Models of Intelligence Operations

Jak Marshall, M.Sci (Hons). M.Res

Submitted for the degree of Doctor of Philosophy

at Lancaster University,

March 2016.

Abstract

It is vital to modern intelligence operations that the cycle of gathering, analysing and acting upon intelligence is as efficient as possible in the face of an ever increasing volume of available information. The collection, processing and subsequent analysis aspect of the intelligence cycle is modelled as a novel finite horizon Bayesian stochastic dynamic programming problem, namely the multi-armed bandit allocation (MABA) problem. The MABA framework models the efforts of a processor to search for intelligence items of the highest importance by making sequential samples from a collection of intelligence sources. Through Bayesian learning the processor learns about the importance distributions of the available sources over time, select a source from which to sample at each decision epoch, and decides whether or not to allocate sampled items for analysis. For source selection, a novel Lagrangian based index heuristic is developed and its performance is compared to existing index heuristics including knowledge gradient and Thompson sampling methods. The allocation policy is handled by thresholds which act as Lagrangian multipliers of the original MABA problem. Both a discrete Dirichlet-Multinomial and a continuous Exponential-Gamma-Gamma implementation of the MABA problem are developed, where the latter also models uncertainty in the processor's own ability to accurately assess the importance of sampled items.

Acknowledgements

Producing this thesis has been an incredible journey, as most PhDs inevitably become to those doing them and I have the Herculean efforts of Jonathan Tawn, Idris Eckley and Kevin Glazebrook (who I have a great deal to thank for besides what I write here) for creating STOR-i in the first place and for recruiting me for its first cohort. Well done on getting the program funded for a second time, by the way!

My supervisory team have been wonderful. Professor Glazebrook's strategic direction and a career's experience of student supervision have been a godsend. Roberto Szetchman and Chris Kirkbride have spent countless hours on Skype to provide greatly appreciated regular feedback and discussion on the project and have often accomodated obstacles such as time difference and my own full time employment to help me. I couldn't ask for a more accommodating and compassionate set of supervisors.

The Naval Postgraduate School's OR department have my thanks for supporting the project and for their hospitality and kindness during my three visits to their campus. I also thank Terry James for enduring quite possibly the worst transatlantic journey I've ever experienced and Hailey for driving us both to Manchester for a beer when our flight was delayed! It's not the highest priority item on the list of reasons to thank Professor Glazebrook, but 38 hours of travel time made the Fisherman's Wharf meal that he treated us to when we landed in Monterey pretty unforgettable so I thank him for that.

Lancaster's Department of Maths and Stats deserves heaps (heaps!) of thanks

for its support over the years, even before the PhD started. Being able to earn some money through lab teaching and talking at open days really made making ends meet that much easier. In particular, Julia Tawn was there on a number of occasions to help me. At this point I'd also like to thank former STOR-i administrator Deborah Stewart for all of her time and help and I wish her all the best. Dan Suen and I can always start a business as white van men (men with ven) if our respective careers don't work out. Dan, Erin and Jamie, thanks for all the laughs and board game evenings! Thanks to Faye Williamson for coming in to the office with a smile and hug. Sarah Taylor's been an incredibly supportive friend and has advised me on numerous occasions on how to really live the PhD life and make it your own and has also shared her ballroom talents with me long enough to win me a medal. Couldn't have done it alone (they don't let you!)

I've been blessed with so many friends that it'd be unfair to single certain individuals out. Ben Winterton, Liam Fielder, Sean Cassidy, and Laura Dean make the cut for starting Lancaster's Comedy Institute with me, which went on to become a burgeoning family of friends, many of whom I'll be in touch with for many years to come. The cast and crew of League of Breadgends, with mad props to Mateusz 'Yourchinski' Yourchinski for running it. I've moved to Malta and back again and have held two separate full time jobs in the last 15 months of this PhD and through it all, Bethany Jones has been there to support me through what's been a very tough time. I only hope that I can return the favour because it has meant the world to me. Slow Hamp!

I also want to thank the good people at Exient Malta and Sega Hardlight studios for being really cool about the whole writing up process and being flexible with me. Also thanks for letting me make video games and for giving me money. That's been awesome of you guys. Mad props.

I also want to thank my parents for checking in on me and being supportive through everything I do, regardless of what it is. Their unexhaustible pride in me is wonderful.

Declaration

I declare that this these is my own work except where noted otherwise and has not been submitted for the award of a higher degree elsewhere.

Jak Marshall

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Ranging from terrorist threats and security affairs in countries of interest to the health status of certain political leaders, gathering and analysing intelligence plays a key role in shaping the course of action in situations of national interest. While enormous resources are spent with the goal of providing timely and accurate intelligence in response to requests for information, the vast volume and type of information emanating from intelligence sources often makes it difficult to achieve these goals.

The intelligence process consists of three stages: Collection, processing, and analysis. In response to a list of information requests, items are collected through a variety of means: human, signals, imagery, and open source. The processing phase is designated to transform the raw information that was collected into products that may be used in the analysis phase. The nature of the collected raw material dictates the type of operations to be included in the processing effort, as well as the required analysis capabilities. Common processing operations include data reduction, noise reduction, decryption, language translations, context clarification and more. In the analysis phase, the processed intelligence is evaluated and put in perspective with respect to current assessments.

The key measures of effectiveness of an intelligence operation are timeliness and accuracy. Timeliness is important because the information requests generally have an explicit deadline or the situation on the ground may develop rapidly. Accuracy

is manifested by the correctness and precision of the information provided to the decision makers. At a lower resolution, the value of single information items also depends on the impact they have on the overall intelligence analysis.

This thesis is concerned with the collection phase of intelligence operations. Since the impact of each source of intelligence is by and large not known in advance, we take a Bayesian learning approach that takes into account the past performance of each intelligence source to evaluate which source is the most promising. Every situation requires an exploration of the available options, which must necessarily converge on the most appropriate information providers.

This short introductory material serves to motivate the rest of the thesis. We begin by talking about the nature of intelligence operations research and the types of operational problems that are considered. We then outline the structure and contributions of this work.

1.1 Intelligence operations research

Incorporating operations research methods into a problem of military interest is nothing new. In fact the origins of operations research (OR), previously known as operations analysis (OA), as a formal field of study during World War II are well known. A review of some of the techniques that arose during that particular conflict are given in [McCloskey, 1987a] and [McCloskey, 1987b].

As we move towards the modern era and the changing nature of warfare that this brings, the increase in international terrorism and attacks by small groups and individuals as opposed to nations and large scale armies [Cronin, 2002] has increasingly placed importance on intelligence gathering operations to tackle these less visible threats. As such, there has been increased interest in applying OR type thinking to applied intelligence problems.

In the 2010 Philip McCord Morse lecture, archived in [Kaplan, 2012], a review of the interface between operations research and intelligence operations was given using the term 'intelligence operations research' for this particular branch of OR.

The 'terror queues' problem [Kaplan, 2010] models undetected terror plots as customers in a collection of queues, where the service corresponds to intervention measures to foil these plots. In queueing terms, an abandonment is an event where a customer leaves a queue before they are served. In this context of these queues, an abandonment represents a failure to detect an aggressor and therefore a successful terror attack. The problem is to dynamically allocate the attention of servers to minimise the number of abandonments and maximise service completions to prevent terrorists from carrying out their activities.

The servers could be human agents infiltrating terror cells, in which case switching costs would be high to reflect that agents are committed for long term operations. The different queues could also represent geographical locations at any scale, which could affect the ease of re-allocation of committed resources. It is also possible to capture the ongoing learning process concerning the various queues given the efforts of past investigations or a history of previous attacks.

A Bayesian learning methodology guides allocation in a dynamic fashion. Adversarial game theoritcal elements can be introduced where the terrorists wishing to evade detection have partial or complete knowledge of the allocation policy that is in use by the counter-terror organisation.

As well as protecting infrastructure and 'soft-targets' such as transport hubs, it may also be in the interests of government to delay or disrupt the proliferation of technologies and projects such as nuclear weapons programmes by other nations. The model of [Brown et al., 2009] assumes that in order for an enemy nation to achieve a major goal such as developing a nuclear weapon, various subgoals need to be achieved, and a dependency structure will exist for this group of tasks. Being able to model this structure in a reasonable way and having sufficient knowledge of the progress that has been made so far in achieving each of the tasks is key. However, the intelligence on these topics is usually a mixture of actual and imprecise information so once more, the proliferating nation will be acting in an adversarial manner in order to detect and mitigate the effects of the disrupting

Figure 1.1: Source: https://www.cia.gov/library/publications/additional-publications/the-work-of-a-nation/images/intel%20cycle-2.jpg.

activity, including the gathering of intelligence. The idea that certain pieces of intelligence or sources of intelligence are more reliable and precise than others is one that I would wish to capture in the models developed in this work.

On the subject of intelligence gathering itself, Kaplan describes the overall intelligence operations process as the 'intelligence cycle', which consists of planning. collection, processing, analysis, and dissemination processes in continuous feedback as shown in Figure 1.1.

Kaplan frames the discussion of intelligence operations research efforts in these terms and many other papers explicitly mention this concept when they are defining what they understand by the real world intelligence gathering process. In a section of the lecture regarding ideas for future work in intelligence operations research, Kaplan highlights the tension between the rates at which data can be collected, processed and then analysed, making particular mention of signals and satellite imagery based intelligence. It is possible for agencies to collect a very large amount of data which is far in excess of the same agency's ability to provide enough technical processing effort to actually use all of it. Further, the processing rate is likely to be faster than the rate at which the processed intelligence can be contextually analysed and prepared for dissemination. This motivates the need to be selective about which pieces of raw intelligence one should commit further pro-

cessing effort to such as language translation, for example. It is for this particular open problem that I wish to develop models in my PhD project.

Contributions towards this problem have since been made on the subject of selective intelligence gathering and analysis by students of the Naval Postgraduate School of Moneterey California. Selective analysis of large communication networks is considered in [Nevo, 2011] and [Ellis, 2013] where the goal is to focus surveillance efforts on node pairs where one or both of the nodes are considered to have high relevance to a particular intelligence objective or set of keywords. The search for important information becomes an exploration-exploitation problem over the graph of individual people within an organisation. This problem setting does not necessarily require the graph theoretic setting and could be adapted to the general problem of sampling information from many distinct sources.

The work of [Zlatsin, 2013] considers the application of detecting the location of drug smugglers at any given time and considers the problem of consolidating many different types of intelligence together (such as human and signals intelligence) that can have varying degress of plausibility considered separately and collectively. It is worth considering that not all intelligence types are of equal reliability or importance and defining the worth of intelligence items will be a necessary part of the modelling process. The NPS theses referenced here all include a Bayesian learning framework that guides future decisions based on the current state of knowledge at any time. In the case of Zlatsin, a particular quantity of interest, e.g. the location of smugglers' boats is explicitly sought after as opposed to the more abstract concept of 'relevance' in general. In this work, we took the approach to keep the real world intelligence operatives in mind in the models that have been developed, but to keep said models as general as possible. Adequate extensions to the new theory should be reasonably easy to add to the base model should a particular application call for additional or modified assumptions and features.

1.2 Outline of thesis

This section sets out the structure of the thesis and highlights the nature and placements of the major contributions of this document.

Chapter 2 serves as the literature review for this document. In addition to covering a range of topics that are of relevance to the contributions made in this thesis.

In Chapter 3 we develop and discuss a single source intelligence operations exemplar problem. The purpose of this material is to illustrate how the methodologies explored in Chapter 2 can apply to a simple model of intelligence operations.

It is the introduction and development of the multi-armed bandit allocation model (MABA) which forms the major contribution of this thesis. It is a model of the relationship between the processing and analysis parts of the intelligence cycle during a time critical investigation. It allows for an abstraction of intelligence operations that incorporates the time pressures of a true investigation in a way which has not previously been done. Chapter 4 outlines the MABA problem setting, which will be our chosen framework for modeling intelligence operations in the numerical work carried out in Sections 5 and 6.

Chapter 5 approaches the MABA problem using a Dirchlet-multinomial model and compares the performance of various combinations of source selection and allocation policies, including those based on knowledge gradient and Thompson sampling methods. A full account of the numerical work carried out is given. Moreover, we develop a Lagrangian relaxation of the model and implement that numerically in some select cases.

In Chapter 6 we generalise the solution approach to a continuous setting and also incorporate the key operational consideration of processor *judgement uncertainty* into the model. An account of preliminary numerical work is provided. We also set up the theoretical model for a Lagrangian relaxation of the model under this setting which is analogous to that set out in Chapter 5.

The numerical work carried out in Chapters 5 and 6 demonstrates the viability

of the MABA model to be used in practice and also provides insights into which heuristic solutions have the most consistent performance so that practitioners can select which method to use in their own time critical investigations. Chapter 7 contains concluding remarks regarding the project as a whole and makes recommendations for future work.

Chapter 2

Literature Review

This chapter forms the literature review of this thesis and finishes the process that the introductory material started which is to cover a breadth of material that frames the specific operations research areas to which this document contributes. Here we also cover the various methods and models from which the later material inherits much of its ideas and intuition.

Chapter 1 contained references to relevant literature in the burgeoning field of intelligence operations research. We begin this chapter with a recent contribution towards developing models of intelligence operations, which we take as a starting point as we consider our own models.

In Section 2.1 we examine the tandem queue model of the processor-analyst relationship in the intelligence cycle. The model, developed by Yinon Costica (2010), is a relatively recent contribution most closely reflects that relationship that we will eventually come to model using the MABA framework. In particular, the tandem queue model expresses the demand of expediency from a processor who is burdened with a large amount of intelligence materials to filter through for the most crucial information.

2.1 Costica's tandem queue model

In this subsection we take a look at a recent existing model of the intelligence processing problem using a queue theoretical approach. Costica's work introduces us to two useful characters, who are the processor and the analyst.

The processor is a technician who performs tasks such as language translation and image refinement to prepare incoming intelligence items so that they are readable by our second character, the analyst. The analyst is assumed to be perfectly capable of placing items that she receives within the context of the broader intelligence objectives of her organisation using her knowledge of other information that she has seen. We use Costica's work as the starting point for modelling intelligence operations.

The model itself is presented in [Costica, 2010]. It is here that we first encounter items of intelligence modelled as customers in a queue. There is no consideration to the size and complexity of the intelligence item itself. All intelligence items are assumed to be processed at the same rate.

Intelligence items are assumed to arrive at a processing station at rate λ_1 forming an M/M/1 queue where items are processed at rate μ_1. The arrival rate λ_1 is formed from the sum of two independent homogeneous Poisson processes with rates λ_P and λ_N which respectively denote arrivals of items that are classed as *positives* (P) and *negatives* (N) where P-type items are of importance and N-type items are not. No middle ground exists, i.e. importance classification is purely binary. The value λ_1 is known whilst the values λ_P and λ_N are unknown.

Costica writes of classification processes that are characterised by their *sensitivity* p and *specificity* q which respectively denote the proportion of the P items that are classified as being positive and the proportion of rejected items that are truly of type N. The processing station then passes along items to the analysis station with an overall rate of $\lambda_2 = p\lambda_P + (1-q)\lambda_N$ at which point the analysts are able to perfectly determine the importance of the items in the M/M/1 queue that forms at their station with rate μ_2. It is assumed that the more time that a

particular item is studied by processing staff, the more accurate their evaluation of its importance will be. The way in which the incoming items progress through the system in shown in Figure 2.1.

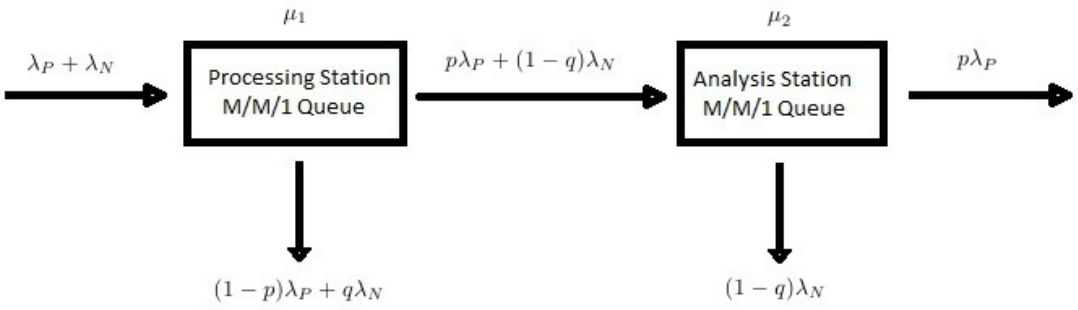

Figure 2.1: The flow of intelligence items in the tandem queue model

The tandem queue is stable if the arrival rate of items at each station is less than the service rate. The long-run waiting times for an item at the processing and analysis stations of the queue are respectively denoted by W_1 and W_2. The rate of processing μ_1 is directly linked to the quality ϵ of the classifications at the processing station which affects the values of both p and q and therefore λ_2.

The task under this model is to choose an optimal processing station service rate μ_1 for the processing staff to maximise an objective function based on p and q as performance measures and the mean waiting time $\bar{W}$ as a constraint, the details of which will follow. Operationally this corresponds to finding the best tradeoff between classification accuracy and timeliness.

The objective in Costica's model involves use of Receiver Operating Characterstic (ROC) curves, which are described in [Parpucea et al., 2011] and [Ahmadi et al., 2010], to characterise the behaviour of any kind of item classification system. ROC curves have an x-axis which is the false positive rate $1-q$ of the classification method and the y-axis is the false positive rate p of the classification method.

The *ROC space* is the feasible region $[0,1] \times [0,1]$ of an ROC curve where the point $(0,1)$ represents a perfect classification system with no chance of false positives or true negatives. The ROC curve in this space is the true positive rate p modelled as a function of the false positive rate $(1-q)$ so we have $p = f(1-q)$.

A particular classification policy is characterised by its ROC curve where the worst case scenario for any policy is for its ROC curve to be the line $p = 1 - q$ which corresponds to performance which is just as effective as random guessing. For any given classification method the sensitivity of the ROC curve to changes in the classification quality of items was of interest to Costica. A parameter ϵ was introduced to model the classification quality where $\epsilon \in [\epsilon_{\min}, \epsilon_{\max}]$, where ϵ_{min} represnts the best classification quality. The general parametric form for the ROC curve is

$$p = f_\epsilon(1 - q, \epsilon). \tag{2.1}$$

If we denote by λ_2 the arrival rate of items to the analysis station, one has that

$$\begin{aligned} \lambda_2 &= p\lambda_P + (1 - q)\lambda_N & (2.2)\\ &= f_\epsilon(1 - q, \epsilon)\lambda_P + (1 - q)\lambda_N, & (2.3) \end{aligned}$$

where we substitute (2.1) for p in (2.2) to obtain (2.3). One can adjust the classification quality ϵ to control the output rate λ_2. It should be noted that as we increase ϵ towards its maximum, mathematically feasible, value $\epsilon_{\max}$ the processors are effectively spending less time with each item and worsening the quality of classification towards the worst case scenario possible for that particular method, which may or may not be at the level of random guessing. The opposite applies as we decrease ϵ towards $\epsilon_{\min}$.

To ensure that the parameter ϵ has this property one assumes that it is monotone increasing in the rate of service μ_1 at the processing station as it is assumed a lesser mean time spent processing each item corresponds to an overall decrease in classification quality. For a given processing rate μ_1 the corresponding quality parameter ϵ is a function of the rate, which we express as $\epsilon(\mu_1)$.

For a given classification policy there is a level of classification quality known as the *complete test* which corresponds to the best possible classification quality which

can arise from implementing that policy. This realistic ceiling on classification quality has the value $\underline{\epsilon}$ where $\epsilon_{\min} \leq \underline{\epsilon} \leq \epsilon_{\max}$. For this complete test there must also be a fastest rate, $\underline{\mu_1}$, at which $\underline{\epsilon}$ can be achieved such that $\epsilon(\underline{\mu_1}) = \underline{\epsilon}$.

Working in the other direction we assume that there is a maximum rate of processing $\bar{\mu_1}$ that can be achieved which will result in the worst possible classification quality ϵ_{max}. Unlike ϵ_{min} it is always possible to reach this theoretical bound. We let $\bar{\mu_1} \geq \underline{\mu_1}$ be the least value of μ_1 we can choose such that $\epsilon(\bar{\mu_1}) = \epsilon_{max}$.

We now have a one to one function between the feasible classification qualities and their corresponding service rates so by substitution we can adapt (2.3) to obtain

$$\lambda_2 = f(1-q, \mu_1)\lambda_P + (1-q)\lambda_N \tag{2.4}$$

where $\mu_1 \in [\underline{\mu_1}, \bar{\mu_1}]$

Although the processors do not know whether each individual item is of importance or not, it is assumed that the relative rates of arrival of the two kinds of item are known so one can write the output rate λ_2 from the processing station in terms of values that we have some level of control over. In (2.4) one need only specify the required average processing service rate μ_1 and the false positive rate $1-q$ of the classification policy which is decided by the choice of classification method.

Maximising specificity p over the stable region of choices for μ_1 is not the complete picture. There is also a desire for a certain level of timeliness so a limit is set on the mean overall delay $\bar{W}$ for an item from entering the system to reaching the analysts, assuming the item reaches that station. Assuming stability in the queue, the mean delay time for an individual item is

$$\begin{aligned} W &= \frac{1}{\mu_1 - \lambda_1} + \frac{1}{\mu_2 - \lambda_2} \\ &= \frac{1}{\mu_1 - \lambda_1} + \frac{1}{\mu_2 - f(1-q, \mu_1)\lambda_P - (1-q)\lambda_N}. \end{aligned} \tag{2.5}$$

So as long as we impose some upper limit, $\bar{W}$ on the overall delay for a fully

worked intelligence item we can achieve the optimal value of μ_1 under whichever delay constraint is chosen by the client. The optimisation problem is set up as follows:

$$\max_{q,\mu_1} f(1-q,\mu_1)$$

Subject to:

$$\begin{aligned}
&\mu_1 \geq \lambda_1 \\
&\mu_2 \geq f(1-q,\mu_1)\lambda_P + (1-q)\lambda_N \\
&q \in [0,1] \\
&\mu_1 \in [\underline{\mu_1}, \bar{\mu_1}] \\
&\frac{1}{\mu_1 - \lambda_1} + \frac{1}{\mu_2 - f(1-q,\mu_1)\lambda_P - (1-q)\lambda_N} \leq \bar{W}
\end{aligned} \tag{2.6}$$

where the constraints ensure the stability of the system, the feasibility of the classification method and the timeliness of the entire process. This model returns a recommendation for a processing service rate which is best suited for the situation it is given. I refer the reader to [Costica, 2010] for the full sensitivity analysis of this optimisation model.

This model provides some useful ideas on how to think about modelling intelligence items and also the relationship between the processor, who is burdened with a heavy workload and imperfect classification ability, with the 'all knowing analyst' who receives the filtered selection from the processor.

Key ideas from this model that are used in the core contributions of this thesis are the processor/analyst dynamic and the modelling of intelligence items as individual entities that attribute identical processing times.

As shall be seen in Chapter 3, when we develop a single source model of our own, we choose to develop the processor's character so that she can refuse to process items according to some bespoke policy rather than having to process all items that she encounters.

For now, we explore further topics considered central to the intelligence processing problem and the work carried out in later chapters in this document.

2.2 Markov Decision Processes

A feature of the Costica and Ellis models of the previous section is that they operate within a framework based on continuous time. However it may be more appropriate in certain models to consider discrete time formulations. The framework supplied by Markov decision processes (MDPs) allow for such formulations. MDPs have been known in some form or another since the works of Bellman (see [Bellman, 1957a] and [Bellman, 1957b]) and model discrete time stochastic processes with control. The control aspect of this description refers to the inclusion of a problem solver that interacts with the process at decision epochs that occur at discete time intervals.

At each of the decision epochs in an MDP, the process is in a state i and the problem solver must choose an action a from the set of actions that are available whilst in state i to advance the process to the next decision epoch. When the action a is chosen, the state then randomly advances to a new state $i^{'}$ which also depends on the choice of a according to the state transition function $P_a(i, i^{'})$ of the process. The Markovian property of MDPs refers to the fact that the process is conditonally independent of all preceding states and actions so only knowledge of the current state i and current action a are necessary. The decision aspect of MDPs are made meaningful with the inclusion of rewards as motivation for the problem solver. The reward $R_a(i, i')$ is earned by the problem solver taking action a whilst in state i and subsequently transitioning to state s'. It is the plurality or actions and their associated rewards which give the problem solver's decisions meaning. The special case MDP where there is only one action for each state, and all associated rewards are the same is the standard *Markov chain* described in [Meyn and Tweedie, 1993].

The problem solver makes use of a *policy* to make decisions based on the

situation in which she finds herself in any given decision epoch. More formally, a policy π is a function that maps states to actions. Since we're working with Markovian provlems, the policies we are concerned only require the current state as an input. We have $a_t = \pi(i_t)$ for all t. An optimal policy $\pi^{'}$ maximises rewards over any instance of the problem horizon and the problem solver would prefer to seek out such policies and deploy them.

Application of MDPs are numerous and diverse. A review of their applications has been published by [White, 1933]. The applications areas cover traditional operations research applications [Gluss, 1959], finance and investment [Rosenfield et al., 1983] and queueing theory [Low, 1974]. There are many more obscure areas covered such as pub darts strategy [Kohler, 1982] and patient admissions to nursing homes [Lopez-Toledo, 1976].

The scope of topics that MDPs cover is demonstrably vast so we now focus on particular sub-topics which lie closer to the the types of problems we wish to solve, and we begin this narrowing of focus by considering stochastic dynamic programming.

2.3 Stochastic Dynamic Programming

Stochastic dynamic programs (SDPs) are a class of optimisation problems that crucially involve uncertainty. As well as the uncertain state transitions that were described in relation to MDPs in the previous section, SDPs are also concerned with uncertainty relating to the problem state i itself. Key parameters relating to the decision at hand (e.g. the probability that a customer defaults on a loan) are unknown and the problem solver must forge ahead with her decision making with imperfect knowledge of the problem state.

A classic stochastic dynamic programming problem is the example of selling a house on the market subject to a fixed deadline, with the objective of maximising the final selling price. The decision epochs correspond to the times at which offers are received, at which time the seller has a one-time opportunity to accept the

offer. If the offer is accepted then the problem ends and the homeowner takes the sum of money offered, otherwise the offer is lost.

It is reasonable to reframe the house selling problem in terms of a processor receiving offers to process intelligence items. Moreover, these offers correspond to individual decisions as to how to react to incoming intelligence items.

In such multi-stage problems an *a priori* static policy may be appropriate depending on the parameters of the problem. It is worth putting thought into whether a dynamic approach will make a significant improvement over the best static policy available, given the additional complexity that the dynamic structure imposes.

With a dynamic policy the decision maker can benefit from learning behaviours as new information arises through the problem horizon. In [Kall and Wallace, 1994] there is a simple example problem of a land developer who needs to decide whether to develop land at a cost and then decide whether to build on the land before or after knowing what the value of doing so would be. There is also the condition that it would cost more to build if you wait to find out the value of building there. The construction of the example shows us that only the dynamic policy provides the best expected gain and outperforms any static strategy.

Dynamic formulations of sufficiently complex problems incur the famous *Curse of Dimensionality* described in [Bellman, 1957b] whereby solving most dynamic programs in their fullest sense is often rendered intractables because of the complications of working with high dimension state spaces. Approximate and heuristic solutions to the full DP using the full formulation as a starting point can often result in significant gains over the best available static policy by making an adequate trade-off between capturing the complexity of a problem and reducing the overall demand for computational resources.

2.3.1 The Bellman Equation

The basic concept that underpins most of stochastic dynamic programming is the Bellman equation. It is used to solve decision problems with discrete time decision epochs which count down from a specified initial start time t to the terminal time 0, where no actions can be taken. A value function V for a general problem setting recursively defines the optimal reward as a function of the current information state $\mathbf{i}$ and also constructively prescribes the actions that should be taken to achieve this reward. The action space A contains all available actions and depends on the current information state.

The notation used in [Ross, 1983] is used to set up the general form of the Bellman equation. The value of state $\mathbf{i}$ with t time periods remaining is $V_t(\mathbf{i})$. The immediate expected reward earned by taking action $a \in A$ of available actions, when in state $\mathbf{i}$ is denoted as $R(\mathbf{i}, a)$. When $t = 1$ there is only one decision left to be made so we can only earn immediate rewards so the following relation holds

$$V_1(\mathbf{i}) = \max_{a \in A} R(\mathbf{i}, a), \tag{2.7}$$

where for completeness one has that $V_0(\mathbf{i}) = 0$.

For each state i and action a there is a probability transition matrix $\mathbf{P_i}(a)$ governing which state we begin the next time phase in given that the action a is taken next. When more than one time period remains we must choose the action that maximises the expected combination of immediate and future rewards. In general we have for $t > 1$

$$V_t(\mathbf{i}) = \max_{a \in A} \left[R(\mathbf{i}, a) + \sum_{\mathbf{j}} \mathbf{P_{ij}}(a) V_{t-1}(\mathbf{j}) \right] \tag{2.8}$$

where $P_{\mathbf{ij}}(a)$ is the probability of moving from state $\mathbf{i}$ to state $\mathbf{j}$ if action a is chosen next. The value function V needs to be evaluated recursively using the $t = 1$ cases such as in (2.7) first and then working in reverse from there. This can be difficult even when the state space is only moderately large so it is often the case that

the value function needs to be approximated in some other way via some heuristic method.

In stochastic dynamic programming terminology, a *policy*, denoted by π, is a rule that recommends an action to take if the problem is in state i at time t. Such policies have a sequence of states X_n that arise which have the Markov property that the transition probabilities are the $\mathbf{P_{ij}}$ as before. When working with policies we can interpret the value functions as the expected reward V_π earned when implementing the policy π.

$$V_\pi(\mathbf{i}) = E_\pi\left[\sum_{k=0}^{n} R(X_k, a_k)|X_0 = \mathbf{i}\right] \tag{2.9}$$

where a_k denotes the action taken at the k^{th} decision phase. This approach allows a manager to specify an exhaustive set of instructions for a process. Thinking of solutions to stochastic dynamic programming problems as policies is a standard way of evaluating competing policies and comparing the performances of competing solutions and the contributions made by this thesis will be expressed in these terms. The applied intelligence problem of information collection is such that the role of a stochastic dynamic programming approach is clear since one wishes to learn in a sequential fashion under uncertainty.

Discounting of future rewards as in [Veinott, 1979] is often incorporated into model formulations, where a power of a *discounting factor* $0 < \gamma < 1$ is applied to rewards for each time epoch after the current one. Such factors are motivated by the economical assumption that a reward earned sooner is more valuable than the same reward earned later and the smaller the value of γ is, the more strongly we place value on more immediate gains. Since timeliness of intelligence delivery is one of the challenges faced by the intelligence community this could be one way of incorporating this, particularly if the time horizon is modelled as infinitely long. We obtain the discounted value function, also known as *the Bellman Equation* by including this discounting factor for $t > 1$

$$V_t(\mathbf{i}) = \max_{a \in A} \left[R(\mathbf{i}, a) + \gamma \sum_{\mathbf{j}} P_{\mathbf{ij}}(a) V_{t-1}(\mathbf{j}) \right] \tag{2.10}$$

and when we consider a policy π

$$V_\pi(\mathbf{i}) = E_\pi \left[\sum_{k=0}^{n} \gamma^k R(X_k, a_k) | X_0 = \mathbf{i} \right] \tag{2.11}$$

where we find the value, for a fixed γ, of being in state $\mathbf{i}$ from this perspective is by maximising (2.11) over all policies π. These discounting factors also give us the option of creating greater sense of urgency by discounting future rewards more heavily. However one must be wary of discounting future rewards so harshly that one acts too hastily by passing along inferior information to the analysts from the processing station for the sake of timeliness in such a way that critically inhibits exploration.

We now look at a specific species of MDP, namely multi-armed bandits.

2.3.2 Multi-armed bandit problems and index policies

In this thesis, we develop models of intelligence operations where the processor has access to many competing sources of intelligence items. From these, exactly one source is to be sampled during each decision epoch. All of this occurs against a backdrop of uncertainty as to the quality of the items that are drawn from each source. Multi-armed bandit theory offers a suitable framework for modelling this kind of scenario.

In the language of bandit theory, the competing alternatives of a decision problem are referred to as *projects* and the sequential allocation of effort between these projects is a process which is to be optimised over several decision epochs. Using what are known as *index theorems* these sequential allocation problems can be solved with computational efficiency and with near optimal performance. Such index policies are the focus of this subsection.

I shall use the language of [Ross, 1983] to introduce the concept of a bandit

from the perspective of stochastic dynamic programming. For a single project we can decide at each discrete decision epoch to allocate effort to that project for the coming time period or retire from the project permanently. Given that we are currently in some state i the immediate reward for operating the project for the next time period is $R(i)$ and the state also changes to k with state transition probability $\mathbf{P}_{ik}$. Retiring at any time earns us a fixed retirement reward M.

We apply a discounting factor, $0 < \gamma < 1$, to the problem and let $V(i : M)$ be the maximum expected discounted reward earned given that we are currently in state i and the retirement reward for the problem is M. The value function V therefore satisfies

$$V(i : M) = \max \left\{ M, R(i) + \gamma \sum_{k} \mathbf{P}_{ik} V(k : M) \right\}. \tag{2.12}$$

For a fixed state i, it is easy to prove by induction that $V(i : M) - M$ is decreasing in M. Since M is constant there must exist a value $\bar{M}(i)$ such that we are indifferent between operating the project and retiring from it in the next time period. Such a value is defined as

$$\bar{M}(i) = \min[M : V(i : M) = M]. \tag{2.13}$$

We now have a critical value policy which instructs us how to act at each decision epoch. We simply need to evaluate $\bar{M}(i)$ at every state and then continue to operate if $M < \bar{M}(i)$ and retire if $\bar{M}(i) < M$ where we can decide arbitrarily for the case where the two quantities are equal.

We would want to be able to optimise the same kind of decision problem for multiple projects of the type discussed so far. So we assume that we have L identical projects, noting that it can be shown that this can be done without loss of generality. We also assume that we can retire from the problem permanently at any decision epoch and earn the reward M for doing so and zero rewards can be earned thereafter. If we denote the state of the problem at any time as $\mathbf{i} = (i_1, \ldots, i_L)$,

where i_j denotes the state of the j^{th} project we can write the maximum expected γ-discounted return if we choose to operate on project j next as

$$O^j(V(\mathbf{i}:M)) = R(i_j) + \gamma \sum_k \mathbf{P}_{i_j k} V(i_1, \ldots, i_{j-1}, k, i_{j+1}, \ldots, i_L : M) \tag{2.14}$$

where $R(i_j)$ is the one-step reward for operating the project j when it is in state i_j. Then we can write the value function V in state $\mathbf{i}$ as

$$V(\mathbf{i}:M) = \max[M, \max_j O^j(V(\mathbf{i}:M))] \tag{2.15}$$

so (2.14) provides us with a theoretically sound decision policy for the action we should take next. However computing each of the $O^j(V(\mathbf{i}:M))$ rapidly becomes very difficult because the number of states can easily get out of control now that we are working in L dimensions, so a direct numerical solution may not be reasonable to attain using this route. However it can be shown that the multi project version of the problem can be decomposed into its constituent projects and we obtain a policy in terms of the indifference values, $\bar{M}(i_j)$ for each of the projects when considered as single project problems.

We find that we should retire from the problem if and only if we have that $\bar{M}(i_j) < M$ for $j = 1, \ldots, n$ and otherwise we should choose the project with the greatest indifference value as long as that value is greater than M also. The proof of why this decomposition is known, at least in its original form, as *Gittin's index theorem* where a proof is provided in [Gittins, 1979] and Chapter 2 of [Gittins et al., 2011], and the reader can also find an alternative proof in [Ross, 1983]. The single project case is also covered by this theorem.

The class of bandit problem Gittins set out in [Gittins, 1979] and [Gittins and Jones, 1979] is referred to as the *standard multi-armed bandit problem* and it was this set-up that Gittins used to prove the optimality of the index that bears his name. Although Gittins indices can be used whether we are working with Bayes

type thinking or not, it is sometimes more useful to approach certain problems with a Bayesian outlook if uncertainty plays a prominent role in the real world problem.

The way to picture a standard multi-armed bandit problem is to think of a one-armed bandit machine with L distinct levers, or arms. The l^{th} arm, when pulled will deliver a unit reward with probability p_i so each of the arm pulls is essentially a single draw from a Bernoulli distribution with a parameter particular to the arm pulled. The complication lies in the fact that we do not necessarily have complete information about the success probabilities p_i, but even with little or no prior information we still have motivation to maximise our expected discounted rewards in this situation. We also assume that these probabilities remain constant between decisions.

In the standard version of the problem, the true probability p_i of a non-zero reward being earned from arm i is constant over the decision horizon. A naive approach to reward maximisation would be to always pull the arm that we believe to have the highest probability of success and continue to do so forever. However this approach ignores the possibility that our lack of information about other arms could lead to the discovery of a more reliable reward stream should we pull alternative arms and learn that one of the other arms has a higher true probability of delivering non-zero rewards.

The key to dealing with this is to view the knowledge we have about an arm as its current state and take a Bayesian approach to learning about each of the arms. By conjugacy, since we're concerned with Bernoulli rewards we assign Beta priors to each of the arms and update these beliefs as pulls are made. If we consider the l^{th} arm of a bandit after t pulls then we know that the success probability p_l has the following pdf

$$(\alpha_l^t + \beta_l^t + 1)!(\alpha_l^t!\beta_l^t!)^{-1} p_l^{\alpha_l^t}(1 - p_l)^{\beta_l^t}$$

where if there have been r successes, α_l^t is equivalent to $\alpha_l^0 + r$ and β_l^t is updated to

be $\beta_l^0 + t - r$. Before discounting, the expected reward from the next pull of an arm that has been pulled t times already is the expected value of the beta distribution that governs it which is equal to $(\alpha_l^t + 1)/(\alpha_l^t + \beta_l^t + 2)$.

We can generalise the standard multi-armed problem by allowing there to be rewards that aren't necessarily unital and follow all manner of known distributions. Indeed, Gittins produced indices that are optimal for bandits with Gaussian rewards.

To make the problem technically explicit for the multi-armed bandit with L arms and exponentially distributed rewards, we say that each arm has an associated belief about the value of the λ_l parameter of the exponential distribution governing the rewards obtainable from the arm with the information available. We have that our initial estimate $\lambda_l \sim$ Gamma (α_l^0, β_l^0) where $\alpha_l^0 > 1$ and $\beta_l^0 > 0$ for all l. We sequentially pull arms and update our posterior beliefs after the $(t+1)^{st}$ pull made such that $\lambda_l \sim$ Gamma $(\alpha_l^{t+1}, \beta_l^{t+1})$.

If we denote by $l^t \in 1, ..., L$ the choice of the $(t+1)^{st}$ arm to be pulled and the reward obtained from that pull as Y_l^{t+1}, we update the shape parameter in the following way:

$$\alpha_l^{t+1} = \begin{cases} \alpha_l^t + 1 & \text{if } l^t = l \\ \alpha_l^t & \text{otherwise.} \end{cases} \tag{2.16}$$

We also update the scale parameter as follows:

$$\beta_l^{t+1} = \begin{cases} \beta_l^t + Y_l^{t+1} & \text{if } l^t = l \\ \beta_l^t & \text{otherwise.} \end{cases} \tag{2.17}$$

So at after t arm pulls we have the vectors $\alpha^t = (\alpha_1^t, \ldots, \alpha_L^t)$ and $\beta^t = (\beta_1^t, \ldots, \beta_L^t)$ which together form the *knowledge state* $s^t = (\alpha^t, \beta^t)$ which gives at time t the complete summary of the beliefs we have for the collection of L bandits.

A policy τ for sequentially choosing arms under this set-up consists of a sequence, $(X^{\tau,t})_{t=0}^{\infty}$ of decision rules which maps knowledge states, s^t to one of the L alternative arms. The challenge here is to maximise expected rewards. The expected reward at time t is $(\lambda_{X^{\tau,t}}(s^t))^{-1}$. For the T stage bandit setup, the problem is to find τ^{max} where

$$\tau^{max} = \max_{\tau} \mathbf{E}^{\tau} \sum_{t=0}^{T} \frac{\gamma^t}{\lambda_{X^{\tau,t}}(s^t)} \tag{2.18}$$

for the discount factor γ. The notation $\mathbf{E}^{\tau}$ denotes the expected rewards receieved where we always use decision rules of the policy τ. Just as we needed Gittins indices to escape the high dimensionality problems of the bandit problems discussed in the previous section, we should also have a computationally cheap heuristic or approximation for these multi-armed bandit problems.

We briefly discuss complexity of bandits problems in general. In the introductory section of [Gittins et al., 2011] the authors motivate their discussion of bandit processes with seven example problems. One of these is a sequential decision problem concerning allocation of several competing medical treatments to patients. The authors go on to show that the SDP cannot solve this problem in polynomial time.

Using an index based policy however, the number of individual computations and the amount of storage space can be respectively reduced to being quadratic and linear in the problem size which is a great saving in both quantities. In the particular case given in the text, the problem can still be solved optimally. The intelligence operation problems that we formulate in this thesis are complex enough to require some kind of heuristic approach to render them solvable in a reasonable amount of time.

That is not to say that bandit problems can't be solved efficiently in some cases. It was shown in [Gittins and Jones, 1974] that a standard MAB can be solved efficiently using index policies and is at most NP-hard depending on the formulation. There is also the concept of a restless bandit, where the states of in-

dividual arms do not necessarily remain unchanged if they are not pulled. Finding optimal policies for such problems was proven to be PSPACE hard in [Bertsimas et al., 1995] and therefore it is more likely that heuristic solutions are required to find optimal solutions.

However it is not always best practice to jump directly to approximation. If your SDP problem is intractable, it is worth investigating whether a Lagrangian relaxation can provide a route to a direct solution first, and then exploit the properties of the relaxed, probably easier proxy problem, if this can't be done.

2.4 Lagrangian Multipliers

Formal optimisation problems tend to consist of an objective function, which is to be minimised or maximised, and a set of constraints, which are not to be violated. Problems such as these can be made easier to solve by using the technique of Lagrangian multipliers, originally developed in [Lagrange, 1811]. The basic premise of the multiplier method is to incorporate one or more of a given problems constraints into its objective function. In doing so, the problem can be often be solved in such a way that directly accommodates the constraints directly, rather than having to verify solutions after the fact.

We proceed by giving an example. We want to minimise

$$\min f(x, y) = x^2 - 8x + y^2 - 12y + 48$$

subject to

$$x + y = 4. \tag{2.19}$$

We can rewrite this problem, taking the constraint into the objective by means of a Lagrangian multiplier.

$$\min F(x, y, \lambda) = x^2 - 8x + y^2 - 12y + 48 - \lambda(x + y - 4) \tag{2.20}$$

and we now take partial derivatives and set them to zero

$$F_x(x, y, \lambda) = 2x - 8 - \lambda = 0$$
$$F_y(x, y, \lambda) = 2y - 12 - \lambda = 0$$
$$F_\lambda(x, y, \lambda) = x + y - 4 = 0$$

which yields

$$x = 1$$
$$y = 3$$
$$\lambda = -6$$

to yield a minimum of 10 when substituting back into 2.19. Observe how we were able to find a valid candidate solution for the original problem directly with no need to explicitly solve the conditions and use them to eliminate extra variables. Even if a solution does not arise directly out of using a Lagragian multiplier to incorporate constraints into the objective of a problem, the act of manipulating the constraints in this way can still make the problem easier to solve by other means. This is commonly referred to as *Lagrangian Relaxation*.

The use of Lagrangian multipliers is commonplace in the world of optimisation and remains a powerful technique. Problems concerning traditional defense operations too have made use of Lagrangian relaxation techniques. Whether it concerns the allocation of defense assets ([Pugh, 1964] and [Cheong, 1985]), minimising the costs associated with weapon use ([Kwon et al., 1999]), or optimising the layout of communication networks on challenging terrain ([Ibrahim and Alfa, 2015], Lagrangian relaxation techniques have found their use in OR problems for a long time. It would not be out of the question to apply Lagrangian techniques to the relatively new field of intelligence operations research.

The work of [Everett, 1963] remarks that although a Lagrangian multiplier method can't always guarantee that a solution can be found directly, any solution that is found this way is guaranteed to be a valid solution that satisfies all constraints. The relaxation process doesn't always yield a solution, but it can certainly pave the way toward a solution by some other means. In such a case we must resort to an approximate solution method. In the case of a hard stochastic dynamic programming problem, we can make a Lagrangian relaxation of the problem first and then employ one of many heuristic methods, some of which are discussed in the next section.

2.5 Approximate Dynamic Programming

When an SDP problem is intractable or otherwise prohibitively costly to solve, and a corresponding Lagrangian approach fails to allievate such obstacles, approximate solutions can often help provide servicable solutions.

The point is made in [Powell, 2011] that dynamic programming problems generally suffer from a vulnerability to high dimensionality of the state and action spaces. However real-world operations require good solutions that exact methods can't provide in reasonable lengths of time and approximate dynamic programs (ADPs) occupy the space left behind by exact and otherwise relaxed methods.

The heuristic nature of solutions to approximate dynamic programming problems almost always incurs a degree of suboptimality in the resulting policies. However, choosing the right heuristic for the right problem can often mean that the perplexed problem solver can access a workable solution for their problem, whilst minimising the negative impact on solution quality. In this section we showcase some examples of currently applied approximate solution types in ADP problems.

2.5.1 ϵ-Greedy Methods

Any discussion on heuristic methods would be incomplete without talking about the greedy approach. The so called ϵ-greedy approach in [T, 2010] states that the problem solver in state i should choose the action a that maximises the immediate expected reward $R_a(i, i^{'})$ with probability $1 - \epsilon$ and choose a random action with probability ϵ. When $\epsilon = 0$, we simply call this special case the greedy approach.

The principle of the ϵ-greedy approach is that the best looking action at any given time is probably the best one, so the problem solver should just go ahead and select it. However when we're dealing with a state space that has any degree of uncertainty about its underlying parameters, we run the risk of becoming trapped in a locally optimal solution by not exploring the action space thorougly enough to find the true best actions for each state. What ϵ-greedy does is force the problem solver to randomly try any of the apparently suboptimal actions with probabiity ϵ in an effort to strike a balance between exploitation of the 'best' actions and the exploration of possibly better actions over time.

Locally optimal solutions occur in practice even when there isn't uncertainty if the problem is complicated enough [Hougardy and Kirchner, 2006] and optimisation problems that deploy greedy type solutions need to find workarounds for this phenomenon. Randomly choosing suboptimal actions and repeating the problem from a series of random starting points are but two of the ways to hunt for other local optima that surpass the best known current local optima.

We must also bear in mind that it is not always possible to deploy such tactics in real operations. If we consider the intelligence operations application in the context, acting completely at random or restarting the problem endlessly aren't likely to be viable approaches. Nor can we rely on methods which are only proven to converge on a good solution over an infinite horizon only. If it is generally expensive to explore suboptimal actions it is worth refining our approach to exploration so that we do it economically.

In the next subsection we discuss a 'greedy-like' heuristic that is designed to

explore in a fashion that is more targeted than electing to act at random some of the time.

2.5.2 Knowledge gradient methods

There is an argument put forward in [Ryzhov and Powell, 2010] and [Ryzhov and Powell, 2011] for the use of knowledge gradient (KG) or *one-period look ahead* policies for such problems to be used in certain multi-armed bandit problems. One reason given in [Ryzhov and Powell, 2010] is that Gittins indices, although proven to be optimal for indexable problems, can be difficult to compute. Also, some experimental evidence is provided to show that KG policies can outperform Gittins index approximation based policies when the rewards are exponentially distributed as opposed to say, Gaussian.

The knowledge gradient method places value on information from decisions that will inform us how to obtain higher expected rewards. Suppose that we have already made t arm pulls of a multi-armed bandit and are told that after the next pull there will be no information gain recieved from any future pulls. At time $t+1$ onwards it is clear that we would continue to keep pulling the arm with the highest expected immediate return forever as it would always be our best available choice.

In this situation we would wish to make our $(t+1)^{st}$ pull such that the highest expected immediate return for arms at time $(t+1)$ improves most from the same value at time t. Explicitly we write what is called the KG *factor*, for the l^{th} arm at time t as

$$\nu_l^{KG,t} = \mathbf{E}^t\left[\left(\max_j \frac{\beta_j^{t+1}}{\alpha_j^{t+1}-1}\right) - \left(\max_j \frac{\beta_j^{t}}{\alpha_j^{t}-1}\right)\right] \tag{2.21}$$

where the right hand side of (2.21) is the expected improvement in our estimate of the maximum expected reward from time t to time $t+1$

Now to state the overall KG policy we need to balance the immediate expected gains with the total future gain in information that would be obtained from all subsequent pulls given that we pull the l^{th} arm next. We can obtain the policy for

the infinite horizon case by first considering the finite case and taking limits. If we recall that the KG method assumes that we adopt a greedy approach after the next arm is pulled, for the decision at time t we select the arm, $X^{KG,t}$ according to the following decision rule.

$$X^{KG,t}(s^t) = \arg\max_l \frac{\beta_l^t}{\alpha_l^t - 1} + (T-t)\nu_l^{KG,t}. \tag{2.22}$$

However we also need to take into account the discount factor γ for the remaining rewards to be completely accurate so we rewrite (2.22) for the finite case as follows

$$X^{KG,t}(s^t) = \arg\max_x \frac{\beta_x^t}{\alpha_x^t - 1} + \gamma\frac{1-\alpha^{T-t}}{1-\alpha}\nu_x^{KG,t}, \tag{2.23}$$

and by letting $T \to \infty$ we obtain the infinite horizon decision rule

$$X^{KG,t}(s^t) = \arg\max_l \frac{\beta_l^t}{\alpha_l^t - 1} + \frac{\gamma}{1-\alpha}\nu_l^{KG,t}. \tag{2.24}$$

So in this way the KG policy always chooses the arm that delivers the best combination of achieving immediate reward and attempting to gain new valuable information about the arms. The researchers setting out the case for KG in [Ryzhov and Powell, 2010] and [Ryzhov and Powell, 2011] also include computational experiments in their work to compare the performance of the KG policy against some other policies, including Gittins indices, for the exponential reward setting and their work would suggest that the KG method outperforms or at least fares well against them.

The Gittins index used in the comparison is set up to deal specifically with Gaussian rewards whereas it is exponential rewards that are used for the test studies so there may be some inadequacy in the experiment design, ergo the topic should be revisited. However, the relative computational cheapness of KG and the reasoning behind its construction are both appealing so KG should still be more than appropriate for use in studies carried out in this thesis.

2.5.3 Thompson sampling and Optimistic Bayes sampling

A problem with the standard greedy approach, and other deterministic policies, is that they are prone to becoming trapped in locally optimal decisions patterns. In a bandit problem, if a particular arm has the highest expected overall reward according to a given policy and belief state, but the arm in question isn't truly the optimal arm to choose, then the policy in question is likely to continue its exploitation of that arm as long as it performs reasonably well, neglecting to explore other arms often enough to discover the true best arm. The purpose behind the two heuristics in this subsection is to incorporate exploration into the decision process in a random fashion in such a way as to favour the arms which are believed to be better but not to choose them deterministically.

The first of these is the Thompson sampling (TS) approach, described in [Thompson, 1933]. When faced with a multi-armed bandit problem, TS first instructs us to randomly sample from the posterior reward distributions of each arm and use the results of these samples as the indices for the arms at that point in time. TS then says we should choose the arm with the highest index and redraw new indices at each time point. It's simplicity and effectiveness has seen TS sampling policies being implemented in web design and analytics.

Optimistic Bayesian sampling (OBS) (see [May et al., 2012]) is a variant of the TS approach which places value on the posterior beliefs about arms in the event that an arm with a good posterior reward distribution produces a bad random draw during a TS style index creation process. As with the TS approach, one makes a random sample for each arm based on its posterior reward distribution. However in the OBS approach, this random integer is compared to the expected immediate return for that arm and the greater of these two values acts as the index for that source. The policy is asymmetric in that is apparently shuns bad draws from good arms, and embraces good draws from bad arms and great draws from good arms. Hence, the 'optimistic' part of the name.

Chapter 3

Single source allocation model: Exemplar Problem

This chapter serves to explore how the material from the literature review of Chapter 2 may be applied to a general intelligence gathering problem. A single source allocation model is formulated here and draws on knowledge from the fields of dynamic programming, Bayesian statistics and multi-armed bandit theory.

Where later chapters of this document are concerned with managing multiple intelligence sources which compete for the attention of intelligence processing staff, the discussion here looks at the behaviour of a single stream of intelligence items and related sampling policies.

Suppose that there is an incoming stream of intelligence items and that it is the job of an intelligence processor to evaluate the importance of these items with respect to an operational objective specified by an analyst, who receives intelligence items passed along by the processor. The processor draws items from the pool sequentially and before they can draw another item from the pool the item currently held must be either permanently discarded or passed along for analysis.

The processor can discard the currently held item if it is deemed to be unimportant or the item can passed further along the chain of command to analysts if the item in question is clearly pertinent to the objective at hand. If the value

judgement is not so simple to make for the item, it is also possible to choose to devote more time and effort into scrutinizing it further to better understand its level of importance. Over a finite time horizon the goal of the processor is to submit a collection of items with the greatest combined importance values possible.

Here a model for the described problem is given and a dynamic program is formulated to solve the processor's problem of maximizing the overall value of submitted items.

3.1 Modelling the Items

Items discussed in the context of the studies in this document all have some form of importance score associated with them. In this single source model, the distribution of these scores in the processor's pool of items is assumed to be Normal with some unknown mean μ. If we briefly assume that μ is known and denote by X the importance score of a single item drawn from the pool we assume that

$$X|\mu \sim N(\mu, \sigma^2) \tag{3.1}$$

where the variance σ^2 is known. When it comes to deciding the fate of the next item drawn from the pool, for simplicity we always deal with the posterior distribution for μ that was current just before the currently held item was drawn from the pool. We write this as

$$\mu \sim N(\xi, \nu^2) \tag{3.2}$$

and both parameters are updated simultaneously to include any new information about μ when and only when we discard an item or when we pass an item along for analysis. The processor is assumed to have an imperfect ability to assess the true importance scores of items. The rationale for this is that the processor sees the items in relative isolation in comparison to the analysts who have a more comprehensive view of the intelligence landscape. When the processor records

their 'best educated guess' of an item's importance, we say that this observed value is the random variable Y. We have that

$$Y|X \sim N(X, \tau^2) \tag{3.3}$$

where τ^2 is known and represents the degree of imprecision which the processor tends to exhibit when making judgements about the true importance value of the item that they are inspecting. The variances σ^2 and τ^2 are common across all items.

Since it is possible for the processor to make multiple observations of a single item, we use the random variable $Y_{(n)}$ to denote the sample mean of n observations (assumed independent given their true score X) made of an item, hence we have

$$Y_{(n)}|X \sim N\left(X, \frac{\tau^2}{n}\right). \tag{3.4}$$

Suppose a request for information concerns the progress made towards the development of nuclear weapons in a certain country of interest, here represented by μ. In this context X could represent signal intelligence (often vast in volume and noisy) about the quantity of highly enriched uranium already created, and the Y_i's are observations resulting from the processors doing a preliminary analysis of the signal intelligence.

From the above we can then infer the unconditional X-distribution by writing

$$X = \mu + \epsilon \tag{3.5}$$

where $\mu \sim N(\xi, \nu^2)$ and $\epsilon \sim N(0, \sigma^2)$ are independent r.v.s. We therefore infer that

$$X \sim N(\xi, \sigma^2 + \nu^2). \tag{3.6}$$

Further, we can write

$$Y \sim X + \zeta \tag{3.7}$$

where $X \sim N(\xi, \sigma^2 + \nu^2)$ and $\zeta \sim N(0, \tau^2)$ are independent r.v.s. We then infer that

$$Y \sim N(\xi, \sigma^2 + \nu^2 + \tau^2). \tag{3.8}$$

We now have unconditional distributions for both the value of an item drawn from the processor's pool X and the observed value of the item Y as reported by the processor.

3.2 Dynamic Program

Formally, the finite time horizon for this problem consists of T discrete decision epochs. We assume that the processor begins the problem with T epochs remaining having already drawn an item from the pool and also already having made on observation of that item. We let V be the value function for the problem and give the general form

$$V_t(\xi, \nu^2, y_{(n)}, n) \tag{3.9}$$

where

- $t :=$ the number of decisions remaining in the problem.
- $\xi :=$ the posterior mean for μ which was current before the current item was drawn from the item pool.
- $\nu^2 :=$ the corresponding posterior variance for μ.
- $y_{(n)} :=$ the mean value of the observations made of the current item so far. The subscript indicates that the sample mean is taken for the n observations of the current item
- $n :=$ the number of observations, that have already been made, of the current item.

At any decision epoch there are three actions one could take:

- Discard (D) := Cease looking at the current item and draw a new item from the item pool. Immediately make one observation of this new item.
- Pass (P) := Cease looking at the current item and pass it to the analysts, earning a reward $(X - C)$ for doing so where C is some user specified global constant. Immediately draw a new item from the pool and make one observation of it.
- Re-Assess (R) := Make an additional observation of the current item.

For clarity, all of the above actions reduce the decision counter t to $t-1$. We can condition on these actions to formulate the dynamic program. First we have that

$$V_t(\xi, \nu^2, y_{(n)}, n) = \max[V_t(\xi, \nu^2, y_{(n)}, n|D), V_t(\xi, \nu^2, y_{(n)}, n|P), V_t(\xi, \nu^2, y_{(n)}, n|R)] \tag{3.10}$$

so the formulation is now reduced to evaluating the three conditional value functions in (3.10).

To obtain V given that we take action D we use

$$V_t(\xi, \nu^2, y_{(n)}, n|D) = \int_{-\infty}^{\infty} V_{t-1}(\xi_+, \nu_+^2, y_{(1)}, 1)\phi\left(\frac{y_{(1)} - \xi_+}{(\sigma^2 + \nu_+^2 + \tau^2)^{\frac{1}{2}}}\right) dy_{(1)} \tag{3.11}$$

where the function ϕ refers to standard Normal cdf and the notation y_1 refers to the new observation of the new item. We also use the subscript notation '+' to make it clear that we have updated the posterior mean and variance for μ on the right hand side of (3.11).

We compute the updated posterior mean and variance for μ

$$\xi_+ = \frac{\xi(\frac{\tau^2}{n} + \sigma^2) + \nu^2 y_{(n)}}{\frac{\tau^2}{n} + \sigma^2 + \nu^2} \tag{3.12}$$

$$\nu_+^2 = \frac{\hat{\nu}^2(\frac{\tau^2}{n} + \sigma^2)}{\frac{\tau^2}{n} + \sigma^2 + \nu^2} \tag{3.13}$$

For the passing action (P) the conseqences of said action are the same as if we discard (D) with the additional immediate reward earned $\mathbb{E}[X|Y_{(n)}] - C$ where C

is a client defined hesitation constant which is used as a quality controlling tuner in the system.

The distributions of X and Y are known to us but we need to use the random variable $X|Y_{(n)}$ in order to include all the information the processor knows about the currently held item from their observations. We have that

$$X|Y_{(n)} = y_{(n)} \sim N\left(\frac{(\sigma^2+\nu^2)y_{(n)} + \frac{\tau^2}{n}\xi}{\sigma^2+\nu^2+\frac{\tau^2}{n}}, \frac{(\sigma^2+\nu^2)\frac{\tau^2}{n}}{\sigma^2+\nu^2+\frac{\tau^2}{n}}\right) \tag{3.14}$$

and from this we obtain

$$V_t(\xi, \nu^2, y_{(n)}, n|P) = V_t(\xi, \nu^2, y_{(n)}, n|D) + \mathbb{E}[X|Y_{(n)}] - C. \tag{3.15}$$

In the next section it will be shown more explicitly how C acts as an effective quality control.

We now have the recursive value functions conditional on the discarding (D) and passing (P) actions so we just need to state the V conditional on the remaining action of re-assessing (R) the currently held item.

When re-assessing occurs, it is important that processors use the information obtained from the observations $y_{(n)}$ to inform their prediction for the value of the $(n+1)^{st}$ observation y_{n+1}. The reader is advised to note the subtle difference in notation. We also need to make use of the previously unmentioned r.v. $Y_{n+1}|Y_{(n)}$ and notice that we can rewrite this r.v. in the following way

$$Y_{n+1}|Y_{(n)} = X|Y_{(n)} + \zeta|Y_{(n)} \tag{3.16}$$

and note that the observation error ζ for the $(n+1)^{st}$ observation is independent of both X and $Y_{(n)}$. Since $\zeta|Y_{(n)} \sim N(0, \tau^2)$ we deduce that

$$Y_{n+1}|Y_{(n)} \sim N\left(\frac{(\sigma^2+\nu^2)y_{(n)} + \frac{\tau^2}{n}\xi}{\sigma^2+\nu^2+\frac{\tau^2}{n}}, \frac{(\sigma^2+\nu^2)\frac{\tau^2}{n}}{\sigma^2+\nu^2+\frac{\tau^2}{n}} + \tau^2\right). \tag{3.17}$$

So we can now explicitly state the value function $V_t(\xi, \nu^2, y_{(n)}, n|R)$ conditional on taking the re-assessing action.

$$V_t(\xi, \nu^2, y_{(n)}, n|R) = \int_{-\infty}^{\infty} V_{t-1}\left(\xi, \nu^2, \frac{ny_{(n)} + y_{n+1}}{n+1}, n+1\right)\phi\left(\frac{y_{n+1} - \frac{(\sigma^2+\nu^2)y_{(n)}+\frac{\tau^2}{n}\xi}{\sigma^2+\nu^2+\frac{\tau^2}{n}}}{\left(\frac{(\sigma^2+\nu^2)\frac{\tau^2}{n}}{\sigma^2+\nu^2+\frac{\tau^2}{n}} + \tau^2\right)^{\frac{1}{2}}}\right)dy_{n+1} \tag{3.18}$$

It is now theoretically possible for us to evaluate V_t by exhaustive recursion and choosing the action that maximises total expected rewards. A reward is earned when an item is passed along for analysis. We discourage the passing of low quality items by subtracting the fixed parameter C from all individual rewards earned by single items.

The resulting effect is that the processor tends not to submit items which have an expected score which is less than the hesitation constant C. We discuss this further in the next section.

3.3 The hesitation constant as a performance tuner

In this section some key results underpinning the discussion in this section are set out. The first result shows us that the hesitation constant C has the property whereby we can increase its value in order to reduce the rate at which items are passed for analysis.

If we have a set Π of allowable policies our goal then is to choose $\pi \in \Pi$ which maximises the Bayes return

$$\mathbb{E}_\pi\left[\left(\sum_{i=1}^{N} X_i\right) - NC\right] \tag{3.19}$$

where N is the total number of items passed along for analysis and is a random variable with properties determined by the choice of π. It has been stated that we can use the constant C as a tuner to control the quantity of items passed along for analysis. For a fixed time period T we write $N(T)$ for the number of items passed

for analysis by time T.

Let $C_1 > C_2$ and suppose $\pi(C_1)$, $\pi(C_2)$ are the optimising policies for C_1 and C_2 respectively. We have,

$$\begin{aligned}
& \mathbb{E}_{\pi(C_2)}\left[\sum_{i=1}^{N(T)} X_i\right] - \mathbb{E}_{\pi(C_2)}\left[N(T)\right]C_1 \\
\leq \quad & \mathbb{E}_{\pi(C_1)}\left[\sum_{i=1}^{N(T)} X_i\right] - \mathbb{E}_{\pi(C_1)}\left[N(T)\right]C_1 \\
\leq \quad & \mathbb{E}_{\pi(C_1)}\left[\sum_{i=1}^{N(T)} X_i\right] - \mathbb{E}_{\pi(C_1)}\left[N(T)\right]C_2 \\
\leq \quad & \mathbb{E}_{\pi(C_2)}\left[\sum_{i=1}^{N(T)} X_i\right] - \mathbb{E}_{\pi(C_2)}\left[N(T)\right]C_2 \\
\rightarrow \quad & (C_1 - C_2)\mathbb{E}_{\pi(C_1)}\left[N(T)\right] \leq (C_1 - C_2)\mathbb{E}_{\pi(C_2)}\left[N(T)\right] \qquad (3.20)
\end{aligned}$$

and hence that $\mathbb{E}_{\mu(C)}\left[N(T)\right]$ is decreasing in C. Informally, one can reach the final implication by writing the above inequality as $A \leq B \leq C \leq D$ and deducing that $C - B \leq D - A$.

The first and third inequalities in (3.20) follow as $\pi(C_1)$ and $\pi(C_2)$ are respectively the optimising policies for C_1 and C_2, and the second inequality holds because the optimising policy for C_1 results in an even greater value for the objective when C_2 is the cost per allocation instead of C_1.

This analysis supports the use of C as a tuning parameter to control the mean rate at which items are passed to the analyst.

3.4 Bayesian updating

It is important that it is understood how the Bayesian updates such as those in (3.12) and (3.13) are derived. For ease of the discussion that follows, we use the notation ξ_0 and ν_0^2 to respectively denote the initial prior mean and prior variance for μ before any observations are made on any item at the very start of the problem horizon.

For μ, a Normal (ξ_0, ν_0^2) prior is assumed and we have a Normal likelihood

function for observed data $y_{(n)}$. If n observations have been made of the 1^{st} item drawn, the posterior density for μ is

$$p(\mu|y_{(n)}) \propto \exp\left\{ -\frac{1}{2\nu_0^2}(\mu-\xi_0)^2 \right\} \exp\left\{ -\frac{1}{2(\frac{\tau^2}{n}+\sigma^2)}(y_{(n)}-\mu)^2 \right\} \tag{3.21}$$

where if we expand the exponent terms in (3.21) we obtain a quadratic of the form $-\frac{1}{2}(a\mu^2 - 2b\mu + c)$ where c is a function of known constants only. We obtain the quadratic and linear terms

$$a = \frac{1}{\nu_0^2} + \frac{1}{\frac{\tau^2}{n}+\sigma^2}, \qquad b = \frac{\xi_0}{\nu_0^2} + \frac{y_{(n)}}{\sigma^2+\frac{\tau^2}{n}} \tag{3.22}$$

where algebraic manipulation gives us that the posterior distribution is also in the correct Normal form:

$$\begin{aligned} p(\mu|y_{(n)}) &\propto \exp\left\{ -\frac{1}{2}(a\mu^2 - 2b\mu) \right\} \\ &= \exp\left\{ -\frac{1}{2}a\left(\mu^2 - \frac{2b\mu}{a} + \frac{b^2}{a^2}\right) + \frac{b^2}{2a} \right\} \\ &\propto \exp\left\{ -\frac{1}{2}a\left(\mu - \frac{b}{a}\right)^2 \right\} \\ &= \exp\left\{ -\frac{1}{2}\left(\frac{\mu - b/a}{1/\sqrt{a}}\right)^2 \right\}. \end{aligned} \tag{3.23}$$

So we have that the posterior distribution for μ has a normal density with variance parameter

$$a^{-1} = \frac{\nu_0^2(\sigma^2+\frac{\tau^2}{n})}{\frac{\tau^2}{n}+\sigma^2+\nu_0^2} \tag{3.24}$$

and mean parameter

$$b/a = \frac{\xi_0(\frac{\tau^2}{n}+\sigma^2) + \nu_0^2 y_{(n)}}{\frac{\tau^2}{n}+\sigma^2+\nu_0^2} \tag{3.25}$$

where a and b are as in (3.22). The equations in (3.24) and (3.25) explicitly show how updates of belief state about μ occur as items are trashed or passed. An induction argument is now given showing that any of our Bayesian updates in our

model take a general form arising from this method.

For ease of notation I define the sets

$$\mathbb{K}_{i,j} = ([1, i] - \{j\}) \cap \mathbb{N}. \tag{3.26}$$

In words, $\mathbb{K}_{i,j}$ is the set of the first i natural numbers with the exception of j.

We also need to make use of the notation n_j which denotes the number of observations that were made on the j^{th} item drawn. We also temporarily alter our notation and let $\bar{y}_j$ denote the sample mean of the observations made of the j^{th} item.

Claim: When we discard or pass the i^{th} item in the problem and when $i \geq 2$, the following holds when we update to obtain ξ_i and ν_i^2 from ξ_0 and ν_0^2.

$$\begin{aligned} \xi_i &= \frac{\xi_0 \prod_{j=1}^{i}(\frac{\tau^2}{n_j} + \sigma^2) + \nu_0^2 \sum_{j=1}^{i} \bar{y}_j \prod_{k \in \mathbb{K}_{i,j}}(\frac{\tau^2}{n_k} + \sigma^2)}{\prod_{j=1}^{n}(\frac{\tau}{n_j} + \sigma^2) + \sum_{j=1}^{i} \nu_0^2 \prod_{k \in \mathbb{K}_{i,j}}(\frac{\tau^2}{n_k} + \sigma^2)} \qquad (3.27) \\ \nu_i^2 &= \frac{\nu_0^2 \prod_{j=1}^{i}(\frac{\tau^2}{n_j} + \sigma^2)}{\prod_{j=1}^{i}(\frac{\tau}{n_j} + \sigma^2) + \sum_{j=1}^{i} \nu_0^2 \prod_{k \in \mathbb{K}_{i,j}}(\frac{\tau^2}{n_k} + \sigma^2)} \qquad (3.28) \end{aligned}$$

Proof: The derivation of the posterior mean part of the claim will be shown first and then the proof will be completed by showing the posterior variance part of the proof.

Proceeding by induction, it is trivial to show that the posterior mean and variance we would obtain after we have discarded or passed the second item take the forms above. Now if I assume that the claim holds for every natural number from 2 to i I just need to show the inductive step which gives us the $(i+1)^{st}$ case from the i^{th}. When we pass or discard the i^{th} item we update to our posterior

mean for μ as follows.

$$\xi_{i+1} = \frac{\xi_i(\frac{\tau^2}{n_{i+1}} + \sigma^2) + \nu_t^2 \bar{y}_{i+1}}{\frac{\tau^2}{n_{i+1}} + \sigma^2 + \nu_t^2} \tag{3.29}$$

We choose to break up the derivation by considering the numerator of (3.29) first and show that we can obtain the $(i+1)^{st}$ case for the numerator. When we substitute in for ξ_i we can take the numerator of (3.27), and multiply it by $(\frac{\tau^2}{n_{i+1}} + \sigma^2)$. We pass the denominator of (3.27) to the denominator of (3.29) and deal with it later. The numerator of (3.29) becomes

$$\left[\xi_0 \prod_{j=1}^{i}\left(\frac{\tau^2}{n_j} + \sigma^2\right) + \nu_0^2 \sum_{j=1}^{i} \bar{y}_j \left(\prod_{k \in \mathbb{K}_{i,j}} \left(\frac{\tau^2}{n_k} + \sigma^2\right)\right)\right]\left[\frac{\tau^2}{n_{i+1}} + \sigma^2\right] + [\nu^2 \bar{y}_{i+1}] \tag{3.30}$$

so if we multiply out the pair of square brackets in (3.30) we obtain

$$\xi_0 \prod_{j=1}^{i+1}\left(\frac{\tau^2}{n_j} + \sigma^2\right) + \left[\nu_0^2 \sum_{j=1}^{i+1} \bar{y}_j \left(\prod_{k \in \mathbb{K}_{i+1,j}} \left(\frac{\tau^2}{n_k} + \sigma^2\right)\right) - \nu_0^2 \bar{y}_{i+1} \prod_{k \in \mathbb{K}_{i+1,i+1}} \left(\frac{\tau^2}{n_k} + \sigma^2\right)\right] + \nu^2 \bar{y}_{i+1} \tag{3.31}$$

where if we take the numerator of (3.28) when we substitute for ν_i^2 in (3.31), again leaving the denominator of (3.31) for now, we find that since the numerator of (3.28) is

$$\nu_0^2 \prod_{j=1}^{i}\left(\frac{\tau^2}{n_k} + \sigma^2\right) = \nu_0^2 \prod_{k \in \mathbb{K}_{i+1,i+1}} \left(\frac{\tau^2}{n_k} + \sigma^2\right) \tag{3.32}$$

which gives us the term we need to cancel out the negative term in (3.31) leaving us with the numerator

$$\xi_0 \prod_{j=1}^{i+1}\left(\frac{\tau^2}{n_j} + \sigma^2\right) + \left[\nu_0^2 \sum_{j=1}^{i+1} \bar{y}_j \left(\prod_{k \in \mathbb{K}_{i+1,j}} \left(\frac{\tau^2}{n_k} + \sigma^2\right)\right)\right] \tag{3.33}$$

which gives us the numerator for ξ_{i+1} which fits with our claimed form for it. Now we need to verify that the denominator is also of the correct form. Remember that during the course of evaluating the numerator we delayed dealing with the

following quotient

$$\frac{1}{\prod_{j=1}^{n}\left(\frac{\tau}{n_j}+\sigma^2\right)+\sum_{j=1}^{i}\nu_0^2\prod_{k\in\mathbb{K}_{i,j}}\left(\frac{\tau^2}{n_k}+\sigma^2\right)} \tag{3.34}$$

when substituting for both ξ_i and ν_i^2. We could have taken out this quotient initially as it is a common factor of the two terms in the numerator of (3.29). We can multiply the denominator of (3.29) by the inverse of (3.34) and evaluate the resulting expression then check it against our proposed form for the denominator of ξ_{i+1}.

We first obtain the denominator of ξ_{i+1} to be

$$\left(\prod_{j=1}^{n}\left(\frac{\tau}{n_j}+\sigma^2\right)+\sum_{j=1}^{i}\nu_0^2\prod_{k\in\mathbb{K}_{i,j}}\left(\frac{\tau^2}{n_k}+\sigma^2\right)\right)\left(\frac{\tau^2}{n_{i+1}}+\sigma^2+\nu_i^2\right) \tag{3.35}$$

Now we substitute ν_i^2 in (3.35) but notice that if we put everything inside the rightmost bracket of (3.35) over a common denominator (the denominator of (3.34)) we find that the leftmost bracket will cancel out with this denominator leaving us with only the numerator of the rightmost bracket. In doing all of this we find that (3.35) is equivalent to

$$\left(\frac{\tau^2}{n_{i+1}}+\sigma^2\right)\left[\prod_{j=1}^{i}\left(\frac{\tau^2}{n_j}+\sigma^2\right)+\sum_{j=1}^{i}\nu_0^2\prod_{k\in\mathbb{K}_{i,j}}\left(\frac{\tau^2}{n_k}+\sigma^2\right)\right]+\nu_0^2\prod_{j=1}^{i}\left(\frac{\tau^2}{n_j}+\sigma^2\right). \tag{3.36}$$

If we simplify (3.36) we get precisely the denominator we need for ξ_{i+1} and complete this part of the induction argument.

So for the variance part of the proof we already established that the basis case was true in the induction. Now we need to be able to obtain the specified form (3.28) for the $(i+1)^{st}$ given that we assume the i^{th}. Since we would be using $\hat{\nu}_i^2$ as the prior variance for μ, the next posterior variance to be computed would be

$\hat{\nu}^2_{i+1}$ and it is done as follows.

$$\nu^2_+ = \frac{\nu^2(\sigma^2 + \frac{\tau^2}{n_i})}{\frac{\tau^2}{n_i} + \sigma^2 + \nu^2} \tag{3.37}$$

If we let ϕ and ψ respectively denote the numerator and denominator of (3.28) we find that (3.37) is equal to

$$\nu^2_+ = \frac{\phi(\sigma^2 + \frac{\tau^2}{n_i})}{\psi\left(\frac{\psi(\frac{\tau^2}{n_i}+\sigma^2)+\phi}{\psi}\right)} = \frac{\phi(\sigma^2 + \frac{\tau^2}{n_i})}{\psi(\frac{\tau^2}{n_i} + \sigma^2) + \phi} \tag{3.38}$$

And if we substitute back in the values of ϕ and ψ we find that we have precisely the form we need for the posterior variance. □

3.5 Numerical Implementation

The main computational challenge of working with this model is evaluating the value function $V_t(\xi, \nu^2, y_{(n)}, n)$ from (3.10), (3.11), (3.15) and (3.18). It has four arguments, three of which are continuous. The range of both ξ and $y_{(n)}$ is $(-\infty, \infty)$ whilst ν^2 has range $[0, \infty)$ and while this alone may not be too much of an implementation issue, one also has to account for the fact that the value function also takes the time remaining t as an argument and also that V is recursively defined. For example we know from (3.11) that

$$V_t(\xi, \nu^2, y_{(n)}, n|D) = \int_{-\infty}^{\infty} V_{t-1}(\xi_+, \nu^2_+, y_{(1)}, 1)\phi\left(\frac{y_{(1)} - \xi_+}{(\sigma^2 + \nu^2_+ + \tau^2)^{\frac{1}{2}}}\right) dy_{(1)}. \tag{3.39}$$

One requirement for evaluating the above integral is that for all $y_{(1)}$ in the range $(-\infty, \infty)$ we have available to us a corresponding value for $V_{t-1}(\xi_+, \nu^2_+, y_{(1)}, 1)$. Bearing in mind that V_{t-1} itself requires infinitely many V_{t-2} and so on and so forth, it becomes immediately obvious that some form of approximation scheme for V needs to be deployed. However any approximation error in evaluating V_t at time $t = 0$ will be inherited when evaluating it at $t = 1$ and so on. It is therefore

important to keep approximation errors at a minimum as we step backwards in time.

One way of doing this is to construct a lattice of points that encompasses the four dimensional statespace (implied by $V_t(\xi, \nu^2, y_{(n)}, n)$) as much as possible and use this to discretise the continuous parts of the statespace to a level of detail that adequately trades off timeliness (or indeed, tractability) and accuracy. Each point in the lattice corresponds to a particular state the system can take, so that when one needs to compute the value of a state that is between these points, one will interpolate within the hypercube, of lattice points it is contained in.

The only other case that needs to be dealt with is the instance where the system requires an evaluation of V_t at a state existing outside of the range of the lattice. In this case one is forced to submit the value V_t corresponding to the nearest point of the statespace that is included within the hypercube but generally the lattice ranges should be chosen to avoid these cases as much as possible. At least the hypercube that is chosen should be large enough so that the value function returns for parameter combinations that exist outside of it contribute very little to the computation of V_t.

So:

- Construct four vectors of equally spaced values of ξ , ν, $y_{(n)}$ and n respectively.
- Let V_1 be a four dimensional array with dimensions corresponding to the length of the vectors in the previous step.
- For each element of V_1, compute $V_1(\xi, \nu^2, y_{(n)}, n|P)$ (as defined in 3.15) with parameters corresponding to the position within the array.
- For t in 2 to T construct V_t from V_{t-1}, where $V_t(\xi, \nu^2, y_{(n)}, n)$ is defined in 3.10, interpolating within V_{t-1} where necessary.
- Directly reference from V_T the desired value $V_T(\xi, \nu^2, y_{(n)}, n|D)$ or interpolate within the smallest hypercube of V_T containing that point.

One can use the psuedocode described in Algorithms 1-3 to approximate the values of $V_t(\xi, \nu^2, y_{(n)}, n|D)$, $V_t(\xi, \nu^2, y_{(n)}, n|P)$ and $V_t(\xi, \nu^2, y_{(n)}, n|R)$ required in computing $V_t(\xi, \nu^2, y_{(n)}, n)$.

Set : $Reward_D \leftarrow 0$
Carry out Bayesian updating that would occur as a result of trashing now;
Set : $\xi_+ \leftarrow \frac{\xi(\frac{\tau^2}{n}+\sigma^2)+\nu^2 y_{(n)}}{\frac{\tau^2}{n}+\sigma^2+\nu^2}$
Set : $\nu_+^2 \leftarrow \frac{\hat{\nu}^2(\frac{\tau^2}{n}+\sigma^2)}{\frac{\tau^2}{n}+\sigma^2+\nu^2}$
Create : vector $PhiArray$ and $InterpArray$ to be the same size as $YArray$
for $a \leftarrow 1$ **to** Dim_Y **do**
  Set : $y_{(1)} \leftarrow YArray[a]$
  Set : $PhiArray[a] \leftarrow \phi\left(\frac{y_1-\xi_+}{(\sigma^2+\nu_+^2+\tau^2)^{\frac{1}{2}}}\right)$
  where ϕ is the standard Normal cdf;
  Search : $XiArray$ for the consecutive pair of elements such that ξ_+ lies between those two values. Call the endpoints of that interval ξ_{low} and ξ_{high}
  Repeat : for $NuArray$ and ν_+^2, creating ν_{low}^2 and ν_{high}^2
  Interpolate: (linearly) between the four lattice points of $Final$ corresponding to the states $V_{t-1}(\xi_{low}, \nu_{low}^2, y_{(1)}, 1)$, $V_{t-1}(\xi_{low}, \nu_{high}^2, y_{(1)}, 1)$, $V_{t-1}(\xi_{high}, \nu_{low}^2, y_{(1)}, 1)$ and $V_{t-1}(\xi_{high}, \nu_{high}^2, y_{(1)}, 1)$ to obtain $V_{t-1}(\xi_+, \nu_+^2, y_{(1)}, 1)$
  Set : $InterpArray[a] \leftarrow V_{t-1}(\xi_+, \nu_+^2, y_{(1)}, 1)$
end
The arrays PhiArray and InterpArray are now fully populated;
Set : $Reward_D \leftarrow \frac{\sum_{i=1}^{Dim_Y} InterpArray[i]PhiArray[i]}{\sum_{i=1}^{Dim_Y} PhiArray[i]}$
return $Reward_D$;

Algorithm 1: Evaluating $V_t(\xi, \nu^2, y_{(n)}, n|D)$

Retrieve: $Reward_D$ from Algorithm 2
Set : $Reward_P \leftarrow Reward_D + \frac{(\sigma^2+\nu^2)y_{(n)}+\frac{\tau^2}{n}}{\sigma^2+\nu^2+\frac{\tau^2}{n}}$
return $Reward_P$;

Algorithm 2: Evaluating $V_t(\xi, \nu^2, y_{(n)}, n|P)$

An optional addition to the above procedure is to create the arrays A_t which track the action taken by the processor from any given state i.e. the maximising action from the options of passing, trashing and re-assessing the currently held item.

Set : $Reward_R \leftarrow 0$
Create : vector $PhiArray$ and $InterpArray$ to be the same size as $YArray$
for $a \leftarrow 1$ **to** Dim_Y **do**
  Set : $y_{n+1} \leftarrow YArray[a]$
  Set : $y_+ \leftarrow \frac{ny_{(n)}+y_{n+1}}{n+1}$
  Set : $PhiArray[a] \leftarrow \phi\left(\frac{y_{n+1}-\frac{(\sigma^2+\nu^2)y_{(n)}+\frac{\tau^2}{n}\xi}{\sigma^2+\nu^2+\frac{\tau^2}{n}}}{\left(\frac{(\sigma^2+\nu^2)\frac{\tau^2}{n}}{\sigma^2+\nu^2+\frac{\tau^2}{n}}+\tau^2\right)^{\frac{1}{2}}}\right)$
  where ϕ is the standard Normal cdf;
  Search : $YArray$ for the consecutive pair of elements such that y_+ lies between those two values. Call the endpoints of that interval y_{low} and y_{high}
  Interpolate: (linearly) between the pair of lattice points of $Final$ corresponding to the states $V_{t-1}(\xi, \nu^2, y_{low}, n)$ and $V_{t-1}(\xi, \nu^2, y_{high}, n)$ and linearly to obtain $V_{t-1}(\xi, \nu^2, y_{(+)}, n)$
  Set : $InterpArray[a] \leftarrow V_{t-1}(\xi, \nu^2, y_{(+)}, n)$
end
The arrays PhiArray and InterpArray are now fully populated;
Set : $Reward_R \leftarrow \frac{\sum_{i=1}^{Dim_Y} InterpArray[i]PhiArray[i]}{\sum_{i=1}^{Dim_Y} PhiArray[i]}$
return $Reward_R$;

Algorithm 3: Evaluating $V_t(\xi, \nu^2, y_{(n)}, n|R)$

```
, , 10, 4

      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
 [1,] "T"  "T"  "T"  "R"  "R"  "R"  "R"  "R"  "R"  "R"
 [2,] "T"  "T"  "T"  "R"  "R"  "R"  "R"  "R"  "R"  "R"
 [3,] "T"  "T"  "T"  "R"  "R"  "R"  "R"  "R"  "R"  "R"
 [4,] "T"  "T"  "T"  "R"  "R"  "R"  "R"  "R"  "R"  "R"
 [5,] "T"  "T"  "T"  "R"  "R"  "R"  "R"  "R"  "R"  "R"
 [6,] "R"  "R"  "R"  "R"  "R"  "R"  "R"  "R"  "R"  "R"
 [7,] "T"  "T"  "T"  "R"  "R"  "R"  "R"  "R"  "R"  "R"
 [8,] "R"  "R"  "R"  "R"  "R"  "R"  "R"  "R"  "R"  "R"
 [9,] "P"  "P"  "P"  "P"  "P"  "P"  "P"  "P"  "P"  "P"
[10,] "P"  "P"  "P"  "P"  "P"  "P"  "P"  "P"  "P"  "P"
```

Figure 3.1: A sample of from the numerical output

By doing this it should be possible in most cases to directly recommend an action to the processor given their current problem state $(\xi, \nu^2, y_{(+)}, n)$ if it lies within the bounds of the created lattice. It is unclear what should be done in the event that the processor must 'interpolate' between non-identical actions. One could choose the action corresponding to action specified at the nearest neighbouring lattice point as a rule of thumb.

I have been able to produce some working code that produces such an action array A_t; an example screen of the output is shown in Figure 3.1. A two dimensional slice of the larger four dimensional action lattice is shown which shows a contiguous region within which the processor is compelled to re-assess their currently held item, which separates two similar such regions where the processor is respectively compelled to trash or pass on her current item.

I've carefully chosen the parameters for this toy problem to show what happens at the borderline between discarding and passing items. Being able to see this behaviour requires that the fidelity of the experiment is fine enough to capture the re-assessment region. In other test cases on a more macroscopic scale, it is common for this interim area to be missed out entirely, having a lattice which jumps from trashing to passing and vice versa.

However, the method takes considerable time to execute and is unsuitable for

anything more than a moderately sized lattice. Given that we ultimately wish to extend to multiple source problems, it is clear that a lattice type approach such as this may not be suitably developed to solve those problems. The level of complexity in the systems we would wish to consider is only going to increase from this stage. A completely exhaustive numerical approximation, even at this level of discretisation, is certain to suffer from dimensionality related difficulties and severely limit the progress that can be made in future research efforts.

We seek instead index-based heuristics to guide our processor and analyst in our intelligence operations models. The exploration and development of these index-based decision approaches form the basis for the content in the research of the chapters to come.

Chapter 4

The Multi-Armed Bandit Allocation Problem

4.1 Overview

This chapter develops a novel variant of the multi-armed bandit problem. Its name is the multi-armed bandit allocation (or *MABA*) model. For those familiar with the literature of bandit problems or for those that have read the relevant material in Chapter 2, the term *source* is used instead of 'arm' when discussing the various sampling choices available throughout the problem horizon. This language is primarily used to serve the intelligence operations setting but will be used from this point onwards.

The crucial difference between MABA and other multi-armed bandit problems is that only a subset of the sampled rewards can be *allocated* and count towards the problem solver's final reward, and the total number of allocations must be made within a finite number of decision epochs. The non-allocated rewards are permanently discarded and count as zero reward at the end of the problem. The decisions to allocate or not are made in a sequential fashion so the problem becomes a matter of efficiently searching for the best sources from which to sample, but also the best way to sequentially allocate the sampled rewards while this search

takes place.

More technically, the multi-armed bandit allocation problem models a situation in which there are K *arms* or *sources* which can yield rewards, only one of which can be *visited* at each time epoch $t \in \mathbb{N} \cap [1, T]$. Once a source has been visited and its current reward observed, a subsequent decision is required concerning whether that reward should be *allocated*. Over a horizon of length T there is a limit of $M = (1-q)T$ of rewards which can be allocated, where $q \in (0, 1)$. Typically $1 - q \ll 1$. Sources evolve in a Markovian fashion when they are visited, but remain unchanged otherwise. The goal is to design a policy, namely a rule for taking decisions regarding *both* sources to visit and rewards to be allocated to maximise the expected total reward allocated.

To be more precise, we introduce the *MABA* $\{(\Omega_i, R_i, P_i), 1 \leq i \leq K; T, M\}$ as follows:

1. The *state space of source* i is denoted by $\Omega_i, 1 \leq i \leq K$. For the rest of this section we shall assume that each such state space is finite or countable, though our later theory will apply more widely;

2. The *state* of the system at each decision epoch $t \in \mathbb{N} \cap [1, T]$ is written $\mathbf{I}(t) = \{I_1(t), I_2(t), \ldots, I_K(t)\}$, where $I_i(t) \in \Omega_i$ is the state of source i at t;

3. A policy π is a rule that decides which source to be visited at each epoch t and mandates a decision concerning whether the reward available at the source should be allocated or not. Policies are allowed to depend upon the entire history of the process to date (states observed, actions taken) and may be randomised;

4. Should source i be visited at epoch t then a reward $R_i\{I_i(t)\}$ is available to be allocated (or not), where $R_i : \Omega_i \to \mathbb{R}^+$ is source i's *return function,* assumed positive-valued. The state of source i also undergoes a transition

$I_i(t) \to I_i(t+1)$ determined by Markovian law P_i. The states of sources not visited at t are frozen;

5. The total number of rewards to be allocated over horizon T is bounded above by M, where $M \leq T$. Typically, we have $M \ll T$.

The MABA model is designed for simplicity, but also to allow for extensions and modules to be added to it for particular applications. Two key assumptions that adhere to this design philosophy are the assumptions of a single processor, and the assumption that discarded items are not accessible to the processor for the rest of time. Relaxing either or both of these assumptions would be valid actions to take but would add unnecessary complexity to the model which is why we've elected for the simpler option in these cases. A single processor model is simpler than having a model which assumes an array of many processors and the reason for choosing to prefer a single processor model, other than for simplicity, is that the policies developed to solve the single processor variant of the MABA can be issued to quasi-autonomous processors individually and we assume that solutions of this type are easier to implement that those that apply to a multi-processor problem. Such multi processor solutions would typically have to be supported by a team management system of some kind, which would require even further, arguably unnecessary, model complexity. If a particular use case would demand a multi-processor solution that could not be comprised from many single processor solutions, then this complexity could be added to the single processor MABA model that we've chosen to proceed with, although this would require additional modelling effort. As for permanent discarding of items, we assume that the processor can not revisit previously discarded items, even if they later regret their decision to discard a particular item. This assumption, if deemed unrealistic for practice, could be remedied by the inclusion of a finite sized 'folder' of items to be considered for re-assessment. Doing so would offer the processor three additional actions at each decision epoch. First, they could store a sampled item in the 'folder' of items which have neither been discarded or allocated. Second and third,

the processor could allocate or permanently discard items from the 'folder'. Increasing the size of the state and action space like this would ameliorate concerns about permanent discards but would also come at the cost of additional model dimensionality, so we've elected to accept the problems of permanent discards in the MABA and move ahead.

In what follows we shall write $\pi(t)$ for the source visited under policy π at epoch t and $X_\pi(t)$ for the indicator which takes the value 1 when the reward available at the source visited by π at t is allocated and is 0 otherwise. We now express the problem P associated with the *MABA* $\{(\Omega_i, R_i, P_i), 1 \leq i \leq K; T, M\}$ as follows:

$$(P) : R(T, M) := \sup_\pi E\left[\sum_{t=1}^{T} R_{\pi(t)}\left\{I_{\pi(t)}(t)\right\} X_\pi(t)\right];$$

$$\text{such that } \sum_{t=1}^{T} X_\pi(t) \leq M. \tag{4.1}$$

Plainly $R(T, T) := R(T)$ is the expected return from the corresponding conventional *MAB* in which all rewards made available are allocated, namely in which the constraint (4.1) becomes

$$\sum_{t=1}^{T} X_\pi(t) \leq T \tag{4.2}$$

and hence vacuous. The following result is immediate.

Lemma 4.1.1. $R(T) \geq R(T, M) \geq R(M), \forall M \leq T.$

Solutions for the conventional *MAB* in which all rewards are allocated are difficult to implement directly. We therefore seek simplification through developing relaxations of P which are more amenable to analysis. We do this by considering a version of the key constraint (4.1) based on expected values. We thus develop problem P^* as follows:

$$(P^*) : R^*(T, M) := \sup_\pi E\left[\sum_{t=1}^{T} R_{\pi(t)}\left\{I_{\pi(t)}(t)\right\} X_\pi(t)\right];$$

$$\text{such that } E\left\{\sum_{t=1}^{T} X_{\pi}(t)\right\} \leq M. \tag{4.3}$$

A rather poor way of solving P^* would be to resolve the allocation part of the problem by simply randomising to allocation (rather than to non-allocation) at each epoch independently with probability $\frac{M}{T}$. This, together with the evident fact that P^* is indeed a relaxation of P yields the following:

Lemma 4.1.2. $R(T) \geq R^*(T,M) \geq \max\left\{R(T,M), \frac{M}{T}R^*(T)\right\}, \forall M \leq T.$

We proceed toward a solution of P^* by developing a further relaxation based on Lagrangian techniques. We develop problem $P^*(C)$ for multiplier $C \in \mathbb{R}^+$ by dropping the constraint (4.3) and instead introduce into the objective for the problem penalties for violations of it. We write

$$(P^*(C)) : R^*(T,M,C) := \max_{\pi} E\left[\sum_{t=1}^{T}(R_{\pi(t)}\left\{I_{\pi(t)}(t)\right\} - C)X_{\pi}(t) + CM\right]. \tag{4.4}$$

In problem $P^*(C)$ the policy class is *unconstrained* with respect to the allocations made. In $P^*(C)$, there is no hard control over the number of allocations made and the flow of items from the processor to the analyst is instead controlled indirectly by the choice of C. Note that in (4.4) the multiplier C has an interpretation as a fixed cost incurred whenever an allocation is made. Please note that the standard theory of MDPs (see, for example, Puterman [1994]) implies that, for given C the maximum in (4.4) will be achieved by a policy whose decisions are deterministic and dependent upon $(t, \mathbf{I}(t))$ only. Further, it is easy to see that $R^*(T,M,C)$, being a maximum taken over an objective which is linear in C, is convex (and piecewise linear) in C. The breakpoints in $R^*(T,M,C)$ will correspond to points at which there are multiple optimal policies. We resolve any non-uniqueness in defined quantities by taking right limits (ie, as C is approached from above). In that spirit, we now write $\pi^*(C)$ for a policy which achieves $R^*(T,M,C)$,and hence which is optimal for $P^*(C)$ and $m^*(C)$ for the mean number of rewards allocated under $\pi^*(C)$. Finally, we use $\Delta^+R^*(T,M,C)$ for the right gradient of

$R^*(T, M, C)$, namely

$$\Delta^+ R^*(T, M, C) := \lim_{c \to C^+} \frac{\partial}{\partial c} R^*(T, M, c). \tag{4.5}$$

We now have the following result:

Proposition 4.1.3. *(i)* $m^*(C)$ *is nonincreasing in* C; *(ii)* $min_{C \in \mathbb{R}^+} R^*(T, M, C) = R^*(T, M)$.

Proof. For (i), we observe from (4.4) that

$$\Delta^+ R^*(T, M, C) = -m^*(C) + M, \tag{4.6}$$

and this is nondecreasing because of the convexity (in C) of $R^*(T, M, C)$. It follows immediately that $m^*(C)$ is nonincreasing, as required. For (ii) we first note that it is trivial from the definitions of the quantities concerned that $R^*(T, M, C) \geq R^*(T, M), C \in \mathbb{R}^+$, and hence that $\min_{C \in \mathbb{R}^+} R^*(T, M, C) \geq R^*(T, M)$. To obtain the reverse inequality we observe trivially that

$$m^*(0) = T > M, \tag{4.7}$$

namely that if there is no cost associated with allocation ($C = 0$) then all rewards will be allocated. Similarly, we must have that

$$\lim_{C \to \infty} m^*(C) = 0 \tag{4.8}$$

and hence from (4.6), we infer that

$$\Delta^+ R^*(T, M, 0) < 0, \tag{4.9}$$

and

$$\lim_{C \to \infty} \Delta^+ R^*(T, M, C) > 0. \tag{4.10}$$

It must then follow that one of two possibilities must occur. The first is that there exists some value C^* for which

$$\Delta^+ R^* (T, M, C^*) = 0 \Rightarrow m^* (C^*) = M \tag{4.11}$$

and hence that the corresponding optimal policy $\pi^* (C^*)$ satisfies the constraint (4.3) with equality. It must then follow that

$$R^* (T, M, C^*) = E \left[\sum_{t=1}^{T} R_{\pi^*(C^*)(t)} \left\{ I_{\pi^*(C^*)(t)} (t) \right\} X_{\pi^*(C^*)} (t) \right] \leq R^* (T, M) \tag{4.12}$$

from which we immediately infer that

$$min_{C \in \mathbb{R}^+} R^* (T, M, C) \leq R^* (T, M), \tag{4.13}$$

as required. This establishes (ii) for such cases. The second possibility is that while there is no C-value at which $\Delta^+ R^* (T, M, C)$ is zero, nonetheless there exists some value C^* for which

$$\Delta^+ R^* (T, M, C) < 0, C < C^*, \tag{4.14}$$

and

$$\Delta^+ R^* (T, M, C) > 0, C > C^*. \tag{4.15}$$

When this happens it is easy to show that there must exist two distinct policies, $\pi^- (C^*)$ (optimal to the left at C^*) and $\pi^+ (C^*)$ (optimal to the right at C^*) , both of which achieve $R^* (T, M, C^*)$, and which satisfy

$$E \left\{ \sum_{t=1}^{T} X_{\pi^- (C^*)} (t) \right\} > M \tag{4.16}$$

and

$$E \left\{ \sum_{t=1}^{T} X_{\pi^+ (C^*)} (t) \right\} < M. \tag{4.17}$$

It will then follow that some randomisation between $\pi^- (C^*)$ and $\pi^+ (C^*)$ will

achieve constraint (4.3) with equality while continuing to achieve $R^*(T, M, C^*)$. Equation (4.12) then holds but with $\pi^*(C^*)$ replaced by this randomisation and we then infer

$$min_{C\in\mathbb{R}^+} R^*(T, M, C) \leq R^*(T, M), \tag{4.18}$$

as before. We have now established (ii) for all cases. This completes the proof. □

There is one further point. In problem $P^*(C)$, the allocation part of the policy is trivial. Plainly an allocation will be optimally made in respect of a reward R if and only if R is no less than the corresponding cost C. It follows that $P^*(C)$ is equivalent to a corresponding MAB problem (ie, in which all rewards are allocated) but with reward contributions R_iI replaced by $(R_i - C)^+$. Hence we can rewrite, with a slight abuse of notation, $P^*(C)$ in the form

$$(P^*(C)) : R^*(T, M, C) := \max_{\pi} E\left[\sum_{t=1}^{T} \left(R_{\pi(t)}\left\{I_{\pi(t)}(t)\right\} - C\right)^+ + CM\right], \tag{4.19}$$

where now the policy π only chooses sources and allocations occur precisely when a positive contribution is made to the above objective. In this sense problem $P^*(C)$ is a finite horizon MAB problem.

Motivated by the above, we could in principle adopt the following approach to developing heuristic policies for the *MABA problem P*. We develop heuristic policies for the relaxation P^* using the above ideas and we then adapt those to achieve policies for P. How this latter step is achieved will be described later, but we now sketch ideas concerning how the former step is accomplished. Using the above results, the following programme, if achievable would yield solutions to P^* :

- Fix C;
- Solve the MAB problem $P^*(C)$ in the form $\max_\pi E\sum_{t=1}^{T}\left[\left(R_{\pi(t)}\left\{I_{\pi(t)}(t)\right\} - C\right)^+\right]$;
  - Infer the value $m^*(C)$;
  - If $m^*(C) < M$, decrease C; if $m^*(C) > M$ increase C;
  - Iterate until $m^*(C)$ is sufficiently close to M;

- Use the final optimal policy $\pi^*(C)$ as the solution to P^*.

Sadly, a programme of that kind is unrealistic, not least because the problem $P^*(C)$ is itself intractable. Hence we shall necessarily adopt heuristic procedures to estimate a $C-$value, C_H say, close to C^* and a policy for $P^*(C_H), \pi_H(C_H)$ say, such that $\pi_H(C_H)$ is close to optimal for P^*. We can then adapt $\pi_H(C_H)$ to achieve a strongly performing policy for the *MABA* problem P.

The focal application of the MABA throughout this thesis is toward the end of aiding intelligence operations. However the generality of this model has potential for application in other areas. This chapter closes by speculating on how the MABA framework could be applied to an existing business setting.

4.1.1 Application of MABA to freemium app user acquisition strategy

In the realm of freemium app development, customers can install applications at a price point of zero. Apps such as Tinder, Spotify, Snapchat, and Skype monetize through a mixture of in-app purchases, which unlock additional functionality, or through exposing their users to advertising whilst using the product. The freemium model assumes that most of the users will spend nothing at all throughout their lifetime as a user and see many adverts, but click very few of them. Freemium apps make money by having an infintely scalable inventory, which is easily available to users through an established payment platform such as iTunes, Google Play or PayPal. If a sufficient number of users install the app, even if the probability of an individual spending money in the app is small, the apps in question can still make enough money to cover the costs of app development, app maintenance, and crucially, user acquisition costs. See [Seufert, 2014] for a more complete view on this industry and the material covered in this subsection.

Organic user acquisition is the phenomenon where users discover apps naturally without targeted advertising suggesting that they install the app. However, this meagre volume is rarely enough to sustain a freemium app's economy. It would be

ideal if every hopeful app developer could count on a their app 'going viral' and acquiring a vast amount of users and brand awareness through social networking shares by a large number of engaged users, but hoping for and even designing apps around this potential virality and doing no further paid user acquisiton is rarely a successful approach.

Users for apps are often acquired through user acquisition services that typically charge app developers a fixed rate per user that they persuade to install the developer's app. However it is not in the developers interest to pay to acquire users if the expected lifetime value (LTV) of that user is less than the cost of their acquisition. So the user acquisition services are presented with a MABA problem. They incur a cost in searching for users to allocate to the various apps in their client network. The various sources they have available for finding users will yield user cohorts with random expected LTV values (whose true LTV only becomes apparent after their allocation to an app) and they can decide on a case by case basis which users to allocate to client apps. There is an incentive to for the user acquisiton services to submit a limited subset of the total number of sampled users, such that the allocated subset are of the highest quality possible (in terms of LTV) such that they can claim a higher average LTV per allocated user than their rival user acquisition services. Additionally, these services need to rapidly find the best user sources for the client app in question (the right audience for the right app) such that the costs incurred in locating the best users are minimised.

We conclude by saying that there is untapped potential for user acquisiton services to provide a higher mean quality of users to their client app developers for a much reduced cost by applying MABA methods to their operations.

Chapter 5

A Dirichlet-Multinomial MABA model

In this chapter we apply the multi-armed bandit allocation (MABA) framework that was developed in Chapter 4 to the problem of intelligence gathering. The opening section of this chapter is dedicated to developing the Dirichlet-Multinomial MABA model. We develop solutions for the problem P via finding solutions to the problem $P^*(C)$ introduced in Chapter 4 and adapted to the Dirichlet-Multinomial setting here.

We then develop a Lagrangian relaxation of $P^*(C)$, which allows the processor to sample from any number of the available sources once per time epoch, provided she is willing to pay a charge W to do so. We refer to this version of the problem as $P^*(C, W)$. This approach is novel for this type of problem and forms an original contribution in this work.

In addition to the Lagrangian relaxation, we will be adapting existing types of heuristic to provide approximate solutions to P. Knowledge gradient, Thompson sampling and Optimistic Bayes sampling methods are also adapted to the MABA framework.

We will be using two methods for conducting studies using the existing heuristics. The first is a 'static C' approach, where a benchmark value of C is chosen

once and for all through the problem horizon. The second is a 'dynamic C' approach, where the value of C is allowed to change after observing the value of each sampled item. A section outlining these item allocation policies is presented.

The latter sections of this chapter document efforts to conduct numerical studies using combinations of heuristic source selection and item allocation policies.

The purpose of these numerical studies is twofold. Firstly, we wish to show that the framework described in Chapter 4 and the heuristics outlined in this Chapter can actually be implemented in a robust and scalable way. Secondly, we compare the relative performances of the various heuristic source selection policies within and between the static C and dynamic C frameworks. In doing so we intend to yield insights into the relative merits of the heuristic policies in each setting and to test whether their relative merits under one framework are consistent under the other.

A key assumption of this chapter's work is that all item importances take discrete values. The motivation for this is how we suspect the MABA model may be used in practice. We believe that real world intelligence operatives are more likely to assign discrete ratings of importance to intelligence items (Item A has an importance rating of 'Mission Critial' and Item B is considered to have a rating of 'Non-Urgent') as opposed to assigning continuous importance values (Item A has an importance rating of 5.43 and Item B is considered to have a rating of 1.34), and we take the former approach in this chapter. We recognise that the assumption of discrete item importance values is potentially restrictive so in Chapter 6 we consider a MABA model where continuous importance scores are supported.

5.1 Development of Dirichlet-Multinomial MABA model

In this section we develop the Dirichlet-Multinomial model for solving $P^*(C)$.

One supposes that the processor encounters a $MABA$ $\{(\Omega_i, R_i, P_i), 1 \leq i \leq K; T, M\}$

which corresponds to a finite-horizon intelligence gathering problem. The processor seeks to sample a total of T items sourced from up to K distinct sources. The convention of counting time periods is that t denotes the number of sampling decisions *remaining*.

We denote by $I_{k,t}, 1 \leq t \leq T$, the importance of the item sampled from source k when there are t decisions remaining. We set $I_{k,t} = 0$ wherever source k is not sampled when t samples remain. The processor seeks to sample from the sources in such a way as to solve the problem P,

$$(P) : \max \mathbb{E} \left\{ \sum_{k=1}^{K} \sum_{t=1}^{T} I_{k,t} X_t \right\}, \tag{5.1}$$

Subject to:

$$\sum_{t=1}^{T} X_t \leq \lfloor T(1 - q_h) \rfloor \tag{5.2}$$

where $X_t = \begin{cases} 1 & \text{if item is allocated (passed to analyst) at time t} \\ 0 & \text{otherwise,} \end{cases}$ and $0 < q_h < 1$.

The value q_h is called the *horizon quartile* and denotes the proportion of items not passed to the analyst by the processor.

The operational interpretation of this is that the processor is capable of processing a greater number of items than the analyst is capable of analyzing in the same amount of time. The processor aims to provide the analyst with a selection of the most important items from the items that she sees over the horizon. However the processor sees items in a sequential fashion.

The processor chooses her sources sequentially, sampling a single item from a single source in each time period. She then assigns an integer score out of N, where N is client-specified. We suppose that she is able to make a perfect judgement concerning the importance of each item sampled. As she progresses forwards through time, she will use this new information to update her posterior beliefs about the population of item importances for items sampled from the K sources.

The problem is similar to the General Secretary problem discussed by [Babaioff et al., 2007] where a manager must hire the best possible subset of candidates for secretarial positions given that she must interview the candidates in a random order and make a permanent decision whether to hire each individual candidate before interviewing the next one in the sequence. It is also true in this MABA problem that an item is discarded permanently by the processor if it is not immediately allocated after sampling.

The k^{th} source/bandit from a collection of K such objects is as follows: An item sampled from source k at time t has importance $I_{k,t}$ which has a multinomial distribution on $[1, 2, ..., N]$ with unknown $\mathbf{p_k}. \in [0, 1]^N$ Hence

$$P(I_{k,t} = i \mid \mathbf{p_k}) = p_{i,k}, 1 \leq i \leq N, 1 \leq k \leq K, 1 \leq t \leq T. \tag{5.3}$$

We define the N x K x T scalar arrays α and c and the K x T scalar arrays n and I and will refer to these objects in what follows. The initial uncertainty relating to $\mathbf{p_k}$ is described by a Dirichlet prior with parameter given by the N-vector $\alpha_{k,T}$, that is,

$$\pi\left(\mathbf{p_k}\right) = \frac{\Gamma\left(\sum_i \alpha_{i,k,T}\right)}{\Pi_i \Gamma\left(\alpha_{i,k,T}\right)} \Pi_i p_{i,k}^{\alpha_{i,k,T}-1}, p_{i,k} > 0, 1 \leq i \leq N, \sum_i p_{i,k} = 1. \tag{5.4}$$

The $\alpha_{i,k,t}$ terms update through time as follows

$$\alpha_{i,k,t-1} = \alpha_{i,k,t} + c_{i,k,t} \tag{5.5}$$

where $c_{i,k,t} \in \{0, 1\}$ denotes whether an item with precision equal to i was observed from source k at with t time periods remaining. The resulting predictive distribution of I_k is given by

$$P(I_{k,t} = i) = \int P(I_{k,t} = i \mid \mathbf{p_k}) d\pi\left(\mathbf{p_k}\right) = \frac{\alpha_{i,k,t}}{\sum_j \alpha_{j,k,t}}, 1 \leq i \leq N, \tag{5.6}$$

and the expected one-step reward for the model, taken with respect to the predic-

tive distribution, as is appropriate for the Bayes' reward, is given by

$$r\left(\alpha_{k,t}\right) = \sum_{i=1}^{N} iP(I_{k,t} = i) = \sum_{i=1}^{N} i\frac{\alpha_{i,k,t}}{\sum_j \alpha_{j,k,t}} \tag{5.7}$$

A multinomial likelihood is assumed for the observed count vector $c_{k,t} = (c_{i,k,t})_{i=1,\ldots,N}$. We have

$$P(c_{k,t}|p_k) \propto \prod_{i=1}^{N} p_{i,k}^{c_{i,k,t}} \tag{5.8}$$

which yields the posterior distribution

$$\pi_{t-1}(p_k) = \pi_t(p_k|c_{k,t}) \propto P(c_{k,t}|p_k)\pi_t(p_k) \propto \prod_{i=1}^{N} p_{i,k}^{\alpha_{i,k,t}+c_{i,k,t}-1}, \tag{5.9}$$

which is Dirichlet with parameter $\alpha_{k,t} + c_{k,t}$. Hence this is a conjugate structure. The update shown in (5.9) only takes place when a source is sampled.

For ease of notation we now write

$$\sum_{s=t}^{T} c_{i,k,s} = n_{i,k,t} \tag{5.10}$$

and

$$\sum_{i=1}^{N} n_{i,k,t} = n_{k,t} \tag{5.11}$$

The updated one-step reward for the next time period is given by

$$r\left(\alpha_{k,t-1}\right) = r\left(\alpha_{k,t}|\, c_{k,t}\right) = \sum_{i=1}^{N} i\frac{\left(\alpha_{i,k,t} + c_{i,k,t}\right)}{\left\{\left(\sum_j \alpha_{j,k,t}\right) + \sum_j c_{j,k,t}\right\}} \tag{5.12}$$

The quantity (5.12) will be greater for larger counts of high importance items. As the number of observations $n_{k,t}$ from source k gets large the effect of the observations will dominate that of the prior values $\alpha_{k,T}$.

In Chapter 4, it was proposed that we develop heuristic policies for MABA problems such as P in (5.2) via the analysis of suitable relaxations such as P^* and $P^*(C)$. The relaxation P^* is obtained from P by replacing the constraint on the number of items allocated to one involving the latter quantity's expected value.

In the case of the problem class we are considering here P^* takes the form:

$$(P^*) : \max \mathbb{E}\left\{\sum_{k=1}^{K}\sum_{t=1}^{T} I_{k,t}X_t\right\} \tag{5.13}$$

Subject to:

$$\mathbb{E}\left(\sum_{t=1}^{T} X_t\right) \leq \lfloor T(1-q_h)\rfloor. \tag{5.14}$$

It is now possible to apply a Lagrangian relaxation $P^*(C)$ by incorporating the constraint in (5.14) into the objective by using the Lagrangian multiplier $C \in \mathbb{R}^+$. From (4.4), $P^*(C)$ is equivalent (ie, has the same optimal policies as) to the MAB problem

$$(P^*(C)) : \max \mathbb{E}\left\{\sum_{k=1}^{K}\sum_{t=1}^{T} (I_{k,t}-C)^+\right\} \tag{5.15}$$

Despite dropping the constant term CM from (4.4), we shall continue with the $P^*(C)$ label for this problem.

The discussion in Chapter 4 yields the conclusion that in problem $P^*(C)$, only items whose importance exceeds C are passed on to the analyst. This is reflected in the form of the objective in (5.15). Looking ahead, once we have solution approaches to $P^*(C)$, the threshold C can be tuned to meet the constraint in (5.14) with equality. In this way we will develop solutions to P^*.

For problem $P^*(C)$, the one-step reward in (5.7) needs to be developed further to take the form

$$r\left(\alpha_{k,t},C\right) = \sum_{i=1}^{N} (i-C)^+ P(I_{k,t}=i) = \sum_{i\geq C} (i-C)\,\frac{\alpha_{i,k,t}}{\sum_j \alpha_{j,k,t}} \tag{5.16}$$

where the posterior one step expected reward in (5.12) is now modified to

$$\begin{aligned} r\left(\alpha_{k,t-1},C\right) = r\left(\alpha_{k,t},C|\, c_{k,t}\right) &= \sum_{i\geq C} (i-C)\,\frac{\alpha_{i,k,t}+c_{i,k,t}}{\sum_j(\alpha_{j,k,t}+c_{j,k,t})} \\ &= \sum_{i\geq C} (i-C)\,\frac{(\alpha_{i,k,T}+n_{i,k,t})}{\left\{\left(\sum_j \alpha_{j,k,T}\right)+n_{k,t}\right\}}. \end{aligned} \tag{5.17}$$

A relationship between one step rewards and the choice of C can be proven analytically. The next result states that these one-step rewards are decreasing in C.

Lemma 5.1.1. *The immediate expected reward* $r\left(\alpha_{k,t},C\right)$ *as defined in* (5.16) *is non-increasing in* C.

Proof. Let $C_1 > C_2$. We have that

$$r\left(\alpha_{k,t},C_2\right) - r\left(\alpha_{k,t},C_1\right) \geq \sum_{i \geq C_1} \left(C_1 - C_2\right) \frac{\alpha_{i,k,T} + n_{i,k,t}}{\sum_j (\alpha_{j,k,T}) + n_{k,t}} \geq 0.$$

hence the result. □

For any given policy for choosing which sources to sample, it is plain that we can decrease the number of items allocated by increasing the value of C. Further, from Proposition 4.0.3, we have that $m^*(C)$, the mean number of items allocated under $\pi^*(C)$, an optimal policy for $P^*(C)$, is nonincreasing in C. This means that it is relatively straightforward to search over values of C to find one C^* say, such that $m^*(C^*) = \lfloor T(1-q_h) \rfloor$. It will then follow from the ideas in Chapter 4 that $\pi^*(C^*)$ will solve P^*.

A finer search can be conducted by making use of the randomised acceptance approach developed to solve the dynamic C version of this problem. This tool is described in a later subsection in detail. See equation (5.39) for its implementation.

Evolving beliefs about the level of importance of items emerging from the K sources will influence the processor as she decides how to sample from the sources to obtain a collection of items for the analyst with high importance. We can now state the full DP for this version of the model. We have the recursively defined value function:

$$V_t(\alpha_t, C) = \max_{1 \leq k \leq K} \left\{ r\left(\alpha_{k,t}, C\right) + \sum_{i=1}^{N} P(I_{k,t} = i) V_{t-1}((\alpha_{j,t})_{j \neq k}; \alpha_{k,t} + 1^i, C) \right\};$$

$$V_1(\alpha_1, C) = \max_{1 \leq k \leq K} \{r\left(\alpha_{k,1}, C\right)\}. \tag{5.18}$$

Note that 1^i is an N-vector with i^{th} component 1 and zeroes elsewhere. $V_0(\alpha_0, C) = 0$ for all input values.

5.2 Lagrangian indices for the Discrete MABA problem

We wish to obtain approximate solutions to the problem $P^*(C)$. In this section we reformulate the problem to use Lagrangian indices as the heuristic of choice. This novel solution approach will be implemented in a later section and its performance compared against existing heuristic methods.

We introduce the sampling cost parameter W which represents a fee that the processor must pay per item sampled regardless of whether it is passed along for analysis or not. We also give the processor the option not to sample from any source at all during any given decision epoch. The processor can sample (or not) each source at each epoch.

We can state the objective of the problem $(P^*(C, W))$ as follows,

$$(P^*(C,W)) : \max V(W) = \max \mathbb{E}\left(\sum_{k=1}^{K}\sum_{t=1}^{T}(I_{k,t} - C)^+ - W\sum_{t=1}^{T}\sum_{k=1}^{K} S_{k,t}\right) + WT \tag{5.19}$$

where $S_{k,t} = \begin{cases} 1 & \text{if source k is sampled at time t} \\ 0 & \text{otherwise,} \end{cases}$

It is also noted at this point that the value of C is regarded as a constant of the problem, obtained in some pre-processing phase similar to that in the problem $P^*(C)$. However, the value of the sampling fee W is one which we are free to tune to obtain source selection index solutions to $(P^*(C, W))$. The first stage in doing this is to notice that $V(W)$ can be decomposed sourcewise as

$$V(W) = \max \sum_{k=1}^{K} V_k(W) + WT \tag{5.20}$$

where

$$V_k(W) := \max \mathbb{E}\left(\sum_{t=1}^{T} (I_{k,t} - C)^+ - W \sum_{t=1}^{T} S_{k,t} \right) \tag{5.21}$$

. The value of $V_k(W)$ can be developed analytically via a recursive definition if we also introduce the subscript t to denote the number of sampling decisions remaining. We have

$$V_{k,t}(\alpha_{k,t}, W) = \max \Bigg\{ \sum_{i \geq C} (i - C) \frac{\alpha_{i,k,t}}{\sum_j \alpha_{j,k,t}} - W + \sum_{i=0}^{N} P(I_{k,t} = i) V_{k,t-1}(\alpha_{k,t} + 1^i, W); V_{k,0}(\alpha_{k,0}, W) \Bigg\} \tag{5.22}$$

where

$$V_{k,0}(\alpha_{k,0}, W) = 0. \tag{5.23}$$

The two actions in (5.22) respectively refer to the actions of sampling an item and choosing not to sample an item from the source k. The rightmost term of (5.22) indicates that once the processor decides to stop sampling from a particular source, she never samples from that source for the remainder of the horizon. It is easy to show that this is a feature of any optimal policy.

Theorem 5.2.1. *$V_{k,t}(\alpha_{k,t}, W)$ is increasing in the value of t under the condition that $\alpha_{k,t} = \alpha_k$ for all t.*

Proof. Proceed by induction. The basis case is trivial as

$$V_{k,t}(\alpha_k, W) \geq V_{k,0}(\alpha_k, W)$$

for $t > 0$. Continuing to the induction step we assume that $V_{k,t}(\alpha_k, W) \geq V_{k,t-1}(\alpha_k, W)$. We have that

$$\begin{aligned} V_{k,t+1}(\alpha_k, W) = \max \Bigg\{ & \sum_{i \geq C} (i - C) \frac{\alpha_{i,k}}{\sum_j \alpha_{j,k}} - W \\ & + \sum_{i=0}^{N} P(I_{k,t+1} = i) V_{k,t}(\alpha_k + 1^i, W), V_{k,0}(\alpha_k, W) \Bigg\} \\ & \geq \max \Bigg\{ \sum_{i \geq C} (i - C) \frac{\alpha_{i,k}}{\sum_j \alpha_{j,k}} - W \\ & + \sum_{i=0}^{N} P(I_{k,t} = i) V_{k,t-1}(\alpha_k + 1^i, W), V_{k,0}(\alpha_k, W) \Bigg\} \\ & = V_{k,t}(\alpha_k, W) \end{aligned}$$

hence the result. □

Theorem 5.2.2. *$V_{k,t}(\alpha_{k,t}, W)$ is non-increasing in the value of W.*

This result is obvious from (5.21).

The decision whether to allocate the item or not is made for the processor implicitly by comparing the importance of the sampled item to the value of C. For ease of this formulation we will not allow the processor to randomly accept items of importance equal $C-1$ and suppose that this concept can be reintroduced in future work done on this problem.

For any given state $\alpha_{k,t}$, the source k is indexable if we can compute an indifference charge, $W_{k,t}(\alpha_{k,t})$ such that sampling from source k with t periods remaining is only optimal if any actual sampling charge W is less than $W_{k,t}(\alpha_{k,t})$ so that it has the following relationship with $\mathrm{V}_{k,t}(\alpha_{k,t}, W)$.

$$V_{k,t}(\alpha_{k,t}, W) = 0 \Leftrightarrow W \geq W_{k,t}(\alpha_{k,t}) \tag{5.24}$$

or alternatively

$$W_{k,t}(\alpha_{k,t}) = \inf\{W : V_{k,t}(\alpha_{k,t}, W) = 0\}. \tag{5.25}$$

If all sources are indexable in this fashion then in the state $\alpha_{k,t}$ the processor can rank them by their indifference charges, opting to sample the source k with the greatest value of $W_k(\alpha_{k,t})$. In this way the problem $P^*(C, W)$ is solved by a policy which samples all sources whose index exceeds W at all times.

Implementation of Lagrangian indices in the MABA problem

Using the standard dynamic programming framework of [Bellman, 2003], as well as the calibration approaches put forward in [Nio-Mora, 2011], [Jacko and Villar, 2012] and [Berry and Fristedt, 1985] one can compute approximate solutions to the DP problem implied by (5.22). To evaluate a given value of $W_{k,t}(\alpha_{k,t})$, one must evaluate $V_{k,t}(\alpha_{k,t}, W)$ over a grid of W values. The size of this grid should be the result of a trade off between accuracy and timeliness of computation.

Since this is a finite horizon problem, one can use the backwards recursion property in (5.22) to compute these quantities by stepping backwards in time from the boundary case (5.23). One also requires a sufficiently populated grid of the state space α for each time point, which is potentially very expensive in terms of memory allocation and computational effort. It is possible however to reduce the overall cost of these calculations by creating a permanent library of $W_{k,t}(\alpha_{k,t})$ values so that they do not need to be calculated in an online fashion for repeated runs in numerical experiments.

It is also possible to reduce the size of the state space considered by observing that for a given value of C, the information considering the posterior probabilities of sampling items with importances less than or equal to C can be consolidated into a single entity in any given vector $\alpha_{k,t}$, which can potentially reduce the dimensionality of the problem significantly depending on the nature of the individual problem. Whichever level of precision and/or state space reduction is chosen, one must create an array of $V_{k,t}$ and $W_{k,t}$ values that span the state space to use as a library for numerical experiments on the given problem. This is done as follows.

1. Decide on the size of the grid of W values to use in the calibration as well

as the subset of the state space that one wishes to use for the arrays of $V_{k,t}$ and $W_{k,t}$ values.

2. Create arrays which span the state space $\alpha_{k,t}$ to respectively store values of $V_{k,t}(\alpha_{k,t}, W)$ and $W_{k,t}(\alpha_{k,t})$ for each of the K sources (and also for each element of the W-grid in the case of the $V_{k,t}$).

3. Compute the value of $V_{k,1}(\alpha_{k,1}, W)$ where $t = 1$, for each k and value of W.

4. For each source k, for each state in the state space where $t = 1$, find the two consecutive values $W_1 < W_2$ for which the interval $[V_{k,1}(\alpha_{k,1}, W_1), V_{k,1}(\alpha_{k,1}, W_2)]$ contains zero. Store the mean of W_1 and W_2 as the value of $W_{k,1}(\alpha_{k,1})$ for that part of the state space.

5. One can now step backwards in time for $t = 2...T$ to compute the remaining values of $V_{k,t}(\alpha_{k,t}, W)$ using (5.22) and previously calculated values at lower values of t. With t time periods remaining one only needs to cover the state space for which the total number of samples made from a source is less than or equal to $T - t$.

6. One can now compute the corresponding values of $W_{k,t}(\alpha_{k,t})$ using the same method as in step 4.

At this point one should have populated arrays of value function and fair charge values which span the chosen subset of the state space, the chosen grid of W values, all time points in the horizon and all sources. From this point one could choose to solve the problem $P^*(C, W)$ by simulating forward in time from the chosen initial conditions for the horizon by following the following procedure.

1. Set an actual sampling charge W_{actual} for the problem.

2. With T time periods to go, retrieve the approprate $W_{k,T}(\alpha_{k,t})$ for each source.

3. For each source k such that $W_{k,T}(\alpha_{k,t}) > W_{actual}$, in descending order of the value of $W_{k,T}(\alpha_{k,t})$, sample an item from that source and accept it if the importance of the sampled item is greater than C.

4. Update the item importance distributions for sources sampled from, also recording the total number of samples and allocations made.

5. Repeat this process for $t = T-1, ..., 1$ using the posterior state of the system from earlier time points.

One is then able to return the number of allocations made, which are not subject to the allocation constraints of $P^*(C, W)$ in this version of the implementation, and repeat the entire process for as many runs is desired by the experimenter.

However, we are actually interested in solving the original problem P which only permits the processor to sample exactly one item per decision epoch so the solution to $P^*(C, W)$ is not what we are truly pursuing. Rather the processor should instead use the $W_{k,t}(\alpha_{k,t})$ values as indices to rank the competing sources, selecting the source k with the greatest value of $W_{k,t}(\alpha_{k,t})$ for the sampling decision at time t. This provides an approximation to $P^*(C)$ which in turn is an approximation to P^*. Finally, by restricting the total number of allocations to be exactly the number desired by the analyst, we can approximate a solution to P. This is achieved by preventing extraneous allocations from being made and by forcing allocations to be made at the end of the problem horizon if there would otherwise be a shortfall.

In this method we must also force sub-C items to be passed if it is necessary in order to allocate $\lfloor T(1 - Q_h) \rfloor$ items within T decisions and restrict the allocation of any number of items in excess of this maximum figure.

The speed at which horizons can be simulated in this implementation is very fast once the $W_{k,t}(\alpha_{k,t})$ values have been computed. Obtaining the W values themselves is the most time consuming aspect of the implementation but in certain applications the processor would be able to precompute likely scenarios for added time efficiency.

5.3 Heuristic approaches to the solution of $P^*(C)$

Evaluating the function V_t defined in Section 4.1 is generally an intractable problem so we've appealed to heuristic approaches such as the Lagrangian heuristic from the previous section. This section will adapt a selection of existing heuristic archetypes as solutions to $P^*(C)$. The Thompson sampling and optimistic Bayesian sampling methods use simulated source samples to drive decisions through the horizon. The knowledge gradient method asks the processor to take quasi-myopic decisions under an assumption that there is only one more time period during which learning can take place. It was established in section 5.1 (in the discussion following Lemma 5.1.1) that it should be possible to find an optimal threshold C^*, such that $m^*(C) = \lfloor T(1-q_h) \rfloor$. We also discuss how this is done. Additionally, a super-optimal perfect information policy is developed as an upper bound benchmark against which other heuristics can be measured in numerical studies. Approaches to inferring solutions to P from those for P^* and $P^*(C)$ are described in this section. We also discuss the possibility of varying the threshold C throughout the horizon to respond to feedback from the sources as time advances in a threshold selection policy which we call the 'dynamic C' approach.

5.3.1 Knowledge gradient approach

The Knowledge Gradient (KG) method was developed by [Gupta and Miescke, 1984] and further analysed by [Frazier et al., 2008] and [Ryzhov et al., 2012] as a heuristic approach to sequential learning problems. The typical objective is to effectively experiment with available alternatives in order to quickly learn which of them is the true best option (ranking and selection) as well as to maximise the total rewards earned in the process of doing so. It captures the trade-off between exploration and exploitation in online learning problems in a computationally tractable way.

Under KG, one assumes at every decision stage that the current decision is the last opportunity for which it is possible for learning to take place, after which the

most attractive of the alternatives will be chosen for the remainder of the problem horizon. The alternative which maximises the total expected rewards under these assumptions is chosen at each stage in a rolling fashion.

There are conditions given in [Frazier et al., 2008] which guarantee that the KG policy asymptotically converges to choose the true best alternative (the one with the highest mean) over an infinite horizon. It is worth noting that in $P^*(C)$ the best alternative refers to the source with the highest associated value of $\sum_{i<C}(i - C)p_{i,k}$. The paper also proves that the policy is one-step optimal as it is identical to a greedy approach in that special case. For the objective in our application, which is the gathering of a collection of intelligence items which together have the greatest total importance, the KG method's asymptotic tendency to converge on the best alternative is attractive. It is not generally known whether this heuristic is appropriate in the case of finite horizon online problems.

In our problem we are faced with K intelligence sources and it is assumed that each of them is capable of generating intelligence items independently with an unknown distribution of level of importance specific to that source.

Since KG methodology guarantees asymptotic convergence (as $t \to \infty$) to the most preferable, i.e. greatest, of these importance levels, one can be confident that as sources are chosen sequentially, the sources that yield the more important items are more likely to be chosen as we learn more about the sources. Therefore, the items of high importance are more likely to be included in the final item collection.

The one-step optimal attribute of the KG policy also gives weight to immediate rewards. So, as one moves through the decision horizon one would expect that exploration of the available sources is balanced against the exploitation of those that are believed to produce intelligence items of great importance. The numerical work in this section will test the adequacy of KG methods for this problem.

The knowledge gradient (KG) approach to producing solutions to $P^*(C)$ is to take decisions on the basis that the current epoch is the only opportunity to learn, beyond which one would be forced to employ a purely greedy policy and select

the source with the greatest expected one step reward $r\left(\alpha_{k,t},C\right)$ for the remainder of the problem horizon. Since no more learning takes place, the resulting total reward earned this way for source k with t time periods remaining is also the KG index for that source at that time.

We construct the KG index for a particular source k at time t by taking the sum of the current immediate expected reward $r\left(\alpha_{k,t},C\right)$ and adding it to a term which quantifies the total expected future reward earned from a greedy policy given that the current decision epoch is the only time the processor can learn. Since no further learning takes place, the processor will choose the same source and earn the same expected reward from the source which is greedy-optimal after the current decision has been made. By conditioning on the importance of the next item to be sampled, one can estimate the per period reward for all future decisions and multiply this by $(t-1)$, the number of future decisions to be made in the time horizon. At $t=1$ the KG policy is identical to the greedy policy. We have the KG index $KG(k,t)$ for source k at time t:

$$KG(k,t) := r\left(\alpha_{k,t},C\right) + (t-1)\sum_{i=1}^{N} P(I_{k,t}=i)\max\left(\max_{j\neq k} r\left(\alpha_{j,t},C\right); r\left(\alpha_{k,t}+1^{i},C\right)\right), \quad (5.26)$$

so the KG policy always directs the processor to choose the source with the greatest value of $KG(k,t)$ at each sampling decision. The processor then updates the posterior belief state for the selected source and the KG index is reapplied in a rolling fashion for each time step. Learning continues to take place as a result of all decisions.

Policy sensitivity to the degree of emphasis on exploration can be investigated by placing limits on the maximum size of the linear scaling term $(t-1)$ in (5.26) by selecting a cap, J using the modified index

$$KG(k,t,J) := r\left(\alpha_{k,t},C\right) + \min(J,(t-1))\sum_{i=1}^{N} P(I_{k,t}=i)\max\left(\max_{j\neq k} r\left(\alpha_{j,t},C\right); r\left(\alpha_k+1^i,C\right)\right), \tag{5.27}$$

and adjusting J to control the weight that expected future rewards has on the KG index. A special case of KG, which we call the greedy policy, can be created by setting $J = 0$. The greedy policy only uses the immediate expected rewards of the sources to choose from which source to sample. The computation of these indices is relatively straightforward and can be done as set out in (5.27). In [**?**], curtailing the effect of the remaining horizon $(t-1)$ in the exploration term proved useful.

The explicit algorithm for implementing a study of this type is shown below.

1. Specify the client's prior array α_T and the desired horizon length T. Set the total reward earned and the total number of items passed to be equal to zero. Decide on a value of C to use.

2. Compute the indices $KG_{k,t}$ as set in (5.27) for each of the K sources.

3. From the source with greatest such $KG_{k,t}$, observe the importance score $I_{k,t}$ for the sampled item where the distribution of $I_{k,t}$ is as in (5.6).

4. Add $(I_{k,t}-C)^+$ to the total reward earned for the horizon and add one to the number of items passed if this value is non-zero.

5. Update the Dirichlet posterior such that $\alpha_{k,t-1} \leftarrow \alpha_{k,t} + 1^i$ where k is the source that was sampled for this time period.

6. Reduce t, the number of time periods remaining, by 1.

7. If $t > 0$, Go to step 2.

8. Record the total reward earned and total number of items passed.

These steps will compute and store the result of one realisation of the system over the complete time horizon of length T. Repeating this a desired number of times will form a study allowing us to compute the mean and standard deviation for the total reward earned and proportion of items that are passed.

5.3.2 Thompson sampling and optimistic Bayes sampling

In this subsection we adapt the Thompson sampling (TS) and optimistic Bayes sampling (OBS) approaches to the discrete MABA setting.

We start with TS. At each time point t and for each source k, the processor uses the posterior distribution of p_k to make a random source k importance draw from among the integers from 1 to N. More formally, at each time point t the Thompson sampling index $TS_{k,t}$ assigned to source k is randomly sampled from the posterior importance distribution of source k. We have

$$P(TS_{k,t} = i) = \frac{\alpha_{i,k,t}}{\sum_{j=1}^{N} \alpha_{j,k,t}} \tag{5.28}$$

. Having observed the Thompson sampling indices $TS_{k,t}, 1 \leq k \leq K$, the policy for $P^*(C)$ is then to choose the source with the largest index, deciding uniformly randomly between the candidate sources in case of ties, and then update $\alpha_{i,k,t-1} = \alpha_{i,k,t} + c_{i,k,t}$.

For OBS, as with the TS approach, the processor makes a random integer draw for each source based on its posterior item importance distribution. However in the OBS approach, this random integer is compared to the expected one step mean return for that source and the greater of the two values acts as the index for that source. We have

$$OBS_{k,t} = \max\left(TS_{k,t}, \sum_{i=1}^{N}\left(\frac{i\alpha_{i,k,t}}{\sum_{j=1}^{N} \alpha_{j,k,t}}\right)\right) \tag{5.29}$$

and the policy for $P^*(C)$ selects the source k with the greatest value of $OBS_{k,t}$.

The effect of adjusting the random indices in this way is to only choose greedy

suboptimal sources if they outperform the greedy optimal choice in a random draw. One would expect the OBS policy to compete well against the standard greedy policy as the OBS introduces a much needed element of exploration which is absent in the greedy policy.

The implentation of either the Thompson or Optimistic Bayes sampling heuristics within the $P^*(C)$ framework is almost identical to that of the implementation of the knowledge gradient heuristic and its capped variants, which was defined at the end of the previous subsection. One simply needs to replace the computation and ranking of the $KG_{k,t}$ indices in steps 2 and 3 with that of the $TS_{k,t}$ or $OBS_{k,t}$ as appropriate.

5.3.3 Perfect information policy

Evaluating and comparing the performance of candidate source selection policies would ideally include comparisons to an optimal policy. In this context, comparisons to optimal are not available as we are not able to compute the optimal policy for problems of realistic size. A superoptimal policy is a plausible alternative.

The superoptimal policy used in this study is referred to as the perfect information (PI) policy. The premise is that the processor is clairvoyant and knows the true nature of the item importance distributions of all sources. It is effectively a greedy policy where no learning is necessary.

For the other candidate policies, computing the mean Bayes return requires both an inner and outer simulation as the true item importance distributions are unknown to the processor. In the case of PI, after the random multinomial vectors α_k have been generated for each of the K sources in each run of the simulation, the inner simulation does not need to take place as one can analytically compute the expected total value of the items passed, given knowledge of the true item importance distributions.

The inner expectation of the mean Bayes return

$$\mathbb{E}_{\alpha_1,\ldots,\alpha_K}\left\{\mathbb{E}\left\{\sum_{k=1}^{K}\sum_{t=1}^{T}(I_{k,t}-C)^{+}X_{k,t}|p\right\}\right\} \tag{5.30}$$

in this study is computed as the total of the importance excesses over C of all items passed to the analyst. The inner expectation presumes a given p to determine sampling outcomes and the outer expectation concerns the prior sampling of p from the prior distribution with parameters α.

Once the prior sampling of p is conducted, a clairvoyant will see what it truly is and will maximise her return by sampling from the source with the highest one-time return at all epochs and also by allocating to the analyst accordingly. Hence the clairvoyant's return for $P^*(C)$ is written

$$\max\mathbb{E}_{\alpha_1,\ldots,\alpha_K}\left\{\mathbb{E}\left\{\sum_{k=1}^{K}\sum_{t=1}^{T}(I_{k,t}-C)^{+}X_{k,t}|p\right\}\right\} = T\mathbb{E}_{\alpha}\left\{\max_{k}\left(\sum_{i>C}(i-C)p_{i,k}\right)\right\} \tag{5.31}$$

where C is tuned via some surrogate policy (e.g. KG) so that it is appropriate for the target problem. This may be an issue for the PI policy if the KG policy performs poorly but the benefit of this is that is makes the computations a lot simpler. If KG is found to be an inappropriate tuning policy then the best performing heuristic policy should be a good substitute for KG here. In this document, KG is always chosen for this tuning of C.

5.4 Application of heuristics to solve P

Now that we have specified some source selection heuristics to use, we now seek to set out how the processor should allocate sampled items in order to provide solutions for the problem P. This amounts to selecting an appropriate allocation threshold C to apply to the stream of observed intelligence items.

In this section we describe two methods for threshold selection. The first is the'static C' method, which makes a one time calibration for the choice of C before processor samples any items. The second is the 'dynamic C' method, where the

processor makes adjustments to her preferred threshold value as she samples more items.

5.4.1 Static C Method

Tuning C to solve P^*

The problem $P^*(C)$ arose as a Lagrangian relaxation of P^*. Policies for $P^*(C)$ allocate items with an importance score at least C and are easy to implement. If the value of C is chosen appropriately, one can approximate a solution to P^*.

There are effectively only $N+1$ choices for C due to the discrete nature of importance scores in this model. One needs only to find the two consecutive values of C such that the expected number of allocations ($\lfloor T(1-q_h) \rfloor$) allowed by the constraint (5.14) of P^* lies between them. We use the notation $m^*(C)$ to denote the expected number of allocations made when using the optimal policy for the problem $P^*(C)$. One can obtain values for $m^*(C)$ numerically by simulating many horizons using C and tracking the mean number of sampled items that are allocated. In practice, a strongly performing heuristic for $P^*(C)$ may need to replace the optimal policy when estimating $m^*(C)$.

Since C is discrete, taking values in integers only, there may be no C for which $m^*(C) = \lfloor T(1-q_h) \rfloor$. We secure the equality we need by randomising between consecutive thresholds as follows. If $m^*(C) < \lfloor T(1-q_h) \rfloor \leq m^*(C-1)$ (recall from Chapter 4 that $m^*(C)$ is nonincreasing in C) then for the probability that the processor uses the threshold $C-1$ we have

$$\pi = \frac{m^*(C) - \lfloor T(1-q_h) \rfloor}{m^*(C) - m^*(C-1)}, \tag{5.32}$$

and the probability that the processor applies the threshold value of C to sampled items during allocation decisions is $1-\pi$. The intention is that the mean number of allocated items when using this hybrid policy is approximately that allowed by the constraint of the problem P^*.

Using a randomised policy for choosing C means that it will simplify and clarify matters for the processor to record absolute importance scores for the allocated items rather than exceedes over C. Hence we amend the objective to maximise:

$$\mathbb{E}\left\{\sum_{k=1}^{K}\sum_{t=1}^{T} I_{k,t}X_t\right\} \tag{5.33}$$

where we restrict the set of allocation policies to those described in this subsection and the processor randomly determines the value of C for each decision epoch. The problem is to choose the appropriate values of C and π such that the mean number of items passed is approximately $\lfloor T(1-q_h)\rfloor$ so by abuse of notation we continue to refer to this problem as $P^*(C)$.

To find the values of C_1 and π to use such that the expected number of allocations made is as close to $\lfloor T(1-q_h)\rfloor$ as possible the processor proceeds in the following way for each of the policies considered:

1. Via simulation, search among the integer values of $C \in [1, N]$ for the two consecutive values for which $m^*(C) < \lfloor T(1-q_h)\rfloor \leq m^*(C-1)$.

2. Use the formula in (5.32) and the C values from step 1 to set the probability π that the threshold value $C-1$ is used for any given allocation decision.

It is also possible to perform a search among the possible values of π if steps 1 and 2 do not tend to allocate near enough to the target value of $\lfloor T(1-q_h)\rfloor$ allocations. This is more time intensive at this stage but the problem setting may make it worthwhile.

Approximation to P via P^*

With only the priors α for the item importance distributions available ahead of time, the processor must decide the nature of both the sampling and allocation policies without any access to online feedback from the true system. In absence of any available opportunity for real-world experimentation, the mean Bayes return $\mathbb{E}_\alpha\left\{\mathbb{E}\left\{\sum_{k=1}^{K}\sum_{t=1}^{T} I_{k,t}X_t|\phi\right\}\right\}$ is a measure of the expected performance of

candidate policy combinations given that the current prior for the system is α. This quantity represents the expected total allocated reward from the problem, averaged over α and in order to compute it one must compute the mean expected horizon reward $\mathbb{E}\left\{\sum_{k=1}^{K}\sum_{t=1}^{T} I_{k,t}X_t|\phi\right\}$ for a sufficient number of random vectors ϕ drawn from the prior α.

The way to generate random instances item importance distributions based on α is to generate for each of the k sources realisations of the independent random variables $Z_1 \ldots Z_N$ where $Z_i \sim Gamma(\alpha_{i,k}, 1)$ and use the fact that the vector $(\frac{Z_1}{Z} \ldots \frac{Z_N}{Z})$ where $Z = \sum_{i=1}^{N} Z_i$ is Dirichlet distributed with parameter α_k (see [Devroye, 1986]).

When randomising with respect α in the outer simulation, one generates realisations of item importance distributions for each source before the horizon simulation begins and one would refer to the pre-generated distributions whenever a sample from a source is made in order to obtain the item's importance value. One would need to repeat this procedure for each repeated horizon in order to compute the mean Bayes return for the problem.

However, it is more computationally efficient and mathematically equivalent to instead incorporate the randomisation of α into the inner simulation itself. Starting with the same priors for each horizon considered, the item importances for the sampled items can be drawn from the respective predictive distributions of the sources sampled, where the importance value of the sampled item is then used to update the posterior for that source. In numerical studies, this eliminates any need to create random draws from α for each repeated horizon before simulating what occurs during the horizon for the given draw ϕ. Instead each α is sampled implicity during each simulated horizon, which greatly reduces the computational time required to conduct a study. In the previous subsection, it was shown how the processor could produce an approximate solution to the problem P^* by tuning the value of C in the problem $P^*(C)$ such that the constraint on the expected number of allocations in problem P^* is satisified. A value of C and a probability

π of using threshold value $C-1$ can be selected via a search method such that the expected number of allocations is close to the upper limit according to the constraint imposed by the processor in problem P^*. The processor truly wishes to solve problem P where the constraint on the number of allocations is less than or equal to a given value absolutely, with no allowances for possibly allocating items to achieve a total in expectation as is the case in P^* and $P^*(C)$.

Since the processor is able to approximate a solution to P^* by tuning the value of C appropriately in $P^*(C)$, she should now be able to approximate a solution to P by forcing the number of allocations to satisfy the constraint on the number of allocations directly. This can be achieved by preventing all intelligence items from being allocated once the maximum number of allocations have been reached or forcing the allocation of items with importance less than C in order to reach the maximum allocation limit if necessary.

Under the problem P, the processor is required to allocate an exact number of sampled items by the end of the problem horizon. To force a solution to P, the processor behaves as if solving $P^*(C)$ until one of two conditions are satisfied. One condition is that the processor allocates a number of items which is equal to the required amount under P, at which point she allocates no further items, even if those items have importances greater than C. Alternatively the processor has yet to allocate the full amount of items and the number of allocations that remain is equal to the number of time periods that remain in the horizon. In this case, the processor allocates all of the remaining items sampled, even if those items have importances less than C. In this way the exact number of allocated items that is required under P is always achieved.

5.4.2 Dynamic C method

The discussion of the *Dirichlet-Multinomial* model so far has treated the threshold C as a pre-calculated constant that applies to all allocation decisions throughout the horizon. The operational motivation for why a client would choose a higher

value of C would be because of an increased scarcity level in resource (time) that can be dedicated to analysing processed items. Previously passable items may need to be rejected by an updated policy which takes into account a more scarce investigation environment. Choosing a greater value of C achieves this effect by filtering out more items.

The problem of choosing a static value of the tuning constant C is difficult and our current tuning method requires a time consuming search subroutine before it can be implemented for a given problem scenario. This situation motivates the development of methods which allow the value of C to be chosen dynamically throughout the horizon. The ability to respond dynamically to what is learned about the sources is attractive. In doing so, we attempt to solve P directly by tuning C across the horizon to ensure that exactly the desired number of allocations are made by the processor.

When solving P^* via $P^*(C)$ in the static case, the choice of C should be such that the expected number of allocations is less than or equal to $\lfloor T(1-q_h) \rfloor$. A proposed way of dynamically choosing a value of C at time t is to first compute the q_h-quantile of the posterior c.d.f. of the item importance distribution for each source k, where we define

$$C_{k,t} = \inf\{i : F_{k,t}(i) \geq q_h, 1 \leq i \leq N\}, \tag{5.34}$$

where

$$F_{k,t} = \sum_{l \leq i} \frac{\alpha_{l,k,T} + n_{l,k,t}}{\sum_j (\alpha_{j,k,T}) + n_{k,t}}, \tag{5.35}$$

and we set $C_t = \max_k C_{k,t}$ to be the tuning constant for that decision period.

We now discuss how a dynamic thresholding policy impacts the design of KG indices. Since C_t updates with incoming information about the sources, one must modify the knowledge gradient indices to accommodate this new feature of the model. The dynamic analogue for the knowledge gradient index for source k at

time t is defined as follows:

$$KG(k,t) := r\left(\alpha_{k,t},C_t\right) + (t-1)\sum_{i=1}^{N} P(I_{k,t}=i)\max\left(\max_{j\neq k} r\left(\alpha_{j,t},C_{t-1}(i)\right); r\left(\alpha_k+1^i,C_{t-1}(i)\right)\right), \tag{5.36}$$

where $C_{t-1}(i)$ is the value of the tuning constant at time $t-1$ given that an item of importance i is observed as a result of selecting source k. The processor selects sources according to this modified KG policy and only allocates the item sampled at time t if its importance is greater than C_t. Proceeding in this way yields a proposed solution to the problem P^* set out in (5.14). The reasoning behind this approach is that the processor can only allocate the proportion $1-q_h$ of the T sampled items in the problem so it holds that she must reject Tq_h items. By rejecting items below the q_h^{th} quantile of the item importance distribution we should tend to filter out the correct proportion of items. These items contribute the least reward upon allocation as they are from the leftmost part of the distribution, so the processor satisfies the expectation based allocation constraint whilst still maximising the total importance of the allocated intelligence items.

As with the static C implementation we can set a cap J, on the multipier applied to future expected rewards and create a capped version of KG as where

$$KG(k,t,J) := r\left(\alpha_{k,t},C_t\right) + \min(J,(t-1))\sum_{i=1}^{N} P(I_{k,t}=i)\max\left(\max_{j\neq k} r\left(\alpha_{j,t},C_{t-1}(i)\right); r\left(\alpha_k+1^i,C_{t-1}(i)\right)\right), \tag{5.37}$$

to tune the importance the processor places on future expected rewards.

Remark: When $F_{k,t}(C_{k,t}) = q_h$, items with importance at least $C_{k,t}$ are passed to the analyst, in a single source problem, with probability equal to $1-q_h$. Otherwise, when $F_{k,t}(C_{k,t}) > q_h$, the threshold policy passes items with probability

smaller than $1 - q_h$, meaning that the senior analyst will typically receive items at rate which deviates from the desired amount. One way to overcome this rate bias is to randomize the passing policy for items with importance $(C_{k,t} - 1)$, by passing them with probability equal to

$$\pi_{k,t} := \frac{(F_{k,t}(C_{k,t}) - q_h)}{(F_{k,t}(C_{k,t}) - F_{k,t}(C_{k,t} - 1))}. \tag{5.38}$$

To accommodate randomised acceptance in this way, we need to reformulate the model that has been presented so far. We redefine the nature of the threshold policy based on $C_{k,t}$ to incorporate $\pi_{k,t}$. We define $\pi_t = \max_k \pi_{k,t}$ and $F_t = F_{k,t} : k = \arg\max_k(C_{k,t} - \pi_{k,t})$. For the processor, the random acceptance policy based on the threshold C_t such that at time t,

- Items of importance $i \in [C_t, N]$ are always accepted.
- Items of importance $i \in [0, C_t - 1)$ are always rejected.
- Items of importance $i = C_t - 1$ are passed to the analyst with probability π_t.

The added accuracy offered by this 'π method' is sufficient that it will also be incorporated into the static C implementation. One interpolates between the two values, $m^*(C-1)$ and $m^*(C)$ (see the definition preceding Proposition 4.0.3) that contain the desired expected mean number of items, $\lfloor T(1 - q_h) \rfloor$ to be passed.

$$\pi_t = \frac{\lfloor T(1 - q_h) \rfloor - m^*(C-1)}{m^*(C) - m^*(C-1)} \tag{5.39}$$

When choosing C_t in a dynamic fashion, given that the processor knows exactly how many rewards have been allocated at time t, the number of remaining allocations that should be made at time t will not necessarily be equal to $\lfloor t(1 - q_h) \rfloor$. The processor can dynamically adjust the quantile used in the computation of the $C_{k,t}$ by taking into account the number of allocations that have already been made. For the horizon wide quantile q_h, we define the time dependent quantile q_t

for $t > 0$. We have

$$q_t = 1 - \min\left[1, \left[\frac{T(1-q_h) - \sum_{s=t+1}^{T} X_s}{t}\right]^+\right], \tag{5.40}$$

where $X_t = \begin{cases} 1 & \text{if item is allocated (passed to analyst) at time } t, \text{and} \\ 0 & \text{otherwise.} \end{cases}$

The quantity $\sum_{s=t+1}^{T} X_s$ in (5.40) denotes the total number of allocations made before the current time period. For a given value of q_t and source k there is a passing threshold $C^*_{k,t}$ defined as the (approximate) posterior q_t quantile of source k at time t:

$$C^*_{k,t} = \inf\{i : F_{k,t}(i) \geq q_t, 1 \leq i \leq N\} \tag{5.41}$$

where $C^*_t = \max_k C^*_{k,t}$.

The knowledge gradient for source k at time t is computed as in (5.36) except in the case of randomised choices we instead use

$$\pi_{k,t} := \frac{(F_{k,t}(C^*_{k,t}) - q_t)}{(F_{k,t}(C^*_{k,t}) - F_{k,t}(C^*_{k,t} - 1))}. \tag{5.42}$$

With dynamically changing thresholds it is no longer appropriate to use the objective function set out in (5.13) as the interpretation of rewards is inconsistent as the threshold value is not held constant over the horizon. Under (5.13), less important items can yield greater rewards than more important items if they clear a lower C^*_t target by a larger margin. We replace the objective in (5.13) with the alternative described in (5.43).

Maximise:

$$\mathbb{E}\left\{\sum_{k=1}^{K}\sum_{t=1}^{T} I_{k,t}X_t\right\} \tag{5.43}$$

subject to (5.14), as previously. We remove the thresholds from the objective function entirely.

We have discussed the specific adapation of the KG indices to the dynamic

setting. It is worth noting how the other source selection policies we have used in the static C environment operate under a dynamic C thresholding environment. In the cases of Thompson sampling and Optimistic Bayes sampling, the indices are generated in exactly the same way as under static C, as there neither policy is calculated with respect to the threshold C_t at any time in the horizon. Only the current posterior item importance distributions for the sources are necessary.

In the case of the perfect information policy, the analyst was able to analytically compute the inner expectation of the mean Bayes return in the static C environment using (5.31) which saved on computational effort. This is because of the clairvoyant nature of the PI policy, which reveals the true nature of the item importance distributions across all sources. In the dynamic C environment, the threshold is not a pre-calculated constant; rather, the threshold evolves throughout the time horizon. To use PI in the dynamic C setting, the processor does not benefit from the computational saving present in the static C case and instead uses her clairvoyant knowledge to choose the source with the greatest one step reward at each time t, subject to the prevailing value of C_t^*.

At this stage, we establish the usability of the dynamic C framework with some preliminary studies.

5.4.3 Preliminary studies for Dynamic C method

The purpose of this subsection is to explore the implementation of the dynamic C methodology over a series of small scale numerical studies before committing to a larger scale study.

In this subsection, two item importance distributions, spanning five sources each, are considered in the experiments. They are shown here in Tables 5.1 and 5.2 and are referred to throughout. We show the prior $\alpha_{i,k,T}$ values in these two tables for the five sources in Studies 1 and 2. We sample from these priors to generate the initial belief states $\alpha_{i,k,T}$ for the sources in each simulation run.

Importance	1-8	9	10	Mean
source 1,2	4	4	2	5.263
source 3	8	9	3	5.250
source 4,5	4	5	1	5.237

Table 5.1: Prior importance distributions for sources in Study 1

The prior importance distributions for Study 1 shown in Table 5.1 have been chosen such that the distributions collection of five sources have relatively similar means (means within 0.026 of each other) but not all identically distributed. Similar means in these experiments make for the most compelling studies as if any one source has a mean that is even moderately lower than the rest, it will tend to be ignored by all of the competing policies. The sources' item importance distributions are also designed in such a way that early observations in the problem horizon can change the posterior mean of these distributions in a meaningfully great way.

The design of the item importance distributions in Table 5.2 also follow this design philosophy of ensuring that the means of the distributions are relatively close together. The means of these sources' item importance distributions are also just below 3 and the long tails of each of these distributions are placed to start for items of importances 4 and above. The rationale for this is to create problems with interesting solutions. We learn less from problems where either a single source

attracting all of the samples in an uncontested manner, or the choice of the source is ultimately irrelevant to the total reward earned. The latter case occurs when each of the sources are so similar that there is no meaningful choice to be made. Having similar means (so that one source doesn't immediately win outright) and having diverse long tails above the mean of the item importance distributions (so that the choice between sources is meaningful) has helped us create more compelling studies. The studies in this subsection are concerned with solving the problem P.

Importance	1-3	4	5	6	7	8	9	10	Mean
source 1	93	6	3	1	7	4	2	8	2.52
source 2	78	1	1	1	1	1	1	20	2.72
source 3	84	7	6	5	4	3	2	1	2.4
source 4	75	1	1	1	1	1	10	10	2.68
source 5	21	2	2	2	2	2	2	2	2.93

Table 5.2: Prior importance distributions for sources in Study 2

We achieve this by first solving $P^*(C)$ which yields a solution to P^*. We then force a solution to P by overruling the thresholding policy to prevent excess allocations over the quota, or by ensuring the quota is met by allocating items if the number of time periods remaining is equal to the number of remaining allocations that are required to fulfil the quota.

Study 1

In this subsection the results of a numerical study which we shall refer to as Study 1 are discussed. The number of sources considered was 5, the time horizons were of length $T = 100$, the maximum importance rating of any source was $N = 10$. Each policy was tested on the same 2000 test case runs.

Four such experiments were run with different values of q_h. The four values used were 0.95, 0.9, 0.85 and 0.80. The prior beliefs for each of the 5 sources at the start of every time horizon are shown in Table 5.1 along with the prior mean item importance for each source.

The values in Table 5.1 are also the basis for randomly generating the 'true'

1-q_h	G Policy	s.e	KG Policy	s.e.	PI policy	s.e.
0.2 (20 items)	170.5	0.10	170.4	0.10	171.1	0.09
0.15 (15 items)	132.2	0.07	131.9	0.07	132.3	0.07
0.1 (10 items)	89.9	0.05	89.8	0.05	90.0	0.05
0.05 (5 items)	45.9	0.02	46.4	0.02	46.7	0.02

Table 5.3: Mean total importance of items passed

underlying distributions in individual horizons from which all item importances were drawn. The PI policy has exclusive access to knowledge of these true distributions in contrast to the other policies considered, which rely entirely on the posterior beliefs informed by the observed sampled item importances from sources. We refer to the greedy policy (G) as the variant of KG that does not take into account future rewards and focuses on immediate rewards only. The G index is obtained by setting $J = 0$ in (5.37).

In Table 5.3 we present the mean total importance of the items allocated in each of the experiments, categorised by total number of item allocations and the source selection policy used. In terms of mean total importance of items allocated, it is apparent that the performance gap between the G and KG policies is in favour of the G policy, with the exception of the $1 - q_h = 0.05$ case. It is unclear how much the choice of policy regarding source selection is a driver of this good performance and how much of it can be attributed to the dynamic threshold selection mechanism but the combination of both of these features appears to work well in this case. Further preliminary studies are required at this point to examine the effect of the problem setting on policy performance and how much the favourable 1% (approx.) value of PI over KG is problem dependent or whether it holds generally. Figures 5.1 and 5.2 were each created by running five separate experiments with the different values of T shown on the x-axes of the figures. The value of q_h used was 0.9 in these two figures. In each experiment, 2000 simulations were run. The item importance distributions are as in Experiment 1 (see Table 5.1) In Figure 5.1 we show the boxplot of the pairwise percentage gains of mean Bayes returns for the PI policy over the KG policy. For each simulation run, the mean

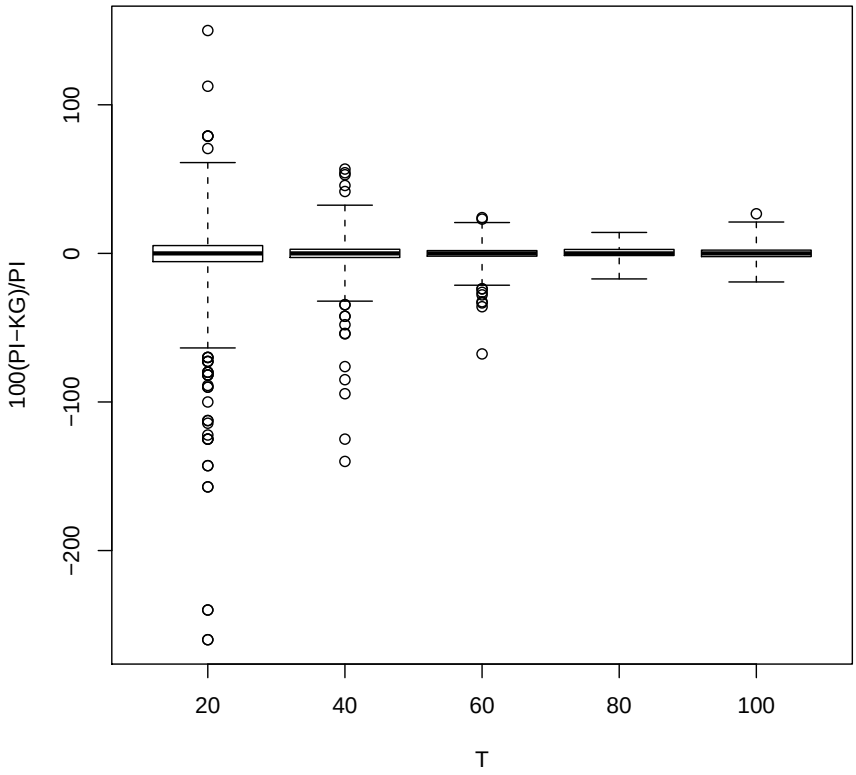

Figure 5.1: Relationship between pairwise percentage gains of mean Bayes returns (PI vs. KG) and horizon length T

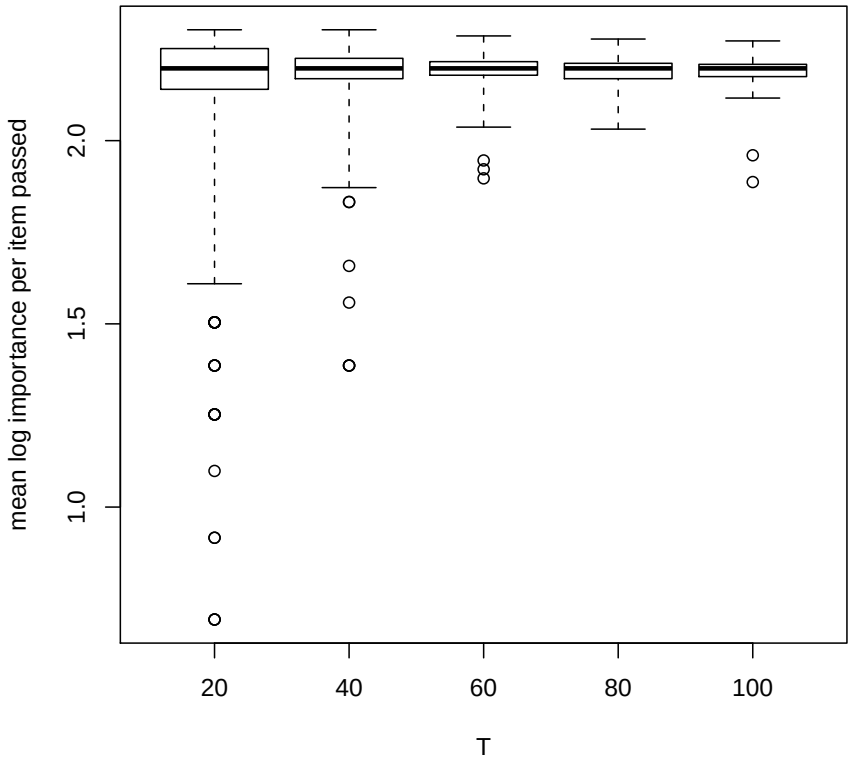

Figure 5.2: Relationship between log mean passed item importance and horizon length T (KG)

Bayes returns for the PI and KG policy are compared directly and the percentage gain of I over KG is computed according to the formula in the y-axis of Figure 5.1. In Figure 5.2 the boxplots are constructed for each experiment by taking the log mean importance of the allocated items in each of the simulation runs. In Figure 5.1, the relationship between the horizon length T and the value of perfect information is shown. The plot suggests that the role of source selection policies is greater when the horizon length is shorter, motivating analysis as to why this is the case. A natural hypothesis is that the KG policy benefits from additional time to learn about the true relative ranking of the sources as T increases or that exploration is increasingly expensive as the horizon length decreases. Before we continue it is worth mentioning that for the simulations depicted in Figures 5.1 and 5.2 the theoretical maximum log mean item importance per allocated item in

1-q_h	G	s.e	KG	s.e
0.2	30.91	0.12	25.75	0.27
0.15	41.14	0.12	27.68	0.32
0.1	37.54	0.16	28.79	0.39
0.05	4.25	0.18	18.88	0.43

Table 5.4: % Optimal source choices, Study 1

each run is $\log 10$, as the maximum total importance for all allocated items in each run is $N\lfloor T(1-q_h)\rfloor$ where we have $N = 10$ and the total number of allocations equal to $\lfloor T(1-q_h)\rfloor$.

This effect is more noticeable in Figure 5.2 where the boxplots of the log mean importance per item passed by the KG policy are plotted for the same scenarios studied in Figure 5.1. The effect of an increased time horizon steadily decreases the variability in the average value of the passed items. The mean value remains steady as the time horizon increases, largely because the log mean importance per item value for $T = 20$ is already close to the theoretical maximum and has no room to increase. This suggests that the problem setting of Study 1 is such that it is not particularly difficult for the policies to attain the highest levels of performance. A more challenging problem setting may be appropriate.

In Table 5.4 the mean percentage of the horizon that the G and KG policies agreed with the PI policy (i.e. chose the same source) is shown above. The knowledge gradient policy chooses the sources favoured by PI less often than the greedy policy preferring to explore other sources in addition to exploiting the supposed 'best source', which the G policy focuses on doing more strongly. We see in Table 5.5 that the significant difference in source selection behaviour between the PI, KG and G policies has not translated into a noticeable difference in the mean horizon wide average threshold $\frac{1}{T}\sum_{t=1}^{T} C_t^*$ which prevail between the three policies.

The lack of noticeable impact of varying source selection also applies to the mean time at which the processor fills their quota of items required to send to the analyst, except in the case of $q_h = 0.05$ where we see that the larger differences

in source selection accuracy between all three policies is reflected in the spread of these mean end times. The KG policy, which in this case is more aligned with the PI policy, has an earlier end time than the far less accurate G policy here.

Although the mean end times do increase with the value of q_h they do not increase in direct proportion to the number of items that are passed. In this example $1 - q_h = 0.05$ and $1 - q_h = 0.2$ respectively imply that 5 and 20 items are to be submitted by the processor to the analyst (a fourfold increase), yet the corresponding mean end times in the PI case are 28.26 and 66.54, which is approximately only a twofold increase. As one would expect, increasing the horizon threshold q_h results in a stricter item acceptance policy, meaning that on average a greater number of items are rejected by the processor per item that she accepts.

The mean end times in Table 5.5 are very low and show that the policies are allocating the full quota of items without using close to the full horizon's worth of T samples. This highlights a problem with the thresholding policy effective in these studies. In particular, it appears that setting the threshold selection policy sets C too low and results in the processor allocating the full quota of items too early. The dynamic adjustments to C across the horizon do not seem to restrict the stream of item allocations strongly enough and an alternative dynamic threshold selection policy may need to be developed.

This gulf in agreement with PI between the G and KG policies does not have a significant impact on the mean total importance of the items that are submitted. The sources in this problem are similar enough that it matters less which sources are chosen as could otherwise be the case. Different scenarios need to be tested to investigate the impact of scenario design on these performance measures. The discrepancy between the prior and the true item importance distributions of the sources is generally not great, so the G policy performs well when exploration isn't paramount but it may be the case the G policy's performance drops sharply in a lower information environment.

1-q_h	Policy	Mean horizon wide threshold	Mean end time	s.e
0.05	G	8.05	34.09	0.30
	KG	8.05	33.82	0.30
	PI	8.05	28.26	0.30
0.1	G	7.49	48.35	0.30
	KG	7.49	48.68	0.30
	PI	7.49	48.21	0.30
0.15	G	7.07	59.28	0.29
	KG	7.07	59.32	0.29
	PI	7.07	59.28	0.29
0.2	G	6.60	66.67	0.27
	KG	6.60	66.55	0.27
	PI	6.60	66.54	0.27

Table 5.5: Mean thresholds, finishing times and rewards in Study 1. All s.e. for Mean Threshold were less than 10^{-4}

This particular numerical study examined one test case of source selection and investigated the sensitivity of policy performance to some key problem attributes. In the cases of both G and KG policies it has been shown numerically that the value of perfect information in this particular case is small, and that the G policy seems to do better than the KG in all but the $1 - q_h = 0.05$ case.

The choice of horizon length T and the horizon quantile, q_h has been shown to have an effect on the performance of source selection policies in particular and motivates further investigation into those relationships so that more challenging scenarios can be created to test source selection policies in particular.

Studies 2 and 3

The impact of making accurate source selections in Study 1 (i.e. selecting the same sources as the PI policy) on the overall performance of the processor was unclear. It did not appear to make a significant difference. The prior beliefs for the sources in Study 1, from which the true distributions were constructed, were possibly too similar for studies of that problem to reveal interesting properties about source selection policy so the problem set up for Study 2 was created.

Policy	Mean threshold	s.e	Mean end time	s.e	Mean reward	s.e
G	2.65	0.01	27.42	0.15	48.52	0.20
KG	2.64	0.01	29.75	0.16	48.62	0.19
PI	2.68	0.01	28.26	0.15	49.06	0.19

Table 5.6: Mean thresholds, finishing times and rewards in Study 2

A revised set of priors for a five source problem is shown in Table 5.2. The idea behind the design on this problem is that the 90% quantile of each of these distributions is 3. With a q_h value of 0.9 this allows more room for variation in the shape and scale of the tails of the item importance distributions, which makes them more distinct than the sources in Study 1. The number of sources considered remains 5, the time horizons are of length $T = 100$, the maximum importance rating of any source is $N = 10$, and 2000 simulations were run for each policy.

The new problem setting of Study 2 did reveal an undesirable property of the dynamic threshold selection policy employed so far. Table 5.6 shows that the average end time for all policies is suspiciously low. Given that $T = 100$, one would expect the processor to make the most of her available sampling opportunities in order to seek out items of greater importance. She does not do so, because the anchoring of the threshold C_t^* to the q_t quantile of posterior distribution of the sources results in the processor being content with accepting any reward above the q_t quantile of the distribution, which in this study will be items typically with minimum importance of 3 or 4.

A policy which readily accepts items of a lower importance quickly exhausts the budget of items the processor can submit for analysis and in doing does not provide the best value for the analyst as the search for items of the highest importance are not prioritised. This policy only strives to provide items from the top $1 - q_t$ of the perceived item importance distribution rather than the top $1 - q_t$ of items seen in the horizon.

A new method for selecting C_t^* values based on q_t values is required to amend

this behaviour. One such way of achieving this is to divorce threshold selection from the beliefs about the sources' item importance distributions. The amended dynamic C policy at time t requires the value of q_t as defined in (5.40) and we define

$$C_t = \lceil Nq_t \rceil \tag{5.44}$$

$$\pi_t = \lceil Nq_t \rceil - Nq_t, \tag{5.45}$$

where N is the maximum importance score for any item.

The effect of doing this is that the evolving C_t values are very strict initially and reject all items which fall below the importance level equal to $\lceil Nq_h \rceil$ but this strictness wanes as the processor continues to reject items and has fewer remaining time periods during which she must fill her quota of item allocations. This approach directly ties the threshold level C_t to the processor's level of urgency with regards to fulfilling her item allocation quota at time t. She accepts fewer lower quality items when she has more time remaining and has fewer items to submit. She becomes less strict as time elapses, and if she has allocated fewer items by time t. This is in direct contrast to the previously trialled method which ties C_t to the current posterior item importance distributions of the sources, and did not use the full length of the horizon to search for the best quality items. This new approach is designed with the intention that the processor does not end her search for items prematurely.

Study 3 replaces the original source based threshold policy implemented in Study 2 with the new policy desribed by (5.44) and (5.45), holding all other parameters equal. The observed improvement which this new threshold policy creates can be seen to be dramatic in Table 5.7. The mean sum total of the importances of submitted items is significantly greater in Study 3 than it is in Study 2. The mean threshold across the horizons in Study 3 are almost double those in Study 2 and the processors of Study 3 tend to use the entire horizon searching for high

Policy	Mean threshold	s.e	Mean end time	s.e	Mean reward	s.e
G	7.375	0.013	95.89	0.23	83.17	0.34
KG	7.371	0.012	96.25	0.23	81.57	0.32
PI	7.375	0.013	95.83	0.23	83.21	0.34

Table 5.7: Mean thresholds, finishing times and rewards in Study 3

Study	G	s.e	KG	s.e
2	42.34	0.44	32.58	0.64
3	99.97	0.00	81.62	0.82

Table 5.8: % Optimal source choices, Studies 2 and 3

quality items, where Study 2 failed to do so. The differences in source selection policy between G, KG and PI still do not seem to impact nearly as much as the change in the threshold selection policy clearly has.

It is a surprising result indeed to see that a policy based on such a simple heuristic vastly outperforms a policy which makes sophisticated use of the posterior information state of the sources. It has not been shown that the superiority of this heuristic holds generally but it is more a damning indictment of the so-called sophisticated policy's competence. The problem of dynamically and efficiently setting a threshold for C has proven to be a non-trivial task.

The source selection behaviour for the G policy largely coincides with that of the PI policy, which causes its reward performance to marginally outperform the KG policy. This is because the KG policy still tends to explore other sources more often as it deviates from the sources with the greatest immediate expected rewards. This is despite KG assigning the majority of its sampling effort to the more exploitable sources on average. It is possible that over-weighting of potential future gains in the KG policy is encouraging too much explorative behaviour which is inappropriate for this setting where rewards are thresholded in such a strict way. The cost of that level of exploration may just be too great in these kinds of problems.

The percentage of source selection decisions made by the G and KG policies

which coincided with the PI policy in Studies 2-3 are shown in Table 5.8 . The G policy was very much aligned (almost identically) with the PI policy in the setting operating in Study 3, agreeing in almost 100% of cases with the PI policy and achieving an essentially identical performance. We find that the KG policy agrees with PI less often as it is designed to explore the sources and deviate from the greedy-optimal source, which in Study 3 also happens to be the PI-optimal source.

These preliminary studies suggest that the key driver of good performance is the adequacy of the policy for accepting and rejecting sampled items that are sampled by the processor more than it is the policy for selecting and learning about the sources from which to sample, despite the value of accurate information about sources being high. It may be the case that relatively short time horizons in these problems do not grant the processor enough flexibility to experiment and learn about sources enough for it to make a large impact on her ability to seek out the most important items that can be found.

In these preliminary studies it is clear that the time horizon T, the horizon quantile, q_h, the source selection policy, the item acceptance policy, and the state of the prior beliefs relative to the true facts all have an impact on the processor's ability to gather high quality items for the analyst. By running tests on various combinations of these conditions the goal is to construct a sufficiently detailed picture so that meaningful insights can be gained into the problem.

5.5 Lagrangian indices: Numerical study

Several heuristics for solving the problem P have been identified in this chapter so far. In the remainder of this chapter, numerical implementations of a multitude of these approaches will be documented. The results of such studies will serve to establish whether these approaches are workable at all and help to identify which heuristics tend to be the best fit for solving P overall by testing them out in a wide array of problem settings.

To test the performance of the Lagrangian charges $W_{k,t}(\alpha_{k,t})$ as an index heuristic for solving problem P, the heuristic was tested on the five source scenario set out in Tables 5.9 and 5.10 with $q_h = 0.95$ and $T = 100$. Hence, the allocation limit is 5 items out of 100.

During the time that the authors have looked at this problem, it has been difficult to devise an implementation of the Lagrangian index heuristic approach that is tractable when applied to moderately large state spaces. It has only been possible for us to apply the Lagrangian index heuristic to problems with a binary state space. In other words, when computing indices, we can only describe items' importance as either being less than C or greater than or equal to C, with no level of granularity in between.

This particular setup has the convenient property that the resulting integer value of C that is most appropriate threshold for this problem is 9 when a value of C is searched for in the problem $P^*(C)$. One can reduce the complexity of the size of the state space considerably by consolidating the parts of the item importance distributions corresponding to importances 1 through 9 into a single category for rewards which are not strictly greater than C. This allows us to evaluate the performance of the Lagrangian indices, even if we cannot consider a wide range of problem settings.

Once the array of $W_{k,t}(\alpha_{k,t})$ was generated for both scenarios, 10000 horizons were run for both studies and the mean Bayes return for the Lagrangian indices are shown in Tables 5.11 and 5.12 as well as those earned by rival policies in the

Importance	1-9	10
source 1,2	36	2
source 3	36.5	1.5
source 4,5	37	1

Table 5.9: Prior importance distributions for sources in Lagrangian studies for Experiment 1

Importance	1-9	10
source 1,2	9	10
source 3	9	1
source 4,5	18	1

Table 5.10: Prior importance distributions for sources in Lagrangian studies for Experiment 2

same experiment setup. In these studies we approximate a solution to P by forcing the total number of allocations in any study to be exactly $T\lfloor(1-q_h)\rfloor$ as described in section 5.4.1. This is achieved in one of two ways in each problem horizon. Under application of a heuristic to the problem P^*, the processor either reaches the required quota of item allocations for P before the horizon ends, in which case she allocates no further items in that horizon. The other case is that the processor does not reach the allocation quota early and has a remaining number of allocations to make which is equal to the number of remaining time periods. In this second case, the processor allocates all remaining sampled items. In this way, the exact number of required allocations are made in every instance of the problem.

We now look at Tables 5.11 and 5.12 and comment on the performance of the various source selection policies in Experiments 1 and 2.

The mean Bayes returns in Experiment 2 make it difficult to differentiate between the performance of the Lagrangian, optimistic Bayes, Thompson, and knowledge gradient policies (as well as PI), as all of these policies are able to achieve approximately the highest possible rewards in this scenario. Sources 1 and 2 in Experiment 2 tend to be the clearly preferred choices over the remaining three. Only

Policy	Mean Bayes return ($T = 100$)	s.e.
Greedy	34.72	0.09
Knowledge Gradient	39.11	0.09
Thompson	41.81	0.09
Optimistic Bayes	42.46	0.08
KG Capped at 2	34.42	0.09
KG Capped at 5	34.42	0.09
KG Capped at 10	37.70	0.09
Lagrangian	43.91	0.1
PI	44.45	0.09

Table 5.11: Bayes returns for Lagrangian indices compared to rival heuristics (Experiment 1)

Policy	Mean Bayes return ($T = 100$)	s.e.
Greedy	36.44	0.13
Knowledge Gradient	49.82	0.01
Thompson	49.32	0.03
Optimistic Bayes	50.00	0.01
KG Capped at 2	36.39	0.14
KG Capped at 5	36.35	0.14
KG Capped at 10	36.51	0.14
Lagrangian	49.64	0.10
PI	50.00	0.11

Table 5.12: Bayes returns for Lagrangian indices compared to rival heuristics (Experiment 2)

the greedy policy and the greedy-like capped KG policies are noticeably weaker. These policies fail to explore the sources sufficiently enough to consistently sample from the best sources in each run of the simulation. The cost of exploration in this kind of problem is worth paying.

Experiment 1 shows more clearly the differences between the Lagrangian, optimistic Bayes, Thompson and KG policies. This is most likely because all five sources in this problem have priors which are similar, as opposed to Experiment 2 where the differences between the sources are more exaggerated. The Lagrangian index policy suffers the least from the more challenging problem whereas the knowledge gradient policy falls further behind the Lagrangian index policy in Experiment 1. Thompson and Optimistic Bayes perform better than KG, but do not keep up with the Lagrangian policy. The Lagrangian policy comes equipped with indices that have been designed to consider the most detailed picture of the future state of the problem at any given time, and we see this reflected in its performance here.

In Experiment 1 we observe again the inadequacy of the greedy policy and capped versions of KG. Policy performance in this problem hinges greatly on the ability of each policy to consider possible future states of the problem when making sampling decisions in the present. The Lagrangian indices perform so well precisely because they extensively consider future states of the problem by design. Knowledge gradient allows a second best source to be explored in the case where it would be efficient to do so. This approach seems to be less good than sampling from the posteriors as is the case with optimistic Bayes and Thompson sampling, which allow the full range of sources to be potentially explored. It is not clear why Optimistic Bayes' tendency to forgive otherwise attractive sources for their poor draws results in a superior performance over Thompson sampling in both experiments. The small edge that Optimistic Bayes has over Thompson sampling may be specific to item importance distributions of these problems. Further studies would be required to find out whether Thompson sampling consistently performs less well across a broader range of problems and also to discern why this is the

10 \1-9	0	1	2	3	4	5
0	0.0525	0.0515	0.0515	0.0505	0.0505	0.0495
1	0.0655	0.0645	0.0645	0.0635	0.0625	0.0615
2	0.0785	0.0775	0.0765	0.0755	0.0755	0.0745
3	0.0925	0.0905	0.0895	0.0885	0.0875	0.0865
4	0.1055	0.1035	0.1025	0.1015	0.1005	0.0985
5	0.1185	0.1165	0.1155	0.1135	0.1125	0.1115

Table 5.13: A subset of the Lagrangian indices for source 1 at $t = 1$ (Experiment 1)

case.

We have seen that the heuristic based on the Lagrangian indices competes the most favourably with the others that have been tested in this chapter in this scenario, only faring worse than the perfect information policy. These positive results for the Lagrangian index heuristic motivates further evaluation of the Lagrangian heuristic's performance in general. We expect the policy to perform very well if we can devise a computationally tractable implementation of the heuristic which supports problems with larger state spaces. We expect that the Lagrangian index policy's performance relative to the others considered in this problem would scale well with the size of the state space.

Comments regarding the indices

A subset of the fair charge $W_{k,t}(\alpha_{k,t})$ values for source 1 in Experiment 1 are shown in Tables 5.13 to 5.16. They show the fair charges for $t = 1, 10, 20$, and 50 for all combinations of 0 to 5 samples above the threshold C (vertical axes in the tables), and less than or equal to C (horizontal axes).

For example, the indices in row 3 and column 2 in Table 5.13 represent the charge $W_{1,1}(3, 2) = 0.0895$, the charge which is applicable at $t = 20$ with 3 samples exceeding the threshold and 2 samples which do not exceed the threshold.

The indices exhibit desirable general properties. The fair charges $W_{k,t}(\alpha_{k,t})$ are increasing in t and the number of sampled items of importance above C, where in

10 \1-9	0	1	2	3	4	5
0	0.0555	0.0545	0.0535	0.0535	0.0525	0.0515
1	0.0685	0.0675	0.0665	0.0665	0.0655	0.0645
2	0.0825	0.0815	0.0805	0.0795	0.0785	0.0775
3	0.0965	0.0945	0.0935	0.0925	0.0915	0.0895
4	0.1095	0.1085	0.1065	0.1055	0.1035	0.1025
5	0.1235	0.1215	0.1195	0.1185	0.1165	0.1155

Table 5.14: A subset of the Lagrangian indices for source 1 at $t = 10$ (Experiment 1)

10 \1-9	0	1	2	3	4	5
0	0.0575	0.0565	0.0565	0.0555	0.0545	0.0535
1	0.0715	0.0705	0.0695	0.0685	0.0675	0.0665
2	0.0885	0.0845	0.0835	0.0825	0.0815	0.0805
3	0.0995	0.0985	0.0965	0.0955	0.0945	0.0935
4	0.1135	0.1125	0.1105	0.1095	0.1075	0.1065
5	0.1275	0.1255	0.1245	0.1225	0.1205	0.1195

Table 5.15: A subset of the Lagrangian indices for source 1 at $t = 20$ (Experiment 1)

this case the only such importance value is 10. The indices also decrease in the number of sampled items with value less than or equal to C. Intuitively we would expect these properties for the fair charges.

For the indices, the marginal increase per sampled item with importance greater than C is generally greater than the marginal decrease per sampled item with importance less than equal to C. If we view sampled items of importance 10 as successes and all other sampled items as failures, the Lagrangian heuristic says that a particular source can still be valuable if the processor samples many failures from that source, as long as she has also sampled a few successes. Since the priors for all of these sources are heavily skewed towards sampling failures, then it would make sense that the aversion to individual failures is not pronounced in the behaviour of the indices.

10 \1-9	0	1	2	3	4	5
0	0.0625	0.0625	0.0615	0.0605	0.0595	0.0585
1	0.0775	0.0765	0.0755	0.0745	0.0735	0.0725
2	0.0925	0.0915	0.0895	0.0885	0.0875	0.0865
3	0.1075	0.1055	0.1045	0.1205	0.1015	0.1005
4	0.1225	0.1205	0.1185	0.1175	0.1155	0.1135
5	0.1375	0.1355	0.1355	0.1315	0.1295	0.1275

Table 5.16: A subset of the Lagrangian indices for source 1 at $t = 50$ (Experiment 1)

5.6 Extended numerical studies using non-Lagrangian heuristics

In the preceding section we conducted studies which established the Lagrangian index method of source selection to be both viable and strong when compared against other source selection policies. Due to computational limitations of working with Lagrangian indices we were only able to conduct studies in a limited number of scenarios. The other source selection policies considered are not subject to the same limitations. In this section we conduct an extension of the previous study which excludes the Lagrangian index method from the list of candidates.

To test the remaining source selection heuristics 60 variants were run in this study. Each variant within the study consists of $K = 5$ competing sources and a maximum importance score of $N = 10$. Each variant operates under a unique combination of threshold policy implementation and initial conditions.

A summary of the various configurations is shown in Table 5.17. One can form all of the variants by selecting one option from each column. In all cases, 5000 simulations were run for each configuration in these studies.

Horizon length T	$1 - q_h$	Prior Distributions	Thresholding Policy
100	0.05	0 (Table 5.18)	Static
20	0.1	1 (Table 5.19)	Dynamic
	0.15	2 (Table 5.20)	
	0.20		
	0.30		

Table 5.17: Summary of study parameter codes

The static thresholding method sets a threshold value C and a random accpetance probability π for the entire horizon before the processor begins sampling items (see section 5.4.1). The processor allocates all sampled items with an importance value greater than or equal to the threshold C and allocates with probability π any sampled item with importance value equal to $C - 1$. The processor conducts some preparatory numerical work to search for values of C and π such that

the mean number of item allocations made is approximately the target number of allocations, i.e a solution to P^*.

The dynamic thresholding policy is the adapted source independent version established during the preliminary studies of the dynamic C implementation (see section 5.4.2). Recall that the thresholding policy at time t is defined by the threshold value C_t, and the random acceptance probability π_t. The processor accepts the item sampled at time t if it has an importance value greater than or equal to C_t. She accepts items with importance equal to $C_t - 1$ with probability π_t. We have

$$C_t = \lceil Nq_t \rceil \tag{5.46}$$

$$\pi_t = \lceil Nq_t \rceil - Nq_t. \tag{5.47}$$

where q_t is defined as in (5.40).

The source selection policies being tested in this study are the greedy and knowledge gradient policies including capped variants of KG (2,5, and 10 period maximum look ahead). As described in sections 5.3.1 (static) and 5.4.2 (dynamic) the desired cap levels are achieved by setting the corresponding value of J. We also have both Thompson and Optimistic Bayes sampling policies and the perfect information policy.

In all cases we are using threshold based policies to solve P^* by finding an appropriate threshold in $P^*(C)$. We then force a solution to P by imposing special allocation rules to overrule the thresholding policy in order to allocate the exact number of items required by the analyst. The processer either prevents additional allocations once the quota of item allocations has been reached, or allocates all remaining items if the number of remaining time periods is equal to the number of remaining allocations required to satisfy the allocation quota.

The experimental designs of Experiments 0 and 1 in Tables 5.18 and 5.19 should be familiar from Tables 5.1 and 5.2 in the preliminary studies. We now add

Experiment 2, described in Table 5.20, where the sources have been designed such that sources 1 and 5 are clearly the best and worst sources in terms of their prior item importance distributions. The rationale here is that we want Experiment 2 to exist as a control, as we very much expect all policies to choose source 1 when applied to the sources in Experiment 2.

Importance	1-8	9	10	Mean
source 1,2	4	4	2	5.263
source 3	4	9	3	5.250
source 4,5	4	5	1	5.237

Table 5.18: Prior importance distributions for sources in Experiment 0

Importance	1-3	4	5	6	7	8	9	10	Mean
source 1	93	6	3	1	7	4	2	8	2.52
source 2	78	1	1	1	1	1	1	20	2.72
source 3	84	7	6	5	4	3	2	1	2.4
source 4	75	1	1	1	1	1	10	10	2.68
source 5	21	2	2	2	2	2	2	2	2.93

Table 5.19: Prior importance distributions for sources in Experiment 1.

Importance	1	2-9	10	Mean
source 1	1	1	10	7.63
sources 2,3,4	1	1	1	5.5
source 5	10	1	1	3.37

Table 5.20: Prior importance distributions for sources in Experiment 2

5.6.1 Analysis

In this subsection, the results of all of the numerical studies relating to the discrete MABA problem will be analysed. The tables of results are collected in the next subsection in Tables 5.21 through 5.35. In this section we hope to establish the viability of the methodology that has been developed to solve this problem and

to make actionable conclusions about the competing source selection and item allocation policies that are being tested.

In Section 4.5 we considered a much narrower range of problem configurations due to the computational limitations we encountered in implementing the Lagrangian index policy. The heuristics in these experiments are not subject to such problems so we have been able to extend the analysis to include a broader range of scenarios.

Validation of data

Before looking at the data it is worth checking its overall viability by checking whether it fits with what we would expect in a broad sense. The perfect information policy in both the static and dynamic cases routinely ranks the highest in all of the individual study variants. This not only fits with what we would expect from this policy but also provides context to the performance of the other non-clairvoyant policies, which will be commented on in more detail as we proceed with the analysis.

Another key sanity check for the data is how the mean Bayes returns relate to total number of items passed, holding all other parameters equal. We expect the mean Bayes returns to be concave increasing in the value of $1 - q_h$. The total importance value of the allocated items increases, because strictly more items are allocated. However, the mean item importance of the allocated items decreases as the processor accepts more items, because the processor must be less selective in order to accept a larger number of items. By examining the upwards progression of the mean Bayes returns for corresponding experiments, we see that in all cases that the overall returns are indeed concave increasing in the number of items for all variants within the study.

We can also make side by by side comparisons for the study variants where $T = 100$ and $T = 20$ within the individual studies. In all directly comparable study variants, the relative performance ranking of the source allocation policies

is the same between the $T = 100$ and $T = 20$ versions of the same study. If a particular policy performs poorly or well under $T = 100$, it performs just as poorly or well with respect to the other policies under $T = 20$ and vice versa.

It does not appear that there are any discrepancies or unexplained patterns in the data that relate to anything else other than the choice of thresholding policy (static or dynamic) or the choice of source selection policy. We can now proceed to analyse the effects of these two key choices' effects on the final mean Bayes return.

Effect of thresholding policies

In this study, two thresholding policies were tested. The first of these is the static C approach, where a single thresholding policy value is tuned in advance of the problem horizon starting. The second being the dynamic C policy, where the thresholding policy value changes over time, responding to learnings made about the various sources.

In Experiments 1 and 2, the dynamic C thresholding policy tends to match or outperform the static C policy, holding all other experiment parameters equal. This also includes the performances of the perfect information policies in these instances too. The static C policy's calibration stage, where it chooses the value of C used in the true problem, is consistently worse at selecting a value of C than the dynamic C policy.

The only instances in which dynamic C doesn't always outperform static C is under the conditions of Experiment 3, where source 1 has a very high probability of producing the highest possible valued items over any other. In this instance, the threshold that results from the calibration phase of the static C method also happens to be a threshold which competes relatively on a par, and sometimes better than, the thresholds returned by dynamic C. It should be noted that this conclusion applies specifically to the perfect information, Optimistic Bayes and knowledge gradient policies. In the Static C cases, the Greedy, Thompson sampling and capped Knowledge Gradient policies' performances are universally weaker across

all variants in Experiment 2. The dynamic C thresholding policy's tendency to lower the threshold slightly over the problem horizon results in it accepting lower importance items at the end of problem horizons voluntarily, without having to pass those types of items by default to fulfil the allocation quota. The static C policy by definition does not alter its behaviour once it is calibrated and in this particular case, its choice of threshold proves to be the best fit for the problem.

From the results that have been gathered, it's clear that the dynamic C thresholding policy is preferred to the static policy, although we have seen that it can be flawed under certain conditions compared to the static C case. However, the dynamic C option, unlike static C, does not require an expensive calibration phase for each unique problem setting that it encounters, making dynamic C the more viable option for operational deployment in a situation which is explicitly time critical. The caveat for recommending dynamic C would be that until it is known what a typical problem setting would look like for an intelligence gathering exercise, it is hard to know how dynamic C would perform. The source independent Dynamic C approach may need adaptation for certain problems as it is not obvious that its good performance is consistent across all problem types. The heuristic that we have so far is likely to need further adaptation going forward.

Now that we have discussed the relative merits of the two thresholding policies, let us now turn our attention to the source selection policies.

Effect of source selection policy

The numerical results paint a clear picture as to how the source selection policies compare in terms of performance.

Optimistic Bayes sampling, second only to perfect information, tends to yield the greatest mean Bayes returns across all of the studies and even though it doesn't always strictly place highest in the relative performance rankings of the non-clairvoyant policies, it is by far the least susceptible to significant failure. The recommendation that emerges from this study would be that optimisitc Bayes

sampling be used in an operational setting wherever possible. However, our limited evaluation of the Lagrangian index heuristic shows promising results as it has performed very well where it can be implemented. We would expect the policy to continue to perform favourably if the computational issues relating to its wider implementation could be overcome.

The knowledge gradient policy would be the next best option after optimistic Bayes for a source selection policy and would also be generally suitable for deployment.

The remaining policies can largely be summarised as inferior cousins of knowledge gradient or optimistic Bayes. The Thompson sampling and greedy approaches are the most consistent after knowledge gradient in terms of performance. The greedy approach tends to be a lesser version of knowledge gradient and Thompson sampling has the same relationship with optimistic Bayes.

The drawback of greedy is that it doesn't actively explore the available sources so its resulting behaviour is slower to adapt. The greedy approach almost functions on a par with knowledge gradient in problems where the value of exploration is low, in which case greedy and knowledge gradient are very similar. The Thompson sampling approach's ability to index sources below their expected sampled item importance value has proven to hinder its performance in comparison to optimistic Bayes. This is particularly noticeable in Experiment 2, where the item importance distributions of all five sources are heavily skewed towards the lower values, causing Thompson samples to routinely undervalue sources.

The degree to which the drawbacks of the greedy and Thompson sampling policies affect their performance (compared to knowledge gradient and optimistic Bayes respectively) is largely problem dependent, but generally speaking, Thompson sampling tends to perform better than greedy because optimistic Bayes performs better than knowledge gradient. Additionally, since the drawback of greedy is largely negated in situations where learning is less valuable, its not a policy that is recommended.

The capped variants of knowledge gradient have performances which lie somewhere between that of the greedy and knowledge gradient policies, which makes sense as the formulation of the capped knowledge gradient policy tends towards greedy as the cap tends to zero and tends to KG as the cap tends towards $t-1$. It is evident from this study that there is no particular benefit to be obtained from adjusting the effect of the future expected gains in the knowledge gradient. The intermediate KG policies created by adjusting J to be less than T and greater than 0 has not resulted in a policy which consistently outperforms either KG or greedy in any experiment that we have conducted.

Anomalous Results

There are some exceptions to the general patterns from the numerical results discussed so far. We now seek to explain these anomalies and gain further insights.

In Experiment 1, for $1 - q_h = 0.05$ (see Table 5.26), we see in the static cases for both $T = 20$ and $T = 100$ that the greedy policy and all four variants of the knowledge gradient perform exceptionally poorly. Checking the mean value of the threshold applied to the sources, we find that each policy's thresholding behaviour in these experiments is always to accept items with value 10 because in both cases it is the thresholding policy which on average returns a number of item allocations closest to the maximum number of allocations desired by the analyst. The strict exclusion of low value items in this experiment is common across all policies in this scenario. The noticeably poor behaviour of the greedy and knowledge gradient policies is not because the thresholding policies are unfavourably different. However these policies are struggling to find enough high value items to compete with their competitors. This is because these policies are somehow ill suited to this particular experimental set up and because the threshold has been set too high, which compounds the problems faced by these policies further. It is not apparent how the specific nature of this problem setting informs us where it may occur in other scenarios nor what those scenarios would look like. Additionally, this par-

ticular case is an extreme case of the general trend of these policies performing poorly.

However in Experiment 1 with $1 - q_h = 0.1$, we find that the optimistic Bayes policy, which is generally the best performing policy in these numerical experiments, does not keep up with the other policies and only surpasses Thompson sampling in both the $T = 100$ and $T = 20$ case. This anomaly is linked to the unequal thresholding values under which the heuristic policies operate. In both cases, the optimistic Bayes policy operates under a higher, and therefore stricter, threshold then all of the other policies. The optimistic Bayes operates under $C = 8$ on average (in the $T = 100$ case) and $C = 7$ (in the $T = 20$ case) whilst the other policies use $C = 5$ or $C = 6$ in both cases. We see that optimistic Bayes may have a slightly lower mean Bayes return in these experiments, but the higher thresholding policy results in the allocated items (other than those allocated at the end of the horizon) being of a higher quality, even if there are fewer of them. This example does highlight the inadequacy of comparing source allocation policies under different thresholding policies when there are outliers. A subject for future work would be to devise an effective way to hold all source selection heuristics to the same standard.

Concluding remarks

This numerical study was conducted to test whether the implementation of the framework for solving the discrete MABA problem (P) was workable and also to test the effectiveness of various source selection policies within that framework.

The study has shown the implementation does work and that it can be used to approximate a solution to P reasonably well. The static C and dynamic C thresholding policies have each shown to have their flaws, but nothing so cataclysmic that it prevents the use of either, although dynamic C proved to be better in these studies. In the case of the static C, it appears as if calibrating C towards passing the specified number of items dictated by $\lfloor T(1 - q_h) \rfloor$ may not be the

best approach. A costly search over the possible thresholds which targets rewards directly may be a good option to pursue in future work. In any case, the static C also suffers operationally from requiring a calibration phase to tune C in the first place, which may be too costly for operational use. Dynamic C thresholding, although the best in this study, would require extensive testing to truly establish its general effectiveness and beyond the scope of this work.

The picture is more clear when we look at the source selection policies. Optimistic Bayes is a firm choice, although knowledge gradient is also viable. The other policies considered can be ruled out as being inferior variants of these two frontrunners, although one would choose the Thompson sampling policy as the third most consistent performer of those tested.

Considering the Lagrangian index policy again, we saw in a previous study that its performance level was second only to the perfect information policy. In future work, efforts should be made to render the Lagrangian index policy computationally tractable in a wider array of scenarios so we can establish whether its superior performance is consistent more broadly. We believe that its performance in the limited studies that we have seen it in so far motivate the effort as the Lagrangian index policy has performed well so far.

5.6.2 Results of study

Policy	Mean Bayes return ($T = 100$)	s.e.	($T = 20$)	s.e.
Greedy	34.72	0.09	6.89	0.03
Knowledge Gradient	39.11	0.09	6.82	0.03
Thompson	41.81	0.09	7.66	0.03
Optimistic Bayes	42.46	0.08	7.85	0.03
KG Capped at 2	34.42	0.09	6.82	0.03
KG Capped at 5	34.42	0.09	6.82	0.03
KG Capped at 10	37.70	0.09	6.82	0.03
PI	44.45	0.09	8.10	0.03
Greedy (Dynamic)	47.32	0.01	9.38	0.01
Knowledge Gradient (Dynamic)	47.89	0.01	9.45	0.01
Thompson (Dynamic)	26.81	0.12	5.30	0.05
Optimistic Bayes (Dynamic)	47.89	0.01	9.46	0.01
KG Capped at 2 (Dynamic)	47.57	0.01	9.44	0.01
KG Capped at 5 (Dynamic)	47.59	0.01	9.45	0.01
KG Capped at 10 (Dynamic)	47.57	0.01	9.44	0.01
PI (Dynamic)	48.35	0.01	9.47	0.01

Table 5.21: Bayes returns for Experiment 0 with $1 - q_h = 0.05$

Policy	Mean Bayes return ($T = 100$)	s.e.	($T = 20$)	s.e.
Greedy	60.81	0.12	12.36	0.05
Knowledge Gradient	65.71	0.12	12.30	0.05
Thompson	70.41	0.15	13.67	0.05
Optimistic Bayes	73.11	0.15	14.13	0.05
KG Capped at 2	60.44	0.12	12.30	0.05
KG Capped at 5	60.44	0.12	12.20	0.05
KG Capped at 10	60.44	0.12	12.30	0.05
PI	73.93	0.12	14.78	0.05
Greedy (Dynamic)	91.71	0.02	18.12	0.02
Knowledge Gradient (Dynamic)	92.45	0.02	18.32	0.02
Thompson (Dynamic)	76.29	0.13	15.17	0.06
Optimistic Bayes (Dynamic)	92.46	0.02	18.30	0.02
KG Capped at 2 (Dynamic)	91.99	0.02	18.28	0.01
KG Capped at 5 (Dynamic)	91.96	0.02	18.26	0.02
KG Capped at 10 (Dynamic)	92.00	0.02	18.27	0.01
PI (Dynamic)	95.01	0.01	18.60	0.02

Table 5.22: Bayes returns for Experiment 0 with $1 - q_h = 0.1$

Policy	Mean Bayes return ($T = 100$)	s.e.	($T = 20$)	s.e.
Greedy	86.34	0.14	17.55	0.05
Knowledge Gradient	91.15	0.14	17.48	0.05
Thompson	95.95	0.17	19.04	0.06
Optimistic Bayes	95.95	0.17	19.65	0.06
KG Capped at 2	85.98	0.14	17.48	0.05
KG Capped at 5	85.98	0.14	17.48	0.05
KG Capped at 10	85.98	0.14	17.48	0.05
PI	97.51	0.13	25.44	0.04
Greedy (Dynamic)	133.98	0.05	26.47	0.02
Knowledge Gradient (Dynamic)	135.73	0.03	26.84	0.02
Thompson (Dynamic)	95.02	0.19	18.91	0.08
Optimistic Bayes (Dynamic)	135.89	0.03	26.83	0.02
KG Capped at 2 (Dynamic)	135.30	0.03	26.80	0.02
KG Capped at 5 (Dynamic)	135.31	0.03	26.83	0.02
KG Capped at 10 (Dynamic)	135.30	0.03	26.81	0.02
PI (Dynamic)	138.00	0.02	26.91	0.03

Table 5.23: Bayes returns for Experiment 0 with $1 - q_h = 0.15$

Policy	Mean Bayes return ($T = 100$)	s.e.	($T = 20$)	s.e.
Greedy	171.02	0.16	32.72	0.05
Knowledge Gradient	171.24	0.16	32.69	0.05
Thompson	169.89	0.14	32.50	0.05
Optimistic Bayes	170.88	0.15	32.50	0.05
KG Capped at 2	171.03	0.16	32.71	0.05
KG Capped at 5	171.10	0.16	32.68	0.05
KG Capped at 10	171.10	0.16	32.66	0.06
PI	172.06	0.13	33.02	0.05
Greedy (Dynamic)	175.47	0.06	34.43	0.03
Knowledge Gradient (Dynamic)	176.35	0.05	34.74	0.03
Thompson (Dynamic)	160.22	0.14	31.52	0.07
Optimistic Bayes (Dynamic)	176.41	0.05	34.76	0.03
KG Capped at 2 (Dynamic)	175.30	0.04	34.63	0.03
KG Capped at 5 (Dynamic)	175.29	0.05	34.63	0.03
KG Capped at 10 (Dynamic)	175.28	0.04	34.66	0.03
PI (Dynamic)	180.02	0.04	35.27	0.03

Table 5.24: Bayes returns for Experiment 0 with $1 - q_h = 0.2$

Policy	Mean Bayes return ($T = 100$)	s.e.	($T = 20$)	s.e.
Greedy	248.07	0.18	47.46	0.07
Knowledge Gradient	248.46	0.18	47.50	0.07
Thompson	244.36	0.18	47.15	0.06
Optimistic Bayes	247.55	0.18	47.32	0.07
KG Capped at 2	248.14	0.19	47.46	0.07
KG Capped at 5	248.23	0.19	47.46	0.07
KG Capped at 10	248.22	0.19	47.44	0.07
PI	250.39	0.16	48.12	0.06
Greedy (Dynamic)	250.30	0.10	48.96	0.05
Knowledge Gradient (Dynamic)	250.49	0.08	49.38	0.04
Thompson (Dynamic)	235.49	0.15	46.38	0.07
Optimistic Bayes (Dynamic)	250.63	0.07	49.39	0.04
KG Capped at 2 (Dynamic)	249.03	0.07	49.25	0.04
KG Capped at 5 (Dynamic)	249.01	0.07	49.28	0.04
KG Capped at 10 (Dynamic)	249.04	0.07	49.26	0.04
PI (Dynamic)	255.07	0.06	50.05	0.04

Table 5.25: Bayes returns for Experiment 0 with $1 - q_h = 0.3$

Policy	Mean Bayes return ($T = 100$)	s.e.	($T = 20$)	s.e.
Greedy	14.54	0.06	2.87	0.02
Knowledge Gradient	14.54	0.06	2.87	0.02
Thompson	33.77	0.15	6.05	0.04
Optimistic Bayes	33.11	0.12	5.66	0.04
KG Capped at 2	14.53	0.06	2.87	0.02
KG Capped at 5	14.54	0.06	2.87	0.02
KG Capped at 10	14.54	0.06	2.87	0.02
PI	34.77	0.16	6.23	0.03
Greedy (Dynamic)	47.69	0.01	9.41	0.01
Knowledge Gradient (Dynamic)	48.19	0.01	9.46	0.01
Thompson (Dynamic)	29.75	0.01	5.95	0.05
Optimistic Bayes (Dynamic)	48.18	0.01	9.46	0.01
KG Capped at 2 (Dynamic)	43.60	0.01	8.72	0.01
KG Capped at 5 (Dynamic)	43.62	0.02	8.71	0.01
KG Capped at 10 (Dynamic)	43.63	0.02	8.74	0.01
PI (Dynamic)	48.36	0.01	9.47	0.01

Table 5.26: Bayes returns for Experiment 1 with $1 - q_h = 0.05$

Policy	Mean Bayes return ($T = 100$)	s.e.	($T = 20$)	s.e.
Greedy	82.72	0.15	14.67	0.06
Knowledge Gradient	82.65	0.15	14.60	0.06
Thompson	71.14	0.16	12.91	0.05
Optimistic Bayes	75.90	0.14	13.58	0.05
KG Capped at 2	82.69	0.15	14.68	0.06
KG Capped at 5	82.65	0.15	14.59	0.06
KG Capped at 10	82.68	0.15	14.61	0.06
PI	92.43	0.15	16.60	0.06
Greedy (Dynamic)	94.34	0.02	18.44	0.02
Knowledge Gradient (Dynamic)	94.97	0.02	18.52	0.02
Thompson (Dynamic)	91.04	0.09	17.90	0.03
Optimistic Bayes (Dynamic)	94.73	0.02	18.47	0.02
KG Capped at 2 (Dynamic)	67.34	0.07	13.47	0.03
KG Capped at 5 (Dynamic)	67.36	0.07	13.47	0.03
KG Capped at 10 (Dynamic)	67.39	0.07	13.48	0.03
PI (Dynamic)	95.01	0.02	18.58	0.02

Table 5.27: Bayes returns for Experiment 1 with $1 - q_h = 0.1$

Policy	Mean Bayes return ($T = 100$)	s.e.	($T = 20$)	s.e.
Greedy	102.44	0.18	18.95	0.08
Knowledge Gradient	103.34	0.21	18.91	0.08
Thompson	87.92	0.18	16.55	0.06
Optimistic Bayes	103.14	0.17	18.84	0.06
KG Capped at 2	103.11	0.22	19.01	0.08
KG Capped at 5	103.27	0.22	19.10	0.08
KG Capped at 10	103.16	0.21	19.00	0.08
PI	106.79	0.23	19.88	0.88
Greedy (Dynamic)	136.01	0.04	26.73	0.03
Knowledge Gradient (Dynamic)	137.94	0.03	26.94	0.03
Thompson (Dynamic)	98.13	0.19	20.99	0.08
Optimistic Bayes (Dynamic)	137.82	0.03	26.95	0.03
KG Capped at 2 (Dynamic)	98.13	0.09	19.59	0.04
KG Capped at 5 (Dynamic)	98.03	0.09	19.62	0.04
KG Capped at 10 (Dynamic)	98.15	0.09	19.64	0.04
PI (Dynamic)	137.99	0.03	26.98	0.03

Table 5.28: Bayes returns for Experiment 1 with $1 - q_h = 0.15$

Policy	Mean Bayes return ($T = 100$)	s.e.	($T = 20$)	s.e.
Greedy	119.36	0.20	22.42	0.07
Knowledge Gradient	120.13	0.20	22.50	0.07
Thompson	103.17	0.20	19.78	0.07
Optimistic Bayes	122.06	0.20	22.62	0.07
KG Capped at 2	119.68	0.20	22.44	0.07
KG Capped at 5	119.96	0.20	22.47	0.07
KG Capped at 10	120.10	0.20	22.45	0.07
PI	123.58	0.17	23.10	0.06
Greedy (Dynamic)	177.99	0.06	34.83	0.04
Knowledge Gradient (Dynamic)	179.87	0.04	35.07	0.04
Thompson (Dynamic)	172.12	0.13	33.66	0.05
Optimistic Bayes (Dynamic)	179.27	0.05	35.04	0.04
KG Capped at 2 (Dynamic)	127.00	0.11	25.42	0.05
KG Capped at 5 (Dynamic)	126.92	0.11	25.42	0.05
KG Capped at 10 (Dynamic)	127.04	0.11	25.46	0.05
PI (Dynamic)	180.05	0.04	35.25	0.03

Table 5.29: Bayes returns for Experiment 1 with $1 - q_h = 0.2$

Policy	Mean Bayes return ($T = 100$)	s.e.	($T = 20$)	s.e.
Greedy	151.37	0.23	29.01	0.08
Knowledge Gradient	152.50	0.23	29.04	0.08
Thompson	133.54	0.21	25.97	0.07
Optimistic Bayes	154.41	0.24	29.15	0.08
KG Capped at 2	152.02	0.23	29.12	0.08
KG Capped at 5	152.33	0.23	29.16	0.08
KG Capped at 10	152.57	0.24	29.19	0.08
PI	154.43	0.21	29.51	0.08
Greedy (Dynamic)	251.77	0.10	49.36	0.05
Knowledge Gradient (Dynamic)	254.70	0.07	49.72	0.05
Thompson (Dynamic)	243.85	0.17	47.89	0.07
Optimistic Bayes (Dynamic)	253.86	0.08	49.72	0.05
KG Capped at 2 (Dynamic)	185.49	0.14	37.04	0.06
KG Capped at 5 (Dynamic)	185.46	0.14	37.04	0.06
KG Capped at 10 (Dynamic)	185.42	0.14	37.04	0.06
PI (Dynamic)	255.00	0.06	49.97	0.04

Table 5.30: Bayes returns for Experiment 1 with $1 - q_h = 0.3$

Policy	Mean Bayes return ($T = 100$)	s.e.	($T = 20$)	s.e.
Greedy	36.44	0.13	6.62	0.04
Knowledge Gradient	49.82	0.01	7.16	0.04
Thompson	49.32	0.03	9.79	0.01
Optimistic Bayes	50.00	0.01	10.00	0.01
KG Capped at 2	36.39	0.14	6.62	0.04
KG Capped at 5	36.35	0.14	6.62	0.04
KG Capped at 10	36.51	0.14	6.62	0.04
PI	50.00	0.11	10.00	0.04
Greedy (Dynamic)	47.68	0.01	9.41	0.01
Knowledge Gradient (Dynamic)	48.18	0.01	9.44	0.01
Thompson (Dynamic)	29.88	0.12	5.88	0.05
Optimistic Bayes (Dynamic)	48.16	0.01	9.46	0.01
KG Capped at 2 (Dynamic)	48.36	0.01	9.44	0.01
KG Capped at 5 (Dynamic)	48.38	0.01	9.44	0.01
KG Capped at 10 (Dynamic)	48.37	0.01	9.47	0.01
PI (Dynamic)	48.85	0.01	9.47	0.01

Table 5.31: Bayes returns for Experiment 2 with $1 - q_h = 0.05$

Policy	Mean Bayes return ($T = 100$)	s.e.	($T = 20$)	s.e.
Greedy	58.98	0.24	11.51	0.07
Knowledge Gradient	95.84	0.09	12.44	0.07
Thompson	95.88	0.10	19.16	0.03
Optimistic Bayes	100.00	0.01	19.99	0.01
KG Capped at 2	58.87	0.24	11.51	0.07
KG Capped at 5	58.82	0.24	11.51	0.07
KG Capped at 10	59.54	0.24	11.52	0.07
PI	100.00	0.01	19.99	0.01
Greedy (Dynamic)	93.38	0.02	18.35	0.02
Knowledge Gradient (Dynamic)	94.08	0.02	18.50	0.02
Thompson (Dynamic)	86.02	0.10	17.01	0.04
Optimistic Bayes (Dynamic)	93.92	0.02	18.43	0.02
KG Capped at 2 (Dynamic)	95.22	0.08	17.53	0.04
KG Capped at 5 (Dynamic)	95.26	0.08	17.45	0.04
KG Capped at 10 (Dynamic)	95.26	0.08	17.49	0.04
PI (Dynamic)	95.65	0.02	18.61	0.02

Table 5.32: Bayes returns for Experiment 2 with $1 - q_h = 0.1$

Policy	Mean Bayes return ($T = 100$)	s.e.	($T = 20$)	s.e.
Greedy	77.17	0.29	15.43	0.08
Knowledge Gradient	131.33	0.21	16.78	0.09
Thompson	138.68	0.20	27.99	0.04
Optimistic Bayes	149.98	0.01	29.96	0.01
KG Capped at 2	77.18	0.29	15.43	0.08
KG Capped at 5	77.18	0.29	15.45	0.08
KG Capped at 10	78.76	0.31	15.48	0.09
PI	150.00	0.26	29.98	0.04
Greedy (Dynamic)	136.30	0.04	26.70	0.03
Knowledge Gradient (Dynamic)	137.98	0.03	26.95	0.03
Thompson (Dynamic)	108.54	0.18	21.41	0.07
Optimistic Bayes (Dynamic)	137.96	0.03	26.93	0.03
KG Capped at 2 (Dynamic)	129.55	0.19	24.42	0.06
KG Capped at 5 (Dynamic)	129.56	0.19	24.50	0.06
KG Capped at 10 (Dynamic)	129.49	0.19	24.53	0.06
PI (Dynamic)	137.98	0.03	26.94	0.03

Table 5.33: Bayes returns for Experiment 2 with $1 - q_h = 0.15$

Policy	Mean Bayes return ($T = 100$)	s.e.	($T = 20$)	s.e.
Greedy	94.63	0.34	19.00	0.10
Knowledge Gradient	157.34	0.30	20.67	0.10
Thompson	177.80	0.29	36.17	0.06
Optimistic Bayes	199.92	0.02	39.87	0.01
KG Capped at 2	94.77	0.34	19.01	0.10
KG Capped at 5	94.87	0.34	19.03	0.10
KG Capped at 10	96.99	0.37	19.10	0.10
PI	199.96	0.28	19.17	0.10
Greedy (Dynamic)	179.39	0.05	34.97	0.04
Knowledge Gradient (Dynamic)	179.93	0.04	35.07	0.04
Thompson (Dynamic)	175.63	0.10	34.42	0.04
Optimistic Bayes (Dynamic)	179.75	0.05	35.13	0.03
KG Capped at 2 (Dynamic)	152.96	0.29	30.66	0.08
KG Capped at 5 (Dynamic)	153.12	0.28	30.61	0.08
KG Capped at 10 (Dynamic)	152.81	0.29	30.52	0.08
PI (Dynamic)	180.10	0.04	35.27	0.03

Table 5.34: Bayes returns for Experiment 2 with $1 - q_h = 0.2$

Policy	Mean Bayes return ($T = 100$)	s.e.	($T = 20$)	s.e.
Greedy	129.19	0.42	25.76	0.12
Knowledge Gradient	198.93	0.40	27.78	0.13
Thompson	247.42	0.49	50.66	0.10
Optimistic Bayes	299.21	0.06	59.27	0.03
KG Capped at 2	129.68	0.43	25.81	0.12
KG Capped at 5	129.87	0.43	25.82	0.12
KG Capped at 10	132.38	0.47	25.93	0.12
PI	299.45	0.45	59.76	0.10
Greedy (Dynamic)	253.99	0.07	49.67	0.05
Knowledge Gradient (Dynamic)	254.85	0.07	49.78	0.05
Thompson (Dynamic)	248.92	0.12	48.83	0.06
Optimistic Bayes (Dynamic)	254.76	0.07	49.85	0.05
KG Capped at 2 (Dynamic)	189.67	0.43	41.30	0.11
KG Capped at 5 (Dynamic)	190.28	0.42	41.30	0.11
KG Capped at 10 (Dynamic)	190.14	0.41	41.27	0.11
PI (Dynamic)	254.97	0.06	50.06	0.04

Table 5.35: Bayes returns for Experiment 2 with $1 - q_h = 0.3$

Chapter 6

Continuous MABA with judgement error

In the previous chapter, we were concerned with solutions to a discrete Dirichlet-Multinomial formulation of the MABA problem. The processor in that problem was faced with K sources from which T items were to be sampled, where the unknown distribution of the importances of these items was Dirichlet-Multinomial. We denoted the importance of the item sampled from source k with t samples remaining by $I_{k,t}$ and within the context of the discrete MABA, problem P was stated,

$$(P) : \max \mathbb{E} \left\{ \sum_{k=1}^{K} \sum_{t=1}^{T} I_{k,t} X_t \right\}, \tag{6.1}$$

Subject to:

$$\sum_{t=1}^{T} X_t \leq \lfloor T(1 - q_h) \rfloor \tag{6.2}$$

where $X_t = \begin{cases} 1 & \text{if item is allocated (passed to analyst) at time t} \\ 0 & \text{otherwise,} \end{cases}$ and $0 < q_h < 1$.

The value q_h is called the *horizon quartile* and denotes the proportion of items not allocated by the processor.

In this chapter we consider a continuous variant of the problem P in which

the item importances are non-negative real valued. We go on to develop an Exponential-Gamma-Gamma conjugate structure to model the processor's learnings about the item importance distribution of items sampled from sources.

We also extend the MABA problem to include the operational issue of processor judgement error, where the processor has explicit uncertainty about their own ability to assess the importance ratings of sampled items. We go on to model this uncertainty by having the distribution of the perceived item importances to be exponentially distributed with mean equal to the true (unknown) item importance. The processor's perceived notion of the item importance is only partially accurate, whereas it was perfect in the discrete model of Chapter 5.

First we develop the framework for the continuous MABA and establish how processor uncertainty is captured by the model. There we also formally set out the continuous version of the problem P. We then develop a Lagrangian relaxation of the continuous MABA problem as we did with the discrete MABA problem. In addition we also adapt the same selection of existing heuristic approaches that were used in the discrete case to analogous solution approaches for the continuous model.

The reader will recall that the implementation of the Lagrangian relaxation based solution in the discrete MABA model was limited by computational resources to a small number of testable cases. Although the implementation method is detailed in this chapter, moving forward with any meaningful numerical work was beyond the resources available during this project. However this chapter does contain numerical work relating to existing heuristic approaches, as well as the preliminary work that informed the design of those experiments.

6.1 Development of Model

In this section we develop a continuous MABA model which also accounts for the processor's uncertainty of her own assessments of the importance value of sampled items. From a horizon of length T, the processor seeks to sample $\lfloor T(1-q_h) \rfloor$

items from K available sources, where q_h is the proportion of items that are to be rejected by the processor. Denote by $Y_t, 1 \leq t \leq T$, the *processor's judgement* of the true importance value, I_t of the item sampled at time t. The processor's judgement is typically centered around the true importance, subject to some noise.

We model source k as follows: An item sampled from source k has a true importance rating R_k distributed as a gamma $\Gamma(\alpha_k, \beta_k)$, random variable and hence has mean $\frac{\alpha_k}{\beta_k}$ and variance $\frac{\alpha_k}{\beta_k^2}$. The R_k used as the precisions are not known. If all of the parameters α_k, β_k were known *a priori* then in order to achieve maximal total importance from some fixed number of items, the processor should focus on items from the source with maximal $\frac{\alpha_k}{\beta_k}$. Suppose, however, that the item importance distributions of the K sources are not fully known in advance. To make this approach tractable, suppose that the scale parameters β_k are known but that there is uncertainty regarding the shape parameters α_k. Taking a Bayesian viewpoint it is supposed that α_k has a gamma prior $\Gamma(\gamma_k, \delta_k)$. These are assumed independent for distinct k.

If R is the true importance of the item then we shall suppose that the conditional distribution of the inverse score $U = Y^{-1} \mid R \backsim \exp(R)$. Hence $E(Y^{-1} \mid R) = R^{-1}$ and the inverse score is unbiased for the inverse importance.

The problem P of Chapter 4 is to allocate a collection of sampled items with the maximum total importance value. The continuous MABA analogue for this problem, which we call P' is to maximise the total expected perceived item importance score of the allocated items instead, as the processor does not know the true importance. The processor intends to pass the most important items to the analyst by solving P'. We have

$$(P') : \max \mathbb{E}\left\{\sum_{t=1}^{T} Y_t X_t\right\}, \tag{6.3}$$

Subject to:

$$\sum_{t=1}^{T} X_t \leq \lfloor T(1-q_h) \rfloor. \tag{6.4}$$

where X_t is defined as in (6.4). As with problem P, the problem P' is hard to solve exactly so we approximate solutions to it. By changing the constraint in (6.4) and allowing the expected number of allocations to be less than $\lfloor T(1-q_h)\rfloor$ rather than the total number of allocations, we create the problem $P^{*\prime}$, the continuous analogue to P^*. We have

$$(P^{*\prime}) : \max \mathbb{E}\left\{\sum_{t=1}^{T} Y_t X_t\right\}, \tag{6.5}$$

Subject to:

$$\mathbb{E}\left(\sum_{t=1}^{T} X_t\right) \leq \lfloor T(1-q_h)\rfloor. \tag{6.6}$$

The problem $P^{*\prime}$ is also difficult to solve so we introduce a further relaxation step. We use a Lagrangian multipler C to bring the constraint of problem $P^{*\prime}$ (see (6.7)) into the objective of a new problem which we call $P^*(C)'$. Items sampled by the processor from any source are allocated and passed to the analyst if and only if their item importance value is greater than C.

We now state the relaxed version of the processor's problem for this chapter

$$(P^*(C)') : \max \mathbb{E}\left\{\sum_{t=1}^{T} \left(C^{-1} - Y_t^{-1}\right)^+ X_t\right\}. \tag{6.7}$$

The form of the objective in (6.7) is taken because Y_t, and some simple functions of it, will have an infinite mean while Y_t^{-1} does not. As with the discrete MABA model of Chapter 5, the processor can search for a value of C (using a bisection method for example) such that the constraint in $P^{*\prime}$ is satisfied. In this way the relaxed problem $P^*(C)'$ can be used as a heuristic to provide solutions to $P^{*\prime}$. In turn, by manipulating the item allocation behaviour at the end of the problem horizon, we can force solutions to P' too. When there is no constraint on the total number of allocations ($q_h = 0$), the processor seeks to sample from the sources in such a way as to maximise the expected value $\mathbb{E}\left\{\sum_{t=1}^{T}\left(C^{-1} - Y_t^{-1}\right)^+\right\}$, where the expectation is understood to be taken with respect to the prior distributions

of the unknown α_k as well as over realisations of the process. The understanding implied by such an objective is that it is only the items for which $Y_t^{-1} - C^{-1} < 0$, equivalently $Y_t > C$, which are passed on to the analyst. We are effectively maximising the sum of the inverse scores of the allocated items.

If the processor solves this problem, she allocates a collection of items which she believes to have a high true importance as she discards items with perceived importance less than C. In turn she passes items to the analyst with the highest true importance scores. This forms the continuous analogue to the $P^*(C)$ problem in the discrete MABA setting.

For ease of notation we drop the source identifying subscript k in what follows.

The implied distribution of the inverse scores U is given by

$$\begin{aligned} f(u \mid \alpha) &= \int f(u \mid r)\, g(r \mid \alpha)\, dr \\ &= \int r e^{-ru} \frac{r^{\alpha-1}\beta^{\alpha} e^{-\beta r}}{\Gamma(\alpha)} dr \\ &= \frac{\alpha\beta^{\alpha}}{(\beta+u)^{\alpha+1}}, u \geq 0. \end{aligned} \tag{6.8}$$

If α has the prior $\pi = \Gamma(\gamma, \delta)$ then its posterior based on a single importance score $y = u^{-1}$ is given by

$$\begin{aligned} \pi(\alpha \mid u) &\propto \pi(\alpha) f(u \mid \alpha) \\ &\propto \alpha^{\gamma-1} e^{-\delta\alpha} . \frac{\alpha\beta^{\alpha}}{(\beta+u)^{\alpha+1}} \\ &\propto \alpha^{\gamma} \exp\left[-\alpha\left\{\delta + \ln\left(\frac{\beta+u}{\beta}\right)\right\}\right] \\ &= \Gamma\left\{\gamma+1, \delta + \ln\left(\frac{\beta+y^{-1}}{\beta}\right)\right\}. \end{aligned} \tag{6.9}$$

Hence this is a conjugate structure. The posterior for α following importance scores $y_s, 1 \leq s \leq T$, on T items is

$$\Gamma\left\{\gamma + (T-t), \delta + \sum_{s=t}^{T} \ln\left(\frac{\beta + y_s^{-1}}{\beta}\right)\right\} \tag{6.10}$$

with associated posterior mean for the true mean importance $\frac{\alpha}{\beta}$ for the source given by

$$E\left(\frac{\alpha}{\beta} \mid y_s, t \leq s \leq T\right) = \frac{\gamma + (T-t)}{\beta\left\{\delta + \sum_{s=t}^{T} \ln\left(\frac{\beta + y_s^{-1}}{\beta}\right)\right\}} \tag{6.11}$$

which will be large for a collection of items with high importance scores.

Evolving beliefs about the importance of items emerging from the K sources will influence the processor as she decides how to sample from the sources to obtain a collection of items for the analyst with high total importance. When computing mean Bayes returns in numerical studies, samples from sources always come from their current posteriors.

In summary, the goal for the analyst is to sample the source with largest precision. However, these are unknown, and have a $Gamma(\alpha_k, \beta_k)$ prior. Furthermore, there is a $Gamma(\gamma_k, \delta_k)$ prior for α_k, and the analyst judgement of the precision is exponentially distributed with mean equal to the true (unknown) precision. As the sources are sampled, the analyst can reduce the uncertainty about the true precision of each source.

From this framework, we can extract some further analytical observations which will help us to devise and implement solutions to the problem $P^*(C)'$. We look at this in the next subsection.

6.1.1 Some preliminary observations/calculations

Using the framework developed so far in this section we will be able to define two useful concepts, the expected one-step return of a source and the Bellman equation for this particular variant of the intelligence problem.

For implementation of numerics and analysis in relation to this model we will need the marginal predictive p.d.f of $U = Y^{-1}$ (continuing to drop the source-identifying suffix k for the moment) when the current posterior for α is $\Gamma(\gamma, \delta)$.

The appropriate calculation is

$$\begin{aligned} f(u \mid \gamma, \delta) &= \int_0^\infty f(u \mid \alpha)\, \pi(\alpha \mid \gamma, \delta)\, d\alpha \\ &= \int_0^\infty \frac{\alpha \beta^\alpha}{(\beta+u)^{\alpha+1}} \frac{\delta^\gamma \alpha^{\gamma-1} e^{-\delta\alpha}}{\Gamma(\gamma)} d\alpha \\ &= \frac{\delta^\gamma \gamma}{(\beta+u)} \left(\delta + \ln\left(\frac{\beta+u}{\beta}\right)\right)^{-\gamma-1}, u \geq 0, \end{aligned} \tag{6.12}$$

and the corresponding value of $E\left(\left(C^{-1}-Y^{-1}\right)^+ \mid \gamma, \delta\right) = E\left(\left(C^{-1}-U\right)^+ \mid \gamma, \delta\right)$ is then given by

$$E\left(\left(C^{-1}-U\right)^+ \mid \gamma, \delta\right) = \int_0^{C^{-1}} \left(C^{-1}-u\right) f(u \mid \gamma, \delta)\, du$$

$$= \int_0^{C^{-1}} \left(C^{-1}-u\right) \frac{\delta^\gamma \gamma}{(\beta+u)} \left(\delta + \ln\left(\frac{\beta+u}{\beta}\right)\right)^{-\gamma-1} du, \tag{6.13}$$

which, using the substitution $x = \ln\left(\frac{\beta+u}{\beta}\right)$, becomes

$$= \int_0^{\ln\left(\frac{\beta+C^{-1}}{\beta}\right)} \left(C^{-1} - \beta\left(e^x - 1\right)\right) \delta^\gamma \gamma\, (\delta+x)^{-\gamma-1}\, dx. \tag{6.14}$$

It will abbreviate things below if we call this quantity in (6.14) $r(\gamma, \delta, C)$, the mean one step return from a source with current posterior $\Gamma(\gamma, \delta)$ in what follows.

We now need to restore the source identifying suffices in order to write the DP optimality equations for a $T-$horizon version of the above multi-armed bandit problem. We use (γ, δ) to denote the $2K-$vector $\left((\gamma_1, \delta_1), (\gamma_2, \delta_2), \ldots, (\gamma_K, \delta_K)\right)$ which, in the formulation below is taken to be the state of the system with time t to go to the end of the horizon, with $V_t(\gamma, \delta, C)$ the corresponding optimal value function. We have the Bellman equations

$$\begin{aligned} V_t(\gamma, \delta, C) = \max_{1 \leq i \leq K} \Bigg\{ & r_i(\gamma_i, \delta_i, C) \\ & + \int_0^\infty f_i(u \mid \gamma_i, \delta_i)\, V_{t-1}\left[(\gamma_j, \delta_j)_{j \neq i}; \left(\gamma_i + 1, \delta_i + \ln\left(\frac{\beta_i+u}{\beta_i}\right)\right), C\right] du \Bigg\}; \end{aligned}$$

$$V_1(\gamma, \delta, C) = \max_{1 \le i \le K} \{r_i(\gamma_i, \delta_i, C)\}. \tag{6.15}$$

For large K, T it will not be possible to develop solutions to the above problem using exact DP methods. We propose a number of heuristic approaches to circumventing this issue. The details of these proposals are given in Sections 6.2 and 6.3 of this chapter.

6.2 Lagrangian index approach to continuous MABA problem

In the same way that Lagrangian indices were developed for the discrete version of the MABA problem in Chapter 5, we now seek to develop analogous indices for the continuous version of the problem.

Technical Description

A Lagrangian index approach to source selection, rather than sampling from exactly one source per time period in the horizon, would instead allow many sources to be sampled per time period and impose a charge for each sample taken. Hence the objective of the Lagrangian relaxation is as follows:

$$V(W) := \max \left\{ \mathbb{E}\left(\sum_{k=1}^{K} A_k(T) - W N_k(T) \right) + WT \right\}, \tag{6.16}$$

where $A_k(T)$ is the total reward received from source k items over the $T-horizon$ and $N_k(T)$ is the number of source k items sampled over the horizon. The class of policies is such that *any number of sources* may be sampled at each time, but that a charge W is paid for each.

It will be true that as the charge W increases, so, under an optimal policy the mean number sampled, namely $\mathbb{E}\sum_{k=1}^{K} N_k(T)$ will decrease. One approach is to focus on the relaxation whose associated charge W^* achieves $\mathbb{E}\sum_{k=1}^{K} N_k(T) = T$,

namely that *on average* one item is sampled at a time (and hence T on average in total over the entire horizon) under an optimal policy.

To develop suitable index policies for the problem, observe that problem (6.16) has a source-wise decomposition. This means that

$$V(W) = \sum_{k=1}^{K} V_k(W) + WT, \tag{6.17}$$

where

$$V_k(W) := \max \mathbb{E}\left(A_k(T) - WN_k(T)\right), \tag{6.18}$$

is the maximal return from a problem defined in terms of source k only in which a decision has to be made at each time point as to whether to sample from the source (and claim the appropriate one-step reward $r_k(\gamma_k, \delta_k, C)$ but also incur the charge W) or not (and receive nothing) at each time point in the horizon. This source k problem is solved by a DP recursion as follows:

$$\begin{aligned} V_{k,t}(\gamma_k, \delta_k; W) = \max \Bigg\{ & r_k(\gamma_k, \delta_k, C) - W \\ & + \int_0^\infty f_k(u \mid \gamma_k, \delta_k) V_{k,t-1}\left(\gamma_k + 1, \delta_k + \ln\left(\frac{\beta_k + u}{\beta_k}\right); W\right) du; \\ & V_{k,t-1}(\gamma_k, \delta_k; W) \Bigg\}, \end{aligned} \tag{6.19}$$

where the first term on the r.h.s. above corresponds to sampling from source k and the second term corresponds to not sampling.

Any solution to (6.19) that samples from the source non-consecutively can be replaced by another solution that samples consecutively and achieves the same value as the non-consecutive solution. This is due to the fact that the state of the system remains constant (i.e. $V_{k,t}(\gamma_k, \delta_k; W) = V_{k,t-1}(\gamma_k, \delta_k; W)$) when the source k is not sampled in period t. More formally, for a solution V_k that samples consecutively to period s and stops sampling until period t for $t > s - 1$ (when a new sample is taken), there exists solution $\tilde{V}_k$ that samples consecutively up to period $s - 1$ and doesn't sample in periods $s - 2, \ldots, t$ such that the rewards

gained under both solutions are the same and $V_{k,t}(\gamma_i, \delta_k; W) = \tilde{V}_{k,s-1}(\gamma_k, \delta_k; W)$. Hence, without loss of generality, we can restrict our sampling schemes to stopping problems such that once the non-sampling action is taken, it remains in force for the rest of the horizon. The above DP is then be simplified to

$$V_{k,t}(\gamma_k, \delta_k; W) = \max\left\{ r_k(\gamma_k, \delta_k, C) - W + \int_0^\infty f_k(u \mid \gamma_k, \delta_k) V_{k,t-1}\left(\gamma_k + 1, \delta_k + \ln\left(\frac{\beta_k + u}{\beta_k}\right); W\right) du; 0 \right\}, \quad (6.20)$$

since it can be shown that the non-sampling option must have an associated future return of zero. Source k will be *indexable* if for all (γ_k, δ_k, t) there exists some charge $W_k(\gamma_k, \delta_k, t)$ for which

$$r_k(\gamma_k, \delta_k, C) - W + \int_0^\infty f_k(u \mid \gamma_k, \delta_k) V_{k,t-1}\left(\gamma_k + 1, \delta_k + \ln\left(\frac{\beta_k + u}{\beta_k}\right); W\right) du \geq 0 \Leftrightarrow W \leq W_k(\gamma_k, \delta_k, t). \quad (6.21)$$

In words, source k is indexable if, for the source k problem in every state (γ_k, δ_k, t) there exists an *indifference charge* $W_k(\gamma_k, \delta_k, t)$ such that it is optimal to sample source k in state (γ_k, δ_k, t) if and only if the actual charge W is below the indifference charge.

If all K sources are indexable then the optimal policy for the Lagrangian relaxation (6.16) has the following form: in every system state (γ, δ, t) sample all sources j for which the appropriate indifference charge $W_j(\gamma_j, \delta_j, t)$ exceeds the actual charge W. Note also that if $W \geq 0$ then $V(W)$ must be an upper bound for the problem in (6.15) on the optimal return (V^{opt}, say) and so we must have

$$\min_{W \geq 0} V(W) \geq V^{opt} \quad (6.22)$$

for the tightest such bound. It is not difficult to show that

$$\min_{W\geq 0} V(W) = V(W^*), \tag{6.23}$$

where W^* is as previously described, namely the value of the charge for sampling such that the mean number of items sampled equals the length of the horizon, T. Such upper bounds are often close to tight and hence provide effective benchmarks against which to measure the performance of heuristics. For ease of notation, we drop the source suffix k in what follows. We recall that the one-step rewards are given by

$$r(\gamma,\delta,C) = \int_0^{\ln\left(\frac{\beta+C^{-1}}{\beta}\right)} \left(C^{-1} - \beta\left(e^x - 1\right)\right)\delta^{\gamma}\gamma\left(\delta + x\right)^{-\gamma-1} dx. \tag{6.24}$$

This can be re-expressed as

$$r(\gamma,\delta,C) = E\left[\left(C^{-1} - \beta\left(e^X - 1\right)\right) I\left(X \leq \ln\left(\frac{\beta + C^{-1}}{\beta}\right)\right)\right], \tag{6.25}$$

where X is a positive-valued random variable with the pdf given by

$$f_X(x) = \delta^{\gamma}\gamma\left(\delta + x\right)^{-\gamma-1}, x \geq 0. \tag{6.26}$$

We develop a sequence of theoretical results, which make use of the following definition

Definition: We say that random variable X is stochastically larger (resp. smaller) than Y if the distribution function of X is everywhere smaller (resp. larger) than that of Y, namely,

$$F_X(y) := P(X \leq y) \leq (resp. \geq) P(Y \leq y) = F_Y(y) \text{ for all } y \tag{6.27}$$

It is easy to show that if X is stochastically larger than Y then $E\{\phi(X)\} \geq E\{\phi(Y)\}$ for any increasing function ϕ and $E\{\phi(X)\} \leq E\{\phi(Y)\}$ for any de-

creasing function ϕ. When X is stochastically smaller than Y these inequalities are reversed.

Suppose that the distribution of X is dependent upon some real-valued parameter θ. We say that $X \mid \theta$ is stochastically increasing (resp. decreasing) in θ if whenever $\theta_1 \geq \theta_2$ then $X \mid \theta_1$ is stochastically larger (resp. smaller) than $X \mid \theta_2$.

We will show that $V_t(\gamma, \delta, W)$ is increasing in γ (for fixed t, δ, W) and decreasing in δ (for fixed t, γ, W). It will then follow from its definition above that that the index $W(\gamma, \delta, t)$ is increasing in γ (for fixed t, δ) and decreasing in δ (for fixed t, γ). We first need the following.

Lemma 6.2.1. *One step return $r(\gamma, \delta, C)$ is non-decreasing in γ and non-increasing in δ.*

Proof. We have that

$$r(\gamma, \delta, C) = E\left(\left(C^{-1} - U\right)^{+} \mid \gamma, \delta\right), \tag{6.28}$$

where $U \mid (\gamma, \delta)$ has the *pdf*

$$\begin{aligned} f(u \mid \gamma, \delta) &= \int_0^\infty f(u \mid \alpha)\, \pi(\alpha \mid \gamma, \delta)\, d\alpha \\ &= \int_0^\infty \frac{\alpha \beta^\alpha}{(\beta + u)^{\alpha+1}} \frac{\delta^\gamma \alpha^{\gamma-1} e^{-\delta\alpha}}{\Gamma(\gamma)} d\alpha \\ &= \frac{\delta^\gamma \gamma}{(\beta + u)} \left(\delta + \ln\left(\frac{\beta + u}{\beta}\right)\right)^{-\gamma-1}, u \geq 0. \end{aligned} \tag{6.29}$$

Now it is straightforward to show, using a change of variable (x=ln $\left(\frac{\beta+u}{\beta}\right)$, that

$$\begin{aligned} P(U \leq v \mid \gamma, \delta) &= \int_0^v f(u \mid \gamma, \delta)\, du = \int_0^v \frac{\delta^\gamma \gamma}{(\beta + u)} \left(\delta + \ln\left(\frac{\beta + u}{\beta}\right)\right)^{-\gamma-1} du \\ &= 1 - \left(\frac{\delta}{\delta + \ln\left(\frac{\beta+v}{\beta}\right)}\right)^{\gamma}, v > 0 \end{aligned} \tag{6.30}$$

which is increasing in γ (for fixed δ) and decreasing in δ (for fixed γ) for each $v > 0$.

It follows (by definition of the terms concerned) that U is decreasing stochastically in γ (for fixed δ) and increasing stochastically in δ (for fixed γ). But the quantity $(C^{-1}-U)^{+}$ is a decreasing function of U. The result now follows from standard results regarding the stochastic ordering of random variables. □

Proposition 6.2.2. *$V_t(\gamma,\delta,W)$ is non-decreasing in γ (for fixed δ) for all t.*

Proof. The proof uses an induction on t. We first rewrite the DP equations determining $V_t(\gamma,\delta,W)$ as

$$V_t(\gamma,\delta,W)=\max\left\{r(\gamma,\delta,C)-W+\int_0^{\infty}f(u\mid\gamma,\delta)\,V_{t-1}\left(\gamma+1,\delta+\ln\left(\frac{\beta+u}{\beta}\right);W\right)du;0\right\}$$

$$=\max\left\{r(\gamma,\delta,C)-W+E_{U|(\gamma,\delta)}\left[V_{t-1}\left(\gamma+1,\delta+\ln\left(\frac{\beta+U}{\beta}\right);W\right)\right];0\right\}. \tag{6.31}$$

We firstly observe that the result is trivial for $t=1$, since we have that

$$V_1(\gamma,\delta,W)=\max\{r(\gamma,\delta,C)-W;0\} \tag{6.32}$$

and we now invoke Lemma (6.2.1). We now suppose the result to be for all horizons $\leq t-1$ and infer it to be true for horizon t. By the induction hypothesis regarding δ for horizon $t-1$ we know that $V_{t-1}\left(\gamma+1,\delta+\ln\left(\frac{\beta+U}{\beta}\right);W\right)$ is a non-increasing function of U. Suppose now that $\gamma_1>\gamma_2$. We know from the calculation in the proof of Lemma (6.2.1) that $U\mid(\gamma_1,\delta)$ is stochastically smaller than $U\mid(\gamma_2,\delta)$ for each fixed δ. It therefore follows from standard results on stochastic ordering that

$$E_{U|(\gamma_1,\delta)}\left[V_{t-1}\left(\gamma_1+1,\delta+\ln\left(\frac{\beta+U}{\beta}\right);W\right)\right]$$
$$\geq E_{U|(\gamma_2,\delta)}\left[V_{t-1}\left(\gamma_1+1,\delta+\ln\left(\frac{\beta+U}{\beta}\right);W\right)\right] \tag{6.33}$$

for each fixed δ. We now invoke the inductive hypothesis regarding γ to infer that

$$E_{U|(\gamma_2,\delta)}\left[V_{t-1}\left(\gamma_1+1,\delta+\ln\left(\frac{\beta+U}{\beta}\right);W\right)\right] \geq E_{U|(\gamma_2,\delta)}\left[V_{t-1}\left(\gamma_2+1,\delta+\ln\left(\frac{\beta+U}{\beta}\right);W\right)\right] \quad (6.34)$$

and hence (combining the two inequalities above) that

$$E_{U|(\gamma_1,\delta)}\left[V_{t-1}\left(\gamma_1+1,\delta+\ln\left(\frac{\beta+U}{\beta}\right);W\right)\right] \geq E_{U|(\gamma_2,\delta)}\left[V_{t-1}\left(\gamma_2+1,\delta+\ln\left(\frac{\beta+U}{\beta}\right);W\right)\right]. \quad (6.35)$$

We also have (from Lemma (6.2.1)) that

$$\gamma_1>\gamma_2 \Rightarrow r(\gamma_1,\delta,C)\geq r(\gamma_2,\delta,C). \quad (6.36)$$

Putting all the above together we see that when $\gamma_1>\gamma_2$ we may deduce that

$$r(\gamma_1,\delta,C)-W+E_{U|(\gamma_1,\delta)}\left[V_{t-1}\left(\gamma_1+1,\delta+\ln\left(\frac{\beta+U}{\beta}\right);W\right)\right]\geq$$

$$r(\gamma_2,\delta,C)-W+E_{U|(\gamma_2,\delta)}\left[V_{t-1}\left(\gamma_2+1,\delta+\ln\left(\frac{\beta+U}{\beta}\right);W\right)\right] \quad (6.37)$$

and hence

$$V_t(\gamma_1,\delta,W)\geq V_t(\gamma_2,\delta,W), \quad (6.38)$$

as required. Hence the inductive hypothesis regarding γ goes through. □

The induction argument regarding δ is similar.

Proposition 6.2.3. *$V_t(\gamma,\delta,W)$ is non-increasing in δ (for fixed γ) for all t.*

Proof. From the proof of Lemma (6.2.1) we infer that when $\delta_1>\delta_2$ that $U \mid (\gamma,\delta_1)$ is stochastically larger than $U \mid (\gamma,\delta_2)$ from which it follows trivially that $\delta_1+\ln\left(\frac{\beta+U}{\beta}\right) \mid (\gamma,\delta_1)$ is stochastically larger than $\delta_1+\ln\left(\frac{\beta+U}{\beta}\right) \mid (\gamma,\delta_2)$ which is in turn trivially stochastically larger than $\delta_2+\ln\left(\frac{\beta+U}{\beta}\right) \mid (\gamma,\delta_2)$. Hence using the

induction hypothesis regarding δ twice we infer that

$$E_{U|(\gamma,\delta_1)}\left[V_{t-1}\left(\gamma+1,\delta_1+\ln\left(\frac{\beta+U}{\beta}\right);W\right)\right] \leq E_{U|(\gamma,\delta_2)}\left[V_{t-1}\left(\gamma+1,\delta_1+\ln\left(\frac{\beta+U}{\beta}\right);W\right)\right]. \leq E_{U|(\gamma,\delta_2)}\left[V_{t-1}\left(\gamma+1,\delta_2+\ln\left(\frac{\beta+U}{\beta}\right);W\right)\right]. \quad (6.39)$$

As above, we combine this with the fact that

$$r(\gamma,\delta_1,C) \leq r(\gamma,\delta_2,C) \quad (6.40)$$

to infer that

$$r(\gamma,\delta_1,C) - W + E_{U|(\gamma,\delta_1)}\left[V_{t-1}\left(\gamma+1,\delta_1+\ln\left(\frac{\beta+U}{\beta}\right);W\right)\right] \leq r(\gamma,\delta_2,C) - W + E_{U|(\gamma,\delta_2)}\left[V_{t-1}\left(\gamma+1,\delta_2+\ln\left(\frac{\beta+U}{\beta}\right);W\right)\right] \quad (6.41)$$

and hence from (6.39) that

$$V_t(\gamma,\delta_1,W) \leq V_t(\gamma,\delta_2,W), \quad (6.42)$$

as required. It now follows that the inductive hypothesis regarding δ goes through.

□

Corollary 6.2.4. *$W(\gamma,\delta,t)$ is increasing in γ (for fixed δ) and decreasing in δ (for fixed γ) for all t.*

Proof. This follows immediately from the preceding results and the characterisation of $W(\gamma,\delta,t)$ in (6.21) as the smallest W value which makes $V_t(\gamma,\delta,W)$ zero. □

The above results provide the building blocks for understanding the structure of index policies. Suppose, for example, that at time t a source with current state (γ,δ) and index $W(\gamma,\delta,t)$ is sampled. At the following epoch, the above theory

tells us that the new index for the source will be $W\left(\gamma+1, \delta+\ln\left(\frac{\beta+y^{-1}}{\beta}\right), t-1\right)$ where y is the sampled importance at t. From the above theoretical results we have that

$$W\left(\gamma+1, \delta+\ln\left(\frac{\beta+y^{-1}}{\beta}\right), t-1\right) \geq W(\gamma, \delta, t) \Leftrightarrow y \geq \Psi(\gamma, \delta, t) \tag{6.43}$$

for some threshold $\Psi(\gamma, \delta, t)$. It will follow that if the sampled importance is large enough (above a threshold) then the index policy is guaranteed to return to the same source for a further sample at $t-1$. Hence the index policy embodies a form of *play on the winner* rule.

6.3 Other heuristic approaches

As with the discrete MABA model, we also develop other heuristic approaches for use in the continuous MABA framework. Doing so will demonstrate a broader number of ways to develop solution approaches to the continuous form of the intelligence problem. Knowledge gradient, Thompson sampling and optimistic Bayes sampling methods are adapted in this section.

Each of the heuristics described in this section are adapted to provide solutions to $P^{*\prime}$ via the relaxed problem $P^*(C)'$. The processor searches for the threshold C in each case that satisfies the constraint on the expected number of allocations in the problem $P^{*\prime}$. It is then possible to force a solution to P' in each case by overruling the thresold policy at the end of each problem horizon to ensure the exact number of required allocations dictated by the constraint are made. In the case that the processor allocates the maximum number of items early, she refuses to allocate any remaining sampled items, even if their importance exceeds C. Conversely, she allocates all remaining sampled items if the total number of time periods remaining in the problem is equal to the maximum allocations allowed minus the total number of allocations made so far. In this way, each of the preceding heuristics can be used to find solutions to P'

6.3.1 Knowledge Gradient methodology

Knowledge gradient methods, when applied to the continuous MABA problem, require an index computation for each source k, at every time period in the horizon. Using the quantities defined in equations (6.12) and (6.14) of Section 5.1.1, the KG index for source k with t time periods remaining is

$$KG(k,t) := r_k\left(\gamma_k, \delta_k, C\right) + (t-1)\int_0^\infty f_k\left(u \mid \gamma_k, \delta_k\right)\Delta\left(k,u\right)du, \tag{6.44}$$

where the $\Delta\left(k,u\right)$ term is defined as,

$$\Delta(k,u) := \max\left(\max_{j\neq k}\left\{r_j\left(\gamma_j,\delta_j,C\right)\right\}; r_k\left[\gamma_k+1, \delta_k+\ln\left(\frac{\beta_k+u}{\beta_k}\right), C\right]\right), \tag{6.45}$$

which, operationally, quantifies the highest expected one-step reward of any of the K sources given that source k was chosen most recently yielding an item with inverse importance score equal to u. At each time t, the processor computes the indices in (6.44) for each of the k sources and samples from the source k with the largest $KG(k,t)$. For a particular importance score u, it is straightforward enough to calculate $\Delta(k,u)$ by just computing each of the rewards individually using the following integral for each of the sources,

$$r\left(\gamma_k,\delta_k,C\right) = \int_0^{C^{-1}} \left(C^{-1}-u\right) f(u|\gamma_k,\delta_k)du, \tag{6.46}$$

where,

$$f(u|\gamma_k,\delta_k) = \frac{\delta_k^{\gamma_k}\gamma_k}{(\beta_k+u)}\left(\delta_k+\ln\left(\frac{\beta_k+u}{\beta_k}\right)\right)^{-\gamma_k-1}, u\geq 0. \tag{6.47}$$

Hence it is possible to compute $f(u|\gamma_k,\delta_k)\Delta(k,u)$ for a fixed u. An appropriate form of numerical integration over the range of u is then required to complete the computation of the KG indices.

It is also of interest to investigate the sensitivity of policy behaviour to the scaling term applied to expected future rewards, which appears as $(t-1)$ in (6.44). Instead one could select a cap, J and use the modified index

$$KG(k,t) := r_k\left(\gamma_k,\delta_k,C\right) + \min(J,(t-1))\int_0^\infty f_k\left(u \mid \gamma_k,\delta_k\right)\Delta\left(k,u\right)du, \tag{6.48}$$

and adjust J to control the weight that expected future rewards has on the KG index. Adjusting the weighting on the future like this was explored in [Glazebrook et al., 2012] with some success so we will try it here.

Knowledge gradient simulation

To compute the knowledge gradient indices for each source, one needs to separately compute the expected immediate rewards and the resulting expected future rewards. Both require some form of approximate integration.

One requires knowledge of the prior information state (γ_k, δ_k) for each of the K competing intelligence sources and also the known rate parameters β_k for the precisions of emerging items. The threshold C, which determines the minimum level of perceived importance an intelligence item needs to possess to be allocated and passed to the analyst, must be specified. The processor searches for a value of C such that the expected number allocations satisfies the constraint of $P^{*\prime}$.

To evaluate the various integrals involved in these computations, both numerical integration and Monte Carlo methods are used so that their performance can be compared. In the Monte Carlo case, the convergence rate is inversely proportional to the square root of the computing power available, while the convergence for the numerical integration methods will be at its fastest when only one dimension is considered. We will use the library GSL library `<gsl/gsl_integration.h>` in C++ to handle the implementation of the numerical integration.

We have a closed form for the cdf $F(u|\gamma, \delta)$ corresponding to the pdf in (6.47), namely

$$F(u|\gamma,\delta) = \int_0^\infty f(x|\gamma,\delta)dx = 1 - \delta^\gamma \left(\delta + \ln\left(\frac{\beta+u}{\beta}\right)\right)^{-\gamma}. \tag{6.49}$$

We can use the inverse transform method to obtain random draws from this distribution. We have the inverse cdf,

$$F^{-1}(u|\gamma,\delta) = \beta\left(\exp\left[\delta\left(1-u\right)^{-\frac{1}{\gamma}} - \delta\right] - 1\right), \tag{6.50}$$

so we can estimate $r(\gamma, \delta, C)$ by sampling from a suitably large sample of uniform random numbers and transforming according to (6.50). We want to estimate the

knowledge gradient for source k via a nested simulation. Let

$$\bar{r}(m;\gamma,\delta,C) = \frac{1}{m}\sum_{\ell=1}^{m}(C^{-1} - x_\ell)I(x_\ell \leq C^{-1}) \tag{6.51}$$

be the reward estimator, where I is an indicator function and the x_ℓ's are IID samples obtained by inverse transform using (6.50). The standard knowledge gradient estimator for source k and cap level J is

$$\bar{\alpha}_k(m,n;\gamma_k,\delta_k,C) := \bar{r}(m;\gamma_k,\delta_k,C) + \min(J,(t-1))\frac{1}{n}\sum_{i=1}^{n}\bar{\Delta}_k(u_i), \tag{6.52}$$

where the u_i's are IID samples drawn from the density $f(\cdot|\gamma_k,\delta_k)$ and

$$\bar{\Delta}_k(u_i) = \max\left\{\max_{j\neq k}\{\bar{r}(m,\gamma_j,\delta_j,C)\}, \bar{r}\left(m,\gamma_k+1,\delta_k+\ln\left(\frac{\beta_k+u_i}{\beta_k}\right),C\right)\right\}. \tag{6.53}$$

Analysis relating to Monte Carlo approach

Using MC approaches, preliminary computations were made to test the adequacy of the estimator $\bar{r}(m,\gamma,\delta,C)$ and it was found that it has a strong tendency to increase with γ and decrease with δ. This is consistent with the behaviour of $r(\delta,\gamma,C)$ described in Lemma 6.2.1.

An unforseen feature of this estimator which became known as a result of the preliminary work was that if one selects any common ratio, $\frac{\gamma}{\delta} = \mu$ and increases the values of γ and δ along the line defined by this common ratio condition, one tends to find that the value of the estimator $r(m,\gamma,\delta,C)$ also increases. This pattern held for all choices of μ tested and varying the choice of C did nothing to change this effect. An analytical proof that the estimator $r(m,\gamma,\delta,C)$ behaves in this way under these conditions is now given.

Lemma 6.3.1. *Let $y > 0$. It follows that $\delta\ln\left(1+\frac{y}{\delta}\right)$ is increasing in δ for the range $\delta > 0$.*

Proof. Note that

$$\ln\left(1+\frac{y}{\delta}\right)=\int_0^{\frac{y}{\delta}}\frac{1}{1+u}du>\frac{\frac{y}{\delta}}{1+\frac{y}{\delta}} \tag{6.54}$$

and hence that

$$\frac{d}{d\delta}\left(\delta\ln\left(1+\frac{y}{\delta}\right)\right)=\ln\left(1+\frac{y}{\delta}\right)-\delta\frac{\frac{y}{\delta^2}}{1+\frac{y}{\delta}}>0 \tag{6.55}$$

as required. □

Fix the value of $\mu>0$ and consider the r.v. X_δ with distribution parametrised by the positive $\delta>0$ with p.d.f.

$$f_\delta(x)=\frac{\delta^{\delta\mu}}{(\delta+x)^{\delta\mu+1}}\qquad(x\geq 0). \tag{6.56}$$

Corollary 6.3.2. *X_δ is stochastically decreasing in δ.*

Proof. The distribution function of X_δ is given by

$$F_\delta(y)=\int_0^y f_\delta(x)\,dx=1-\left(1+\frac{y}{\delta}\right)^{-\delta\mu}=1-\exp\left[-\delta\mu\ln\left(1+\frac{y}{\delta}\right)\right], \tag{6.57}$$

which by the above lemma is increasing in δ, $\forall y>0$, as required. □

Proposition 6.3.3. *$r(\delta\mu,\delta,C)$ is increasing in δ for any fixed $\mu>0$.*

Proof. We express $r(\delta\mu,\delta,C)$ as $E(\phi(X_\delta))$ where X_δ is as above and

$$\phi(x)=\left(C^{-1}-\beta\left(e^x-1\right)\right)^+,\qquad(x\geq 0), \tag{6.58}$$

is a decreasing function. Hence $r(\delta\mu,\delta,C)$ is the expectation of a decreasing function of a X_δ which is stochastically decreasing in δ. The proposition now follows. □

Comment: If $\mu>0$ is fixed, and $\gamma=\delta\mu$ then $E(\alpha)=\frac{\gamma}{\delta}=\mu$ is fixed and $Var(\alpha)=\frac{\gamma}{\delta^2}=\frac{\mu}{\delta}$ is decreasing in δ. Hence the above result concerns one-step rewards for a range of cases in which the mean of α is fixed at μ but its variance

is decreasing in δ. As this decrease in variance takes place the one-step rewards increase.

Such a result would mean that the quantity $r(\gamma, \delta, C)$ punishes exploration of more variable sources to some extent, in particular in such cases where a subset of competing sources have similar means. Proposition 6.4.3 tells us that the trade-off between exploitation and exploration in this problem setting is skewed in favour of exploitation. The degree to which exploitation is favoured in practice isn't apparent from the analysis alone but upcoming numerical work in this chapter will demonstrate that the imbalanced tradeoff can make this problem very difficult.

6.3.2 Thompson and OBS sampling

The continuous MABA analogue to the discrete version of the Thompson sampling heuristic requires that the processor make a random draw from source k to form the Thompson sample index TS_k for that source. We have

$$P\left(TS_k \leq (C^{-1} - u)\right) = F(u|\gamma_k, \delta_k) \tag{6.59}$$

To generate TS_k, one samples from $F^{-1}(U|\gamma_k, \delta_k)$ where U is $U[0, 1]$. This is also the method for generating random samples from the sources once a source has been chosen. It is very similar to the Thompson sampling method for the discrete case in its execution. In each time step, the processor samples from the source k with the greatest associated value of TS_k after all such indices have been generated.

The optimistic Bayes sampling index for source k, OBS_k is defined as

$$OBS_k = \max\left(TS_k, r(u|\gamma_k, \delta_k)\right), \tag{6.60}$$

which forms a parallel with the relationship between Thompson and Optimistic Bayes sampling methods in the discrete case. In the case of OBS, the processor samples from the source k with the greatest OBS_k value at each time t.

6.4 Preliminary numerical studies - KG based heuristics

Throughout the remainder of this chapter, the rewards shown are the respective values of $\sum_{t=1}^{T}\left((C^{-1}-1)-Y_t^{-1}\right)^+$ where the Y_t^{-1} are the inverse scores of the samples sourced and passed to the analyst at time t.

Before any implementation of tractable numerical studies took place, some preliminary work was undertaken using Monte Carlo integration and numerical integration to test whether these approaches are the candidate numerical implementations of the knowledge gradient based heuristics. We conduct some simple studies to observe whether anticipated behaviours occur and compare the peformance of these two implementations. Ultimately, we elect to proceed with the numerical integration approach on the grounds of its superior computational efficiency and it becomes the standard method we use to compute KG indices and their variants.

6.4.1 Trialling Monte Carlo integration implementation

We now document the preliminary studies designed to test the capabilities of the Monte Carlo integration implementation of the KG heuristics for this problem. Wherever the processor needs to compute a one step return, she uses the estimating method which was set out in section 6.4.1.

In these studies, scenarios involving three competing sources were considered. The sources were set up such that the prior parameters (γ, δ) governing their importance distributions are distinct pairs which have the property that the resulting value of $\bar{r}(m, \gamma, \delta, C)$ are reasonably close together. The purpose of this is to make the three sources equally attractive to a purely greedy policy before the first sampling choice is made. This places emphasis on differences in future expected rewards between the three sources. For any three source experiment considered, the same initial parameters were chosen.

	Source 1	Source 2	Source 3
γ	3	6	9
δ	2	4.128	6.27
$\mathbb{E}(\alpha)$	1.5	1.45	1.44
$Var(\alpha)$	0.75	0.35	0.22

Table 6.1: Parameter choices for 3 source scenarios

The global known scale parameter was set to $\beta = 2$, the hesistation constant to $C = 1$ and the length of the time horizon to be $T = 100$. For outer level computations the number of iterations was set to $m = 10^4$ whilst $n = 10^3$ iterations were used for inner computations. We consider 300 problem horizons and place no constraints on the number of allocations that the processor makes in these preliminaries.

The corresponding values of $\bar{r}(m; \gamma, \delta, C)$ (see equation (6.51) in Section 6.4.1) for the sources described in Table 6.1 were approximately 0.252, a figure based on the mean of 10^6 random MC iterates for each of the three pairs. The standard errors for all three sources were approximately 3×10^{-4}, which brings the immediate expected rewards for the sources reasonably close together on average.

Scenario Generation

Each experiment has fixed parameter values for γ_k and δ_k for each source k. For each horizon simulated, it is necessary to generate values for the processor's initial priors for source k from these values.

The approach is to take advantage of the fact that $\mathbb{E}(\alpha) = \frac{\gamma}{\delta}$ and $Var(\alpha) = \frac{\gamma}{\delta^2}$ to compute values for γ and δ for each scenario. This is achieved by taking a moderately large number of random samples from a $\Gamma(\gamma_k, \delta_k)$ distribution and recording the sample mean $\bar{x_k}$ and variance s_k of these draws. We can then set the prior values $\delta_{k,sim}$ and $\gamma_{k,sim}$ for the current run to be

$$\begin{aligned} \gamma_{k,sim} &= \bar{x_k} * \delta_{k,sim}. \\ \delta_{k,sim} &= \frac{\bar{x_k}}{s_k} \end{aligned} \tag{6.61}$$

Table 6.2: KG mean Bayes returns (MC approach, 300 runs)

Policy	Mean Bayes Return	s.e
KG	25.2	0.79
Cap 50	25.7	0.78
Cap 20	25.3	0.75
Cap 1	25.4	0.88
Greedy	25.8	0.89

Table 6.3: Mean proportion of samples from each source (MC, 300 runs)

Policy	Source 1	s.e	Source 2	s.e	Source 3	s.e
KG	25.2	1.86	35.5	2.12	39.3	2.13
Cap 50	25.9	1.89	34.7	2.11	39.3	2.13
Cap 20	31.1	2.02	32.9	2.08	36.0	2.10
Cap 1	41.1	2.17	28.6	2.01	30.3	2.03
Greedy	34.7	2.10	33.0	2.13	32.3	2.09

and these values form the prior parameters for the K sources for the given horizon. Using common random numbers, we ensure that each policy is given the same sequence of randomised horizons in any numerical studies performed.

The mean Bayes returns for problem $P^*(C)'$ (with $C = 1$) that the processor earns using some KG based source selection policies under this experimental set up are shown in Table 6.2 alongside the standard errors for those experiments. Monte Carlo methods are used for the numerical implementation. These preliminary results suggest that that in this particular problem setting, the processor should be indifferent between using the greedy and knowledge gradient source selection policies as the performance difference between the various cap levels looks to be insignificant.

The proportions of time (averaged across all runs) for which each source was selected by each policy are shown in Table 6.3. For the KG and Cap 50 policies, the choice of sources significantly favours sources 2 and 3 over source 1, but otherwise the three sources attract approximately equal attention from the source selection policies, which could imply that the sources are too similar on average for the MC integration method to pick a clear winner. The fact that the greedy policy doesn't significantly favour one source suggests that the MC integration method or the

problem setting is inappropriate so revising both of those issues before more work is carried out will be necessary.

There are also high levels of random noise in the MC methods used to compute the KG indices. Tables 6.2 and 6.3 show that the standard errors for these experiments are large. One way to decrease these standard errors would be to increase the number of problem instances from 300 to a high enough level to reduce the errors, but this MC approach has proven very expensive in terms of computing time. Obtaining the current study size took an excess of one day even with parallel computing. The approach is already too slow to be of use practically in an intelligence gathering context where expediency has already been established as an essential property for any proposed method. In the next subsection we will see that the direct numerical integration approach compares favourably to Monte Carlo integration in this regard and ultimately becomes our method of choice for the larger study in this chapter.

6.4.2 Trialling numerical integration implementation

In an effort to speed up the process of obtaining data and reduce the noise associated with these data, direct numerical integration was implemented to replace the Monte Carlo approach. One can use direct numerical integration to make the computations for all the quantities of the form $r(m; \gamma, \delta, C)$ wherever they appear in the algorithm, replacing the MC methods previously used to compute such terms. We also compute the KG indices using direct integrations and remove MC methods from the index computation altogether. The terms in (6.47) are computed using direct integration.

An effect of this is that the overall duration of individual runs in the experiment is greatly reduced, allowing more runs to be completed within a fixed time budget. To illustrate this, the 300 runs required to produce the data in Table 6.4 and Table 6.5 took approximately half an hour per experiment to generate whilst the data in Table 6.2 and Table 6.3 took at least twenty four hours per experiment to

Table 6.4: KG Mean Bayes Returns (Numerical integration, 300 runs)

Policy	Mean Bayes Return	s.e
KG	25.28	0.04
Cap 50	25.51	0.03
Cap 20	25.13	0.04
Cap 1	25.31	0.04
Greedy	25.42	0.04

Table 6.5: Rate of source selection (Numerical integration, 300 runs)

Policy	Source 1	s.e	Source 2	s.e	Source 3	s.e
KG	13.8	0.17	53.6	0.33	32.6	0.30
Cap 50	12.7	0.16	55.8	0.31	31.5	0.29
Cap 20	12.9	0.17	65.6	0.30	21.5	0.26
Cap 1	30.8	0.32	55.1	0.36	14.1	0.22
Greedy	100.0	0.0	0.0	0.00	0.0	0.00

produce. In some large part this addresses an earlier held concern that this solution method would be impractical for use in the an operational setting. There is also a major reduction in the standard error as a result of the change of approach for the same number of runs. Comparing the mean Bayes returns of Table 6.4 with the analogous MC data in Table 6.5 we see do not see any apparent effect that varying the look-ahead cap has on the mean Bayes returns. Any amount of exploration seems to produce the same mean reward level. This effect is now most likely to be a consequence of the parameters of the experiment; it may be that the choice of C is so low that most sampled items are likely to be allocated. The initial set-up of the sources makes the choice between exploration and exploitation irrelevant, because the cap level, unlike in the MC case, has a significant effect on source selection. Comparing Table 6.3 to Table 6.5 we see that the rate at which the various sources are selected for sampling in each instance vary between the two approaches. The direct integration approach significantly favours source 2 over sources 1 and 3 for non-zero cap levels whereas source 1 is chosen exclusively in the pure exploitation case. For caps of 20 and greater source 3 is also chosen significantly more often than source 1 but source 2 still prevails as the preferred choice on average.

The direct numerical integration approach favours source 1 outright in the

Table 6.6: KG Mean Bayes Returns (Numerical integration, 2500 runs)

Policy	Mean Bayes Return	s.e
KG	25.24	0.02
Cap 50	25.28	0.02
Cap 20	25.23	0.02
Cap 1	25.21	0.02
Greedy	25.27	0.02

greedy case, which has the greatest mean and variance, and when the cap level is non-zero the policy strongly prefers source 2 and decreasingly selects source 1 as the cap increases to favour the two sources with lower variances, possibly preferring to sample from source 2 more often than source 3 because of its slightly higher mean importance value.

This behaviour shows a counter-intuitive property of the non-zero cap (KG) policies in that a source with a lower $\mathbb{E}(\alpha)$ and lower $Var(\alpha)$ (source 2) is more likely to be chosen than a source with a a higher $\mathbb{E}(\alpha)$ and $Var(\alpha)$ (source 1) which suggests that exploration of more variable sources for potentially greater rewards is not taking place. Rather, the KG-type policies are displaying a more risk averse nature, opting to avoid sources with higher values of $Var(\alpha)$ to avoid potentially lesser rewards. This behaviour is most likely explained by Proposition 6.4.3, which tells us that this problem is skewed in such a way to disincentivise exploration to an extent. What is striking here is that this punishment of higher $Var(\alpha)$ is of a scale that a source which also has a higher $\mathbb{E}(\alpha)$ can still be passed over for a source where the same value is lower. It is a particular weakness of the KG based heuristics in this setting.

Since it is possible to perform more runs per unit time using direct numerical integration than the MC approach, this was exploited to collect the same statistics from 2500 runs in approximately the same amount of time it would take to obtain 300 runs from the MC approach. These results are shown in Tables 6.6 and 6.7.

The direct numerical integration approach does not render the greedy policy superior to any of the policies that place any weight on exploration by having a non-

Table 6.7: Rate of source selection (Numerical integration, 2500 runs)

Policy	Source 1	s.e	Source 2	s.e	Source 3	s.e
KG	11.8	0.10	56.6	0.20	31.6	0.19
Cap 50	12.7	0.10	55.8	0.20	31.5	0.18
Cap 20	12.6	0.11	63.9	0.20	23.5	0.17
Cap 1	32.3	0.21	53.5	0.22	14.2	0.14
Greedy	100.0	0.00	0.0	0.00	0.0	0.00

zero cap level. We find in this particular scenario that in terms of the Bayes returns, one may generally choose any source selection policy that has been considered so far and be confident that the rewards will be as high as if any other cap were chosen. The invariance of rewards to the cap level chosen for the source selection policy reflects the similarity of the mean importance across the three sources in this study but Table 6.7 shows us that the cap level does have a noticeable impact on which source is chosen, even if this does not translate into dramatic differences in rewards. We see once more that increasing cap levels, corresponding to placing increasing weight on the exploration term, results in source 2, and to a lesser extent source 3, receiving a greater proportion of the sampling effort from the source selection policy.

The greatest change is between the greedy policy and the Cap 1 policy, which includes some element of exploration, where the higher $Var(\alpha)$ value of source 1 appears to repel this policy and others with non-zero caps on exploration. This is a further indication that sources with greater variances are seen as less attractive by policies which take a one-step look into the future, despite source 1 having a greater $\mathbb{E}(\alpha)$ value in its item importance distribution. Further studies will be conducted to examine whether this trend holds in other scenarios.

6.4.3 Two source comparisons

To understand more clearly how variants of the KG policy are affected by the initial parameter choices for the item importance distributions of the sources, a series of two source scenarios were studied. The two source format should make

Table 6.8: Parameter choices for source 2 in studied scenarios

	γ	δ	$\mathbb{E}(\alpha)$	$Var(\alpha)$
Identical	3	2	1.50	0.750
Lesser $Var(\alpha)$	30	20	1.50	0.075
Lesser $\mathbb{E}(\alpha)$ and $Var(\alpha)$	8	6	1.33	0.222
Greater $\mathbb{E}(\alpha)$ and lesser $Var(\alpha)$	30	15	2.00	0.133

it easier to make conclusions regarding source selection behaviour without a third source creating interference.

The mean Bayes returns for the total importance of allocated items in problem $P^*(C)'$ (for $C = 1$) and source selection rates for the three studies are shown in Tables 6.9 through 6.16 and the parameter choices for each study are shown in Table 6.8. In all four of these studies, source 1 has $\gamma = 3$ and $\delta = 2$ and acts as a control source in each of the studies. The mean Bayes returns for the identical source version of the two source study do not vary with the cap level in any significant fashion. One does not expect to be able to gain much through exploration when the sources are so similar, as neither source will stand out as particular worthy of exploration, since both sources are identical. The rate at which the two sources are sampled from was evenly split between the two sources, as one would expect.

Table 6.9: Mean Bayes returns (Identical Sources, 10000 runs)

Policy	Mean Bayes Return	s.e
KG	25.26	0.035
Cap 50	25.28	0.034
Cap 20	25.26	0.035
Cap 1	25.24	0.034
Greedy	25.19	0.035

When source 1 has a higher $Var(\alpha)$ value, the mean Bayes returns decrease with the cap level as shown in Table 6.11. In this study it is possible to see a significant effect on rewards that can be attributed to the size of the cap placed on the exploration term in the KG index.

In Table 6.12, the greedy and Cap 1 policies choose significantly more often the

Table 6.10: Rate of source selection (Identical Sources, 10000 runs)

Policy	Source 1	s.e	Source 2	s.e
KG	50.8	0.41	49.2	0.41
Cap 50	50.6	0.41	49.4	0.41
Cap 20	50.3	0.41	49.7	0.41
Cap 1	50.6	0.41	49.4	0.41
Greedy	50.1	0.03	49.9	0.03

source with the smaller $Var(\alpha)$ value than the policies with caps of 20 or above. At this point one may believe that the exploration aspect of the KG index seeks source 1 because of its higher $Var(\alpha)$, contradicting the behaviour seen in the preliminary experiments where the sources with the lower $Var(\alpha)$ (and $\mathbb{E}(\alpha)$) were favoured as the weight on the exploration term went up. However the pure exploitation case favoured source 2 and we have already shown analytically with Proposition 5.4.3 that a lower $Var(\alpha)$ is favoured when the values of $\mathbb{E}(\alpha)$ are equal in this setting. Something else about the KG index must be driving sampling traffic to the alternative source as the cap level increases.

Table 6.11: Mean Bayes returns (Lesser $Var(\alpha)$ (source 2), 10000 runs)

Policy	Mean Bayes Return	s.e
KG	25.47	0.035
Cap 50	25.50	0.035
Cap 20	25.54	0.036
Cap 1	26.40	0.035
Greedy	26.47	0.035

Table 6.12: Rate of source selection (Lesser $Var(\alpha)$ (source 2), 10000 runs)

Policy	Source 1	s.e	Source 2	s.e
KG	82.3	0.25	17.7	0.25
Cap 50	81.7	0.26	18.3	0.26
Cap 20	81.3	0.26	18.7	0.26
Cap 1	7.7	0.13	92.3	0.13
Greedy	0.0	0.00	100.0	0.00

When source 1 is initialised with a larger $\mathbb{E}(\alpha)$ and a higher $Var(\alpha)$ parameter than source 2, there is a significant decrease in the rewards as the cap level is

increased as shown in Table 6.13, which coincides with a decrease in the sampling rate of source 1 as the cap level increases as shown in Table 6.14. It has already

Table 6.13: Mean Bayes returns (Lesser $\mathbb{E}(\alpha)$ and $Var(\alpha)$ (source 2), 10000 runs)

Policy	Mean Bayes Return	s.e
KG	23.81	0.035
Cap 50	23.86	0.034
Cap 20	23.87	0.034
Cap 1	24.42	0.034
Greedy	25.27	0.035

been observed when the $\mathbb{E}(\alpha)$ values are equal a high cap version of the KG index will choose the source with the lower $Var(\alpha)$ in this study. Source 2 has both the lower $Var(\alpha)$ and the lower $\mathbb{E}(\alpha)$ value of the two sources considered and still the high cap versions of the KG index significantly favour it. The greedy choice in all the studies so far appears to be consistent with the theory that higher $\mathbb{E}(\alpha)$ values are typically favoured and sources with lower $Var(\alpha)$ prevail in cases where sources have equal $\mathbb{E}(\alpha)$. However, the only consistent pattern with regards to the prevailing choice in the high cap cases is that the KG policy increasingly disagrees with the greedy policy as the cap increases. To further demonstrate this, the fourth study's source 2 possesses both a greater $\mathbb{E}(\alpha)$ and smaller $Var(\alpha)$ than source 1. Source 2 should be preferred very strongly to source 1 and in the pure exploitation case (and in the Cap 1 case) we see that source 2 is chosen without deviation in Table 6.16. However as the cap level becomes sufficiently high, we witness a strong deviation to the inferior source 1 in Table 6.16 despite the detrimental effect this has on rewards, which fall sharply as the cap level increases as shown in Table

Table 6.14: Rate of source selection (Lesser $\mathbb{E}(\alpha)$ and $Var(\alpha)$ (source 2), 10000 runs)

Policy	Source 1	s.e	Source 2	s.e
KG	8.5	0.11	91.5	0.11
Cap 50	8.6	0.11	91.4	0.11
Cap 20	8.8	0.12	91.2	0.12
Cap 1	47.1	0.38	53.9	0.38
Greedy	100.0	0.00	0.0	0.00

Table 6.15: Mean Bayes returns (Greater $\mathbb{E}(\alpha)$ and smaller $Var(\alpha)$ (source 2), 10000 runs)

Policy	Mean Bayes Return	s.e
KG	27.20	0.045
Cap 50	27.32	0.045
Cap 20	27.45	0.047
Cap 1	33.18	0.037
Greedy	33.11	0.037

Table 6.16: Rate of source selection (Greater $\mathbb{E}(\alpha)$ and smaller $Var(\alpha)$ (source 2), 10000 runs)

Policy	Source 1	s.e	Source 2	s.e
KG	74.5	0.27	25.5	0.27
Cap 50	74.0	0.28	26.0	0.28
Cap 20	72.0	0.30	28.0	0.30
Cap 1	0.0	0.00	100.0	0.00
Greedy	0.0	0.00	100.0	0.00

6.15.

It would appear in these two source studies the KG policy, with a sufficiently high weight placed on the future rewards term, will choose the next best source and rarely choose the source which prevails under pure exploitation. This can be shown analytically under certain conditions.

Rationale: Denote by k^* the greedy-optimal source. For sufficiently large t, the KG policy as defined in (6.44) will tend to select the source with the second greatest value of $r(\gamma, \delta, C)$ if $r(\gamma_{k^*}, \delta_{k^*}, C) > r\left(\gamma_{k^*} + 1, \delta_{k^*} + ln\left(\frac{\beta+u}{\beta}\right), C\right)$, where u is the inverse importance of the next sampled item.

Any source s in a given problem which is not k^* has a value of $r(\gamma_s, \delta_s, C)$ which is less than $r(\gamma_{k^*}, \delta_{k^*}, C)$ by definition. For source j, the value of the $\max_{j \neq s} r(\gamma_j, \delta_j, C) = r(\gamma_{k^*}, \delta_{k^*}, C)$, whereas for source k^* the equivalent value is again smaller by definition of k^*. If $r(\gamma_{k^*}, \delta_{k^*}, C) < r\left(\gamma_{k^*} + 1, \delta_{k^*} + ln\left(\frac{\beta+u}{\beta}\right), C\right)$ then the expected future rewards term of the KG index of the greedy-optimal source k^* will be strictly less than the equivalent terms for all other sources. When comparing the KG indices of the greedy-optimal source k^* with some other source

s, for t large enough, we have that

$$(t-1)\int_0^\infty f_s\left(u \mid \gamma_s, \delta_s\right)\Delta\left(s,u\right) - f_{k^*}\left(u \mid \gamma_{k^*}, \delta_{k^*}\right)\Delta\left(k^*,u\right)du$$

$$> r(\gamma_{k^*}, \delta_{k^*}, C) - r(\gamma_s, \delta_s, C) \tag{6.62}$$

and therefore $KG(s,t) > KG(k^*,t)$ for all $s \neq k^*$ and the maximal such $KG(s,t)$ is associated with the source with the greatest $r(\gamma_s, \delta_s, C)$, which is the source with the second greatest value of $r(\gamma, \delta, C)$.

Numerically, for $u > 0$ I found that the act of sampling from a source causes an update which results in the decrease of its immediate expected returns, at least for the cases considered in the studies shown so far. I am unclear whether this pattern holds generally so I'll leave it as an open issue for now. For $u > 0$, is $r(\gamma, \delta, C) > r\left(\gamma + 1, \delta + \ln\left(\frac{\beta+u}{\beta}\right), C\right)$ generally or are there specific conditions under which this is true?

6.5 Numerical Study: Existing approaches

We now look to examine the performance of heuristic approaches, which have been adapted to fit within the framework of the continuous MABA problem.

6.5.1 Experiment set-up

Six numerical studies were undertaken to make an assessment of the relative performance level of several source sampling policies. Two varieties of three source problem were considered which we will refer to as Experiments 1 and 2. The pararmeters for each of the sources in these experiments are shown in Tables 6.17 and 6.18. Both experiment types are run for each of the target horizon quantiles $q_h = 0.95, 0.85, 0.70$ with $T = 100$. The source selection policies considered are knowledge gradient, greedy, Thomspon sampling and optimistic Bayes sampling. We also consider capped variants of the KG policy with the cap J set to 1, 20,

	Source 1	Source 2	Source 3
γ	3	6	9
δ	2	4.128	6.27

Table 6.17: Parameter choices for 3 Experiment 1

	Source 1	Source 2	Source 3
γ	3	6	9
δ	4	8.257	12.54

Table 6.18: Parameter choices for Experiment 2

and 50.

In all of the experiments, the processor is trying to solve the problem P' for the various values of q_h. She does this by first solving $P^{*\prime}$ via $P^*(C)'$ and then forcing the number of allocations to be $\lfloor T(1-q_h) \rfloor$ to obtain a solution to P'.

As with the discrete MABA numerics, for each policy one first makes a binary search for the smallest integer values of C such that at least $\lfloor T(1-q_h) \rfloor$ items are allocated on average. For the purposes of the numerics here, the term on average refers to the mean allocation rate over 1000 horizons.

The resulting value of C for each heuristic tested is used as the threshold for that policy over 10000 randomly generated horizons. For each allocation decision there is a probability p that $C-1$ is used as the threshold (if $C > 1$) instead of C which is computed using the formula in (5.32) in Chapter 5. Due to the changing nature of the threshold, all rewards are computed relative to $(C-1)^{-1}$ instead of C^{-1}. Unlike the discrete case, items which are passed along to the processor at the end of the horizon such that exactly $\lfloor T(1-q_h) \rfloor$ items are allocated have a value of *zero* if their importance value is not large enough to be allocated under normal circumstances. This has the consequence of making poor runs slightly worse in extreme cases.

6.5.2 Results

Six sets of results showing the mean Bayes returns obtained by the processor under the various source selection policies are displayed in Tables 6.19 to 6.24. With regard to the results as a whole, the variants of the KG policy appear to differ very little in terms of their performance. There are some instances where one of these policies stands out from the others in an individual study, but no single KG variant consistently stands out superior.

The Thompson sampling method tends to perform significantly better than all other policies whereas the OBS policy's performance is inconsistent when compared to the other policies, and generally performs the worst of all. The only difference between the implementation of the two policies is that OBS always uses the expected item importance $r(u|\gamma_k, \delta_k)$ as the minimum index for each source k, regardless of the random draw from the posterior distribution. It is possible that in doing so, OBS neglects to explore the sources as much as Thompson sampling does. However, we previously saw in the discrete MABA problem (Chapter 5) that OBS performed very well in the numerical experiments there. This suggests that the problem with OBS in the continuous MABA problem isn't exclusively because of the policy itself. Instead the specific relationships between the continuous MABA problem (and the discrete MABA problem) require further analysis to determine why their performance varies so much between the two settings. We leave this as a subject for future work. The need to examine this variation was discovered at such a time that rendered it impossible for us to embark on this analysis in earnest.

In Tables 6.19 and 6.22 we see the results for Experiments 1 and 2 where $q_h = 0.95$ so the processor only allocates exactly 5 out of the 100 items seen. In these scenarios, the difference in performances among the policies is the narrowest. This is partly due to there being too few items in total for there to be significant variation in their performance. In any further studies it may be worth omitting problem cases where the number of allocated items is this small. Additionally, the

Policy	Mean Bayes return	s.e	C	p
KG	0.6809	0.0016	11	1.00
Greedy	0.6848	0.0016	11	1.00
KG Capped at 1	0.6831	0.0016	11	1.00
KG Capped at 20	0.6853	0.0016	11	1.00
KG Capped at 50	0.6835	0.0016	11	1.00
Thompson	0.6874	0.0016	11	1.00
Optimistic	0.6824	0.0016	11	1.00

Table 6.19: Mean Bayes returns for Experiment 1 with $1 - q_h = 0.05$

Policy	Mean Bayes return	s.e	C	p
KG	1.4228	0.0036	8	0.55
Greedy	1.4209	0.0037	8	0.55
KG Capped at 1	1.4171	0.0036	8	0.57
KG Capped at 20	1.4308	0.0037	8	0.52
KG Capped at 50	1.4296	0.0037	8	0.55
Thompson	1.8652	0.0044	8	0.13
Optimistic	1.4310	0.0039	8	0.33

Table 6.20: Mean Bayes returns for Experiment 1 with $1 - q_h = 0.15$

threshold value C has to be at its highest setting in order to filter the stream of incomng items sufficiently to only submit 5 of them.

In Table 6.21 the converse effect is observed where the number of allocations in the experiment is too great for there to be any noticeable difference in the policies' performances. In Experiment 1 30 items out of 100 are allocated and the policies are all equally capable of searching and allocating the best 30 items. This again highlights the importance of adequate parameter selection when designing experiments to test the capabilities of competing policies. This particular problem setup is too easy to allow superior policies to differentiate themselves from inferior ones.

The optimistic Bayes sampling heuristic has performed relatively poorly in the continuous MABA problem, which is in stark contrast to its performance in the discrete MABA problem, where it was second only to PI and Lagrangian, where applicable. It is not apparent why the optimistic nature of OBS is detrimental to its performance but it is clear that it is where the problem lies, seeing as Thompson

Policy	Mean Bayes return	s.e	C	p
KG	8.082	0.016	3	0.35
Greedy	8.028	0.017	3	0.83
KG Capped at 1	8.065	0.017	3	0.44
KG Capped at 20	8.056	0.017	3	0.56
KG Capped at 50	8.020	0.017	3	0.76
Thompson	8.073	0.018	3	0.34
Optimistic	7.914	0.019	3	0.89

Table 6.21: Mean Bayes returns for Experiment 1 with $1 - q_h = 0.30$

Policy	Mean Bayes return	s.e	C	p
KG	0.6764	0.0017	11	1.00
Greedy	0.6791	0.0017	11	1.00
KG Capped at 1	0.6773	0.0017	11	1.00
KG Capped at 20	0.6751	0.0016	11	1.00
KG Capped at 50	0.6765	0.0017	11	1.00
Thompson	0.6783	0.0017	11	1.00
Optimistic	0.6681	0.0017	11	1.00

Table 6.22: Mean Bayes returns for Experiment 2 with $1 - q_h = 0.05$

Policy	Mean Bayes return	s.e	C	p
KG	3.869	0.008	4	0.57
Greedy	3.875	0.008	4	0.38
KG Capped at 1	3.840	0.008	4	0.78
KG Capped at 20	3.867	0.008	4	0.57
KG Capped at 50	3.876	0.008	4	0.57
Thompson	3.873	0.009	4	0.55
Optimistic	3.736	0.011	4	0.89

Table 6.23: Mean Bayes returns for Experiment 2 with $1 - q_h = 0.15$

Policy	Mean Bayes return	s.e	C	p
KG	14.978	0.045	2	0.23
Greedy	14.899	0.046	2	0.23
KG Capped at 1	14.985	0.045	2	0.19
KG Capped at 20	14.894	0.046	2	0.27
KG Capped at 50	14.845	0.046	2	0.76
Thompson	16.210	0.046	2	0.01
Optimistic	14.077	0.044	2	0.89

Table 6.24: Mean Bayes returns for Experiment 2 with $1 - q_h = 0.30$

sampling has a much better performance when the two policies are compared in the continuous setting. Sampling from the posteriors to create indices seems to be effective, but augmenting those indices in the OBS case does not seem to work. It would require further scrutiny of how OBS differs in behaviour to Thompson to gain the insights required. In particular, it would be worth tracking how often OBS indices result in sampling from a source with a lower $\mathbb{E}(\alpha)$ because of a favourable draw from the posterior versus how often OBS indices fail to sample from sources with higher $\mathbb{E}(\alpha)$ because of unfavourable draws in the index creation stage.

It may be that the performance of OBS and Thompson sampling methods are entirely problem dependent. Conducting a study large enough to ascertain whether or not this is the case would be a task for some future work. The question of what exactly differentiates Thompson sampling's performance from that of OBS sampling in either the continuous or the discrete setting is not clear from the work that has been carried out. Apart from this concern, Thompson sampling is the most effective source selection heuristic across all studies in the continuous setting. However, because of the arising concern that its performance may not be consistent across a fuller range of problems, the standard KG heuristic would be a recommended alternative because of its consistent high level of performance across all of the experiments in the discrete and continuous settings. The standard KG heuristic is also an appealing choice in the discrete setting following this argument, as the consistency concerns relating to the Thompson sampling method in the continuous setting apply equally to the OBS method in the discrete case. Furthermore the Lagrangian index heuristic performs exceptionally well where it has been possible to implement it, but we are not yet in a position where it can be broadly used in all problem settings because of outstanding computational tractability issues.

The standard KG heuristic seems to be the most appropriate to use overall. There doesn't appear to be a tangible reason to use any of the capped variants for any specific reason from the data. Capping the KG has the effect of making it increasingly like the greedy policy (the lower the cap, the more like a greedy

policy KG becomes), and although we have shown analytically (see Prop. 6.4.3) and practically (see section 6.5.3) that exploration type behaviour in this setting can be costly, a pure exploitation policy can only marginally outperform KG within these edge case scenarios. From the results in this section, the cost of exploration appears to be worth paying as the greedy policy falls far behind when it cannot match the performance of KG.

What remains to be addressed in future work is to examine more closely the behaviour of Thompson and OBS sampling methods to determine the causes for the gulf in their relative performance levels. Additionallly, finding a heuristic to approximate a Lagrangian index approach for the continuous setting is desirable, as we have seen some evidence from the discrete setting that such an approach could work very well, even if we have only seen it in a limited setting in Chapter 5.

6.6 Future work: Existing heuristic approaches for implementing Lagrangian relaxation

Computational limitations have hindered our efforts to provide an implementation of the Lagrangian index source selection policy in the continuous MABA problem for a general horizon length T. We close this chapter with a brief discussion into how we could use existing techniques from the literature to develop a more computationally tractable solution approach.

A closed form approximation to the Whittle index is developed in [Brezzi and Lai, 2002] to simplify the exploration aspect of the indices. It is compatible with our Bayesian Gamma model and for clarity I'll state results to be consistent with the notation used in this document. For an alternative with reward distribution with shape parameter γ, rate parameter δ and t time periods reamining, the Brezzi-Lai index is

$$I_t^{BL}(\gamma,\delta) = \frac{\gamma}{\delta} + \frac{\sqrt{\gamma}}{\delta}\psi\left(\frac{1}{\delta \ln(\frac{t}{t-1})}\right) \tag{6.63}$$

where

$$\psi(x) = \begin{cases} \sqrt{x/2} & \text{if } s \leq 0.2 \\ 0.49 - 0.11(x/2)^{-\frac{1}{2}} & \text{if } 0.2 < s \leq 1 \\ 0.63 - 0.26(x/2)^{-\frac{1}{2}} & \text{if } 1 < s \leq 5 \\ 0.77 - 0.58(x/2)^{-\frac{1}{2}} & \text{if } 5 < s \leq 15 \\ (2\ln x - \ln\ln x - \ln 16\pi)^{\frac{1}{2}} & \text{if } s > 15. \end{cases} \tag{6.64}$$

The literature explains that I_t^{BL} has the same desirable properties as our true index $W(\gamma,\delta,t)$ in that it is increasing in t and γ and decreasing in δ. An alternative approximate index was developed in [Caro and Gallien, 2007] which performs favourably against the Brezzi-Lai index in the numerical experiments carried out in the literature. It is defined as

$$I_t^{CG}(\gamma,\delta) = \frac{\gamma}{\delta} + \frac{z_t\sqrt{\gamma}}{\sqrt{\delta^2+\delta^3}}, \tag{6.65}$$

where z_t is the solution to $(t-1)\Psi(z) = z$ where Ψ is the normal error function. Again, this index has the desired properties in relation to its arguments, so it is a candidate heuristic for $W(\gamma, \delta, t)$. The only obstacle to using both of these heuristics is adapting them to be compatible with the rewards framework of our MABA problem. It is not immediately obvious how this would be done and would require some further analytical work.

Implementation of existing heuristics to Lagrangian formulation of the MABA problem

A natural way to include these heuristic indices into our Lagrangian approximation of the intelligence management problem would be to replace the computationally expensive $E_u V_t\left(\gamma+1, \delta+\log\left(\frac{\beta+u}{\beta}\right)\right)$, (for reasonably large t at least) and replace it with either $I_t^{CG}(\gamma, \delta)$ or $I_t^{BL}(\gamma, \delta)$.

The immediate issue with doing this is that neither of these indices incorporates the hesitiation constant C, so using them in this way would mean that estimates of the fair sampling charge W would not approximate those based on $r(\gamma, \delta, C)$. The desire to find the highest quality, top quartile items would be diminished by this. Replacing $r(\gamma, \delta)$ with say, $I_t^{BL}(\gamma, \delta)$ would completely remove this desirable aspect of the model.

However if one were to use a hybrid approach, where the immediate expected rewards at least took the hesitation constant C into account, then we could use the approximation

$$\tilde{W} = r(\gamma, \delta) + \frac{1}{n}\sum_{i=1}^{j^*}\left(I_t^{BL}(\gamma+1, \delta_{[i]}) - \hat{W}\right), \tag{6.66}$$

where

$$j^* = \max\left\{j : 1, \ldots, n : I_t^{BL}(\gamma, \delta_{[j]}) \geq \bar{r}(\gamma, \delta) + \frac{1}{n}\sum_{i=1}^{n}\max\left\{\bar{I}_t^{BL}(\gamma, \delta_{[i]}) - I_t^{BL}(\gamma, \delta_{[j]}), 0\right\}\right\}. \tag{6.67}$$

This has the benefit of including the one-step rewards in a way which is true to

the model and also giving weight to the more variable sources in a way which is very cheap computationally. There is no guarantee that this will be robust as the hybrid isn't likely to have the same order of magnitude. It may be better to fully commit to using the heuristic indices and instead use

$$\tilde{W} = I_t^{BL}(\gamma, \delta) + \frac{1}{n} \sum_{i=1}^{j^*} \left(I_t^{BL}(\gamma + 1, \delta_{[i]}) - \hat{W} \right), \tag{6.68}$$

which would solve the robustness issue but would abandon a key attribute of the model by disregarding the importance of C entirely. This issue could be worked around by incorporating $r(\gamma, \delta)$ into a hybrid index at each stage so that

$$I_t^H(\gamma, \delta) = r(\gamma, \delta) \left(\frac{\gamma}{\delta} + \frac{z_t \sqrt{\gamma}}{\sqrt{\delta^2 + \delta^3}}, \right) \tag{6.69}$$

which takes a similar approach to that used in [Glazebrook et al., 2012] where the authors added problem-specific features to $I_t^{BL}(\gamma, \delta)$ when considering an assortment problem. We could then use (6.68) with I_t^H, although doing so would render the computation of $\tilde{W}$ at least as computationally expensive as evaluating the fair charge for $T = 2$. Whether one can avoid or reduce the amount of computations related to this (via omission or discretisation of the (γ, δ) space) and still maintain a robust, near-optimal decision policy is yet to be seen and further work on this subject would be required.

Chapter 7

Conclusions and future considerations

The largest component of the work carried out in this document relates to the multi-armed bandit allocation problem (MABA) which was introduced in Chapter 4 and then developed into numerical studies in subsequent chapters. Chapter 5 pursued a discrete Dirichlet-multinomial model and Chapter 6 focused on a continuous Exponential-Gamma-Gamma formulation, which also incorporated the feature of the processor's judgement uncertainty.

Overall, this thesis has managed to develop some robust methodologies to attack a collection of problems usually not tackled in the literature and that are of importance to those in the intelligence community. The contributions to solving the discrete MABA here have already provided solutions based on numerical testing. This thesis has also explored possible avenues for attack in the continuous case as well as incorporating the operational concern of processor uncertainty.

7.1 Discrete MABA problem

For the discete MABA model, a wide range of numerical experiments were conducted to test the relative performance level of various item allocation policies. Where the Lagrangian policy was tested, it performed second only to the super-

optimal 'perfect information' policy.

The optimistic Bayes sampling method and knowledge gradient methods also performed well very well, with optimistic Bayes faring better than knowledge gradient across the range of numerical experiments that were carried out in Chapter 5. The standard Knowledge gradient typically outperformed any variation of the policy which involved capping the look ahead multiplier on future expected rewards, including the greedy policy. The Thompson sampling method outperformed the greedy policy but less well than knowledge gradient.

Although the Lagrangian policy has shown to be the most effective in the numerical studies that it has featured in, it should be noted that the range of examples considered was limited to those in which the value of C was set to be $N-1$ to decrease the computational cost of running numerical studies. Expanding the work done here so that more reward states per source can be considered would be a priority subject for further study of the Lagrangian allocation policy in the discrete MABA model.

The robustness of the thresholding policies for allocated sampled items was sufficiently good to yield usable results, and the dynamic C policy outperformed static C in the numerical study, but both thresholding variants could be improved upon. Investigating ways to tune the static C policy more efficiently and without using the mean number of allocated items in the unconstrained case would certainly be an avenue for further work. With dynamic C, simply giving it an extensive range of test cases would help to uncover under which circumstances it performs less well so that any policy redesign can then take place.

The judgement uncertainty feature included in the continuous MABA model is one which has been described as desirable by intelligence professionals during informal discussions and it is also called for in [Friedman and Zeckhauser, 2012] and [Kaplan, 2012]. A priority for future work on the discrete MABA model in this document would be to find a suitable way to incorporate this feature into the discrete MABA work and evaluate how policies perform there.

7.2 Continuous MABA problem

The main issues arising when formulating the continuous MABA model are those of the complexity of the model itself and of selecting appropriate problem settings to fully test the capabilities of various allocation policies. The Exponential-Gamma-Gamma structure is not as operationally transparent as the discrete model.

The analytical framework that has been built to incorporate the judgement uncertainty feature has come at the cost of not being able to attribute a meaningful absolute importance value to intelligence items. Rather we must compare them to a chosen threshold C. One is not able to set $C = 0$ in the continuous formulation as was the case in the discrete model.

Any mean Bayes returns obtained are hard to compare to other results obtained under a different value of C. Using a common value of C across many experiments is a poor solution to this problem, as the choice of C plays a key part in achieving the best results for any given policy and could bias experiments in favour of certain policies. However, it may be possible to first use a set of threshold values for the purposes of allocation decisions, unique to each individual policy. We would then separately use a common threshold across all policies for the purposes of scoring and guaranteeing comparable results.

The policies' relative performances across the pilot study indicate that the Thompson sampling policy is best suited to the continuous MABA model so far. Knowledge gradient type policies seem to be largely unaffected by the size of the look-ahead cap value placed on expected future rewards, and the Optimistic Bayes sampling policy as it stands is inconsistent. There is currently no adequate super-optimal policy formulated for this problem that performs consistently well.

The pilot study for the continuous MABA model is limited in scope and further numerical work is required if more meaningful insights are to be gained. The framework itself may also need to be redesigned to be more user-friendly to non-technicians.

A Lagrangian reformulation of the continuous MABA model has been provided

as an analogue to the discrete Lagrangian model in Chapter 5. However, at this time there is no computationally tractable way to produce numerical studies in a similar fashion to those carried out elsewhere. The computational shortcut used in the discrete case has no natural analogue in the continuous setting.

Bibliography

J Ahmadi, M Doostparast, and T Parsian. Bayes estimation based on random censored data for some life time models under symmetric and asymmetric loss functions. *Communications in Statistics*, 39(17):pp. 3058–3071, August 2010.

S. Asmussen and Glynn P.W. *Stochastic Simulation: Algorithms and Analysis*. Springer, 2007.

M Babaioff, N Immorlica, D Kempe, and R Kleinberg. A knapsack secretary problem with applications. In M Charikar, K Jansen, O Reingold, and J. D. P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques: Lecture Notes in Computer Science*, pages 16–28. Springer, Berlin Heidelberg, 2007. URL `http://dx.doi.org/10.1007/978-3-540-74208-1_2`.

J Bather. Randomised allocation of treatments in sequential trials. *Advances in Applied Probability*, 12(1):pp. 174–182, 1980. ISSN 00018678. URL `http://www.jstor.org/stable/1426500`.

R Bellman. A problem in the sequential design of experiments. *Sankhya: The Indian Journal of Statistics Vol*, 16(3):pp. 221–229, 1956.

R. Bellman. A markovian decision process. *Journal of Mathematics and Mechanics*, (6), 1957a.

R. Bellman. *Dynamic Programming*. Rand Corporation research study. Princeton

University Press, 1957b. ISBN 9780691079516. URL `http://books.google.it/books?id=wdtoPwAACAAJ`.

R. Bellman. *Dynamic Programming*. Dover Publications, 2003.

P Beraldi, A Violi, N Scordino, and N. Sorrentino. Short-term electricity procurement: A rolling horizon stochastic programming approach. *Applied Mathematical Modelling*, 35(8):pp. 3980–3990, 2011.

J M Bernardo and A F M Smith. *Bayesian Theory*. Wiley, 1994.

D. Berry and B. Fristedt. *Bandit Problems: Sequential Allocation of Experiments. Monographs on Statistics and Applied Probability Series*. Chapman and Hall, 1985.

D Bertsimas and J Niño Mora. Restless bandits, linear programming relaxations, and a primal-dual index heuristic. *Oper. Res.*, 48(1):80–90, January 2000. ISSN 0030-364X. doi: 10.1287/opre.48.1.80.12444. URL `http://dx.doi.org/10.1287/opre.48.1.80.12444`.

D. Bertsimas, I. C. Paschalidis, and J. N. Tsitsiklis. Branching bandits and Klimov's problem: achievable region and side constraints. *IEEE Transactions on Automatic Control*, 40(12):2063–2075, Dec 1995. ISSN 0018-9286. doi: 10.1109/9.478231.

M. Brezzi and L. L. Lai. Optimal learning and experimentation in bandit problems. *Journal of Economic Dynamics and Control*, 27(1):pp. 87–108, 2002.

G C. Brown, W M. Carlyle, R C. Harney, E M. Skroch, and Wood R K. Interdicting a nuclear-weapons project. *Operations Research*, 57(4):pp. 866–877, 2009.

F. Caro and J. Gallien. Dynamic assortment with demand learning for seasonal consumer goods. *Management Science*, 53(2):pp. 276–292, 2007.

K.C. Cheong. Survey of investigation into the missile allocation problem. 1985. URL `www.dtic.mil/dtic/tr/fulltext/u2/a159385.pdf`.

P. F. Christoffersen and F. X. Diebold. Optimal prediction under asymmetric loss. Working Paper 167, National Bureau of Economic Research, October 1994. URL `http://www.nber.org/papers/t0167`.

Y Costica. Optimizing classification in intelligence processing. 2010. URL `http://hdl.handle.net/10945/4986`.

A K. Cronin. Behind the curve: Globalisation and international terrorism. *International Security*, 27(3):pp. 30–58, 2002.

M. DeGroot. *Optimal Statistical Decisions*. McGraw-Hill, New York, 1970.

L. Devroye. *Non-Uniform Random Variate Generation*. 1986.

D R. Ellis. Algorithms for efficient intelligence collection. 2013. URL `http://hdl.handle.net/10945/37621`.

H. Everett. Generalized lagrange multiplier method for solving problems of optimum allocation of resources. *Operations Research*, 11(3):399–417, 1963. ISSN 0030364X, 15265463. URL `http://www.jstor.org/stable/168028`.

P. I. Frazier, W. B. Powell, and S. Dayanik. A knowledge-gradient policy for sequential information collection. *SIAM Journal on Control and Optimization*, 47(5):2410–2439, 2008. doi: 10.1137/070693424. URL `http://dx.doi.org/10.1137/070693424`.

J.A. Friedman and R. Zeckhauser. Assessing uncertainty in intelligence. *Intelligence and National Security*, 27(6):pp. 824–847, 2012.

D. Gamerman. *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference*. Chapman-Hall/CRC, 1997.

J. Gittins, K. Glazebrook, and R. Weber. *Multi-armed bandit allocation indices*. John Wiley and Sons, 2011.

J C. Gittins. Bandit processes and dynamic allocation indices. *Journal of the Royal Statistics Society*, 41(2):pp. 148–177, 1979.

J C Gittins and D M Jones. A dynamic allocation index for the discounted multi-armed bandit problem. *Biometrika*, 66(3):pp. 561–565, 1979.

J.C. Gittins and D.M. Jones. A dynamic allocation index for the sequential design of experiments. In J. Gani, editor, *Progress in Statistics*, pages 241–266. North-Holland, Amsterdam, NL, 1974.

K.D. Glazebrook, J. Meissner, and J. Schurr. How big should my store be? on the interplay between shelf-space, demand learning and assortment decisions. *Working Papers*, 2012.

B. Gluss. An optimum policy for detecting a fault in a complex system. *Operations Research*, 7(4):468–477, 1959. doi: 10.1287/opre.7.4.468. URL `http://dx.doi.org/10.1287/opre.7.4.468`.

S. S. Gupta and K. J. Miescke. Sequential selection procedures–a decision theoretic approach. *Ann. Statist.*, 12(1):336–350, 03 1984. doi: 10.1214/aos/1176346411. URL `http://dx.doi.org/10.1214/aos/1176346411`.

S. Hougardy and S. Kirchner. Lower bounds for the relative greedy algorithm for approximating steiner trees. *Networks*, 47(2):pp. 111–115, 2006.

C I Hsu, H C Li, Liu S M, and C C. Chao. Aircraft replacement scheduling: A dynamic approach. *Transportation Research Part E*, 47(1):pp. 41–60, 2011.

A. Ibrahim and A. S. Alfa. Using lagrangian relaxation for radio resource allocation in high altitude platforms. *IEEE Transactions on Wireless Communications*, 14 (10):5823–5835, Oct 2015.

P. Jacko and S. S. Villar. Opportunistic schedulers for optimal scheduling of flows in wireless systems with ARQ feedback. In *Teletraffic Congress (ITC 24), 2012 24th International*, pages 1–8, Sept 2012.

P Kall and S W Wallace. *Stochastic Programming*. John Wiley and Sons, 2 edition, 1994.

E S. Kaplan. Terror queues. *Operations Research*, 58(4):pp. 773–784, 2010.

E S. Kaplan. OR forum - intelligence operations research - the 2010 Philip McCord Morse Lecture. *Operations Research*, 2012.

S. Karlin. Dynamic inventory policy with varying stochastic demands. *Management Science*, 6(3):pp. 231–258, 1960.

D. Kohler. Optimal strategies for the game of darts. *The Journal of the Operational Research Society*, 33(10):871–884, 1982. ISSN 01605682, 14769360. URL `http://www.jstor.org/stable/2580993`.

O Kwon, D Kang, K Lee, and S Park. Lagrangian relaxation approach to the targeting problem. *Naval Research Logistics (NRL)*, 46(6):640–653, 1999. ISSN 1520-6750. doi: 10.1002/(SICI)1520-6750(199909)46:6⟨640::AID-NAV3⟩3.0.CO;2-Q. URL `http://dx.doi.org/10.1002/(SICI)1520-6750(199909)46:6<640::AID-NAV3>3.0.CO;2-Q`.

J.L. Lagrange. *Mecanique Analytique*. 1811.

Dong Li and K D. Glazebrook. A Bayesian approach to the triage problem with imperfect classification. *European Journal of Operational Research*, 215(1): pp. 169–180, November 2011. URL `http://ideas.repec.org/a/eee/ejores/v215y2011i1p169-180.html`.

A. A. Lopez-Toledo. A controlled markov chain model for nursing homes. *SIMULATION*, 27(5):161–169, 1976. doi: 10.1177/003754977602700505. URL `http://sim.sagepub.com/content/27/5/161.abstract`.

David W. Low. Optimal dynamic pricing policies for an m/m/s queue. *Operations Research*, 22(3):545–561, 1974. ISSN 0030364X, 15265463. URL `http://www.jstor.org/stable/169504`.

B C. May, N Korda, A Lee, and D S. Leslie. Optimistic Bayesian sampling in contextual-bandit problems. *J. Mach. Learn. Res.*, 13:2069–2106, June 2012. ISSN 1532-4435. URL `http://dl.acm.org/citation.cfm?id=2188385.2343711`.

J F. McCloskey. OR forum - British Operations Research in World War II. *Operations Research*, 35(3):pp. 453–470, 1987a.

J F. McCloskey. US Operations Research in World WarII. *Operations Research*, 35(6):pp. 910–925, 1987b.

S. P. Meyn and R. L. Tweedie. *Markov Chains and Stochastic Stability*. Springer-Verlag, 1993.

S J Moura, H K Fathy, D S Callaway, and J L. Stein. A stochastic optimal control approach for power management in plug-in hybrid electric vehicles. *IEEE transactions on control systems technology*, 19(3):pp. 545–555, 2011.

Y Nevo. Information selection in intelligence processing. 2011. URL `http://hdl.handle.net/10945/10660`.

José Nio-Mora. Computing a classic index for finite-horizon bandits. *INFORMS Journal on Computing*, 23(2):254–267, 2011. doi: 10.1287/ijoc.1100.0398. URL `http://dx.doi.org/10.1287/ijoc.1100.0398`.

I Parpucea, B Parv, and T Socaciu. Modeling uncertainty in a decision problem by externalising information. *Int. J. of Computers, Communications and Control*, 6(2):pp. 328–336, June 2011.

W.B. Powell. *Approximate Dynamic Programming, Solving the curses of dimensionality*. 2011.

G. E. Pugh. Lagrange multipliers and the optimal allocation of defense resources. *Operations Research*, 12(4):543–567, 1964. ISSN 0030364X, 15265463. URL `http://www.jstor.org/stable/167702`.

D.L. Puterman. *Markov Decision Processes*. John Wiley Sons, New York, 1994.

D. B. Rosenfield, R. D. Shapiro, and D. A. Butler. Optimal strategies for selling an asset. *Management Science*, 29(9):1051–1061, 1983. ISSN 00251909, 15265501. URL `http://www.jstor.org/stable/2630932`.

S M. Ross. *Introduction to Stochastic Dynamic Programming*. Elsevier, 1983. ISBN 0125984219.

I O Ryzhov and W B Powell. The value of information in multi-armed bandits with exponentially distributed rewards. *International Conference on Computational Science*, 2011.

I. O. Ryzhov, W. B. Powell, and P. I. Frazier. The knowledge gradient algorithm for a general class of online learning problems. *Operations Research*, 60(1):180–195, 2012. doi: 10.1287/opre.1110.0999. URL `http://dx.doi.org/10.1287/opre.1110.0999`.

I. O. Ryzhov, W. B. Powell, and P. I. Frazier. The knowledge gradient algorithm for a general class of online learning problems. *Operations Research*, 60(1):pp. 180–195, January/February 2012.

P I Ryzhov, I O. Frazier and W B Powell. On the robustness of a one-period look-ahead policy in multi-armed bandit problems. *International Conference on Computational Science*, pages pp. 180–195, 2010.

E. Seufert. *Freemium Economics, Leveraging Analytics and User Segmentation to Drive Revenue*. 2014.

J A. Swets. Measuring the accuracy of diagnostic systems. *Science*, 240(4857):pp. 1285–1293, 1988.

R. W. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3):pp. 285–294, 1933.

M Tovik. Adaptive ε-greedy exploration in reinforcement learning based on value differences. In *Annual Conference on Artificial Intelligence*, pages 203–210. Springer, 2010.

A F. Veinott. Bandit processes and dynamic allocation indices. *Journal of the Royal Statistics Society*, 40(5):pp. 1635–1660, 1979.

D. J. White. A survey of applications of markov decision processes. *Journal of the Operations Research Society*, 44(11):pp. 1073–1096, 1933.

P. Whittle. Restless bandits: Activity allocation in a changing world. *Journal of Applied Probability*, 25:287–298, 1988. ISSN 00219002. URL `http://www.jstor.org/stable/3214163`.

P Zlatsin, 2013. URL `https://calhoun.nps.edu/handle/10945/37751`.