t(I_t) - M_{I_t}(\mathbf{Z}_t)|\mathcal{F}_{t-1}]
\end{aligned}$$

For the first term, note that for any random variable X, $\mathbb{E}[X] \leq \mathbb{E}[X\mathbb{I}\{X > 0\}] = \int_0^\infty \mathbb{P}(X \geq x)dx$. Then, by Lemma 6.13 and using the fact that $\varsigma_t^2(i) \leq \sum_{\ell=0}^{d-1} k(z_\ell, z_\ell) \leq$

d, it follows that,

$$\begin{aligned}
&\mathbb{E}[M_{I_t^*}(\mathbf{Z}_t) - \eta_t(I_t^*) - \alpha_t\varsigma_t(I_t^*)|\mathcal{F}_{t-1}] \\
&\quad\leq \int_0^\infty \mathbb{P}(M_{I_t^*}(\mathbf{Z}_t) - \eta_t(I_t^*) - \alpha_t\varsigma_t(I_t^*) \geq x|\mathcal{F}_{t-1})dx \\
&\quad\leq \int_0^\infty \sum_{i=1}^{K^d}\sum_{z\in\mathcal{Z}^d} \mathbb{P}(M_i(z) - \eta_t(i) - \alpha_t\varsigma_t(i) \geq x|\mathcal{F}_{t-1})dx \\
&\quad= \sum_{i=1}^{K^d}\sum_{z\in\mathcal{Z}^d} \int_{\alpha_t\varsigma_t(i)}^\infty \mathbb{P}(M_i(z) - \eta_t(i) \geq x|\mathcal{F}_{t-1})dx \\
&\quad= \sum_{i=1}^{K^d}\sum_{z\in\mathcal{Z}^d} \sqrt{2\pi}\varsigma_t(i)\exp\left\{-\frac{(\alpha_t\varsigma_t(i))^2}{2\varsigma_t^2(i)}\right\} \\
&\quad\leq \sum_{i=1}^{K^d}\sum_{z\in\mathcal{Z}^d} \sqrt{2d\pi}\frac{1}{(t+d-1)^2(K|\mathcal{Z}|)^d} \\
&\quad= \frac{\sqrt{2d\pi}}{(t+d-1)^2},
\end{aligned}$$

where the last inequality follows from the definition of α_t.

For the second term, recall that $\eta_t(i) = \mathbb{E}[M_i(\mathbf{Z}_t)|\mathcal{F}_{t-1}]$ and I_t is $\mathcal{F}_{t-1}$ measurable. Hence,

$$\mathbb{E}[\eta_t(I_t) + \alpha_t\varsigma_t(I_t) - M_{I_t}(\mathbf{Z}_t)|\mathcal{F}_{t-1}] = \eta_t(I_t) + \alpha_t\varsigma_t(I_t) - \eta_t(I_t) = \alpha_t\varsigma_t(I_t).$$

Combining both terms gives the result.

□

We now prove the regret bounds for dRGP-UCB in the repeating and non-repeating cases.

6.B.1 Non-Repeating

Theorem 6.2. *The d-step single play lookahead regret of dRGP-UCB satisfies,*

$$\mathbb{E}[\mathfrak{R}_T^{(d,s)}] \leq O(\sqrt{KT\gamma_T \log(TK|\mathcal{Z}|)}).$$

Proof. For ease of notation define $\mathfrak{R}_T$ as the d-step lookahead regret with single plays that we are interested in (i.e. $\mathfrak{R}_T = \mathfrak{R}_T^{(d,s)}$) and note that,

$$\mathbb{E}[\mathfrak{R}_T] \leq \sum_{h=0}^{\lfloor T/d \rfloor} \mathbb{E}\Bigg[\sum_{s=hd+1}^{(h+1)d} \mathbb{E}[r_s|\mathcal{F}_{hd}]\Bigg].$$

Then, using Lemma 6.14, and the fact that since we cannot repeat plays, $\sigma_t(J_{t+\ell}) = \sigma_{t+\ell}(J_{t+\ell})$ for any $\ell = 0, \ldots, d-1$,

$$\begin{aligned}
\mathbb{E}[\mathfrak{R}_T] &\leq \sum_{h=0}^{\lfloor T/d \rfloor} \mathbb{E}\Bigg[\sum_{s=hd+1}^{(h+1)d} \mathbb{E}[r_s|\mathcal{F}_{hd}]\Bigg] \\
&\leq \sum_{h=0}^{\lfloor T/d \rfloor} \mathbb{E}\Bigg[\frac{\sqrt{2d\pi}}{(h+1)^2 d^2} + \alpha_{hd+1}\sqrt{\varsigma^2_{hd+1}(I_{hd+1})}\Bigg] \\
&\leq \frac{\sqrt{2\pi}}{d}\sum_{h=1}^{\lfloor T/d \rfloor+1} \frac{1}{h^2} + \sum_{h=0}^{\lfloor T/d \rfloor} \sqrt{2\log((K|\mathcal{Z}|)^d(h+1)^2d^2)}\mathbb{E}\Bigg[\sqrt{\sum_{\ell=0}^{d-1}\sigma_{hd+1}(J_{hd+1+\ell})}\Bigg] \\
&\leq \frac{\pi^{5/2}}{\sqrt{2}3d} + \sqrt{4\log((K|\mathcal{Z}|)^d(T+d))}\sqrt{\lfloor T/d \rfloor + 1}\mathbb{E}\Bigg[\sqrt{\sum_{t=1}^{T}\sigma_t^2(J_t)}\Bigg] \\
&\leq \frac{\pi^{5/2}}{\sqrt{2}3d} + \sqrt{4\log((K|\mathcal{Z}|)^d(T+d))}\sqrt{\lfloor T/d \rfloor + 1}\mathbb{E}\Bigg[\sqrt{\sum_{j=1}^{K}\sum_{t=1}^{T}\sigma_t^2(j)\mathbb{I}\{J_t = j\}}\Bigg] \\
&\leq \frac{\pi^{5/2}}{\sqrt{2}3d} + \sqrt{4\log((K|\mathcal{Z}|)^d(T+d))}\sqrt{\lfloor T/d \rfloor + 1}\sqrt{C_1 K\gamma_T}
\end{aligned}$$

where $C_1 = 1/\log(1+\sigma^{-2})$ and the last line follows by Lemma 6.9. This gives the result.

□

6.B.2 Repeating

Theorem 6.5. *The d-step multiple play lookahead regret of dRGP-UCB satisfies,*

$$\mathbb{E}[\mathfrak{R}_T^{(d,m)}] \leq O\left(\sqrt{KT\gamma_T \log((K|\mathcal{Z}|)^d T)}\right).$$

Proof. For ease of notation define $\mathfrak{R}_T$ as the d-step lookahead regret with multiple plays that we are interested in (i.e. $\mathfrak{R}_T = \mathfrak{R}_T^{(d,m)}$) and note that,

$$\mathbb{E}[\mathfrak{R}_T] = \sum_{h=0}^{\lfloor T/d \rfloor} \mathbb{E}\left[\sum_{s=hd+1}^{(h+1)d} \mathbb{E}[r_s|\mathcal{F}_{hd}]\right].$$

Then, note that from Lemma 6.11, it follows that

$$\begin{aligned}\varsigma_t^2(i) &\leq 3\sum_{j=1}^{K}\left(\sum_{m=N_j(t)+1}^{N_j(t+d)} \frac{N_j(t+d)-m+1}{\sigma^2}\sigma_j^2(z^{(m)};m-1)\right)\\ &\leq \frac{3d}{\sigma^2}\sum_{j=1}^{K}\sum_{m=N_j(t)+1}^{N_j(t+d)}\sigma_j^2(z^{(m)};m-1).\end{aligned}$$

Hence, by Lemma 6.14 and summing over all time points where we start a d-step look

ahead policy, it follows that,

$$\begin{aligned}
\mathbb{E}[\mathfrak{R}_T] &= \sum_{h=0}^{\lceil T/d \rceil - 1} \mathbb{E}\Bigg[\sum_{s=hd+1}^{(h+1)d} \mathbb{E}[r_s|\mathcal{F}_{hd}]\Bigg] \\
&\leq \sum_{h=0}^{\lfloor T/d \rfloor} \mathbb{E}\Bigg[\frac{\sqrt{2d\pi}}{(h+1)^2 d^2} + \alpha_{hd+1}\sqrt{\varsigma^2_{hd+1}(I_{hd+1})}\Bigg] \\
&\leq \frac{\sqrt{2\pi}}{d} \sum_{h=1}^{\lfloor T/d \rfloor + 1} \frac{1}{h^2} \\
&+ \sum_{h=0}^{\lfloor T/d \rfloor} \sqrt{2\log((K|\mathcal{Z}|)^d (h+1)^2 d^2)}\mathbb{E}\Bigg[\sqrt{\frac{3d}{\sigma^2}\sum_{j=1}^{K}\sum_{m=N_j(d(h+1))}^{N_j(d(h+1))} \sigma_j^2(z^{(m)}; m-1)}\Bigg] \\
&\leq \frac{\pi^{5/2}}{\sqrt{2}3d} \\
&+ \sqrt{\frac{12d}{\sigma^2}\log((K|\mathcal{Z}|)^d (T+d))}\sqrt{\lfloor T/d \rfloor + 1}\mathbb{E}\Bigg[\sqrt{\sum_{h=0}^{\lfloor T/d \rfloor}\sum_{j=1}^{K}\sum_{m=N_j(dh)+1}^{N_j(d(h+1))} \sigma_j^2(z^{(m)}; m-1)}\Bigg]
\end{aligned}$$

Then, from Lemma 6.9 and the fact that γ_n is increasing in n,

$$\begin{aligned}
\sqrt{\sum_{h=0}^{\lfloor T/d \rfloor}\sum_{j=1}^{K}\sum_{m=N_j(dh)+1}^{N_j(d(h+1))} \sigma_j^2(z^{(m)}; m-1)} &= \sqrt{\sum_{j=1}^{K}\sum_{m=1}^{N_j(T)} \sigma_j^2(z^{(m)}; m-1)} \\
&\leq \sqrt{\sum_{j=1}^{K} C_1 \gamma_{N_j(T)}} \leq \sqrt{C_1 K \gamma_T}
\end{aligned}$$

for $C_1 = (1 + \log(\sigma^{-2}))^{-1}$. Hence,

$$\mathbb{E}[\mathfrak{R}_T] \leq \frac{\pi^{5/2}}{\sqrt{2}3d} + \sqrt{\frac{12d}{\sigma^2}\log((K|\mathcal{Z}|)^d (T+d))}\sqrt{T/d+1}\sqrt{C_1 K \gamma_T}$$

and so the result follows. □

6.C Theoretical Results for dRGP-TS

The regret bounds for the Thompson sampling approach(dRGP-TS) follow in a similar manner to those for dRGP-UCB using the techniques of Russo and Van Roy (2014). Specifically, using (Russo and Van Roy, 2014), we get the following result which is equivalent to Lemma 6.14, which can then be used to get the regret bound much in the same way as Theorem 6.2 and Theorem 6.5.

Lemma 6.15. *Assume we start a d-step look ahead policy at time t, selecting leaf node I_t, then*

$$\sum_{s=t}^{t+d-1} \mathbb{E}[r_s|\mathcal{F}_{t-1}] \leq \frac{\sqrt{2d\pi}}{(t+d-1)^2} + \alpha_t \varsigma_t(I_t).$$

Proof. As in (Russo and Van Roy, 2014) we relate the Bayesian regret of Thompson sampling to the upper confidence bounds used in our upper confidence bound approach. Specifically, by Proposition 1 in (Russo and Van Roy, 2014),

$$\begin{aligned}\sum_{s=t}^{t+d-1} \mathbb{E}[r_s|\mathcal{F}_{t-1}] &= \mathbb{E}[M_{I_t^*}(\mathbf{Z}_t) - M_{I_t}(\mathbf{Z}_t)|\mathcal{F}_{t-1}] \\ &= \mathbb{E}[M_{I_t^*}(\mathbf{Z}_t) - \eta_t(I_t^*) - \alpha_t\varsigma_t(I_t^*)|\mathcal{F}_{t-1}] + \mathbb{E}[\eta_t(I_t) + \alpha_t\varsigma_t(I_t) - M_{I_t}(\mathbf{Z}_t)|\mathcal{F}_{t-1}]\end{aligned}$$

The same argument as Lemma 6.14 then gives the result.

□

6.C.1 Non-Repeating

Theorem 6.3. *The d-step non-repeating lookahead regret of dRGP-TS satisfies,*

$$\mathbb{E}[\mathfrak{R}_T^{(d,s)}] \leq O(\sqrt{KT\gamma_T \log(TK|\mathcal{Z}|)}).$$

Proof. Given Lemma 6.15, the proof follows in the same manner as the proof of

Theorem 6.2. □

6.C.2 Repeating

Theorem 6.6. *The d-step multiple play lookahead regret of dRGP-TS satisfies,*

$$\mathbb{E}[\mathfrak{R}_T^{(d,m)}] \leq O\left(\sqrt{KT\gamma_T \log((K|\mathcal{Z}|)^d T)}\right).$$

Proof. The proof follows by the same argument as Theorem 6.5 using Lemma 6.15. □

6.D Regret Bounds for Non-Parametric Approach

Recall the non-parametric approach described in Section 6.5. We model each (arm, z) combination as an 'arm' and let $\mu_{j,z}$ denote the expected reward of arm j when $z_j = z$. We can then create estimates $\bar{Y}_{j,z,t}$ of the reward of each arm from the $N_{j,z}(t)$ samples of arm j with $Z_j = z$ we receive up to time t. These estimates can be used to define an upper confidence bound style algorithm over the 'arms' $\{(j,z)\}_{j=1,z=0}^{K,Z_{\max}}$.

We define confidence bound based on UCB1 (Auer et al., 2002a) and Russo and Van Roy (2014)

$$U(j,z,t) = \bar{Y}_{z,j,t} + \sqrt{\frac{\sigma^2(2+6\log(T))}{N_{j,z}(t)}}.$$

where σ is the standard error of the noise. After playing each j,z combinations once, we proceed to play the arm with largest $U(j, Z_{j,t}, t)$ at time t. We now bound the regret of this algorithm to horizon T.

Theorem 6.1. *The instantaneous regret up to time T of the UCB1 algorithm with $K|\mathcal{Z}|$ arms can be bounded by*

$$\mathbb{E}[\mathfrak{R}_T^{(1)}] \leq O(\sqrt{K|\mathcal{Z}|T\log(T)} + K|\mathcal{Z}|^2)$$

Proof. First let $t_0 = K|\mathcal{Z}|(|\mathcal{Z}|+1)$ and note that since we need to wait z steps after playing arm j to have $Z_j = z$, after t_0 steps, we can guarantee to have played each arm at least once. Then by Lemma 6.12, for any $1 \leq t \leq t_0$,

$$\mathbb{E}[f_{J_t^*}(Z_{J_t^*,t}) - f_{J_t}(Z_{J_t,t})] \leq \mathbb{E}[\max_{1\leq t\leq t_0}\{f_{J_t^*}(Z_{J_t^*,t}) - f_{J_t}(Z_{J_t,t})\}] \leq 2\sqrt{2\log(t_0)}$$

since the distribution of the difference of two zero mean Gaussian random variables is also a Gaussian random variable with mean 0 and variance $\sigma_1^2 + \sigma_2^2 \leq 2$ here. Then, we can use a similar technique to Russo and Van Roy (2014) to bound the cumulative regret in the remaining $t_0 + 1 \leq t \leq T$ steps but using Lemma 6.12 again to bound the maximal difference in f_j's.

$$\begin{aligned}
\mathbb{E}[\mathfrak{R}_T] &= \sum_{t=t_0}^{T} \mathbb{E}[f_{J_t^*}(Z_{J_t^*,t}) - f_{J_t}(Z_{J_t,t})\mathbb{I}\{\forall j, z; f_j(z) \in [L(j,z,t), U(j,z,t)]\}] \\
&\qquad + \sum_{t=t_0}^{T} \mathbb{E}[f_{J_t^*}(Z_{J_t^*,t}) - f_{J_t}(Z_{J_t,t})\mathbb{I}\{\exists j, z; f_j(z) \notin [L(j,z,t), U(j,z,t)]\}] \\
&\leq \sum_{t=t_0}^{T} \mathbb{E}[U(J_t^*, Z_{J_t^*,t}, t) - L(J_t, Z_{J_t,t}, t)] \\
&\qquad + 2\sqrt{2\log(T)}T\mathbb{P}(\exists j, z; f_j(z) \notin [L(j,z,t), U(j,z,t)]) \\
&\leq \sum_{t=t_0}^{T} \mathbb{E}[U(J_t, Z_{J_t,t}, t) - L(J_t, Z_{J_t,t}, t)] \\
&\qquad + 2\sqrt{2\log(T)}T\sum_{j=1}^{K}\sum_{z\in\mathcal{Z}}\mathbb{P}(f_j(z) \notin [L(j,z,t), U(j,z,t)])
\end{aligned}$$

Since $\epsilon_t \sim \mathcal{N}(0, \sigma^2)$, by Lemma 1 in (Russo and Van Roy, 2014),

$$2\sqrt{2\log(T)}T\sum_{j=1}^{K}\sum_{z\in\mathcal{Z}}\mathbb{P}(f_j(z) \notin [L(j,z,t), U(j,z,t)]) \leq \frac{1}{T|\mathcal{Z}|K} \leq 2\sqrt{2\log(T)}.$$

Then, for the first term, by the same argument as Russo and Van Roy (2014),

$$\begin{aligned}
&\sum_{t=t_0}^{T}\mathbb{E}[U(J_t, Z_{J_t,t}, t) - L(J_t, Z_{J_t,t}, t)] \\
&\quad\leq \sum_{t=t_0}^{T}\sum_{j=1}^{K}\sum_{z\in\mathcal{Z}}\mathbb{E}[U(j,z,t) - L(j,z,t)\mathbb{I}\{J_t = j, Z_{J_t,t} = z\}] \\
&\quad\leq 2\sqrt{\sigma^2(2+6\log(T)}\sum_{t=t_0}^{T}\sum_{j=1}^{K}\sum_{z\in\mathcal{Z}}\mathbb{E}\left[\frac{1}{\sqrt{2N_{j,z}(t)}}\mathbb{I}\{J_t = j, Z_{J_t,t} = z\}\right] \\
&\quad\leq 2\sqrt{\sigma^2(2+6\log(T)}\sum_{j=1}^{K}\sum_{z\in\mathcal{Z}}\mathbb{E}\left[\sum_{l=0}^{N_{j,z}(T)-1}\frac{1}{\sqrt{l+1}}\right] \\
&\quad\leq 2\sqrt{\sigma^2(2+6\log(T)}\sum_{j=1}^{K}\sum_{z\in\mathcal{Z}}\mathbb{E}\left[\sqrt{N_{j,z}(T)}\right] \\
&\quad\leq 2\sqrt{\sigma^2(2+6\log(T)}\sqrt{K|\mathcal{Z}|T}
\end{aligned}$$

where the last line follows by Cauchy-Schwartz. This concludes the proof.

□

6.E Theoretical Guarantees on Optimistic Planning Procedure

Proposition 6.8. *In the multiple play case, for the optimistic planning procedure with a budget of N samples, if the procedure is stopped at step $n < N$ because we selected a node i_n of depth d to expand, then $v^* - v(i_n) = 0$. Otherwise, if there exists some $\lambda \in (\frac{1}{K}, 1]$ and $d_0 \in \{1, \dots, d\}$ such that $\forall l \geq d_0$, $p_l((d-l)\Delta) \leq \lambda^l$, then for $N > n_0 = \frac{K^{d_0+1}-1}{K-1}$,*

$$v^* - v(i_N) \leq \left(d - \frac{\log(N-n_0)}{\log(\lambda K)} - \frac{\log(\lambda K - 1)}{\log(\lambda K)} + 1\right)\Delta. \tag{6.6}$$

Proof. Since our $\tilde{f}_j(z)$'s are samples from a Gaussian posterior, they can be negative.

Hence it will be convenient to work with a transformation that guarantees positivity. To this end, let $\delta = -\min_{j,z} \tilde{f}_j(z)$ if $\min_{j,z} \tilde{f}_j(z) < 0$ and $\delta = 0$ if $\min_{j,z} \tilde{f}_j(z) \geq 0$ and for any arm j and covariate z, define,

$$\tilde{f}'_j(z) = \tilde{f}_j(z) + \delta \geq 0.$$

Then we define the corresponding v, b and u values of any node $i \in \mathcal{S}_n$ at step n and Ψ functions as,

$$v'(i) = v(i) + d\delta \qquad b'_n(i) = b_n(i) + d\delta \qquad u'(i) = u(i) + l(i)\delta$$
$$\Psi'(\mathbf{z}(i), d - l(i)) = \Psi(\mathbf{z}(i), d - l(i)) + (d - l(i))\delta \qquad \Psi'^*(l) = \Psi^*(l) + l\delta,$$

where $l(i)$ is the depth of node i. Note that node i^* maximizing $v(i)$ will also maximize $v'(i)$ and that if at step n we select a node maximizing $b_n(i)$ this will also be the node maximizing $b'_n(i)$ and so $v(i_1) \geq v(i_2) \iff v'(i_1) \geq v'(i_2)$ and $b(i_1) \geq b(i_2) \iff b'(i_1) \geq b'(i_2)$ for all nodes i_1, i_2. Furthermore, it holds that $v'(i) \geq u'(i)$ and that $b'(i)$ is an upper bound on $v'(i)$ for all nodes i and in particular $b'(i) = u'(i) + \Psi'(\mathbf{z}(i), d - l(i))$.

We begin with the case where the algorithm is stopped after some number n of nodes have been expanded because the selected node is of depth d. Let $i^*_1, \ldots, i^*_d$ be the nodes on the path to i^* and let j be the maximal depth of this path in $\mathcal{T}_n \cup \mathcal{S}_n$. If i_n is the node at depth d selected to be expanded at time n, then,

$$0 \leq v^* - v(i_n) = v'(i^*_j) - v'(i_n) \leq b'(i^*_j) - v'(i_n) \leq b'(i_n) - v'(i_n) = \Psi'(\mathbf{z}(i_n), d - d) = 0,$$

since we select node i_n at time n so it must have the largest $b_n(i)$ and $b'_n(i)$ value. This proves the first statement.

For the other case, define the set

$$\Gamma = \bigcup_{l=0}^{d} \{ \text{ node } i \text{ of depth } l \text{ such that } v^* - v(i) \leq \Psi'^*(d-l)\},$$

and note that if $v^* - v(i) \leq \Psi'^*(d-l)$ then also $v'^* - v'(i) \leq \Psi'^*(d-l)$ As in (Hren and Munos, 2008), we will show that all nodes expanded by our algorithm are in Γ. For this, let node i of depth l be chosen to be expanded at time n. This means it has the largest $b_n(i)$ (and $b'_n(i)$) value of all nodes in $\mathcal{S}_n$. We also now need to define the b value of a node in $\mathcal{T}_n$ as $b_n(i) = \max_{j \in C(i)} b_n(j)$ where $C(i)$ is the set of all children of node i, and we define $b'_n(i)$ correspondingly. This definition together with the previous remark means that for any $j \in \mathcal{T}_n$, $b'_n(i) \geq b'_n(j)$. Then for some $1 \leq j \leq d$, $i^*_j \in \mathcal{T}_n$, so it follows that $b'_n(i^*_j) \leq b'_n(i_n)$. But, the best value of any continuation of a path to the optimal node is simply v^* and so by definition of the b values $b'_n(i^*_j) \geq v'(i^*_j) = v'^*$. Hence, since $v'(i) \geq u'(i)$ and $\Psi'(\mathbf{z}(i), d-l) \leq \Psi'^*(d-l)$,

$$\begin{aligned} v'(i) \geq u'(i) = b'_n(i) - \Psi'(\mathbf{z}(i), d-l) \geq b'_n(i^*_j) - \Psi'(\mathbf{z}(i), d-l) \geq v'^* - \Psi'(\mathbf{z}(i), d-l) \\ \geq v'^* - \Psi'^*(d-l), \end{aligned}$$

it follows that $i \in \Gamma$. Then, we bound from below the maximal depth at which a node is chosen to be expanded. Let n_0 be the number of policies in Γ up to depth d_0 and let d_N be the maximal depth of any node expanded before the algorithm is stopped at time N. By the assumption in the proposition, the proportion of $(d-l)\Delta$-optimal nodes at depth l is bounded by λ^l. Then, $\Psi'^*(d-l) = \Psi(d-l) + (d-l)\delta \leq (d-l)\max_{j,z} \tilde{f}_j(z) - (d-l)\min_{j,z} \tilde{f}_j(z) = (d-l)\Delta$ by definition of Ψ and so $p_l(\Psi'^*(d-l)) \leq p_l((d-l)\Delta) \leq \lambda^l$. Hence,

$$N \leq n_0 + \sum_{l=d_0}^{d_N} \lambda^l K^l = n_0 + \sum_{l=d_0}^{d_N} A^l \leq n_0 + A^{d_0+1} \frac{A^{d_N - d_0} - 1}{A - 1}$$

for $A = \lambda K > 1$. Rearranging gives,

$$\begin{aligned} d_N &\geq d_0 + \log_A \left(\frac{(N - n_0)(A-1)}{A^{d_0+1}} + 1 \right) \geq d_0 + \log_A \left(\frac{(N - n_0)(A-1)}{A^{d_0+1}} \right) \\ &\geq \frac{\log(N - n_0)}{\log(K\lambda)} - 1 + \frac{\log(\lambda K - 1)}{\log(\lambda K)} \end{aligned}$$

Let i_N be the node the algorithm outputs at step N when the computational resources have been exceeded and note that this is the node in $\mathcal{T}_N$ with largest depth (i.e. $l(i_N) = d_N$) that has the largest b_N (or b'_N) value. Since $i_N \in \mathcal{T}_N$, there is some step $n \leq N$ when node i_N was expanded. Then, let j be the maximal depth of nodes on the path $i_1^*, \ldots, i_d^*$ in $\mathcal{S}_n$. It then follows that

$$v'^* - v'(i_N) \leq b'_n(i_j^*) - v'(i_N) \leq b'_n(i_N) - v(i_N) \leq \Psi'(\mathbf{z}(i_N), d - l(i_N)) \leq \Psi'^*(d - d_N).$$

Hence,

$$\begin{aligned} v^* - v(i_N) = v'^* - v'(i_N) &\leq \Psi'^*(d - d_N) = \Psi^*(d - d_N) + (d - d_N)\delta \\ &\leq (d - d_N)(\max_{j,z} \tilde{f}_j(z) - \min_{j,z} \tilde{f}_j(z)) \\ &\leq \left(d - \frac{\log(N - n_0)}{\log(K\lambda)} - \frac{\log(\lambda K - 1)}{\log(\lambda K)} + 1 \right) \Delta \end{aligned}$$

which gives the result.

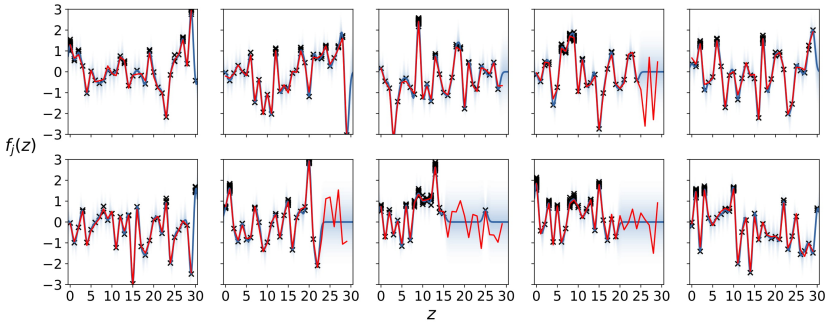
□

6.F Further Experimental Results

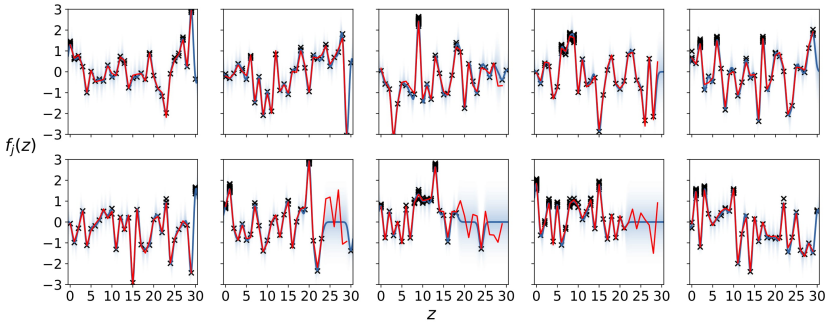
6.F.1 Posterior Distributions and Covariates

*d*RGP-UCB

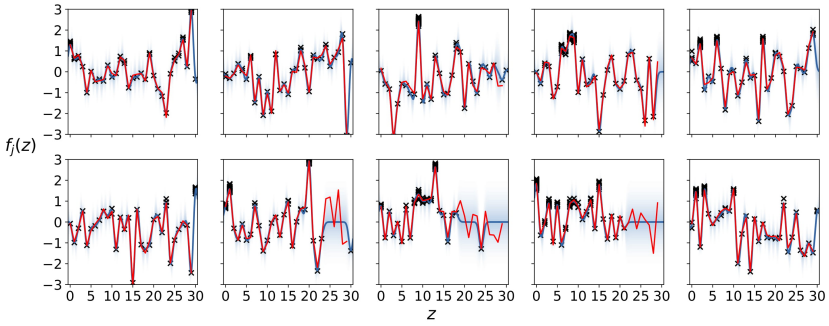
In this section, we plot the posterior (blue) of *d*RGP-UCB with density given by the blue region in the instantaneous case for various values of d and different kernels. The red curve is the true recovery curve and the crosses are our observed samples. Note that as the kernel gets smoother, the algorithm places more samples in the good regions. This is to be expected as for smoother kernels, there is less need to explore as many sub-optimal regions. Also, as d increases more samples are at the peak and there are less poorly estimated areas.



(a) $d = 1$

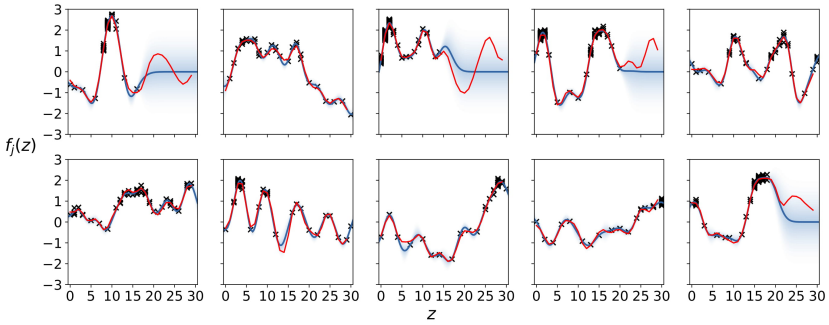


(b) $d = 2$

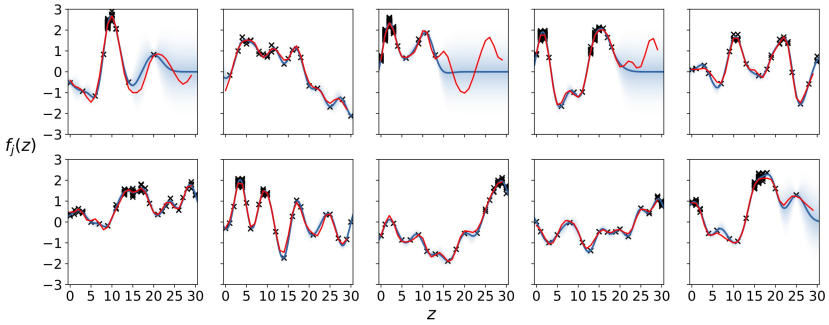


(c) $d = 3$

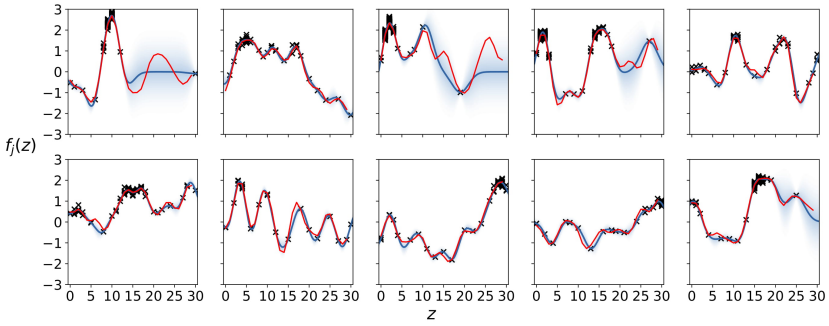
Figure 6.6: dRGP-UCB with squared exponential kernel with $l = 0.5$.



(a) $d = 1$

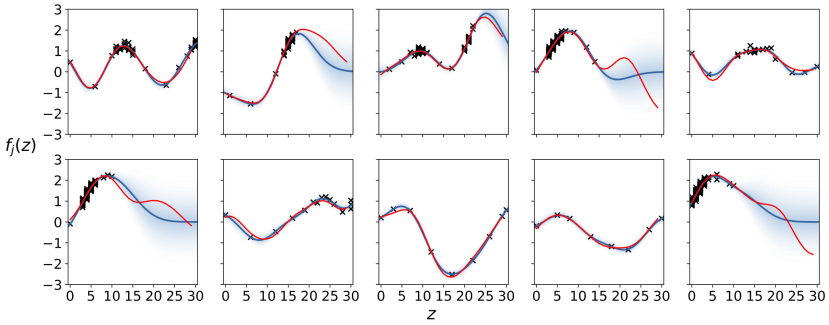


(b) $d = 2$

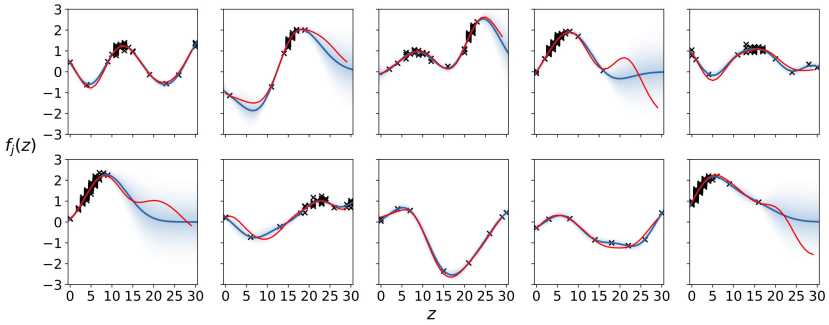


(c) $d = 3$

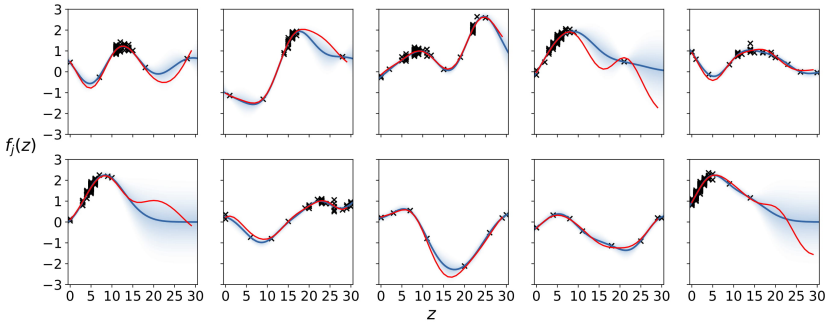
Figure 6.7: dRGP-UCB with squared exponential kernel with $l = 2$.



(a) $d = 1$



(b) $d = 2$

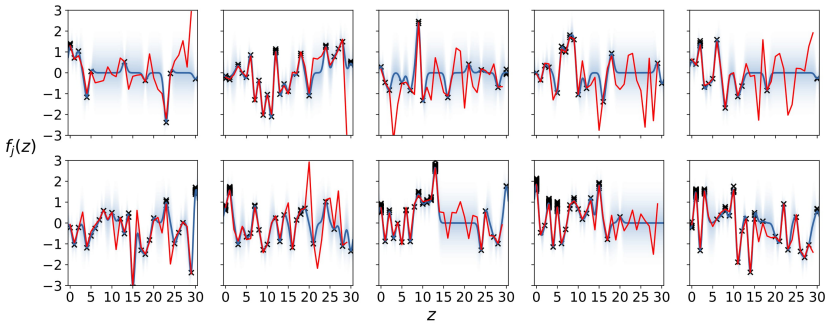


(c) $d = 3$

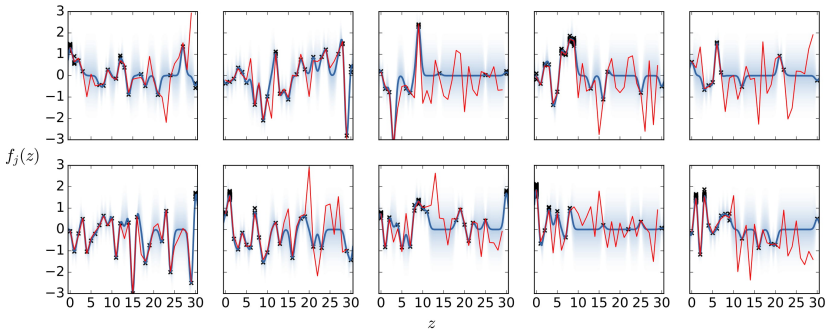
Figure 6.8: dRGP-UCB with squared exponential kernel with $l = 5$.

*d*RGP-TS

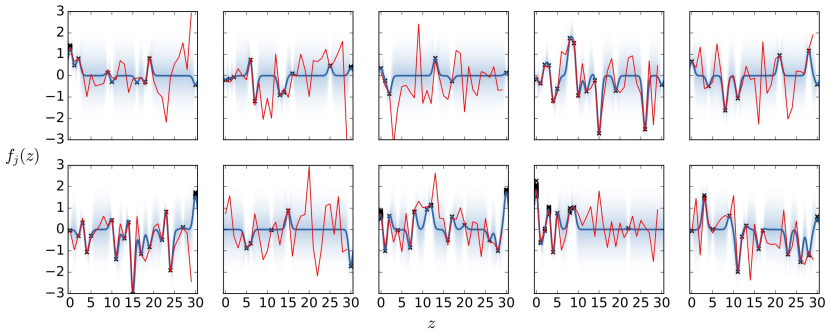
In this section, we plot the posterior (blue) of *d*RGP-TS. with density given by the blue region with different l's and d's. We see much the same pattern as for *d*RGP-UCB, although it does seem to demonstrate poorer estimation of the recovery curve in the single step case. However, it is worth noting that the algorithms have only been run once for these plots.



(a) $d = 1$

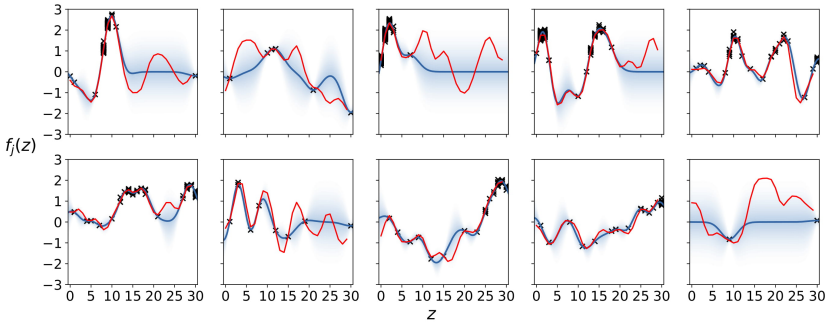


(b) $d = 2$

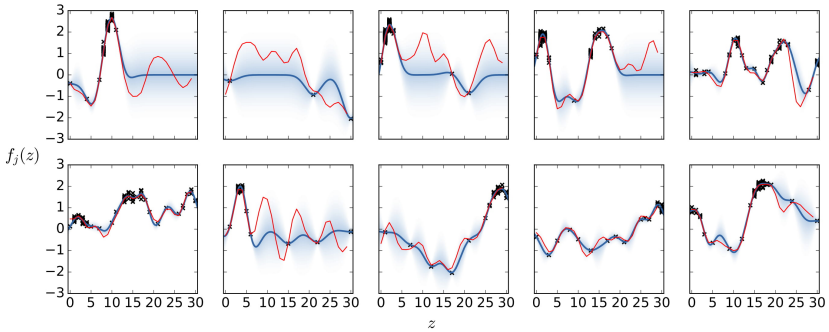


(c) $d = 3$

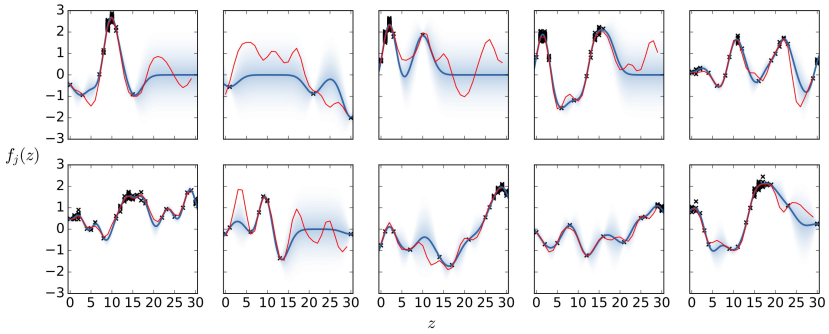
Figure 6.9: *d*RGP-TS for squared exponential kernel with $l = 0.5$.



(a) $d = 1$

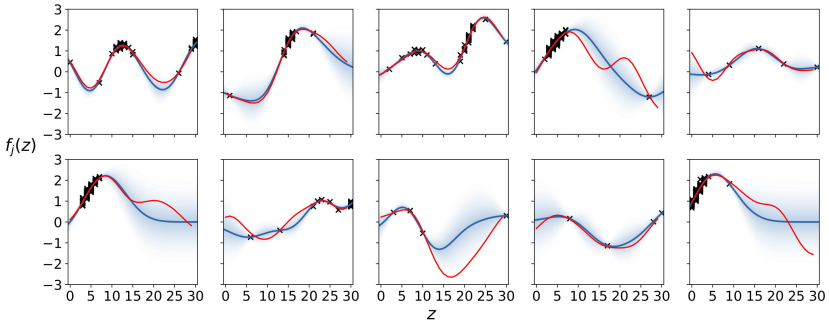


(b) $d = 2$

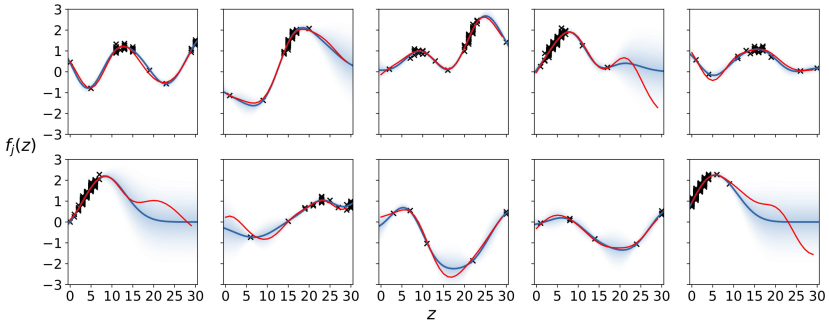


(c) $d = 3$

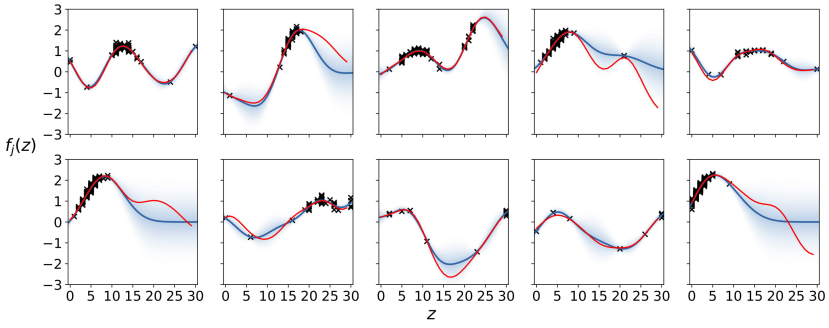
Figure 6.10: dRGP-TS for squared exponential kernel with $l = 2$.



(a) $d = 1$



(b) $d = 2$



(c) $d = 3$

Figure 6.11: dRGP-TS wit squared exponential kernel with $l = 5$.

6.F.2 Values of Theta in Parametric Experiments

Here we give the values of θ (to 3dp) which were used in the logistic and gamma experiments in Section 6.8.

Logistic

Table 6.2: θ values used in experiments with logistic recovery functions

	θ		
Arm 1	0.584	0.521	12.239
Arm 2	0.971	0.357	10.460
Arm 3	0.121	0.622	25.631
Arm 4	0.240	0.943	18.870
Arm 5	0.613	0.925	20.310
Arm 6	0.480	0.914	1.452
Arm 7	0.974	0.484	10.128
Arm 8	0.780	0.422	0.396
Arm 9	0.658	0.591	23.264
Arm 10	0.687	0.753	7.908

Gamma

Table 6.3: θ values used in experiments with gamma recovery functions

	θ		
Arm 1	2.068	0.249	0.508
Arm 2	5.023	0.375	0.551
Arm 3	3.657	0.470	0.772
Arm 4	0.560	0.176	0.569
Arm 5	3.901	0.747	0.500
Arm 6	0.600	0.145	0.266
Arm 7	6.482	0.522	0.554
Arm 8	13.645	0.748	0.678
Arm 9	7.365	0.562	0.288
Arm 10	2.705	0.593	0.381

6.F.3 Results for Different Lengthscales

In this section, we present results for the parametric setting where we have used different lenghtscales for the kernel of the Gaussian process in our methods. The parametric functions that we are considering are quite smooth so we choose a squared exponential kernel and used $l = 5$ in the main text, and present results here for $l = 2.5$ and $l = 7.5$. Note that in this setting looking at the smoothness of the recovery functions to inform a decision about the lengthscale is reasonable since we are comparing our algorithms to RogueUCB-Tuned of Mintz et al. (2017) which requires knowledge of the parametric family and Lipschitz constant of the recovery function.

The results for $l = 2.5$ are shown in Table 6.4 and Figure 6.12. The results for $l = 7.5$ are in Table 6.5 and Figure 6.13. From these results, we can see that in the Gamma case, our algorithms are almost invariant to the choice of l, obtaining similar results for all choices of l. In particular, for all three choices of l considered, our algorithms considerably outperform RogueUCB-Tuned of Mintz et al. (2017). In the logistic setting, there is slightly more variation in the performance of our algorithms when the lengthscale changes, although the results are still fairly similar. In this case, we see that choosing $l = 7.5$ leads to the best results for both of our algorithms. This

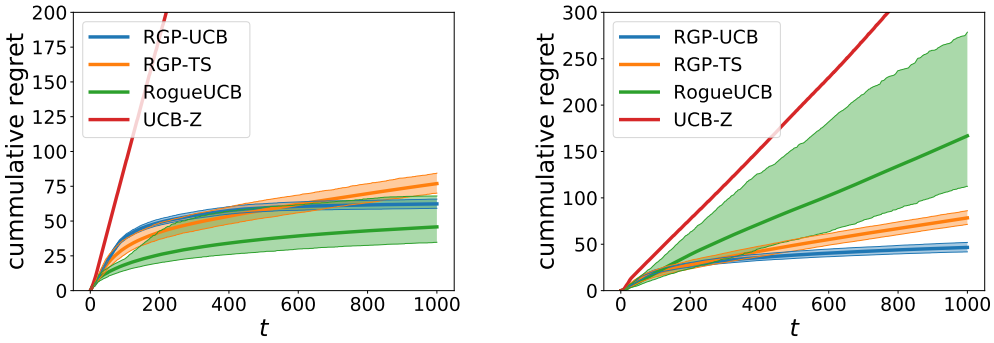
Table 6.4: Total reward at $T = 1000$ for single step experiments with parametric functions and $l = 2.5$

Setting	1RGP-UCB ($l = 2.5$)	1RGP-TS ($l = 2.5$)	RogueUCB-Tuned	UCB-Z
Logistic	448.6 (441.1,456.6)	452.5 (443.7,460.3)	446.2 (438.2,453.5)	242.6) (229.6,256.0
Gamma	145.1 (138.5, 151.5)	155.8 (148.8,162.5)	132.7 (111.0,144.5)	116.8 (108.4,125.5)

Table 6.5: Total reward at $T = 1000$ for single step experiments with parametric functions and $l = 7.5$

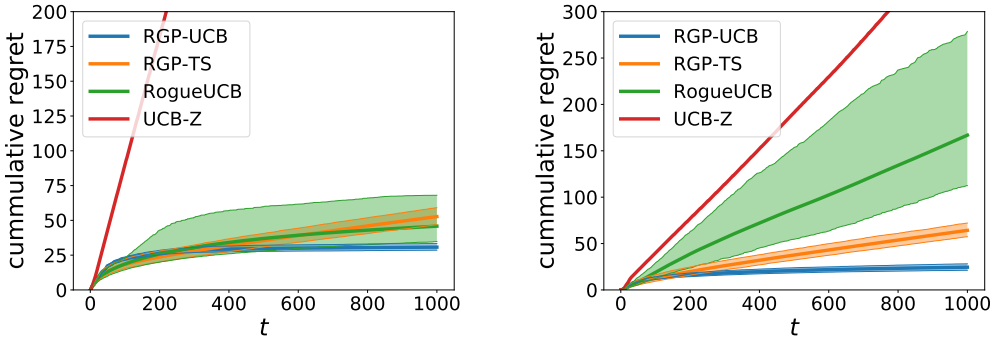
Setting	1RGP-UCB ($l = 7.5$)	1RGP-TS ($l = 7.5$)	RogueUCB-Tuned	UCB-Z
Logistic	465.1 (457.3,472.9)	465.1 (457.4,472.7)	446.2 (438.2,453.5)	242.6 (229.6,256.0)
Gamma	145.2 (139.8, 151.0)	155.8 (149.0,162.5)	132.7 (111.0,144.5)	116.8 (108.4,125.5)

is most likely due to the fact that logistic functions are quite smooth and $l = 7.5$ represents the smoothest GPs we have considered.



(a) Logistic setup, $l = 2.5$ (b) Gamma setup, $l = 2.5$

Figure 6.12: Cumulative instantaneous regret for parametric setup with $l = 2.5$



(a) Logistic setup, $l = 7.5$ (b) Gamma setup, $l = 7.5$

Figure 6.13: Cumulative instantaneous regret for parametric setup with $l = 7.5$

Chapter 7

Conclusions

In this thesis we have presented and analyzed sequential decision problems motivated by the problem of selecting questions to present to students in online education. We now summarize the contributions of each chapter and relate the work back to the motivating problems in education (see Chapter 3).

In Chapter 4, we considered the problem of constructing an adaptive sequence of questions to maximize a student's learning in a homework task of fixed length. We modeled this as the stochastic knapsack problem, where each question was an item that could be placed in the knapsack and the knapsack capacity was the duration of the homework task. The size of the question was the length of time it took the student to answer it, and the reward was the benefit to the student from answering it. For this problem, we assumed we had access to a generative model of item sizes and rewards. This is a reasonable assumption in the education context since there has been much work in the educational data mining community on constructing such models (see e.g. Corbett and Anderson (1994); Hambleton and Swaminathan (2013); Jarušek and Pelánek (2012)). Under the further assumption that the item sizes were discrete, we modeled the problem of selecting the best adaptive sequence of items as the problem of finding the best policy from a given decision tree. This was done

offline, and we were able to estimate the value of a potential policy by sampling from the generative models. The objective was to be able to output a near optimal policy after relatively few samples from the generative model.

Our algorithm `OpStoK`, presented in Chapter 4, was an optimistic planning algorithm specifically adapted to the stochastic knapsack problem. Here, instead of using discount factors, as in other optimistic planning algorithms, we directly estimated the remaining capacity. Confidence bounds on both the remaining capacity and accumulated reward were then constructed, and these were used to bound the potential reward of an extension of a partial policy. We proved that, with high probability, our algorithm returned an ϵ-optimal policy and bounded the number of samples from the generative models required for this. The `OpStoK` algorithm was an anytime algorithm and returned a good solution even if stopped early. This was demonstrated experimentally, where we also demonstrated favorable performance compared to a state of the art algorithm for the stochastic knapsack problem (Dean et al., 2008), in terms of the number of policies sampled.

The work in Chapter 5 was motivated by the issue that students do not learn from a question immediately after answering it. Instead, learning will actually take place some time after the student has answered the question, and it will not necessarily be possible to identify the individual effect of each question the student answered. Specifically, the student may answer many questions on a topic, and then after some delay, we will observe an increase in their understanding of the topic in the form of a test score or equivalent, but we will not know the individual contribution of each question to this. Modeling this as a bandit problem, we defined each question as an arm and assumed that the reward of each question was not observed by the algorithm immediately, instead it was stochastically delayed and only received as a part of an aggregated reward some time later. This aggregated reward was the summed reward of some unknown number of previous plays. We referred to this bandit problem as

'bandits with delayed, aggregated anonymous feedback'.

For the bandits with delayed, aggregated anonymous feedback problem, we presented an algorithm, ODAAF, in Chapter 5 and analyzed its performance under various assumptions about the delay distributions. Our algorithm was a rarely switching algorithm and ran in phases. In each phase, each active arm was played consecutively, thus minimizing contamination from delayed samples of other arms. From the samples received while playing a given arm, confidence bounds were constructed using the assumptions on the delay to control the bias in the observations. At the end of each phase an arm was eliminated if its upper confidence bound was less than a lower confidence bound of a different arm. The lengths of the phases were determined by the assumptions on the delay. The assumptions we made on the delay were weak, and we showed that under these assumptions the regret of our algorithm nearly matched the rate of regret of Joulani et al. (2013) for the simpler delayed feedback bandit problem (where the observations were delayed but non-anonymous, so which arm generated which reward was known). Specifically, under only the assumption that the expected delay was bounded and known, the regret of our algorithm matched that of Joulani et al. (2013) up to logarithmic factors. If we also knew that the delay was bounded with known bound, our algorithm matched the rate of Joulani et al. (2013) exactly, whereas if it had known bounded variance, we were penalized by an additive variance term in our regret. We also demonstrated these rates experimentally.

The recovering bandits problem presented in Chapter 6 aimed to capture the effect of the time between repetitions of the same question on the benefit the student gained from answering the question. Specifically, we assumed that for each question (arm) there was some unknown recovery function modeling the reward of the question as a function of the time since it was last asked. Consider the problem of teaching times tables via an online education system. Clearly, if the student has just answered a question and the same question is asked again straight away, the benefit to the

student will be less than if we wait until they have forgotten the answer and ask it again. We did not make any parametric assumptions on this recovery function, but assumed that it was smooth enough to be modeled by a Gaussian process with known kernel. We assumed that the noise on our observations was Gaussian.

In recovering bandits, the reward of each arm at a given time step depended on the entire sequence of past plays, since these determined how long it had been since each arm was played. This dependence meant that instead of just selecting one arm per time step, it was better to look ahead and select sequences of d arms that played each arm near its optimal value. In Chapter 6, we presented two algorithms for the recovering bandits problem. They both consisted of placing Gaussian Process priors on the recovery function of each arm and then updating the posterior with the observations whenever the arms were played. Since we had a GP prior and Gaussian noise, our posteriors were conjugate. These posteriors were then used to lookahead and select a sequence of d arms to play, either using a Thompson sampling or a UCB selection procedure. We showed that both these algorithms satisfied strong Bayesian regret guarantees with respect to an oracle which selects the optimal sequence of d arms. Our algorithms also performed well experimentally. Particularly, it was demonstrated in experiments that our algorithms learned to only play arms at times when the recovery function corresponded to high reward. We also considered using techniques from optimistic planning to make our Thompson sampling algorithm more computationally efficient in the case where d was large.

7.1 Further Work

In this section, we consider future directions for research relating to the material in this thesis. We begin by discussing particular extensions to the work in each chapter, and then consider more general issues arising from using bandit algorithms in education

software.

7.1.1 Optimistic Planning for the Stochastic Knapsack Problem

The algorithm `OpStoK`, proposed in Chapter 4, is an anytime algorithm for the stochastic knapsack problem when we have access to a generative model of item sizes and rewards. It was observed experimentally that the algorithm attained high reward even when it was stopped early. This is clearly a beneficial property of the algorithm, and so it would be good to obtain theoretical guarantees on the performance of the algorithm when it is stopped early. Related to this is the problem of determining the accuracy of the algorithm for a given number of samples per policy evaluated. Our guarantees give a bound on the number of samples required for a given accuracy, so it would be interesting to consider these reverse guarantees. This would allow the user to specify a number of samples, rather than a desired accuracy, and get an estimate of how accurate the algorithm would be if it was allowed this number of samples. One way to achieve this could be to use the results from the literature on best arm identification. Specifically, Theorem 1 of Gabillon et al. (2012) gives a bound on the accuracy of a best arm identification procedure similar to the one we use in Algorithm 4.1 for a specified number of samples. Alternatively, Hren and Munos (2008) provide such guarantees for optimistic planning of deterministic systems so we may be able to adapt their results to our setting.

When interested in applying the `OpStoK` algorithm to the education setting, one obvious obstacle is our assumption that item sizes (question duration) and rewards (benefit to the student of answering the question) do not depend on the previous items in the policy. Clearly, in the educational domain, this is not a realistic assumption as the benefit to a student of answering a question will depend on how many similar questions they have answered recently. The challenge of incorporating

factors like this into a model of student attainment has been considered recently in the educational data mining community (Harpstead and Aleven, 2015; Martin et al., 2011; Pelánek, 2014). These sorts of models could then be used in our algorithm to provide a generative model of the reward of a particular question given the history of past questions in the policy. This would restrict the potential for sharing samples across policies, and so would increase the sample complexity, but would allow us to capture this more realistic phenomenon. Note that the decision tree we use to model the policies would stay the same as it already captures the dependence of the total reward on the sequence in a weaker manner.

Throughout, we have assumed that the models presented in the educational data mining community have been correct, and that they are able to appropriately deal with uncertainty. We are proposing to use these models as a generative model for our algorithm, so we need to be sure that they are able to generate samples which accurately represent the true data. Therefore, a challenge when applying this algorithm to the educational domain would be to check the model output was correct, and if necessary develop our own generative models for educational data. For this a Bayesian approach may be appropriate since this would provide us with a distribution over item rewards and sizes from which to sample. A final broader open problem is whether the assumption that the item sizes are discrete (or can be discretized) can be removed. In the case of continuous item sizes, the previous decision tree representation would no longer be feasible, so a new approach may be necessary.

7.1.2 Bandits with Delayed, Aggregated Anonymous Feedback

The algorithm we presented in Chapter 5 for the delayed, aggregated anonymous feedback bandits problem is a rarely switching algorithm. In the education setting, and in many other application domains, the use of rarely switching algorithms is not practically beneficial. In particular, in education, when we model each question as

an arm, it is undesirable to repeatedly ask a student the same question. Therefore, a natural direction for future work is to consider whether, under some further assumptions on the reward or delay distribution, it is possible to develop algorithms for the delayed, aggregated anonymous feedback problem which switch arms more often. One approach to this would be to assume that the reward at each time step can be represented by a mixture model of the reward from the different arms played previously. Under the additional assumption that the reward and delay distributions were from an exponential family distribution with known parametric form, the EM algorithm could be used to obtain estimates of the parameters of the model (see (Dempster et al., 1977) for more details on the EM algorithm). Online variants of the EM algorithm have also been proposed (Cappé and Moulines, 2009; Cappé, 2011) with some theoretical guarantees on performance. One could then try to use an online EM algorithm within the bandits with delayed, aggregated anonymous feedback problem to obtain estimates of the reward parameter of each arm in the case where the algorithm switches arms frequently.

An additional algorithmic question relating to our work on the delayed, aggregated anonymous feedback problem studied in Chapter 5 is whether the 'bridge period' in our algorithm (ODAAF) can be removed. This was added in order to deal with the dependencies between arms being active and the reward they contribute to each observation. It would be interesting to see if theoretical guarantees on the performance of this algorithm could be obtained without this bridge period. A further extension to the delayed, aggregated anonymous feedback problem would be to extend the model to allow for composite rewards. This problem was studied in the adversarial setting by Cesa-Bianchi et al. (2018). In the delayed composite rewards setting, the reward R_{J_t} obtained by playing arm J_t at time t can be split into m components, $R_{J_t}^{(1)} + \cdots + R_{J_t}^{(m)}$ and each component $R_{J_t}^{(i)}$ is delayed, possibly by a different amount. At time t, the player then receives the sum of all *components* of past plays that have arrived at time t.

This is particularly relevant in the education setting since it can be assumed that the benefit of asking a particular question can be broken down into the amount of learning about a set of specific skill components (this underpins the approach of Bayesian Knowledge Tracing and many other approaches to modeling student performance, e.g. Corbett and Anderson (1994); Hambleton and Swaminathan (2013); Shahiri et al. (2015)) and the delay in learning of each skill component may be different. Lastly, to the best of our knowledge, there is currently no lower bound for the stochastic delayed feedback bandits problem that involves a delay parameter. Hence, an interesting open problem is to find a tight lower bound. This would tell us whether the additive expected delay term seen in the regret of our algorithm, and many other algorithms for delayed feedback bandits, is unavoidable.

7.1.3 Recovering Bandits

Within the approach for the recovering bandits problem proposed and analyzed in Chapter 6, there are further open questions which would be interesting to address. In particular, when selecting the number of steps to lookahead in order to define the optimal lookahead policy, we argued that since, in expectation, we see a local maxima of a GP with lengthscale ℓ every 2ℓ steps (Murray, 2016; Rasmussen and Williams, 2006), this would be a good number of steps to lookahead. It would be advantageous to formalize this intuition more, and if possible bound the expected difference in reward of an optimal policy which looks 2ℓ steps ahead and one which considers the entire horizon.

One limitation of the work presented in Chapter 6 is the assumption that the rewards must be Gaussian. The reason for this was to ensure that the posterior distributions were conjugate, so Gaussian concentration could be used to obtain upper confidence bounds and samples from the posterior. In order to obtain similar bounds for non-Gaussian noise, it would be necessary to have some guarantees on the con-

centration of a non-conjugate posterior. This is typically very challenging, although progress has been made in (van der Vaart et al., 2008a) where asymptotic convergence guarantees on the expected distance between the posterior mean and true mean were given in the non-conjugate Gaussian process classification setting (see Appendix A.4). The guarantees in Chapter 6 were also given in terms of the Bayesian regret of the algorithm. It would be interesting to consider frequentist regret guarantees as well. Here we would assume that there is a true underlying recovery curve for each arm. The results of van der Vaart et al. (2008a) are frequentist so, again, obtaining finite time results equivalent to those in (van der Vaart et al., 2008a) may be one way to get frequentist regret guarantees in the recovering bandits problem. van der Vaart and van Zanten (2011) give explicit finite time frequentist rates for the concentration of a Gaussian process posterior under conjugate Gaussian noise, so the challenge would be to combine the techniques there with those in (van der Vaart et al., 2008a) to obtain explicit finite time rates for the concentration of the posterior in the Gaussian process classification setting.

In order to apply the recovering bandits algorithms to the education setting, we would model each question as an arm. In this case, one practical extension of the recovering bandits problem would be to allow for the recovery curve to also depend on the correctness of the question. In particular, we would expect the reward of asking a student a question to depend on how long it has been since they have seen it, and also on whether they got it correct at the last attempt. One might imagine that it might be possible to achieve this by extending $Z_{j,t}$ to be a vector of two variables; the time since the question was last answered correctly, and the time since the question was last answered incorrectly. However, increasing the covariate dimension makes learning the Gaussian process more difficult, so we would need to check that our algorithm is still able to accurately learn the recovery curves in this case.

7.1.4 Bandit Problems in Online Education

Although, all of the work in this thesis was motivated by the problem of selecting questions in education software, unfortunately, we are yet to test any of the algorithms in a real life educational environment. Therefore, testing the practical performance of our approaches remains an area for further work. In particular, it would be interesting to investigate how well the recovering bandits approach to simultaneously estimating the forgetting curve and using this to decide when to give the students questions works in practice. Given the promising experimental results on simulated data (see Section 6.8), one would hope that the algorithm could be applied to the educational domain to yield good results. In order for this to happen, it will be necessary to construct a good definition of reward. As discussed in Chapter 3, this is not straightforward. Future research would therefore need to involve working with educational practitioners to come up with a good definition of reward that is both pedagogically appropriate and that yields good results when placed into a bandit algorithm.

Each problem considered in this thesis has been studied in isolation. In practice, it also would be necessary to combine these techniques in order to develop an algorithm that can deal with all these problems simultaneously. Of particular interest would be incorporating the recovering structure into either the knapsack or delayed problem.

The problems discussed in this thesis do not cover all the issues which may arise from using bandit approaches in educational software. In particular, throughout this thesis, we have tried to develop algorithms for one student individually. However, it may be beneficial to share information between students. In the bandit setting, a natural way to capture this would be to define a set of features which characterize each student and consider contextual bandit algorithms which aim to maximize some function of these features. In the recovering bandits problem, this could be achieved by increasing the covariate space of the GPs to incorporate these student features, and changing the definition of time since a question was asked to be student spe-

cific. In the bandits with delayed aggregated anonymous feedback problem, a more significant change to the algorithm would need to be made in order to use this contextual information. In the stochastic knapsack problem studied in Chapter 4, often information will be shared between students in order to define the generative models used. However, it would be interesting to investigate whether information about tree structure could also be shared between students.

There are also many other problems arising from the educational domain that would lead to interesting variants of the standard multi-armed bandit problem. In particular, in education, the sequencing of plays of the arm is important and can effect the total reward from that sequence. This was touched upon with the recovering bandits problem of Chapter 6. However, there are various other sequencing effects that would make interesting bandit problems to study. For instance, if you ask a student question A and then question B, the benefit to their learning is not guaranteed to simply be the summed benefit of asking each question in isolation. Particularly, if questions A and B are both necessary to understand a topic, their combined reward could be considerably greater than their summed individual rewards. This is related to the combinatorial bandits problem introduced by Cesa-Bianchi and Lugosi (2012). A key difference is that, in this case, we would expect the reward to be a non-linear combination of individual rewards that also depends on the sequencing of actions.

A related problem is to define a bandit model that can capture the necessity for pre-requisite questions to have been answered (correctly) before certain questions can be given. A naive approach would be to consider arms as blocks of questions, but this would not allow us to choose between different follow-up questions efficiently, nor would it have the flexibility to stop giving students a string of questions if they were seen to struggle with the first (often easiest) pre-requisite question. Other interesting areas for future research are the inclusion of revision exercises from a different topic, so that, for example, an algorithm could learn to go back and revise past topics after a

certain knowledge level has been achieved in the current topic. Conversely, the reward of asking a question (or repeating a topic) could decay with the number of times it has been previously seen. This problem has been studied in the rotting bandits problem (Levine et al., 2017) and so it would be interesting to combine this with our recovering bandits framework.

Appendix A

Useful Results and Definitions

A.1 Definitions

Definition 1 (KL Divergence, (Cover and Thomas, 2012)). *The relative entropy or* Kullback-Leibler (KL) divergence *between two probability distributions* $p(x)$ *and* $q(x)$ *on* $\mathcal{X}$ *is defined as*

$$KL(p|q) = \mathbb{E}_p\left[\log\left(\frac{p(X)}{q(X)}\right)\right].$$

Hence, for discrete distributions,

$$KL(p|q) = \sum_{x\in\mathcal{X}} p(x)\log\left(\frac{p(x)}{q(x)}\right),$$

and for continuous distributions

$$KL(p|q) = \int_{\mathcal{X}} p(x)\log\left(\frac{p(x)}{q(x)}\right)dx.$$

For some common distributions there is an exact analytic expression for the KL divergence. For $\mathcal{X} = [0,1]$ and $p(x)$ and $q(x)$ Bernoulli distributions with success parameters θ_p and θ_q respectively, $KL(p|q) = (1-\theta_p)\log\left(\frac{1-\theta_p}{1-\theta_q}\right) + \theta_p\log\left(\frac{\theta_p}{\theta_q}\right)$, while for univariate Gaussian distributions on $\mathcal{X} = \mathbb{R}$ with means μ_p and μ_q and common

variance σ^2, $KL(p|q) = \frac{(\mu_p - \mu_q)^2}{2\sigma^2}$.

Definition 2 (λ-sub-Gaussian, Boucheron et al. (2013)). *A random variable X is said to be λ-sub-Gaussian if $\mathbb{E}[X] = 0$ and for all $a > 0$,*

$$\mathbb{P}(X > a) \leq \exp\left\{ - \frac{a^2}{2\lambda^2} \right\}.$$

A.2 Inequalities

Theorem A.1 (Hoeffding's Inequality). *Let $X_1, \ldots, X_n$ be independent random variables such that $X_i \in [a_i, b_i]$ for all $i = 1, \ldots, n$. Then, for every $a > 0$,*

$$\mathbb{P}\left(\sum_{i=1}^{n} (X_i - \mathbb{E}[X_i]) \geq a \right) \leq \exp\left\{ - \frac{2a^2}{\sum_{i=1}^{n} (b_i - a_i)^2} \right\}$$

Proof. See e.g. (Boucheron et al., 2013). □

Note that a similar result holds for $X_1, \ldots, X_n$ i.i.d λ-sub-Gaussian random variables. In this case, $\mathbb{P}(\sum_{i=1}^{n} (X_i - \mathbb{E}[X_i]) \geq a) \leq \exp\{-\frac{a^2}{2\lambda^2}\}$

Theorem A.2 (Azuma-Hoeffding Inequality). *Let $X_1, \ldots X_n$ be a martingale difference sequence such that $|X_i - X_{i-1}| \leq c_i$ and $X_0 = 0$ for some positive constants c_i. Then, for any $a > 0$,*

$$\mathbb{P}\left(\sum_{i=1}^{n} > t \right) \leq \exp\left\{ - \frac{2a^2}{\sum_{i=1}^{n} c_i} \right\}$$

Proof. See e.g. (Cesa-Bianchi and Lugosi, 2006). □

Theorem A.3 (Bernstein's Inequality). *Let $X_1, \ldots, X_n$ be independent real valued random variables with $\mathbb{E}[X_i] = 0$ and $X_i \leq 1$ for all $i = 1, \ldots, n$. Define $\sigma^2 = \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}[x_i^2]$, then, for any $a > 0$,*

$$\mathbb{P}\left(\sum_{i=1}^{n} (X_i - \mathbb{E}[X_i]) \geq a \right) \leq \exp\left\{ - \frac{na^2}{2\sigma^2 + 2a/3} \right\}$$

Proof. See e.g. (Cesa-Bianchi and Lugosi, 2006). □

Theorem A.4 (Freedman's Inequality)**.** *Let* $\{Y_k\}_{k=0}^{\infty}$ *be a real-valued martingale with respect to the filtration* $\{\mathcal{F}_k\}_{k=0}^{\infty}$ *with increments* $\{Z_k\}_{k=1}^{\infty}$*:* $\mathbb{E}[Z_k|\mathcal{F}_{k-1}] = 0$ *and* $Z_k = Y_k - Y_{k-1}$*, for* $k = 1, 2, \ldots$*. Assume that the difference sequence is uniformly bounded on the right:* $Z_k \leq b$ *almost surely for* $k = 1, 2, \ldots$*. Define the predictable variation process* $W_k = \sum_{j=1}^{k} \mathbb{E}[Z_j^2|\mathcal{F}_{j-1}]$ *for* $k = 1, 2, \ldots$*. Then, for all* $t \geq 0$*,* $\sigma^2 > 0$*,*

$$\mathbb{P}\left(\exists k \geq 0 : Y_k \geq t \text{ and } W_k \leq \sigma^2\right) \leq \exp\left\{-\frac{t^2/2}{\sigma^2 + bt/3}\right\}.$$

Proof. See (Freedman, 1975). □

This result implies that if for some deterministic constant, σ^2, $W_k \leq \sigma^2$ holds almost surely, then $\mathbb{P}\left(Y_k \geq t\right) \leq \exp\left\{-\frac{t^2/2}{\sigma^2+bt/3}\right\}$ holds for any $t \geq 0$.

Theorem A.5 (Doob's Maximal Inequality)**.** *Let* $\{X_i\}_{i=0}^{n}$ *be a sub-martingale with respect to* $\{\mathcal{F}_i\}_{i=0}^{n}$*. Then for any* $a > 0$*,*

$$\mathbb{P}\left(\max_{0\leq i\leq n} X_i \geq a\right) \leq \frac{\mathbb{E}[X_n]}{\lambda}$$

Proof. See e.g. (Shiryaev, 1995). □

Theorem A.6 (Pinsker's Inequality)**.** *Let* p *and* q *be probability distributions on* $(\Omega, \mathcal{A})$*, then,*

$$\sup_{a\in\mathcal{A}} |p(a) - q(a)| \leq \sqrt{\frac{1}{2} KL(q|p)}.$$

Proof. See e.g. (Boucheron et al., 2013). □

A.3 Markov Decision Processes

A Markov decision process (Sutton and Barto, 1998; Puterman, 2014) (MDP), is a tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$ where $\mathcal{S}$ is a finite set of states, $\mathcal{A}$ is a finite set of actions, P is

a set of matrices of state transition probabilities, R is a set of reward probabilities, and $\gamma \in [0, 1]$ is a discount factor. At time t, if the MDP is in state $S_t = s \in \mathcal{S}$, if the player takes action $A_t = a \in \mathcal{A}$, they will transition to state $s' \in \mathcal{S}$ with probability $P_a[s, s'] = \mathbb{P}(S_{t+1} = s'|S_t = s, A_t = a)$ and receive reward r with probability $R_a[s, r]$. Here P_a is the matrix of transition probabilities for action a and $P_a[s, s']$ is the (s, s')th element of this, and $R_a[s, r]$ is the probability of receiving reward r after taking action a from state s.

A.4 Gaussian Processes and RKHS's

A Gaussian process (GP) represents a distribution over functions. More formally, a Gaussian process is a stochastic process such that any finite collection of the random variables has a multi-variate Gaussian density. Let f be a Gaussian process on $[0, 1]^d$. A Gaussian process is completely specified by its mean function $\mu(x) = \mathbb{E}[f(x)]$ and covariance function $k(x, x') = \mathbb{E}[(f(x)-\mu(x))(f(x')-\mu(x'))]$. The covariance function specifies the smoothness of the function. Some popular choices of covariance functions are given in Section A.4.3

A.4.1 Regression and Classification

Assume for now that we have a Gaussian process with mean 0, that is $\mu(x) = 0$ for all $x \in [0, 1]^d$. Gaussian process regression refers to the problem where for $i = 1, \ldots, n$, we observe

$$Y_i = f(x_i) + \epsilon_i$$

for $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ iid with known standard deviation σ. If we take a Bayesian approach and place a GP prior on f, the posterior is conjugate. Specifically, for $\mathbf{k}_N(z) = (k(z_1, z), \ldots, k(z_N, z))^T$ and positive semi-definite kernel matrix $\mathbf{K}_N = [k(z_i, z_j)]_{i,j=1}^N$,

the posterior mean and covariance are given by,

$$\mu_N(z) = \mathbf{k}_N(z)^T(\mathbf{K}_N + \sigma^2\mathbf{I})^{-1}\mathbf{y}_N,$$
$$k_N(z,z) = k(z,z') - \mathbf{k}_N(z)^T(\mathbf{K}_N + \sigma^2\mathbf{I})^{-1}\mathbf{k}_N(z')$$

so $\sigma_N^2(z) = k_N(z,z)$. Then, for any $z \in \mathcal{Z}$, the posterior distribution of $f(z)$ is $\mathcal{N}(\mu_N(z), \sigma_N^2(z))$.

In the classification setting, our observations take the form,

$$Y_i \sim \text{Bern}(\phi(f(x_i)))$$

where $\text{Bern}(\theta)$ represents the Bernoulli distribution with success probability θ and $\phi(\cdot)$ is some link function. Normally, ϕ is taken to be the logistic link so $\phi(z) = (1+\exp(-z))^{-1}$, or the probit link in which case $\phi(z) = \Phi(z)$ for $\Phi(\cdot)$ the standard Gaussian cdf. In the classification case, the Gaussian process prior is non-conjugate so there exists no closed form expressions for the posterior mean and covariance functions. Instead these should be found using MCMC methods or approximations (Nickisch and Rasmussen, 2008). See (Rasmussen and Williams, 2006) for more details on Gaussian process regression and classification.

A.4.2 RKHS

An RKHS or *Reproducing Kernel Hilbert Space* is a Hilbert space $\mathbb{H}$ of real functions defined on an index set $\mathcal{X}$ endowed with an inner product $\langle\cdot,\cdot\rangle_{\mathcal{H}}$ such that there exists a function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ satisfying:

(i) for every $x \in \mathcal{X}$, $k(x,x')$ is a function of x' and is in $\mathcal{H}$,

(ii) k has the reproducing property, $\langle f(\cdot), k(\cdot,x)\rangle_{\mathbb{H}} = f(x)$.

Note that also $k(\cdot,x) \in \mathbb{H}$ and $k(x',\cdot) \in \mathbb{H}$, and that also $\langle k(x,\cdot), k(x',\cdot)\rangle_{\mathcal{H}} = k(x,x')$.

The RKHS attached to a Gaussian process with covariance function k is the completion $\mathbb{H}$ of the linear space of all functions

$$x \to \sum_{i=1}^{m} a_i k(s_i, x) \qquad \text{such that } a_1, \ldots, a_m \in \mathbb{R}, s_1, \ldots, s_m \in \mathcal{X}, m \in \mathbb{N}$$

relative to the norm,

$$\left\langle \sum_{i=1}^{m} a_i k(s_i, \cdot), \sum_{j=1}^{r} b_j k(t_j, \cdot) \right\rangle_{\mathbb{H}} = \sum_{i=1}^{m} \sum_{j=1}^{r} a_i b_j k(s_i, t_j).$$

Intuitively it is the set of all linear combinations of kernel functions. See (van der Vaart et al., 2008b) for more details.

A.4.3 Covariance Functions

One popular choices of covariance function in the machine learning literature is the *squared exponential* covariance function with lengthscale $l > 0$,

$$k(x, x') = \exp\left\{ -\frac{(x - x')^2}{2l^2} \right\}.$$

Intuitively the lengthscale measures the smoothness of the Gaussian process.

Another common covariance function is the Matérn covariance function with lengthscale $l > 0$ and positive parameter $\nu > 0$,

$$k(x, x') = \frac{2^{l-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}|x - x'|}{l} \right)^{\nu} K_\nu \left(\frac{\sqrt{2\nu}|x - x'|}{l} \right)$$

where K_ν is a modified Bessel function. Often we choose $\nu = 3/2$, in which case,

$$k(x, x') = \left(1 + \frac{\sqrt{3}|x - x'|}{l} \right) \exp\left\{ -\frac{\sqrt{3}|x - x'|}{l} \right\},$$

or $\nu = 5/2$ in which case,

$$k(x, x') = \left(1 + \frac{\sqrt{5}|x - x'|}{l} + \frac{\sqrt{5}(x - x'|)^2}{3l^2}\right) \exp\left\{ - \frac{\sqrt{5}|x - x'|}{l} \right\}.$$

There are several other covariance functions which may be used, see (Rasmussen and Williams, 2006) for details.

Bibliography

Abbasi-Yadkori, Y., Pál, D., and Szepesvári, C. (2011). Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 2312–2320.

Abeille, M. and Lazaric, A. (2017). Linear thompson sampling revisited. *Electronic Journal of Statistics*, 11(2):5165–5197.

Agrawal, R. (1995). Sample mean based index policies by o (log n) regret for the multi-armed bandit problem. *Advances in Applied Probability*, 27(4):1054–1078.

Agrawal, R., Hedge, M., and Teneketzis, D. (1988). Asymptotically efficient adaptive allocation rules for the multiarmed bandit problem with switching cost. *IEEE Transactions on Automatic Control*, 33(10):899–906.

Agrawal, S. and Devanur, N. (2014). Bandits with concave rewards and convex knapsacks. In *Conference on Economics and Computation*, pages 989–1006.

Agrawal, S. and Devanur, N. (2016). Linear contextual bandits with knapsacks. In *Advances in Neural Information Processing Systems*, pages 3450–3458.

Agrawal, S., Devanur, N. R., and Li, L. (2016). An efficient algorithm for contextual bandits with knapsacks, and an extension to concave objectives. In *Conference on Learning Theory*, pages 4–18.

Agrawal, S. and Goyal, N. (2012). Analysis of thompson sampling for the multi-armed bandit problem. In *Conference on Learning Theory*.

Agrawal, S. and Goyal, N. (2013a). Further optimal regret bounds for thompson sampling. In *International Conference on Artificial Intelligence and Statistics*, pages 99–107.

Agrawal, S. and Goyal, N. (2013b). Thompson sampling for contextual bandits with linear payoffs. In *International Conference on Machine Learning*, pages 127–135.

Alshammari, M., Anane, R., and Hendley, R. J. (2014). Adaptivity in e-learning systems. In *IEEE International Conference on Complex, Intelligent and Software Intensive Systems*, pages 79–86.

Andersen, E., Gulwani, S., and Popovic, Z. (2013). A trace-based framework for analyzing and synthesizing educational progressions. In *SIGCHI Conference on Human Factors in Computing Systems*, pages 773–782.

Antonova, R., Runde, J., Lee, M. H., and Brunskill, E. (2016). Automatically learning to teach to the learning objectives. In *ACM Conference on Learning@ Scale*, pages 317–320.

Arora, R., Dekel, O., and Tewari, A. (2012). Online bandit learning against an adaptive adversary: from regret to policy regret. *International Conference on Machine Learning*.

Audibert, J.-Y. and Bubeck, S. (2009). Minimax policies for adversarial and stochastic bandits. In *Conference on Learning Theory*, pages 217–226.

Audibert, J.-Y. and Bubeck, S. (2010). Best arm identification in multi-armed bandits. In *Conference on Learning Theory*, pages 41–53.

Auer, P. (2002). Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422.

Auer, P., Cesa-Bianchi, N., and Fischer, P. (2002a). Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256.

Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E. (1995). Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *IEEE Foundations of Computer Science*, page 322.

Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E. (2002b). The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77.

Auer, P. and Chiang, C.-K. (2016). An algorithm with nearly optimal pseudo-regret for both stochastic and adversarial bandits. In *Conference on Learning Theory*, pages 116–120.

Auer, P. and Ortner, R. (2010). UCB revisited: Improved regret bounds for the stochastic multi-armed bandit problem. *Periodica Mathematica Hungarica*, 61(1-2):55–65.

Averell, L. and Heathcote, A. (2011). The form of the forgetting curve and the fate of memories. *Journal of Mathematical Psychology*, 55(1):25–35.

Azuma, K. (1967). Weighted sums of certain dependent random variables. *Tohoku Mathematical Journal, Second Series*, 19(3):357–367.

Badanidiyuru, A., Kleinberg, R., and Slivkins, A. (2013). Bandits with knapsacks. In *IEEE Annual Symposium on Foundations of Computer Science*.

Bahrick, H. P. and Phelphs, E. (1987). Retention of spanish vocabulary over 8 years. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 13(2):344.

Bellman, R. (1956). A problem in the sequential design of experiments. *Sankhyā: The Indian Journal of Statistics (1933-1960)*, 16(3/4):221–229.

Besbes, O., Gur, Y., and Zeevi, A. (2014). Stochastic multi-armed-bandit problem with non-stationary rewards. In *Advances in Neural Information Processing Systems*, pages 199–207.

Bhalgat, A., Goel, A., and Khanna, S. (2011). Improved approximation results for stochastic knapsack problems. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 1647–1665.

Biedka, T., Dreger, B., Kachlicki, J., Krawiec, K., Plazewski, M., Wierzejewski, P., and Wozniak, P. (1998). Employing a neural network to solving the repetition spacing problem. https://www.supermemo.com/english/ol/nn.htm. Accessed: 2018-21-10.

Bogunovic, I., Scarlett, J., and Cevher, V. (2016). Time-varying gaussian process bandit optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 314–323.

Boucheron, S., Lugosi, G., and Massart, P. (2013). *Concentration inequalities: A nonasymptotic theory of independence*. Oxford University Press.

Bouneffouf, D. and Feraud, R. (2016). Multi-armed bandit problem with known trend. *Neurocomputing*, 205:16–21.

Browne, C. B., Powley, E., Whitehouse, D., Lucas, S. M., Cowling, P. I., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S., and Colton, S. (2012). A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 4(1):1–43.

Bubeck, S. and Cesa-Bianchi, N. (2012). *Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems.* Foundations and Trends in Machine Learning.

Bubeck, S. and Liu, C.-Y. (2013). Prior-free and prior-dependent regret bounds for thompson sampling. In *Advances in Neural Information Processing Systems*, pages 638–646.

Bubeck, S. and Munos, R. (2010). Open loop optimistic planning. In *Conference on Learning Theory*, pages 477–489.

Bull, A. D. (2011). Convergence rates of efficient global optimization algorithms. *Journal of Machine Learning Research*, 12(Oct):2879–2904.

Burnetas, A. N., Kanavetas, O., and Katehakis, M. N. (2015). Asymptotically optimal multi-armed bandit policies under a cost constraint. *arXiv preprint arXiv:1509.02857*.

Busoniu, L. and Munos, R. (2012). Optimistic planning for markov decision processes. In *International Conference on Artificial Intelligence and Statistics*, pages 182–189.

Cappé, O. (2011). Online expectation maximisation. In *Mixtures: Estimation and Applications*, pages 31–53. Wiley Online Library.

Cappé, O., Garivier, A., Maillard, O.-A., Munos, R., and Stoltz, G. (2013). Kullback–Leibler upper confidence bounds for optimal sequential allocation. *The Annals of Statistics*, 41(3):1516–1541.

Cappé, O. and Moulines, E. (2009). On-line expectation–maximization algorithm for latent data models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(3):593–613.

Cepeda, N. J., Pashler, H., Vul, E., Wixted, J. T., and Rohrer, D. (2006). Distributed practice in verbal recall tasks: A review and quantitative synthesis. *Psychological Bulletin*, 132(3):354.

Cesa-Bianchi, N., Gentile, C., and Mansour, Y. (2018). Nonstochastic bandits with composite anonymous feedback. In *Conference On Learning Theory*, pages 750–773.

Cesa-Bianchi, N., Gentile, C., Mansour, Y., and Minora, A. (2016). Delay and cooperation in nonstochastic bandits. In *Conference on Learning Theory*, pages 605–622.

Cesa-Bianchi, N. and Lugosi, G. (2006). *Prediction, learning, and games*. Cambridge university press.

Cesa-Bianchi, N. and Lugosi, G. (2012). Combinatorial bandits. *Journal of Computer and System Sciences*, 78(5):1404–1422.

Chapelle, O. and Li, L. (2011). An empirical evaluation of Thompson sampling. In *Advances in Neural Information Processing Systems*, pages 2249–2257.

Chen, S., Lin, T., King, I., Lyu, M. R., and Chen, W. (2014). Combinatorial pure exploration of multi-armed bandits. In *Advances in Neural Information Processing Systems*, pages 379–387.

Chu, W., Li, L., Reyzin, L., and Schapire, R. (2011). Contextual bandits with linear payoff functions. In *International Conference on Artificial Intelligence and Statistics*, pages 208–214.

Clement, B., Roy, D., Oudeyer, P.-Y., and Lopes, M. (2014). Online optimization of teaching sequences with multi-armed bandits. In *International Conference on Educational Data Mining*.

Clement, B., Roy, D., Oudeyer, P.-Y., and Lopes, M. (2015). Multi-armed bandits for intelligent tutoring systems. *Journal of Educational Data Mining*, 7(2).

Contal, E. and Vayatis, N. (2016). Stochastic process bandits: Upper confidence bounds algorithms via generic chaining. *arXiv preprint arXiv:1602.04976*.

Cooper, H., Robinson, J. C., and Patall, E. A. (2006). Does homework improve academic achievement? a synthesis of research, 1987–2003. *Review of Educational Research*, 76(1):1–62.

Coquelin, P.-A. and Munos, R. (2007). Bandit algorithms for tree search. In *International Conference on Uncertainty in Artificial Intelligence*, pages 67–74.

Corbett, A. T. and Anderson, J. R. (1994). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4(4):253–278.

Cortes, C., DeSalvo, G., Kuznetsov, V., Mohri, M., and Yang, S. (2017). Discrepancy-based algorithms for non-stationary rested bandits. *arXiv preprint arXiv:1710.10657*.

Cover, T. M. and Thomas, J. A. (2012). *Elements of information theory*. John Wiley & Sons.

Cowan, N. (2008). What are the differences between long-term, short-term, and working memory? *Progress in Brain Research*, 169:323–338.

Dani, V., Hayes, T. P., and Kakade, S. M. (2008). Stochastic linear optimization under bandit feedback. In *Conference on Learning Theory*.

Dantzig, G. B. (1957). Discrete-variable extremum problems. *Operations Research*, 5(2):266–288.

Dean, B. C., Goemans, M. X., and Vondrák, J. (2008). Approximating the stochastic knapsack problem: The benefit of adaptivity. *Mathematics of Operations Research*, 33(4):945–964.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (methodological)*, pages 1–38.

Dempster, F. N. (1989). Spacing effects and their implications for theory and practice. *Educational Psychology Review*, 1(4):309–330.

Desautels, T., Krause, A., and Burdick, J. (2014). Parallelizing exploration-exploitation tradeoffs in Gaussian process bandit optimization. *Journal of Machine Learning Research*, 15(1):3873–3923.

Devroye, L. and Lugosi, G. (2001). *Combinatorial methods in density estimation*. Springer.

Doob, J. L. (1953). *Stochastic processes*. John Wiley & Sons.

Dudai, Y., Karni, A., and Born, J. (2015). The consolidation and transformation of memory. *Neuron*, 88(1):20–32.

Dudik, M., Hsu, D., Kale, S., Karampatziakis, N., Langford, J., Reyzin, L., and Zhang, T. (2011). Efficient optimal learning for contextual bandits. In *Conference on Uncertainty in Artificial Intelligence*, pages 169–178.

Ebbinghaus, H. (2013). Memory: A contribution to experimental psychology. *Annals of Neurosciences*, 20(4):155.

Edge, D., Fitchett, S., Whitney, M., and Landay, J. (2012). Memreflex: adaptive flashcards for mobile microlearning. In *International Conference on Human-Computer Interaction with Mobile Devices and Services*, pages 431–440.

Erraqabi, A., Lazaric, A., Valko, M., Brunskill, E., and Liu, Y.-E. (2017). Trading off rewards and errors in multi-armed bandits. In *International Conference on Artificial Intelligence and Statistics*.

Even-Dar, E., Mannor, S., and Mansour, Y. (2006). Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *The Journal of Machine Learning Research*, 7:1079–1105.

Filippi, S., Cappé, O., Garivier, A., and Szepesvári, C. (2010). Parametric bandits: The generalized linear case. In *Advances in Neural Information Processing Systems*, pages 586–594.

Frazier, P. I. (2018). A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*.

Freedman, D. A. (1975). On tail probabilities for martingales. *The Annals of Probability*, 3(1):100–118.

Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139.

Gabillon, V., Ghavamzadeh, M., and Lazaric, A. (2012). Best arm identification: A unified approach to fixed budget and fixed confidence. In *Advances in Neural Information Processing Systems*, pages 3212–3220.

Gabillon, V., Lazaric, A., Ghavamzadeh, M., Ortner, R., and Barlett, P. (2016). Improved learning complexity in combinatorial pure exploration bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 1004–1012.

Galloway, M., Conner, J., and Pope, D. (2013). Nonacademic effects of homework in privileged, high-performing high schools. *The Journal of Experimental Education*, 81(4):490–510.

Garivier, A. and Moulines, E. (2011). On upper-confidence bound policies for non-stationary bandit problems. In *International Conference on Algorithmic Leaning Theory*.

Gelman, A., Stern, H. S., Carlin, J. B., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013). *Bayesian data analysis*. Chapman and Hall/CRC.

Gilks, W. R., Richardson, S., and Spiegelhalter, D. (1995). *Markov chain Monte Carlo in practice*. CRC press.

Gittins, J., Glazebrook, K., and Weber, R. (2011). *Multi-armed bandit allocation indices*. John Wiley & Sons.

Gittins, J. C. (1979). Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 148–177.

GPy (2012). GPy: A gaussian process framework in python. http://github.com/SheffieldML/GPy.

Grünewälder, S., Audibert, J.-Y., Opper, M., and Shawe-Taylor, J. (2010). Regret bounds for gaussian process bandit problems. In *International Conference on Artificial Intelligence and Statistics*, pages 273–280.

Hambleton, R. K. and Swaminathan, H. (2013). *Item response theory: Principles and applications*. Springer Science & Business Media.

Harpstead, E. and Aleven, V. (2015). Using empirical learning curve analysis to inform design in an educational game. In *Annual Symposium on Computer-Human Interaction in Play*, pages 197–207.

Hartland, C., Gelly, S., Baskiotis, N., Teytaud, O., and Sebag, M. (2006). Multi-armed bandit, dynamic environments and meta-bandits. *Hal Archive: hal-00113668*.

Heidari, H., Kearns, M., and Roth, A. (2016). Tight policy regret bounds for improving and decaying bandits. In *International Joint Conference on Artificial Intelligence*, pages 1562–1570.

Hinkley, D. V. and Hinkley, E. A. (1970). Inference about the change-point in a sequence of binomial variables. *Biometrika*, 57(3):477–488.

Hren, J.-F. and Munos, R. (2008). Optimistic planning of deterministic systems. In *European Workshop on Reinforcement Learning*, pages 151–164.

Jaksch, T., Ortner, R., and Auer, P. (2010). Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11(Apr):1563–1600.

Jarušek, P., Klusácek, M., and Pelánek, R. (2013). Modeling students' learning and variability of performance in problem solving. *Educational Data Mining*, pages 256–259.

Jarušek, P. and Pelánek, R. (2012). Analysis of a simple model of problem solving times. In *International Conference on Intelligent Tutoring Systems*, pages 379–388. Springer.

Joulani, P., György, A., and Szepesvári, C. (2013). Online learning under delayed feedback. In *International Conference on Machine Learning*, pages 1453–1461.

Jun, K.-S., Bhargava, A., Nowak, R., and Willett, R. (2017). Scalable generalized linear bandits: Online computation and hashing. In *Advances in Neural Information Processing Systems*, pages 99–109.

Kaufmann, E., Cappé, O., and Garivier, A. (2012a). On bayesian upper confidence bounds for bandit problems. In *International Conference on Artificial Intelligence and Statistics*, pages 592–600.

Kaufmann, E., Korda, N., and Munos, R. (2012b). Thompson sampling: An asymptotically optimal finite-time analysis. In *International Conference on Algorithmic Learning Theory*, pages 199–213. Springer.

Kocsis, L. and Szepesvári, C. (2006). Bandit based monte-carlo planning. In *European Conference on Machine Learning*, pages 282–293.

Korda, N., Kaufmann, E., and Munos, R. (2013). Thompson sampling for 1-dimensional exponential family bandits. In *Advances in Neural Information Processing Systems*, pages 1448–1456.

Krause, A. and Ong, C. S. (2011). Contextual gaussian process bandit optimization. In *Advances in Neural Information Processing Systems*, pages 2447–2455.

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

Lan, A. S. and Baraniuk, R. G. (2016). A contextual bandits framework for personalized learning action selection. In *Educational Data Mining*, pages 424–429.

Langford, J. and Zhang, T. (2008). The epoch-greedy algorithm for multi-armed bandits with side information. In *Advances in Neural Information Processing Systems*, pages 817–824.

Lattimore, T. (2016). Regret analysis of the finite-horizon gittins index strategy for multi-armed bandits. In *Conference on Learning Theory*, pages 1214–1245.

Lattimore, T. and Szepesvari, C. (2017). The end of optimism? an asymptotic analysis of finite-armed linear bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 728–737.

Lattimore, T. and Szepesvári, C. (2018). *Bandit Algorithms*. Cambridge University Press, draft version 1016 edition.

Lee, J. I. and Brunskill, E. (2012). The impact on individualizing student models on necessary practice opportunities. *Educational Data Mining*.

Leitner, S. (1995). *So lernt man lernen.* Herder.

Levine, N., Crammer, K., and Mannor, S. (2017). Rotting bandits. In *Advances in Neural Information Processing Systems*, pages 3077–3086.

Li, L., Chu, W., Langford, J., and Schapire, R. (2010). A contextual-bandit approach to personalized news article recommendation. In *International Conference on World Wide Web*, pages 661–670.

Li, L., Lu, Y., and Zhou, D. (2017). Provably optimal algorithms for generalized linear contextual bandits. In *International Conference on Machine Learning*, pages 2071–2080.

Lindsey, R. V., Mozer, M. C., Huggins, W. J., and Pashler, H. (2013). Optimizing instructional policies. In *Advances in Neural Information Processing Systems*, pages 2778–2786.

Liu, Y.-E., Mandel, T., Brunskill, E., and Popovic, Z. (2014). Trading off scientific knowledge and user learning with multi-armed bandits. In *International Conference on Educational Data Mining*.

Lomas, J. D., Forlizzi, J., Poonwala, N., Patel, N., Shodhan, S., Patel, K., Koedinger, K., and Brunskill, E. (2016). Interface design optimization as a multi-armed bandit problem. In *CHI Conference on Human Factors in Computing Systems*, pages 4142–4153.

Luckin, R. (2001). Designing children's software to ensure productive interactivity through collaboration in the zone of proximal development (zpd). *Information Technology in Childhood Education Annual*, 2001(1):57–85.

Ma, Y., Agnihotri, L., Baker, R. S., and Mojarad, S. (2016). Effect of student ability and question difficulty on duration. In *Educational Data Mining*, pages 135–142.

Mandel, T., Liu, Y.-E., Brunskill, E., and Popovic, Z. (2015). The queue method: Handling delay, heuristics, prior data, and evaluation in bandits. In *AAAI Conference on Artificial Intelligence*, pages 2849–2856.

Martin, B., Mitrovic, A., Koedinger, K. R., and Mathan, S. (2011). Evaluating and improving adaptive educational systems with learning curves. *User Modeling and User-Adapted Interaction*, 21(3):249–283.

Matiisen, T., Oliver, A., Cohen, T., and Schulman, J. (2017). Teacher-student curriculum learning. *arXiv preprint arXiv:1707.00183*.

May, B. C., Korda, N., Lee, A., and Leslie, D. S. (2012). Optimistic bayesian sampling in contextual-bandit problems. *Journal of Machine Learning Research*, 13(Jun):2069–2106.

May, B. C. and Leslie, D. S. (2011). Simulation studies in optimistic bayesian sampling in contextual-bandit problems. *Statistics Group, Department of Mathematics, University of Bristol*.

Mellor, J. and Shapiro, J. (2013). Thompson sampling in switching environments with bayesian online change detection. In *International Conference on Artificial Intelligence and Statistics*, pages 442–450.

Mintz, Y., Aswani, A., Kaminsky, P., Flowers, E., and Fukuoka, Y. (2017). Nonstationary bandits with habituation and recovery dynamics. *arXiv preprint arXiv:1707.08423*.

Morton, D. P. and Wood, R. K. (1998). On a stochastic knapsack problem and generalizations. In *Advances in Computational and Stochastic Optimization, Logic Programming, and Heuristic Search*, pages 149–168. Springer.

Mu, T., Goel, K., and Brunskill, E. (2017). Combining adaptivity with progression ordering for intelligent tutoring systems. In *NIPS'17 Workshop: Teaching Machines, Robots, and Humans*.

Mu, T., Wang, S., Andersen, E., and Brunskill, E. (2018). Combining adaptivity with progression ordering for intelligent tutoring systems. In *ACM Conference on Learning @ Scale*, page 15.

Munos, R. et al. (2014). From bandits to monte-carlo tree search: The optimistic principle applied to optimization and planning. *Foundations and Trends® in Machine Learning*, 7(1):1–129.

Murray, I. (2016). Gaussian processes and kernels. https://www.inf.ed.ac.uk/teaching/courses/mlpr/2016/notes/w7c_gaussian_process_kernels.pdf.

Neu, G. (2015). Explore no more: Improved high-probability regret bounds for non-stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 3168–3176.

Neu, G., Antos, A., György, A., and Szepesvári, C. (2010). Online Markov decision processes under bandit feedback. In *Advances in Neural Information Processing Systems*, pages 1804–1812.

Neu, G., György, A., Szepesvári, C., and Antos, A. (2014). Online markov decision processes under bandit feedback. *IEEE Transactions on Automatic Control*, 59(3):676–691.

Nickisch, H. and Rasmussen, C. E. (2008). Approximations for binary gaussian process classification. *Journal of Machine Learning Research*, 9(Oct):2035–2078.

Ortner, R., Ryabko, D., Auer, P., and Munos, R. (2012). Regret bounds for restless markov bandits. In *International Conference on Algorithmic Learning Theory*, pages 214–228.

Page, E. (1955). A test for a change in a parameter occurring at an unknown point. *Biometrika*, 42(3/4):523–527.

Papadimitriou, C. H. and Tsitsiklis, J. N. (1999). The complexity of optimal queuing network control. *Mathematics of Operations Research*, 24(2):293–305.

Pavlik, P. I. and Anderson, J. R. (2008). Using a model to compute the optimal schedule of practice. *Journal of Experimental Psychology: Applied*, 14(2):101.

Pelánek, R. (2014). Application of time decay functions and the elo system in student modeling. In *Educational Data Mining*.

Perchet, V., Rigollet, P., Chassang, S., and Snowberg, E. (2016). Batched bandit problems. *The Annals of Statistics*, 44(2):660–681.

Puterman, M. L. (2014). *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Rafferty, A. N., Brunskill, E., Griffiths, T. L., and Shafto, P. (2011). Faster teaching by pomdp planning. In *International Conference on Artificial Intelligence in Education*, pages 280–287.

Raj, V. and Kalyani, S. (2017). Taming non-stationary bandits: A bayesian approach. *arXiv preprint arXiv:1707.09727*.

Rasmussen, C. E. and Williams, C. K. (2006). Gaussian processes for machine learning. MIT press.

Reddy, S., Labutov, I., Banerjee, S., and Joachims, T. (2016). Unbounded human learning: Optimal scheduling for spaced repetition. In *ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1815–1824.

Rigollet, P. and Zeevi, A. (2010). Nonparametric bandits with covariates. *Conference on Learning Theory*, page 54.

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

Russo, D. and Van Roy, B. (2014). Learning to optimize via posterior sampling. *Mathematics of Operations Research*, 39(4):1221–1243.

Russo, D. and Van Roy, B. (2016). An information-theoretic analysis of thompson sampling. *The Journal of Machine Learning Research*, 17(1):2442–2471.

Scarlett, J. (2018). Tight regret bounds for bayesian optimization in one dimension. In *International Conference on Machine Learning*.

Scarlett, J., Bogunovic, I., and Cevher, V. (2017). Lower bounds on regret for noisy gaussian process bandit optimization. In *Conference on Learning Theory*, pages 1723–1742.

Scott, S. L. (2010). A modern bayesian look at the multi-armed bandit. *Applied Stochastic Models in Business and Industry*, 26(6):639–658.

Segal, A., David, Y. B., Williams, J. J., Gal, K., and Shalom, Y. (2018). Combining difficulty ranking with multi-armed bandits to sequence educational content. In *International Conference on Artificial Intelligence in Education*, pages 317–321.

Seldin, Y. and Lugosi, G. (2017). An improved parametrization and analysis of the exp3++ algorithm for stochastic and adversarial bandits. In *Conference on Learning Theory*, pages 1743–1759.

Shahiri, A. M., Husain, W., et al. (2015). A review on predicting student's performance using data mining techniques. *Procedia Computer Science*, 72:414–422.

Shekhar, S., Javidi, T., et al. (2018). Gaussian process bandits with adaptive discretization. *Electronic Journal of Statistics*, 12(2):3829–3874.

Shiryaev, A. N. (1995). *Probability (2nd Ed.)*. Graduate Texts in Mathematics. Springer-Verlag.

Slivkins, A. and Upfal, E. (2008). Adapting to a changing environment: the brownian restless bandits. In *Conference on Learning Theory*, pages 343–354.

Srinivas, N., Krause, A., Kakade, S., and Seeger, M. (2010). Gaussian process optimization in the bandit setting: No regret and experimental design. In *International Conference on Machine Learning*.

Steinberg, E. and Parks, M. (1979). A preference order dynamic program for a knapsack problem with stochastic rewards. *Journal of the Operational Research Society*, pages 141–147.

Sutton, R. S. and Barto, A. G. (1998). *Introduction to reinforcement learning*, volume 135. MIT press Cambridge.

Szita, I. and Szepesvári, C. (2011). Agnostic KWIK learning and efficient approximate reinforcement learning. In *Conference on Learning Theory*, pages 739–772.

Szörényi, B., Kedenburg, G., and Munos, R. (2014). Optimistic planning in markov decision processes using a generative model. In *Advances in Neural Information Processing Systems*, pages 1035–1043.

Theocharous, G., Beckwith, R., Butko, N., and Philipose, M. (2009). Tractable pomdp planning algorithms for optimal teaching in "spais". In *IJCAI PAIR Workshop*.

Thompson, W. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3–4):285–294.

van der Vaart, A. and van Zanten, H. (2011). Information rates of nonparametric gaussian process methods. *Journal of Machine Learning Research*, 12(Jun):2095–2119.

van der Vaart, A. W., van Zanten, J. H., et al. (2008a). Rates of contraction of posterior distributions based on gaussian process priors. *The Annals of Statistics*, 36(3):1435–1463.

van der Vaart, A. W., van Zanten, J. H., et al. (2008b). Reproducing kernel hilbert spaces of gaussian priors. In *Pushing the limits of contemporary statistics: contributions in honor of Jayanta K. Ghosh*, pages 200–222. Institute of Mathematical Statistics.

Vernade, C., Cappé, O., and Perchet, V. (2017). Stochastic bandit models for delayed conversions. In *Conference on Uncertainty in Artificial Intelligence*.

Vernade, C., Carpentier, A., Zappella, G., Ermis, B., and Brueckner, M. (2018). Contextual bandits under delayed feedback. *arXiv preprint arXiv:1807.02089*.

Wang, Z., Shakibi, B., Jin, L., and Freitas, N. (2014). Bayesian multi-scale optimistic optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 1005–1014.

Wang, Z., Zhou, B., and Jegelka, S. (2016). Optimization as estimation with gaussian processes in bandit settings. In *International Conference on Artificial Intelligence and Statistics*, pages 1022–1031.

Whittle, P. (1988). Restless bandits: Activity allocation in a changing world. *Journal of Applied Probability*, 25(A):287–298.

Williams, D. (1991). *Probability with martingales*. Cambridge University Press.

Williams, J. J., Kim, J., Rafferty, A., Maldonado, S., Gajos, K. Z., Lasecki, W. S., and Heffernan, N. (2016). Axis: Generating explanations at scale with learnersourcing and machine learning. In *ACM Conference on Learning@ Scale*, pages 379–388.

Wozniak, P. and Gorzelanczyk, E. J. (1994). Optimization of repetition spacing in the practice of learning. *Acta Neurobiologiae Experimentalis*, 54:59–59.

Xu, J., Xing, T., and Van Der Schaar, M. (2016). Personalized course sequence recommendations. *IEEE Transactions on Signal Processing*, 64(20):5340–5352.

Yi, J., Hsieh, C.-J., Varshney, K. R., Zhang, L., and Li, Y. (2017). Scalable demand-aware recommendation. In *Advances in Neural Information Processing Systems*, pages 2409–2418.