Evaluation of the Intelligence Collection and Analysis Process

Livia Zsuzsanna Stark, B.Sc.(Hons.), M.Res

Lancaster University

STOR-i

excellence with impact

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

August 2022

Abstract

Intelligence is a critical tool in modern security operations that provides insight into current and future operational conditions. It is a concept that transfers to other applications where monitoring activities or situations is imperative, such as ecological research. As technological advances in the past decades lead to increased availability of potential intelligence, we concentrate on source selection to ensure the resulting intelligence is of high quality and fit for purpose. We wish to bring focus to the more varied nature of intelligence than what is currently reflected in models of its collection and evaluation. Therefore, we examine the intelligence collection and analysis process in two separate scenarios; one treats it as a ongoing strategic activity, in another intelligence collection is carried out with an investigative intent.

The first problem we formulate concerns source selection with a random time delay in feedback, corresponding to the collection and evaluation time of the intelligence. Both the distributions of such time delay and the outcome of the intelligence evaluation are unknown, giving rise to the classic exploration-exploitation dilemma in a long-run setting. We develop promising approaches to accommodate the novel features of the model based on Gittins indices and the knowledge gradient, and examine the issues presented when incorporating structures of dependence between the time delay and the outcome of the evaluation.

Next, we develop a novel intelligence collection problem rooted in tactical level source selection, aiming to piece together an intelligence picture comprised of multiple types of information, for example, where and when an attack is planned. We demonstrate that when all elements of the model are known, dynamic programming provides the optimal policy. When some elements are unknown, which introduces an exploration-exploitation aspect to the model, we find that in certain cases the ability to learn is severely limited.

Acknowledgements

First and foremost I would like to thank STOR-i and all its staff over the years for the opportunity to do a PhD. I know if it was not for STOR-i I would not have done so. A special thank you goes to Jon Tawn, Idris Eckley and Kevin Glazebrook for creating this unique community of students, and steering it through the unprecedented times of the pandemic. STOR-i would not be the same without its people and the cohort system, and so I would like to give a shout-out to the Tigersharks, the best year group STOR-i has ever seen.

I must also express my gratitude towards my supervisors, Kevin Glazebrook, Peter Jacko and Michael Atkinson for their unrelenting faith in me and my abilities. On top of that, I must thank Kevin for being an amazing guide in Monterey, California, and introducing me to its natural and gastronomic wonders. While they are not as good as those you can get in the Monterey Bay area, I have added both seafood chowder and cioppino to my repertoire. To Peter I owe thanks for being an understanding ear during hard times, and especially for making time to address my rising panic as my submission date grew closer. Without his and Mike's support during this last month I would without doubt be still writing.

While not part of the original plan, I must thank STOR-i and its matchmaking service for recruiting Stan. He has been my rock in the past few years and provided immense support (and good meals) during the writing up stage. Finally, I must thank my parents for enabling me to take this path in life. Without their encouragement I would not be who I am and where I am today, and for that I will forever be grateful.

Declaration

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

Livia Zsuzsanna Stark

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The rapid technological advancement of the past few decades has significantly enhanced intelligence collection capabilities, making a large amount of data available. For example: in 2010 the National Security Agency collected roughly two billion communications per day (Kaplan, 2012). However, data in its raw form carries little meaning and further resources must be expended to produce intelligence from it, through what has been termed as the *intelligence cycle*.

Kaplan (2012) describes five components of the intelligence cycle: planning, collection, processing, analysis, and dissemination. Examples of collected data include tips from informants, intercepted emails, and satellite images. The processing and analysis components allow the intelligence team to determine the quality of the items. Examples of processing include translating an email and image enhancement. The resulting information is ”made sense of” in the following analysis phase, where the the intelligence team ultimately determines the value of the item, producing intelligence. However, a sizeable proportion of data collected is discarded without ever being examined due to insufficient manpower and resources, leading to the loss of potentially valuable intelligence. To improve efficiency and reduce such loss, collection efforts should concentrate on high quality intelligence sources that are reliable and provide relevant intelligence.

Often, the quality of intelligence sources is initially unknown, only to be determined through analysing the incoming items. This makes the decision about which sources to utilise challenging. The same limitations on resources such as time and manpower that are responsible for the discarded items mean that only a small number of intelligence items from various sources can be analysed. Hence the intelligence team must balance exploration and exploitation in choosing which sources to use, as the intelligence team needs to explore many sources to generate reasonable estimates of source quality. In terms of the intelligence cycle, our research covers the collection, processing, and analysis phases, gathered under the term *evaluation*, with a particular focus on collection: which source should the intelligence team query for its next piece of intelligence? In this thesis we devise approaches to guide the intelligence team in how to choose among intelligence sources of different but unknown quality. The quality of sources can be established by sampling from the sources one by one, learning about them while trying to find relevant intelligence, making this a sequential decision problem.

1.2 Models of Intelligence Collection and Analysis

There is limited work in the non-confidential literature on mathematical models of intelligence collection, processing, and analysis, Most of the relevant literature fall under one of two streams of research: one by Kaplan and collaborators (Kaplan, 2010; Kaplan and Feinstein, 2012; Seidl et al., 2016; Wrzaczek et al., 2017) and the other out of the Naval Postgraduate School, primarily lead by Szechtman (Costica, 2010; Marshall, 2016; Tekin, 2016).

The Kaplan research lines examine intelligence collection and processing at macro level: plots are hatched, detected, investigated, and interdicted at known rates. In the original model, information regarding potential terror plots queue for analysis in a role equivalent to that of a customer and intelligence agents are treated as the servers of such a terror queue (Kaplan, 2010). The intelligence agents successfully complete a service if they are able to investigate, infiltrate, and interdict the terror plot. Attacks are detected and interdicted at constant, known rates. While in the queue waiting for service, plots may be successfully executed which is equivalent to the customer leaving the queue. There are several follow-on studies to the original terror queue paper. Seidl et al. (2016) examines optimal control for how many agents should perform detection (against undetected plots) versus interdiction (against detected plots). The work of Wrzaczek et al. (2017) pushes this concept further by formulating a game where the terror group chooses its attack rate and the government chooses its staffing level. The extension in Kaplan and Feinstein (2012), which also uses a queue to track plots (customers) and intelligence analysts (servers) focuses on the trade-off between collecting more raw intelligence versus the processing and analysis of that raw intelligence to guide action to thwart attacks.

In these papers the intelligence collection and vetting of sources is not modelled explicitly, though the underlying mechanism by which agents detect and interdict plots is presumably very similar to our focus. In that sense our work is complimentary, as we examine the micro level of how agents actually execute their investigation: examining intelligence from many different sources of varying and potentially unknown quality.

The work out of the Naval Postgraduate School also examines the micro level. Costica (2010) uses a queue to examine the arrival of incoming tips, some of which are relevant and some are worthless. An initial processing server determines whether each tip should be sent to the analysis server or discarded. The processing server makes both false positives and false negatives. The primary metric of interest is the rate of relevant tips evaluated by the analysts. However, this model does not consider separate intelligence sources and assumes all parameters are known. The theses of Marshall (2016) and Tekin (2016) started a stream of research led by Szechtman with a focus similar to ours. Marshall (2016) and Tekin (2016) have multiple intelligence sources of unknown qualities. Each period the decision maker obtains intelligence items called tips from a subset of the sources, and they must consider the trade-off between exploration and exploitation to obtain the largest cumulative number of relevant tips over time. Carmeli (2021) assumes each piece of intelligence has a value associated with it and the value distribution depends upon the source and is unknown. Collectors must spend resources to obtain intelligence each period and desire to obtain high value intelligence. If the resources spent are insufficient, the intelligence obtained is worthless. Dimitrov et al. (2016) considers an investigation of a network (e.g., phone calls), where the collector examines one edge per period. The likelihood an edge contains relevant intelligence and the value of each node are unknown and the distributions for these quantities update as the investigation proceeds. Kronzilber (2017) also considers a network but from a cyber perspective: an attacker can either gather information from a computer it has access to or attack another computer in the network. The likelihood of obtaining relevant information from each computer is unknown.

These papers focus on screening sources of intelligence for relevant or high value items, with two overarching assumptions, each presenting a gap in the models available. The first gap identified is that intelligence is collected at discrete, evenly spaced periods. However, this does not account for inherent variation in the required resources, for example time, to distill the data gathered into intelligence. While Marshall (2016) allows the processor to make multiple observations of the intelligence item before reaching a decision, the time spent on an item is controlled by the decision maker. Similarly, the resources allocated to a source are controlled by the intelligence team in Carmeli (2021) with a constant budget for every period. We are not aware of any literature on intelligence collection that addresses stochasticity in either the time or other resources required to obtain and assign a value to an intelligence item.

The second assumption present in the cited research is that only one type of intelligence exists, and individual items only differ from each other by their values. This is a too significant aspect to ignore, as in an operational setting intelligence items may possess attributes not captured by their value, which could mean they cannot be treated equivalent. Examples of possible attributes include the type of information they carry (*who* is planning an attack and *where*), or the intelligence question they pertain to (i.e. surveillance on different organisations). Such a feature has not made an appearance in the study of intelligence collection or processing to the best of our knowledge.

While not the primary focus of such models, intelligence plays a role in a variety of problems. For example, Kress and MacKay (2014) includes intelligence in a Lanchester combat model that allows one side to better aim their fire when they obtain better intelligence. The search problem in Atkinson et al. (2016) also incorporates the concept of intelligence. Every piece of intelligence provides information to a searcher who may use the intelligence received so far to take action, or decide to wait for further intelligence.

1.3 Contributions

In this section we briefly outline the structure of this thesis and take the opportunity to highlight the contributions made.

Chapter 2 provides an overview of the relevant theoretical background. We introduce Markov and semi-Markov decision processes, bandit processes and multi-armed bandits, with dynamic programming, Gittins indices and the knowledge gradient as potential solution approaches.

Chapter 3 contains the first research problem of this thesis, which we termed the *intelligence problem*. Our main contributions in this chapter are:

- Addressing one of the gaps in modelling approaches currently present in the literature as discussed in Section 1.2, by explicitly modelling the time required to obtain and analyse an intelligence item.

- Developing two novel heuristic policies to address the new features of the model based on the ideas underlying the Gittins index, namely the Expected Generalised Gittins Index (EGGI) and the Calibrated Knowledge Gradient Index (CKGI).

In our setting, both the relevance of the intelligence item and the time required to determine it is a characteristic of the source of the item, and both are quantities which we permit to be random with an initially unknown distribution. Incorporating a time delay so that rewards associated with obtaining a relevant piece of intelligence are only received once such determination is made, bringing our model even closer to a real-life intelligence collection and analysis process. We consider two variants: one where the quality of an item is independent from its evaluation time, the other where dependence is present.

In Chapter 4 we develop the problem of the *intelligence puzzle*, which is a novel intelligence collection problem featuring multiple types of intelligence. Our most important contributions are:

- Proposal of a completely new problem, which captures a previously ignored aspect of intelligence collection, namely the availability of distinct types of information.
- Initial exploration of the new problem, with the decision makers having access to both perfect and imperfect information on the problem parameters.
- Establishing the optimal policy when decision makers possess perfect information, and developing the Expected Completion Cost (ECC) policy, a heuristic designed for when only imperfect information is available.

The intelligence puzzle is motivated by intelligence collection on the tactical level, in which case intelligence is gathered with the purpose of investigating a specific intelligence question in mind. The intelligence team must examine intelligence items from a selection of sources in search of one of each required types to complete the puzzle. However, both the occurrence and the relevance of the intelligence items are random, and their distributions depend on which source they originate from. As the intelligence puzzle corresponds to an open intelligence question, completion of the puzzle is time sensitive and must be done as quickly as possible. As it is a novel problem, we first examine a version in which the distribution of both the occurrence and the relevance of the types are known, followed by variants in which the distribution of either is only learnt through repeated observation.

In Section 5.1 we summarise our achievements and has some concluding remarks. Finally, Section 5.2 we presents potential avenues of future work for both the intelligence problem and the intelligence puzzle, including some extensions which may be applied to both models.

Chapter 2

Methodologies

The overarching methodological theme of this thesis is to model problems inspired by the intelligence collection and analysis process as Markov and semi-Markov decision processes, and as multi-armed bandits. In Chapter 2 we set out the relevant modelling and solution approaches that we consequently employ in later chapters. The main sources that informed this discussion are Puterman (1994) for Section 2.1 and Section 2.2, and Gittins et al. (2011) for Section 2.3 and Section 2.4. The thought experiment we introduce in Section 2.5 is well laid out with an abundance of additional discussion in Powell and Ryzhov (2012).

2.1 Markov and Semi-Markov Decision Processes

Markov Decision Processes

A *Markov decision process*, or MDP for short, is a type of sequential decision-making process. Properties and solution approaches to MDPs have been studied at large scale, being the standard approach to model not only academically, but industrially interesting problems. A survey of applications was presented by White (1993), and since then the variety of settings it has been studied in has grown.

We describe the key elements of such a decision problem in the same order as encountered in Puterman (1994). First of all, we must define when decisions take place.

These points in time are referred to as *decision epochs*, and are denoted as e. In the setting of Markov decision processes the set of decision epochs, denoted as $\mathcal{S}_e$, is discrete and its elements spaced evenly so that we may say without loss of generality that the time between decision epochs is 1. In that case $\mathcal{S}_e$ is written as

$$\mathcal{S}_e = \{0, 1, ..., H\} \qquad\qquad H \leq \infty, \tag{2.1.1}$$

where H denotes the horizon of the decision problem. If $H = \infty$, the decision problem is said to be of an infinite horizon with no clear end. Otherwise, it is a finite horizon problem of H periods where the last decision is made at decision epoch $H - 1$. The decision problem terminates at decision epoch H: despite referring to it as such, no decision takes place and is only included to record the final state of the system. This is the same as the convention used by Puterman (1994). Note that when not stated otherwise, by decision epoch e we mean decision epoch $e \leq H - 1$.

The next element of the decision problem is the aforementioned system *state*, denoted as $s \in \mathcal{S}_s$ where $\mathcal{S}_s$ is the set of states the system may occupy. Depending on the application, these states may represent conditions of a physical system or knowledge regarding the system. At every decision epoch e the decision maker is faced with a choice. Given that the state of the system is s at decision epoch e, they may choose to take any *action* $a \in \mathcal{S}_a(s)$, where $\mathcal{S}_a(s)$ is the set of permitted actions is state s. As a consequence of choosing action a in state s at decision epoch e, the system transitions to state s' with *transition probability* $\mathbb{P}(s' \mid s, a)$ and the decision maker receives a reward. While such a reward may take many forms, it must not be affected by actions taken at later decision epochs and either its value or expected value, denoted as $r(s, a)$, must be known before the action is chosen. In a finite horizon setting a reward might also be received at the last decision epoch H for the decision process terminating in state s. We denote the expectation of such a terminal reward as $r(s)$.

There may exist a preference towards receiving rewards earlier rather than later in the process, which can be expressed via the use of discounting. The *discount factor*

$0 < d < 1$, measures the value provided at decision epoch $e + 1$ by what is 1 unit of reward at the current decision epoch e. It is most commonly present in infinite horizon processes, but can also form part of the decision process in the finite horizon case.

What sets a Markov decision process apart from a more general, discrete-time sequential decision process is that the available set of actions, and the rewards and transition probabilities associated with them are independent of past actions taken, states visited, and rewards received. The history of the decision process only affects these through the state the system currently occupies.

A Markov decision process may be alternatively defined, building it up from Markov chains and Markov reward processes. In that case a Markov decision process is a type of Markov reward process where the transition probabilities can be influenced through the actions taken. We do not wish to delve too deep into the details of the above. For a detailed examination of Markov chains and Markov reward processes we refer the reader to Tijms (2003).

A *decision rule* determines which action to take in each state of a given decision epoch e. A *policy* π is a sequence of decision rules which prescribes the action to take at all decision epochs: it provides a decision rule for all possible combinations of state and decision epoch. A Markov decision rule only depends on the history of the process through the current decision epoch e and system state s_e. A generalisation of a Markov policy is a history dependent one, in which the state and decision epoch alone is not enough to dictate which action a policy prescribes. We also wish to distinguish between deterministic and randomised policies. As the name suggests randomised policies contain random elements, while deterministic ones do not. Our primary focus will be on the class of deterministic Markov policies, which we denote as Π^{MD}. Implementing a policy produces a random sequence of rewards of length H, one for each decision epoch when an action was made, or of $H + 1$ if a terminal reward is present. The decision maker wishes to apply the policy which results in the highest valued sequence of rewards and therefore they need to be able to assign a value to these reward sequences. In the finite horizon case, the reward sequences tend to be compared based on the *expected total reward* of applying policy π, defined as

$$\mathbb{E}_{\pi,s}\left[\sum_{e=0}^{H-1} r\left(s_e, \pi(s_e)\right) + r_H\left(s_H\right)\right], \tag{2.1.2}$$

which may be adapted to include the discount factor as necessary. In the infinite horizon case of the problem the decision problem produces an infinite, random reward stream. Sometimes the expected total reward we get from letting H approach ∞ in (2.1.2) is still an appropriate measure, but such approach often leads to assigning a value to the reward stream that does not converge, possibly taking on a value of $-\infty$ or ∞. To be able to compare different reward streams, we need a finite-valued measure to compare them by. In the case of undiscounted rewards, it is typical to define the value of the reward stream as the *expected average reward* of policy π, given as

$$\lim_{H\to\infty}\frac{1}{H}\mathbb{E}_{\pi,s}\left[\sum_{e=0}^{H-1} r\left(s_e, \pi(s_e)\right)\right]. \tag{2.1.3}$$

In the presence of a discount factor the value of the reward stream is given by the *expected total discounted reward* of policy π, namely

$$\mathbb{E}_{\pi,s}\left[\sum_{e=0}^{\infty} d^e r\left(s_e, \pi(s_e)\right)\right]. \tag{2.1.4}$$

Note that while neither the limit in (2.1.3) or (2.1.4) is guaranteed to converge, they do converge under quite general assumptions such as discrete decision epochs, finite set of states and finite set of actions. As the decision problems at the focus of this thesis are of the undiscounted variety, here and in following sections discounted decision problems are only briefly discussed. For more details see Puterman (1994).

The *optimality criterion* states the condition a policy π must satisfy to be considered optimal, which is to produce the highest valued reward sequence or stream with respect to a model-appropriate measure of value. Following the convention of Puterman (1994), designate a Markov decision process coupled with an optimality criterion, as a Markov decision problem. We also refer to the optimality criterion as the *objective* of the decision maker. While we have introduced the decision problem in terms of maximising rewards, it is trivial to adapt it to $r(s, a)$ representing a cost. We can do so by either defining costs as negative rewards so that $r(s, a) \leq 0$, alternatively, the objective is to be redefined to minimise a sequence of positive costs.

Semi-Markov Decision Processes

Semi-Markov decision processes (SMDP) generalise Markov decision processes to a continuous-time setting, where the decision epochs occur at random according to a probability distribution, and may be, but are not necessarily triggered by a change in the state of the system. Such a decision problem offers more flexibility than an MDP, and is used in queueing (Afanasyeva et al., 2012) and in modelling financial risks (Janssen and Manca, 2007) to name a few.

The elements of the semi-Markov decision process are the same as that of a Markov decision process, but adapted to the continuous-time setting. Note that in some special cases semi-Markov decision processes can be reformulated as Markov decision processes. The set of decision epochs is defined as a collection of random points in time, and the time elapsed between two consecutive decision epochs e and $e + 1$ is referred to as the *sojourn time* T_e. While the set of system states is defined similarly as it is for Markov decision processes, a state must be defined for any given time, not just at the decision epochs. The action a taken in a given state s jointly controls the probability of the system occupying state s' at the next decision epoch $\mathbb{P}(s' \mid s, a)$ and the probability distribution of the associated sojourn time. In a continuous time setting there are two types of rewards the decision maker can receive. They may receive a lump-sum reward in response to taking action a in state s which is analogous to the rewards defined for MDPs and denoted as $k(s, a)$. Rewards may also be accrued continuously at rate $w(s''_t, s, a)$ for the process occupying state s''_t at time t following action a in state s. The inclusion of s''_t is to keep the formulation general as in semi-Markov models multiple transitions may occur between decision epochs. However, the reward rate depending on either s''_t, s, or a is not a necessity. Then the reward associated with taking action

a in state s is written as

$$R(s,a) = k(s,a) + \int_{t_e}^{t_{e+1}} w(s''_t, s, a)\mathrm{d}t, \tag{2.1.5}$$

where t_e is the time of decision epoch e. Note that this reward is a random quantity as both s''_t and t_e are random.

The semi-Markov qualifier is used for such a continuous-time decision process because the sojourn times are random, and the distribution of sojourn times and the transition probabilities as defined above only depend on the history of the process through the current state and action chosen at the current decision epoch.

The reward sequence or reward stream of a semi-Markov decision process is also a random quantity, and therefore we need to define what we mean by the value of such sequence. Finite horizon semi-Markov decision processes are much less studied (Mamer, 1986; Huang and Guo, 2011) than those with an infinite horizon for which Baykal-Gűrsoy (2011) provides an overview.

The optimality criterion of the semi-Markov decision problem is then defined analogously to that of the Markov decision problem. We consider two optimality criteria for infinite horizon semi-Markov decision processes, the first of which is the one favoured by Puterman (1994), the *ratio average reward* criterion. It uses a reward measure formulated as the ratio of the expected sum of rewards and the expected sum of sojourn times, giving rise to an optimality criterion of the form

$$\max_{\pi} \liminf_{H\to\infty} \frac{\mathbb{E}_{\pi,s}\left[\sum_{e=0}^{H} R_e\left(s_e, \pi(s_e)\right)\right]}{\mathbb{E}_{\pi,s}\left[\sum_{e=0}^{H} T_e\right]}. \tag{2.1.6}$$

The second optimality criterion we include here quantifies the reward stream in terms of the *long-run average reward* of policy π, and so is defined as

$$\max_{\pi} \liminf_{\mathcal{H}\to\infty} \frac{1}{\mathcal{H}} \mathbb{E}_{\pi,s}\left[\sum_{e=0}^{e_{\mathcal{H}}-1} k(s_e, \pi(s_e)) + \int_0^{\mathcal{H}} w(s''_t, s_{e_t}, \pi(s_{e_t}))\mathrm{d}t\right]. \tag{2.1.7}$$

Note that in the infinite horizon limit $\mathcal{H} \to \infty$ the number of decision epochs up to $\mathcal{H}$, denoted as $e_{\mathcal{H}}$ also goes to infinity. While the reward measure used here is similar in form to $\sum_{e=0}^{e_{\mathcal{H}}-1} R(s_e, \pi(s_e))$, the upper limit of the integral stretches to $\mathcal{H}$ instead of the time that corresponds to $e_{\mathcal{H}} - 1$. The optimality criteria in (2.1.6) defines the horizon in terms of how many decision epochs take place, similarly to discrete time Markov processes. At the same time (2.1.7) interprets it as a point in real time, $\mathcal{H}$. Under some circumstances the two decision criteria agree, although that is not the case in general.

We omit discussion of discounted semi-Markov decision problems here, but refer the reader to Chapter 11 of Puterman (1994), as it offers a detailed examination of semi-Markov decision problems.

2.2 The Bellman Equation and Backward Induction

In this section we discuss finding the optimal policy for some of the optimality criteria set out in Section 2.1, with primary focus on Markov decision problems.

We preface this discussion by stating a well established result without proof, which is that for Markov and semi-Markov decision problems the optimal policy will belong to the class of deterministic Markov policies (Puterman, 1994). This allows us to introduce the *optimality equations*, otherwise known as *Bellman equations* (Bellman, 1957), in the framework of deterministic Markov policies without having to examine a more general case first.

The Optimality Equations

First of all, we consider a finite-horizon Markov decision problem with an optimality criterion that requires the optimal policy to maximise the expected total reward over the horizon, calculated via (2.1.2). Then a deterministic Markov policy $\pi \in \Pi^{\mathrm{MD}}$ is considered optimal in state s_e, if starting from state s_e in decision epoch e it achieves the highest expected total reward out of all other policies. Since we only care about the optimality of π from decision epoch e onwards, it does not matter how the policy performed at earlier decision epochs. This leads to the idea of constructing the optimal policy of the entire decision problem $\pi^* = \pi_0^*$ by finding the optimal policy at decision epoch $H - 1$, namely π_{H-1}^*, and incrementally expanding it to include earlier decision epochs of the problem in a manner that preserves the optimality of the policy. The above line of thought is formalised in the Bellman equations, which are the series of H equations written as

$$\mathbb{V}\left(e, s_e\right) = \max_{a \in \mathcal{S}_a} \left\{\mathbb{V}^a\left(e, s_e\right)\right\}, \qquad e = \{0, ..., H-1\} \tag{2.2.1}$$

where $\mathbb{V}\left(e, s_e\right)$ is referred to as the *value function* of state s_e at decision epoch e, and the expression the equations maximise over

$$\mathbb{V}^a\left(e, s_e\right) = r(s_e, a) + \sum_{s' \in \mathcal{S}_s} \mathbb{P}\left(s' \mid s_e, a\right) \mathbb{V}\left(e+1, s'\right), \tag{2.2.2}$$

is called the *value function of action a* in state s_e at epoch e. We must combine the above with the boundary condition of the decision problem, which is determined by the terminal value of the final state s_H and is written as

$$\mathbb{V}\left(H, s_H\right) = r_H(s_H). \tag{2.2.3}$$

The equations in (2.2.1) together with (2.2.3) can be used to find the value function at any decision epoch e for any state s_e the system may occupy at that time, which is the expected total reward of the optimal policy from e onward. The value function of the starting state $\mathbb{V}\left(0, s_0\right)$ is solved for in the same manner to obtain the expected total reward of the decision problem with starting state s_0 under the optimal policy π^*. Furthermore, any policy π that achieves $\mathbb{V}\left(0, s_0\right)$ is optimal, and therefore the Bellman equations can also be used to determine the optimality of a given policy.

Backward Induction

Backward induction (Puterman, 1994) is an algorithm based on the optimality equations that is commonly used to efficiently solve finite-horizon discrete time Markov decision problems. The term ”dynamic programming” is often used synonymously to backward induction, but it may be used to refer to methods and results for sequential decision problems in general. Let us denote by $\mathcal{S}_{s_e}$ the set of states the system may occupy at decision epoch e, and by $\mathcal{S}_{a_{s_e}}$ the set of actions available at decision epoch e in state s_e. We set out the backward induction algorithm as follows.

The Backward Induction Algorithm

1. Start at $e = H$ and record

$$\mathbb{V}\left(H, s_H\right) = r_H(s_H) \qquad\qquad \text{for all } s_H \in \mathcal{S}_{s_H},$$

2. Move on to the previous decision epoch $e := e - 1$.

   Compute

$$\mathbb{V}^a\left(e, s_e\right) = r(s_e, a) + \sum_{s' \in \mathcal{S}_s} \mathbb{P}\left(s' \mid s_e, a\right) \mathbb{V}\left(e+1, s'\right) \qquad \text{for all } s_e \in \mathcal{S}_{s_e}.$$

   Record

$$\begin{aligned} \mathbb{V}\left(e, s_e\right) &= \max_{a \in \mathcal{S}_a} \left\{\mathbb{V}^a\left(e, s_e\right)\right\}, \\ a^*\left(e, s_e\right) &= \arg\max_{a \in \mathcal{S}_a} \left\{\mathbb{V}^a\left(e, s_e\right)\right\}. \end{aligned}$$

3. If $e = 0$, stop.

   Otherwise, repeat step 2.

Upon completion, the backward induction algorithm will have determined and recorded the value function of all states $s \in \mathcal{S}_s$ and the optimal policy π^* as the collection of optimal actions $a^*(e, s)$.

The Bellman equations are straightforward to adapt to the discounted total reward criterion over an infinite horizon. Using the expanded form of $\mathbb{V}^a\left(e, s_e\right)$ they take the form

$$\mathbb{V}\left(e, s_e\right) = \max_{a \in \mathcal{S}_a} \left\{ r(s_e, a) + \sum_{s' \in \mathcal{S}_s} d\ \mathbb{P}\left(s' \mid s_e, a\right) \mathbb{V}\left(e+1, s'\right) \right\}. \tag{2.2.4}$$

However, when faced with an infinite horizon MDP, backward induction is not suitable. One notable exception is when the infinite horizon problem can be truncated to that of a finite horizon in a way that the expected total reward of the decision problems are comparable. When such approach is not appropriate, there are alternatives in the form of the value iteration and policy iteration algorithms. We do not include a discussion of those here, instead redirecting those interested to Puterman (1994) for details. That, specifically Chapter 11 is also where an in depth discussion on the optimality equations of semi-Markov bandits can be found. Here we only mention that while one can analogously define them with discounted rewards over infinite horizon, in general there are differences present.

2.3 Bandit Processes and Multi-Armed Bandits

In this section we introduce bandit processes and multi-armed bandits. Here and in Section 2.4 we often refer to (Gittins et al., 2011) as a resource that covers the topics in both sections.

Families of Alternative Bandit Processes

A *bandit process* is a type of semi-Markov decision process in which at every decision epoch e there are two actions, often referred to as *controls*, available to the decision maker. The effect of applying control 1, also known as *continuation* control, is to allow the underlying reward process in state s to transition to its next state s' with transition probability $\mathbb{P}(s' \mid s)$ and receive the associated reward $r(s)$. This reward may be discounted with discount factor d. We must note the probability that the sojourn time is no more than t depends on the transition that takes place and is denoted as $\mathbb{P}(T \leq t \mid s, s')$. Applying control 0 is equivalent to *freezing* the bandit. For the duration control 0 is in place no transitions occur and no reward is received. However, every point in time becomes a decision epoch where the decision maker can opt to carry on with freezing the bandit or apply the continuation control.

As the bandit process may spend some time frozen, we distinguish the time the process was under the continuation control as the *process time*. Then the sequence of process times when the continuation control is applied, along the sequence of states at those times is called the *realisation* of the bandit process. It is important to note that such a realisation of any bandit process b is independent of the sequence of controls that was applied, as control 0 only freezes the process. In a *Markov bandit process* the above still applies, except that under the continuation control all sojourn times are equal and therefore can be set to 1.

By *family of alternative bandit processes* (FABP), which is an old fashioned but in our opinion apt term, we refer to a decision process defined as a collection of N independent bandit processes. Starting from decision epoch $e = 0$ only one of the N bandit processes is chosen to apply the continuation control to, and the rest are frozen until the next decision epoch of the continued bandit process is reached. This practice of applying control 1 to one of the bandits while applying control 0 to the rest until the continued bandit arrives at its next decision epoch is repeated for the duration of the decision process. The frozen arms do not influence the one that is being continued, and vice versa.

The set of available actions for a FABP that consists of N arms is a subset of $\mathcal{S}_a = \{1, ..., N\}$, where taking action $a \in \mathcal{S}_a$ is equivalent to continuing bandit a and freezing all others. If at all decision epochs the set of available actions is the entirety of the set $\mathcal{S}_a$ it is a *simple family of alternative bandit processes* (SFABP). The state of the system s can be expressed as a vector of the states of the individual bandits so that $\boldsymbol{s} = (s^1, ..., s^N)$, and therefore the state space of the multi-armed bandit is the product of their state space, $\mathcal{S}_s = \prod_{a=1}^{N} \mathcal{S}_{s^a}$.

Being a semi-Markov decision process, the decision epochs of a semi-Markov FABP occur at random points in time so that the sojourn time following decision epoch e is a random variable T_e. Considering the definition of FABP, if at decision epoch e bandit a was chosen to continue, then $T_e = T^a(s_e)$ which is the sojourn time of bandit process a in state s_e. For Markov bandits the question of decision epochs is much simpler. Since the sojourn time of all continued Markov bandit processes are 1, the set of decision epochs of Markov FABP are discrete, $e = 0, 1, ...$

Note that while a Markov policy for Markov or semi-Markov FABP prescribes the bandit to continue at each decision epoch e based on the complete state of the decision problem at that decision epoch s_e, the transition probabilities, rewards, and in the case of semi-Markov FABPs the following sojourn time only depend on the bandit process a chosen for continuation and its state at that particular decision epoch s^a_e. As we will see in Section 2.4, this feature of the decision problem can be exploited to great effect in devising a solution approach. Same as in the decision problems in Section 2.1, applying policy π to a FABP produces a sequence of rewards which can be compared to those of other policies through the performance measure (*payoff*) used to formulate the objective of the decision problem.

Multi-Armed Bandits

A large area of interest in bandit literature is that of *multi-armed bandits*, MAB for short, which is an *online learning* problem. A multi-armed bandit is a SFABP in which the expected value of the reward received for continuing bandit a is unknown, and the distribution of said reward may only be learnt from repeatedly continuing bandit process a and observing the rewards received. In the context of multi-armed bandits we may refer to the constituent bandits as the arms of the multi-armed bandit, and the action of continuing arm a as pulling or playing arm a. These terms derive from the original motivating example of the model, a series of fruit machines (one armed bandit) with initially unknown rewards over which a gambler wishes to maximise their earnings.

In such a decision problem the states of the different arms are *information* or *knowledge states*, which capture the knowledge the decision maker possesses regarding the distribution of possible rewards. The reward received for pulling any given arm is not only a reward, but also provides information regarding the reward distribution of said arm. For that reason we may use the expression ”make an observation on arm a” synonymously to pulling an arm. The phrase online learning refers to the fact that the decision maker must learn the reward distributions simultaneously to accumulating the rewards which they wish to maximise, as opposed to *offline learning*, in which rewards are only gathered after an initial period focused on learning. The online nature of learning in multi-armed bandits requires a balance to be struck between exploring the arms to obtain reliable estimates of their expected rewards and exploiting the arm believed to have the highest expected reward. While the terms exploration and exploitation have been long in use, learning and earning has been increasing in popularity to refer to the same concept.

Depending on if we are discussing semi-Markov or discrete-time multi-armed bandits they are still either a semi-Markov or a discrete-time Markov decision problem, and as such the Bellman equations can be used to produce an optimal policy. However, due to the so called curse of dimensionality such approach becomes increasingly less helpful the more arms a multi-armed bandit possesses. For that reason other options are sought. Under some circumstances heuristic rules approach optimality (Kelly, 1981), but that is not the case in general.

Two well established bandit algorithms are *Thompson sampling* and the *Upper Confidence Bound* (UCB) approach. Thompson Sampling was introduced in Thompson (1933). It generates a sample of the unknown parameters from a posterior distribution at each decision epoch and then chooses the action that maximizes the expected reward, assuming the sampled values are the true parameter values. The Upper Confidence Bound approach (Auer et al., 2002) chooses an action in each decision epoch that maximizes the sum of the current expected reward with an extra bounding term. This bounding term increases with the number of decision epochs and decreases with the number of times the particular action is chosen. UCB usually performs well by ensuring a sufficient amount of exploration is done while still primarily making exploitative choices. We do not go into further detail regarding either Thompson sampling or UCB, as these methods are not the primary focus of our thesis. For more details on Thompson sampling see Chapter 36 of Lattimore and Szepesvári (2020), and part II of the same for UCB.

The solution approaches that are relevant to the research in this thesis are discussed in the remaining sections of this chapter. In Section 2.4 we consider the *Gittins index policy*, which is optimal for infinite horizon discounted SFABPs and may serve as a heuristic in other cases. Another method is the *knowledge gradient* algorithm, which can be applied to a wide variety of learning problems both online and offline. We discuss knowledge gradient in Section 2.5.

2.4 The Gittins Index

The Gittins index theorem was first published in Gittins and Jones (1974) then Gittins (1979), developed not for multi-armed bandits, but for a different bandit process. In this section we use discrete-time infinite horizon discounted SFABPs to introduce the concept Gittins indices. Note that the same ideas can be extended for semi-Markov infinite horizon discounted SFABPs. For more details on that, see Section 2.8 of Gittins et al. (2011).

First, we define an *index policy* as a policy in which an index $\nu(a, s^a) \in \mathbb{R}$ is assigned to each bandit dependent only on the current state s^a of said bandit and the bandit a with the biggest index is chosen to continue. Such an index policy is optimal for SFABPs, and may be calculated based on a comparison with a so called *standard bandit*. In the discrete time setting a standard bandit process is guaranteed to produce a reward of λ every decision time it has been chosen for continuation. Let us construct a new SFABP for every bandit process in the original SFABP made up of a standard bandit and the bandit of interest. For the remainder of the section we focus on the two-bandit SFABP created for bandit a. If from the start only the standard bandit is continued, the expected discounted total reward over the infinite horizon is $\frac{\lambda}{1-d}$. Presume bandit a was continued at decision epoch $e = 0$, after which the bandit to continue is determined by the optimal policy. Imagine, that according to the optimal policy at decision epoch e^* the standard bandit must be continued. In that case, at e^*+1 the state of the two bandit SFABP is identical to what it was as the standard bandit only has one state, and bandit a has been frozen and therefore remained in the state it was in at e^*. Consequently, if it was optimal to continue the standard bandit at e^* it will also be the optimal action at e^*+1, and following the same logic it must be for all $e \geq e^*$. Then the expected discounted total reward of the optimal policy is

$$\sup_{e^*>0} \mathbb{E}\left[\sum_{e=0}^{e^*-1} d^e r(s_e^a, a) + d^{e^*} \frac{\lambda}{1-d} \mid s_0^a = s^a\right]. \tag{2.4.1}$$

Note that while the superscript a signals that s_e^a is the state of bandit a, d^e is to be read as d to the power of e.

The next step in finding the Gittins index is why this method of deriving it is called the calibration approach, as we wish to find the value of λ for which we are indifferent between continuing either bandit with both actions optimal. By equating the payoffs of the two options and rearranging it to standard form we get

$$0 = \sup_{e^*>0} \mathbb{E}\left[\sum_{e=0}^{e^*-1} d^e r(s_e^a, a) + \left(1 - d^{e^*}\right) \frac{\lambda}{1-d} \mid s_0^a = s^a\right] \tag{2.4.2}$$

which has one unique root

$$\lambda = (1-d) \sup_{e^*>0} \frac{\mathbb{E}\left[\sum_{e=0}^{e^*-1} d^e r(s_e^a, a) \mid s_0^a = s^a\right]}{1 - \mathbb{E}\left[d^{e^*} \mid s_0^a = s^a\right]} = \sup_{e^*>0} \frac{\mathbb{E}\left[\sum_{e=0}^{e^*-1} d^e r(s_e^a, a) \mid s_0^a = s^a\right]}{\mathbb{E}\left[\sum_{e=0}^{e^*-1} d^e \mid s_0^a = s^a\right]}. \tag{2.4.3}$$

This unique root $\lambda = \nu(a, s^a)$ is the Gittins index of bandit process a. Then the Gittins index policy chooses to continue at every decision time e the bandit with the largest index given by $\nu(a, s_e^a)$, which is the optimal policy for a discrete-time SFABPs. Proof of the above and further examination of the Gittins index is found in Gittins et al. (2011). Use of the Gittins index drastically reduces the computational complexity of the decision problem as it allows the decision maker to evaluate and compare the constituent bandits separately instead of having to consider the whole SFABP.

In some special cases, such as Beta-Bernoulli MABs a good approximation of the Gittins index can be obtained via dynamic programming. For that, we need to approximate the infinite horizon problem with a finite horizon one. For a comprehensive survey on finite horizon Beta-Bernoulli bandits we direct the reader to Jacko (2019). Due to the effect of discounting, if the finite horizon is sufficiently large it has an expected cumulative reward comparable to that of the infinite horizon problem, and backward induction can be used to find the Gittins index via the above calibration approach.

2.5 The Knowledge Gradient

The knowledge gradient method was initially proposed in Gupta and Miescke (1996) then further developed in Frazier et al. (2008) for a different learning problem, namely ranking and selection. It was shortly after adapted to multi-armed bandits by Ryzhov et al. (2012). We introduce the underlying idea using a thought experiment in terms of a discrete time undiscounted finite horizon MAB.

Imagine the following scenario in which we wish to control a discrete-time undiscounted multi-armed bandit with N arms. In total we may make H observations corresponding to having a finite horizon H to maximise the observed rewards over. At epoch e the MAB is in state $s_e = (s_e^1, ..., s_e^N)$ after learning from the e observations made so far. The expected reward of each arm a is given by $r(s_e^a, a)$. In this thought experiment we become unable to update our beliefs after n observations. Without learning, the knowledge states remain the same and therefore the MAB occupies state s_n with expected rewards $r(s_n^a, a)$ for all decision epochs $n \leq e \leq H$ corresponding to the remaining $H - n + 1$ observations. With learning infeasible, attention must be turned to exploiting the information already gathered through continuing the arm we believe best till the horizon is reached. Then the expected reward from $e = n$ onwards is

$$(H - n + 1) \max_{a \in \mathcal{S}_a} \mathbb{E} \left[r_n(s_n^a, a) \right]. \tag{2.5.1}$$

Now let us alter the thought experiment in a way that while we may still learn from the n+1th observation made at $e = n$, learning ceases at $e = n$+1. Similarly to the previous setting, the expected reward from $e = n+1$ onward is $(H - n) \max_{a \in \mathcal{S}_a} r_{n+1}(s_{n+1}^a, a)$. Since at decision epoch $e = n$ we still have one more observation available to us before learning stops, the expected total reward of continuing arm a at decision epoch $e = n$ is the combined reward of observing arm a at $e = n$ and the rewards accumulated from the decision epochs $n + 1 \leq e \leq H$. We write it as

$$\mathbb{E}\left[r_n(s_n^a, a)\right] + (H - n) \max_{a' \in \mathcal{S}_{a'}} \mathbb{E}\left[r_{n+1}(s_{n+1}^{a'}, a') \mid s_n, a_e = a\right]. \qquad (2.5.2)$$

To maximise the expected reward from $e = n$ onward we must choose the arm a that maximises (2.5.2).

The knowledge gradient policy uses the thought process from the above thought experiment to select which arm to play at every decision epoch e. It regards every decision epoch e to be the last opportunity to gather further information on the reward distributions of the arms, then uses

$$\underset{a \in \mathcal{S}_a}{\arg\max} \mathbb{E}\left[r(s_e^a, a)\right] + (H - e) \max_{a' \in \mathcal{S}_{a'}} \mathbb{E}\left[r_{e+1}(s_{e+1}^{a'}, a') \mid s_e, a\right] \qquad (2.5.3)$$

to select the arm which contributes the most, either by virtue of its high expected reward or via knowledge the observation provides. Note that this is not an index policy as the states of the other arms are necessary to calculate the second half of the sum.

While the Gittins index simplifies the MAB by considering the arms separately, in essence the knowledge gradient breaks them into a series of one step learning problems. For some versions of the MAB problem the knowledge gradient policy has been shown to be optimal (Ryzhov et al., 2012). However, Edwards et al. (2017) demonstrated its weak performance in a Beta-Bernoulli setting.

Chapter 3

The Intelligence Problem with Delays

3.1 Introducing the Intelligence Problem

The first contribution in this thesis concerns what we dub the *intelligence problem*, which falls under the multi-armed bandit paradigm, but with the addition of non-trivial complications, namely delayed reward observations and binding time commitment. Our model is set out as follows. An intelligence team has access to a number of sources. Each source generates a stream of intelligence items, which we call *tips*. The intelligence team only has enough resources to collect and analyse one tip from one source at time. At each decision point, one source is chosen and that source produces a tip. The tip is evaluated as either *relevant* or *nuisance*. While in reality the usefulness of a tip may be better captured as a continuum, this work views them as binary, which is consistent with much of the literature (Costica (2010); Marshall (2016); Tekin (2016)). Intuitively, even in the presence of continuous values some items are clearly too low quality to be useful, with the opposite also true. By treating intelligence items in a binary fashion, only the nuance for mid-value intelligence items is lost. There are no false positives or false negatives. Examples of sources include informants, phone taps, monitoring chat rooms, Intelligence, Surveillance, and Reconnaissance (ISR) assets (e.g., UAV), and surveillance of a suspect.

There may be a delay between when a source is chosen to when the tip is collected, which we call the *collection time*. For example if the intelligence team decides to monitor a phone tap, a chat room, or surveil a suspect's house, it will take some time before something interesting occurs that qualifies as a tip. After the tip is obtained, there will be further processing and analysis time to determine the result of the tip: relevant or nuisance. This additional time analysing a tip is the *analysis time*. For example, a suspicious phone conversation about a meeting might require additional investigation to determine if it is relevant to the investigation. We combine the collection time and analysis time into one entity that encapsulates all temporal delays between the selection of a source and the evaluation result and call this the *evaluation time*. Once the intelligence team has chosen a source, they must complete the evaluation of the tip obtained from it. By picking one of the sources they commit to waiting the full evaluation time to observe the results and receive the associated reward. Only after that are they able to switch to a different source. While prior work has focused on the quality of sources, very little research on the intelligence cycle has incorporated this time delay. Explicit modelling of the evaluation time and the potential delays and lulls involved in the intelligence cycle is the most significant contribution of this work.

The intelligence team desires not just high quality sources, but sources that generate intelligence that moves quickly from the start of collection through to the end of analysis. As the quality of sources may initially be unknown, the team needs to balance exploitation and exploration. The team must spend effort using all available sources to obtain accurate estimates of the relevance and evaluation times of the tips belonging to each of the sources (exploration) to ensure it utilizes the best sources. Once the intelligence team is reasonably confident in the quality of sources, they should exploit the highest quality sources to generate the most accurate intelligence picture.

While we focus on the intelligence scenario, our model applies for any application where a decision maker must choose among several entities that produce items of varying quality at different frequencies. An ecology scenario would involve the placement of sensors (e.g., cameras, observers, traps) at various locations to monitor wildlife. There may be some activity of interest (e.g., mating, hunting, childcare) that the researchers want to monitor.

The remainder of this chapter consists of four parts. First, as a precursor to tackling the intelligence problem, in Section 3.2 we consider multiple models which could be used to describe the intelligence problem. We consider these models in the simplest scenario in which all relevant parameters are known, then gradually introduce layers of stochasticity to them. The main body of this chapter consists of two parts. Section 3.3 delves into a variant of the intelligence problem in which the relevance and evaluation time of tips are independent of each other, while Section 3.4 considers the case in which dependence is present.

3.2 Modelling of the Intelligence Problem

In order to proceed, we need to precisely define what the exact problem of interest faced by the intelligence team is. We need to specify the objective of the problem and the metrics of interest, as well as the known and unknown quantities present in the model. Depending on the underlying scenarios and assumptions different metrics are appropriate. In this section we discuss our assumptions and come up with our choice of metric to use in Section 3.3 and Section 3.4. Throughout, we model the sources of intelligence as a multi-arm bandit, which is suitable for sources that are independent of each other. We recognise that this is a major assumption and may not always apply. However, it is necessary to gain an understanding of the problem. Table 3.2.1 shows correspondence between typical bandit terminology and the elements of the intelligence problem.

An intelligence problem with $\mathcal{A}$ available sources is modelled as a semi-Markov multi-armed bandit with $\mathcal{A}$ arms, where each arm corresponds to a source. The outcomes of tips from arm a, namely X^a, $a \in \{1, ..., \mathcal{A}\}$, form a sequence of i.i.d. Bernoulli random

Intelligence	**Bandits**
source	arm
tip	outcome
relevant tip	success outcome
nuisance tip	failure outcome
relevance probability	success probability
evaluation time	inter-decision time

Table 3.2.1: Terminology equivalence of the intelligence and bandit problems

variables, where a success $X^a = 1$ represents a tip deemed relevant and a failure $X^a = 0$ a nuisance tip. Each tip also has an associated evaluation time, which depends on the arm chosen. These evaluation times determine the sojourn times of the process, and the timing of the decision epochs. The system state of the multi-armed bandit describes the knowledge of the decision maker regarding the success probabilities and evaluation times of the arms.

There are several modelling choices for the rewards received and the objective the decision maker might wish to optimise against, each motivated by different real life intelligence problems. They are described in more detail further on in this section.

IP1. Obtain as many relevant tips as possible in the long run.

IP2. Obtain as many relevant tips as possible in the long run, with preference to do so early on.

IP3. Obtain as many relevant tips as possible before a given deadline. There may or may not be a preference towards finding relevant tips early on.

IP4. Obtain a good rate of relevant tips per unit resource consumed.

While we spend little time considering IP2 and IP3, we include them in this discussion for completeness and to demonstrate the range of modelling options that was available to us.

We first consider these under a simplified scenario, in which all the probabilities of observing a relevant tip are known and the associated evaluation times are deterministic rather than random. From there, we progressively add layers of randomness to the problem to work up to the scenario in which both the success probabilities and the parameters governing the evaluation times are unknown. These cases are summarised in Table 3.2.2. Throughout these investigations all previous assumptions of the problem, such as evaluating tips one at a time, binary outcomes, delayed rewards, and not being able to renege on a chosen source remain.

	Tip Relevance	**Evaluation Time**
Case 1	Random, known success probability	Deterministic
Case 2	Random, unknown success probability	Deterministic
Case 3	Random, unknown success probability	Random, known distribution
Case 4	Random, unknown success probability	Random, unknown distribution

Table 3.2.2: Nature of the relevance probabilities and the evaluation times considered in the cases of Section 3.2.

Case 1

In the first case we examine, the probability that a tip from arm a is relevant is known and given by p^a, so that $X^a \sim \text{Bernoulli}(p^a)$. Likewise, the associated evaluation time, which we denote as t^a is also known. The state of any arm a is given by the decision maker's knowledge of said arm, $s^a = (p^a, t^a)$, which are both known quantities as mentioned before and therefore remain the same throughout the problem. Since the arms are independent, the state of the multi-armed bandit as a whole is given by $s = (s^1, ..., s^{\mathcal{A}})$. Note that as a consequence of the individual arms only occupying one state throughout the problem the system is also limited to one state only.

At every decision epoch e an arm a is selected to continue, which is equivalent to sampling a tip from source a to evaluate. Regardless of the arm chosen, only self transitions occur, with a probability of $\mathbb{P}(s \mid s, a) = 1$. There is no continuously accrued reward gained from playing arm a, however, there is a lump-sum reward based on the expected outcome of the evaluation.

While all four intelligence problems share some similarities, they need different modelling approaches. As sources may have different evaluation times and therefore in general $t^a \neq t^{a'}$, for all $a \neq a'$, in the case of intelligence problems IP1-IP3 the time between decisions is not constant. This necessitates their formulation as a semi-Markov multi-armed bandit. Since the t^a's are known, the sojourn time T_e that follows decision epoch e is directly determined by the arm a chosen as $\mathbb{P}\left(T_e = t^a \mid a\right) = 1$. When formulating the reward structure of IP1.-IP3., we find that for all three cases the lump-sum reward is the expected outcome of the evaluation received upon its completion,

$$k\left(s, a\right) = p^a. \tag{3.2.1}$$

However, their associated optimality criteria differ.

IP1. In the case of IP1, the problem description suggests we are in an undiscounted infinite horizon regime. In a semi-Markov setting this leads to two possible objective criteria, one in terms of the long-run average reward (2.1.7), the other in terms of the ratio average reward (2.1.6).

Let us first consider the long-run average reward criterion. For the reward in (3.2.1) it simplifies to

$$\max_{\pi} \lim_{\mathcal{H}\to\infty} \frac{1}{\mathcal{H}} \mathbb{E}_{\pi,s}\left[\sum_{e=0}^{e_{\mathcal{H}}-1} k(s_e, \pi(s_e))\right] = \max_{\pi} \lim_{\mathcal{H}\to\infty} \frac{1}{\mathcal{H}} \mathbb{E}_{\pi}\left[\sum_{e=0}^{e_{\mathcal{H}}-1} p^{\pi(s_e)}\right]. \tag{3.2.2}$$

Since in all deterministic Markov policies the action prescribed only depends on the state of the process of which there is only one here $s_e = s \; \forall e \in \mathcal{S}_e$, any such policy would select an arm a at the beginning of the decision problem and continue playing that arm throughout. Given that, we can rewrite the objective as

$$\max_{a\in\mathcal{S}_a}\lim_{\mathcal{H}\to\infty}\frac{1}{\mathcal{H}}\mathbb{E}_\pi\left[\sum_{e=0}^{e_{\mathcal{H}}-1}p^a\right]=\max_{a\in\mathcal{S}_a}\lim_{\mathcal{H}\to\infty}\frac{e_{\mathcal{H}}p^a}{\mathcal{H}}=\max_{a\in\mathcal{S}_a}\frac{p^a}{t^a},\tag{3.2.3}$$

where we noted that the number of observations made if arm a was continued from the beginning till the end of the horizon is $e_{\mathcal{H}}=\mathcal{H}/t^a$. What (3.2.3) shows is that the policy which chooses the arm a with the highest ratio of p^a/t^a, a quantity we refer to as the *reward rate* of that arm, is optimal.

The ratio average reward criterion is adapted as follows. Then

$$\max_{\pi}\liminf_{H\to\infty}\frac{\mathbb{E}_{\pi,s}\left[\sum_{e=0}^{H-1}k(s_e,\pi(s_e))\right]}{\mathbb{E}_{\pi,s}\left[\sum_{e=0}^{H-1}T_e\right]}=\max_{\pi}\lim_{H\to\infty}\frac{\mathbb{E}_\pi\left[\sum_{e=0}^{H-1}p^{\pi(s_e)}\right]}{\mathbb{E}_\pi\left[\sum_{e=0}^{H-1}t^{\pi(s_e)}\right]},\tag{3.2.4}$$

and after invoking the argument based on only considering deterministic Markov policies we find that the objective is of the form

$$\max_{a\in\mathcal{S}_a}\lim_{H\to\infty}\frac{\sum_{e=0}^{H-1}p^a}{\sum_{e=0}^{H-1}t^a}=\max_{a\in\mathcal{S}_a}\lim_{H\to\infty}\frac{(H)p^a}{(H)t^a}=\max_{a\in\mathcal{S}_a}\frac{p^a}{t^a},\tag{3.2.5}$$

which is the same policy we found to be optimal with respect to the long-run average reward criterion.

IP2. The intelligence problem described in IP2 is that of a discounted infinite horizon decision problem, for which the expected discounted total reward criterion is most suitable. Discounting in a semi-Markov setting was not discussed in detail in Chapter 2, and for reasons similar to those stated there, we will not consider the corresponding optimality criterion further.

IP3. The third potential definition of the intelligence problem suggests a finite horizon decision problem either with or without discounting. We only consider the undiscounted variety of this intelligence problem, but it may be adapted to discounting in a similar manner to IP2. Over a finite horizon $\mathcal{H}$ the objective is to find the policy which maximises the total expected reward over said horizon. In the reward structure defined, that can be written as

$$\mathbb{E}_{\pi,s}\left[\sum_{e=0}^{e_{\mathcal{H}}-1} k(s_e,\pi(s_e))\right] = \mathbb{E}_{\pi}\left[\sum_{e=0}^{e_{\mathcal{H}}-1} p^{\pi(s_e)}\right]. \tag{3.2.6}$$

While the evaluation times of the tips do not explicitly feature in the expected total reward, alongside the horizon they constrain how many evaluations may take place. Then the objective criterion is

$$\max_{\pi} \mathbb{E}_{\pi}\left[\sum_{e=0}^{e_{\mathcal{H}}-1} p^{\pi(s_e)}\right] \tag{3.2.7}$$

with the implicit constraint that for every policy π and every realisation of the decision problem

$$\sum_{e=0}^{e_{\mathcal{H}}-1} t^{\pi(s_e)} \leq \mathcal{H}. \tag{3.2.8}$$

The equations (3.2.7) and (3.2.8) in essence describe a knapsack constrained bandit problem (Tran-Thanh et al., 2012; Graczová and Jacko, 2014).

IP4. The last intelligence problem in our list of examples stands apart as we can formulate it as a discrete time decision problem instead of a semi-Markov one, where the evaluation times no longer dictate when the decision epochs occur, instead $e \in \{0, 1, ..., H-1\}$. We achieve this by disconnecting the the evaluation times of the tips from the decision epochs, treating them as usage of some kind of resource. In this reading of the problem an infinite horizon, such as the one present in IP1, would be translated to an infinite budget which we wish to utilise wisely, despite being infinite. Then the time spent on evaluating tips is the equivalent of the intelligence team exchanging some amount of resource for an opportunity to observe a tip, wishing to obtain high rewards for the resource invested.

The rewards are still only received upon completion of the evaluation process, set to coincide with the next decision epoch so that the reward from decision epoch e only materialises when decision epoch $e+1$ is triggered. To capture the resources spent on obtaining the tip, the reward of an arm is given by the utilisation rate of the time-resource of the chosen arm, which for arm a in state s is

$$r(s,a) = \mathbb{E}_s\left[\frac{X^a}{t^a}\right] = \frac{p^a}{t^a}, \tag{3.2.9}$$

which is received upon completion of the evaluation. High rewards are achieved by arms that either have high success probabilities, short evaluation times or both. If (but only if) the problem is over an infinite horizon, we expect such a model to be equivalent to IP1. With an infinite horizon and no discounting we find the long-run average reward criterion to be the most suitable, taking the form

$$\max_{\pi} \lim_{H\to\infty} \frac{1}{H}\mathbb{E}_{\pi}\left[\sum_{e=0}^{H-1} r(s_e, \pi(s_e))\right] = \max_{\pi} \lim_{H\to\infty} \frac{1}{H}\mathbb{E}_{\pi}\left[\sum_{e=0}^{H-1} \frac{p^{\pi(s_e)}}{t^{\pi(s_e)}}\right]. \tag{3.2.10}$$

Since we expect all policies to continually play only one of the arms a it simplifies to

$$\max_{a\in\mathcal{S}_a} \lim_{H\to\infty} \frac{1}{H-1}\sum_{e=0}^{H} \frac{p^a}{t^a} = \max_{a\in\mathcal{S}_a} \frac{p^a}{t^a}. \tag{3.2.11}$$

which is the same as the optimality criterion of IP1.

While the intelligence problems IP2 and IP3 lead to interesting model formulations, for the remainder of the cases we focus on IP1 and IP4.

Case 2

Next, we deliberate a version of the intelligence problem in which the evaluation time associated with arm a remains a known quantity for all $a \in \mathcal{S}_a$, but the success probabilities of the arms are unknown, and are modelled as random variables P^a. To incorporate our initial beliefs and facilitate Bayesian learning of P^a we place a prior on it which is updated with the outcomes of the evaluated tips. In such a case the knowledge available to the decision maker is summarised by the collection of posterior parameters, denoted as $\boldsymbol{\rho}^a$ and the values of the t^a. Then the state of the system in any epoch e is given by $s = (s^1, ..., s^{\mathcal{A}})$ where $s^a = (\boldsymbol{\rho}^a, t^a)$. However, as all t^a's remain constant through the problem, one might not even consider them part of the state. While everything regarding the sojourn times and the spacing of the decision epochs is the same as in the previous case, the system is no longer restricted to one state only and it transitions to a new state in response to evaluating a tip from arm a. Let s_{a^+} and s_{a^-} denote the state of the system after a tip sampled from arm a in system state s evaluated to a relevant and nuisance tip respectively. Then the transition probabilities from state s given arm a is chosen are

$$\mathbb{P}\left(s_{a^+} \mid s, a\right) = \mathbb{P}\left(X^a = 1 \mid \boldsymbol{\rho}^a\right) = \mathbb{E}_s\left[P^a\right] \tag{3.2.12}$$

$$\mathbb{P}\left(s_{a^-} \mid s, a\right) = \mathbb{P}\left(X^a = 0 \mid \boldsymbol{\rho}^a\right) = \mathbb{E}_s\left[1 - P^a\right]. \tag{3.2.13}$$

IP1 Let us first look at what these changes mean for the intelligence problem IP1. The lump-sum reward $k(s, a)$ associated with continuing arm a in state s, which now includes the belief regarding P^a through $\boldsymbol{\rho}^a$, is given by the expected outcome of the evaluation

$$k(s, a) = P^a. \tag{3.2.14}$$

Note that the lump sum reward is a random quantity, of which the expectation with respect to the state of the system is taken within the optimality criteria.

The long-run average reward criterion takes a form similar to (3.2.2)

$$\max_{\pi} \lim_{\mathcal{H}\to\infty} \frac{1}{\mathcal{H}} \mathbb{E}_\pi \left[\sum_{e=0}^{e_{\mathcal{H}}-1} \mathbb{E}_{s_e}\left[P^{\pi(s_e)}\right]\right]. \tag{3.2.15}$$

We note that as $\mathcal{H} \to \infty$ so does $e_{\mathcal{H}} \to \infty$.

Let us examine the reward measure we wish to maximise more closely with the intent of simplifying it. We start by multiplying the expression within the first expectation by $e_{\mathcal{H}}/e_{\mathcal{H}}$ to get

$$\lim_{\mathcal{H}\to\infty} \frac{1}{\mathcal{H}} \mathbb{E}_\pi \left[\sum_{e=0}^{e_{\mathcal{H}}-1} \mathbb{E}_{s_e}\left[P^{\pi(s_e)}\right]\right] = \lim_{\mathcal{H}\to\infty} \mathbb{E}_\pi \left[\frac{1}{\mathcal{H}} \frac{e_{\mathcal{H}}}{e_{\mathcal{H}}} \sum_{e=0}^{e_{\mathcal{H}}-1} \mathbb{E}_{s_e}\left[P^{\pi(s_e)}\right]\right]. \tag{3.2.16}$$

Assuming all conditions as per the Lebesgue convergence theorem (Bartle, 1995) are met, we can exchange the limit and the expectation with respect to policy π and (3.2.16) takes the form

$$\mathbb{E}_\pi \left[\lim_{\mathcal{H}\to\infty} \frac{e_{\mathcal{H}}}{\mathcal{H}} \times \lim_{\mathcal{H}\to\infty} \frac{1}{e_{\mathcal{H}}} \sum_{e=0}^{e_{\mathcal{H}}-1} \mathbb{E}_{s_e}\left[P^{\pi(s_e)}\right]\right] = \mathbb{E}_\pi \left[\lim_{\mathcal{H}\to\infty} \frac{e_{\mathcal{H}}}{\mathcal{H}} \times \lim_{e_{\mathcal{H}}\to\infty} \frac{1}{e_{\mathcal{H}}} \sum_{e=0}^{e_{\mathcal{H}}-1} \mathbb{E}_{s_e}\left[P^{\pi(s_e)}\right]\right]. \tag{3.2.17}$$

If only one arm a is ever sampled from, through learning, the expectation $\mathbb{E}_s\left[P^a\right]$ converges to its underlying true value $\hat{p}^a$. We assume that as the process progresses, any policy that satisfies the optimality criteria would eventually "settle" on one of the arms, deviating from it with smaller and smaller probabilities. The arm the policy settles on in such way is the one that performs best according to some appropriate metric, namely arm $\pi(s_{e\to\infty})$. In the context of the intelligence problem this is not an unreasonable assumption as we expect an optimal policy to be able to identify and exploit the best source. Admittedly, one may construe a deterministic Markov policy that continues exploring the arms forever, but such policy would not be optimal as it fails to exploit the knowledge it gains from the exploration and therefore would never satisfy the optimality criteria. Consequently we write,

$$\lim_{e_{\mathcal{H}}\to\infty} \mathbb{E}_\pi\left[\frac{1}{e_{\mathcal{H}}}\sum_{e=0}^{e_{\mathcal{H}}-1}\mathbb{E}_{s_e}\left[P^{\pi(s_e)}\right]\right] = \mathbb{E}_\pi\left[\hat{p}^{\pi(s_{e\to\infty})}\right], \tag{3.2.18}$$

and

$$\lim_{\mathcal{H}\to\infty}\frac{e_{\mathcal{H}}}{\mathcal{H}} = \frac{1}{t^{\pi(s_{e\to\infty})}}, \tag{3.2.19}$$

which can be used to simplify (3.2.17) to get

$$\mathbb{E}_\pi\left[\frac{\hat{p}^{\pi(s_{e\to\infty})}}{t^{\pi(s_{e\to\infty})}}\right]. \tag{3.2.20}$$

Therefore the optimality criteria in terms of the long-run average reward is

$$\max_\pi \mathbb{E}_\pi\left[\frac{\hat{p}^{\pi(s_{e\to\infty})}}{t^{\pi(s_{e\to\infty})}}\right], \tag{3.2.21}$$

and any policy which eventually, and presumably through learning correctly identifies and chooses to exploit the arm $\mathbb{E}_\pi\left[s_{e\to\infty}\right] = a$ which has the highest $\hat{p}^a/t^a$ is optimal.

For Case 2, the ratio average reward criterion is written as

$$\max_\pi \liminf_{H\to\infty}\frac{\mathbb{E}_{\pi,s}\left[\sum_{e=0}^{H-1} r_e(s_e,\pi(s_e))\right]}{\mathbb{E}_{\pi,s}\left[\sum_{e=0}^{H-1} T_e\right]} = \max_\pi \lim_{H\to\infty}\frac{\mathbb{E}_\pi\left[\sum_{e=0}^{H-1}\mathbb{E}_{s_e}\left[P^{\pi(s_e)}\right]\right]}{\mathbb{E}_\pi\left[\sum_{e=0}^{H-1} t^{\pi(s_e)}\right]}. \tag{3.2.22}$$

For the same reasons as earlier, we presume policy π to settle on arm $\pi(s_{e\to\infty})$. Then

$$\lim_{H\to\infty}\frac{1}{H}\mathbb{E}_\pi\left[\sum_{e=0}^{H-1} t^{\pi(s_e)}\right] = \mathbb{E}_\pi\left[t^{\pi(s_{e\to\infty})}\right], \tag{3.2.23}$$

together with (3.2.18) can be used to re-write the objective as

$$\max_{\pi} \frac{\mathbb{E}_{\pi}\left[\hat{p}^{\pi(s_{e\to\infty})}\right]}{\mathbb{E}_{\pi}\left[t^{\pi(s_{e\to\infty})}\right]}, \tag{3.2.24}$$

therefore a policy for which settles on arm $\mathbb{E}_{\pi}\left[s_{e\to\infty}\right] = a$ so that

$$a := \arg\max_{a' \in \mathcal{S}_a} \frac{\hat{p}^{a'}}{t^{a'}}, \tag{3.2.25}$$

is optimal. Note that while (3.2.21) and (3.2.24) are not the same, identification and exploitation of the arm considered best with respect to the metric $\hat{p}^a/t^a$ leads to optimality in both cases.

IP4. The introduction of unknown success probabilities affects the expected reward of continuing arm a in state s of IP4 in a similar way it did IP1, so that

$$r(s,a) = \mathbb{E}_s\left[\frac{X^a}{t^a}\right] = \frac{\mathbb{E}\left[P^a\right]}{t^a}. \tag{3.2.26}$$

Following the same reasoning that any optimal policy converges to playing only one of the arms, as well as the arguments for $\mathbb{E}\left[P^a\right] \to \hat{p}^a$ apply, and therefore the optimality criterion of IP4 is

$$\max_{\pi} \lim_{H\to\infty} \frac{1}{H}\mathbb{E}_{\pi}\left[\sum_{e=0}^{H} \frac{\mathbb{E}\left[P^{\pi(s_e)}\right]}{t^{\pi(s_e)}}\right] = \max_{\pi} \mathbb{E}_{\pi}\left[\frac{\hat{p}^{\pi(s_{e\to\infty})}}{t^{\pi(s_{e\to\infty})}}\right], \tag{3.2.27}$$

which is the same optimality criterion as seen in (3.2.21). Therefore once again we conclude that any policy that is optimal for either formulation of IP1 or IP4 will be optimal for the others. In addition, any policy that ultimately identifies the arm with the highest underlying reward rate is optimal. However, we find it important to note that this conclusion only holds in the infinite horizon limit as we have made use of asymptotic results. If the horizon is not infinite, only very large, we cannot guarantee that the optimal polices will be identical. Furthermore, the speed at which the best arm is identified will play a role in maximising the expected reward from a given policy, as the time available to exploit said arm is not longer infinite.

Case 3

In the third case of the intelligence puzzle both P^a and $T^a \sim f(t \mid \boldsymbol{\theta}^a)$ are random quantities, but through $\boldsymbol{\theta}^a$ the distribution of the evaluation time is known. The state s^a of arm a is described very similarly to the previous case where the evaluation times were known, so that $s^a = (\boldsymbol{\rho}^a, \boldsymbol{\theta}^a)$ and the system state of the multi-armed bandit is given by $s = \left(s^1, ..., s^{\mathcal{A}}\right)$. Choosing arm a no longer guarantees a certain evaluation time, instead, it determines the distribution of said evaluation time at epoch e so that

$$f(T_e = t \mid a) = f(T^a = t \mid \boldsymbol{\theta}^a), \tag{3.2.28}$$

and therefore its expected duration is

$$\mathbb{E}\left[T_e \mid s, a\right] = \mathbb{E}\left[T^a \mid s^a\right]. \tag{3.2.29}$$

Since the $\boldsymbol{\theta}^a$'s are known, the second component of any s^a is constant. Therefore transitions only occur with respect to the first components, the $\boldsymbol{\rho}^a$, with the associated transition probabilities the same as stated in (3.2.12) and (3.2.13).

IP1. The reward associated with selecting arm a in state s is the same as in case 2, given by the expected outcome of the evaluation defined in terms of the lump sum reward $k(s, a)$, and shown in (3.2.14).

We found that non-deterministic evaluation times severely limit the ways in which we can manipulate the long-run average reward criterion. We start by writing

$$\max_{\pi} \lim_{\mathcal{H}\to\infty} \frac{1}{\mathcal{H}} \mathbb{E}_{\pi,s}\left[\sum_{e=0}^{e_{\mathcal{H}}-1} k(s_e, \pi(s_e))\right] = \max_{\pi} \lim_{\mathcal{H}\to\infty} \frac{1}{\mathcal{H}} \mathbb{E}_{\pi,s}\left[\sum_{e=0}^{e_{\mathcal{H}}-1} P^{\pi(s_e)}\right]. \tag{3.2.30}$$

However, $e_{\mathcal{H}}$ is no longer only a function of policy π but also of $\boldsymbol{\theta}$ and consequently s, through the realisation of the random sequence of evaluation times that result in $e_{\mathcal{H}}$ decision epochs within horizon $\mathcal{H}$. For that reason manipulating the above expression into showing explicit dependence of the evaluation times is difficult.

In contrast, there is little in addition to the steps needed in case 2 to suitably adapt the ratio average reward criterion to case 3. We write

$$\max_{\pi} \liminf_{H\to\infty} \frac{\mathbb{E}_{\pi,s}\left[\sum_{e=0}^{H-1} r_e(s_e, \pi(s_e))\right]}{\mathbb{E}_{\pi,s}\left[\sum_{e=0}^{H-1} T_e\right]} = \max_{\pi} \lim_{H\to\infty} \frac{\mathbb{E}_{\pi}\left[\sum_{e=0}^{H-1} \mathbb{E}_{s_e}\left[P^{\pi(s_e)}\right]\right]}{\mathbb{E}_{\pi}\left[\sum_{e=0}^{H-1} \mathbb{E}_{s_e}\left[T^{\pi(s_e)}\right]\right]}. \tag{3.2.31}$$

A set of arguments similar to those leading to (3.2.24) can be used here as well. As the number of samples taken from arm a increases, $\mathbb{E}_s\left[P^a\right]$ converges to $\hat{p}^a$, while $\mathbb{E}_s\left[T^a\right]$ remains constant. Same as before, we posit that optimal policies eventually settle on one of the arms which we denote by $\pi(s_{e\to\infty})$, and that we may use this property in manipulating the optimality criteria. The observation in (3.2.18) applies as stated, and for the current numerator we write

$$\lim_{H\to\infty} \frac{1}{H} \mathbb{E}_{\pi}\left[\sum_{e=0}^{H} \mathbb{E}_{s_e}\left[T^{\pi(s_e)}\right]\right] = \mathbb{E}_{\pi}\left[\mathbb{E}_{\boldsymbol{\theta}^{\pi(s_{e\to\infty})}}\left[T^{\pi(s_{e\to\infty})}\right]\right]. \tag{3.2.32}$$

Then the objective can be rewritten as

$$\max_{\pi} \frac{\mathbb{E}_{\pi}\left[\hat{p}^{\pi(s_{e\to\infty})}\right]}{\mathbb{E}_{\pi}\left[\mathbb{E}_{s_{e\to\infty}}\left[T^{\pi(s_{e\to\infty})}\right]\right]}, \tag{3.2.33}$$

which is a similar result to that of the previous case.

IP4. Our approach to IP4 changes little with the introduction of random evaluation times, the expected reward of continuing arm a in state s is still given by the expected reward rate

$$r(s,a) = \mathbb{E}_s\left[\frac{X^a}{T^a}\right] = \mathbb{E}_s\left[\frac{P^a}{T^a}\right]. \tag{3.2.34}$$

Then the optimality criterion of IP4 is

$$\max_{\pi} \lim_{H\to\infty} \frac{1}{H} \mathbb{E}_{\pi}\left[\sum_{e=0}^{H-1} \mathbb{E}_s\left[\frac{P^{\pi(s_e)}}{T^{\pi(s_e)}}\right]\right]. \tag{3.2.35}$$

As before, we made use of the asymptotic convergence of policy π.

Note that while in cases 1 and 2 the three optimality criteria considered in more detail were satisfied by the same policies, by allowing the evaluation times to be random we moved to a regime where that is no longer true. While we have not been able to write the long-run average reward criterion of IP1 in terms of $\hat{p}$ and T, it is clear that (3.2.33) and (3.2.35) are not equivalent.

Case 4

Last but not least we consider the intelligence problem in its most complex form, where the evaluation times are not only random with $T^a \sim f(t \mid \boldsymbol{\Theta}^a)$, but the $\boldsymbol{\Theta}^a$'s are only known through their prior and posterior distributions, characterised by parameters $\boldsymbol{\psi}^a$. In such a multi-armed bandit the state of arm a is given by the posterior parameters of both P^a and $\boldsymbol{\Theta}^a$, that is $s^a = (\boldsymbol{\rho}^a, \boldsymbol{\psi}^a)$. As we have seen before, the state of the system is $s = (s^1, ..., s^{\mathcal{A}})$. At every decision epoch e the distribution of the evaluation time and the transition probabilities are determined by the state s and the arm a chosen. The distribution and expectation of the evaluation time T_e following decision epoch e are

$$f(T_e = t \mid a) = f(T^a = t \mid s^a) \tag{3.2.36}$$

$$\mathbb{E}_{s_e}\left[T_e \mid a\right] = \mathbb{E}_{s_e^a}\left[T^a\right]. \tag{3.2.37}$$

Note that the expectation with respect to the state s includes taking the expectation with respect to both $\boldsymbol{\Theta}^a$ and $\boldsymbol{\psi}^a$ as well as $\boldsymbol{\rho}^a$. State transitions must account for changes in the state based on both the observed outcome of evaluating a tip and the observed evaluation time. Let us denote by $s_{a^{+,t}}$ the state after choosing arm a and consequently having observed a relevant tip $X^a = 1$ following an evaluation time of $T^a = t$. Then the transition probability from s to $s_{a^{+,t}}$ is simply the joint probability of the two required events

$$\mathbb{P}\left(s_{a^{+,t}} \mid a, s^a\right) = \mathbb{P}\left(X^a = 1, T^a = t \mid a, s^a\right), \tag{3.2.38}$$

and the transition probability from s to $s_{a^{-,t}}$ is similarly defined as

$$\mathbb{P}\left(s_{a^{-,t}} \mid a, s^a\right) = \mathbb{P}\left(X^a = 0, T^a = t \mid a, s^a\right). \tag{3.2.39}$$

The discussion around the reward structure and optimality criteria for IP1 and IP4 is very similar to what came before. So much so that we refer to case 3 for all relevant quantities, with the exception that when an expectation is taken with respect to s, the state is as defined for case 4 instead.

While it is not our main focus, we must mention Cayci et al. (2019), which proposes a model similar to IP3 under case 4. Nonetheless, there are some major differences between the problem as we described it and the problem defined by the work above, the most significant of which is that it allows interruption to the evaluation process.

Choosing the Intelligence Problem

The question that remains is which of these intelligence problems and optimality criteria we wish to investigate for the remainder of this chapter. While it may be up for debate which of the four intelligence problems reflects the everyday reality of an intelligence team the best, we thought IP1, coupled with the long-run average reward criterion described our aim the best, set out in Section 3.1. Unfortunately, we have seen that when evaluation times are random they only feature in the objective criteria implicitly, making construction of decision rules that take them into account difficult.

As demonstrated, in the presence of deterministic evaluation times (cases 1 and 2), both the ratio average reward criterion of IP1 and the average reward criterion of IP4 are asymptotically equivalent to the reward criterion we wish to optimise against. In Section 3.3 and Section 3.4 we approximate the long-run average reward criterion with the optimality criterion of IP4, and use the expected reward rate $\mathbb{E}_s\left[P^a/T^a\right]$ as a basis for developing an index of any arm a of the multi-armed bandit. While any method based on this approximation is not expected to be optimal, we reckon they are sufficient for designing heuristic polices.

We recognise that a different metric, inspired by the ratio average reward criterion of IP1 and defined as $\mathbb{E}_s\left[P^a\right]/\mathbb{E}_s\left[T^a\right]$ could be just as valid. For further consideration of the metric $\mathbb{E}_s\left[P^a\right]/\mathbb{E}_s\left[T^a\right]$ we refer the reader to Section 3.5.1, where we briefly consider how such a choice changes the heuristics we develop in Section 3.3.

3.3 The Intelligence Problem with Independent Parameters

In this section we examine the intelligence problem chosen in Section 3.2, which is to obtain as many relevant tips as possible in the long run. In a perfect world we would use a metric motivated by IP1, but as discussed in Section 3.2 we use the metric motivated by IP4 as an approximation. We consider a variant in which the probability that a tip is relevant is independent of its evaluation time and develop heuristic policies based on the expected reward rate of the available sources.

3.3.1 Model Description

In this section we detail the model of the intelligence problem, including the underlying Bayesian structure of the learning problem. While we repeat some aspects already outlined in Section 3.2, we do so for the sake of completeness.

The tips from source a, $a \in \{1, ..., \mathcal{A}\}$, form a sequence of IID Bernoulli random variables, $X^a \sim \text{Bernoulli}(P^a)$, therefore the jth outcome from arm a is $x_j^a \in \{0, 1\}$, and the tip is deemed relevant with $\mathbb{P}(x_j^a = 1) = P^a$ or a nuisance tip with $\mathbb{P}(x_j^a = 0) = 1 - P^a$. The time required to evaluate a tip from source a, determining which category it falls into is represented by random variable T^a, with realisations $t^a \in (0, \infty)$. The observed outcomes of X^a and T^a are recorded in the vectors $\boldsymbol{x}^a$ and $\boldsymbol{t}^a$.

Since in this section all P^a and T^a are independent, our chosen metric, the predictive expectation of the reward rate simplifies to

$$\mathbb{E}\left[\frac{P^a}{T^a}\right] = \mathbb{E}\left[P^a\right]\mathbb{E}\left[\frac{1}{T^a}\right], \tag{3.3.1}$$

but no further.

We consider two model variants, which differ only in the distribution of T^a. To distinguish them from models introduced in Section 3.4, we refer to them (perhaps unimaginatively) as *Independent Model 1* (IM1) and *Independent Model 2* (IM2). They are introduced with focus on a single arm to allow us to drop the arm-identifying superscripts.

Independent Model 1

In IM1 a conjugate Beta prior is placed on P, so that the relevance of a tip is given by

$$X \sim \text{Bernoulli}(P), \tag{3.3.2}$$

$$P \mid \boldsymbol{x} \sim \text{Beta}(\alpha, \beta), \tag{3.3.3}$$

where α and β are posterior shape parameters. Once another observation $X = x$ is made they are updated according to

$$\alpha' = \alpha + x, \tag{3.3.4}$$

$$\beta' = \beta + 1 - x. \tag{3.3.5}$$

For this model the evaluation times are given by the sum of a known and deterministic minimum evaluation time, which we denote as t^{min}, and an exponentially distributed excess evaluation time $\widetilde{T}$ so that $T = \widetilde{T} + t^{\text{min}}$;

$$\widetilde{T} \sim \text{Exponential}(\Lambda), \tag{3.3.6}$$

$$\Lambda \mid \widetilde{\boldsymbol{t}} \sim \text{Gamma}(\gamma, \delta), \tag{3.3.7}$$

where Λ is an unknown rate parameter on which a Gamma prior was placed. The parameters γ, δ are its posterior shape and rate parameters. After observing $\widetilde{T} = \widetilde{t}$ we update them using

$$\gamma' = \gamma + 1, \tag{3.3.8}$$

$$\delta' = \delta + \widetilde{t}. \tag{3.3.9}$$

As discussed in Section 3.2 the state of any arm is a knowledge state, which consist of its posterior parameters. In this case that state is given as $(\alpha, \beta, \gamma, \delta)$. Since t^{min} is a constant, it need not form part of the state.

The expectation of P has a well known closed form,

$$\mathbb{E}\left[P\right] = \frac{\alpha}{\alpha+\beta}. \tag{3.3.10}$$

However, no closed form formula exists for $\mathbb{E}\left[1/T\right]$, so it needs to be evaluated numerically from the following

$$\begin{aligned}\mathbb{E}\left[\frac{1}{T}\right] &= \int_0^\infty \int_0^\infty \frac{1}{\widetilde{t}+t^{\min}} \Lambda \exp(-\Lambda \widetilde{t}) \frac{\delta^\gamma}{\Gamma(\gamma)} \Lambda^{\gamma-1} \exp(-\delta\Lambda)\ \mathrm{d}\Lambda \mathrm{d}\widetilde{t} \\ &= \int_0^\infty \frac{1}{\widetilde{t}+t^{\min}} \frac{\gamma\delta^\gamma}{(\widetilde{t}+\delta)^{\gamma+1}} \mathrm{d}t'. \end{aligned} \tag{3.3.11}$$

After having determined the expectations of P and T, we can simply use (3.3.1) to find $\mathrm{E}\left[P/T\right]$.

Sampling provides an alternative way to obtain the distribution and expectation of P/T. To generate the jth sample:

1. Sample Λ_j from Gamma(γ, δ).
2. Sample $\widetilde{t}_j$ from Exponential(Λ_j).
3. Sample P_j from Beta(α, β).
4. Then the jth sample is given by $\dfrac{P_j}{\widetilde{t}_j + t^{\min}}$.

Then the expectation is given by the sample mean. Here, sampling is used mainly to visualise the distribution of P/T, by plotting a histogram of the samples as shown in Figure 3.3.1. The value of $t^{\min}$ is observed to have a significant effect on the shape of the distribution, as it determines the maximum achievable P/T.

Note that in the following investigations whenever $\mathrm{E}\left[P/T\right]$ is required (3.3.10) and (3.3.11) are used, not sampling. Sampling was only used to visualise the distribution of P/T.

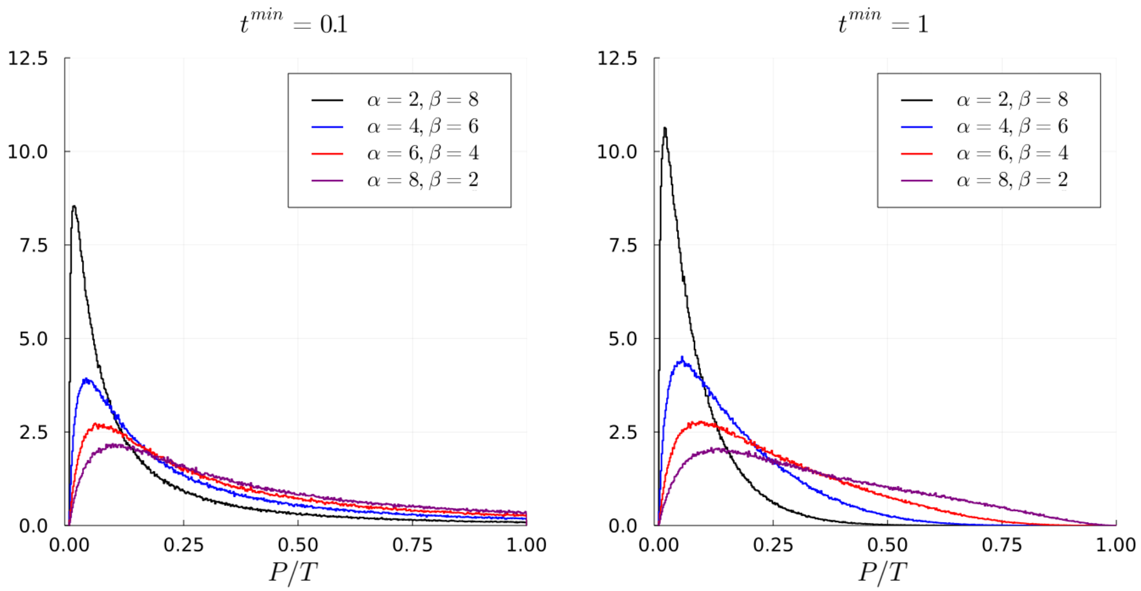


Figure 3.3.1: Distribution of P/T for a range of expected success probabilities with small $t^{\text{min}} = 0.1$ and large $t^{\text{min}} = 1$ minimum evaluation times. We can see that the choice of t^{min} influences the shape of the distribution.

Independent Model 2

The relevance of a tip in IM2 is modelled identically to IM1, as a Beta-Bernoulli random variable with its distribution and Bayesian updating shown (3.3.2) to (3.3.5). The two models are set apart by how the evaluation times are modelled; in this case T follows a Log-Normal distribution. The evaluation times are given by $T = \exp(\widetilde{T})$ where $\widetilde{T}$ is the log-evaluation time, which is Normally distributed with unknown mean Υ. To facilitate Bayesian learning we place a Normal prior on on Υ. We summarise the above as

$$\widetilde{T} \sim \text{Normal}(\Upsilon, s^2), \tag{3.3.12}$$

$$\Upsilon \mid \widetilde{\boldsymbol{t}} \sim \text{Normal}(\mu, \sigma^2), \tag{3.3.13}$$

where μ and σ^2 are the posterior mean and variance of Υ. The variance of the log-evaluation times, s^2, is a known constant. After observing another evaluation time with

$\widetilde{T} = \widetilde{t}$, we update μ and σ^a according to

$$\sigma'^2 = \left(\frac{1}{s^2} + \frac{1}{\sigma^2} \right)^{-1}, \tag{3.3.14}$$

$$\mu' = \sigma'^2 \left(\frac{\mu}{\sigma^2} + \frac{\widetilde{t}}{s^2} \right). \tag{3.3.15}$$

The state of an arm is given by its posterior parameters, which in this case is $(\alpha, \beta, \mu, \sigma^2)$.

The expectation of P is given by (3.3.10), same as in the previous model. Unlike in Section 3.3.1, a close form formula is available for $\mathbb{E}\left[1/T\right]$, shown in (3.3.16).

$$\mathbb{E}\left[\frac{1}{T}\right] = \exp\left(-\mu + \frac{\sigma^2 + s^2}{2}\right), \tag{3.3.16}$$

Then the expectation of P/T can also be given in a closed form.

$$\mathbb{E}\left[\frac{P}{T}\right] = \frac{\alpha}{\alpha + \beta} \exp\left(-\mu + \frac{\sigma^2 + s^2}{2}\right) \tag{3.3.17}$$

Just like in Section 3.3.1, sampling provides an alternative way to obtain the distribution and expectation of P/T. To generate a sample:

1. Sample Υ_j from Normal(μ, σ^2).
2. Sample $\widetilde{t}_j$ from Normal(Υ_j, s^2).
3. Sample P_j from Beta(α, β).
4. Then the jth sample is given by $\dfrac{P_j}{\exp(\widetilde{t}_j)}$.

Note that sampling was only used to visualise the distribution of P/T. Whenever $\mathrm{E}\left[P/T\right]$ is required (3.3.10) and (3.3.16) are used, not sampling. Figure 3.3.2 shows the distribution of the reward rate. The distribution of P/T under IM2 is very similar to that of IM1, but without a minimum evaluation time there is no upper limit on the reward rates.

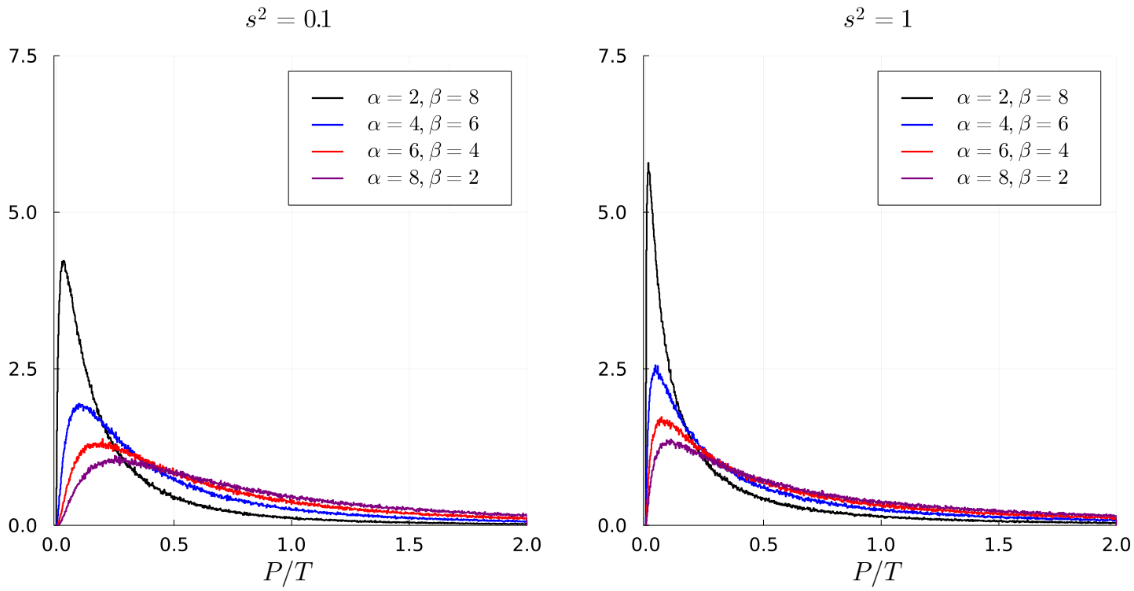


Figure 3.3.2: Distribution of P/T for a range of expected success probabilities with small $s^2 = 0.1$ and large $s^2 = 1$. The appearance of the distributions shown is very similar to what we have seen for IM1 in Figure 3.3.1.

3.3.2 Policies

This section describes the policies that have been developed or adapted to determine the source a tip should be evaluated from at any given decision time. Since this requires us to consider multiple arms again, we return to our previous notation of distinguishing arms using index a as a superscript.

Myopic Policy

The first policy to consider is a simple myopic policy that chooses the arm to play in a greedy manner: that the arm with the highest expected reward is chosen. Therefore at every decision time arm a^* is continued, where

$$a^* := \arg\max_a \mathbb{E}\left[\frac{P^a}{T^a}\right]. \tag{3.3.18}$$

This policy is pure exploitation without exploration. It does not aim to learn about the arms to make a more informed decision at the next decision time, only focusing on achieving the best reward from the current decision, making it myopic.

Bather Index

The Bather Index, also known as the Randomised Allocation Index, is an asymptotically optimal modification of the above myopic policy, first discussed in Bather (1980). The index for arm a is defined as the sum of the greedy term $\mathbb{E}\left[P^a/T^a\right]$ and a random term that decreases with the number of observations made on arm a. Here, a general case of such index was considered, so that the arm continued at the current decision time is given by

$$a^* := \arg\max_a \mathbb{E}\left[\frac{P^a}{T^a}\right] + b_1 b_2^{n^a} Y, \tag{3.3.19}$$

where $b_1 \geq 0$ is a scaling term, $0 \leq b_2 \leq 1$ governs how fast the random term decreases with the number of observations n^a and Y is an Exponential(1) random variable independently observed for all arms and decision times.

The parameters b_1 and b_2 control the exploration process, with either $b_1 = 0$ or $b_2 = 0$ equating the method to the myopic policy, where no exploration takes place. To achieve good performance b_1 and b_2 need to be tuned; the best values can be determined using a grid-search over a reasonable range of parameters. This is further discussed in Section 3.3.3.

Expected Generalised Gittins Index

One of the two major methodological contributions found in Section 3.3.2, the Expected Generalised Gittins Index (EGGI) heuristic defines a novel decision rule, based on an extension of the Gittins Index that allows for deterministic, but non-unit inter-decision times.

The Gittins Index provides an optimal policy over an infinite horizon, and serves as a good heuristic over a finite horizon. In the well studied case of deterministic unit length inter-decision times it arises from comparing the arm in question, arm a, independently of the other arms of the multi-armed bandit to a standard bandit arm that yields a guaranteed reward of λ on every pull. Depending on the properties of arm a and the value of λ, we may prefer to pull one or the other. However, we are free to adjust λ so that the expected rewards of arm a equal the expected rewards of the standard arm. In that case we are indifferent to which arm is pulled; we have found a standard arm of equivalent value to the original bandit arm a. In essence, the Gittins Index is this equivalent value $\lambda = \nu$ where it is equally optimal to pull either the standard arm or arm a. We discussed Gittins indices in more detail in Section 2.4. As the Gittins Index was developed for infinite horizons with discounted rewards, the value of the index also depends on the discount factor d.

We can define the Gittins index for a bandit with deterministic, non-unit inter-decision time k^a the same way. Note that this is the same regime as defined in Case 2 of Section 3.2. Then the index $\nu^a(k^a, d)$ is determined by equating the expected rewards from the standard arm and arm a as follows.

$$\frac{\nu^a(k^a,d)}{1-d} = \sup_{\tau} \mathbb{E}\left[\sum_{e=0}^{\tau-1} d^{k^a e} r^a(e) + d^{k^a \tau}\frac{\nu^a(k^a,d)}{1-d}\right], \tag{3.3.20}$$

$$\nu^a(k^a,d) = (1-d)\sup_{\tau>0} \frac{\mathbb{E}\left[\sum_{e=0}^{\tau-1} d^{k^a e} r^a(e)\right]}{1-\mathbb{E}\left[d^{k^a \tau}\right]}, \tag{3.3.21}$$

where τ is a positive stopping time, defined in terms of the decision epochs, after which the standard arm is continued instead of arm a, $r^a(e)$ is the reward at decision epoch e, and d is the discount factor. For $k^a = 1$ the original Gittins index is recovered.

For Beta-Bernoulli bandits the calibration approach as described above does not only provide a definition for the Gittins index, but also a computationally efficient, direct way to calculate it. Every arm of the bandit is considered independently of the others, only compared to a standard arm that acts as a measuring device.

In every state, determined by the number of successes α^a and failures β^a observed, two actions are available; pull the standard arm for a reward of λ, or the non-standard arm that is under evaluation. The best action is to pull the arm that results in the maximum expected reward over the infinite horizon. For any given λ, backward recursion is used to calculate the maximum expected reward $\mathcal{R}(\lambda, \alpha^a, \beta^a)$ of each state and determine which of the two arms should be continued. While the decision problem considered is an infinite horizon one, we can approximate it with a finite horizon equivalent due to the presence of discounting. Note that the closer d is to 1, the larger the finite horizon must be to obtain accurate indices. For the purposes of calculating the index, the horizon is the total number of observations in the decision problem, N.

The recursion starts from states where no more decisions are left, so the terminal states must satisfy $\alpha^a + \beta^a = N$. Their values are set irrespective of λ to $\mathbb{E}\left[P^a\right]$ so that

$$\mathcal{R}(\lambda, \alpha^a, \beta^a) = \frac{\alpha^a}{\alpha^a + \beta^a}. \tag{3.3.22}$$

The recursion equation for all other states $\alpha^a + \beta^a < N$ is given as

$$\begin{aligned}\mathcal{R}(\lambda, \alpha^a, \beta^a) = \max \Bigg\{ \frac{\lambda}{1-d}, &\frac{\alpha^a}{\alpha^a + \beta^a}\left[1 + d^{k^a}\mathcal{R}(\lambda, \alpha^a + 1, \beta^a)\right] \\ &+ \frac{\beta^a}{\alpha^a + \beta^a}\left[d^{k^a}\mathcal{R}(\lambda, \alpha^a, \beta^a + 1)\right] \Bigg\},\end{aligned} \tag{3.3.23}$$

a version of what appears in Gittins et al. (2011), modified to allow for $k \neq 1$. The recursion stops at the state the first decision must be made in. The first term in (3.3.23) is the total return if the standard arm is activated, while the second term is the expected total reward if the non-standard arm is activated. Note that once the standard arm is optimal to pull, it will remain optimal to pull in all subsequent decision times, as the state of the non-standard arm does not change.

The above process of finding the optimal policy for every state is repeated for gradually increasing values of $\lambda \in [0, 1)$. The Gittins Index $\nu(k^a, d)$ for a state is given by the mid-point between the largest λ for which it is optimal to activate the non-standard arm, and the smallest λ for which it is optimal to activate the standard arm.

Calculation of the index in such a way is very efficient. Table 3.3.1 shows the recorded run-times required in Julia to generate the indices for a range of N's. Note that the run-times are quick for integer instances of k, and given that tables of indices can be

N	**Integer** k	**Non-integer** k
100	0.17s	12.89s
200	0.70s	49.72s
500	5.14s	304.3s
1000	32.06s	1371s
2000	141.2s	1657s

Table 3.3.1: Table of CPU times for generation of the non-unit inter-decision time Gittins index.

pre-generated even for non-integer k's the run-times are reasonable. Note that the difference is due to having to exponentiate to k, which is a computationally more intensive step for non-integer values.

A useful property of $\nu^a(k^a, d)$ is that in the undiscounted limit $d \to 1$ it takes the form

$$\lim_{d\to 1} \nu^a(k^a, d) = \sup_{\tau > 0} \frac{\mathbb{E}\left[\sum_{e=0}^{\tau-1} r^a(e)\right]}{k^a \mathbb{E}\left[\tau\right]} = \frac{\lim_{d\to 1} \nu^a(1, d)}{k^a}, \tag{3.3.24}$$

which is the well established Gittins index for deterministic unit inter-decision times divided by the non-unit inter-decision time, as illustrated in Figure 3.3.3. This property suggests that in the undiscounted limit there is no need to calculate and store a table of indices for every arm with a different k^a, it is sufficient to obtain a single table of Gittins indices where $k = 1$ and adjust the values using (3.3.24) and the appropriate k^a. Tables of Gittins Indices for $k = 1$ with $d = 0.99$ are available in Gittins et al. (2011), but can also be calculated for Beta-Bernoulli bandits via (3.3.23) for any d.

In the context of the intelligence problem the deterministic non-unit inter-decision times equate to tips requiring a known, fixed evaluation time to determine the relevance of a tip. However, the bandits considered in Chapter 3 have random evaluation times, and therefore the Gittins index policy cannot be applied in the above form.

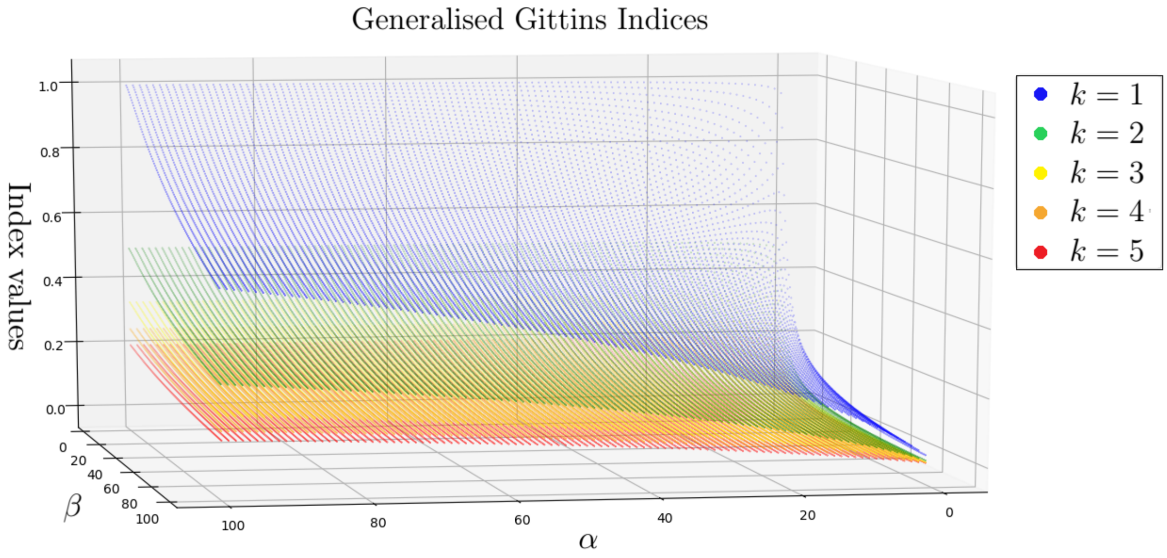


Figure 3.3.3: Generalised Gittins indices for the Beta-Bernoulli bandit with discount factor $d = 0.999$ for deterministic evaluation times $k = 1, 2, 3, 4, 5$.

Since the intelligence problem was defined without discounting, we propose a heuristic based on (3.3.24). By replacing k^a with T^a, the random evaluation time, and taking the expectation with respect to the posterior distribution of T^a, we define a new index given as

$$\mathbb{E}\left[\lim_{d\to 1}\nu^a(T^a)\right] = \mathbb{E}\left[\frac{\lim_{d\to 1}\nu^a(1,d)}{T^a}\right] = \mathbb{E}\left[\frac{1}{T^a}\right]\lim_{d\to 1}\nu^a(1,d). \tag{3.3.25}$$

Since $\mathrm{E}\left[1/T^a\right]$ is straightforward to calculate using either (3.3.11) or (3.3.16), obtaining this index is similarly clear cut. Therefore we can use it to construct a heuristic policy which we call the Expected Generalised Gittins Index (EGGI) policy. At every decision time it continues arm a^*, where

$$a^* := \arg\max_a \mathbb{E}\left[\lim_{d\to 1}\nu^a(T^a)\right]. \tag{3.3.26}$$

Knowledge Gradient

As we have already introduced the knowledge gradient policy for multi-armed bandits in Section 2.5, here we only provide a brief run-down. The knowledge gradient for MAB assumes that while rewards are collected till the end of the horizon, there is only one further opportunity to learn, after which we will cease to update our beliefs. The knowledge gradient policy then chooses one of the arms to learn about as to maximise the expected total reward over the horizon. The knowledge gradient policy for a MAB with random evaluation times is defined following the same logic.

Define the sets $\mathcal{T}_x^a \subset \mathbb{R}^+, x \in \{0,1\}$ so that

$$\mathcal{T}_x^a = \left\{t; \mathbb{E}\left[\frac{P^a}{T^a} \mid \boldsymbol{x}^a, \boldsymbol{t}^a, x, t\right] \geq \max_{a' \neq a} \mathbb{E}\left[\frac{P^{a'}}{T^{a'}} \mid \boldsymbol{x}^{a'}, \boldsymbol{t}^{a'}\right]\right\}, \tag{3.3.27}$$

the sets of outcomes $T^a = t$ accompanied by $X^a = x$ from observing arm a that result in arm i having the highest expected reward rate.

Suppose that there is one more opportunity to learn within the residual horizon $\mathcal{H}_r$, and arm a is chosen. The sample from a is observed, and the posterior of arm a is then updated. Any remainder of the horizon is then taken up by continuous sampling from whichever source has the highest predictive expected success rate. From the definition of $\mathcal{T}_x^a$, if $t \in \mathcal{T}_x^a$ arm a has a higher expected reward rate than the best alternative, and therefore will be continued for the rest of the horizon. Otherwise the best alternative, arm $\underset{a' \neq a}{\operatorname{argmax}}\, \mathbb{E}\left[P^{a'}/T^{a'} \mid \boldsymbol{x}^{a'}, \boldsymbol{t}^{a'}\right]$ will be continued. Note that for the arm with the current highest expected predictive reward rate, the best alternative is the arm with the second highest rate, while for all other arms the best alternative is the one with the current highest rate. The expected predictive reward rate of the best alternative is independent of the new observation, and remains unchanged.
Then the total expected reward is given as

$$\mathcal{R}_{\text{total}}^a = \mathbb{E}\left[P^a \mid \boldsymbol{x}^a, \boldsymbol{t}^a\right] + \mathcal{R}_{t \in \mathcal{T}_0^a}^a + \mathcal{R}_{t \notin \mathcal{T}_0^a}^a + \mathcal{R}_{t \in \mathcal{T}_1^a}^a + \mathcal{R}_{t \notin \mathcal{T}_1^a}^a, \tag{3.3.28}$$

where $\mathbb{E}\left[P^a \mid \boldsymbol{x}^a, \boldsymbol{t}^a\right]$ is the immediate expected reward of observing arm a, the terms

$$\mathcal{R}_{t \in \mathcal{T}_0^a}^a = \int_{t \in \mathcal{T}_0^a} \left[\mathcal{H}_r - t\right]^+ \mathbb{E}\left[\frac{P^a}{T^a} \mid \boldsymbol{x}^a, \boldsymbol{t}^a, 0, t\right] \mathrm{d}\pi\left(0, t \mid \boldsymbol{x}^a, \boldsymbol{t}^a\right),$$
$$\mathcal{R}_{t \in \mathcal{T}_1^a}^a = \int_{t \in \mathcal{T}_1^a} \left[\mathcal{H}_r - t\right]^+ \mathbb{E}\left[\frac{P^a}{T^a} \mid \boldsymbol{x}^a, \boldsymbol{t}^a, 1, t\right] \mathrm{d}\pi\left(1, t \mid \boldsymbol{x}^a, \boldsymbol{t}^a\right)$$

are the probability weighted expected rewards if $t \in \mathcal{T}_x^a$ so that arm i is continued for the rest of the horizon and the terms

$$\begin{aligned}
\mathcal{R}^a_{t \notin \mathcal{T}_0^a} &= \max_{a' \neq a} \mathbb{E}\left[\frac{P^{a'}}{T^{a'}} \mid \boldsymbol{x}^{a'}, \boldsymbol{t}^{a'}\right] \int_{t \notin \mathcal{T}_0^a} \left[\mathcal{H}_r - t\right]^+ \mathrm{d}\pi\left(0, t \mid \boldsymbol{x}^a, \boldsymbol{t}^a\right), \\
\mathcal{R}^a_{t \notin \mathcal{T}_1^a} &= \max_{a' \neq a} \mathbb{E}\left[\frac{P^{a'}}{T^{a'}} \mid \boldsymbol{x}^{a'}, \boldsymbol{t}^{a'}\right] \int_{t \notin \mathcal{T}_1^a} \left[\mathcal{H}_r - t\right]^+ \mathrm{d}\pi\left(1, t \mid \boldsymbol{x}^a, \boldsymbol{t}^a\right)
\end{aligned}$$

are the probability weighted expected rewards if $t \notin \mathcal{T}_x^a$ so that the best alternative arm $a' \neq a$ is continued for the rest of the horizon. Note that both $\mathcal{R}^a_{t \in \mathcal{T}_x^a}$ and $\mathcal{R}^a_{t \notin \mathcal{T}_x^a}$ are functions of the best alternative reward rate $\max_{a' \neq a} \mathbb{E}\left[P^{a'}/T^{a'} \mid \boldsymbol{x}^{a'}, \boldsymbol{t}^{a'}\right]$ through $\mathcal{T}_x^a$.

This knowledge gradient policy for a bandit would always play arm a^*, so that

$$a^* := \arg\max_a \mathcal{R}^a_{\text{total}}. \tag{3.3.29}$$

By assuming P^a and T^a independent, the following terms simplify to

$$\mathbb{E}\left[P^a \mid \boldsymbol{x}^a, \boldsymbol{t}^a\right] = \mathbb{E}\left[P^a \mid \boldsymbol{x}^a\right], \tag{3.3.30}$$

$$\mathbb{E}\left[\frac{P^a}{T^a} \mid \boldsymbol{x}^a, \boldsymbol{t}^a, x, t\right] = \mathbb{E}\left[P^a \mid \boldsymbol{x}^a, x\right] \mathbb{E}\left[\frac{1}{T^a} \mid \boldsymbol{t}^a, t\right], \tag{3.3.31}$$

$$\mathrm{d}\pi\left(x, t \mid \boldsymbol{x}^a, \boldsymbol{t}^a\right) = \mathrm{P}\left(X = x \mid \boldsymbol{x}^a\right) \pi(t \mid \boldsymbol{t}^a) dt, \tag{3.3.32}$$

and the sets $\mathcal{T}_x^a$ take the form

$$\mathcal{T}_x^a = \left\{t; \mathbb{E}\left[P^a \mid \boldsymbol{x}^a, x\right] \mathbb{E}\left[\frac{1}{T^a} \mid \boldsymbol{t}^a, t\right] \geq \max_{a' \neq a} \mathbb{E}\left[\frac{P^{a'}}{T^{a'}} \mid \boldsymbol{x}^{a'}, \boldsymbol{t}^{a'}\right]\right\}. \tag{3.3.33}$$

Theorem 3.3.1. *Given that P^a and T^a are independent, $\mathcal{T}_x^a$ is defined by a threshold t_x^a;*

$$\mathcal{T}_x^a = \left\{t; 0 < t \leq t_x^a\right\}, \tag{3.3.34}$$

so that $t \in \mathcal{T}_x^a$ if and only if $0 < t \leq t_x^a$.

Theorem 3.3.1 can be proven using stochastic dominance, which we provide in Appendix A.1. The conditions under which a similar assumption can be made for the dependent case need further examination, but in general a single threshold does not define $\mathcal{T}_x^a$.

Considering all of the above, the expected total reward and its components simplify to

$$\mathcal{R}^a_{\text{total}} = \mathbb{E}\left[P^a \mid \boldsymbol{x}^a\right] + \mathcal{R}^a_{t\leq t^a_0} + \mathcal{R}^a_{t>t^a_0} + \mathcal{R}^a_{t\leq t^a_1} + \mathcal{R}^a_{t>t^a_1}, \tag{3.3.35}$$

where

$$\begin{aligned}
\mathcal{R}^a_{t\leq t^a_0} &= \mathbb{P}\left(X=0 \mid \boldsymbol{x}^a\right)\mathbb{E}\left[P^a \mid \boldsymbol{x}^a, 0\right]\int_{t\in\mathcal{T}^a_0}\left[\mathcal{H}_r - t\right]^+ \mathbb{E}\left[\frac{1}{T^a} \mid \boldsymbol{t}^a, t\right]\pi(t \mid \boldsymbol{t}^a)\mathrm{d}t,\\
\mathcal{R}^a_{t\leq t^a_1} &= \mathbb{P}\left(X=1 \mid \boldsymbol{x}^a\right)\mathbb{E}\left[P^a \mid \boldsymbol{x}^a, 1\right]\int_{t\in\mathcal{T}^a_1}\left[\mathcal{H}_r - t\right]^+ \mathbb{E}\left[\frac{1}{T^a} \mid \boldsymbol{t}^a, t\right]\pi(t \mid \boldsymbol{t}^a)\mathrm{d}t,\\
\mathcal{R}^a_{t>t^a_0} &= \mathbb{P}\left(X=0 \mid \boldsymbol{x}^a\right)\max_{a'\neq a}\mathbb{E}\left[\frac{P^{a'}}{T^{a'}} \mid \boldsymbol{x}^{a'}, \boldsymbol{t}^{a'}\right]\int_{t\notin\mathcal{T}^a_0}\left[\mathcal{H}_r - t\right]^+ \pi(t \mid \boldsymbol{t}^a)\mathrm{d}t,\\
\mathcal{R}^a_{t>t^a_1} &= \mathbb{P}\left(X=1 \mid \boldsymbol{x}^a\right)\max_{a'\neq a}\mathbb{E}\left[\frac{P^{a'}}{T^{a'}} \mid \boldsymbol{x}^{a'}, \boldsymbol{t}^{a'}\right]\int_{t\notin\mathcal{T}^a_1}\left[\mathcal{H}_r - t\right]^+ \pi(t \mid \boldsymbol{t}^a)\mathrm{d}t.
\end{aligned}$$

The largest value of t to consider equals $\mathcal{H}_r$, as we cannot get rewards past the horizon. Therefore a threshold will fall under one of three cases. When $t^a_x \leq t^{\min}$, the arm a cannot produce a higher expected predictive reward rate than the best alternative.

Proposition 3.3.2. *If evaluation times can take on arbitrarily small values (i.e. there is no minimum evaluation time associated with the model), any arm a can produce a better reward rate than that of the best alternative.*

The second possibility is $t^{\min} < t^a_x < \mathcal{H}_r$. If the observed outcome of T^a falls below the threshold, the expected predictive reward rate of arm a exceeds that of the best alternative arm. The last and third case is $\mathcal{H}_r \leq t^a_x$, when arm a will always be better than the best alternative.

The knowledge gradient is not an index policy, the expected total reward is heavily dependent on the reward rate of the best alternative arm. This adversely affects the decision making. Observing an arm that is not the current best has limited downsides; we either discover that it has a reward rate that is better than the current best, or in the worst case we fall back on the current best. On the other hand, there are many potential downsides evaluating the current best arm. Getting a bad result (failure and/or long evaluation time) reduces the rewards expected then on, or even worse, we have to fall back on the best available alternative. This is only avoided when $t_x^{\text{best}} \geq \mathcal{H}_r$, and we cannot observe an evaluation time long enough to make the current best arm lose its position. As a consequence, the arm with the highest expected total reward could have a low observed reward rate, with the expectation being high only due to the high reward rate of the best alternate. If the policy picks the arm with the highest expectation, it is bound to make some sub-optimal choices.

Calibrated Knowledge Gradient Index

To counter the problems with the knowledge gradient, we can define a new policy in which the expected total reward is still calculated assuming only one opportunity to learn, but we no longer compare arm a to the best alternative, but to a standard arm with a known and guaranteed reward stream with rate λ.

Inspired by the calibration approach from the Gittins index literature we consider each arm a separately. There are two options available; play the standard arm with a guaranteed constant reward rate of λ until the horizon is reached, or observe arm a one more time and continue either arm i or the standard arm for the remainder of the horizon, choosing the one with the higher expected reward rate. Then the value of λ, where we are indifferent between observing arm a one more time and continuing the standard arm straight away denoted as ν^a_{CKGI}, can serve as an index and we name it the Calibrated Knowldge Gradient Index (CKGI).

We define ν^a_{CKGI} as

$$\mathcal{H}_r \nu^a_{\text{CKGI}} = \mathcal{R}^a_{\text{total}}(\nu^a_{\text{CKGI}}), \tag{3.3.36}$$

where $\mathcal{R}^a_{\text{total}}(\lambda)$ is given as

$$\begin{aligned}
\mathcal{R}^a_{\text{total}}(\lambda) &= \mathbb{E}\left[P^a \mid \boldsymbol{x}^a, \boldsymbol{t}^a\right] + \mathcal{R}^a_{t\in\mathcal{T}^a_0(\lambda)}(\lambda) + \mathcal{R}^a_{t\notin\mathcal{T}^a_0(\lambda)}(\lambda) + \mathcal{R}^a_{t\in\mathcal{T}^a_1(\lambda)}(\lambda) + \mathcal{R}^a_{t\notin\mathcal{T}^a_1(\lambda)}(\lambda) \\
&= \mathbb{E}\left[P^a \mid \boldsymbol{x}^a, \boldsymbol{t}^a\right] + \int_{t\in\mathcal{T}^a_0(\lambda)} \left[\mathcal{H}_r - t\right]^+ \mathbb{E}\left[\frac{P^a}{T^a} \mid \boldsymbol{x}^a, \boldsymbol{t}^a, 0, t\right] \mathrm{d}\pi\left(0, t \mid \boldsymbol{x}^a, \boldsymbol{t}^a\right) \\
&\qquad + \lambda \int_{t\notin\mathcal{T}^a_0(\lambda)} \left[\mathcal{H}_r - t\right]^+ \mathrm{d}\pi\left(0, t \mid \boldsymbol{x}^a, \boldsymbol{t}^a\right) \\
&\qquad + \int_{t\in\mathcal{T}^a_1(\lambda)} \left[\mathcal{H}_r - t\right]^+ \mathbb{E}\left[\frac{P^a}{T^a} \mid \boldsymbol{x}^a, \boldsymbol{t}^a, 1, t\right] \mathrm{d}\pi\left(1, t \mid \boldsymbol{x}^a, \boldsymbol{t}^a\right) \\
&\qquad + \lambda \int_{t\notin\mathcal{T}^a_1(\lambda)} \left[\mathcal{H}_r - t\right]^+ \mathrm{d}\pi\left(1, t \mid \boldsymbol{x}^a, \boldsymbol{t}^a\right).
\end{aligned} \tag{3.3.37}$$

and the sets $\mathcal{T}^a_x(\lambda)$ are defined as

$$\mathcal{T}^a_x(\lambda) = \left\{t; \mathbb{E}\left[\frac{P^a}{T^a} \mid \boldsymbol{x}^a, \boldsymbol{t}^a, x, t\right] \geq \lambda\right\}.$$

These equations are the same as in Section 3.3.2, but the expected reward here depends on λ instead of $\max_{a'\neq a}\mathbb{E}\left[P^{a'}/T^{a'} \mid \boldsymbol{x}^{a'}, \boldsymbol{t}^{a'}\right]$.

Let us restrict our view to cases where Theorem 3.3.1 holds. We have to make some assumptions to ensure indexability.

Assumption 3.3.3. *For every arm a there is exactly one indifference value of $\lambda = \nu^a_{CKGI}$.*

Assumption 3.3.4. *Once the standard arm becomes optimal to play, it will stay optimal to play.*

Then we can define an index policy, where the arm to play is given by

$$a^* := \arg\max_a \nu^a_{\text{CKGI}}.$$

Even though this index does not depend on the state of the other arms, it does depend on the residual horizon, $\mathcal{H}_r$, and needs to be calculated for every arm at all decision times.

3.3.3 Applying the Policies

This section describes the specifics of how the policies described in Section 3.3.2 can be applied to the models in Section 3.4.1.

Applying the myopic policy

The myopic policy is the simplest to apply to the models as it only requires the expected reward rate $\mathbb{E}\left[P^a/T^a\right]$, given in (3.3.11) for IM1 and (3.3.16) for IM2. While the decision maker needs to know $\mathrm{E}\left[P^a/T^a\right]$ of every arm at every decision time, it is only necessary to recalculate for the arm that has just been observed; the expected reward rates of the arms not observed do not change.

Applying the Bather index

As per the discussion in Section 3.3.2, the Bather index is the sum of the expected reward rate, and a random term random term which consists of an Exponential(1) random variable modified by scale parameter b_1 and decay parameter b_2. These were determined by conducting a grid search over a range of parameter combinations. For IM1 (Section 3.3.1), an 11×11 grid with 0.1 between the grid points was used, so that both the values of b_1 and b_2 fell within the interval (0; 1). For IM2 (Section 3.3.1), a 21 $\times 11$ grid with 0.1 between the grid points was used, so that the values of b_1 fell within (0; 2) and b_2 fell within (0; 1). The range of values the scale parameter b_1 could take was chosen with the intent to ensure the best parameter combination was covered by the grid. To find the best parameters, for every combination of b_1 and b_2 the policy was applied to the same dataset used in the numerical experiments and the combination that achieved the largest reward was chosen. The best parameter values depend not only on the model but also on the horizon and to a lesser extent on the dataset itself. They are are shown in Appendix A.2.

The above represents a significant effort to tune b_1 and b_2, making the approach impractical to use. However, the purpose of developing such a method was to give a lower bound estimate of the optimal rewards, to better contextualise the performance of the other policies.

The performance of different combinations of b_1 and b_2 for the IM1 and IM2 have been visualised using a form of regret, as shown in Figure 3.3.4 and Figure 3.3.5. This measure provides the relative regret of using a certain policy based on the Bayes reward of said policy and the Bayes reward of an omniscient decision maker. As it used primarily to interpret the results of numerical experiments, we defer the more detailed introduction until Section 3.3.4 and (3.3.40). The contour plots reveal a structure in the rewards achieved that differs between the models and is influenced by the horizon and the number of arms present. Across both models and independent of the number of arms present, as the horizons increased the regions of lower regret shifted upwards, towards higher b_2s, meaning that it is worth investing more resources into exploration when the horizon is large, compared to when it is small. These low regret regions were also smaller for $\mathcal{A} = 5$ than for $\mathcal{A} = 2$ with the regret increasing quickly as the parameter values deviated; the policy is more sensitive to the values of the exploration parameters when there are more arms to explore.

Applying the EGGI

To find the Expected Generalised Gittins Index we can take advantage of the property in (3.3.24). As discussed in Section 3.3.2 it is sufficient to pre-generate a single table of Gittins indices with $k = 1$, which was done using backwards induction and the recursion equation (3.3.23) with a discount factor of $d = 1 - 10^{-10}$.

Note that for $d \sim 1$ some errors will be present in values of the Gittins indices which come from using backwards recursion to calculate the total expected rewards, which is a finite horizon approximation to an infinite horizon problem. These errors can be reduced if the horizon is sufficiently large to approximate infinity, but as d approaches 1, longer and longer horizons are needed to approximate infinity which are not compu-

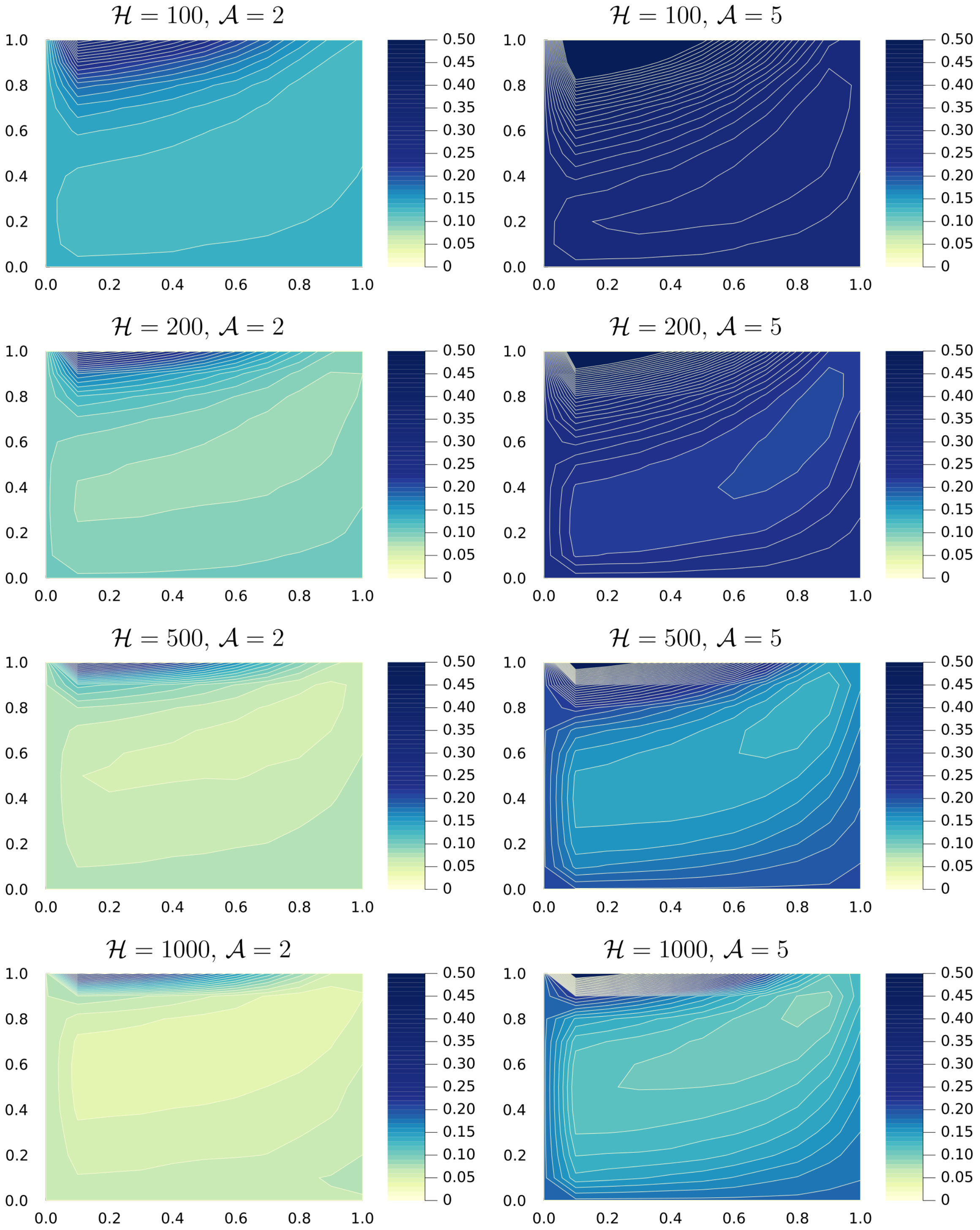


Figure 3.3.4: Relative regret achieved by the Bather index as defined in (3.3.40), observed for parameter combinations of b_1 (horizontal axis) and b_2 (vertical axis) with a range of horizons and number of arms present, for IM1.

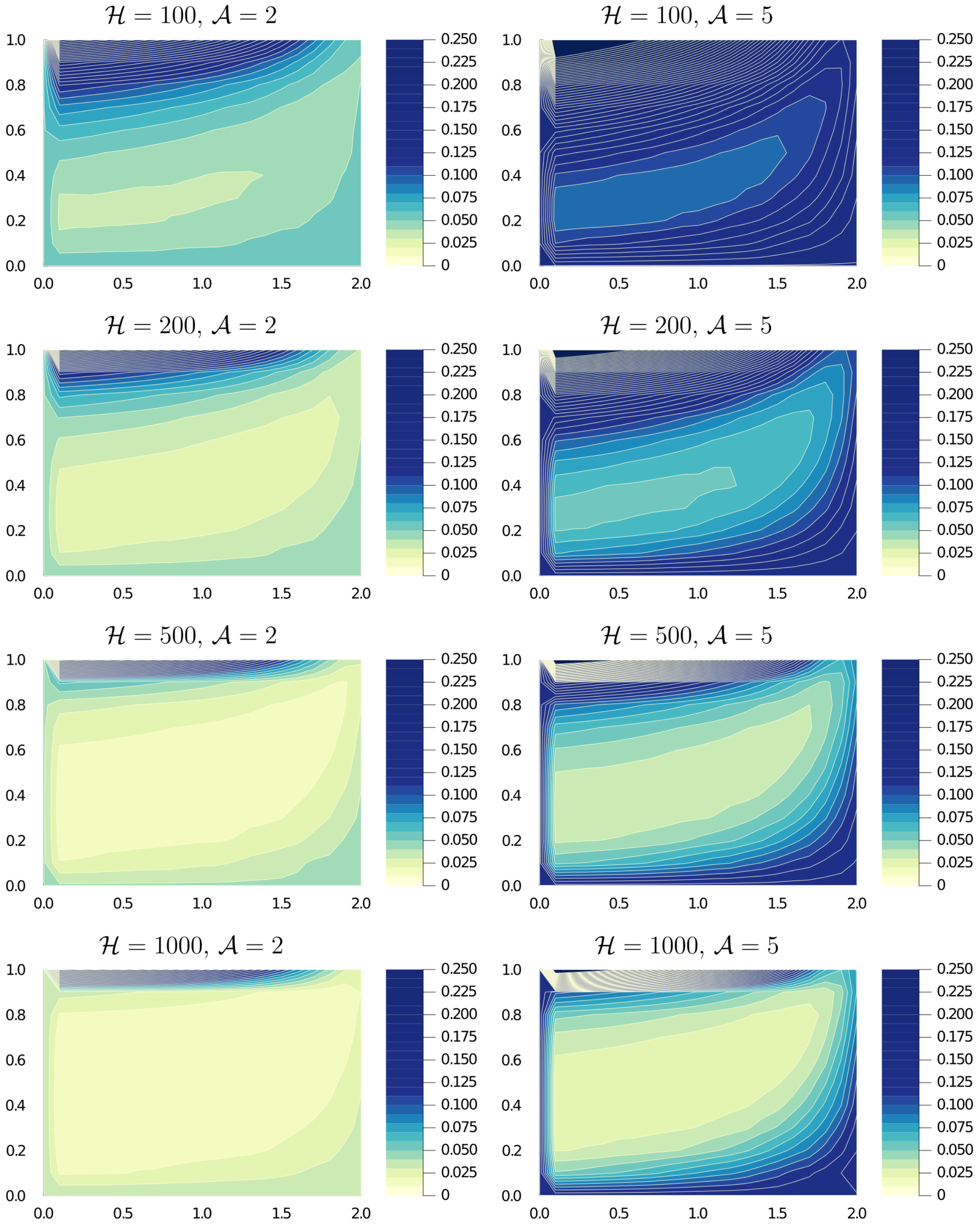


Figure 3.3.5: Relative regret achieved by the Bather index as defined in (3.3.40), observed for parameter combinations of b_1 (horizontal axis) and b_2 (vertical axis) with a range of horizons and number of arms present, for IM2.

tationally practical.

At every decision time the EGGI, $\mathbb{E}\left[\nu^a(T^a)\right]$, can be obtained for every arm a by multiplying the index found in the pre-generated table of Gittins indices that corresponds to the α^a and β^a of the arm by $\mathbb{E}\left[1/T^a\right]$. Note that neither the state or $\mathbb{E}\left[\nu^a(T^a)\right]$ of the arms not observed change, and so their EGGI also remains unchanged.

Applying the knowledge gradient and the CKGI

As seen in Section 3.3.2, both the knowledge gradient and the CKGI policies require similar quantities to be calculated. To be able to discuss them together, let as use λ to mean the best alternative reward rate, which, in the context of the knowledge gradient is $\lambda = \max_{a' \neq a} \mathrm{E}\left[P^{a'}/T^{a'} \mid \boldsymbol{t^{a'}}, \boldsymbol{x^{a'}}\right]$, while in the context of the CKGI λ is the guaranteed reward stream of a standard arm. Past the different roles of λ, calculation of $\mathcal{R}^a_{\text{total}}(\lambda)$ and the threshold $t^a_x(\lambda)$ are the same for both policies. For ease of notation let us focus on the calculation of these quantities for one of the arms and drop the arm-identifying superscript for the rest of this section.

As we have seen before, the threshold is the value of t, denoting the observed evaluation time where the expected reward rate of the arm after one last observation of t and x equals that of the best alternative, λ. In the case of IM1, this expected reward rate is only obtainable numerically, therefore numerical methods are also required to find such a threshold. It is a root of the function

$$f(t) = \mathbb{E}\left[P \mid \boldsymbol{x}, x\right] \mathbb{E}\left[\frac{1}{T} \mid \boldsymbol{t}, t\right] - \lambda.$$

Several numerical root finding algorithms exist, we opted to use the Newton-Rhapson method which is an iterative algorithm. At each iteration we evaluate

$$t_{(n+1)} = t_{(n)} - \frac{f(t_{(n)})}{f'(t_{(n)})},$$

where

$$
\begin{aligned}
f(t) &= \frac{\alpha+x}{\alpha+\beta+1}\int_0^\infty \frac{1}{\widetilde{t}+t^{\min}}\,\frac{\gamma+1}{\widetilde{t}+\delta+t}\left(\frac{\delta+t}{\widetilde{t}+\delta+t}\right)^{\gamma+1}\mathrm{d}\widetilde{t}-\lambda,\\
f'(t) &= -\frac{\alpha+x}{\alpha+\beta+1}\int_0^\infty \frac{1}{\widetilde{t}+t^{\min}}\,\frac{\gamma+1}{\left(\widetilde{t}+\delta+t\right)^3}\left(\frac{\delta+t}{\widetilde{t}+\delta+t}\right)^{\gamma}\left(\delta+t-\widetilde{t}(\gamma+1)\right)\mathrm{d}\widetilde{t},
\end{aligned}
$$

both of which are calculated numerically.

We consider to have found the root if the difference between $t_{(n+1)}$ and $t_{(n)}$ is smaller than 10^{-5}, with the threshold given by $t_x(\lambda) = t_{(n+1)}$.

Once the thresholds are known, the total expected reward is given by

$$
\mathcal{R}_{\text{total}}(\lambda) = \mathbb{E}\left[P \mid \boldsymbol{x}\right] + \mathcal{R}_{t\le t_0}(\lambda) + \mathcal{R}_{t>t_0}(\lambda) + \mathcal{R}_{t\le t_1}(\lambda) + \mathcal{R}_{t>t_1}(\lambda). \tag{3.3.38}
$$

Its components $\mathcal{R}_{t\le t_0}(\lambda)$, $\mathcal{R}_{t\le t_1}(\lambda)$, $\mathcal{R}_{t>t_0}(\lambda)$ and $\mathcal{R}_{t>t_1}(\lambda)$ are given as

$$
\begin{aligned}
\mathcal{R}_{t\le t_x}(\lambda) &= \mathbb{P}\left(X=x \mid \boldsymbol{x}\right)\int_0^{\max\{t_x(\lambda),\mathcal{H}_r\}}\left(\mathcal{H}_r-(t+t^{\min})\right)\mathbb{E}\left[\frac{P}{T}\mid \boldsymbol{t},\boldsymbol{x},t,x\right]\pi(t\mid \boldsymbol{t})\mathrm{d}t,\\
\mathcal{R}_{t>t_x}(\lambda) &= \mathbb{P}\left(X=x \mid \boldsymbol{x}\right)\lambda\int_{\min\{t_x(\lambda),\mathcal{H}_r\}}^{\mathcal{H}_r}\left(\mathcal{H}_r-(t+t^{\min})\right)\pi(t\mid \boldsymbol{t})\mathrm{d}t,
\end{aligned}
$$

with

$$
\pi\left(t\mid \boldsymbol{t}\right) = \frac{\gamma\delta^\gamma}{(t+\delta)^{\gamma+1}},
$$

evaluated numerically.

We need to calculate these same quantities for IM2. We can find the threshold t_x by making use of the Bayesian update given in (3.3.15) to get

$$
\begin{aligned}
&\mathbb{E}\left[P \mid \boldsymbol{x},x\right]\mathbb{E}\left[\frac{1}{T}\mid \boldsymbol{t},t_x\right] = \lambda,\\
&\frac{\alpha+x}{\alpha+\beta+1}\exp\left(-\frac{\sigma^{2\prime}}{\sigma^2}\mu+\frac{\sigma^{2\prime}+s^2}{2}\right)\exp\left(-\frac{\sigma^{2\prime}}{s^2}\log t_x\right) = \lambda,\\
&(t_x)^{-\frac{\sigma^2}{s^2}} = \lambda\,\frac{\alpha+\beta+1}{\alpha+x}\exp\left(\frac{\sigma^{2\prime}}{\sigma^2}\mu-\frac{\sigma^{2\prime}+s^2}{2}\right),\\
&t_x = \left[\lambda^{-1}\,\frac{\alpha+x}{\alpha+\beta+1}\exp\left(-\frac{\sigma^{2\prime}}{\sigma^2}\mu+\frac{\sigma^{2\prime}+s^2}{2}\right)\right]^{\frac{s^2}{\sigma^{2\prime}}},
\end{aligned}
$$

where s^2 is known and constant, α, β, μ and σ^2 are the current prior parameters, and ${\sigma^2}'$ is a posterior parameter after one more observation, given in (3.3.14). After expanding ${\sigma^2}'$ the threshold can be found as

$$t_x = \exp\left(-\frac{s^2}{\sigma^2}\mu + \frac{s^2}{\sigma^2}\frac{2\sigma^2+s^2}{2}\right)\left(\lambda^{-1}\ \frac{\alpha+x}{\alpha+\beta+1}\right)^{\frac{s^2+\sigma^2}{\sigma^2}} . \tag{3.3.39}$$

(3.3.39) gives the threshold t_x below which all t belong to τ_x.

The next step is to find $\mathcal{R}_{t\leq t_x}(\lambda)$ and $\mathcal{R}_{t>t_x(\lambda)}$. We write

$$\begin{aligned}
\mathcal{R}_{t\leq t_x}(\lambda) =& \mathbb{P}\left(X = x \mid \boldsymbol{x}\right)\int_0^{\max\{t_x(\lambda),\mathcal{H}_r\}} \left(\mathcal{H}_r - t\right)\mathbb{E}\left[\frac{P}{T} \mid \boldsymbol{t}, \boldsymbol{x}, t, x\right]\pi(t \mid \boldsymbol{t})\mathrm{d}t,\\
=& \mathbb{P}\left(X = x \mid \boldsymbol{x}\right)\int_0^{\max\{t_x(\lambda),\mathcal{H}_r\}} \left(\mathcal{H}_r - t\right)\frac{\alpha+x}{\alpha+\beta+1}\\
& \exp\left(-\frac{{\sigma^2}'}{\sigma^2}\mu + \frac{{\sigma^2}'+s^2}{2} - \frac{{\sigma^2}'}{s^2}\log t\right)\frac{1}{t\sqrt{2\pi\sigma^2}}\exp\left(-\frac{\left(\log t-\mu\right)^2}{2\sigma^2}\right)\mathrm{d}t\\
=& \mathbb{P}\left(X = x \mid \boldsymbol{x}\right)\int_0^{\max\{t_x(\lambda),\mathcal{H}_r\}} \left(\mathcal{H}_r - t\right)\frac{\alpha+x}{\alpha+\beta+1}\\
& \exp\left(\frac{-s^2\mu + \left(2\sigma^2+s^2\right)s^2 - \sigma^2\log\left(t\right)}{\sigma^2+s^2}\right)\frac{1}{t\sqrt{2\pi\sigma^2}}\exp\left(-\frac{\left(\log(t)-\mu\right)^2}{2\sigma^2}\right)\mathrm{d}t\\
=& \mathbb{P}\left(X = x \mid \boldsymbol{x}\right)\frac{\alpha+x}{\alpha+\beta+1}\exp\left(\frac{-s^2\mu + \left(2\sigma^2+s^2\right)s^2}{\sigma^2+s^2}\right)\frac{1}{\sqrt{2\pi\sigma^2}}\\
& \int_{-\infty}^{\max\{\log(t_x(\lambda)),\log(\mathcal{H}_r)\}} \left(\mathcal{H}_r - \exp(\widetilde{t})\right)\exp\left(-\frac{\left(\widetilde{t}-\mu\right)^2}{2\sigma^2} - \frac{\sigma^2\widetilde{t}}{\sigma^2+s^2}\right)\mathrm{d}\widetilde{t},
\end{aligned}$$

taking the integral with respect to the log time $\widetilde{t}$ instead of the real time and get

$$\begin{aligned}
\mathcal{R}_{t\leq t_x}(\lambda) =& \mathbb{P}\left(X = x \mid \boldsymbol{x}\right)\frac{\alpha+x}{\alpha+\beta+1}\frac{1}{2}\exp\left(-\mu + s^2 + \frac{s^2\sigma^2}{\sigma^2+s^2}\right)\\
& \times\left[\exp\left(\frac{2\mu\left(\sigma^4+s^4\right)+\sigma^2 s^2\left(4\mu+s^2\right)}{2\left(\sigma^2+s^2\right)^2}\right)\right.\\
& \times\left(\operatorname{erf}\left(\frac{\sigma^2\left(\mu+s^2-\log\left(\max\left\{t_x(\lambda),\mathcal{H}_r\right\}\right)\right)+s^2\left(\mu-\log\left(\max\left\{t_x(\lambda),\mathcal{H}_r\right\}\right)\right)}{\sqrt{2\sigma^2}\left(\sigma^2+s^2\right)}\right)-1\right)\\
& \left.+\exp\left(\frac{\sigma^6}{2\left(\sigma^2+s^2\right)^2}\right)\times\left(\operatorname{erf}\left(\frac{\sigma^4+\left(\log\left(\max\left\{t_x(\lambda),\mathcal{H}_r\right\}\right)-\mu\right)\left(\sigma^2+s^2\right)}{\sqrt{2\sigma^2}\left(\sigma^2+s^2\right)}\right)+1\right)\right].
\end{aligned}$$

Finding $\mathcal{R}_{t>t_x(\lambda)}$ does not require a change of variable and can be found in terms of real time. It is given as

$$\begin{aligned}\mathcal{R}_{t>t_x(\lambda)} =& \mathbb{P}\left(X=x \mid \boldsymbol{x}\right)\lambda \int_{t_x}^{\mathcal{H}_r} \left(\mathcal{H}_r - t\right)\pi(t \mid \boldsymbol{t})\mathrm{d}t,\\ =& \mathbb{P}\left(X=x \mid \boldsymbol{x}\right)\lambda \int_{t_x}^{\mathcal{H}_r} \frac{\mathcal{H}_r - t}{t\sqrt{2\pi\sigma^2}}\exp\left(-\frac{\left(\log(t)-\mu\right)^2}{2\sigma^2}\right)\mathrm{d}t\\ =& \mathbb{P}\left(X=x \mid \boldsymbol{x}\right)\lambda\\ \times& \left[\exp\left(\mu_i + \frac{\sigma^2}{2}\right)\left(\mathrm{erf}\left(\frac{\mu+\sigma^2-\log(\mathcal{H}_r)}{\sqrt{2\sigma^2}}\right) - \mathrm{erf}\left(\frac{\mu+\sigma^2-\log\left(\min\left\{t_x,\mathcal{H}_r\right\}\right)}{\sqrt{2\sigma^2}}\right)\right)\right.\\ +& \left.\mathcal{H}_r\left(\mathrm{erf}\left(\frac{\mu-\log\left(\min\left\{t_x,\mathcal{H}_r\right\}\right)}{\sqrt{2\sigma^2}}\right) - \mathrm{erf}\left(\frac{\mu-\log(\mathcal{H}_r)}{\sqrt{2\sigma^2}}\right)\right)\right].\end{aligned}$$

Then the total expected reward is given by (3.3.38). Having obtained $\mathcal{R}_{\text{total}}(\lambda)$, from this point onward we revert to using the arm-identifying superscripts again.

The differences in how the knowledge gradient and the CKGI are applied appear once $\mathcal{R}^a_{\text{total}}(\lambda)$ is obtained. The knowledge gradient policy bases its decisions on directly comparing $\mathcal{R}^a_{\text{total}}(\lambda)$ where λ provides the best alternative reward rate $\lambda = \max_{a'\neq a}\mathrm{E}\left[P^{a'}/T^{a'} \mid \boldsymbol{t^{a'}}, \boldsymbol{x^{a'}}\right]$.

The CKGI requires a few more steps as described in Section 3.3.2 to find the index ν^a_{CKGI}. It must satisfy (3.3.36) and therefore ν^a_{CKGI} is the root of the function

$$f\left(\lambda\right) = \mathcal{R}^a_{\text{total}}\left(\lambda\right) - \mathcal{H}_r\lambda.$$

To obtain the root, iterative root finding algorithms such as the Newton-Rhapson method can be used, where one iteration is given as

$$\lambda_{(n+1)} = \lambda_{(n)} - \frac{f\left(\lambda_{(n)}\right)}{f'\left(\lambda_{(n)}\right)}.$$

The required differential $f'\left(\lambda\right)$ can be found numerically.

We wish to finish the discussion around applying the different policies by considering the time required to compute the above discussed quantities. These are shown in

	Greedy	Bather	EGGI*	KG	CKGI
IM1	23.95 μs	25.03 μs	46.54 μs	837.24 μs	10.35 ms
IM2	0.15 μs	0.76 μs	0.20 μs	1.29 μs	21.23 μs

Table 3.3.2: Computational time of the knowledge gradient and the indices used by the various policies for IM1 and IM2. Note that both micro and milliseconds are used.

* The times shown do not include the time taken to generate the index tables. For index generation times see Table 3.3.1.

Table 3.3.2. It is clear that the lack of closed form formulas in the case of IM1 makes obtaining these more costly, with the impact most significant for the knowledge gradient and the CKGI. While these times appear quite small, they represent the effort to obtain a single quantity, and the calculations to obtain those may need to be repeated few thousand times in the course of one instance of a simulation.

3.3.4 Numerical Experiments

Let us apply the policies described in Section 3.3.2 to the models, described in Section 3.3.1. The priors used were $\alpha^a_{(0)} = 1, \beta^a_{(0)} = 1$ for both IM1 and IM2, $\gamma^a_{(0)} = 1, \delta^a_{(0)} = 1$ for IM1 and $\mu^a_{(0)} = 0, {\sigma^a}^2{}_{(0)} = 1$ for IM2. All $t^{\min}$ and s^2 were the same for all arms. Whenever closed form formulas were not available for a quantity of interest numerical methods were used. We ran these simulations for a range of horizons ($\mathcal{H} = 100, 200, 500, 1000$, and in the case of IM2 also $\mathcal{H} = 2000$) and for $\mathcal{A} = 2$ and $\mathcal{A} = 5$ arms. Note that the use of a finite horizon in the numerical study was motivated by necessity, as we cannot simulate the problem with an infinite horizon. We calculated the Bayes reward for all policies, simulating 10000 instances of the problem where the parameters P^a, Λ^a and Υ^a were sampled from their prior distributions.

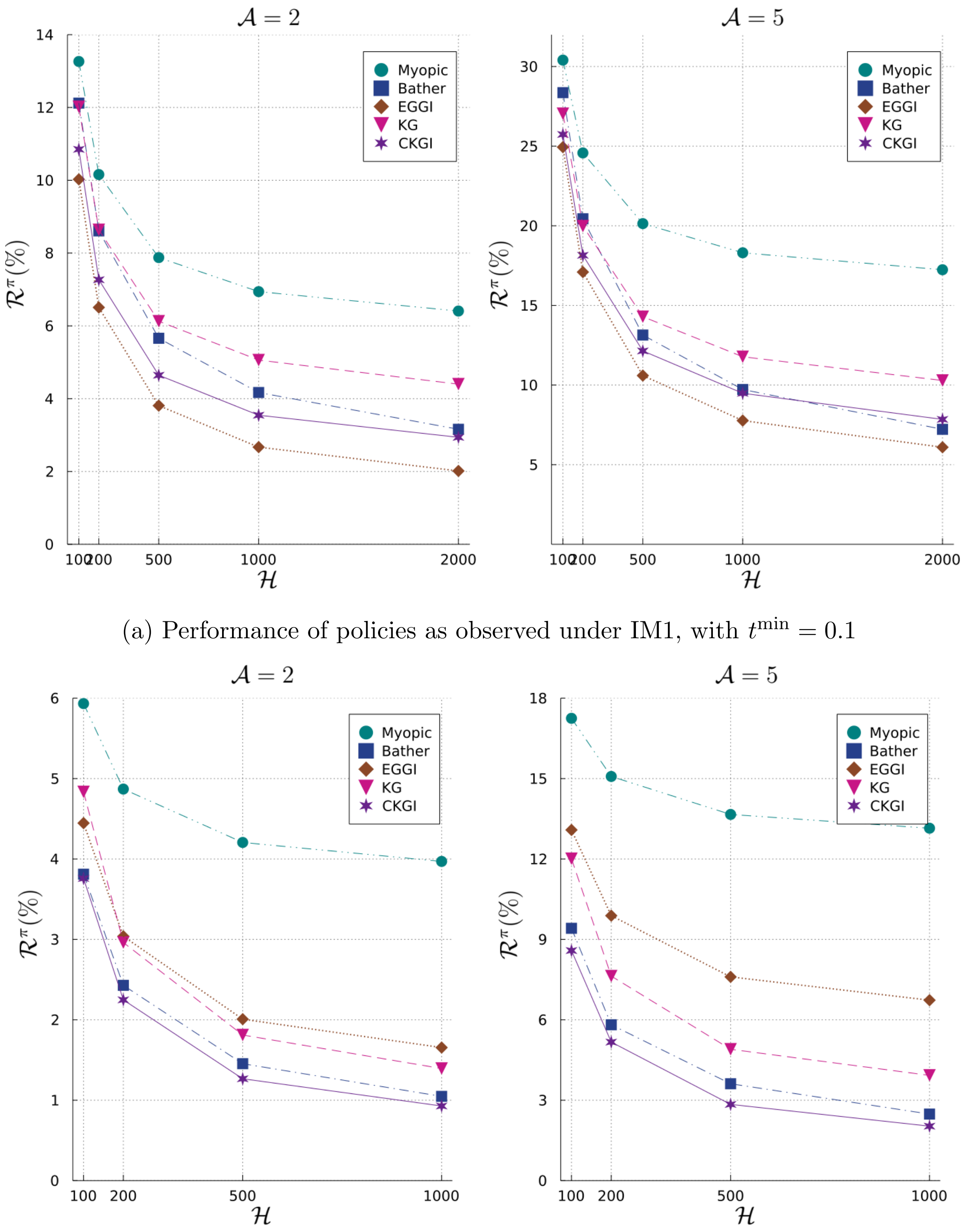
To compare these rewards, we used a regret-type measure. We consider the superoptimal policy to be one in which an omniscient decision maker that does not need to learn about the arms and picks a single arm at the start to observe for all decision times, identifying for every instance of the problem the arm that produces the highest realised reward over the horizon. Note that only a super-omniscient decision maker can make such a selection. Then the Bayes reward of following the optimal policy is obtained the same way as for any other policies. We define the (relative) regret of policy π as

$$\mathcal{R}^{\pi} = \frac{\text{Reward}^{\pi^{\text{SO}}} - \text{Reward}^{\pi}}{\text{Reward}^{\pi^{\text{SO}}}} \tag{3.3.40}$$

the relative difference between the Bayes reward of the super-optimal policy π^{SO} and the Bayes reward of policy π.

Figure 3.3.6 displays the results of the numerical experiments, comparing the regrets across the two models, IM1 with $t^{\min} = 0.1$ and IM2 with $s^2 = 0.1$. In general, a higher regret was observed for $\mathcal{A} = 5$ compared to $\mathcal{A} = 2$ as the observations needed to be split over more arms. What is clearly evident, is the importance of learning. The naive, myopic policy consistently achieves the highest regret, and the Bather index performs well by exploring the arms. When compared to the myopic policy, the EGGI, knowledge gradient and CKGI policies realise a much lower regret for all horizons. However, there is an issue with the Bather index; it requires a degree of omniscience as its performance highly depends on the initial tuning of the exploration parameters. As we don't have an effective method to determine the best exploration parameters without experimentation, the Bather policy proves impractical. Fortunately, none of the other methods suffer from the same issue, which at the same time makes comparing the performance of the EGGI, the knowledge gradient and the CKGI to the Bather policy unfair.

It is interesting to note, that while the EGGI achieves the lowest regret for IM1, for IM2 it is the CKGI that performs best. This can be explained by the different ways learning is approached by the two policies and the amount of information contained in a single observation. In essence, when calibration is used to find either the EGGI or the CKGI, a guaranteed reward stream is compared to the expected future reward. While both heuristics are based on the Gittins index with some kind of approximation of the expected future reward, the nature of the approximations make a big difference. While the EGGI calculates the expected future rewards using and updating the poste-



(a) Performance of policies as observed under IM1, with $t^{\min} = 0.1$

(b) Performance of policies as observed under IM2, with $s^2 = 0.1$

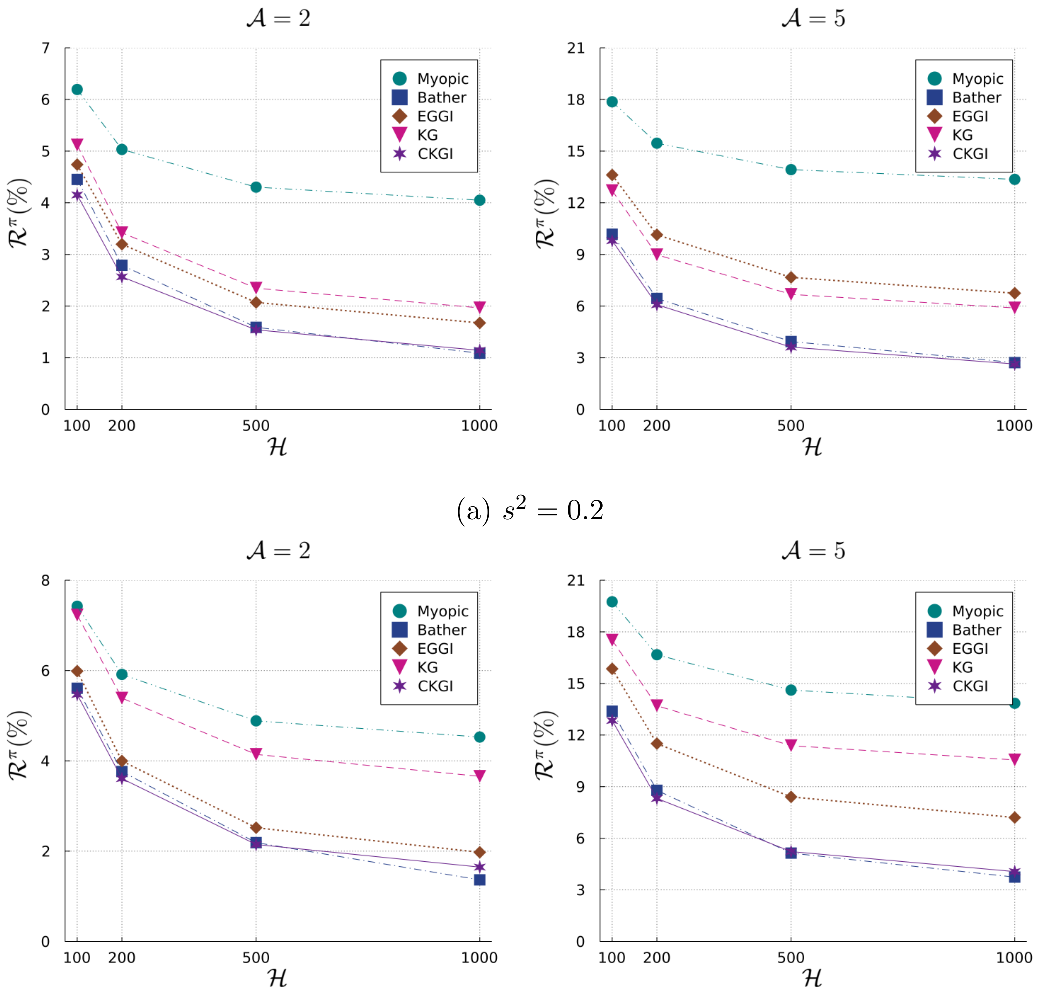
Figure 3.3.6: Relative regret $\mathcal{R}^{\pi}$, as defined in (3.3.40), achieved by the policies discussed in Section 3.4.2 in the numerical experiments of IM1 and IM2.

rior distribution of P to an infinite horizon, the distribution of the evaluation time T only contributes a point estimate. On the other hand, for the purposes of calculating the expected future reward the CKGI only updates the posterior distribution of P with the next observation, but also updates and uses the distribution of T.

In the case of IM2 the variance of the log-time s^2 is a known parameter which we have direct control over. When it is small, such as in the example shown in Figure 3.3.6, a single observation carries a large enough amount of information about the distribution of T^a and where the true log-mean might lie that it is worth using that information, even if learning needs to be limited to a single step ahead. In such scenario the knowledge gradient and CKGI policies learn efficiently. On the other hand, when a single observation does not carry enough information about the distribution of T^a, it is more beneficial to focus on learning about P^a and use a point estimate instead of the distribution of T^a.

To illustrate and further examine this point, we increased s^2 as shown in Figure 3.3.7 and Figure 3.3.8. As s^2 was increased, the amount of information on where the true log-mean might lie was reduced and the performance of all but the EGGI policy quickly deteriorated. The regrets achieved by the EGGI increased much slower, and therefore its performance in comparison to the others improved. It first overtook the knowledge gradient policy at $s^2 = 0.2$ for $\mathcal{A} = 2$ and at $s^2 = 0.5$ for $\mathcal{A} = 5$. At an even larger values of s^2 it even outperformed the CKGI. With s^2 sufficiently increased, it was preferable to use EGGI as seen with IM1. It is important to note that s^2 had to be drastically increased to achieve this; using $s^2 = 2$ is a 20-fold increase from the original s^2=0.1. Figure 3.3.7 and Figure 3.3.8 also revealed that the KG and CKGI kept their advantage over the EGGI for longer as s^2 was increased for $\mathcal{A} = 5$ than they did for $\mathcal{A} = 2$. This is quite intuitive; when there are more arms to chose the best from, making use of the posterior distributions of both P and T is beneficial.

It is possible to choose an s^2 large enough that looking only one step ahead the way the knowledge gradient and the CKGI do becomes simply not good enough. For $\mathcal{A} = 2$ and

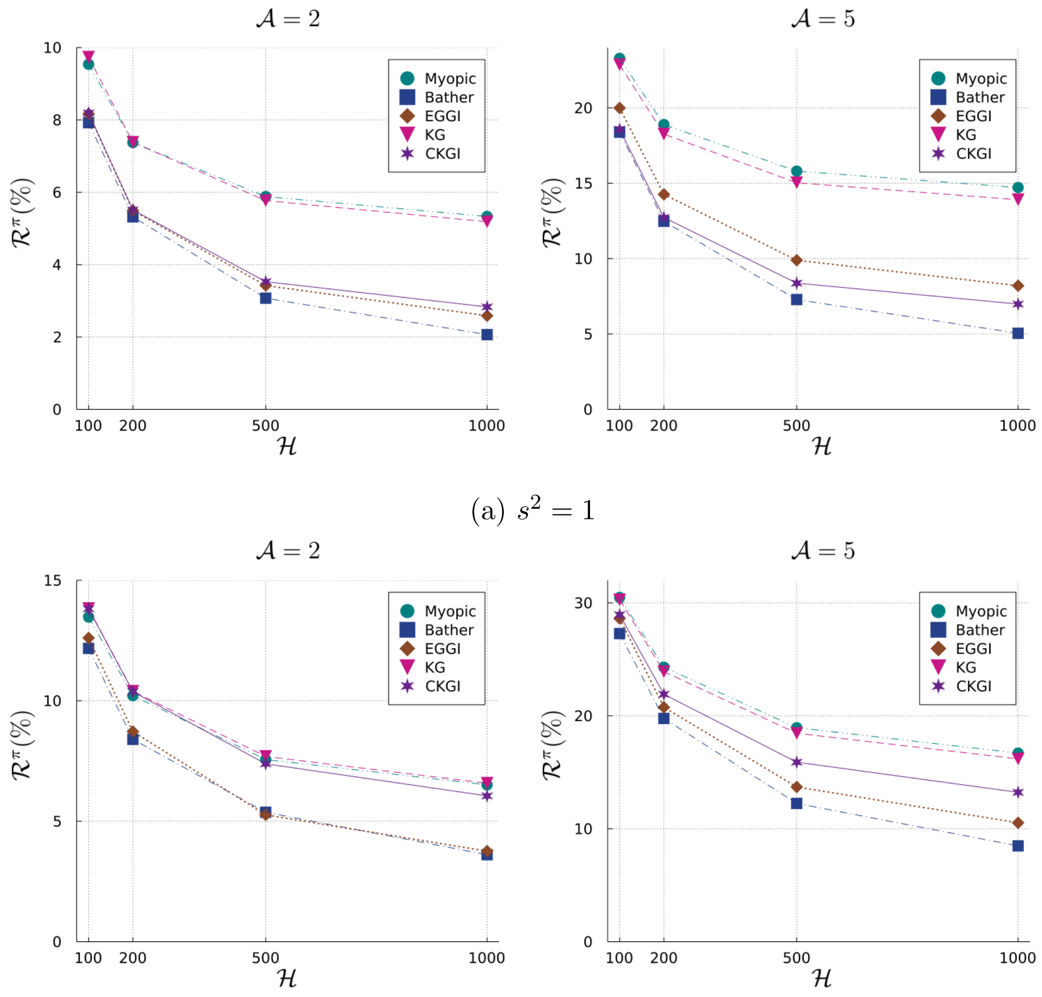


(a) $s^2 = 0.2$

(b) $s^2 = 0.5$

Figure 3.3.7: Performance of the policies in the extended study for IM2, with increasing s^2. Part 1, showing $s^2 = 0.2$ and $s^2 = 0.5$.

$s^2 = 2$ the myopic index produces lower regrets than the KG for all observed horizons, and lower regrets than the CKGI for $\mathcal{H} = 100$ and $\mathcal{H} = 200$. While the knowledge gradient lost its advantage over the myopic index around $s^2 = 1$ for both $\mathcal{A} = 2$ and $\mathcal{A} = 5$, the CKGI still performed markedly better than the myopic index for all but the $\mathcal{A} = 2$, $s^2 = 2$ case.



(a) $s^2 = 1$

(b) $s^2 = 2$

Figure 3.3.8: Performance of the policies in the extended study for IM2, with increasing s^2. Part 2, showing $s^2 = 1$ and $s^2 = 2$.

3.4 The Intelligence Problem with Dependent Parameters

The work in Section 3.4 follows on from having developed models and solutions in Section 3.3 where the relevance of tips was independent from the time it took to evaluate them. Here we wish to establish models where the relevance of a tip depends on its evaluation time.

3.4.1 Model Description

We model the intelligence collection process very similarly to Section 3.3, but consider models where the relevance probability of a tip p^a is a function of T^a, and changes across observations depending on the realisation of $T^a = t^a$ via the logistic link function. This encapsulates the idea that evaluating a relevant and a nuisance tip may take different amounts of time; evaluating a relevant tip is a longer process than determining a nuisance tip to be irrelevant. To characterise arm a, the expected reward rate $\mathbb{E}\left[p^a/T^a\right]$ remains the metric of interest.

The tips from source a, $a \in \{1, ..., \mathcal{A}\}$, form a sequence of IID Bernoulli random variables, $X^a \sim \text{Bernoulli}(p^a)$, therefore the jth outcome from arm a is $x_j^a \in \{0, 1\}$, and the tip is deemed relevant with $\mathbb{P}(x_j^a = 1) = p^a$ or a nuisance tip with $\mathbb{P}(x_j^a = 0) = 1 - p^a$. The time required to evaluate a tip from source a, determining which category it falls into is represented by random variable T^a, with realisations $t^a \in (0, \infty)$. The observed outcomes of X^a and T^a are recorded in the vectors $\boldsymbol{x}^a$ and $\boldsymbol{t}^a$ respectively. As neither the parameters of the logistic link function or the distribution of T^a are known initially, we learn about them by making observations.

We consider two model variants, which differ only in the distribution of T^a, namely *Dependent Model 1* (DM1) and *Dependent Model 2* (DM2). Furthermore, these models only differ from those in Section 3.3.1 in the distributions of the p^a's. We describe the models focusing on a single arm only, and therefore drop the superscript denoting the arm in question.

Dependent Model 1

Same as in IM1, the evaluation times under DM1 are given by the sum of a known, deterministic minimum and random excess evaluation time so that $T = t^{\min} + \widetilde{T}$. The distribution of $\widetilde{T}$ is that of a Gamma-Exponential, given by (3.3.6) and (3.3.7), and the Bayesian updating of the parameters are shown in (3.3.8) and (3.3.9). As stated before, the relevance of a tip p depends on it evaluation time via the logistic link function,

$$X \sim \text{Bernoulli}\left(p(\widetilde{t})\right), \tag{3.4.1}$$

$$p(\widetilde{t}) = \frac{1}{1+\exp\left(-(B_1 + B_2\widetilde{t})\right)}, \tag{3.4.2}$$

$$B_1, B_2 \mid \boldsymbol{x}, \widetilde{\boldsymbol{t}} \sim \text{BvNormal}(\boldsymbol{m}, \boldsymbol{V}). \tag{3.4.3}$$

where B_1 and B_2 are unknown coefficients which we estimate in a Bayesian fashion. The logistic link function was chosen as a it provides a straightforward approach to translate measures of time to probabilities between 0 and 1. As no conjugate models exist for such purpose, the Bayesian updating procedure is not trivial and will be discussed at the end of this section. As seen before, the state of an arm is given by its posterior parameters, in this case by $(\boldsymbol{m}, \boldsymbol{V}, \gamma, \delta)$.

To evaluate $\mathbb{E}\left[p/T\right]$, we need to integrate over B_1, B_2, Λ and $\widetilde{t}$. It is found as

$$\begin{aligned}\mathbb{E}\left[\frac{p}{T}\right] = &\int_0^\infty \int_0^\infty \int_{-\infty}^\infty \int_{-\infty}^\infty \frac{1}{\widetilde{t}+t^{\min}} \frac{1}{1+\exp\left(-\left(B_1+B_2\widetilde{t}\right)\right)} \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} \\ &\exp\left(-\frac{1}{2(1-\rho^2)}\left[\frac{(B_1-m_1)^2}{\sigma_1^2} - \frac{2\rho(B_1-m_1)(B_2-m_B)}{\sigma_1\sigma_2} + \frac{(B_2-m_B)^2}{\sigma_2^2}\right]\right) \\ &\Lambda \exp(-\Lambda\widetilde{t}) \frac{\delta^\gamma}{\Gamma(\gamma)} \Lambda^{\gamma-1} \exp(-\delta\Lambda)\ \mathrm{d}B_1 \mathrm{d}B_2 \mathrm{d}\Lambda \mathrm{d}\widetilde{t}\end{aligned}$$

where m_1 and m_2 are the elements of the mean vector $\boldsymbol{m}$ of the Bivariate Normal posterior of B_1 and B_2, while variances $\sigma_1^2 = V_{1,1}$, $\sigma_2^2 = V_{2,2}$ and correlation $\rho = \frac{V_{1,2}}{\sqrt{V_{1,1}V_{2,2}}}$ are found from their covariance matrix $\boldsymbol{V}$.

Jaakkola and Jordan (2000) provides a simplification that allows the integral over B_1 and B_2 to be reduced to a one dimensional integral

$$\begin{aligned}&\int_{-\infty}^\infty \int_{-\infty}^\infty \frac{1}{1+\exp\left(-\left(B_1+B_2\widetilde{t}\right)\right)} \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} \\ &\exp\left(-\frac{1}{2(1-\rho^2)}\left[\frac{(B_1-m_1)^2}{\sigma_1^2} - \frac{2\rho(B_1-m_1)(B_2-m_B)}{\sigma_1\sigma_2} + \frac{(B_2-m_B)^2}{\sigma_2^2}\right]\right) \mathrm{d}B_1 \mathrm{d}B_2 \\ &= \int_{-\infty}^\infty \frac{1}{1+\exp(-\theta)} \frac{1}{\sqrt{2\pi\sigma_{\text{eff}}^2}} \exp\left(-\frac{(\theta-\mu_{\text{eff}})^2}{2\sigma_{\text{eff}}^2}\right) \mathrm{d}\theta,\end{aligned} \tag{3.4.4}$$

where μ_{eff} and σ^2_{eff} are the effective mean and variance, given by

$$\begin{aligned} \mu_{\text{eff}} &= m_1 + m_B\widetilde{t}, \\ \sigma^2_{\text{eff}} &= \sigma_1^2 + 2\rho\sigma_1\sigma_2\widetilde{t} + \sigma_2^2\widetilde{t}^2. \end{aligned}$$

Using the above simplification and taking the integral with respect to Λ, the expectation $\mathbb{E}\left[p/T\right]$is reduced to

$$\mathbb{E}\left[\frac{p}{T}\right] = \int_0^\infty \int_{-\infty}^\infty \frac{1}{\widetilde{t}+t^{\min}} \frac{1}{1+\exp\left(-\theta\right)} \frac{1}{\sqrt{2\pi\sigma^2_{\text{eff}}}} \exp\left(-\frac{(\theta-\mu_{\text{eff}})^2}{2\sigma^2_{\text{eff}}}\right) \frac{\gamma\delta^\gamma}{(\widetilde{t}+\delta)^{\gamma+1}}\, \mathrm{d}\theta \mathrm{d}\widetilde{t}, \tag{3.4.5}$$

which can be determined using numerical integration. The expectations found using the simplification from Jaakkola and Jordan (2000) were compared to those found via the full integrals and have found both to be accurate approximations.

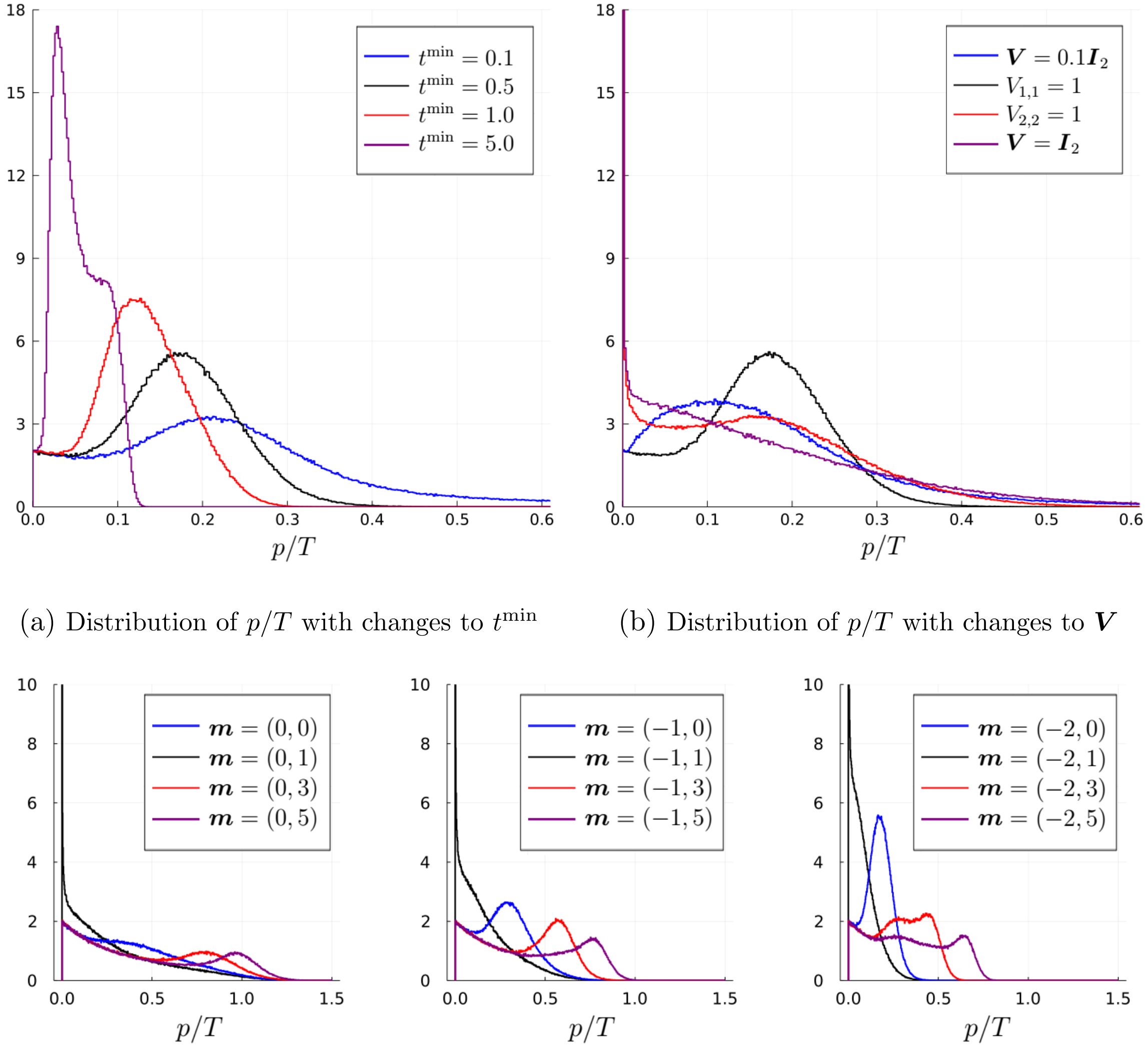
Sampling provides an alternative way to obtain the expectation and visualise the distribution of p/T. To generate a sample:

1. Sample Λ_j from Gamma(γ, δ).
2. Sample $\widetilde{t}_j$ from Exponential(Λ_j).
3. Sample B_{1j} and B_{2j} from BvNormal$(\boldsymbol{m}, \boldsymbol{V})$.
4. Then $p_j = \dfrac{1}{1+\exp\left(-(B_{1j}+B_{2j}\widetilde{t}_j)\right)}$.
5. Finally, the jth sample is given by $\dfrac{p_j}{\widetilde{t}_j + t^{\min}}$.

The expected value of p/T is then given by the sample mean.

Visualising the distribution of p/T with different values of $\boldsymbol{m}$, $\boldsymbol{V}$ and $t^{\min}$ showed that it is very sensitive to the values of these parameters, and that multiple modes may be present. These findings can be seen in Figure 3.4.1.

Note that in the following investigations whenever $\mathrm{E}\left[p/T\right]$ is required (3.4.5) is used, not sampling. Sampling was only used for visualisation of the distribution of p/T.



(a) Distribution of p/T with changes to $t^{\min}$

(b) Distribution of p/T with changes to $\boldsymbol{V}$

(c) Distribution of p/T with changes to $\boldsymbol{m}$

Figure 3.4.1: The shape of the distribution of p/T, dependent on $t^{\min}$, $\boldsymbol{m}$ and $\boldsymbol{V}$ under DM1. If not stated on the legend, the parameter values are $\gamma = 1, \delta = 2$, $t^{\min} = 0.5$, $\boldsymbol{m} = (-2, 1)$ and $\boldsymbol{V} = 0.1\boldsymbol{I}_2$.

Dependent Model 2

The evaluation times in DM2 are the same as in IM2 and follow a lognormal distribution so that $T = \exp(\widetilde{T})$ where $\widetilde{T}$ follows a Normal distribution according to (3.3.12) and (3.3.13), and the posterior parameters are found via (3.3.15) and (3.3.14). The outcome of the evaluation is distributed the same way as it was in DM1, given by (3.4.1)-(3.4.3). Under these conditions the state of an arm is given by $(\boldsymbol{m}, \boldsymbol{V}, \mu, \sigma^2)$.

We defer discussion regarding the Bayesian updating procedure to the end of the section.

The expectation $\mathrm{E}\left[\frac{p}{T}\right]$ is found as

$$\begin{aligned}\mathbb{E}\left[\frac{p}{T}\right] = &\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{1}{\exp(\widetilde{t})} \frac{1}{1+\exp\left(-\left(B_1+B_2\widetilde{t}\right)\right)} \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} \\ &\exp\left(-\frac{1}{2(1-\rho^2)}\left[\frac{(B_1-m_1)^2}{\sigma_1^2} - \frac{2\rho(B_1-m_1)(B_2-m_B)}{\sigma_1\sigma_2} + \frac{(B_2-m_B)^2}{\sigma_2^2}\right]\right) \\ &\frac{1}{\sqrt{2\pi\sigma'^2}}\exp\left(-\frac{(\widetilde{t}-\Upsilon)^2}{2s^2}\right)\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(\Upsilon-\mu)^2}{2\sigma^2}\right)\,\mathrm{d}B_1\mathrm{d}B_2\mathrm{d}\Upsilon\mathrm{d}\widetilde{t}.\end{aligned}$$

Taking the integral with respect to Υ and using the simplification from Jaakkola and Jordan (2000) as seen before in (3.4.4), the formula reduces to

$$\begin{aligned}\mathbb{E}\left[\frac{p}{T}\right] = &\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{1}{\exp(\widetilde{t})}\frac{1}{1+\exp(-\theta)}\frac{1}{\sqrt{2\pi\sigma_{\text{eff}}^2}}\exp\left(-\frac{(\theta-\mu_{\text{eff}})^2}{2\sigma_{\text{eff}}^2}\right) \\ &\frac{1}{\sqrt{2\pi\left(\sigma^2+\sigma'^2\right)}}\exp\left(-\frac{(\widetilde{t}-\mu)^2}{2\left(\sigma^2+\sigma'^2\right)}\right)\,\mathrm{d}\theta\mathrm{d}\widetilde{t}.\end{aligned} \tag{3.4.6}$$

Again, sampling provides an alternative way to obtain the expectation $\mathrm{E}[p/T]$. To generate a sample:

1. Sample Υ_j from Normal(μ, σ^2).

2. Sample $\widetilde{t}_j$ from Normal(Υ, s^2).

3. Sample B_{1j} and B_{2j} from BvNormal$(\boldsymbol{m}, \boldsymbol{V})$.

4. Then $p_j = \dfrac{1}{1+\exp\left(-(B_{1j}+B_{2j}\widetilde{t}_j)\right)}$.

5. Finally, the jth sample is given by $\dfrac{p_j}{\exp(\widetilde{t}_j)}$.

The expected value of p/T is given by the sample mean.

Similarly to DM1, the distribution of p/T under this model was sensitive to the choice of parameters, as seen in Figure 3.4.2.

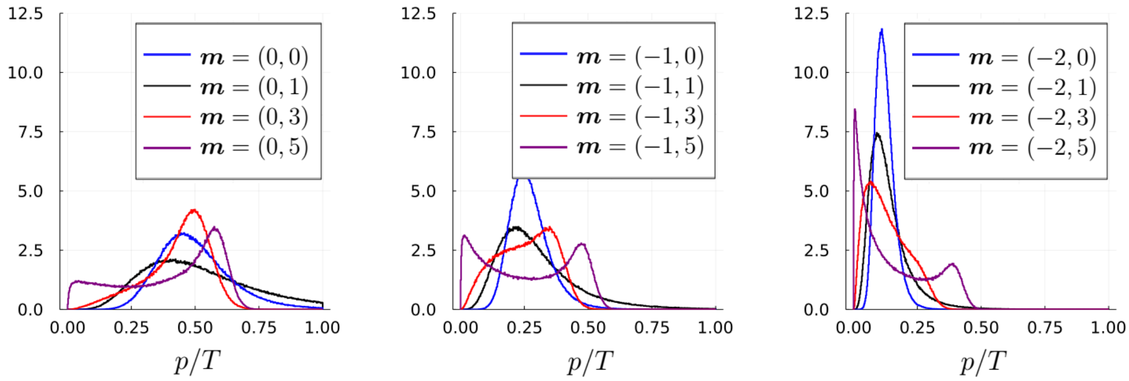


Figure 3.4.2: The shape of the distribution of p/T, dependent on $\boldsymbol{m}$ under DM2. If not stated on the legend, the parameter values are $\mu = 0, \sigma^2 = 0.1$, $s^2 = 0.1$ and $\boldsymbol{V} = 0.1\boldsymbol{I}_2$.

Bayesian Learning of the Logistic Coefficients

Since p is a function of either the excess or log-evaluation time $\widetilde{T}$ via the logistic link function, the parameters of the logistic regression B_1 and B_2 need to be estimated. As we set out to do so in a Bayesian fashion, a joint Bivariate Normal prior with mean vector $\boldsymbol{m}_{(0)}$ and covariance matrix $\boldsymbol{V}_{(0)}$ is placed on B_1 and B_2. While these are not conjugate priors, no options for such are available, but there are multiple methods that use a normal approximation. For completeness of discussion we repeat the summary of the Bayesian learning problem;

$$\begin{aligned} X &\sim \text{Bernoulli}(p(\widetilde{t})), \\ p(\widetilde{t}) &= \frac{1}{1 + \exp(-(B_1 + B_2\widetilde{t}))}, \\ B_1, B_2 &\sim \text{BvNormal}(\boldsymbol{m}_{(0)}, \boldsymbol{V}_{(0)}), \\ B_1, B_2 \mid \boldsymbol{x}, \widetilde{\boldsymbol{t}} &\sim \text{BvNormal}(\boldsymbol{m}_{(n)}, \boldsymbol{V}_{(n)}). \end{aligned}$$

The methods considered for updating the parameters are an approximation based on the Laplace method, described in Spiegelhalter and Lauritzen (1990), which will be referred to as the SL approximation, and the variational Bayesian approach discussed in Jaakkola and Jordan (2000).

We denote the logistic function by $s(y)$ so that

$$s(y) = \text{logistic}(y) = \frac{1}{1+\exp(-y)}.$$

Under the S-L approximation, the updates for the mean vector and covariance matrix resulting from absorbing observation i are as follows.

$$\boldsymbol{V}^{-1}{}_{(j)} = \boldsymbol{V}^{-1}{}_{(j-1)} + s\left(\boldsymbol{m}_{(j-1)}{}^{\mathsf{T}}\boldsymbol{\mathcal{T}}_{(j)}\right)\left[1 - s\left(\boldsymbol{m}_{(j-1)}{}^{\mathsf{T}}\boldsymbol{\mathcal{T}}_{(j)}\right)\right]\boldsymbol{\mathcal{T}}_{(j)}\,\boldsymbol{\mathcal{T}}_{(j)}{}^{\mathsf{T}}, \tag{3.4.7}$$

$$\boldsymbol{m}_{(j)} = \boldsymbol{m}_{(j-1)} + \left(x_{(j)} - s\left(\boldsymbol{m}_{(j-1)}{}^{\mathsf{T}}\boldsymbol{\mathcal{T}}_{(j)}\right)\right)\boldsymbol{V}_{(j)}\boldsymbol{\mathcal{T}}, \tag{3.4.8}$$

where $\boldsymbol{\mathcal{T}}_{(j)} = \left(1, \widetilde{t}_{(j)}\right)^{\mathsf{T}}$, and where $x_{(j)}$ and $\widetilde{t}_{(j)}$ represent the jth paired observation of X and $\widetilde{T}$.

Even though (3.4.7) and (3.4.8) fully define the updates, (3.4.7) can be rewritten using the Sherman-Morrison formula (Sherman and Morrison, 1950) to avoid having to take the inverse of the covariance matrix as part of the updating procedure. Then $\boldsymbol{V}$ is found via

$$\boldsymbol{V}_{(j)} = \boldsymbol{V}_{(j-1)} - \frac{s\left(\boldsymbol{m}_{(j-1)}{}^{\mathsf{T}}\boldsymbol{\mathcal{T}}_{(j)}\right)\left[1 - s\left(\boldsymbol{m}_{(j-1)}{}^{\mathsf{T}}\boldsymbol{\mathcal{T}}_{(j)}\right)\right]\left(\boldsymbol{V}_{(j-1)}\boldsymbol{\mathcal{T}}_{(j)}\right)\left(\boldsymbol{\mathcal{T}}_{(j)}{}^{\mathsf{T}}\boldsymbol{V}_{(j-1)}\right)}{1 + s\left(\boldsymbol{m}_{(j-1)}{}^{\mathsf{T}}\boldsymbol{\mathcal{T}}_{(j)}\right)\left[1 - s\left(\boldsymbol{m}_{(j-1)}{}^{\mathsf{T}}\boldsymbol{\mathcal{T}}_{(j)}\right)\right]\boldsymbol{\mathcal{T}}_{(j)}{}^{\mathsf{T}}\boldsymbol{V}_{(j-1)}\boldsymbol{\mathcal{T}}_{(j)}}. \tag{3.4.9}$$

The updates of the SL approximation are straightforward, using only the data and the prior parameters. On the other hand, the variational Bayesian method includes a new, variational parameter, resulting in a more complex but at the same time flexible approach.

The logistic function has a quadratic variational lower bound given by

$$s(y) \geq s(\xi)\exp\left(\frac{y-\xi}{2} - \eta(\xi)(y^2 - \xi^2)\right),$$

where

$$\eta(\xi) = \frac{1}{2\xi}\left(\frac{1}{2} - s(\xi)\right),$$

and ξ is the variational parameter. Then the parameters of the Bivariate Normal prior can be updated according to

$$\begin{aligned} \boldsymbol{V}^{-1}{}_{(j)} &= \boldsymbol{V}^{-1}{}_{(j-1)} + 2\eta(\xi)\boldsymbol{\mathcal{T}}_{(j)}\ \boldsymbol{\mathcal{T}}_{(j)}{}^{\mathsf{T}}, \\ \boldsymbol{m}_{(j)} &= \boldsymbol{V}_{(j)}\left[\boldsymbol{V}^{-1}{}_{(j-1)}\,\boldsymbol{m}_{(j-1)} + \left(x_{(j)} - \frac{1}{2}\right)\boldsymbol{\mathcal{T}}_{(j)}\right]. \end{aligned}$$

Analogously to (3.4.9), we can use the Sherman-Morrison formula to find

$$\boldsymbol{V}_{(j)} = \boldsymbol{V}_{(j-1)} - \frac{2\eta(\xi)\,\boldsymbol{V}_{(j-1)}\,\boldsymbol{\mathcal{T}}_{(j)}\,\boldsymbol{\mathcal{T}}_{(j)}{}^{\mathsf{T}}\,\boldsymbol{V}_{(j-1)}}{1 + 2\eta(\xi)\,\boldsymbol{\mathcal{T}}_{(j)}{}^{\mathsf{T}}\,\boldsymbol{V}_{(j-1)}\,\boldsymbol{\mathcal{T}}_{(j)}},$$

and compute both $\boldsymbol{V}^{-1}{}_{(j)}$ and $\boldsymbol{V}_{(j)}$ in parallel. In addition, it is useful to consider updating $\ln|\,\boldsymbol{V}_{(j)}\,|$ instead of calculating it for every iteration

$$\ln|\,\boldsymbol{V}_{(j)}\,| = \ln|\,\boldsymbol{V}_{(j-1)}\,| - \ln\left(1 + 2\eta(\xi)\,\boldsymbol{\mathcal{T}}_{(j)}{}^{\mathsf{T}}\,\boldsymbol{V}_{(j-1)}\,\boldsymbol{\mathcal{T}}_{(j)}\right),$$

as this quantity is needed to evaluate the log of the variational lower bound $\mathcal{L}(\xi)$, found in Drugowitsch (2013)

$$\mathcal{L}(\xi) = \frac{1}{2}\boldsymbol{m}_{(j)}{}^{\mathsf{T}}\,\boldsymbol{V}^{-1}{}_{(j)}\,\boldsymbol{m}_{(j)} + \frac{1}{2}\ln|\,\boldsymbol{V}_{(j)}\,| + \ln(s(\xi)) - \frac{\xi}{2} + \eta(\xi)\xi^2,$$

that we aim to maximise.

The variational parameter ξ also requires estimation. It can be found using expectation maximisation of the variational bound, resulting in a closed form update

$$\xi^2 = \boldsymbol{\mathcal{T}}_{(j)}{}^{\mathsf{T}}\boldsymbol{V}_{(j)}\boldsymbol{\mathcal{T}}_{(j)} + \left(\boldsymbol{\mathcal{T}}_{(j)}{}^{\mathsf{T}}\boldsymbol{m}_{(j)}\right)^2.$$

Every time a new observation is made, the variational parameter is set to zero so that $\xi_{(j)0} = 0$ with $\eta = 1/8$, and the initial updates of $\boldsymbol{V}_{(j)}$, $\boldsymbol{V}^{-1}{}_{(j)}$, $\ln|\,\boldsymbol{V}_{(j)}\,|$ and $\boldsymbol{m}_{(j)}$ are found based on those values. We then iterate between calculating ξ based on the current values of $\boldsymbol{V}_{(j)}(\xi)$, $\boldsymbol{V}^{-1}{}_{(j)}(\xi)$, $\ln|\,\boldsymbol{V}_{(j)}(\xi)|$ and $\boldsymbol{m}_{(j)}(\xi)$, and using its newly obtained value to re-calculate $\boldsymbol{V}_{(j)}(\xi)$, $\boldsymbol{V}^{-1}{}_{(j)}(\xi)$, $\ln|\,\boldsymbol{V}_{(j)}(\xi)|$ and $\boldsymbol{m}_{(j)}(\xi)$. The next iteration of the procedure will then use these values to obtain an updated value for ξ. We continue this procedure until the variational bound plateaus, so that $\mathcal{L}(\xi_{\text{new}}) - \mathcal{L}(\xi_{\text{old}}) < 10^{-5}\mathcal{L}(\xi_{\text{new}})$, at which point we arrive at the final variational

Bayesian estimate of $\boldsymbol{m}$ and $\boldsymbol{V}$, and can move onto the next observation when it becomes available.

The priors suggested in Drugowitsch (2013) are

$$\boldsymbol{m}_{(0)} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \qquad \boldsymbol{V}_{(0)} = \begin{pmatrix} 1/2 & 0 \\ 0 & 1/2 \end{pmatrix}.$$

Table 3.4.1 shows the computational effort required to obtain the posterior values of $\boldsymbol{m}$ and $\boldsymbol{V}$, and the expectation $\mathrm{E}\left[p^a/T^a\right]$. As we can see, computing the required expected values is quite costly, which is due to the necessity of multidimensional integration.

	Expectation		**Updating**	
	SL	VB	SL	VB
M1D	22.45 ms	4.33 ms	6.97 μs	115.04 μs
M2D	38.78 ms	4.52 ms	7.41 μs	139.25 μs

Table 3.4.1: Computation time of the expectation $\mathrm{E}\left[p^a/T^a\right]$ and that of the updating procedure used for $\boldsymbol{m}$ and $\boldsymbol{V}$ with SL and VB.

A further, even more significant downside of both of the methods described is that to achieve estimates close to the true parameters, a large number of observations are required. As we have not addressed this weakness further, we see the effect it has on the performance of the policies in Section 3.4.4, where it is discussed further.

3.4.2 Policies

In this section the applicability of the five policies proposed for the independent models to the dependent models is considered.

Myopic Policy

The first policy that was considered is a simple heuristic that always chooses the arm with the highest expected reward. At every decision time arm a^* is continued, where

$$a^* := \arg\max_a \mathbb{E}\left[\frac{p^a}{T^a}\right]. \tag{3.4.10}$$

This is the same myopic policy of pure exploitation that was applied to the independent models, and could be applied without change to the dependent models.

Bather Index

The Bather Index, also known as the Randomised Allocation Index, is similarly straightforward to apply in the dependent case. The arm continued at the current decision time is given by

$$a^* := \arg\max_a \mathbb{E}\left[\frac{p^a}{T^a}\right] + b_1 b_2^{n^a} Y, \tag{3.4.11}$$

where Y is an Exponential(1) random variable, $b_1 \geq 0$ is a scaling term and $0 \leq b_2 \leq 1$ governs how fast the random term decreases with with the number of observations n^a.

The parameters b_1 and b_2 determine the balance between exploration and exploitation. The scale parameter b_1 sets the upper limit of the initial random perturbation, and the larger b_2 is, the the faster the perturbation term tends to 0. The best values can be determined using a grid-search over a reasonable range of parameters. Since evaluating the expected reward rate in the dependent case requires more computational effort than in the independent case, conducting a grid search to find the best parameter combination in the search space also requires a significantly larger computational investment for the dependent models. To address this, we propose a method to find good exploration parameters that makes use of the independent models in Section 3.3.

First we note that the distribution of evaluation times of IM1 are the same as those of DM1, and similarly the evaluation times of IM2 and DM2 are identically distributed. Next, the distribution of the time dependent success probability $p(T)$ is approximated as a time independent mixture of symmetric Beta distributions. Then good exploration parameters can be found by conducting a grid search using the corresponding independent model, but instead of sampling the success probabilities from their prior distributions to generate observations with, they are sampled from the Beta mixture based on the dependent model. The parameters relating to the evaluation times are still sampled from their respective priors.

To generate a sample of $p(T)$s that the mixture distribution can be fitted to, Λ or Υ alongside (B_1, B_2) are sampled from their priors. The number of samples this parameter combination contributes is given by the expected number of observations over a horizon of 100. Then a single sample of $p(T)$ is found by sampling an evaluation time T and using it together with B_1 and B_2 to calculate $p(T)$. The above process is repeated 10000 times to produce a large sample. Then the parameters of the mixture model are determined using a maximum likelihood estimate.

We refer to this method as the Bather Index with Approximate Parameters, BIAP for short.

Expected Generalised Gittins Index

The Expected Generalised Gittins Index (EGGI) rule was developed based on the assumption that the evaluation times and the probability of a positive outcome are independent which is explicitly not the case here. Therefore, we cannot use this policy.

Knowledge Gradient

The knowledge gradient rule for multi-armed bandits assumes that while rewards are collected till the end of the horizon, there is only one more opportunity to learn, and after that we will cease to update our beliefs. The knowledge gradient policy then chooses one of the arms to learn about as to maximise the expected total reward over the horizon. The extension of the above for multi-armed bandits with random evaluation times was initially defined with dependence between p and T assumed and therefore there is no need to further adapt the policy. The policy always plays arm a^*, so that

$$a^* := \arg\max_a R^a_{\text{total}}. \tag{3.4.12}$$

This total expected reward R^a_{total} is approximated as

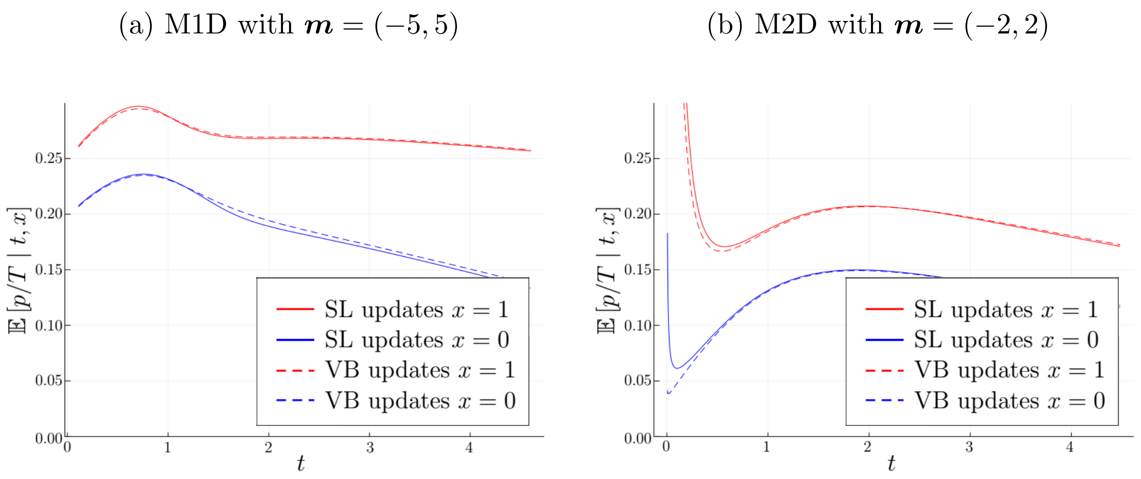
$$\begin{aligned}
R^a_{\text{total}} &= \mathbb{E}\left[p^a \mid \boldsymbol{x}^a, \boldsymbol{t}^a\right] + R^a_{t\in\tau^a_0} + R^a_{t\notin\tau^a_0} + R^a_{t\in\tau^a_1} + R^a_{t\notin\tau^a_1} && (3.4.13)\\
&= \mathbb{E}\left[p^a \mid \boldsymbol{x}^a, \boldsymbol{t}^a\right] + \int_{t\in\tau^a_0} [H-t]^+ \,\mathbb{E}\left[\frac{p^a}{T^a} \mid \boldsymbol{x}^a, \boldsymbol{t}^a, 0, t\right] \mathrm{d}\pi\left(0, t \mid \boldsymbol{x}^a, \boldsymbol{t}^a\right)\\
&\quad + \max_{a'\neq a}\mathbb{E}\left[\frac{p^{a'}}{T^{a'}} \mid \boldsymbol{x}^{a'}, \boldsymbol{t}^{a'}\right] \int_{t\notin\tau^a_0} [H-t]^+ \,\mathrm{d}\pi\left(0, t \mid \boldsymbol{x}^a, \boldsymbol{t}^a\right)\\
&\quad + \int_{t\in\tau^a_1} [H-t]^+ \,\mathbb{E}\left[\frac{p^a}{T^a} \mid \boldsymbol{x}^a, \boldsymbol{t}^a, 1, t\right] \mathrm{d}\pi\left(1, t \mid \boldsymbol{x}^a, \boldsymbol{t}^a\right)\\
&\quad + \max_{a'\neq a}\mathbb{E}\left[\frac{p^{a'}}{T^{a'}} \mid \boldsymbol{x}^{a'}, \boldsymbol{t}^{a'}\right] \int_{t\notin\tau^a_1} [H-t]^+ \,\mathrm{d}\pi\left(1, t \mid \boldsymbol{x}^a, \boldsymbol{t}^a\right), && (3.4.14)
\end{aligned}$$

where τ^a_x were sets of outcomes $T^a = t \in \tau^a_x$ that result in arm i having the highest expected reward rate, defined as

$$\tau^a_x = \left\{t; \mathbb{E}\left[\frac{p^a}{T^a} \mid \boldsymbol{x}^a, \boldsymbol{t}^a, x, t\right] \geq \max_{a'\neq a}\mathbb{E}\left[\frac{p^{a'}}{T^{a'}} \mid \boldsymbol{x}^{a'}, \boldsymbol{t}^{a'}\right]\right\}. \tag{3.4.15}$$

In the cases where p and T were independent, each of the sets τ^a_x could be defined by a single threshold, as $\mathbb{E}\left[p^a/T^a \mid \boldsymbol{x}^a, \boldsymbol{t}^a, t, x\right]$ was decreasing in t. However, that is not always the case with either of the two dependent models, as shown in Figure 3.4.3. Instead of a single threshold, there can be multiple values of t where the equality $\mathbb{E}\left[p^a/T^a \mid \boldsymbol{x}^a, \boldsymbol{t}^a, x, t\right] = \max_{a'\neq a}\mathbb{E}\left[p^{a'}/T^{a'} \mid \boldsymbol{x}^{a'}, \boldsymbol{t}^{a'}\right]$ holds, which can define start or end points of sections that make up τ^a_x.

Since there is no closed form formula for $\mathbb{E}\left[p^a/T^a \mid \boldsymbol{x}^a, \boldsymbol{t}^a, x, t\right]$, such values can only be found numerically, but numerical root-finding algorithms give no guarantee of finding all roots when multiple are present, and the values found will depend on the starting point given to them. Therefore τ^a_x is incredibly challenging to find, which prohibits the calculation of R^a_{total}.


(a) M1D with $\boldsymbol{m} = (-5, 5)$

(b) M2D with $\boldsymbol{m} = (-2, 2)$

Figure 3.4.3: $\mathbb{E}[p/T \mid t, x]$ after observing $T = t$ and $X = x$. All other parameters are assumed to be prior parameters with values as stated in Section 3.4.4.

Calibrated Knowledge Gradient Index

As the Calibrated Knowledge Gradient Index (CKGI) requires the knowledge gradient type expected reward R^a_{total}, which we could not obtain, we cannot apply this policy either.

Ideas for approximate KG and CKGI

As discussed above, exact implementation of the knowledge gradient and the calibrated knowledge gradient policies is not feasible. However, if the dependent posteriors could be effectively approximated by independent posteriors we may be able to apply some version of these policies.

In the dependent models the distributions used for T are the same as for the independent models and is no need for approximations. On the other hand the success probability has a Beta prior and posterior in the independent models but is a function of the evaluation time in the dependent models. To approximate $p^a(T^a)$ with a Beta distribution, we can match its expectation and variance to a Beta(α, β) distribution with

$$\alpha = -\mathrm{E}\left[p^a(T^a)\right] \frac{\mathrm{Var}(p^a(T^a)) + \mathrm{E}\left[p^a(T^a)\right]^2 + \mathrm{E}\left[p^a(T^a)\right]}{\mathrm{Var}(p^a(T^a))},$$
$$\beta = \left(\mathrm{E}\left[p^a(T^a)\right] - 1\right) \frac{\left(\mathrm{Var}(p^a(T^a)) + \mathrm{E}\left[p^a(T^a)\right]^2 + \mathrm{E}\left[p^a(T^a)\right]\right)}{\mathrm{Var}(p^a(T^a))},$$

where $\mathrm{E}\left[p^a(T^a)\right]$ and $\mathrm{Var}(p^a(T^a))$ are the expectation and variance of the success probability in the dependent case, found numerically.

An issue with the above approach is that to get a valid Beta distribution both $\alpha, \beta > 0$ must be true, which means

$$\mathrm{Var}(p^a(T^a)) < \mathrm{E}\left[p^a(T^a)\right]\left(1 - \mathrm{E}\left[p^a(T^a)\right]\right)$$

must be satisfied, which we cannot guarantee.

Another option would be to sample from the posterior distribution of $p^a(T^a)$ and fit a Beta distribution to that sample using maximum likelihood estimates. The drawback of this idea is that generating the sample and fitting to it is a time consuming process and would significantly slow the simulations.

Whichever approach is used to obtain an approximating Beta distribution for $p^a(T^a)$, it could then be used along with the distribution of T in the knowledge gradient and calibrated knowledge gradient policies as if we were dealing with he independent models. However, the practicalities of implementing these approximations would need to be further investigated.

3.4.3 Applying the Policies

As discussed in Section 3.4.2, only the myopic policy, the Bather index and the Bather Index with Approximate Parameters (BIAP) can be applied to the dependent models.

To follow the myopic policy, the decision maker needs to know the expected reward

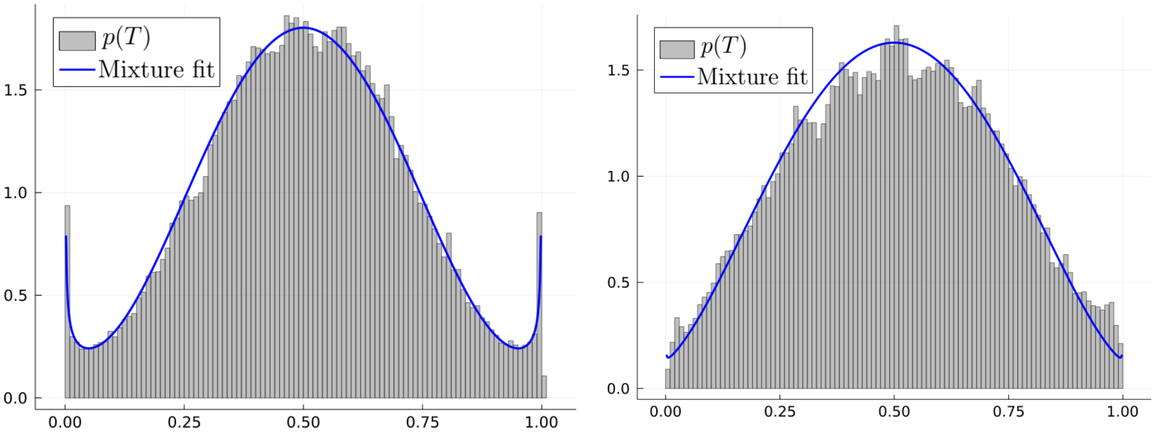
rate $\mathrm{E}\left[p^a/T^a\right]$ of every arm at every decision time, which we calculate using (3.4.5) and (3.4.6) for DM1 and DM2 respectively. Note that the expectation only needs to be recalculated for the arm that has just been observed, the expected reward rates of the arms not observed do not change.
As mentioned previously, the Bather Index is given by the sum of the expected reward rate (calculated the same way as for the myopic policy) and a random term which consists of an Exponential(1) random variable modified by scale parameter b_1 and decay parameter b_2. To find the combination of values for b_1 and b_2 which produces the highest reward a grid search was conducted. For DM1, an 11×11 grid with 0.1 between the grid points was used, so that both the values of b_1 and b_2 fell within the interval $(0, 1)$. For DM2, a 21×11 grid with 0.1 between the grid points was used, so that the values of b_1 fell within $(0, 2)$ and b_2 fell within $(0, 1)$. The range of values the scale parameter b_1 could take was chosen with the intent to ensure the best parameter combination was covered by the grid. The best parameter combinations can be seen in Table 3.4.2.

The BIAP finds good parameter combinations by carrying out the grid search on the independent models with modified distributions, so that the distribution of P^a closely approximates that of $p^a(T^a)$.

The distribution of $p^a(T^a)$ is found via sampling. Since there is only one sample generated, for the rest of this paragraph the superscript a holds no meaning and is therefore omitted. First, B_1, B_2 and either Λ or Υ (dependent on the model) were sampled from their prior. The number of observations this parameter combination will contribute to the sample is given by $\lceil \mathrm{E}\left[\frac{100}{T}\right] \rceil$, the expected number of observations (rounded up) over a horizon of 100. Then for every observation, T is sampled and the resulting $p(T)$ added to the sample. Once observations have been obtained from 10000 parameter combinations, maximum likelihood estimates are used to fit to this sample a mixture distribution of the form

$$w_1\mathrm{Beta}(\alpha_1, \alpha_1) + w_2\mathrm{Beta}(\alpha_2, \alpha_2) + w_3\mathrm{Beta}(\alpha_3, \alpha_3),$$

where the w are the weights of the components, and the α are the two shape parameters which in this case are always equal. This mixture model is used to approximate the distribution of $p(T)$. The sampled distribution of $p(T)$ with the mixture models are shown in Figure 3.4.4 and the parameters of the mixtures are given in Table 3.4.2.



(a) DM1

(b) DM2

Figure 3.4.4: Fitting the Beta mixture to $p(T)$ under DM1 and DM2.

	w_1	α_1	w_2	α_2	w_3	α_3
DM1	0.1469	0.7510	0.8282	3.4451	0.0249	0.1641
DM2	0.0211	0.5939	0.7617	2.7266	0.2173	1.3328

Table 3.4.2: Best fitting shape parameters and their associated weights used in BIAP.

In the independent case of the intelligence problem, P follow a Beta distribution and are independent of T. In the independent approximation of the dependent case the prior for all P is given by a Beta(1,1), same as in the independent case, but P follow and their realisations are sampled from the above determined mixture of Beta distributions as opposed to its prior. Therefore, to generate a dataset on which the grid search can be performed, for each simulation the P are sampled from the appropriate Beta mixture model, and the Λ or Υ from the priors used in the dependent case. Then the grid search for the exploration parameters of the BIAP can be performed the same way as the grid search in independent case, the only difference being a modified dataset.

The results of the grid search for both the Bather index based on the SL approximation and the BIAP are given in Table 3.4.3 for DM1 and Table 3.4.4 for DM2.

	$\mathcal{A} = 2$				$\mathcal{A} = 5$			
	Bather Index		BIAP		Bather Index		BIAP	
$\mathcal{H}$	b_1	b_2	b_1	b_2	b_1	b_2	b_1	b_2
100	0.1	0.7	0.1	0.9	0.1	0.7	0.1	0.8
200	0.2	0.6	0.1	0.9	0.1	0.9	0.2	0.8
500	0.2	0.8	0.2	0.9	0.2	0.8	0.2	0.9
1000	0.2	0.8	0.2	0.9	0.2	0.9	0.2	0.9

Table 3.4.3: Exploration parameters of the Bather Index and the BIAP for DM1

	$\mathcal{A} = 2$				$\mathcal{A} = 5$			
	Bather Index		BIAP		Bather Index		BIAP	
$\mathcal{H}$	b_1	b_2	b_1	b_2	b_1	b_2	b_1	b_2
100	1.4	0.2	1.3	0.3	2.0	0.2	1.6	0.3
200	1.7	0.2	1.6	0.3	2.0	0.3	1.6	0.3
500	1.0	0.3	1.6	0.4	1.6	0.3	0.9	0.5
1000	1.7	0.3	0.7	0.7	2.0	0.3	0.9	0.6

Table 3.4.4: Exploration parameters of the Bather Index and the BIAP for DM2

3.4.4 Numerical Experiments

As discussed in Section 3.4.2, we are only able to apply the myopic policy and the Bather Index rule to the dependent models. The policies were considered using both the Spiegelhalter-Lauritzen (SL) and the variational Bayesian (VB) updates.

The priors used were $\boldsymbol{m} = (0, 0)$, $\boldsymbol{V} = \left(\begin{smallmatrix} 1/2 & 0 \\ 0 & 1/2 \end{smallmatrix}\right)$ for both models, $\gamma^a_{(0)} = 1, \delta^a_{(0)} = 1$

for DM1 and $\mu^a_{(0)} = 0$, $\sigma^{a2}{}_{(0)} = 1$ for DM2, with s^2 given as 0.1. We ran these simulations for a range of horizons and for $\mathcal{A} = 2$ and $\mathcal{A} = 5$ arms. We calculated the Bayes reward for all policies, using 10000 simulated scenarios where the parameters (B^a_1, B^a_2), Λ^a and Υ^a were sampled from their prior distributions.

We compared these policies using the same regret type measure as in Section 3.3. It is given by the relative difference between the Bayes reward of the super-optimal policy π^{SO} and the Bayes reward of policy π, as shown in Section 3.3.4. We consider a policy to achieve a so called super-optimum if it is set by an omniscient decision maker that knows all outcomes and picks a single arm to observe at every decision time, identifying for every instance the arm that produces the highest realised reward over the horizon. The Bayes reward of following the super-optimal policy is obtained the same way as for any other policies. The "SL" and "VB" in the name of a policy shows which approximate method was used to update $\boldsymbol{m}$ and $\boldsymbol{V}$.

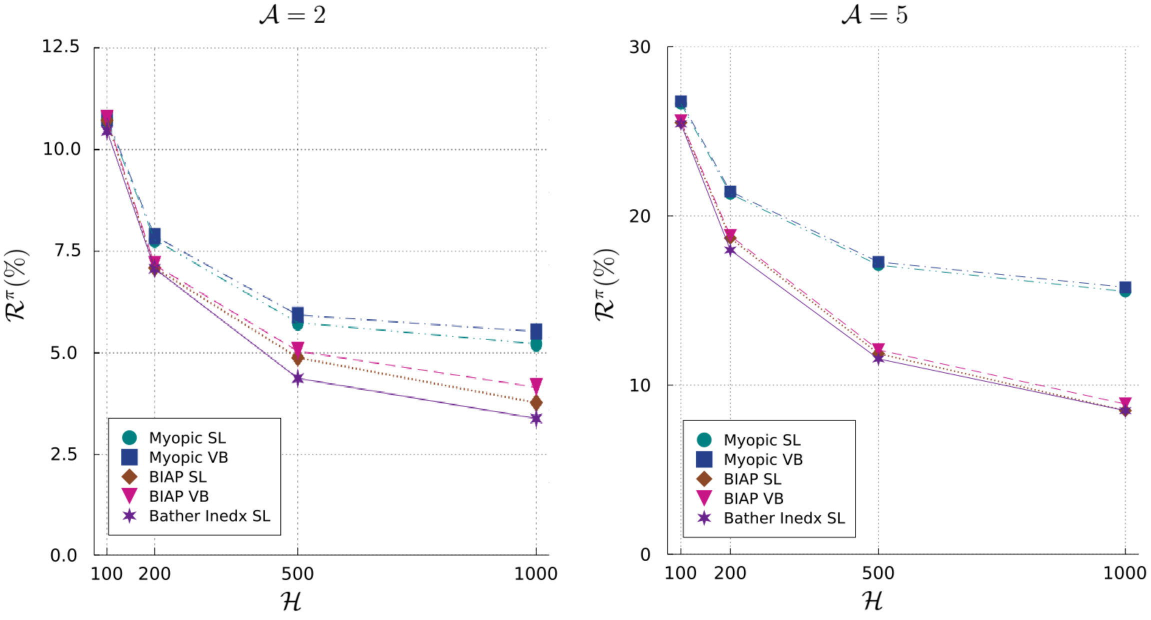


Figure 3.4.5: Relative regrets $\mathcal{R}^\pi$ as defined in (3.3.40), achieved by the policies discussed in Section 3.4.2 in the numerical experiments of DM1.

With one exception, the Bather type policies (Bather index, BIAP) achieve much lower

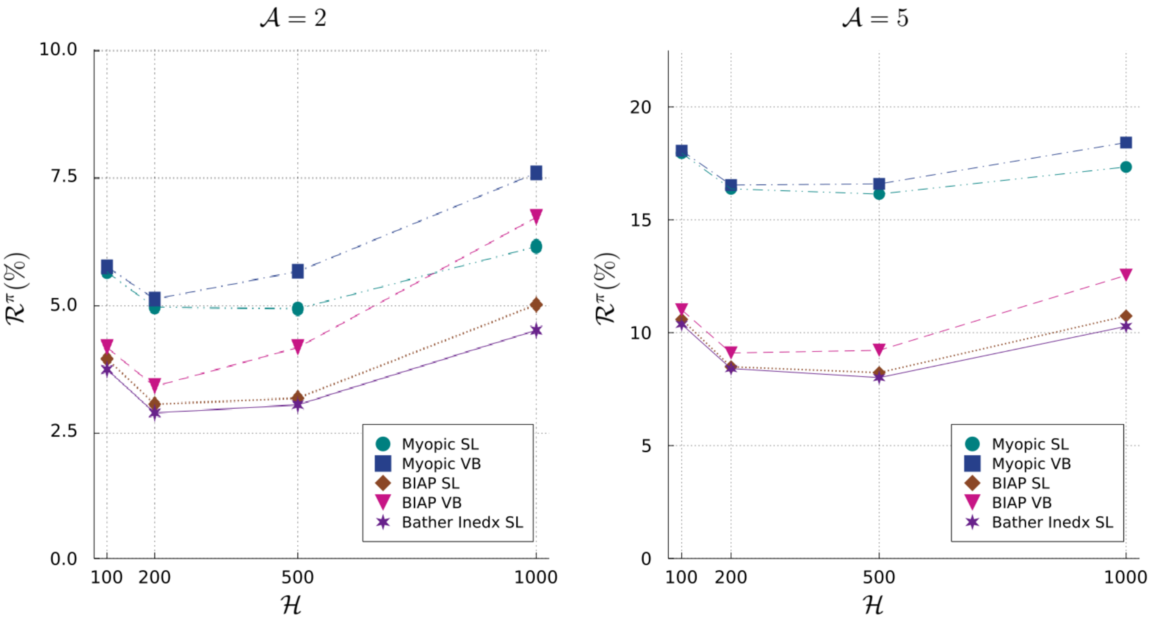
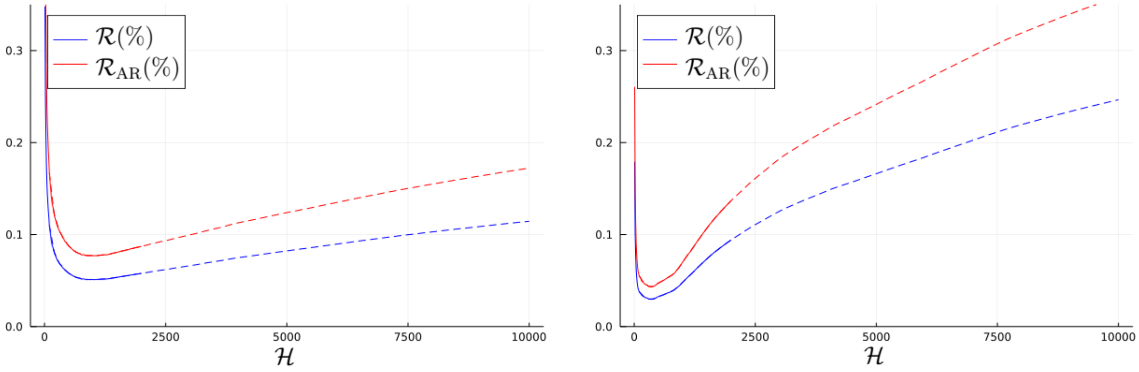


Figure 3.4.6: Relative regrets $\mathcal{R}^\pi$ as defined in (3.3.40), achieved by the policies discussed in Section 3.4.2 in the numerical experiments of DM2.

regrets than the myopic policy, regardless of the method used for inference. As expected, the best results were accomplished by the Bather Index, achieving the lowest regret consistently and independent of the number of arms of the model. The BIAP with SL is a close second. When the SL updates are used (as opposed to the variational Bayesian updates) the same policies produce lower regrets. Note that in the case of DM2 $\mathcal{A} = 2$ using BIAP with the SL approximation results in regrets that are very close to those observed with the Bather Index, but the regret of BIAP with the VB approximation increases quickly, realising a regret even higher than one of the myopic policies at $\mathcal{H} = 1000$.

Similarly to the findings in the independent case, the regrets observed were higher for DM1 than DM2, and the regrets observed for $\mathcal{A} = 5$ were higher than for $\mathcal{A} = 2$. We can also see that the separation between the myopic and the Bather type policies is more pronounced in the case of $\mathcal{A} = 5$ compared to $\mathcal{A} = 2$.

An unexpected observation is that while both types of regret decreased as the horizon increased for DM1, for DM2 the regret initially decreased only to increase again at large horizons. The effect increased horizons had on the observed regret was examined in more detail using the myopic policy. This policy was chosen as the vehicle of the investigation since it does not require tuning and it is the least computationally intensive policy developed both for DM1 and DM2. From $\mathcal{H} = 0$ to $\mathcal{H} = 2000$ the horizon was increased in increments of 10, then from $\mathcal{H} = 2000$ to $\mathcal{H} = 10000$ increased in increments of 100. The results can be seen in Figure 3.4.7.



(a) DM1

(b) DM2

Figure 3.4.7: The regret measure is demonstrated to increase after increasing horizons past a certain point. $\mathcal{R}_{\text{AR}}$ is a now defunct regret measure, which has been moved to the appendix.

These experiments focusing on the regrets replicate the results seen in the initial numerical experiments for $\mathcal{H} = 100, 200, 500$ and 1000. However, they also uncovered that past the original set of horizons the regrets of DM1 increase similarly to what was seen for DM2. This was not observed in the initial numerical experiments, since the increase is slow and starts around $\mathcal{H} = 1000$, the last horizon initially considered. In the case of DM2 the regrets increase more rapidly and from an earlier point onward which simply made this behaviour easier to uncover in the original experiments. This increase in regrets is contrary to our expectations, since a larger horizon implies more observations and therefore more accurate estimates of the reward rates. However, as described in Section 3.4.1, in the absence of conjugate models for $p(T)$ approximate methods were used to update the posterior distribution. This reduced the accuracy of the parameter estimates, possibly leading to misidentification of the best arm. We found that the estimates were more inaccurate for DM2 than for DM1, which could explain the more rapid increase in regrets observed with DM2. For more details, see Appendix A.4.

3.5 Concluding Remarks

3.5.1 Discussion of Model Assumptions

Throughout Chapter 3 some choices and modelling assumptions have been made on which we reflect here. The most important of those is the choice of metric made in Section 3.2. We opted to approximate the semi-Markov long-run average reward optimality criterion with the optimality criterion of a discrete-time multi-armed bandit problem, for which the expected reward rate of an arm $\mathbb{E}\left[P/T\right]$ was an appropriate metric, and therefore was used throughout Chapter 3. However, we had the option of choosing a different semi-Markov optimality criterion to use in approximation, which is based on the ratio average reward. This would have resulted in characterising the arms of the MAB by a different metric, $\mathbb{E}\left[P\right]/\mathbb{E}\left[T\right]$.

The most immediate advantage we see of using the metric $\mathbb{E}\left[P\right]/\mathbb{E}\left[T\right]$ over $\mathbb{E}\left[P/T\right]$ is the ease with which it can be calculated. While considering IM1, we had to use numerical integration as shown in (3.3.11) to evaluate $\mathbb{E}\left[1/T\right]$. Using the other metric would have required calculation of $\mathbb{E}\left[T\right]$ instead, which is straightforward and can be done analytically. Consequently, application of the knowledge gradient and CKGI policies would have required less numerical computation. In the dependent models $\mathbb{E}\left[T\right]$ would be obtained identically to the independent models, and only $\mathbb{E}\left[P\right]$ would be needed to be evaluated with respect to the full state. While this would not reduce the dimensions needed to numerically integrate over in either (3.4.5) or (3.4.6), it would lead to simpler integrands.

Given the above and that it is based on a semi-Markov optimality criterion, $\mathbb{E}\left[P\right]/\mathbb{E}\left[T\right]$ might have been a better choice of metric. However, without re-examining the intelligence problem, we cannot say how the performance of the solution methods would compare; if $\mathbb{E}[P/T]$ or $\mathbb{E}[P]/\mathbb{E}[T]$ would yield better results.

One of our observations is regarding the knowledge gradient and the CKGI. We must note that the intelligence problem is defined with an infinite horizon and that in the numerical experiments a horizon is only present due to necessity. However, as the knowledge gradient was adapted to our problem from a finite horizon formulation, it makes use of the residual horizon in directing which arm should be continued. The same applies to the CKGI as it was built on our adaptation of the knowledge gradient. Given that the myopic, EGGI policies remain ignorant of the artificially present horizon, the knowledge gradient and the CKGI may have an unfair advantage in knowing when the problem terminates, potentially inflating the rewards seen in Section 3.3.4 and Section 3.4.4. In terms of applying these policies, this poses some difficulties.

To mitigate the above, the policies can be adjusted by keeping the residual horizon $\mathcal{H}_r$ in their formulation a large constant value for every decision epoch. Ryzhov et al. (2012) adapted the knowledge gradient policy to an infinite horizon by considering a finite horizon problem then letting the horizon tend to infinity. One may wish to investigate the applicability of such an approach to our problem, though we recognize it may require significant effort.

3.5.2 Summary of Results

For the independent case of the intelligence problem, the myopic policy performed poorly. However, all other approaches devised to direct source selection for the independent case of the intelligence problem performed well, even if the specifics depended on the model and the number of sources present. While the Bather index is highly impractical due to the required tuning, the EGGI, the knowledge gradient and the CKGI can be applied directly. We must emphasise the achievements of the CKGI. Although it struggled in the presence of high observational uncertainties, extreme values of s^2 were required to produce the decline in performance. Otherwise, the CKGI consistently attained one of the lowest regrets, for some cases of IM2 coming within $\approx 1\%$ of the super-optimal policy. While the knowledge gradient was shown to be less resilient to increases in the observational uncertainties, the EGGI was the least affected. Consequently, in those cases the EGGI clearly outperformed both the knowledge gradient and the CKGI.

We found that the dependence structure examined between the relevance probabilities of the tips and their evaluation times severely limited what we can achieve due to computational difficulties. Even so, the BIAP, which selected the exploration parameters for the Bather index based on our prior beliefs performed similarly to the Bather index, for which these parameters were directly tuned. Both of these methods clearly outperformed the myopic policy. We found that for our purposes the simpler SL approximation outperformed the more complex variational inference method when updating the coefficients of the logistic link function. Ultimately, due to the approximate nature of these methods both are prone to incorrectly estimating the parameter values.

Chapter 4

The Intelligence Puzzle

4.1 Introducing the Intelligence Puzzle

4.1.1 Motivation

While the work in Chapter 3 considers intelligence collection and analysis on strategic time-scales, here in Chapter 4 we approach the problem on a tactical level. The collection and analysis process is placed in an investigative setting, where intelligence is viewed as a tool to solve a specific question, such as plans surrounding a terrorist attack. Be it a friendly game of Cluedo or one of Agatha Christie's detective novels, we all know that an investigation often requires multiple types of information to be uncovered, such as motives, opportunity and the weapon used. Similarly, an investigation that aims to prevent crime or other adverse events is in need of a variety of types of information.

The *intelligence puzzle* is a novel problem, where the intelligence team has to consider a diverse assortment of tips; while the usefulness of a tip is still binary, either relevant or nuisance, they all have an inherent type which corresponds to different topics they may provide information on, such as ”what”, ”where” or ”when”. To answer the intelligence question at hand, the intelligence team must solve an intelligence puzzle where each piece is a single relevant tip providing a certain type of information. Such a tip from now on will be referred to as a *magic bullet* as only one needs to be uncovered for each type to complete the puzzle. However, further complicating the issue there are multiple sources and all may provide any type of information. The intelligence question remaining unanswered carries a risk with it; the attack may be executed in the absence of all relevant information for interdiction, and therefore the team must complete the intelligence puzzle urgently. We set out to model such an intelligence puzzle as a semi-Markov decision process, and develop policies to guide source selection to minimise completion time of the puzzle.

The checklist problem studied in Atkinson and Kress (2018) can be interpreted as a tactical level intelligence collection with multiple types of intelligence if the mission represents a military operation, and the tasks that must be completed beforehand represent collection of intelligence. However, Atkinson and Kress (2018) allowed tasks to go uncompleted, in turn carrying the risk of mission failure.

Some stochastic assignment problems include features similar to those of the intelligence puzzle. The models most resembling ours are found in Ross and Wu (2013) and its follow up Ross et al. (2021). Ross and Wu (2013) considered a selection of boxes and a stream of sequentially arriving coupons, each characterised by an independent identically distributed binary vector determining which box the coupon can be assigned to. Their objective was to appoint a coupon to every box in manner that minimised the number of coupons to fill the boxes. In Ross et al. (2021) the problem was examined for unknown distributions of the binary vector. Formulations similar to Ross and Wu (2013) started to emerge for dynamic multi-item search (Dizon-Ross and Ross, 2020) and assortment planning problems (Ross and Seshadri, 2021). Although some of these papers also studied a problem where each of a collection of distinct types of items must be found, there are some key differences compared to the intelligence puzzle. While their items all originated from a single source and any single item may belong to more than one type, in the intelligence puzzle there are multiple sources present and an item must have one, but only one type. Furthermore, a key element of the intelligence puzzle is the evaluation of the tips which is not present in the papers mentioned.

In Section 4.1.2 the main elements of the intelligence puzzle are introduced. Section 4.2, Section 4.3 and Section 4.4 build on these to develop the intelligence puzzle for three different versions, as well as explore some policy options.

4.1.2 Elements of the Intelligence Puzzle

In this section we introduce the main elements of the intelligence puzzle which provide the structure common across all versions considered here. These are the physical state of the intelligence puzzle, and three vector quantities that characterise the quality of the sources.

We define the *physical state* of the intelligence puzzle as the collection of information types which we have and have not found a magic bullet for. For an intelligence puzzle consisting of $\mathcal{M}$ types of information its physical state $\boldsymbol{s} \in \{0,1\}^{\mathcal{M}}$ is given by the vector

$$\boldsymbol{s} = \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_{\mathcal{M}} \end{bmatrix}, \tag{4.1.1}$$

where the elements s_m represent the state of intelligence type m. A value of $s_m = 0$ means that a magic bullet of type m intelligence has been uncovered, while a value of $s_m = 1$ means that we are still looking for a magic bullet of type m intelligence. This state only depends on which types of information have or have not been uncovered, and therefore only changes upon discovery of a magic bullet. There are two states that deserve special mention; state $\mathbf{1}_{\mathcal{M}}$ is the starting state where no pieces of information have been discovered yet, and state $\mathbf{0}_{\mathcal{M}}$ which is the end state where all required pieces of information have been discovered.

The *stage* $g \in \{1, ..., \mathcal{M}\}$ of the intelligence puzzle is a sub-problem of the puzzle, characterised by the number of pieces of information discovered. Each physical state unambiguously belongs to a stage of the puzzle. Stage g refers to the phase between the $(g-1)^{\text{th}}$ and g^{th} discoveries. For example, stage 1 of the intelligence puzzle is when no magic bullets have been encountered yet, and stage $\mathcal{M}$ is when only one type remains to be discovered. Note that an intelligence puzzle with $\mathcal{M}$ types of information has $\mathcal{M}$ stages.

The novelty of the intelligence puzzle is in introducing types to the tips, leading to a more complex characterisation of sources, provided by three vector quantities each. Every source $a \in \{1, ..., \mathcal{A}\}$, where $\mathcal{A}$ is the number of sources available, produces a series of tips. The type of the tip is a random attribute M, which follows a Categorical distribution. The probability of encountering a tip of type $M = m \in \{1, ..., \mathcal{M}\}$ on source a is given by the *encounter probability* denoted as q_m^a . For a given source a the encounter probabilities can be grouped into a parameter vector;

$$\boldsymbol{q}^a = \begin{bmatrix} q_1^a \\ q_2^a \\ \cdots \\ q_{\mathcal{M}}^a \end{bmatrix}, \tag{4.1.2}$$

which is the parameter vector of the Categorical distribution of M. For all a and m $0 \leq q_m^a \leq 1$. Since each tip must belong to one of the $\mathcal{M}$ categories, the encounter probabilities must also satisfy the constraint

$$\sum_{m=1}^{\mathcal{M}} q_m^a = 1 \qquad \forall a \in \{1, ..., \mathcal{A}\}\,. \tag{4.1.3}$$

The probability that a tip of type m is a magic bullet is given by its *conditional success probability*, or *success probability* for short, and is denoted as p_m^a. It is conditional on the type of information $(M = m)$ the tip contains. When the success probabilities are grouped into parameter vectors they are denoted as

$$\boldsymbol{p}^a = \begin{bmatrix} p_1^a \\ p_2^a \\ \dots \\ p_{\mathcal{M}}^a \end{bmatrix}, \tag{4.1.4}$$

For all a and m $0 \leq p_m^a \leq 1$ and are assumed independent of each other. Note that there is no requirement on the p_m^a's to add up to 1.

It is important to reiterate that only one magic bullet per type is needed for the completion of the intelligence puzzle. If a second magic bullet was to be encountered it makes no difference as that aspect of the problem has already been solved. In practice this suggests that the success probabilities are state dependent and the success probability of an already discovered type is 0 on every source. The state dependent success probability vector is then denoted as $\boldsymbol{p}^a(\boldsymbol{s})$ and is calculated as

$$\boldsymbol{p}^a(\boldsymbol{s}) = \begin{bmatrix} p_1^a(\boldsymbol{s}) \\ p_2^a(\boldsymbol{s}) \\ \dots \\ p_{\mathcal{M}}^a(\boldsymbol{s}) \end{bmatrix} = \begin{bmatrix} p_1^a s_1 \\ p_2^a s_2 \\ \dots \\ p_{\mathcal{M}}^a s_{\mathcal{M}} \end{bmatrix}. \tag{4.1.5}$$

Since a tip is either a magic bullet (success) or a nuisance (failure), each of them is represented by a random variable X_m^a, which is a shorthand for $X_M^a \mid M = m$. It follows a Bernoulli distribution

$$X_m^a \sim \text{Bernoulli}(p_m^a). \tag{4.1.6}$$

The X_m^a of a given source and type are independent and identically distributed, while X_m^a of different sources and types are independent but may be distributed differently.

To determine the type and relevance of a tip, resources are expended which is captured in the *evaluation time* of the tips. The *base evaluation time* of a type m tip from source a is then given by t_m^a. A distinction must be made between the base evaluation time and the evaluation time as the latter is a function of the base evaluation time and may evolve throughout the problem as described shortly. When grouped by sources the base evaluation times are denoted as

$$\boldsymbol{t}^a = \begin{bmatrix} t_1^a \\ t_2^a \\ \dots \\ t_{\mathcal{M}}^a \end{bmatrix}. \tag{4.1.7}$$

Having discovered a given type m, no more of the same type will be deemed relevant. In such cases once the type has been established, the evaluation process can be halted and therefore will take a shorter amount of time than if we were still looking for a relevant tip. For this reason we assert that the evaluation time is a function of the physical state of the problem; denoted as $\boldsymbol{t}^a(\boldsymbol{s})$, and we reserve $\boldsymbol{t}^a = \boldsymbol{t}^a(\mathbf{1}_{\mathcal{M}})$ to denote the starting value of the evaluation-time vector; its value when no information has been discovered yet and $\boldsymbol{s} = \mathbf{1}_{\mathcal{M}}$.

The following rule for how the evaluation times change was chosen:

$$t_m^a(\boldsymbol{s}) = \begin{cases} t_m^a & \text{if } s_m = 1, \\ t_m^a/c & \text{if } s_m = 0, \end{cases} \tag{4.1.8}$$

where c is a known constant. Setting $c = 1$ means that the evaluation times do not depend on the types discovered, and setting $c = \infty$ means that no time is spent on evaluating tips of a type where a discovery has already been made.

Initially, the intelligence puzzle was envisioned with a very specific objective in mind, which is minimising its completion time. However, this can be generalised with the inclusion of the state dependent cost rate $w(\boldsymbol{s})$ to aiming to minimise the completion cost of the intelligence puzzle.

Note that while the evaluation time of any type m on any source a is known, since the types occur at random, the evaluation time of a tip in general is random. Consequently, the intelligence puzzle is best described as a semi-Markov decision problem, and is set out using such a framework for three variants over the next three sections.

4.2 The Basic Intelligence puzzle

In what we consider to be the most basic version of the intelligence puzzle all three parameter vectors introduced in Section 4.1.2, $\boldsymbol{p}^a(\boldsymbol{s}), \boldsymbol{q}^a$ and $\boldsymbol{t}^a(\boldsymbol{s})$ characterising the sources are known and their base equivalents remain constant throughout the puzzle. For the duration of any given stage all elements of the intelligence puzzle remain the same, and only change when the puzzle enters a new stage upon discovery of a magic bullet. These changes are a direct consequence of the change in the physical state of the system, and the rules governing them are described in (4.1.5) and (4.1.8).

As per the discussion in Section 4.1.2, Section 4.2.1 is devoted to describing this basic intelligence puzzle as a semi-Markov decision process, which is followed by a discussion on policy options in Section 4.2.2. These policies are then compared in a numerical study presented in Section 4.2.3.

4.2.1 The Intelligence Puzzle in the Framework of Semi-Markov Decision Processes

In the following, we formulate the intelligence puzzle described in Section 4.1.2 in the framework of semi-Markov Decision Processes. The framework requires to define the state space, decision epochs, action set, transition probabilities between the states, the expected cost of actions and the criterion of the decision process.

When all parameter vectors are known, the *state space* of the intelligence puzzle is given by its set of physical states as shown in Section 4.1.2; $\mathcal{S}_s = \{0,1\}^{\mathcal{M}}$, which is a finite set of size $2^{\mathcal{M}}$.

Decision epochs occur at random points in time; at the start of the process then strictly upon state transitions. This is quite intuitive: if the state does not change, neither does the prescribed action. In the context of the intelligence puzzle it corresponds to the beginning of the puzzle and subsequently the times of discoveries of the magic bullets. Since the number of information pieces to discover is finite, so is the number of decision epochs present in the puzzle, given by $\mathcal{M}$.

The distinction between the decision epoch and the stage of the puzzle needs to be made clear. When the state of the intelligence puzzle is simply given by the physical state of the puzzle, (such as when all parameters of the intelligence puzzle are known,) stage g follows on after the $(g-1)^{\text{th}}$ decision epoch and is terminated by the g^{th} decision epoch, and the durations of the stages correspond to the sojourn times. However, such correspondence can not be made in general as we will see in Section 4.3.2 and Section 4.4.2.

At every decision epoch e an *action* $a(e)$ is chosen by the decision maker from the action set $\mathcal{S}_a$, the set of sources which provide tips to evaluate. The action set is state independent and remains unchanged throughout the progression of the intelligence puzzle. It is given as $\mathcal{S}_a = \{1, ..., \mathcal{A}\}$. Note that according to the natural process of the intelligence puzzle an action may be applied multiple times until a relevant tip is found, but the action must remain the same until the next decision epoch. The action a and the state $\boldsymbol{s}$ in which that action was taken together determine both the probability distribution of the subsequent state and the distribution of time elapsed between the action and the resulting transition.

To compute the probability distribution of the subsequent state we have to consider the underlying natural process of the intelligence puzzle. First, let us denote by $\mathbb{P}(\boldsymbol{s}_{m^+} \mid \boldsymbol{s}, a)$ the probability that the process transitions to state $\boldsymbol{s_{m^+}}$ by discovering a magic bullet of type m, given that in state $\boldsymbol{s}$ action a was taken. Remember, the same source may be sampled repeatedly, since observing a failure leaves the state unchanged, giving no reason to deviate from the previous action. Consequently, let $\mathbb{P}(\boldsymbol{s}_{m^+}, k \mid \boldsymbol{s}, a)$ denote the probability that the magic bullet of type m is discovered in the k^{th} sample. Then

$$\mathbb{P}(\boldsymbol{s}_{m^+} \mid \boldsymbol{s}, a) = \mathbb{P}(\boldsymbol{s}_{m^+}, k = 1 \mid \boldsymbol{s}, a) + \mathbb{P}(\boldsymbol{s}_{m^+}, k > 1 \mid \boldsymbol{s}, a). \tag{4.2.1}$$

Any tip sampled from source a is type m with probability q^a_m, and given that it is type m, the probability of it being a magic bullet is $p^a_m(\boldsymbol{s})$. Therefore the probability that in any sequence of tips sampled from source a the first one leads to a magic bullet is independent of the length of the sequence and is given by $\mathbb{P}(\boldsymbol{s}_{m^+}, k = 1 \mid \boldsymbol{s}, a) = q^a_m p_m(\boldsymbol{s})$. For the type m magic bullet to be discovered later in the sequence ($k \geq 2$) two independent events must occur: no magic bullet to be discovered from the first tip, and that the process eventually transitions to state $\boldsymbol{s_{m^+}}$ by discovering a magic bullet of type m. The probability of the first event is $\sum_{m'} q_{m'}(1 - p^a_{m'}(\boldsymbol{s})) = \sum_{m'}(1 - q_{m'} p^a_{m'}(\boldsymbol{s}))$. Since no transition has occurred, both the state and the action in space has remained the same, and therefore the probability of the second event is $\mathbb{P}(\boldsymbol{s}_{m^+} \mid \boldsymbol{s}, a)$, the quantity we are looking for. Then

$$\mathbb{P}(\boldsymbol{s}_{m^+} \mid \boldsymbol{s}, a) = q^a_m p_m(\boldsymbol{s}) + \sum_{m'} (1 - q_{m'} p^a_{m'}(\boldsymbol{s})) \mathbb{P}(\boldsymbol{s}_{m^+} \mid \boldsymbol{s}, a),$$

which we can rearrange to find

$$\mathbb{P}(\boldsymbol{s}_{m^+} \mid \boldsymbol{s}, a) = \frac{q^a_m p_m(\boldsymbol{s})}{\sum_{m'} q_{m'} p^a_{m'}(\boldsymbol{s})}. \tag{4.2.2}$$

This quantity may also be referred to as the *transition probability* from state $\boldsymbol{s}$ to state $\boldsymbol{s}_{m^+}$ under action a. Observe that all non-zero probability transitions have a clear directionality, they bring the intelligence puzzle closer to its end state. Consequently, transition probabilities from state $\boldsymbol{s} = \boldsymbol{0}_{\mathcal{M}}$ are 0, and the puzzle never leaves the end state once it has been reached.

In the intelligence puzzle there is no lump-sum cost associated with the actions taken, but a continuous cost is incurred for the process staying in state $\boldsymbol{s}$ at rate $w(\boldsymbol{s})$. Other than the requirement that it stays constant while $\boldsymbol{s}$ is occupied, there are little restrictions on the cost rate. A notable exception is $w(\boldsymbol{0}_{\mathcal{M}})$, the reward rate of the end stage.

Once the end state is reached, the intelligence puzzle terminates, and no cost should be further accrued. For that reason we assert that $w(\mathbf{0}_{\mathcal{M}}) = 0$ for any version of the intelligence puzzle.

This cost rate is used alongside the expected sojourn times to compute the expected cost of choosing action a in state $\boldsymbol{s}$. Let us denote by $\mathcal{T}^a(\boldsymbol{s})$ the sojourn time that follows taking action a in state $\boldsymbol{s}$. Then the cost of taking action a is simply $w(\boldsymbol{s})\mathcal{T}^a(\boldsymbol{s})$, and the expected cost $w(\boldsymbol{s})\mathbb{E}\left[\mathcal{T}^a(\boldsymbol{s})\right]$. To find the expected sojourn time we again refer back to the natural process of the intelligence puzzle. To discover a magic bullet of any type and induce a transition, at least one tip must be evaluated. Conditional on this tip being type m the evaluation takes t_m^a, and therefore its expected evaluation time is $\sum_{m=1}^{\mathcal{M}} q_m^a t_m^a(\boldsymbol{s})$. The probability that the first tip did not provide a magic bullet is $1 - \sum_{m=1}^{\mathcal{M}} q_m^a p_m^a(\boldsymbol{s})$. If that occurs, further tips must be evaluated and time invested. Since the outcome of the tips is independent of how many have been evaluated previously, the expected time until a magic bullet is discovered after the first one failed to do so is still the expected sojourn time. Therefore we have

$$\mathbb{E}\left[\mathcal{T}^a(\boldsymbol{s})\right] = \sum_{m=1}^{\mathcal{M}} q_m^a t_m^a(\boldsymbol{s}) + \left(1 - \sum_{m=1}^{\mathcal{M}} q_m^a p_m^a(\boldsymbol{s})\right) \mathbb{E}\left[\mathcal{T}^a(\boldsymbol{s})\right],$$

which we rearrange to get

$$\mathbb{E}\left[\mathcal{T}^a(\boldsymbol{s})\right] = \frac{\sum_{m=1}^{\mathcal{M}} q_m^a t_m^a(\boldsymbol{s})}{\sum_{m=1}^{\mathcal{M}} q_m^a p_m^a(\boldsymbol{s})}. \tag{4.2.3}$$

Therefore the expected one-period cost of action a in state $\boldsymbol{s}$, which we denote as $r(\boldsymbol{s}, a)$ is

$$r(\boldsymbol{s}, a) = w(\boldsymbol{s})\mathbb{E}\left[\mathcal{T}^a(\boldsymbol{s})\right] = w(\boldsymbol{s})\frac{\sum_{m=1}^{\mathcal{M}} q_m^a t_m^a(\boldsymbol{s})}{\sum_{m=1}^{\mathcal{M}} q_m^a p_m^a(\boldsymbol{s})}. \tag{4.2.4}$$

This quantity may also be referred to as the *cost function* of action a in state $\boldsymbol{s}$.

Note that the expected time spent in state $\boldsymbol{s} = \mathbf{0}_{\mathcal{M}}$ is a special case. As it is the end state the process never moves on from $\mathbf{0}_{\mathcal{M}}$. However, since the cost rate for $\mathbf{0}_{\mathcal{M}}$ has been set to 0 a process staying in its end state will not accumulate costs either.

A *policy* π for a decision process is defined as any rule that selects the action to be applied as a function of the entire history of the decision process. For the intelligence puzzle we restrict our attention to deterministic stationary Markov policies, which contain no randomisation, explicit time dependence, and is only a function of the state. Then a policy directly prescribes the action based on the state of the decision process; $\pi(\boldsymbol{s}) = a$.

Next we define the objective function of the decision process. The original aim of the intelligence puzzle to minimise the time taken to discover a magic bullet for all required types of information corresponds to a cost rate of $w(\boldsymbol{s}) = 1$ for all states $\boldsymbol{s} \neq \mathbf{0}_{\mathcal{M}}$. However, to keep the formulation more general we define the objective function to minimise the undiscounted cumulative cost

$$\text{Obj: } \min_{\pi} \sum_{e=0}^{\mathcal{M}-1} \mathbb{E}\left[r\left(\boldsymbol{s}_e, \pi(\boldsymbol{s}_e)\right)\right], \tag{4.2.5}$$

where $\boldsymbol{s}_e$ is the state at decision epoch e and $\pi(\boldsymbol{s}_e)$ the action prescribed by policy π.

4.2.2 Policies

In this section we discuss the policies the decision maker may use to determine which action to take, i.e., which source to sample tips from. We consider the optimal policy which is obtained via dynamic programming, and a myopic policy. The circumstances required for optimality of the myopic policy are considered, and some sufficient conditions given. Other policy options such as look-ahead policies, policies based on aggregating the states of the intelligence puzzle and considering the sources separately are also briefly examined. All policies considered in detail are to be applied as described in the flowchart in Figure 4.2.1.

Optimal Policy via Dynamic Programming

Dynamic programming can be used to find the policy that optimally solves the decision process. While the intelligence puzzle is an infinite horizon problem, a clear end state exits and the number of decision epochs is finite and known. This structure lends itself

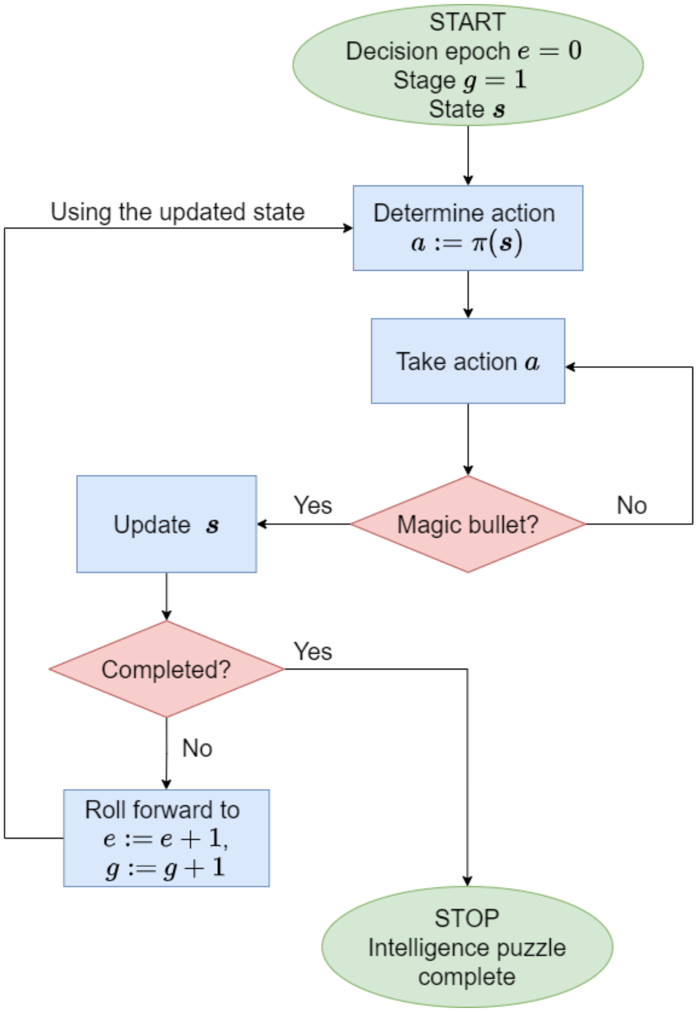


Figure 4.2.1: The process of applying policy $\pi(\boldsymbol{s})$ to the basic intelligence puzzle. The updating process of physical state $\boldsymbol{s}$ has been described in detail in Section 4.1.2, and $\pi(\boldsymbol{s})$ may be any policy from Section 4.2.2.

to a solution approach based on backward recursion.

To construct the Bellman equation for backward recursion, for every available action in every state we need the one-period transition probabilities between the states and the expected cost of the action. As part of defining the semi-Markov decision process in Section 4.2.1, we have derived the transition probability $\mathbb{P}(\boldsymbol{s}_{m^+} \mid \boldsymbol{s}, a)$ and is shown in (4.2.2). In that same section the expected cost $r(\boldsymbol{s}, a)$ of taking action a in state $\boldsymbol{s}$ has been established and is given by (4.2.4).

The last component of the backward recursion is the value function $\mathbb{V}^a(\boldsymbol{s})$ which provides the total expected cost of taking action a in state $\boldsymbol{s}$, given that after the resulting transition the optimal policy is followed. The value of a state $\mathbb{V}(\boldsymbol{s})$ is the lowest value of $\mathbb{V}^a(\boldsymbol{s})$, formally defined as

$$\mathbb{V}(\boldsymbol{s}) = \min_{a \in \mathcal{A}} \left\{ \mathbb{V}^a(\boldsymbol{s}) \right\},$$

and the minimising action the optimal action to take in that state. By calculating the value $\mathbb{V}(\boldsymbol{s})$ of all states starting from the end of the problem where $\boldsymbol{s} = \mathbf{0}_{\mathcal{M}}$ and working sequentially backward to the first stage where $\boldsymbol{s} = \mathbf{1}_{\mathcal{M}}$ we identify the optimal reward $\mathbb{V}(\mathbf{1})$ for the initial state, which is the minimised expected cumulative cost from (4.2.5). The recursion sequence of choices that lead to it is the optimal policy.

Once the end state is reached, the cost rate drops to $w(\mathbf{0}_{\mathcal{M}}) = 0$, and therefore the expected cost of any action taken in $\mathbf{0}_{\mathcal{M}}$ is 0. As stated before, the process cannot transition out of the end state, and therefore its value function is given as

$$\mathbb{V}(\mathbf{0}_{\mathcal{M}}) = 0 \tag{4.2.6}$$

In any other state $\boldsymbol{s}$ the value of action a is obtained as the sum of the expected immediate cost of the action, $r(\boldsymbol{s}, a)$ and the expected future cost due to the resulting transition, $\sum_{m=1}^{\mathcal{M}} \mathbb{P}\left(\boldsymbol{s}_{m^+} \mid \boldsymbol{s}, a\right) \mathbb{V}\left(\boldsymbol{s}_{m^+}\right)$. Therefore the value of every state can be obtained via the following recursion

$$\begin{aligned}
\mathbb{V}(\boldsymbol{s}) &= \min_{a \in \mathcal{A}} \left\{ r(\boldsymbol{s}, a) + \sum_{m=1}^{\mathcal{M}} \mathbb{P}\left(\boldsymbol{s}_{m^+} \mid \boldsymbol{s}, a\right) \mathbb{V}\left(\boldsymbol{s}_{m^+}\right) \right\} \\
&= \min_{a \in \mathcal{A}} \left\{ w\left(\boldsymbol{s}\right) \frac{\sum_{m=1}^{\mathcal{M}} q_m^a t_m^a(\boldsymbol{s})}{\sum_{m=1}^{\mathcal{M}} q_m^a p_m^a(\boldsymbol{s})} + \frac{\sum_{m=1}^{\mathcal{M}} q_m^a p_m^a(\boldsymbol{s}) \mathbb{V}\left(\boldsymbol{s}_{m^+}\right)}{\sum_{m=1}^{\mathcal{M}} q_m^a p_m^a(\boldsymbol{s})} \right\}.
\end{aligned} \tag{4.2.7}$$

The last line of (4.2.7) was obtained by substituting (4.2.2) and (4.2.4). Then the optimal policy in every state $\boldsymbol{s}$ is to take action a^* so that

$$a^* = \pi^{\mathrm{OPT}}(\boldsymbol{s}) := \arg\min_{a} \left\{ \mathbb{V}^a(\boldsymbol{s}) \right\}. \tag{4.2.8}$$

While the optimal policy of the basic intelligence puzzle has been found, it can be impractical to use as it requires computing the value function for $\mathcal{A}2^{\mathcal{M}}$ state and action combinations. Since its complexity grows exponentially with $\mathcal{M}$, obtaining the optimal policy for large puzzles is expected to be computationally challenging.

Myopic Policy

The myopic policy chooses the action with the smallest expected one-period cost $r(\boldsymbol{s}, a)$, given by (4.2.4). It chooses the source that is expected to obtain the next magic bullet the cheapest, with no regards to the states the puzzle may transition to afterwards. Note that the cost rate $w(\boldsymbol{s})$ is independent of the source chosen, and therefore the cheapest option is equivalent to the quickest one and so the value of the cost rate does not affect the decision making.

$$\arg\min_{a\in\mathcal{A}} \{r(\boldsymbol{s}, a)\} = \arg\min_{a\in\mathcal{A}} \left\{ w(\boldsymbol{s}) \frac{\sum_{m=1}^{\mathcal{M}} q_m^a t_m^a(\boldsymbol{s})}{\sum_{m=1}^{\mathcal{M}} q_m^a p_m^a(\boldsymbol{s})} \right\} = \arg\min_{a\in\mathcal{A}} \left\{ \frac{\sum_{m=1}^{\mathcal{M}} q_m^a t_m^a(\boldsymbol{s})}{\sum_{m=1}^{\mathcal{M}} q_m^a p_m^a(\boldsymbol{s})} \right\}$$

Consequently, the myopic policy not only aims to minimise the cost incurred in the current stage, but also the time spent in it. Then the myopic policy in every state is to take action a^* so that

$$a^* = \pi^{\text{MYO}}(\boldsymbol{s}) := \arg\min_{a\in\mathcal{A}} \left\{ \frac{\sum_{m=1}^{\mathcal{M}} q_m^a t_m^a(\boldsymbol{s})}{\sum_{m=1}^{\mathcal{M}} q_m^a p_m^a(\boldsymbol{s})} \right\}. \tag{4.2.9}$$

It is expected to be considerably faster than the optimal policy as it can be implemented in an online fashion so that actions need to be only evaluated for the $\mathcal{M}$ states that are visited throughout the puzzle.

Cases in which the myopic policy is optimal

In general, the myopic policy is not optimal as it ignores the path taken through the state space to complete the intelligence puzzle. However, there are some special cases where it is on par with the optimal policy.

By rearranging (4.2.7),

$$\mathbb{V}^a(\boldsymbol{s}) = w(\boldsymbol{s})\frac{\sum_{m=1}^{\mathcal{M}} q_m^a t_m^a(\boldsymbol{s})}{\sum_{m=1}^{\mathcal{M}} q_m^a p_m^a(\boldsymbol{s})} + \sum_{m=1}^{\mathcal{M}} \frac{q_m^a p_m^a(\boldsymbol{s})}{\sum_{m'=1}^{\mathcal{M}} q_{m'}^a p_{m'}^a(\boldsymbol{s})}\mathbb{V}(\boldsymbol{s}_{m^+}) \tag{4.2.10}$$

$$= \frac{w(\boldsymbol{s})\sum_{m=1}^{\mathcal{M}} q_m^a t_m^a(\boldsymbol{s}) + \sum_{m=1}^{\mathcal{M}} q_m^a p_m^a(\boldsymbol{s})\mathbb{V}(\boldsymbol{s}_{m^+})}{\sum_{m=1}^{\mathcal{M}} q_m^a p_m^a(\boldsymbol{s})} \tag{4.2.11}$$

$$= \frac{w(\boldsymbol{s})\sum_{m=1}^{\mathcal{M}} q_m^a t_m^a(\boldsymbol{s})}{\sum_{m=1}^{\mathcal{M}} q_m^a p_m^a(\boldsymbol{s})}\left(1 + \frac{\sum_{m=1}^{\mathcal{M}} q_m^a p_m^a(\boldsymbol{s})\mathbb{V}(\boldsymbol{s}_{m^+})}{w(\boldsymbol{s})\sum_{m=1}^{\mathcal{M}} q_m^a t_m^a(\boldsymbol{s})}\right) \tag{4.2.12}$$

$$= r(\boldsymbol{s}, a)\left(1 + \frac{\sum_{m=1}^{\mathcal{M}} q_m^a p_m^a(\boldsymbol{s})\mathbb{V}(\boldsymbol{s}_{m^+})}{w(\boldsymbol{s})\sum_{m=1}^{\mathcal{M}} q_m^a t_m^a(\boldsymbol{s})}\right), \tag{4.2.13}$$

the immediate cost function, which the myopic policy is based on can be factored out of $\mathbb{V}^a(\boldsymbol{s})$. If the multiplier of $r(\boldsymbol{s}, a)$ equals 1, $r(\boldsymbol{s}, a) = \mathbb{V}^a(\boldsymbol{s})$ therefore the myopic and the optimal policy are identical. This only occurs in the last stage of the intelligence puzzle where $\mathbb{V}(\boldsymbol{s}_{m^+}) = 0$.

However, $r(\boldsymbol{s}, a)$ and $\mathbb{V}^a(\boldsymbol{s})$ need not equal for the two policies to be the same, only that

$$\arg\min_{a\in\mathcal{A}} \{r(\boldsymbol{s}, a)\} = \arg\min_{a\in\mathcal{A}} \{\mathbb{V}^a(\boldsymbol{s})\}, \tag{4.2.14}$$

$$\arg\min_{a\in\mathcal{A}} \left\{\frac{\sum_{m=1}^{\mathcal{M}} q_m^a t_m^a(\boldsymbol{s})}{\sum_{m=1}^{\mathcal{M}} q_m^a p_m^a(\boldsymbol{s})}\right\} = \arg\min_{a\in\mathcal{A}} \left\{r(\boldsymbol{s}, a)\left(1 + \frac{\sum_{m=1}^{\mathcal{M}} q_m^a p_m^a(\boldsymbol{s})\mathbb{V}(\boldsymbol{s}_{m^+})}{w(\boldsymbol{s})\sum_{m=1}^{\mathcal{M}} q_m^a t_m^a(\boldsymbol{s})}\right)\right\}. \tag{4.2.15}$$

Note that (4.2.15) is a very strong condition, which in general would not be satisfied. However, it is possible to identify some cases in which the condition holds. Since (4.2.15) is a complicated condition, we restrict our attention to the smallest instance of the intelligence puzzle, where $\mathcal{M} = 2$ and $\mathcal{A} = 2$; there are two types of information and two sources that can supply them. To further simplify the problem we set the reward rate for all states $w(\boldsymbol{s}) = 1$.

This is a 2 stage problem with 4 available states,

$$\boldsymbol{s}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \qquad \boldsymbol{s}_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \qquad \boldsymbol{s}_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad \boldsymbol{s}_3 = \begin{bmatrix} 1 \\ 1 \end{bmatrix},$$

and the sources are characterised by

$$\boldsymbol{q}^a = \begin{bmatrix} q_1^a \\ q_2^a \end{bmatrix} = \begin{bmatrix} q_1^a \\ 1 - q_1^a \end{bmatrix} = \begin{bmatrix} q^a \\ 1 - q^a \end{bmatrix}, \qquad \boldsymbol{p}^a = \begin{bmatrix} p_1^a \\ p_2^a \end{bmatrix}, \qquad \boldsymbol{t}^a = \begin{bmatrix} t_1^a \\ t_2^a \end{bmatrix}.$$

The state $\boldsymbol{s}_0$ is the end state of the puzzle and is not otherwise encountered. Stage 2 is the last stage of the problem and the puzzle is either in state $\boldsymbol{s}_1$ or $\boldsymbol{s}_2$. From either of those states the only valid transition is to $\boldsymbol{s}_0$. For compactness sake we discuss the conditions for optimality in terms of $\mathbb{P}\left(\boldsymbol{s}_{m^+} \mid \boldsymbol{s}, a\right)$, $\mathbb{E}\left[\mathcal{T}^a\left(\boldsymbol{s}\right)\right]$ and $\mathbb{V}(\boldsymbol{s})$ as given in (4.2.2), (4.2.3) and (4.2.7) respectively. Then the value functions of states $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ are

$$\begin{aligned} \mathbb{V}^a(\boldsymbol{s}_1) = &\ \mathbb{E}\left[\mathcal{T}^a\left(\boldsymbol{s}_1\right)\right] + \sum_m^{\mathcal{M}} \mathbb{P}\left(\boldsymbol{s}_{m^+} \mid \boldsymbol{s}_1, a\right) \mathbb{V}\left(\boldsymbol{s}_{m^+}\right) \\ = &\ \mathbb{E}\left[\mathcal{T}^a\left(\boldsymbol{s}_1\right)\right] + \mathbb{P}\left(\boldsymbol{s}_0 \mid \boldsymbol{s}_1, a\right) \mathbb{V}\left(\boldsymbol{s}_0\right) \\ = &\ \mathbb{E}\left[T_a\left(\boldsymbol{s}_1\right)\right], \end{aligned} \tag{4.2.16}$$

$$\begin{aligned} \mathbb{V}^a(\boldsymbol{s}_2) = &\ \mathbb{E}\left[\mathcal{T}^a\left(\boldsymbol{s}_2\right)\right] + \sum_m^{\mathcal{M}} \mathbb{P}\left(\boldsymbol{s}_{m^+} \mid \boldsymbol{s}_2, a\right) \mathbb{V}\left(\boldsymbol{s}_{m^+}\right) \\ = &\ \mathbb{E}\left[\mathcal{T}^a\left(\boldsymbol{s}_2\right)\right] + \mathbb{P}\left(\boldsymbol{s}_0 \mid \boldsymbol{s}_2, a\right) \mathbb{V}\left(\boldsymbol{s}_0\right) \\ = &\ \mathbb{E}\left[\mathcal{T}^a\left(\boldsymbol{s}_2\right)\right]. \end{aligned} \tag{4.2.17}$$

Consequently, for the states available in stage 2 we have

$$\arg\min_{a \in \mathcal{S}_{\mathcal{A}}} \mathbb{V}^a(\boldsymbol{s}_1) = \arg\min_{a \in \mathcal{S}_{\mathcal{A}}} \mathbb{E}\left[\mathcal{T}^a(\boldsymbol{s}_1)\right], \tag{4.2.18}$$

$$\arg\min_{a \in \mathcal{S}_{\mathcal{A}}} \mathbb{V}^a(\boldsymbol{s}_2) = \arg\min_{a \in \mathcal{S}_{\mathcal{A}}} \mathbb{E}\left[\mathcal{T}^a(\boldsymbol{s}_2)\right] \tag{4.2.19}$$

meaning that the myopic policy is optimal in states $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ regardless of the values of $\boldsymbol{q}^a, \boldsymbol{p}^a$, and $\boldsymbol{t}^a$, as predicted based on (4.2.13).

In stage 1 no magic bullets have been discovered yet, and the puzzle is guaranteed to be is state $\boldsymbol{s}_3$. From there it may transition either to $\boldsymbol{s}_1$ or $\boldsymbol{s}_2$. The value function of state $\boldsymbol{s}_3$ is then

$$\begin{aligned}\mathbb{V}^a(\boldsymbol{s}_3) =& \ \mathbb{E}\left[\mathcal{T}^a\left(\boldsymbol{s}_3\right)\right] + \sum_m^{\mathcal{M}} \mathbb{P}\left(\boldsymbol{s}_{m^+} \mid \boldsymbol{s}_3, a\right) \mathbb{V}\left(\boldsymbol{s}_{m^+}\right) \\ =& \ \mathbb{E}\left[\mathcal{T}^a\left(\boldsymbol{s}_3\right)\right] + \mathbb{P}\left(\boldsymbol{s}_1 \mid \boldsymbol{s}_3, a\right) \mathbb{V}\left(\boldsymbol{s}_1\right) + \mathbb{P}\left(\boldsymbol{s}_2 \mid \boldsymbol{s}_3, a\right) \mathbb{V}\left(\boldsymbol{s}_2\right) \qquad (4.2.20) \\ =& \ \mathbb{E}\left[\mathcal{T}^a\left(\boldsymbol{s}_3\right)\right] + \mathbb{P}\left(\boldsymbol{s}_1 \mid \boldsymbol{s}_3, a\right) \mathbb{V}\left(\boldsymbol{s}_1\right) + \left(1 - \mathbb{P}\left(\boldsymbol{s}_1 \mid \boldsymbol{s}_3, a\right)\right) \mathbb{V}\left(\boldsymbol{s}_2\right). \qquad (4.2.21)\end{aligned}$$

For the myopic policy to pick the same source as the optimal policy in state $\boldsymbol{s}_3$

$$\arg\min_a \mathbb{V}^a(\boldsymbol{s}_3) = \arg\min_a \mathbb{E}\left[\mathcal{T}^a(\boldsymbol{s}_3)\right] \qquad (4.2.22)$$

needs to be satisfied, which means either (4.2.23) and (4.2.24)

$$\mathbb{V}^1(\boldsymbol{s}_3) \leq \mathbb{V}^2(\boldsymbol{s}_3) \qquad (4.2.23)$$

$$\mathbb{E}\left[\mathcal{T}^1\left(\boldsymbol{s}_3\right)\right] < \mathbb{E}\left[\mathcal{T}^2\left(\boldsymbol{s}_3\right)\right] \qquad (4.2.24)$$

or (4.2.25) and (4.2.26)

$$\mathbb{V}^1(\boldsymbol{s}_3) \geq \mathbb{V}^2(\boldsymbol{s}_3) \qquad (4.2.25)$$

$$\mathbb{E}\left[\mathcal{T}^1\left(\boldsymbol{s}_3\right)\right] > \mathbb{E}\left[\mathcal{T}^2\left(\boldsymbol{s}_3\right)\right] \qquad (4.2.26)$$

need to be simultaneously true. From that we can derive the necessary condition for the optimality of the myopic policy to be

$$1 \geq \ \Delta\mathbb{P}\left(\boldsymbol{s}_1 \mid \boldsymbol{s}_3\right) \ \frac{\min\limits_a \mathbb{E}\left[\mathcal{T}^a(\boldsymbol{s}_1)\right] - \min\limits_a \mathbb{E}\left[\mathcal{T}^a(\boldsymbol{s}_2)\right]}{\mathbb{E}\left[\mathcal{T}^2\left(\boldsymbol{s}_3\right)\right] - \mathbb{E}\left[\mathcal{T}^1\left(\boldsymbol{s}_3\right)\right]} \qquad (4.2.27)$$

where

$$\Delta\mathbb{P}\left(\boldsymbol{s}_1 \mid \boldsymbol{s}_3\right) = \mathbb{P}\left(\boldsymbol{s}_1 \mid \boldsymbol{s}_3, a=1\right) - \mathbb{P}\left(\boldsymbol{s}_1 \mid \boldsymbol{s}_3, a=2\right). \qquad (4.2.28)$$

The case $\mathbb{E}\left[\mathcal{T}^1\left(\boldsymbol{s}_3\right)\right] = \mathbb{E}\left[\mathcal{T}^2\left(\boldsymbol{s}_3\right)\right]$ is considered separately along with the derivation of the above condition in Appendix B.1.1. It is clear from (4.2.27) that strict conditions are required even in the $\mathcal{M} = 2, \mathcal{A} = 2$ case for the myopic policy to be identical to the optimal policy. By restricting the values of the relevant parameters, two cases have been identified where these conditions are met. The common restrictions are as follows.

$t_1^1 = t_2^1 = t_1^2 = t_2^2 = 1$ identical unit evaluation times,

$\boldsymbol{t}^a(\boldsymbol{s}) = \boldsymbol{t}^a$ state independent evaluation times.

In the first restricted case the success probabilities only depend on the type of information, not on the source

$$p_1^1 = p_1^2 = p_1, \qquad\qquad p_2^1 = p_2^2 = p_2.$$

From this, two sub-cases emerge for which the myopic policy is optimal

$$\frac{(1-q^1)}{q^1}\frac{(1-q^2)}{q^2} < \frac{p_1}{p_2} \qquad \text{where } \frac{p_1}{p_2} < 1, \tag{4.2.29}$$

$$\frac{(1-q^1)}{q^1}\frac{(1-q^2)}{q^2} > \frac{p_1}{p_2} \qquad \text{where } \frac{p_1}{p_2} > 1. \tag{4.2.30}$$

In the second restricted case the success probabilities only depend on the source and not on the type of information so that

$$p_1^1 = p_2^1 = p^1, \qquad\qquad p_1^2 = p_2^2 = p^2.$$

Then 2 sub-cases can be identified based on the relationship of p^1 and p^2. The condition for $p^1 > p^2$ is

$$\frac{1}{2}\left(1 - \sqrt{1 - 4\frac{p^1}{p^2}q^1\left(1-q^1\right)}\right) \le q^2 \le \frac{1}{2}\left(1 + \sqrt{1 - 4\frac{p^1}{p^2}q^1\left(1-q^1\right)}\right), \tag{4.2.31}$$

and for $p^1 < p^2$ the condition is

$$q^2 \le \frac{1}{2}\left(1 - \sqrt{1 - 4\frac{p^1}{p^2}q^1\left(1-q^1\right)}\right) \quad \text{or} \quad \frac{1}{2}\left(1 + \sqrt{1 - 4\frac{p^1}{p^2}q^1\left(1-q^1\right)}\right) \le q^2. \tag{4.2.32}$$

The derivation of these conditions can be found in Appendices B.1.2 and B.1.3.

Look-ahead Policies

A compromise between the reduced computational complexity of the myopic policy and the performance of the optimal policy can be made by considering a k-step look-ahead policy. Instead of only looking at the costs acquired by selecting source a in the current stage of the problem, a k-step look-ahead policy considers the cost of selecting source a in the current stage and the costs that are acquired over the next k stages of the problem. Most commonly costs acquired after the kth step are ignored, which makes the myopic policy equivalent to a 0-step look-ahead policy and the optimal policy to an $(\mathcal{M}-1)$-step look-ahead policy, and for $0 < k < \mathcal{M}-1$ the k-step look-ahead policy a compromise between the two.

While one may define a look-ahead policy in which the costs after the kth step are considered in an approximate manner, here we focus on the 1-step look-ahead policy as defined above. At every stage of the puzzle source a^* is selected so that

$$a^* = \pi^{\mathrm{LA}}(\boldsymbol{s}) =: \arg\min_{a} \left\{ w(\boldsymbol{s})\mathbb{E}\left[\mathcal{T}^{a}\left(\boldsymbol{s}\right)\right] + \sum_{m=1}^{\mathcal{M}} \mathbb{P}\left(\boldsymbol{s}_{m^+} \mid \boldsymbol{s}, a\right) w(\boldsymbol{s}_{m^+}) \min_{a'}\mathbb{E}\left[\mathcal{T}^{a'}\mathbb{E}\left(\boldsymbol{s}_{m^+}\right)\right] \right\}. \tag{4.2.33}$$

Note that $\mathbb{E}\left[\mathcal{T}^{a}\left(\boldsymbol{s}_0\right)\right] = 0$. While it is not expected to be optimal apart from the $\mathcal{M} = 2$ case, it requires significantly fewer calculations than finding the optimal policy via a dynamic program as it only needs to consider the states that can be reached from the current state.

Ordered Discoveries

Let us consider a variant of the intelligence puzzle, where the order in which the magic bullets must be discovered is given. In such a case even if a magic bullet is encountered, if it is not the specific type of magic bullet that we are looking for it will not contribute to solving the intelligence puzzle and is discarded. Let us refer to such a magic bullet as *out of order*. A model like this might be appropriate when one magic bullet provides context for another one, without which the information cannot be interpreted and therefore the magic bullet could not be discovered.

Let us denote the stages of the puzzle by $g = \{1, ..., \mathcal{M}\}$, and the required order of discovery as $\mathcal{O}$. Then the type that needs to be discovered during stage g, denoted as m^* is found as

$$m^* = \mathcal{O}(g). \tag{4.2.34}$$

Observation 4.2.1. *With the order of discoveries fixed, the physical state space of the intelligence puzzle is only a subset of $\mathcal{S}_{\boldsymbol{s}}$, uniquely determined by $\mathcal{O}$ and denoted* *as $\mathcal{S}_{\mathcal{O}}$. This $\mathcal{S}_{\mathcal{O}}$ contains only one state per stage which we denote as $\boldsymbol{s}(\mathcal{O}, g)$ and $\boldsymbol{s}(\mathcal{O}, g+1) \equiv \boldsymbol{s}_{m^*+}$.*

Since in stage g only the discovery of a type m^* magic bullet results in state transition, the expected sojourn time is dependent on m^* and simply equals the expected time till discovery of m^*, modified from (4.2.3) as

$$\mathbb{E}\left[\mathcal{T}^a(\boldsymbol{s}, m^*)\right] = \frac{\sum_{m=1}^{\mathcal{M}} q_m^a t_m^a(\boldsymbol{s})}{q_{m^*}^a p_{m^*}^a}, \tag{4.2.35}$$

since the expected evaluation time of any given tip includes time spent on evaluating out of order ($m \neq m^*$) tips but only a tip of type m^* may provide a magic bullet and trigger a transition.

Proposition 4.2.2. *When the magic bullets need to be discovered according to a fixed order, a policy that at each stage selects the source a^* given by*

$$a^* := \arg\min_a \mathbb{E}\left[\mathcal{T}^a(\boldsymbol{s}, m^*)\right] \tag{4.2.36}$$

is optimal.

Proof. The optimal policy is found via dynamic programming. Since the order of discovery is fixed and therefore Observation 4.2.1 holds, the state transitions are independent of the source a chosen and deterministic with a transition probability of $\mathrm{P}(\boldsymbol{s}_{m^*+} \mid \boldsymbol{s}, a) = \mathrm{P}(\boldsymbol{s}(\mathcal{O}, g+1) \mid \boldsymbol{s}(\mathcal{O}, g), a) = 1$. Then the Bellman equation for backwards recursion takes the form

$$\begin{aligned}
\mathbb{V}_{\mathcal{O}}(\mathbf{0}_{\mathcal{M}}) &= 0, \\
\mathbb{V}_{\mathcal{O}}(\boldsymbol{s}) &= \min_a \left\{ r\left(\boldsymbol{s}, a\right) + \mathrm{P}(\boldsymbol{s}_{m*+} \mid \boldsymbol{s}, a)\mathbb{V}_{\mathcal{O}}(\boldsymbol{s}_{m^*+}) \right\} \\
&= \min_a \left\{ \boldsymbol{w}(\boldsymbol{s})\mathbb{E}\left[\mathcal{T}^a(\boldsymbol{s}, m^*)\right] + \mathbb{V}_{\mathcal{O}}(\boldsymbol{s}_{m^*+}) \right\} \\
&= \min_a \left\{ \boldsymbol{w}(\boldsymbol{s})\mathbb{E}\left[\mathcal{T}^a(\boldsymbol{s}, m^*)\right] \right\} + \mathbb{V}_{\mathcal{O}}(\boldsymbol{s}_{m^*+}).
\end{aligned}$$

The optimal policy is to take the minimising action a^*,

$$a^* := \arg\min_a \left\{ \boldsymbol{w}(\boldsymbol{s})\mathbb{E}\left[\mathcal{T}^a(\boldsymbol{s}, m^*)\right] \right\} = \arg\min_a \left\{ \mathbb{E}\left[\mathcal{T}^a(\boldsymbol{s}, m^*)\right] \right\},$$

since neither $\boldsymbol{w}(\boldsymbol{s})$ or $\mathbb{V}_{\mathcal{O}}(\boldsymbol{s}_{m^*+})$ depends on a. □

Next suppose that while there needs to be a predetermined order of discovery, the decision maker is allowed to set that order at the start of the puzzle. Since the objective of the intelligence puzzle is to minimise the cost of completion, the best order of discovery is the one that achieves the lowest expected cost. The best order can be found via backwards recursion, but we have to re-define the available actions and the associated costs.

At every decision epoch instead of choosing a source the decision maker must choose which type of information $m^* \in \mathcal{S}_m(\boldsymbol{s})$ is best to discover next, where $\mathcal{S}_m(\boldsymbol{s})$ is the state dependent action space which is the the set of types not yet discovered. The cost of each action m is given by

$$r\left(\boldsymbol{s}, m\right) = \min_{a} \mathbb{E}\left[\mathcal{T}^a(\boldsymbol{s}, m)\right], \tag{4.2.37}$$

since Proposition 4.2.2 states that if only type m may produce a magic bullet it is optimal to choose the source a that minimises $\mathbb{E}\left[\mathcal{T}^a(\boldsymbol{s}, m)\right]$. Note that action m guarantees transition from state $\boldsymbol{s}$ to $\boldsymbol{s}_{m^+}$ and so we only need to consider the value of state $\boldsymbol{s}_{m^+}$, $\mathbb{V}(\boldsymbol{s}_{m^+})$.

Then the Bellman equation is given as

$$\begin{aligned}
\mathbb{V}\left(\mathbf{0}_{\mathcal{M}}\right) &= 0 && (4.2.38)\\
\mathbb{V}\left(\boldsymbol{s}\right) &= \min_{m}\left\{r\left(\boldsymbol{s}, m\right) + \mathbb{V}\left(\boldsymbol{s}_{m^+}\right)\right\} && (4.2.39)\\
&= \min_{m}\left\{\min_{a}\, w(\boldsymbol{s})\mathbb{E}\left[\mathcal{T}^a(\boldsymbol{s}, m)\right] + \mathbb{V}\left(\boldsymbol{s}_{m^+}\right)\right\} && (4.2.40)\\
&= \min_{m}\left\{\min_{a}\, w(\boldsymbol{s})\frac{\sum_{m'=1}^{\mathcal{M}} q^a_{m'} t^a_{m'}(\boldsymbol{s})}{q^a_m p^a_m} + \mathbb{V}\left(\boldsymbol{s}_{m^+}\right)\right\}, && (4.2.41)
\end{aligned}$$

and the sequence of minimising m^*'s that achieves $\mathbb{V}(\mathbf{1}_{\mathcal{M}})$, the value of the starting state, is the optimal order $\mathcal{O}^*$ to discover the magic bullets in.

In Section 4.1.2 the dependence of the evaluation times on the state of the puzzle is described; the evaluation time of tips belonging to already discovered types may be reduced. Along similar lines we can motivate a model where the evaluation times depend on the type of information that needs to be obtained to complete the current stage. The reduced evaluation times correspond to only needing to evaluate which m does a tip belong to, and once they are evaluated to be not the correct type there is no need for further evaluation and no more resources are spent on them. Then the evaluation times are no longer dependent on $\boldsymbol{s}$ but on the information type m^* of the stage, and are given as

$$t_m^a(m^*) = \begin{cases} t_m & \text{for } m = m^* \\ t(t_m^a, ...) & \text{otherwise} \end{cases}$$

where $t(t_m^a, ...)$ is any function of t_m^a and potentially other factors, for example the function $t(t_m^a, c) = t_m^a/c$ seen in Section 4.1.2. Note that in this scenario out of order and already discovered types are treated the same. As a consequence, the expected time to discover a relevant tip of prescribed type m^* on any source a does not depend on the state of the puzzle, only on m^*;

$$\mathbb{E}\left[\mathcal{T}^a(\boldsymbol{s}, m^*)\right] = \mathbb{E}\left[\mathcal{T}^a(m^*)\right] \tag{4.2.42}$$

$$= \frac{\sum_{m=1}^{\mathcal{M}} q_m^a t_m^a(m^*)}{q_{m^*}^a p_{m^*}^a} \tag{4.2.43}$$

$$= \frac{q_{m^*}^a t_{m^*}^a + \sum_{m \neq m^*} q_m^a t(t_m^a, ...)}{q_{m^*}^a p_{m^*}^a}. \tag{4.2.44}$$

Since under such circumstances the time to discovery of type m^* is independent of the state of the problem: if all $w(\boldsymbol{s}) = w$ are equal, the order in which the different types of magic bullets are discovered do not matter. We formalise this in Proposition 4.2.3 and Proposition 4.2.4.

Proposition 4.2.3. *When the following applies to magic bullets:*

(i) they must be discovered in a predetermined order,

(ii) the evaluation time of an out of order tip of a given type is the same as after discovering a magic bullet of said type,

(iii) $w(\boldsymbol{s}) = w$,

the expected time to completion of the optimal policy equals $\sum_{g=1}^{\mathcal{M}} \min\limits_{a} \{w\mathbb{E}\left[\mathcal{T}^a(\mathcal{O}(g))\right]\}$.

Proof. Let us assume that the order of discovery is given by $\mathcal{O}$. Since in the fixed order regime the state $\boldsymbol{s}$ is uniquely determined by the order and the stage $\boldsymbol{s} = \boldsymbol{s}(\mathcal{O}, g)$, for clarity we may change the notation and rewrite the Bellman equation using $\mathbb{V}(\boldsymbol{s}) = \mathbb{V}(\mathcal{O}, g)$ and the expanded form $\boldsymbol{s}(\mathcal{O}, g)$.

$$\mathbb{V}(\mathcal{O}, \mathcal{M}+1) = 0, \tag{4.2.45}$$

$$\mathbb{V}(\mathcal{O}, g) = \min_{m \in \mathcal{S}_m(\mathcal{O},g)} \left\{ \min_a \{w(\boldsymbol{s}(\mathcal{O}, g))\mathbb{E}\left[\mathcal{T}^a(\boldsymbol{s}(\mathcal{O}, g), m)\right]\} + \mathbb{V}\left(\mathcal{O}, g+1\right) \right\}. \tag{4.2.46}$$

Then using (4.2.42) and the assumption that $w(\boldsymbol{s}(\mathcal{O}, g)) = w$ (4.2.46) can be rewritten as

$$\mathbb{V}(\mathcal{O}, g) = \min_{m \in \mathcal{S}_m(\mathcal{O},g)} \left\{ \min_a \{w\mathbb{E}\left[\mathcal{T}^a(m)\right]\} + \mathbb{V}\left(\mathcal{O}, g+1\right) \right\} \tag{4.2.47}$$

and the value function of the starting state $\mathbb{V}(\mathcal{O}, 1)$ can be calculated recursively

$$\begin{aligned}
\mathbb{V}(\mathcal{O}, \mathcal{M}) &= \min_{m \in \mathcal{S}_m(\mathcal{O},g)} \left\{ \min_a \{w\mathbb{E}\left[\mathcal{T}^a(m)\right]\} + \mathbb{V}\left(\mathcal{O}, \mathcal{M}+1\right) \right\} \\
&= \min_a \{w\mathbb{E}\left[\mathcal{T}^a(\mathcal{O}(\mathcal{M}))\right]\} \\
\mathbb{V}(\mathcal{O}, \mathcal{M}-1) &= \min_{m \in \mathcal{S}_m(\mathcal{O},g)} \left\{ \min_a \{w\mathbb{E}\left[\mathcal{T}^a(m)\right]\} + \mathbb{V}\left(\mathcal{O}, \mathcal{M}\right) \right\} \\
&= \min_a \{w\mathbb{E}\left[\mathcal{T}^a(\mathcal{O}(\mathcal{M}-1))\right]\} + \min_a \{w\mathbb{E}\left[\mathcal{T}^a(\mathcal{O}(\mathcal{M}))\right]\} \\
&= \sum_{g=\mathcal{M}-1}^{\mathcal{M}} \min_a \{w\mathbb{E}\left[\mathcal{T}^a(\mathcal{O}(g))\right]\} \qquad (4.2.48) \\
&\vdots \\
\mathbb{V}(\mathcal{O}, 1) &= \sum_{g=1}^{\mathcal{M}} \min_a \{w\mathbb{E}\left[\mathcal{T}^a(\mathcal{O}(g))\right]\} \qquad (4.2.49)
\end{aligned}$$

since the pre-determined order prescribes $m^* = \mathcal{O}(g)$ at every stage so that $\mathcal{S}_m(\mathcal{O}, g)$ only ever has a single element. □

Proposition 4.2.4. *When the following applies to magic bullets:*

(i) they must be discovered in a predetermined order,

(ii) the evaluation time of an out of order tip of a given type is the same as after discovering a magic bullet of said type,

(iii) $w(\boldsymbol{s}) = w$*,*

the expected time to completion does not depend on the prescribed order of discovery and $\sum_{g=1}^{\mathcal{M}} \min\limits_{a} \{w\mathbb{E}\left[\mathcal{T}^a(\mathcal{O}(g))\right]\} = \sum_{m=1}^{\mathcal{M}} \min\limits_{a\in\mathcal{A}} w\mathbb{E}\left[\mathcal{T}^a(m)\right]$ *for every order* $\mathcal{O}$*.*

Proof. Assume that the order of discovery $\mathcal{O}^*$ is optimal. Let us define a modified order $\mathcal{O}'$ so that $\mathcal{O}'(l) = \mathcal{O}^*(n)$ and $\mathcal{O}'(n) = \mathcal{O}^*(l)$ where $l, n \in \{1, ..., \mathcal{M}\}$ and $l < n$. Then the value of the starting state given order $\mathcal{O}'$ can be found by rewriting (4.2.49)

$$\mathbb{V}(\mathcal{O}', 1) = \sum_{g=1}^{\mathcal{M}} \min_{a} \{w\mathbb{E}\left[\mathcal{T}^a(\mathcal{O}'(g))\right]\} \tag{4.2.50}$$

$$\begin{aligned} &= \min_{a} \{w\mathbb{E}\left[\mathcal{T}^a(\mathcal{O}'(1))\right]\} + ... + \min_{a} \{w\mathbb{E}\left[\mathcal{T}^a(\mathcal{O}'(n))\right]\} + ... \\ &+ \min_{a} \{w\mathbb{E}\left[\mathcal{T}^a(\mathcal{O}'(l))\right]\} + ... + \min_{a} \{w\mathbb{E}\left[\mathcal{T}^a(\mathcal{O}'(\mathcal{M}))\right]\}. \end{aligned} \tag{4.2.51}$$

Since summation is commutative we can rearrange

$$\begin{aligned} \mathbb{V}(\mathcal{O}', 1) &= \min_{a} \{w\mathbb{E}\left[\mathcal{T}^a(\mathcal{O}'(1))\right]\} + ... + \min_{a} \{w\mathbb{E}\left[\mathcal{T}^a(\mathcal{O}'(l))\right]\} + ... \\ &+ \min_{a} \{w\mathbb{E}\left[\mathcal{T}^a(\mathcal{O}'(n))\right]\} + ... + \min_{a} \{w\mathbb{E}\left[\mathcal{T}^a(\mathcal{O}'(\mathcal{M}))\right]\} \qquad (4.2.52) \\ &= \mathbb{V}(\mathcal{O}^*, 1), \qquad (4.2.53) \end{aligned}$$

therefore $\mathcal{O}'$ is also an optimal order achieving the minimising expected completion time. Since we made no assumptions on which two elements of the orders were exchanged, the above is also true for any other $\mathcal{O}' \neq \mathcal{O}^*$. Consequently, the order of discovery does not matter.

Make note that $\mathbb{E}\left[\mathcal{T}^a(\mathcal{O}(g))\right]$ only depend on $\mathcal{O}$ and g since they determine the type m to be discovered. As $\mathbb{E}\left[\mathcal{T}^a(m)\right]$ is the same irrespective of the stage it is assigned to be discovered in, the summation can be rewritten in terms of the types to get

$$\sum_{e=1}^{\mathcal{M}} \min_{a} \{w\mathbb{E}\left[\mathcal{T}^a(\mathcal{O}(g))\right]\} = \sum_{e=1}^{\mathcal{M}} \min_{a} \{w\mathbb{E}\left[\mathcal{T}^a(\mathcal{O}'(g))\right]\} = \sum_{m=1}^{\mathcal{M}} \min_{a\in\mathcal{A}} w\mathbb{E}\left[\mathcal{T}^a(m)\right]. \tag{4.2.54}$$

Intuitively, the orders $\mathcal{O}' \neq \mathcal{O}^*$ are simply different permutations of the $\mathcal{M}$ types of information, and the terms of $\mathbb{V}(\mathcal{O}^*)$ and $\mathbb{V}(\mathcal{O}')$ can be rearranged into any order of discovery through the commutative property of summation. □

If the evaluation times of out of order types of information are treated differently to those of already discovered types the symmetry is broken and the order in which the types are discovered affects the expected cost of completion. However, backwards recursion can still be used to find the optimal order.

Note that the above observations about optimality only hold when the specific set of circumstances are present; in general we expect a better solution to exist without the constraints on the parameters and the discoveries. Nevertheless, the above discussion opens the door to developing heuristic approaches.

Order Based Heuristics

Based on the findings of **??**, 4.2.3 and 4.2.4 we can define heuristic decision rules for the version of the intelligence puzzle with no ordering present, as set out in Section 4.2.1. We keep the idea of focusing our attention on one given type per stage, but acknowledge that the magic bullet discovered may not be of the type we intended to discover, as not only the type we focus on may yield magic bullets. This is in contrast to when a strict ordering of discoveries was present.

The backwards recursion as described in (4.2.40) can be used to give an order to discover the magic bullets in. This order is only optimal when discoveries must be made in order, but may still be a good order to aim to achieve when no order constraints are present.

Then the order-based dynamic program (O-DP) policy can be defined as follows. First, use (4.2.40) recursively to suggest the type m^* we wish to discover during the next stage

$$m^* = \underset{m \in \mathcal{S}_m(\boldsymbol{s})}{\arg\min} \left\{ \min_a \, w(\boldsymbol{s}) \mathbb{E}\left[\mathcal{T}^a(\boldsymbol{s}, m)\right] + \mathbb{V}\left(\boldsymbol{s}_{m^+}\right) \right\}. \tag{4.2.55}$$

This is an easier dynamic program to solve than the one in (4.2.7), as all stage transitions are deterministic. There is no point in setting an order for the rest of the problem as there is no guarantee it will be followed, as discoveries which in the restricted problem would be out of order still count. The source a^* is then chosen based on which type we aim to discover so that in every state $\boldsymbol{s}$

$$a^* = \pi^{\text{O-DP}}(\boldsymbol{s}) := \underset{a \in \mathcal{S}_a}{\arg\min} \left\{ \mathbb{E}\left[\mathcal{T}^a(\boldsymbol{s}, m^*)\right] \right\} = \underset{a \in \mathcal{S}_a}{\arg\min} \left\{ \frac{\sum_{m=1}^{\mathcal{M}} q_m^a t_m^a(\boldsymbol{s})}{q_{m^*}^a p_{m^*}^a} \right\}. \tag{4.2.56}$$

Note that the name is somewhat misleading as it is built on the assumption of fixed orders that is not guaranteed to be satisfied: while it is obtained via a dynamic program the policy is still only a heuristic one.

The other heuristic is a short-sighted version of the above, motivated by Propositions 4.2.3 and 4.2.4 showing that in some cases the order of discovery does not matter. Due to its short-sighted nature we name it the order-based myopic policy, O-MYO for short. The conditions of Propositions 4.2.3 and 4.2.4 (except for the order constraint) are met when there is no state dependence of the evaluation times and $w(\boldsymbol{s}) = w$. Under such circumstances in the ordered regime the order of discovery would not actually matter, and if the targeted type m^* could be announced along with the source chosen it would be optimal to pick the source a^* and type m^* combination with the shortest expected discovery time so that

$$a^* = \pi^{\text{O-MYO}}(\boldsymbol{s}) := \arg\min_a \left\{ \min_{m* \in \mu} \left\{ \frac{\sum_{m=1}^{\mathcal{M}} q_m^a t_m^a(\boldsymbol{s})}{q_{m^*}^a p_{m^*}^a} \right\} \right\}. \tag{4.2.57}$$

Since these conditions are quite strict and are not generally met, (4.2.57) is not expected to perform optimally. Note that for all a

$$\min_{m^* \in \mu} \left\{ \frac{\sum_{m=1}^{\mathcal{M}} q_m^a t_m^a(\boldsymbol{s})}{q_{m^*}^a p_{m^*}^a} \right\} \geq \frac{\sum_{m=1}^{\mathcal{M}} q_m^a t_m^a(\boldsymbol{s})}{\sum_{m=1}^{\mathcal{M}} q_m^a p_m^a(\boldsymbol{s})} = \mathbb{E}\left[\mathcal{T}^a(\boldsymbol{s})\right], \tag{4.2.58}$$

where the equality occurs at the last stage where only one type of magic bullet remains to be discovered. Therefore we find that the decision rule in (4.2.57) is based on the upper bound of $\mathbb{E}\left[\mathcal{T}^a(\boldsymbol{s})\right]$ as shown in (4.2.58).

Some Discarded Policies

Some other policies were also considered but quickly discarded. They are included only for the sake of completeness.

The first idea was to aggregate all states belonging to the same stage reducing the state space and therefore the computational complexity of the dynamic program used to solve the intelligence puzzle. In that case decisions are made based not on the state of the problem, but based on the stage of the problem.
We can write the modified dynamic program as

$$\begin{aligned}\mathbb{V}\left(\mathcal{M}+1\right) &= 0,\\ \mathbb{V}\left(g\right) &= \min_{a}\left\{w(g)\mathbb{E}\left[\mathcal{T}^{a}\left(g\right)\right] + \mathbb{P}(g+1 \mid g,a)\mathbb{V}\left(g+1\right)\right\}\\ &= \min_{a}\left\{w(g)\mathbb{E}\left[\mathcal{T}^{a}\left(g\right)\right]\right\} + \mathbb{V}\left(g+1\right),\end{aligned}$$

where $\mathbb{E}\left[\mathcal{T}^{a}\left(g\right)\right]$ is the expected time spent in stage g given source a is selected, $\mathbb{P}(g+1 \mid g,a)$ is the transition probability from stage g to $g+1$ given source a is selected and $\mathbb{V}\left(g+1\right)$ is the value function of stage $g+1$. Since states are no longer considered, $w(g)$ is stage dependent surrogate to $w(\boldsymbol{s})$.

However, $\mathbb{P}(g+1 \mid g,a) = 1$ for all g and a due to progression to the next stage being guaranteed. As a consequence the value function of the next stage does not play a role in selecting an source to continue. Such a policy is not only short-sighted but also ignores the information contained in the current state without any benefit to balance that ignorance out. Therefore a policy based only on the stage of the puzzle would be expected to perform worse than the myopic policy in every scenario and will not be further considered.

The next discarded idea was to consider the sources independently of the others within the puzzle. Let us assume that once a source is selected, it will be played until all types of magic bullets are discovered. At every stage the same source a is selected so that the expected cost of discovering all remaining pieces of intelligence under the above assumptions, $\mathbb{V}(\boldsymbol{s}, a)$, is minimised. That is

$$a^* := \arg\min_a \{\mathbb{V}(\boldsymbol{s}, a)\}.$$

However, to be able to calculate the expected costs, all states and the transition probabilities between them need to be considered, and so it is easiest to calculate recursively as follows.

$$\begin{aligned}\mathbb{V}(\boldsymbol{s}_0, a) &= 0,\\ \mathbb{V}(\boldsymbol{s}, a) &= w_e \mathbb{E}\left[\mathcal{T}^a(\boldsymbol{s})\right] + \sum_{m=1}^{\mathcal{M}} \mathbb{P}\left(\boldsymbol{s}_{m^+} \mid \boldsymbol{s}, a\right) \mathbb{V}\left(\boldsymbol{s}_{\boldsymbol{m}^+}, a\right).\end{aligned}$$

Since all states of the puzzle must be considered to recursively calculate $\mathbb{V}(\boldsymbol{s}, a)$, we expect no reduction in the computational complexity compared to that of the optimal policy in Section 4.2.2. For that reason this policy of considering the sources separately is not taken further forward.

4.2.3 Numerical Experiments

The performance of the policies described in Section 4.2.2 is evaluated in a numerical study based on simulations. For simplicity we set the cost rate of every non-terminal ($\boldsymbol{s} \neq \boldsymbol{0}_{\mathcal{M}}$) state to $w(\boldsymbol{s}) = 1$ so that costs accrue at the same rate while the puzzle remains incomplete. In this scenario the cost of completing the puzzle is the time taken to complete the puzzle, and therefore the objective of minimising the completion cost is equivalent to minimising the completion time. In this numerical study the parameter vectors of the simulated puzzles $\boldsymbol{q}^a$ and $\boldsymbol{p}^a$ are selected randomly, but all base evaluation times were set to be $t_m^a = 1$. Note that once all parameter values are sampled, they remain the same throughout the puzzle and are made known to the decision maker. The encounter probability vectors $\boldsymbol{q}^a$ are sampled from a Dirichlet($\boldsymbol{1}_{\mathcal{M}}$) distribution, while all conditional success probabilities p_m^a are simply sampled from a Beta(1,1)≡Uniform(0,1) distribution.

The simulations were run for $\mathcal{A} = 2$ and $\mathcal{A} = 5$, each with $\mathcal{M}$ ranging between 2 and 10. Every such combination of $\mathcal{A}$ and $\mathcal{M}$ was examined with 3 different values of c, the factor governing the evolution of the evaluation times. These were $c = \{1, 5, \infty\}$. As per (4.1.8) $c = 1$ corresponds to the evaluation times remaining unchanged throughout the puzzle, and $c = \infty$ means no time is spent on evaluating tips of already discovered types.

For every combination of $\mathcal{A}$, $\mathcal{M}$ and c we sampled 10^5 parameter combinations to generate intelligence puzzles with. While we wish to make general observations on the performance of the policies discussed, we must recognise that every parameter combination creates a different intelligence puzzle and we cannot simply average over the instances. Instead, every sampled parameter combination was simulated 10 times and averaged over to provide the observation for that parameter combination. The fact that this is quite a low number to average results over is recognised, but our efforts were limited by the computational capacity of the systems available.

We compare the policies mainly in terms of the relative sub-optimality. To obtain the mean relative sub-optimality of a policy π, the relative sub-optimality $\mathcal{SO}^{\pi}$ is calculated for every parameter combination as

$$\mathcal{SO}^{\pi} = \frac{\mathcal{C}^{\pi} - \mathcal{C}^{\pi^{\mathrm{OPT}}}}{\mathcal{C}^{\pi^{\mathrm{OPT}}}}, \tag{4.2.59}$$

which is the relative difference between the cost of completion $\mathcal{C}$ of the intelligence puzzle using the optimal policy π^{OPT} given in Section 4.2.2 and of policy π. It is then averaged over all simulated parameter combinations to produce the mean relative sub-optimality $\overline{\mathcal{SO}}^{\pi}$. To quantify the uncertainty in this measure 95% bootstrap confidence intervals are used. Figure 4.2.2 and Figure 4.2.3 show the mean relative sub-optimality as a function of $\mathcal{M}$ for $\mathcal{A} = 2$ and $\mathcal{A} = 5$ respectively. Acronyms are used to refer to the policies as seen before in Section 4.2.2: MYO stands for the myopic policy, LA for the 1-step look-ahead policy, O-DP for the order based dynamic programming policy and O-MYO for the order based myopic policy.

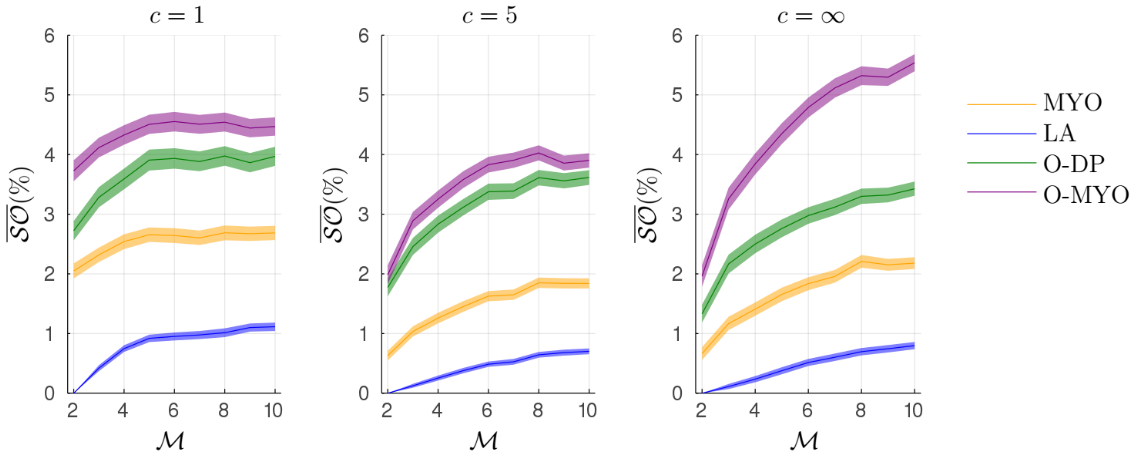


Figure 4.2.2: Mean relative sub-optimality $\overline{\mathcal{SO}}$ for $\mathcal{A} = 2$ as a function of $\mathcal{M}$.

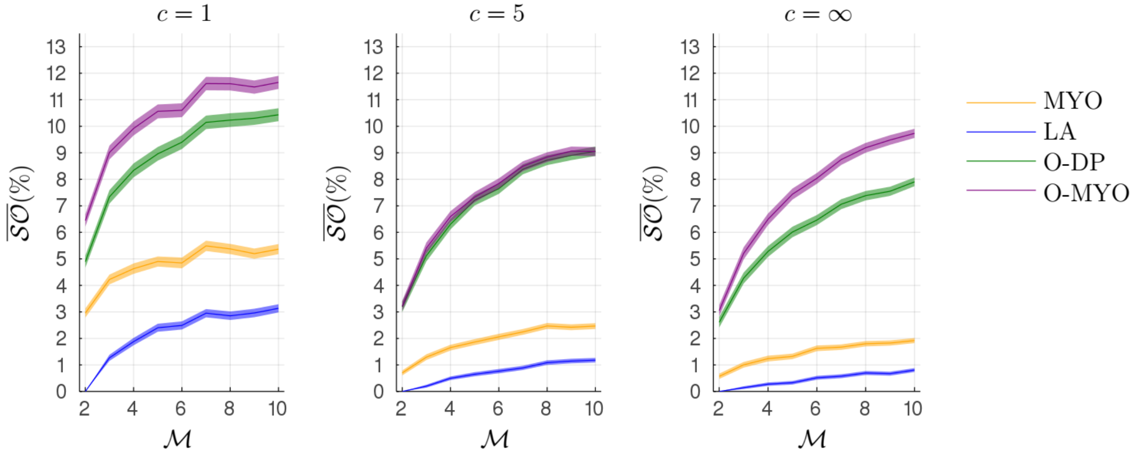


Figure 4.2.3: Mean relative sub-optimality $\overline{\mathcal{SO}}$ for $\mathcal{A} = 5$ as a function of $\mathcal{M}$.

It is evident that the methods established under the assumption that there is a set order in which the magic bullets need to be discovered perform worse than those without this assumption. Since such an assumption significantly changes the dynamics of the intelligence puzzle, it is not surprising that policies do not transfer well to the full problem. Looking at the relative sub-optimalities of the myopic policy and the one-step look-ahead policy the benefit of considering what might happen in the next stage is clear. At $\mathcal{M} = 2$ the one-step look-ahead policy is the same as the optimal policy, but even for $\mathcal{M} > 2$ it consistently outperforms the myopic policy, as is expected of a method that is a compromise between the myopic and the optimal policy.

As Table 4.2.1 and Table 4.2.2 demonstrates, the computational effort to determine which source should be chosen is significantly higher for the optimal policy than it is for the 1-step look-ahead or the myopic policy. The difference increases with the number of types of information present. However, it is important to note that we only need to compute the optimal policy once for any set of parameters, and refer back to that throughout the problem. Unless many puzzles need to be considered in quick succession or the number types increases beyond those shown here, even the optimal policy computes in a reasonable time-frame.

$\mathcal{M}$	MYO	LA	DP
2	0.45 μs	9.36 μs	12.20 μs
5	0.46 μs	28.21 μs	335.81 μs
10	0.47 μs	122.64 μs	18.78 ms

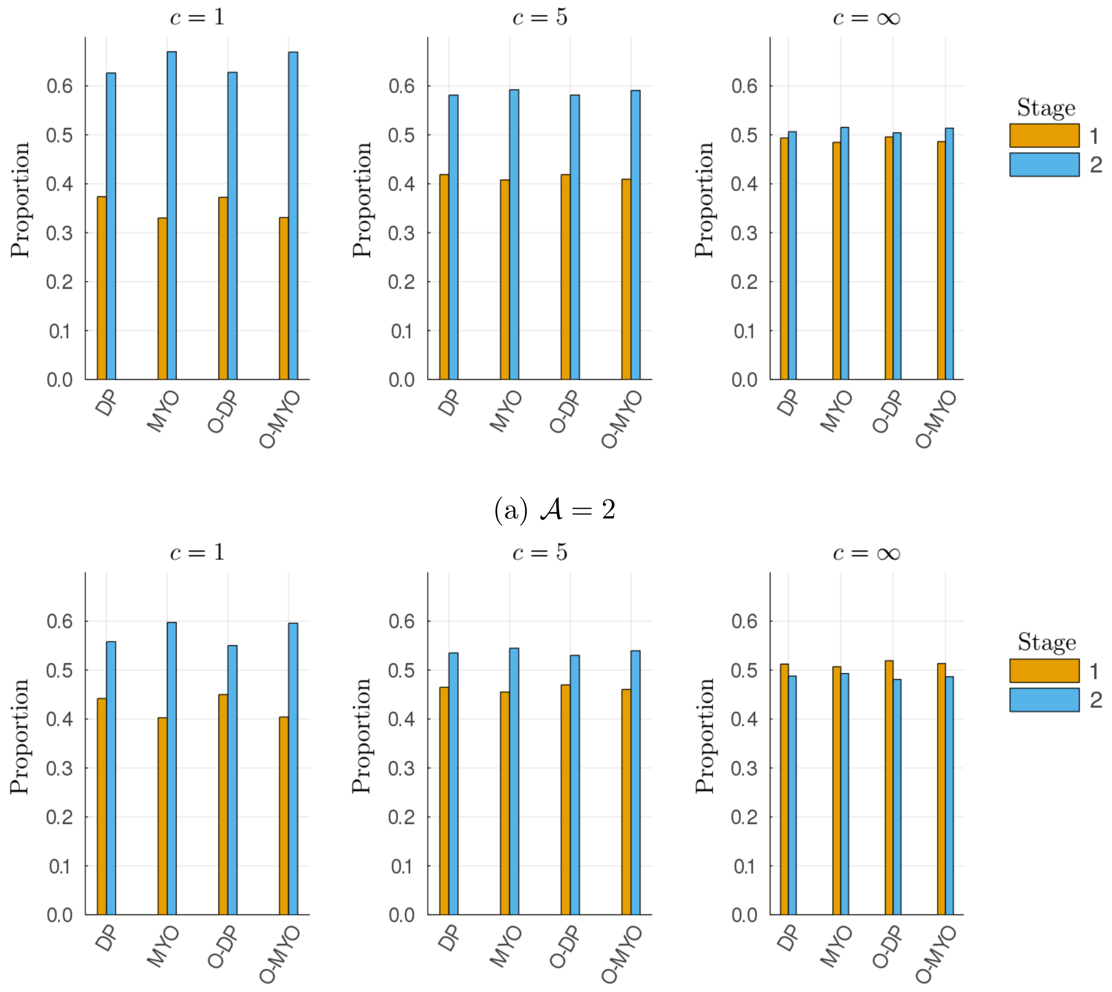
Table 4.2.1: Computation time to determine the source to sample from in state $\boldsymbol{s} = \mathbf{1}_{\mathcal{M}}$, for $\mathcal{A} = 2$

M	MYO	LA	DP
2	2.67 μs	143.13 μs	222.25 μs
5	2.84 μs	251.69 μs	2.05 ms
10	2.87 μs	480.49 μs	87.79 ms

Table 4.2.2: Computation time to determine the source to sample from in state $\boldsymbol{s} = \mathbf{1}_{\mathcal{M}}$, for $\mathcal{A} = 5$

Comparing Figure 4.2.2 and Figure 4.2.3 against each other we can see that in general $\overline{\mathcal{SO}}$ is larger for $\mathcal{A} = 5$ than for $\mathcal{A} = 2$. This is because more sub-optimal choices in the form of more sources are available. The increase in sub-optimality as $\mathcal{M}$ is increased has similar origins; the more stages there are to the problem the more choices of the decision-maker have the chance to be sub-optimal.

To shed a bit more light on how the completion time is split between the different stages of the puzzle, the completion times of the individual stages are examined in detail, focusing on the $\mathcal{M} = 2$ and $\mathcal{M} = 5$ cases. First, we compare the policies in terms of the proportion of the total completion time spent on the different stages as shown in Figure 4.2.4 for $\mathcal{M} = 2$.



(a) $\mathcal{A} = 2$

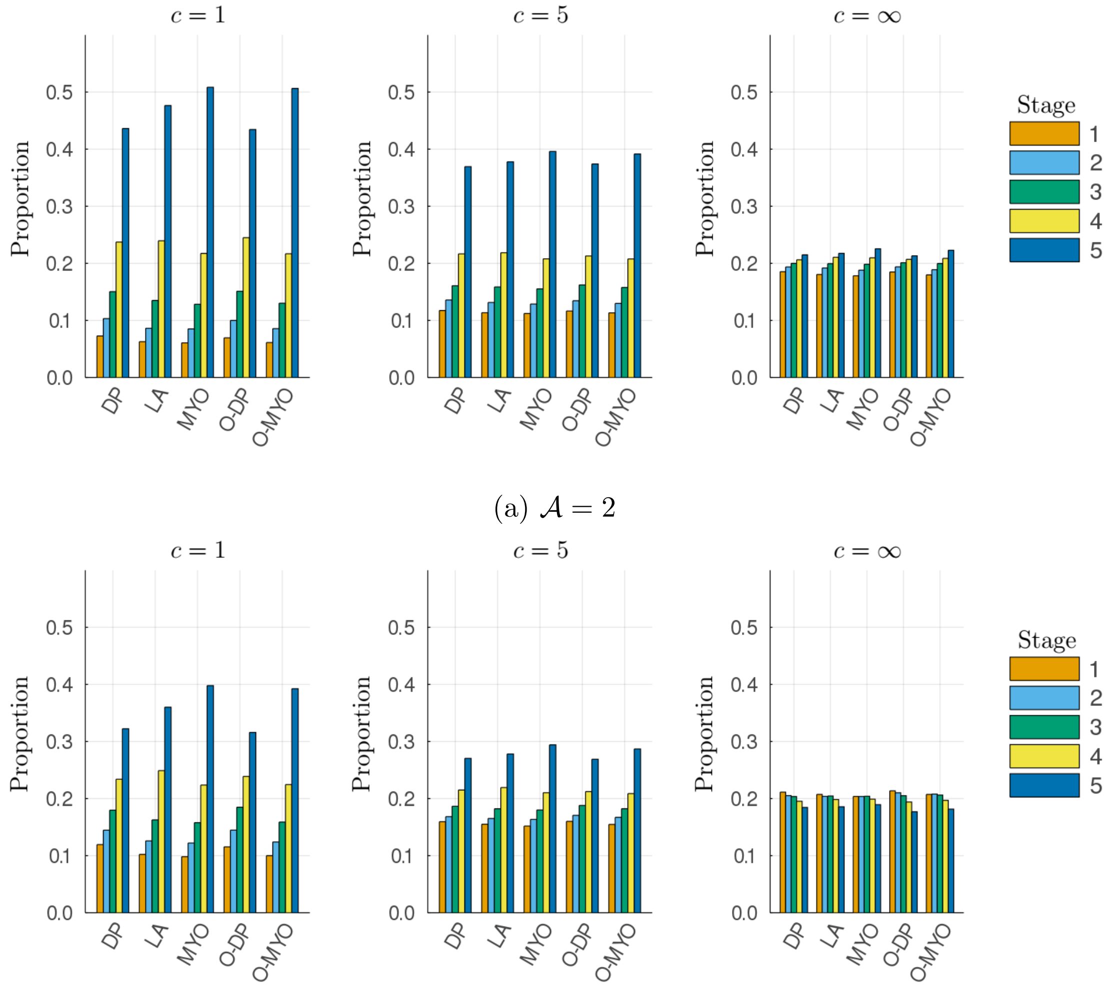
(b) $\mathcal{A} = 5$

Figure 4.2.4: Proportion of completion time spent on each stage for $\mathcal{M} = 2$. Since in the $\mathcal{M} = 2$ case the LA policy is identical to the DP policy it is not shown here.

At first glance the policies examined perform similarly, with their differences slightly more distinct in the $\mathcal{A} = 5$ case. While the difference is subtle, the policies that look ahead to future stages spend a higher proportion of the completion time on the first stage compared to those that do not. Note that the difference in the proportion of time spent on stage 1 and stage 2 is much larger for $\mathcal{A} = 2$ than for $\mathcal{A} = 5$, which is a direct consequence of the number of sources available. In the last stage of the puzzle, all but one type of tips will exclusively produce nuisance tips, and the decision maker will spend a large amount of time evaluating these. When there are more sources to choose from, it is more likely that at least one source will have a reasonably good combination of encounter and success probabilities so that this wasted time is reduced. While having a larger selection of these randomly generated sources shortens the duration of all stages of the puzzle, it is most difficult to find a magic bullet in the last stage and therefore the biggest effect is on the discovery time of the final magic bullet. Shortening the duration of the last stage leads to it taking up a smaller proportion of the completion time, automatically increasing the proportion of time taken up by earlier stages.

Figure 4.2.5 shows the proportion of time spent on the stages for the $\mathcal{M} = 5$ case. Similar observations can be made; for most combinations of $\mathcal{M}$, $\mathcal{A}$ and c the later the stage is, the larger the proportion of the evaluation time it takes up, and these differences are more pronounced in the $\mathcal{A} = 2$ case.

The time spent on already discovered types can be used to explain another, quite interesting feature. As c is increased, the difference between the proportion of time taken up by the individual stages of the puzzle is seen to decrease. Since the effect of c is to shorten the evaluation time of already discovered types of information, the larger c is the less time is wasted on these types of tips. By spending less time on tips that cannot yield a magic bullet the amount of time spent on discovering magic bullets at later stages is significantly reduced, lowering the proportion of time late stages contribute to the total completion time. This is taken to the extreme when $c = \infty$ and no time is spent on already discovered types, resulting in almost equal time spent on all stages. Furthermore, combined with the previously described effect having a wider variety of sources has, in the case of $\mathcal{A} = 5$ the order of proportions have flipped, with



(a) $\mathcal{A} = 2$

(b) $\mathcal{A} = 5$

Figure 4.2.5: Proportion of completion time spent on each stage for $\mathcal{M} = 5$.

earlier stages taking more time than later ones. This is not too surprising as all types are encountered with the same probability throughout the stages, but while during the first stage all of the encountered tips need evaluating and therefore time spent on, during the last stage time is spent only on evaluating one type of tips.

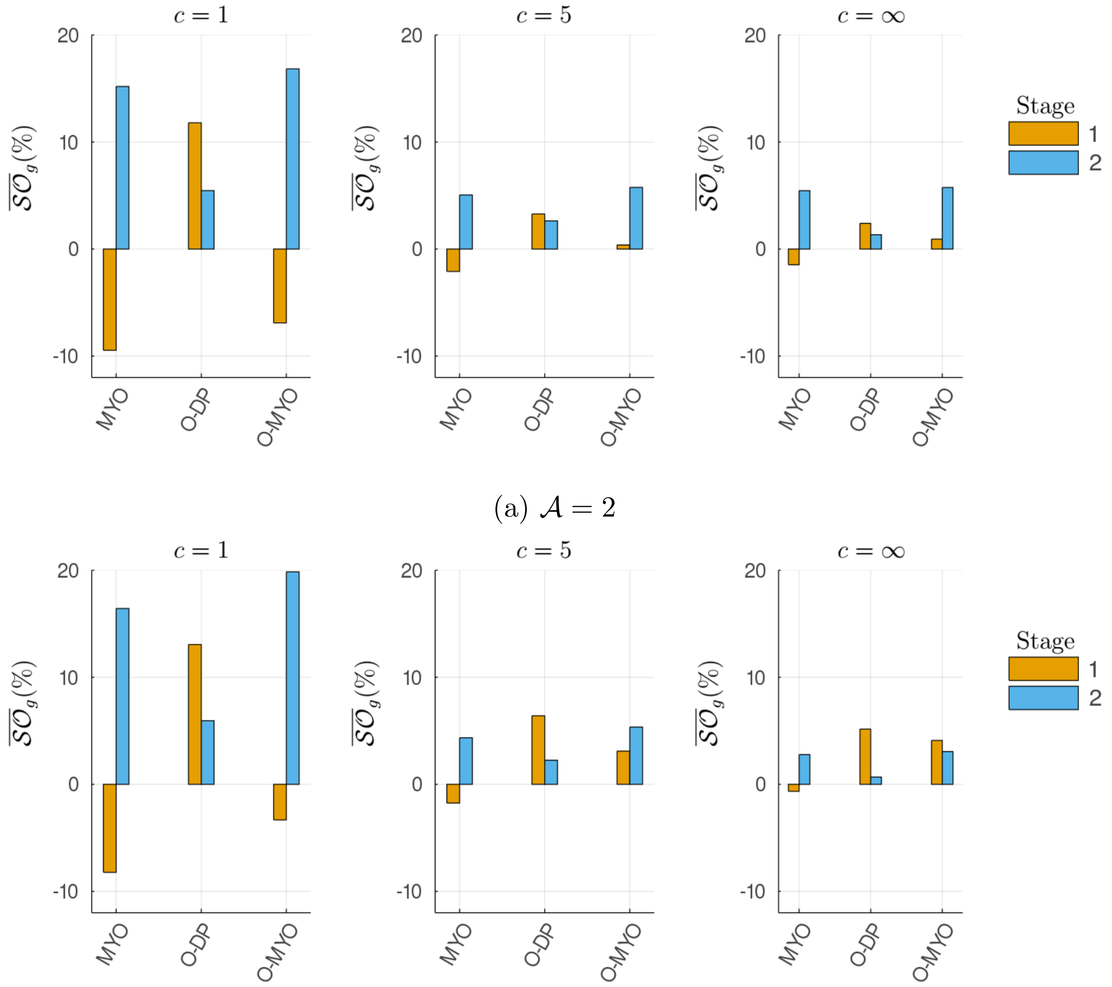
The sub-optimality measure used earlier can be adapted to use on the completion times of the stages instead of the completion time of the entire puzzle. The relative sub-optimality of stage g, $\mathcal{SO}_g^{\pi}$, is calculated for every parameter combination as

$$\mathcal{SO}_g^{\pi} = \frac{\mathcal{C}_g^{\pi} - \mathcal{C}_g^{\pi^*}}{\mathcal{C}_g^{\pi^*}}, \tag{4.2.60}$$

which is the relative difference between the cost of completion of stage g, $\mathcal{C}_g$, of the intelligence puzzle using the optimal policy π^* given in Section 4.2.2 and of policy π. Note that as opposed to the sub-optimality measure before, the stage-wise relative sub-optimality can take on negative values if a particular stage takes less time to complete under policy π than under π^*. This does not discredit π^* as the optimal policy since it is only optimal in terms of the expected total completion time. To obtain the mean relative sub-optimality of the stage $\overline{\mathcal{SO}}_g^{\pi}$ the $\mathcal{SO}_g^{\pi}$ are averaged over all simulated parameter combinations. Figure 4.2.6 shows this stage-wise sub-optimality for $\mathcal{M} = 2$, and Figure 4.2.7 for $\mathcal{M} = 5$.

These figures demonstrate that the different stages do not contribute equally to the overall sub-optimality; short sighted policies demonstrate a high level of sub-optimality in the last stage, while in earlier stages they have negative sub-optimality, having found magic bullets quicker than the optimal policy. Indeed, both the myopic and the order-based myopic policy aim to minimise the discovery time in the current stage, which the results show they are successful at in the early stages of the problem. On the other hand, their short-sightedness can lead them into a disadvantageous state in the last stage which results in a large stage-wise relative sub-optimality. Since the duration of the last stage tends to be the longest, a large stage-wise sub-optimality will have a large impact on the overall sub-optimality. The impact of incorrect assumptions can be seen when comparing the myopic policy and the order-based myopic policy. While the myopic behaviour is clearly seen in the figures, the order-based myopic policy has consistently greater sub-optimality than the myopic policy.

The one-step look-ahead policy is also short-sighted, even if to a lesser extent. As expected, in the $\mathcal{M} = 2$ case the relative sub-optimality associated with either of the stages is 0, as the one-step look-ahead policy is identical to the optimal policy. How-

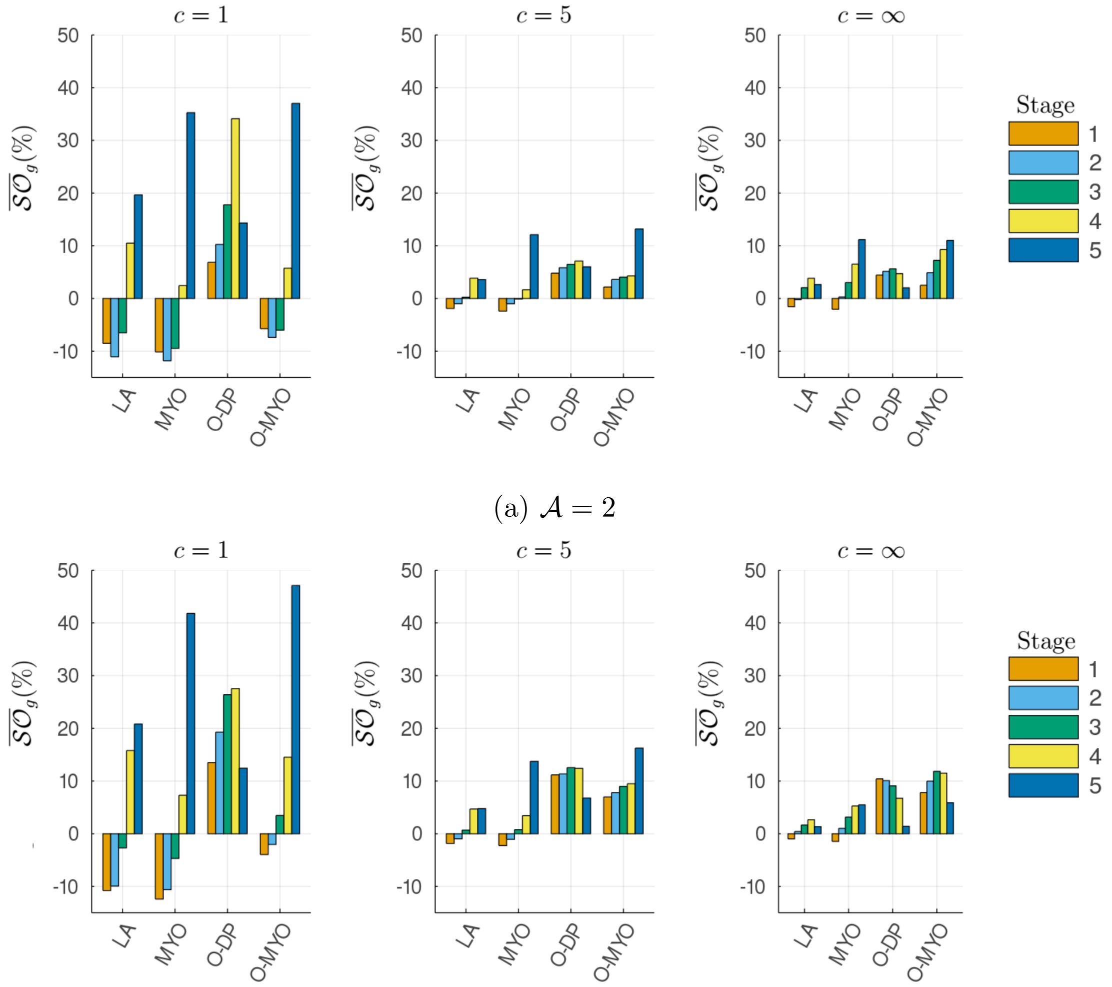


(a) $\mathcal{A} = 2$

(b) $\mathcal{A} = 5$

Figure 4.2.6: Stage-wise mean relative sub-optimality $\overline{\mathcal{SO}}_g$ for $\mathcal{M} = 2$. Since in the $\mathcal{M} = 2$ case the one-step look-ahead policy is identical to the optimal policy it is not shown here.

ever, for $\mathcal{M} > 2$ looking ahead for one step only is no longer optimal. As shown in Figure 4.2.7, for the $\mathcal{M} = 5$ case this policy's ability to look ahead only one step makes relatively little difference during the early stages of the intelligence puzzle and it behaves similarly to the myopic policy. The difference between the one-step look-ahead and the myopic policy becomes apparent when looking at the previous to last stage of the puzzle. At that point the look ahead policy can consider all the remaining stages, and minimise the completion time of the puzzle from that point onward. This reduces



(a) $\mathcal{A} = 2$

(b) $\mathcal{A} = 5$

Figure 4.2.7: Stage-wise mean relative sub-optimality $\overline{\mathcal{SO}}_g$ for $\mathcal{M} = 5$.

the stage-wise sub-optimality of the last and longest stage, giving a clear advantage to the look-ahead policy over the myopic policy.

While the order-based dynamic program does not suffer from the same issues as the other methods and so are sub-optimal to a smaller extent than the myopic and the order-based myopic policies for the last stage, the incorrect assumption of an enforced discovery order impacts it strongly. For no stage does it achieve a shorter completion time than the optimal policy and therefore becomes increasingly sub-optimal with every stage, leading to a generally poor performance as seen in Figure 4.2.2 and Figure 4.2.3.

4.3 The Intelligence Puzzle with Unknown Conditional Success Probabilities

The intelligence puzzle can be extended so that the information available to the decision maker regarding some aspects of the problem is limited. An example of limited availability of information is a scenario where any or all of the parameter vectors are random, and only known to the decision maker through learning their probability distributions. Such a model is more reflective of what may be encountered in aiming to solve a real intelligence question, as the quality of their sources is rarely pre-established.

In this section we focus on the version of the intelligence puzzle where the conditional success probabilities are given by the random vector $\boldsymbol{P}^a$. The task of the decision maker is not only to solve the puzzle, but in order to do so learn about the distribution of the $\boldsymbol{P}^a$'s. The encounter probabilities $\boldsymbol{q}^a$ and the evaluation times $\boldsymbol{t}^a(\boldsymbol{s})$ are known quantities and behave as described in Section 4.1.2, no different than their behaviour in the basic intelligence puzzle. Then the base parameter vectors characterising each source $a \in \{1, ..., A\}$ are

$$\boldsymbol{q}^a = \begin{bmatrix} q_1^a \\ q_2^a \\ \cdots \\ q_{\mathcal{M}}^a \end{bmatrix}, \qquad \boldsymbol{P}^a = \begin{bmatrix} P_1^a \\ P_2^a \\ \cdots \\ P_{\mathcal{M}}^a \end{bmatrix}, \qquad \boldsymbol{t}^a = \begin{bmatrix} t_1^a \\ t_2^a \\ \cdots \\ t_{\mathcal{M}}^a \end{bmatrix}.$$

While $\boldsymbol{q}^a$ and the base evaluation times $\boldsymbol{t}^a$ remain unchanged throughout the puzzle, our belief of the distribution of $\boldsymbol{P}^a$ evolves with every tip evaluated from source a. Note that the state dependence of the random conditional success probability follows the same pattern as its known counterpart in Section 4.1.2.

Section 4.3.1 describes the conjugate Bayesian learning structure used to update our beliefs about $\boldsymbol{P}^a$, while Section 4.3.2 describes the extended intelligence puzzle with all its new features in the framework of semi-Markov decision processes. Section 4.3.3 follows on with a discussion of potential policies, many of which have been adapted from Section 4.2.2. Finally, a numerical study to evaluate these policies is presented in Section 4.3.4.

4.3.1 Bayesian Learning of $\boldsymbol{P}^a$

Bayesian learning is used to estimate the distribution of all conditional success probabilities. For this, we assume that all P_m^a are independent of each other and the other parameter vectors. For the remainder of this subsection we focus on a single source a, and omit the superscript in $\boldsymbol{P}^a$ denoting the source in question.

As described in Section 4.1.2 the relevance of a tip of type m is modelled by a Bernoulli random variable X_m. However, now the success probability given by P_m, which itself is a random variable. A success corresponds to discovering the one and only magic bullet of type m, while a failure corresponds to a nuisance tip. To incorporate our initial beliefs we place a conjugate Beta$(\alpha_m^{(0)}, \beta_m^{(0)})$ prior on P_m, where $\alpha_m^{(0)}$ and $\beta_m^{(0)}$ are the initial shape parameters of the Beta distribution and the superscript denotes the number of observations that have been made. The structure of the Bayesian learning problem can be summarised as

$$\begin{aligned} X_m \mid P_m &\sim \text{Bernoulli}(P_m), \\ P_m &\sim \text{Beta}(\alpha_m^{(0)}, \beta_m^{(0)}), \\ P_m \mid \boldsymbol{x}_m &\sim \text{Beta}(\alpha_m^{(n)}, \beta_m^{(n)}), \end{aligned}$$

where $\boldsymbol{x}_m$ is the vector of n observations of X_m. The posterior parameters are updated as observations are absorbed. Due to the simple conjugate structure the updates are as follows

$$\begin{aligned} \alpha_m^{(n)} &= \alpha_m^{(n-1)} + x_m^{(n)}, \\ \beta_m^{(n)} &= \beta_m^{(n-1)} + 1 - x_m^{(n)}. \end{aligned}$$

In the above $x_m^{(n)}$ is the n^{th} realisation of X_m. In essence $\alpha_m^{(n)}$ is the sum of $\alpha_m^{(0)}$ and all observed successes, while $\beta_m^{(n)}$ is the sum of $\beta_m^{(n)}$ and all observed failures. However, it must be noted that all success probabilities of type m are set to 0 once a magic bullet of said type $x_m = 1$ is encountered, and therefore in practice the $\alpha_m^{(n)}$'s are never updated, making their initial values critical.

It is useful to note the expected values of P_m, which is

$$\mathbb{E}\left[P_m \mid \boldsymbol{X}_m = \boldsymbol{x}_m\right] = \frac{\alpha_m^{(n)}}{\alpha_m^{(n)} + \beta_m^{(n)}}, \tag{4.3.1}$$

From (4.3.1) it is clear that the expectation only exists if $\alpha_m^{(n)} > 1$ for all n. Due to the update structure of $\alpha^{(n)}$ this can be ensured by requiring $\alpha_m^{(0)} > 1$. We can see from (4.3.1) the detrimental effect on the expectations never updating $\alpha_m^{(0)}$ has. In such a case, the expectation of P_m never goes above $\frac{\alpha_m^{(0)}}{\alpha_m^{(0)}+\beta_m^{(0)}}$, and further decreases with every update of β_m.

With the Bayesian learning aspect of the problem discussed, we return to considering all sources and denoting a specific source by the superscript a.

4.3.2 Setting Out the SMDP Framework

In this section we formulate the intelligence puzzle as a semi-Markov decision process, closely following and referring back to Section 4.2.1.

When all parameter vectors were known, the state space of the intelligence puzzle was sufficiently described by the collection of physical states as set out in Section 4.1.2. However, in this version of the puzzle our belief of P_m^a can change through updates to its posterior distribution, and the physical states are no longer sufficient to capture the state of the intelligence puzzle on its own. We need to introduce the *knowledge state* of the puzzle, which corresponds to the collection of our beliefs regarding every P_m^a each best captured by their posterior β_m^a parameters. While the α_m^a's form part of our belief, as highlighted in Section 4.3.1 for all practical purposes they are never updated and therefore need not form part of the knowledge state.

Let us define $\boldsymbol{\alpha}^a$ as the vector of posterior α_m^a parameters and $\boldsymbol{\beta}^a$ as the vector of posterior β^a_m parameters

$$\boldsymbol{\alpha}^a = \begin{bmatrix} \alpha^a_1 \\ \dots \\ \alpha^a_{\mathcal{M}} \end{bmatrix}, \qquad \boldsymbol{\beta}^a = \begin{bmatrix} \beta^a_1 \\ \dots \\ \beta^a_{\mathcal{M}} \end{bmatrix}.$$

Then the *knowledge state space* of the intelligence puzzle $\mathcal{S}_{\boldsymbol{\beta}}$ is given by the set of all possible $\boldsymbol{\beta}$'s where

$$\boldsymbol{\beta} = (\boldsymbol{\beta}^1, ..., \boldsymbol{\beta}^{\mathcal{A}}).$$

Since the number of trials required to observe a magic bullet of type m has no upper limit, neither does the value of β^a_m's and the knowledge state space $\mathcal{S}_{\boldsymbol{\beta}}$ is infinite.

The *physical state space* $\mathcal{S}_{\boldsymbol{s}}$ of the intelligence puzzle is as set out in Section 4.1.2, with the individual states denoted by a vector $\boldsymbol{s}$. Its elements s_m represent the state of intelligence type m, $s_m = 0$ and $s_m = 1$ corresponding to having or having not been discovered.

The *combined state space* (or state space for short) of the intelligence puzzle is the product of the knowledge and physical state space; $\mathcal{S}_{\boldsymbol{s},\boldsymbol{\beta}} = \mathcal{S}_{\boldsymbol{s}} \times \mathcal{S}_{\boldsymbol{\beta}}$, and its state is the combination of its physical and knowledge state, denoted as the tuple $(\boldsymbol{s}, \boldsymbol{\beta})$. While the physical state space is finite, the knowledge state space is not, and therefore the state space of the intelligence puzzle with learning is also infinite.

Similarly to Section 4.2.1, *decision epochs* occur at random, upon state transitions. In contrast to that previous model, the transitions do not have to be with respect to the physical state, knowledge state transitions also trigger a new decision epoch. In context, decision epochs correspond to the completion of the evaluation of individual tips, which either result in physical state transition if a magic bullet is discovered or a knowledge state transition if not. Therefore the *sojourn times* directly correspond to the evaluation times of the tips. The start and end-point of individual *stages* of the puzzle, described in detail in Section 4.1.2, coincide with decision epochs, but multiple decision epochs may occur during a single stage. As before, a puzzle consisting of $\mathcal{M}$

types of information has $\mathcal{M}$ stages.

At every decision epoch an action a must be taken, chosen from the set of available actions $\mathcal{S}_a$ which corresponds to the sources available. As in the previous section, the action set $\mathcal{S}_a = \{1, ..., \mathcal{A}\}$ is independent of the state of the decision process; any action may be taken at any decision epoch. We consider only deterministic stationary Markov policies in our investigations, so that policy π which determines the action taken at a given epoch only depends on the state of the system and prescribes the action $\pi(\boldsymbol{s}, \boldsymbol{\beta})$.

The action a taken in state s determines the probability distribution of both the subsequent states, and the time till the next decision epoch. Since only one tip is evaluated in each epoch, obtaining these quantities is straight-forward. First let us denote the probability that the process transitions from state $(\boldsymbol{s}, \boldsymbol{\beta})$ to $(\boldsymbol{s}_{m^+}, \boldsymbol{\beta})$ by $\mathbb{P}(\boldsymbol{s}_{m^+}, \boldsymbol{\beta} \mid \boldsymbol{s}, \boldsymbol{\beta}, a)$ in response to action a. The probability that it transitions to $(\boldsymbol{s}, \boldsymbol{\beta}^a_{m^-})$, where $\boldsymbol{\beta}^a_{m^-}$ denotes that the knowledge state has changed in response to observing a failure of type m on source a is given by $\mathbb{P}(\boldsymbol{s}, \boldsymbol{\beta}^a_{m^-} \mid \boldsymbol{s}, \boldsymbol{\beta}, a)$. Note that one-epoch transitions involve either the physical or the knowledge state, never both. Since the physical state transitions to $\boldsymbol{s}_{m^+}$ upon discovery of a type m magic bullet, the probability of such transition is the product of the probability of encountering a type m tip and the expectation of the conditional probability that it leads to discovering a magic bullet

$$\mathbb{P}(\boldsymbol{s}_{m^+}, \boldsymbol{\beta} \mid \boldsymbol{s}, \boldsymbol{\beta}, a) = q^a_m \mathbb{E}\left[P^a_m(\boldsymbol{s}, \alpha^a_m, \beta^a_m)\right] = q^a_m \frac{\alpha^a_m s_m}{\alpha^a_m + \beta^a_m}. \tag{4.3.2}$$

Similarly, the knowledge state transitions to $\boldsymbol{\beta}^a_{m^-}$ when the tip does not provide a magic bullet. This occurs with probability

$$\mathbb{P}(\boldsymbol{s}, \boldsymbol{\beta}^a_{m^-} \mid \boldsymbol{s}, \boldsymbol{\beta}, a) = q^a_m \left(1 - \mathbb{E}\left[P^a_m(\boldsymbol{s}, \alpha^a_m, \beta^a_m)\right]\right) = q^a_m \left(1 - \frac{\alpha^a_m s_m}{\alpha^a_m + \beta^a_m}\right). \tag{4.3.3}$$

When $s_m = 0$ the probability in (4.3.3) simplifies to the encounter probability of type m on source a as a magic bullet of that hype has already been discovered and therefore observing a type m tip is guaranteed to be a nuisance tip. While observing tips of already discovered types provide no useful information, they still lead to transitions on the knowledge state of the corresponding P_m^a's. However, this is only a formality as these knowledge states are no longer relevant.

Since the process transitions after every evaluated tip, given action a the distribution of the sojourn times $\mathcal{T}^a(\boldsymbol{s}, \boldsymbol{\beta})$ is simply the the distribution of the evaluation times of the tips

$$\mathbb{P}\left(\mathcal{T}^a(\boldsymbol{s}, \boldsymbol{\beta}) = t_m^a(\boldsymbol{s})\right) = q_m^a \tag{4.3.4}$$

Since this shows that the sojourn times do not depend on the information state we drop the dependence in the notation as well and will denote it as $\mathcal{T}^a(\boldsymbol{s})$. From its distribution, the expected sojourn time may also be obtained

$$\mathbb{E}\left[\mathcal{T}^a(\boldsymbol{s})\right] = \sum_{m=1}^{\mathcal{M}} q_m^a t_m^a(\boldsymbol{s}). \tag{4.3.5}$$

Let us define the cost structure of the decision process analogous to Section 4.2.1; with no lump sum cost or reward present, and so that the continuous cost rate of the process only depends on the physical state and is denoted as $w(\boldsymbol{s})$. Then the expected cost of action a in state $(\boldsymbol{s}, \boldsymbol{\beta})$ is

$$r(\boldsymbol{s}, \boldsymbol{\beta}, a) = w(\boldsymbol{s}) \sum_{m=1}^{\mathcal{M}} q_m^a t_m^a(\boldsymbol{s}). \tag{4.3.6}$$

Note that regardless of the information state, once the physical end state is reached $w(\mathbf{0}_{\mathcal{M}})$ and therefore $r(\mathbf{0}_{\mathcal{M}}, \boldsymbol{\beta}, a)$ is 0.

We define the objective function of the decision process to minimise the undiscounted total accumulated cost over a finite, but unknown number of decision epochs

$$\text{Obj: } \min_{\pi} \sum_{e} \mathbb{E}\left[r\left(\boldsymbol{s}_e, \boldsymbol{\beta}_e, \pi(\boldsymbol{s}_e)\right)\right], \tag{4.3.7}$$

where $\boldsymbol{s}_e$ and $\boldsymbol{\beta}_e$ are the physical and knowledge states at the e^{th} epoch and $\pi(\boldsymbol{s}_e, \boldsymbol{\beta}_e)$ the action prescribed by policy π.

4.3.3 Policies

In this section we develop policies to determine good actions to take in any given state of the intelligence puzzle with random conditional probabilities. We start by deriving a policy based on dynamic programming, which is followed by an assortment of heuristic decision rules based on the policies considered in Section 4.2.2.

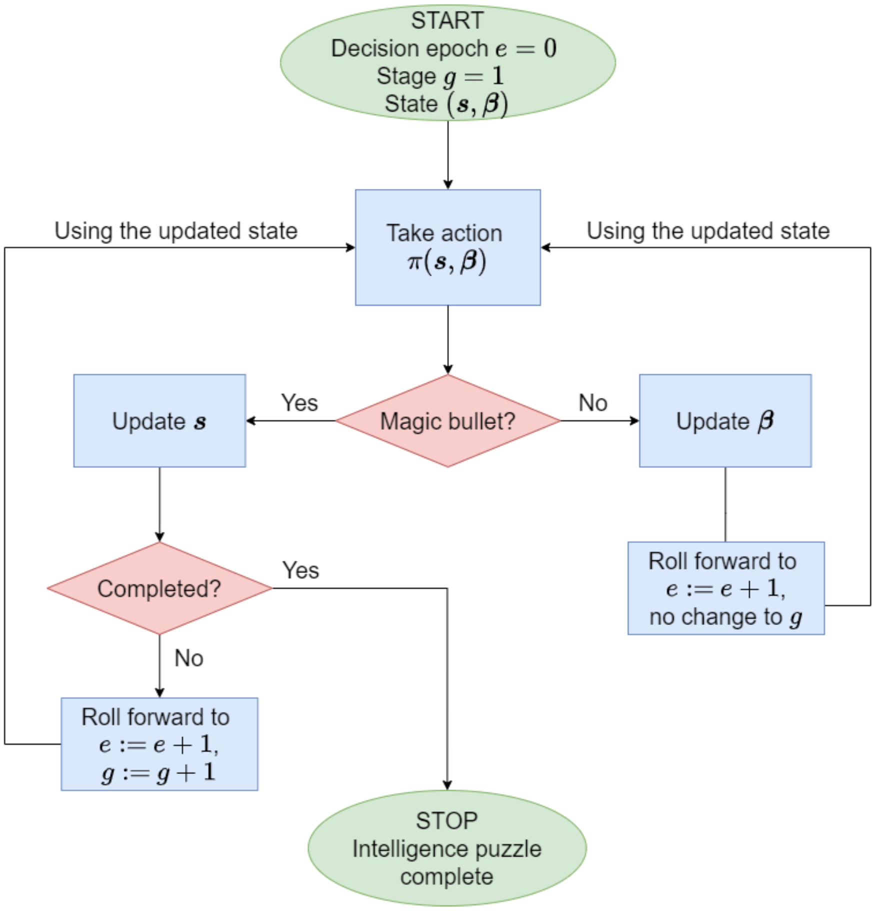


Figure 4.3.1: The process of applying policy $\pi(\boldsymbol{s}, \boldsymbol{\beta})$ to the intelligence puzzle with unknown conditional success probabilities. The updating process of physical state $\boldsymbol{s}$ has been described in detail in Section 4.1.2. The process of updating $\boldsymbol{\beta}$ takes place through updating the knowledge state of the source corresponding to the prescribed action, which is $\boldsymbol{\beta}^{\pi(\boldsymbol{s},\boldsymbol{\beta})}$ and is discussed in Section 4.3.1. The policy $\pi(\boldsymbol{s}, \boldsymbol{\beta})$ may be any policy in Section 4.3.3.

As in Section 4.2.2, we provide a flowchart (see Figure 4.3.1) that describes how these policies are applied to the intelligence puzzle with unknown conditional success probabilities and further illustrates how the decision process moves through the different states.

Dynamic Programming Approach

Following on from Section 4.3.2 we aim to recursively construct a policy for the intelligence puzzle with random conditional probabilities. Due to the infinite knowledge state space and lack of discounting, the decision process does not naturally lend itself to such an approach. To get around this problem we truncate the infinite recursion sequence by introducing artificial end-states to the knowledge state space by capping the values of the β_m^a's at an arbitrary $\beta_{\max}$.

Let us start by constructing the infinite recursion sequence. As stated in Section 4.3.2, the immediate expected cost incurred is the cost of taking action a which is $r(\boldsymbol{s}, \boldsymbol{\beta}, a)$, shown in (4.3.6). For each type m of information 2 types of transitions are possible. Either to state $(\boldsymbol{s}_{m^+}, \boldsymbol{\beta})$ with probability $\mathbb{P}(\boldsymbol{s}_{m^+}, \boldsymbol{\beta} \mid \boldsymbol{s}, \boldsymbol{\beta}, a)$ or to $(\boldsymbol{s}, \boldsymbol{\beta}^a_{m^-})$ with probability $\mathbb{P}(\boldsymbol{s}_{m^+}, \boldsymbol{\beta} \mid \boldsymbol{s}, \boldsymbol{\beta}, a)$. The value function of action a in state $(\boldsymbol{s}, \boldsymbol{\beta})$ is denoted as $\mathbb{V}^a(\boldsymbol{s}, \boldsymbol{\beta})$. The only natural end state of the decision process is when the physical end state $\boldsymbol{s} = \boldsymbol{0}_{\mathcal{M}}$ is reached. Since no costs are associated with such a state and the only permitted transitions lead to states where $\boldsymbol{s} = \boldsymbol{0}_{\mathcal{M}}$ is still true, the value function for any state $(\boldsymbol{0}_{\mathcal{M}}, \boldsymbol{\beta})$ is

$$\mathbb{V}^a(\boldsymbol{0}_{\mathcal{M}}, \boldsymbol{\beta}) = 0. \tag{4.3.8}$$

In any other state we write the value function as the sum of the immediate cost of action $r(\boldsymbol{s}, \boldsymbol{\beta}, a)$ and the expected future costs due to either type of transition, $\mathbb{P}(\boldsymbol{s}_{m^+}, \boldsymbol{\beta} \mid \boldsymbol{s}, \boldsymbol{\beta}, a)\mathbb{V}(\boldsymbol{s}_{m^+}, \boldsymbol{\beta})$ and $\mathbb{P}(\boldsymbol{s}, \boldsymbol{\beta}^a_{m^-} \mid \boldsymbol{s}, \boldsymbol{\beta}, a)\mathbb{V}(\boldsymbol{s}, \boldsymbol{\beta}^a_{m^-})$. Then the value of every state is recursively obtained via

$$\begin{aligned}
\mathbb{V}(\boldsymbol{s},\boldsymbol{\beta}) &= \min_{a\in\mathcal{S}_a}\{\mathbb{V}^a(\boldsymbol{s},\boldsymbol{\beta})\}\\
&= \min_{a\in\mathcal{S}_a}\Bigg\{r(\boldsymbol{s},\boldsymbol{\beta},a) + \sum_{m=1}^{\mathcal{M}}\mathbb{P}(\boldsymbol{s}_{m^+},\boldsymbol{\beta}\mid\boldsymbol{s},\boldsymbol{\beta},a)\mathbb{V}(\boldsymbol{s}_{m^+},\boldsymbol{\beta})\\
&+ \sum_{m=1}^{\mathcal{M}}\mathbb{P}(\boldsymbol{s},\boldsymbol{\beta}^a_{m^-}\mid\boldsymbol{s},\boldsymbol{\beta},a)\mathbb{V}(\boldsymbol{s},\boldsymbol{\beta}^a_{m^-})\Bigg\}\\
&= \min_{a\in\mathcal{S}_a}\Bigg\{w(\boldsymbol{s})\sum_{m=1}^{\mathcal{M}}q^a_m t^a_m(\boldsymbol{s}) + \sum_{m=1}^{\mathcal{M}}q^a_m\frac{\alpha^a_m s_m}{\alpha^a_m+\beta^a_m}\mathbb{V}(\boldsymbol{s}_{m^+},\boldsymbol{\beta})\\
&+ \sum_{m=1}^{\mathcal{M}}q^a_m\left(1-\frac{\alpha^a_m s_m}{\alpha^a_m+\beta^a_m}\right)\mathbb{V}(\boldsymbol{s},\boldsymbol{\beta}^a_{m^-})\Bigg\}. \qquad (4.3.9)
\end{aligned}$$

Since no limit exists on $\boldsymbol{\beta}$, (4.3.8) and (4.3.9) have no end-knowledge states from which the backwards recursion could be started. As stated earlier, to enable calculation of the recursion sequence we create such end states by truncating the knowledge state space, and capping all β^a_m at a maximum value, denoted $\beta_{\max}$. Under this scheme the posterior distribution of all P^a_m are only updated while $\beta^a_m < \beta_{\max}$, after that no further learning of the distribution takes place. While this makes the knowledge state no longer infinite, the knowledge state space may still be large dependent on $\mathcal{A}$, $\mathcal{M}$ and the $\beta_{\max}$ chosen. Assuming identical caps for all β^a_ms, the size of the knowledge state space is bounded by

$$\mid \mathcal{S}_{\boldsymbol{\beta}} \mid = (\beta_{\max} - \beta_0 + 1)^{\mathcal{AM}}\,. \qquad (4.3.10)$$

When a type m is encountered under action a for which the cap is in effect so that $\beta^a_m = \beta_{\max}$, the system can no longer transition to $(\boldsymbol{s},\boldsymbol{\beta}^a_{m^-})$. Instead, with probability $\mathbb{P}(\boldsymbol{s},\boldsymbol{\beta}^a_{m^-}\mid\boldsymbol{s},\boldsymbol{\beta},a)$ it self transitions to $(\boldsymbol{s},\boldsymbol{\beta})$. Let us introduce the set $\mu(\beta^a)$ defined as the set of types m for which $\beta^a_m = \beta_{\max}$

$$\mu(\boldsymbol{\beta}^a) = \{m;\beta^a_m = \beta_{\max}\}\,. \qquad (4.3.11)$$

By making use of $\mu(\boldsymbol{\beta}^a)$ the sum associated with the cost of knowledge state transitions can be split and the recursion can be rewritten as

$$\begin{aligned}
\mathbb{V}(\boldsymbol{s},\boldsymbol{\beta}) &= \min_{a\in\mathcal{S}_a}\Bigg\{ r(\boldsymbol{s},\boldsymbol{\beta},a) + \sum_{m=1}^{\mathcal{M}} \mathbb{P}(\boldsymbol{s}_{m^+},\boldsymbol{\beta} \mid \boldsymbol{s},\boldsymbol{\beta},a)\mathbb{V}(\boldsymbol{s}_{m^+},\boldsymbol{\beta}) \\
&\quad + \sum_{m\notin\mu(\boldsymbol{\beta}^a)} \mathbb{P}(\boldsymbol{s},\boldsymbol{\beta}^a_{m^-} \mid \boldsymbol{s},\boldsymbol{\beta},a)\mathbb{V}(\boldsymbol{s},\boldsymbol{\beta}^a_{m^-}) + \sum_{m\in\mu(\boldsymbol{\beta}^a)} \mathbb{P}(\boldsymbol{s},\boldsymbol{\beta}^a_{m^-} \mid \boldsymbol{s},\boldsymbol{\beta},a)\mathbb{V}(\boldsymbol{s},\boldsymbol{\beta}) \Bigg\} \\
&= \min_{a\in\mathcal{S}_a}\Bigg\{ w(\boldsymbol{s})\sum_{m=1}^{\mathcal{M}} q^a_m t^a_m(\boldsymbol{s}) + \sum_{m=1}^{\mathcal{M}} q^a_m \frac{\alpha^a_m s_m}{\alpha^a_m+\beta^a_m}\mathbb{V}(\boldsymbol{s}_{m^+},\boldsymbol{\beta}) \\
&\quad + \sum_{m\notin\mu(\boldsymbol{\beta}^a)} q^a_m\left(1-\frac{\alpha^a_m s_m}{\alpha^a_m+\beta^a_m}\right)\mathbb{V}(\boldsymbol{s},\boldsymbol{\beta}^a_{m^-}) + \sum_{m\in\mu(\boldsymbol{\beta}^a)} q^a_m\left(1-\frac{\alpha^a_m s_m}{\alpha^a_m+\beta^a_{\max}}\right)\mathbb{V}(\boldsymbol{s},\boldsymbol{\beta}) \Bigg\},
\end{aligned} \tag{4.3.12}$$

which can be rearranged to get

$$\mathbb{V}(\boldsymbol{s},\boldsymbol{\beta}) = \min_{a\in\mathcal{S}_a}\left\{ \frac{w(\boldsymbol{s})\displaystyle\sum_{m=1}^{\mathcal{M}} q^a_m t^a_m(\boldsymbol{s}) + \sum_{m=1}^{\mathcal{M}} q^a_m \frac{\alpha^a_m s_m}{\alpha^a_m+\beta^a_m}\mathbb{V}(\boldsymbol{s}_{m^+},\boldsymbol{\beta})}{\displaystyle\sum_{m\notin\mu(\boldsymbol{\beta}^a)} q^a_m + \sum_{m\in\mu(\boldsymbol{\beta}^a)} q^a_m\frac{\alpha^a_m s_m}{\alpha^a_m+\beta^a_{\max}}} + \frac{\displaystyle\sum_{m\notin\mu(\boldsymbol{\beta}^a)} q^a_m\left(1-\frac{\alpha^a_m s_m}{\alpha^a_m+\beta^a_m}\right)\mathbb{V}(\boldsymbol{s},\boldsymbol{\beta}^a_{m^-})}{\displaystyle\sum_{m\notin\mu(\boldsymbol{\beta}^a)} q^a_m + \sum_{m\in\mu(\boldsymbol{\beta}^a)} q^a_m\frac{\alpha^a_m s_m}{\alpha^a_m+\beta^a_{\max}}} \right\}. \tag{4.3.13}$$

The steps followed to get from (4.3.12) to (4.3.13) can be found in Appendix B.2.1.

Unfortunately, such a policy may not be practical to use. Even though truncating the knowledge state space made it finite, it is still large. Since $\mid \mathcal{S}_{\boldsymbol{s},\boldsymbol{\beta}} \mid = \mid \mathcal{S}_{\boldsymbol{s}} \mid \times \mid \mathcal{S}_{\boldsymbol{\beta}} \mid$, the size of the state space is

$$\mid \mathcal{S}_{\boldsymbol{s},\boldsymbol{\beta}} \mid = 2^{\mathcal{M}} \prod_{a=1}^{\mathcal{A}} \prod_{m=1}^{\mathcal{M}} \left(\beta_{\max} - \beta^{a\,(0)}_m\right).$$

which if all $\beta^{a\,(0)}_m = \beta^{(0)}$ simplifies to $2^{\mathcal{M}}(\beta_{\max}-\beta^{(0)})^{\mathcal{AM}}$. Another issue lies in the limited information available to learn from due to the reaction of the system to the different outcomes of the tips. This affects the usefulness of considering transitions through the knowledge states. As discussed earlier, the only occasion parameter α_m^a is updated is when source a finds a magic bullet of type m. However, at that point all interest in the distribution of P_m^a and all other $P_m^{a'\neq a}$ of the discovered type m is lost. This manifests in the knowledge state being characterised only by the collection of β_m^a's, which keep increasing. While it is only logical that the probability mass of P_m^a shifts to lower values with the observation of a failure, it has no opportunity to recover to higher values. Therefore with every action a taken that results in the evaluation of a type m tip the expectation of its success probability

$$q_m^a \mathbb{E}\left[P_m^a(\boldsymbol{s}, \alpha_m^a, \beta_m^a)\right] = q_m^a \frac{\alpha_m^a s_m}{\alpha_m^a + \beta_m^a}$$

decreases until it is set to zero upon discovery of the magic bullet, either through the increase in β_m^a or by s_m changing to 0. Similarly, the expected probability that action a results in the discovery of any magic bullet

$$\sum_{m=1}^{\mathcal{M}} \mathbb{P}(\boldsymbol{s}_{m^+}, \boldsymbol{\beta} \mid \boldsymbol{s}, \boldsymbol{\beta}, a) = \sum_{m=1}^{\mathcal{M}} q_m^a \frac{\alpha_m^a s_m}{\alpha_m^a + \beta_m^a}$$

is also decreasing every time action a is taken. From the above we can deduce that $\mathbb{V}(\boldsymbol{s}, \boldsymbol{\beta})$ is an increasing function of $\boldsymbol{\beta}$ and the outlook of the decision maker worsens with every decision epoch. The truncated recursion suffers from the same problem. Since in the knowledge end state all $\beta_m^a = \beta_{\max}$, the larger its permitted value, the smaller the minimum permitted value of $\mathbb{E}\left[P_m^a(\boldsymbol{s}, \alpha_m^a, \beta_m^a)\right]$, which is encountered at the end state becomes. As $\beta_{\max} \to \infty$ the expected success probability $\mathbb{E}\left[P_m^a(\boldsymbol{s}, \alpha_m^a, \beta_{\max})\right] \to 0$ for all a and m and the expected completion time and cost of the intelligence puzzle diverges to ∞. Similarly, the larger β is the higher the expected completion cost of the puzzle is. Furthermore, with every non-magic-bullet observed, the expected probability of observing a magic bullet in the following decision epoch reduces. To avoid this ever worsening outlook regarding the future expectation of P_m^a's, the policies constructed in the rest of the section do not consider how the knowledge state may change as a consequence of the action taken.

There is another layer of difficulty to this version of the intelligence puzzle. While the decision maker can take any of the $a \in \mathcal{S}_a$ actions, they have no influence over which type of tip may be encountered, only their knowledge of the $\boldsymbol{q}^a$'s. Since types with larger q_m^a's are encountered more often, a large variation in how many samples are used to estimate the distribution of the P_m^a's may be present.

Despite these challenges, in the rest of this section we present three policies which disregard state transitions with regards to the knowledge states and are analogous the policies in Section 4.2.2. Another method which is later used to obtain a lower bound on the completion cost of the puzzle is discussed at the end.

Expected Completion Cost Policy (ECC)

As we have seen in Section 4.3.3, policies that consider the evolution of the knowledge state of the system are not suitable. However, for a fixed knowledge state the value of the physical state can be recursively calculated.

The value of the physical end state is always 0 independent of the knowledge state. Assuming that the knowledge state is constant is equivalent to setting a non uniform $\beta_{\max}$, positing that the knowledge state space is truncated for all a and m at $\beta_{\max} = \beta_m^a$, the state currently occupied by the process. Therefore the value function of the physical state $\boldsymbol{s}$ conditional on the knowledge state remaining at $\boldsymbol{\beta}$ throughout is $\mathbb{V}(\boldsymbol{s} \mid \boldsymbol{\beta}) = \mathbb{V}(\boldsymbol{s}, \boldsymbol{\beta}_{\max} = \boldsymbol{\beta})$. Under this scenario the recursion in (4.3.13) simplifies to

$$\begin{aligned}\mathbb{V}(\boldsymbol{s} \mid \boldsymbol{\beta}) &= \min_{a \in \mathcal{S}_a} \{\mathbb{V}^a(\boldsymbol{s} \mid \boldsymbol{\beta})\} \\ &= \min_{a \in \mathcal{S}_a} \left\{ w(\boldsymbol{s}) \frac{\sum\limits_{m=1}^{\mathcal{M}} q_m^a t_m^a(\boldsymbol{s})}{\sum\limits_{m=1}^{\mathcal{M}} q_m^a \dfrac{\alpha_m^a s_m}{\alpha_m^a + \beta_m^a}} + \sum_{m=1}^{\mathcal{M}} \left(\frac{q_m^a \dfrac{\alpha_m^a s_m}{\alpha_m^a + \beta_m^a}}{\sum\limits_{m'=1}^{\mathcal{M}} q_{m'}^a \dfrac{\alpha_{m'}^a s_{m'}}{\alpha_{m'}^a + \beta_{m'}^a}} \mathbb{V}\left(\boldsymbol{s}_{m^+} \mid \boldsymbol{\beta}\right) \right) \right\},\end{aligned} \tag{4.3.14}$$

which is analogous to the recursion equation of the basic intelligence puzzle found in Section 4.2.2. The first term of the sum is the expected cost accrued during the current stage, while the second is the sum of the expected cost accrued in future stages of the intelligence puzzle. In effect, the point estimates of the $P_m^a(\boldsymbol{s}, \boldsymbol{\alpha}, \boldsymbol{\beta})$'s fulfil the same role as the $p_m^a(\boldsymbol{s})$'s did in (4.2.7), making the recursion in (4.3.14) an approximation based on deterministic equivalence. Therefore, just as how the value function $\mathbb{V}(\boldsymbol{s})$ in Section 4.2.2 was a measure of the expected completion cost of the basic intelligence puzzle in $\boldsymbol{s}$, so is $\mathbb{V}(\boldsymbol{s} \mid \boldsymbol{\beta})$ an estimate of the expected completion cost in $\boldsymbol{s}$ based on the current knowledge state $\boldsymbol{\beta}$. More precisely, $\mathbb{V}(\boldsymbol{s} \mid \boldsymbol{\beta})$ is the approximate expected completion cost of the intelligence puzzle based on deterministic equivalence.

For every state $(\boldsymbol{s}, \boldsymbol{\beta})$, in which $\boldsymbol{\beta}$ is the most up-to-date knowledge state based on all previous observations, we define the expected completion cost policy (ECC) to prescribe the action that minimises the expected completion cost, recursively calculated using (4.3.14) from the end state till the current state. Which means that under the ECC in every state $(\boldsymbol{s}, \boldsymbol{\beta})$ action a^* is taken so that

$$a^* = \pi^{\mathrm{ECC}}(\boldsymbol{s}, \boldsymbol{\beta}) := \underset{a \in \mathcal{S}_a}{\arg\min} \left\{ \mathbb{V}^a(\boldsymbol{s} \mid \boldsymbol{\beta}) \right\}. \tag{4.3.15}$$

We must emphasise that in the regime of random P_m^a's the knowledge state is not fixed and transitions with respect to it occur. Therefore the recursion has to be re-evaluated from its end state $(\mathbf{0}_{\mathcal{M}}, \boldsymbol{\beta})$ to its current state for every knowledge state $\boldsymbol{\beta}$ the decision process visits. This makes the ECC a computationally complex policy, requiring repeated calculations of a recursive cost sequence.

The application of the ECC policy can be further clarified with the help of Figure 4.3.1. At every decision epoch e a source is selected to sample from as prescribed by (4.3.15), choosing the source with the lowest estimated expected completion cost. If it yields a magic bullet, the physical state is updated and the knowledge state remains unchanged. If this completes the puzzle, the decision process ends, otherwise we roll forward to the next decision epoch $e + 1$ and enter the next stage $g + 1$ of the problem. If the action taken in decision epoch e does not yield a magic bullet, the physical state stays the same, but the knowledge state is updated. Since no magic bullet was discovered we remain in the same stage g of the puzzle, but regardless of the stage we roll forward to the next decision epoch $e+1$. In decision epoch $e+1$ when the next action to take is being determined, the expected completion costs are calculated using the updated state, regardless of which aspect of the state was updated following decision epoch e.

Let us mention on the side another measure which could alternatively be used for the expected completion time; one that uses not only the point estimate of the P_m^a's but their full distribution when computing the expected completion time. If this alternative expected completion time is denoted by $\mathbb{V}'(\boldsymbol{s} \mid \boldsymbol{\beta})$, the alternate recursion takes the form

$$\begin{aligned}
\mathbb{V}'(\mathbf{0}_{\mathcal{M}} \mid \boldsymbol{\beta}) &= 0, \\
\mathbb{V}'(\boldsymbol{s} \mid \boldsymbol{\beta}) &= \min_{a \in \mathcal{S}_a} \{\mathbb{V}'^a(\boldsymbol{s} \mid \boldsymbol{\beta})\} \\
&= \min_{a \in \mathcal{S}_a} \left\{ w(\boldsymbol{s}) \mathbb{E}\left[\frac{\sum_{m=1}^{\mathcal{M}} q_m^a t_m^a(\boldsymbol{s})}{\sum_{m=1}^{\mathcal{M}} q_m^a P_m^a(\boldsymbol{s}, \alpha_m^a, \beta_m^a)} \right] \right. \\
&\qquad \left. + \sum_{m=1}^{\mathcal{M}} \mathbb{E}\left[\frac{q_m^a P_m^a(\boldsymbol{s}, \alpha_m^a, \beta_m^a)}{\sum_{m'=1}^{\mathcal{M}} q_{m'}^a P_m^a(\boldsymbol{s}, \alpha_m^a, \beta_m^a)} \right] \mathbb{V}'\left(\boldsymbol{s}_{m^+} \mid \boldsymbol{\beta}\right) \right\},
\end{aligned}$$

where the expectations are taken with respect to the posterior distributions of the P_m^a's. Generally, those expectations are only obtainable numerically, which would further increase the computational effort required. For that reason this variant of the ECC will not further be considered.

However, even without numerical integration the ECC is a computationally intensive policy which may be slow for even moderate sized problems where $\mathcal{M} \gtrsim 5$. We refer the reader back to Table 4.2.1 and Table 4.2.2 for an indication of the time required to obtain the prescribed action at any one decision time, as the dynamic program we are required to solve here is in essence the same as that in Equation 4.2.7. By restricting the number of subsequent physical state transitions to consider in the recursion, its complexity can be reduced.

Myopic policy

The most drastic simplification of the ECC is to not consider any state transition at all, and construct a policy aiming to minimise the cost accrued during the current stage of the puzzle given its current state $(\boldsymbol{s}, \boldsymbol{\beta})$. We refer to such a policy as myopic since it is short-sighted both in terms of the physical and the knowledge states the intelligence puzzle may transition to as a consequence of its recommended action. As described in Section 4.3.3, the expected duration of the current stage is given by the first term of the sum in (4.3.14), and therefore the myopic policy prescribes action a^* so that

$$\begin{aligned} a^* = \pi^{\text{MYO}}(\boldsymbol{s}, \boldsymbol{\beta}) :&= \underset{a \in \mathcal{S}_a}{\arg\min} \left\{ w(\boldsymbol{s}) \frac{\sum_{m=1}^{\mathcal{M}} q_m^a t_m^a(\boldsymbol{s})}{\sum_{m=1}^{\mathcal{M}} q_m^a \frac{\alpha_m^a s_m}{\alpha_m^a + \beta_m^a}} \right\} \\ &= \underset{a \in \mathcal{S}_a}{\arg\min} \left\{ \frac{\sum_{m=1}^{\mathcal{M}} q_m^a t_m^a(\boldsymbol{s})}{\sum_{m=1}^{\mathcal{M}} q_m^a \frac{\alpha_m^a s_m}{\alpha_m^a + \beta_m^a}} \right\}. \end{aligned} \tag{4.3.16}$$

Observe that due to the cost rate only depending on the physical state, choosing the action which minimises the expected cost accrued during the current stage is equivalent to picking the action which minimises the duration of the stage. This has also been observed in the myopic policy of the basic intelligence puzzle.

Note that the action prescribed by the myopic policy is re-evaluated at every decision epoch in accordance with Figure 4.3.1.

Similarly to how an alternative measure of the expected completion cost was available based on the posterior distribution instead of the point-estimate of P_m^a, there is one for the expected duration of the current stage, written as

$$\mathbb{E}\left[\frac{\sum_{m=1}^{\mathcal{M}} q_m^a t_m^a(\boldsymbol{s})}{\sum_{m=1}^{\mathcal{M}} q_m^a P_m^a(\boldsymbol{s}, \alpha_m^a, \beta_m^a)}\right] = \int_0^1 \cdots \int_0^1 \frac{\sum_{m=1}^{\mathcal{M}} q_m^a t_m^a(\boldsymbol{s})}{\sum_{m=1}^{\mathcal{M}} q_m^a P_m^a(\boldsymbol{s}, \alpha_m^a, \beta_m^a)} \prod_{m=1}^{\mathcal{M}} \pi(P_m^a) \mathrm{d}P_1^a \ldots \mathrm{d}P_{\mathcal{M}}^a, \tag{4.3.17}$$

where $\pi(P_m^a)$ is the Beta posterior distribution of P_m^a with parameters α_m^a and β_m^a. Since the denominator contains a weighted sum of the P_m^a's, the integral can only be evaluated analytically if exactly one type m^* of magic bullet still awaits discovery. Then (4.3.17) simplifies to

$$\mathbb{E}\left[\frac{\sum_{m=1}^{\mathcal{M}} q_m^a t_m^a(\boldsymbol{s})}{\sum_{m=1}^{\mathcal{M}} q_m^a P_m^a(\boldsymbol{s}, \alpha_m^a, \beta_m^a)}\right] = \frac{\alpha_{m^*}^a + \beta_{m^*}^a - 1}{q_{m^*}^a(\alpha_{m^*}^a - 1)} \sum_{m=1}^{\mathcal{M}} q_m^a t_m^a(\boldsymbol{s}),$$

otherwise (4.3.17) can only be evaluated numerically. The integral-myopic policy may be defined as follows; at every decision time source a^* is continued, where

$$a^* := \arg\min_{a \in \mathcal{S}_a} \left\{ \mathbb{E}\left[\frac{\sum_{m=1}^{\mathcal{M}} q_m^a t_m^a(\boldsymbol{s})}{\sum_{m=1}^{\mathcal{M}} q_m^a P_m^a(\boldsymbol{s}, \alpha_m^a, \beta_m^a)}\right] \right\}. \tag{4.3.18}$$

Unfortunately as $\mathcal{M}$ increases so do the dimensions of the integral and the computational effort required to obtain it. Furthermore, the requirement that all $\alpha_m^a > 1$ is limiting, and so this version of the myopic policy is considered no further.

Look-ahead policy

A compromise between the complexity of the ECC and the drastic simplification of the myopic policy is a one-stage look-ahead policy. We construct such a policy by adapting the expected completion cost in (4.3.14) to consider only the expected cost of completing the current stage and the one immediately after, similarly to how the one-step look-ahead policy of the basic intelligence puzzle was adapted from (4.2.7). In doing so, the same assumptions were made during the calculations as in Section 4.3.3, that is the knowledge state is fixed and is only used to provide the posterior point estimates for P_m^a.

Then the one-stage look-ahead policy for the intelligence puzzle with unknown conditional success probabilities takes action a^* so that

$$\begin{aligned} a^* &= \pi^{\mathrm{LA}}(\boldsymbol{s}, \boldsymbol{\beta}) \\ &:= \min_{a \in \mathcal{S}_a} \left\{ w(\boldsymbol{s}) \frac{\sum_{m=1}^{\mathcal{M}} q_m^a t_m^a(\boldsymbol{s})}{\sum_{m=1}^{\mathcal{M}} q_m^a \frac{\alpha_m^a s_m}{\alpha_m^a + \beta_m^a}} \right. \\ &\left. + \sum_{m=1}^{\mathcal{M}} \left(\frac{q_m^a \frac{\alpha_m^a s_m}{\alpha_m^a + \beta_m^a}}{\sum_{m'=1}^{\mathcal{M}} q_{m'}^a \frac{\alpha_{m'}^a s_{m'}}{\alpha_{m'}^a + \beta_{m'}^a}} \min_{a' \in \mathcal{S}_a} \left\{ w(\boldsymbol{s}_{m^+}) \frac{\sum_{m=1}^{\mathcal{M}} q_m^{a'} t_m^{a'}(\boldsymbol{s}_{m^+})}{\sum_{m=1}^{\mathcal{M}} q_m^{a'} \frac{\alpha_m^{a'} s_{m^+,m}}{\alpha_m^{a'} + \beta_m^{a'}}} \right\} \right) \right\}. \end{aligned} \tag{4.3.19}$$

Observe that just as the ECC policy used a deterministically equivalent approximation of the expected completion cost, the one-stage look-ahead cost here is a deterministically equivalent approximation of the one-stage look-ahead cost. Based on this, parallels may be drawn between the one-stage look ahead policy here and the one-step look-ahead policy (4.2.33) of the basic intelligence puzzle. It is important to note that while in the case of the basic intelligence puzzle "one-step" refers to one stage, one stage in the case of the intelligence puzzle with unknown P_m^a's and the look-ahead policy currently under discussion would include many decision epochs. For the sake of calculating the expected costs in (4.3.19) the knowledge state is assumed to remain the same for all decision epochs, but the policy is applied in a way that acknowledges that is not the case. Again, we refer back to Figure 4.3.1. At every decision epoch e (4.3.19) is used to determine the action to take given the current state $(\boldsymbol{s}_e, \boldsymbol{\beta}_e)$ of the intelligence puzzle, which transitions to its next state $(\boldsymbol{s}_{e+1}, \boldsymbol{\beta}_{e+1})$ based on the outcome of the action taken. At the next decision epoch $e+1$, the action to take is again determined by (4.3.19), but it will use the state at decision epoch $e+1$ $(\boldsymbol{s}_{e+1}, \boldsymbol{\beta}_{e+1})$.

We finish discussion of the one-stage look-ahead policy by noting that for $\mathcal{M} = 2$ the one-step look-ahead policy is identical to the ECC policy, as the decision process consists only of two stages.

Super-optimal policy

The super-optimal policy is not intended as realistic policy a decision maker may adhere to, but rather as a method of acquiring a lower bound on the expected cost of completing the intelligence puzzle. In its formulation it is identical to the optimal policy found via dynamic programming in Section 4.2.2 and ignores the need to learn about the conditional success probabilities. It treats the unknown parameter values as if they were known, and therefore requires an omniscient decision maker.

The super-optimal policy is then applied as follows. For every realisation of the intelligence puzzle, the decision-maker is in possession of perfect information regarding the underlying values of the conditional success probabilities. In that case all parameter vectors are known quantities to them. When all parameter vectors are known, the intelligence puzzle is as described in Section 4.2, for which an optimal policy can be obtained using the recursion in (4.2.7). As the P_m^a's are treated like known quantities, information states are unnecessary and the optimal action only depends on the physical state of the puzzle. Based on the above, the value of the super-optimal policy is then obtained by repeatedly applying it to independently sampled instances of the puzzle and averaging over the observed completion costs.

Since the super-optimal policy has access to information which is not otherwise available, it is able to achieve lower completion costs than any other policy that has no access to additional information. Furthermore, since the super-optimal policy is given by the optimal policy of the problem with additional information, no other policy can achieve lower completion cost. For this property and its relative ease of computation the super-optimal policy is used to provide a lower bound on the Bayes value of the completion cost.

4.3.4 Numerical Experiments

The performance of the expected completion cost, myopic and one-step look-ahead policies described in Section 4.3.3 is evaluated numerically, using simulated instances of the intelligence puzzle. The same instances of the intelligence puzzle are also used to obtain the value of the super-optimal policy. We examine them using the same measures as in Section 4.2.3 to which this study is set up similarly.

In this numerical study we set the cost rate of every physical state (apart from $\mathbf{0}_{\mathcal{M}}$) to $w(\boldsymbol{s}) = 1$, so that the objective of minimising the expected cost of completion is equivalent to minimising the expected time till completion of the intelligence puzzle. To keep the problem simple all evaluation times were set to be equal so that $t_m^a = 1$ for all m and a. Like in the simulation study of the basic intelligence puzzle, the encounter probability vectors are sampled from a Dirichlet$(\mathbf{1}_{\mathcal{M}})$ distribution. To evaluate the Bayes value of the expected completion cost of the puzzle, the underlying value of the P_m^a's, which we use in generating the data and in applying the super-optimal policy, is sampled from its prior distribution. We have opted to use the same, vague conjugate Beta(1,1) prior distribution for all P_m^a. Notice that the methods of obtaining the underlying parameters of the intelligence puzzle here match those in Section 4.2.3. Given that, we made concious effort to ensure that the dataset used in this numerical study is a subset of the one used in Section 4.2.3. This allows us to compare the two studies and identify the effects having to learn P_m^a has on the outcome of puzzle.

We examine the intelligence puzzle for $\mathcal{A} = 2$ and $\mathcal{A} = 5$, each with $\mathcal{M}$ ranging between 2 and 6. For every such combination 3 values of c, the factor that controls the time dependence of the $t_m^a(\boldsymbol{s})$ as per (4.1.8), is considered. These are $c = \{1, 5, \infty\}$. At the two extremes $c = 1$ corresponds to no state dependence of evaluation times while $c = \infty$ results in no time spent evaluating types for which $s_m = 0$. For every combination of $\mathcal{A}$, $\mathcal{M}$ and c the dataset contains 10^4 intelligence puzzles defined by the combination of the randomly sampled $\boldsymbol{q}^a$'s and realisations of P_m^a. Each such puzzle is simulated 10 times and averaged over. We have been limited to such a low number of replications by the available computational capacity.

On the figures found in this section abbreviations are used to identify the policies; ECC-P stands for the expected completion cost, LA-P the one step look-ahead and MYO-P the myopic policy. The suffix ”P” is to signal that these are the versions of the policy in which the distribution of P_m^a are initially unknown.

First we compare the policies in terms of the mean regret of a policy π, which we denote as $\overline{\mathcal{R}}^\pi$ and is obtained it as follows. We start by calculating the regret of said policy, denoted as $\mathcal{R}^\pi$, for every realisation of the intelligence puzzle via

$$\mathcal{R}^\pi = \frac{\mathcal{C}^\pi - \mathcal{C}^{\pi^{\text{SO}}}}{\mathcal{C}^{\pi^{\text{SO}}}}, \tag{4.3.20}$$

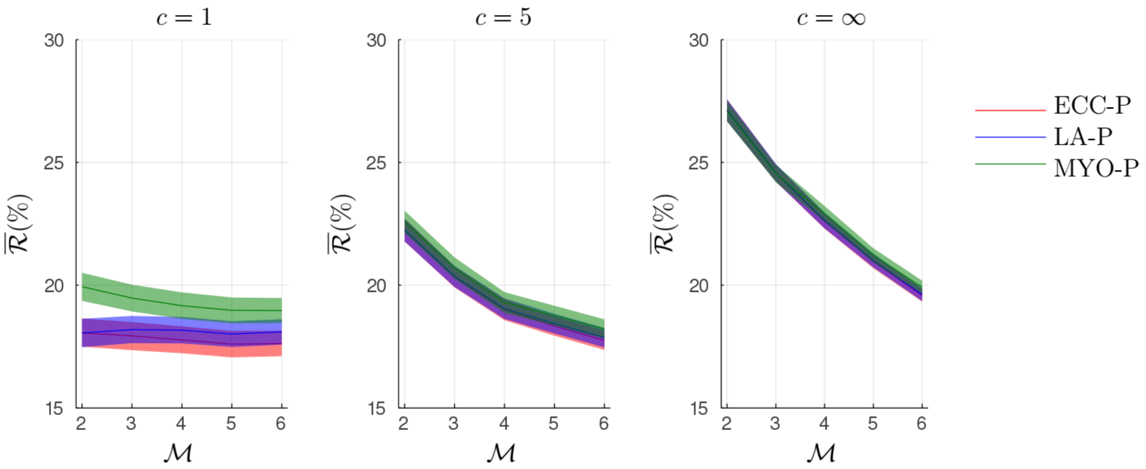


Figure 4.3.2: Mean relative regret $\overline{\mathcal{R}}$ for $\mathcal{A} = 2$ as a function of $\mathcal{M}$ with unit evaluation times.

which is the relative difference between the cost of completion $\mathcal{C}$ of the intelligence puzzle using the super-optimal policy π^{SO} and of policy π. These individual regret values are averaged over to provide the mean regret $\overline{\mathcal{R}}^{\pi}$. We use the super-optimal policy as our benchmark as the optimal policy is not known to us. Note the strong resemblance this measure has to the relative sub-optimality used in Section 4.2.3. That, along with the common dataset allows for direct comparison between numerical studies to examine the effect of reduced information. Figure 4.3.2 and Figure 4.3.3 show the mean regret as a function of $\mathcal{M}$ for $\mathcal{A} = 2$ and $\mathcal{A} = 5$ respectively.

One may notice that mean regrets are quite large for all $\mathcal{A}$, $\mathcal{M}$ and c. This is partly due to how little information the priors used provide. To reassure ourselves and the reader the numerical experiments were repeated with more informative Beta(2,2) and Beta(5,5) priors and figures equivalent to Figure 4.3.2 and Figure 4.3.3 were produced. While these figures have been consigned to Appendix B.3, they show that the mean regret of all policies is reduced with more informative priors in place.

As expected, the mean regret of puzzles with $\mathcal{A} = 5$ are generally much larger than those with $\mathcal{A} = 2$ as in the $\mathcal{A} = 5$ case fewer observations per source are made generally.

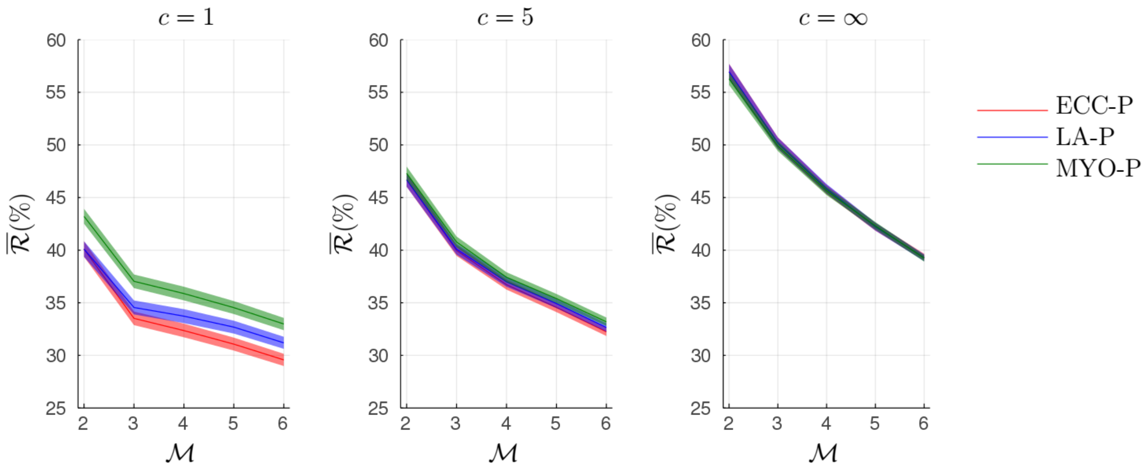


Figure 4.3.3: Mean relative regret $\overline{\mathcal{R}}$ for $\mathcal{A} = 5$ as a function of $\mathcal{M}$ with unit evaluation times.

More available sources to choose from increase the chance of having good combinations q^a_m and $P^a_m = p^a_m$, reducing the average number of observations required to complete the puzzle. This reduced number of observations is then split among 5 sources instead of 2. Not surprisingly the regrets associated with both of these cases are well in excess of the mean regret seen with the basic intelligence puzzle.

The overarching trend of decreasing mean regret as $\mathcal{M}$ is increased can be explained by the increased number of observations required to complete the intelligence puzzle. As discussed in Section 4.3.3, only learning through observed failures limits the decision makers ability to learn the distribution of P^a_m. Due to the inefficient nature of the learning process, having the outcome of even a few additional observations makes a noticeable impact on the mean regret. That is, until the limit of what can be learnt from observing failures only is reached. This conjecture is supported by the decreasing rate of improvement seen for all values of c for both $\mathcal{A} = 2$ and $\mathcal{A} = 5$.

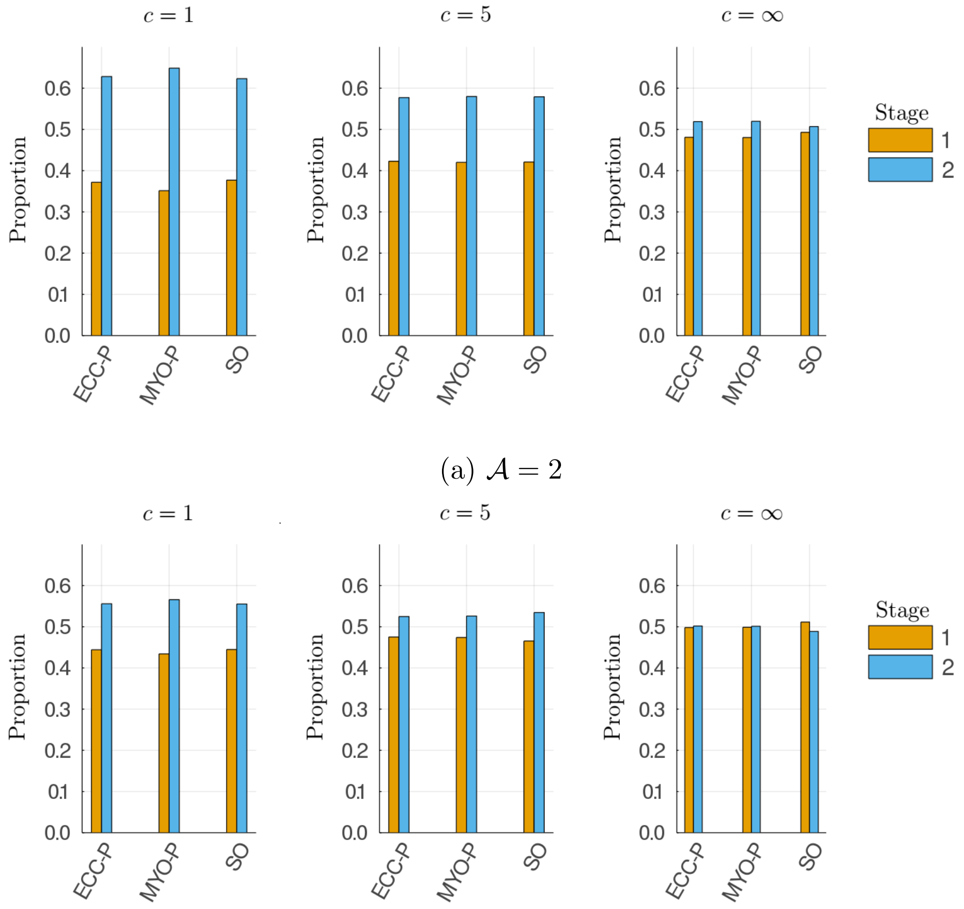
In the case of $c = 1$ the expected order in terms of the performance of the policies is clearly visible; while at $\mathcal{M} = 2$ the ECC and the look ahead policy are identical, for all other sizes of puzzle the ECC policy achieves the lowest regret, followed by the look-ahead and myopic policies. However the distinction between the policies decreases as c is increased, becoming near imperceptible for $c = \infty$. Any differences in the performances of the policies are dominated by common behaviours that can be explained by the need to learn about the P_m^a's and how limited the ability to learn about these parameters is. Both the increase in regret with c and the near-identical performance of the policies can be attributed to the fact that neither of the policies can take effective advantage of the reduction in evaluation times if they don't possess reliable estimates of the conditional probabilities. At the same time the super-optimal policy is free to exploit this feature of the puzzle, reducing its average completion cost and therefore increasing the mean regret of the other policies.

Figure 4.3.4 and Figure 4.3.5 show how the total completion time is distributed among the stages of the decision process. Broadly speaking very similar observations may be made as for the basic intelligence puzzle. A notable difference is that while the proportion of time spent on later stages decreases as c is increased, they still took up a larger proportion of the total completion time that earlier stages. The only exception to this is the super-optimal policy in Figure 4.3.5b. Spending less time on already discovered types of tips reduces the proportion of time spent on later stages, but without accurate posterior estimates of the success probabilities the effect is not strong enough to change the trend that later stages contribute more to the completion cost.

The regret measure used for the completion cost of the puzzle can be adapted to examine individual stages. We call this the stage-wise mean regret, and is calculated on a stage by stage basis. The regret of a puzzle in any stage g is given by

$$\mathcal{R}_g^\pi = \frac{\mathcal{C}_g^\pi - \mathcal{C}_g^{\pi^{\mathrm{SO}}}}{\mathcal{C}_g^{\pi^{\mathrm{SO}}}}, \tag{4.3.21}$$

which is averaged over all randomly generated puzzles to get the stage-wise mean regret $\overline{\mathcal{R}}_g^\pi$ of policy π. It compares the cost accrued during stage g by policy π to the cost accrued during that same stage by the super-optimal policy. Note an important difference from the mean regret; the stage-wise mean regret can take on a negative value if a policy

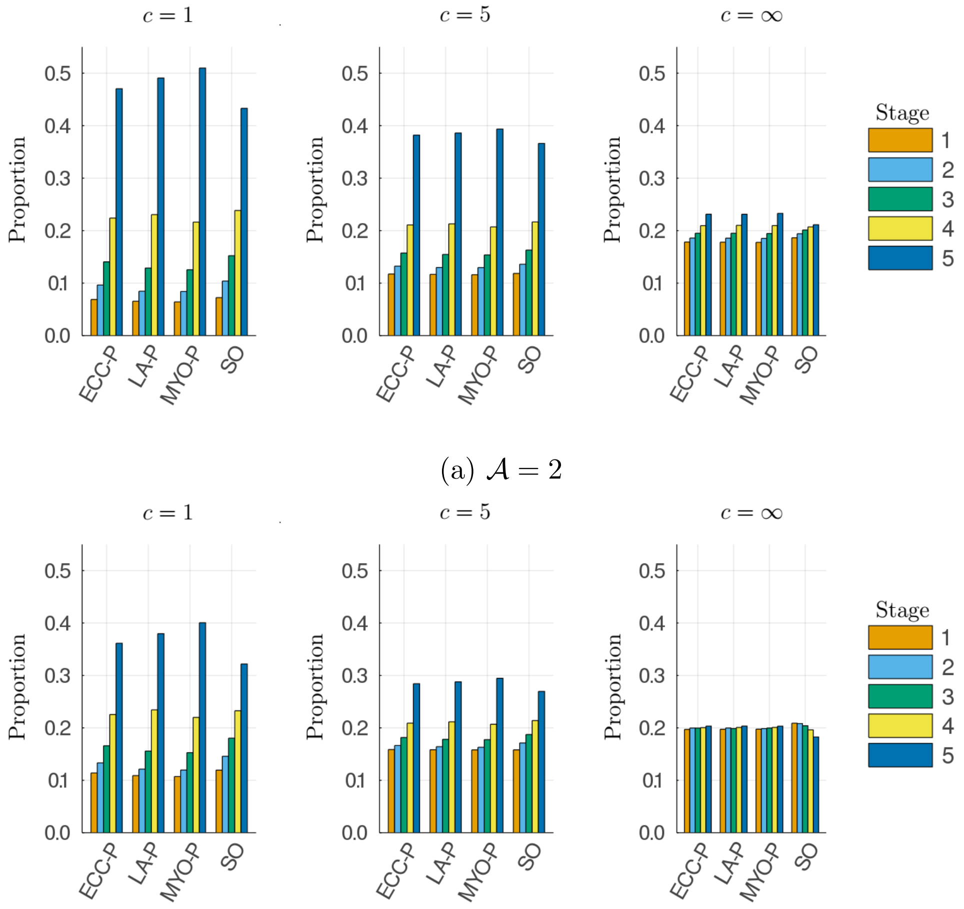


(a) $\mathcal{A} = 2$

(b) $\mathcal{A} = 5$

Figure 4.3.4: Proportion of completion time spent on each stage for $\mathcal{M} = 2$. The look-ahead policy is not represented as it is identical to the ECC for $\mathcal{M} = 2$. The one-stage look-ahead policy does not feature in this figure as it is equivalent to the ECC policy when $\mathcal{M} = 2$.

completes a stage with less cost accrued than the super-optimal policy. Figure 4.3.6 and Figure 4.3.7 show the stage-wise mean regret for $\mathcal{M} = 2$ and $\mathcal{M} = 5$ respectively, each with both $\mathcal{A} = 2$ and $\mathcal{A} = 5$. Just as the mean regret of every policy was higher than those based on the basic intelligence puzzle, so are the stage-wise regrets. For all but the $c = 1$ case of the puzzle with $\mathcal{A} = 2$ and $\mathcal{M} = 5$ all stage-wise regrets are positive, which means that all stage durations were longer than the stage durations of the super-optimal policy. While by itself this is not a surprising result, note that the



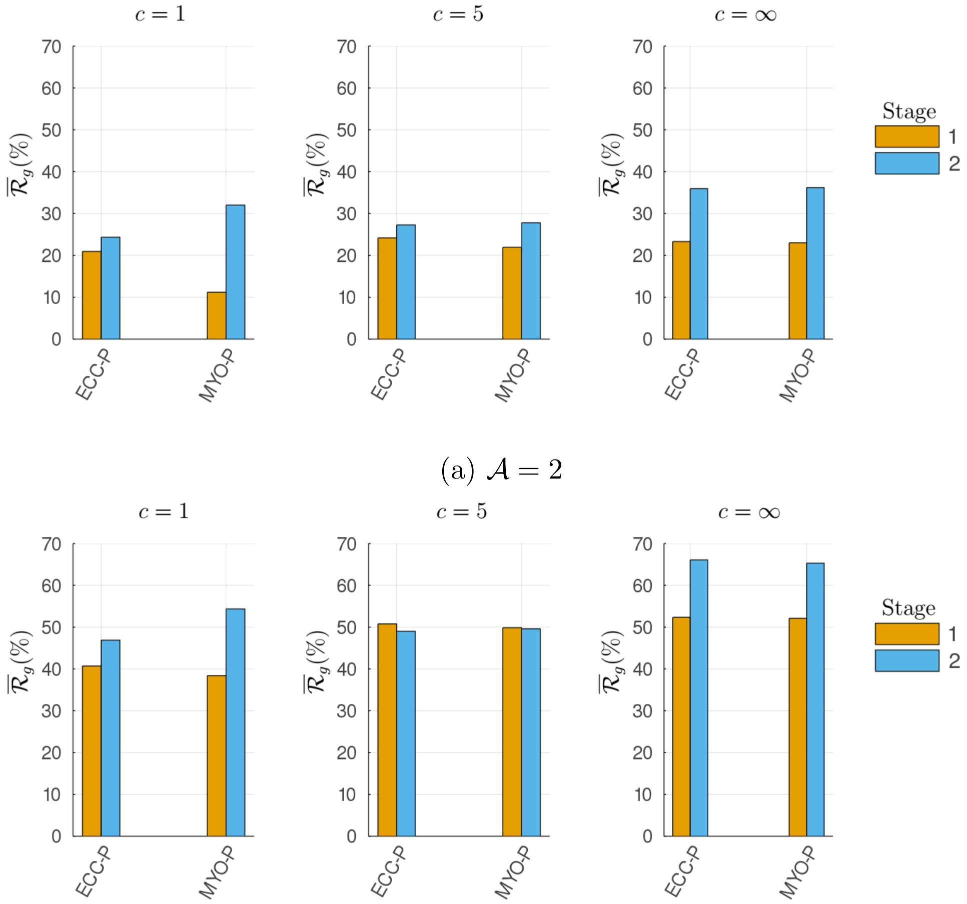
(a) $\mathcal{A} = 2$

(b) $\mathcal{A} = 5$

Figure 4.3.5: Proportion of completion time spent on each stage for $\mathcal{M} = 5$.

myopic and look-ahead policies developed for the basic intelligence puzzle did achieve negative stage-wise mean regrets for the early stages of the puzzle.

The stage-wise regrets of the myopic and the look-ahead policy for $\mathcal{M} = 5$ for all but the $c = \infty$ case initially decrease then increase so that $\overline{\mathcal{R}}_1 > \overline{\mathcal{R}}_2$ but $\overline{\mathcal{R}}_g < \overline{\mathcal{R}}_{g+1}$ from $g = 2$ onwards. Both policies are short sighted to a degree, focusing only on minimising the costs from the current and in the case of the look ahead policy the subsequent stage. This is difficult to do when the P^a_m's are estimated based on a small number of observations, leading to a high regret for stage $g = 1$. During $g = 2$ estimates are based on more observations and the decisions taken are more informed, which reduces



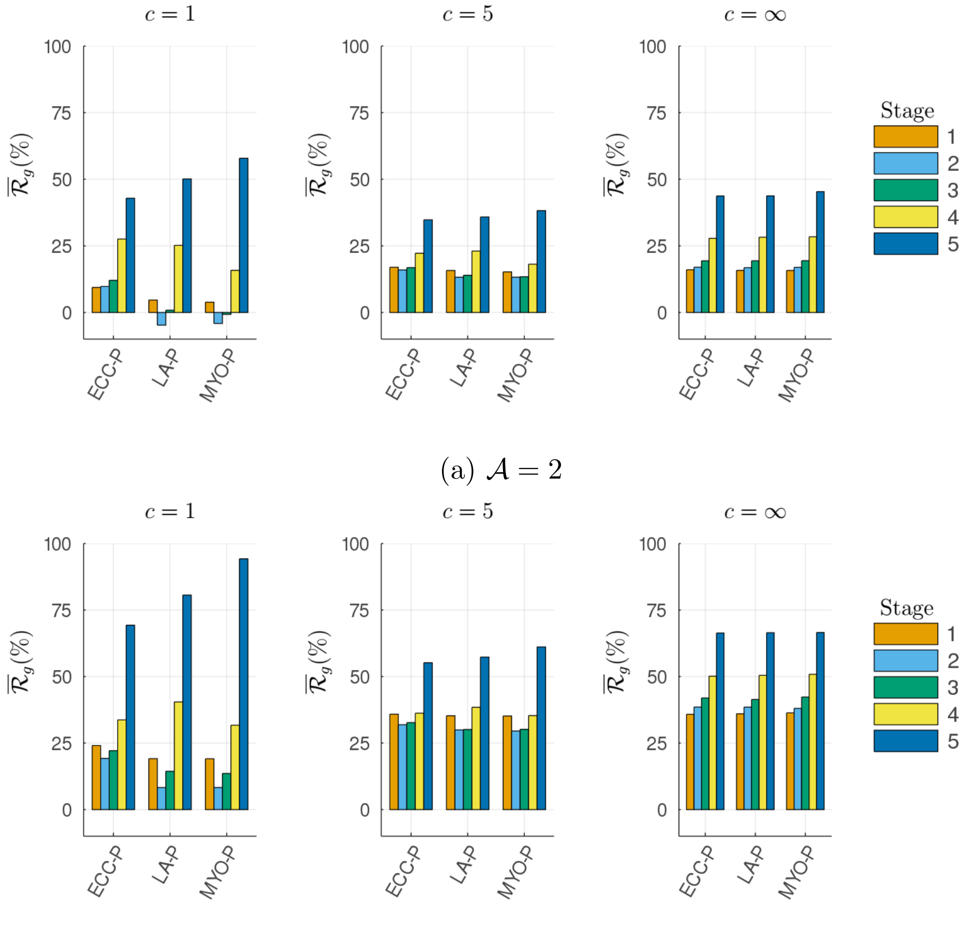
(a) $\mathcal{A} = 2$

(b) $\mathcal{A} = 5$

Figure 4.3.6: Stage-wise mean relative regret $\overline{\mathcal{R}}_g$ of each stage for $\mathcal{M} = 2$. The one-stage look-ahead policy does not feature in this figure as it is equivalent to the ECC policy when $\mathcal{M} = 2$.

the associated regret. However, early sub-optimal actions can lead to physical states which are hard to transition out from, resulting in the increasing stage-wise regret for later stages. The ECC is affected in a similar way; early actions are taken based on little information, and the resulting transitions can force the process into physical states from which completion of the intelligence puzzle is costly.

We conclude the numerical study by summarising our findings. We have seen that while considering the physical state till the end of the problem will reduce the expected



(a) $\mathcal{A} = 2$

(b) $\mathcal{A} = 5$

Figure 4.3.7: Stage-wise mean relative regret $\overline{\mathcal{R}}_g$ of each stage for $\mathcal{M} = 5$.

completion cost of the puzzle, the dominating issue is what sets this puzzle apart from the basic intelligence puzzle; learning the distribution of the P_m^a. The limitation on our ability to learn due to basing posterior estimates only on repeated observations of failures has a negative impact on the costs accrued throughout the intelligence puzzle. However, the degree of learning which is possible under this model can still provide actionable information; in puzzles which require more types of information and therefore need more observations to complete the mean regret is lower for all policies. The inaccurate estimates of P_m^a which the decision maker must rely on also limit any policy's ability to take advantage of reductions in evaluation times for the already discovered types. Our results have also shown the negative impact sub-optimal actions taken early on have on the final stages of the intelligence puzzle. However, we expect this issue to present in any version of the intelligence puzzle that involves the need to learn one or more parameters of the problem.

4.4 The Intelligence Puzzle with Unknown Encounter Probabilities

Similar to extending the intelligence puzzle to incorporate random conditional success probabilities as shown in Section 4.3, the basic intelligence puzzle may be extended to limit the information initially available on the encounter probabilities. In this section we look at a a version of the intelligence puzzle where the encounter probabilities are given by the random vector $\boldsymbol{Q}^a$. The task of the decision maker is not only to solve the puzzle, but in order to do so learn about the distribution of Q^a. The conditional success probabilities $\boldsymbol{p}^a(\boldsymbol{s})$ and the evaluation times $\boldsymbol{t}^a(\boldsymbol{s})$ are treated as known quantities and behave as described in Section 4.1.2, no different than their behaviour in the basic intelligence puzzle. Then the base parameter vectors characterising each source $a \in \{1, ..., A\}$ are

$$\boldsymbol{Q}^a = \begin{bmatrix} Q_1^a \\ Q_2^a \\ \cdots \\ Q_{\mathcal{M}}^a \end{bmatrix}, \qquad \boldsymbol{p}^a = \begin{bmatrix} p_1^a \\ p_2^a \\ \cdots \\ p_{\mathcal{M}}^a \end{bmatrix}, \qquad \boldsymbol{t}^a = \begin{bmatrix} t_1^a \\ t_2^a \\ \cdots \\ t_{\mathcal{M}}^a \end{bmatrix}.$$

While $\boldsymbol{p}^a$ and $\boldsymbol{t}^a$ remain unchanged throughout the puzzle, our belief of the distribution of $\boldsymbol{Q}^a$ evolves with every tip evaluated from source a. As seen with q_m, for all a and m $0 \leq Q_m^a \leq 1$ and $0 \leq Q_m^a \leq 1$. Note that since each tip must belong to one of the $\mathcal{M}$ categories, the random encounter probabilities must still satisfy the constraint $\sum_{m=1}^{\mathcal{M}} Q_m^a = 1$.

Section 4.4.1 describes conjugate Bayesian learning structure used to update our beliefs of $\boldsymbol{Q}^a$, while Section 4.4.2 describes the intelligence puzzle with unknown encounter probabilities in the framework of semi-Markov decision processes. Section 4.4.3 follows on with a discussion of potential policies which have been adapted from Section 4.2.2 in a manner to mirror the policies constructed in Section 4.3.3. Finally, a numerical study to evaluate these policies is presented in Section 4.4.4.

4.4.1 Bayesian Learning of $\boldsymbol{Q}^a$

Bayesian learning is used to estimate the distribution of all encounter probabilities. For this, we assume that all $\boldsymbol{Q}^a$ are independent of each other and the other parameter vectors. Note that $\sum_{m=1}^{\mathcal{M}} Q_m^a = 1$ means that a dependence between the elements of $\boldsymbol{Q}^a$ is present. For the remainder of this subsection we focus on a single source a, and omit the superscript in $\boldsymbol{Q}^a$ denoting the source in question.

As described in Section 4.1.2, M is a random variable and its realisations determine the type of intelligence a tip provides. It follows a Categorical distribution. Now the parameter vector of this Categorical distribution is itself random, given by the vector $\boldsymbol{Q}$ of length $\mathcal{M}$. The Dirichlet distribution has the probability density function

$$f(\boldsymbol{\phi}) = \frac{1}{B(\boldsymbol{\phi})} \prod_{d=1}^{\mathcal{M}} x_d^{\phi_d - 1} \quad \text{where } \sum_{d=1}^{\mathcal{M}} x_d = 1 \text{ and } x_d \in [0,1] \ \forall d \in \{1, \ldots, \mathcal{M}\} \tag{4.4.1}$$

The vectors $\boldsymbol{x}$ and $\boldsymbol{\phi}$ are both of length $\mathcal{M}$ and $B(\boldsymbol{\phi})$ is the multivariate beta function. It is a conjugate prior to the Categorical distribution, therefore we place a conjugate Dirichlet($\boldsymbol{\phi}^{(0)}$) distribution on $\boldsymbol{Q}$. We will use the superscript to denote the number of observations made. The structure of the Bayesian learning problem is summarised as

$$\begin{aligned} M \mid Q &\sim \text{Categorical}(\boldsymbol{Q}), \\ \boldsymbol{Q} &\sim \text{Dirichlet}(\boldsymbol{\phi}^{(0)}), \\ \boldsymbol{Q} \mid \boldsymbol{m} &\sim \text{Dirichlet}(\boldsymbol{\phi}^{(n)}), \end{aligned}$$

where $\boldsymbol{m}$ contains n observations of M. The vectors $\boldsymbol{\phi}^{(0)}$ and $\boldsymbol{\phi}^{(n)}$ are the prior and the posterior concentration parameters respectively. The posterior parameters are updated as observations are observed. For all $j \in 1, ..., n$

$$\phi_m^{(n)} = \begin{cases} \phi_m^{(n-1)} + 1 & \text{if } y_n = m, \\ \phi_m^{(n-1)} & \text{if } y_n \neq m, \end{cases}$$

where y_n is the outcome of the n^{th} observation of M, and m referring to the type of the information in general. In essence $\phi_m^{(n)}$ is the sum of $\phi_m^{(0)}$ and the number of tips of type m observed. Then the expected value of every $q_m^{(n)}$ is

$$\mathbb{E}\left[Q_m^{(n)}\right] = \frac{\phi_m^{(n)}}{\sum_{m'=1}^{\mathcal{M}} \phi_{m'}^{(n)}}. \tag{4.4.2}$$

Note that observing any outcome of M changes the expected value of all Q_m.

With the Bayesian learning aspect of the problem discussed, we return to considering all sources and denoting a specific source by the superscript a.

4.4.2 Setting out the SMDP framework

In this section we formulate the intelligence puzzle with unknown encounter probabilities as a semi-Markov decision process. There are many similarities between this formulation and that of the intelligence puzzle with unknown conditional success probabilities, therefore Section 4.3.2 is often referred to.

Since not all parameter vectors are known, the state of the intelligence puzzle is not sufficiently described by physical states. Similarly to Section 4.3.2 the state of the intelligence puzzle will be a combination of the physical state and the knowledge state of the unknown parameter vector. In this case the unknown parameter vectors are $\boldsymbol{Q}^a$, and therefore the knowledge state of source a is given by its posterior $\boldsymbol{\phi}^a$ parameter, which captures our most up to date beliefs on the distribution of $\boldsymbol{Q}^a$. Then the knowledge state of the intelligence puzzle $\mathcal{S}_{\boldsymbol{\phi}}$ is given by the set of all possible $\boldsymbol{\phi}$'s where

$$\boldsymbol{\phi} = (\boldsymbol{\phi}^1, ..., \boldsymbol{\phi}^{\mathcal{A}}).$$

Since there is no limit on the number of observations permitted of a given type and source, the values of ϕ_m^a have no upper limit and the knowledge state space $\mathcal{S}_{\boldsymbol{\phi}}$ is infinite.

The physical state space $\mathcal{S}_{\boldsymbol{s}}$ of the intelligence puzzle is as set out in Section 4.1.2, with the individual states denoted by a vector $\boldsymbol{s}$. Its elements s_m represent the physical state of intelligence type m, $s_m = 0$ and $s_m = 1$ corresponding to having or having not been discovered.

The combined state space of the intelligence puzzle is the product of the knowledge and physical state space; $\mathcal{S}_{\boldsymbol{s},\boldsymbol{\phi}} = \mathcal{S}_{\boldsymbol{s}} \times \mathcal{S}_{\boldsymbol{\phi}}$, and its state is the combination of its physical and knowledge state, denoted as the tuple $(\boldsymbol{s}, \boldsymbol{\phi})$. While the physical state space is finite, the knowledge state space is not, and therefore the state space of the intelligence puzzle with unknown encounter probabilities is also infinite.

The decision epochs and possible actions of the intelligence puzzle with unknown encounter probabilities is identical to those of the intelligence puzzle with unknown conditional success probabilities, and is described in detail in Section 4.3.2. Here we offer a condensed recap. Decision epochs occur at random intervals, triggered by state transitions with respect to either the information and physical state or only the information state. The timing of these decision epochs translate to having to make a new decision after a tip is evaluated, and the sojourn times directly correspond to the evaluation times of the tips. The start and end-point of individual *stages* of the puzzle, described in detail in Section 4.1.2, always coincide with a decision epoch, but a single stage may contain multiple decision epochs. As before, a puzzle consisting of $\mathcal{M}$ types of information has $\mathcal{M}$ stages. At every decision epoch an action a must be taken, chosen from the set of available actions $\mathcal{S}_a = \{1, ..., \mathcal{A}\}$ which is independent of the state of the decision process; any action may be taken at any decision epoch. We consider mainly deterministic stationary Markov policies in our investigations, so that policy π which determines the action taken at a given epoch only depends on the state of the system and prescribes the action $a = \pi(\boldsymbol{s}, \boldsymbol{\phi})$.

The action a taken in state s determines the probability distribution of both the subsequent states, and the time till the next decision epoch. Since only one tip is evaluated in each epoch, obtaining these quantities is straightforward. There is an important difference between the puzzles with unknown success and encounter probabilities in what constitutes a one-epoch transition. When the success probabilities are unknown, the system transitions either with respect to the knowledge state, or with respect to the physical state. However, in the problem version with unknown encounter probabilities every state transition is a transition with respect to the knowledge state, and may or may not include a physical transition depending on the outcome of the evaluated tip. This is due to the fact that every single tip belongs to one the $\mathcal{M}$ types, and therefore the distribution of the encounter probability $\boldsymbol{Q}^a$ is updated after every tip-evaluation. It does not matter if that tip produced a magic bullet or not.

Let us denote by $\boldsymbol{s}_{m^+}$ the resulting physical state of the puzzle after a type m magic bullet is discovered, and by $\boldsymbol{\phi}^a_{m^+}$ the knowledge state resulting from encountering a type m tip after taking action a. Then the probability that the process transitions from state $(\boldsymbol{s}, \boldsymbol{\phi})$ to $(\boldsymbol{s}_{m^+}, \boldsymbol{\phi}^a_{m^+})$ in response to action a is given by $\mathbb{P}(\boldsymbol{s}_{m^+}, \boldsymbol{\phi}^a_{m^+} \mid \boldsymbol{s}, \boldsymbol{\phi}, a)$. The probability that in response to the same action it transitions to $(\boldsymbol{s}, \boldsymbol{\phi}^a_{m^+})$, is given by $\mathbb{P}(\boldsymbol{s}, \boldsymbol{\phi}^a_{m^+} \mid \boldsymbol{s}, \boldsymbol{\phi}, a)$. Since the physical state transitions to $\boldsymbol{s}_{m^+}$ upon discovery of a type m magic bullet, for which encountering a type m tip is necessary, the probability of a transition to $(\boldsymbol{s}_{m^+}, \boldsymbol{\phi}^a_{m^+})$ is the product of the probability of encountering a type m tip and the expectation of the conditional probability that it leads to discovering a magic bullet

$$\mathbb{P}(\boldsymbol{s}_{m^+}, \boldsymbol{\phi}^a_{m^+} \mid \boldsymbol{s}, \boldsymbol{\phi}, a) = \mathbb{E}\left[Q^a_m(\boldsymbol{\phi})\right] p^a_m(\boldsymbol{s}) = \frac{\phi^a_m}{\sum_{m'=1}^{\mathcal{M}} \phi^a_{m'}} p(\boldsymbol{s}). \tag{4.4.3}$$

Similarly, the state transitions to $(\boldsymbol{s}, \boldsymbol{\phi}^a_{m^+})$ when a type m tip is encountered, but it does not provide a magic bullet. This occurs with probability

$$\mathbb{P}(\boldsymbol{s}, \boldsymbol{\phi}^a_{m^+} \mid \boldsymbol{s}, \boldsymbol{\phi}, a) = \mathbb{E}\left[Q^a_m(\boldsymbol{\phi})\right] \left(1 - p^a_m(\boldsymbol{s})\right) = \frac{\phi^a_m}{\sum_{m'=1}^{\mathcal{M}} \phi^a_{m'}} \left(1 - p^a_m(\boldsymbol{s})\right). \tag{4.4.4}$$

Since the process transitions after every evaluated tip, given action a the distribution of the sojourn times $\mathcal{T}^a(\boldsymbol{s}, \boldsymbol{\phi})$ is simply the the distribution of the evaluation times of the tips, namely

$$\mathbb{P}\left(\mathcal{T}^a(\boldsymbol{s},\boldsymbol{\phi}) = t^a_m(\boldsymbol{s})\right) = \mathbb{E}\left[Q^a_m\right] = \frac{\phi^a_m}{\sum_{m'=1}^{\mathcal{M}} \phi^a_{m'}}. \tag{4.4.5}$$

Since this shows that the sojourn times do not depend on the physical state we drop the dependence in the notation as well and will denote it as $\mathcal{T}^a(\boldsymbol{\phi})$. From its distribution, the expected sojourn time may also be obtained as

$$\mathbb{E}\left[\mathcal{T}^a(\boldsymbol{\phi})\right] = \frac{\sum_{m=1}^{\mathcal{M}} \phi^a_m t^a_m(\boldsymbol{s})}{\sum_{m=1}^{\mathcal{M}} \phi^a_m}. \tag{4.4.6}$$

The cost structure of the decision process is identical to that in Section 4.3.2 where the continuous cost rate of the process only depends on the physical state and is denoted as $w(\boldsymbol{s})$. The only difference is how the expected sojourn time is obtained so that the expected cost of action a in state $(\boldsymbol{s},\boldsymbol{\phi})$ is

$$r(\boldsymbol{s},\boldsymbol{\phi},a) = w(\boldsymbol{s})\frac{\sum_{m=1}^{\mathcal{M}} \phi^a_m t^a_m(\boldsymbol{s})}{\sum_{m=1}^{\mathcal{M}} \phi^a_m}. \tag{4.4.7}$$

Note that regardless of the information state, once the physical end state is reached $w(\mathbf{0}_{\mathcal{M}})$ and therefore $r(\mathbf{0}_{\mathcal{M}},\boldsymbol{\phi},a)$ is 0.

Finally, the objective of the decision process is to minimise the undiscounted total accumulated cost, and therefore

$$\text{Obj: } \min_{\pi} \sum_{e} \mathbb{E}\left[r\left(\boldsymbol{s}_e,\boldsymbol{\phi}_e,\pi(\boldsymbol{s}_e,\boldsymbol{\phi}_e)\right)\right], \tag{4.4.8}$$

where $\boldsymbol{s}_e$ and $\boldsymbol{\phi}_e$ are the physical and knowledge states at the e^{th} epoch and $\pi(\boldsymbol{s}_e,\boldsymbol{\phi}_e)$ the action prescribed by policy π.

4.4.3 Policies

In this section we develop policies to determine good actions to take in any given state of the intelligence puzzle with unknown encounter probabilities. Following the same structure as we did in Section 4.3.3 we derive a policy based on dynamic programming, then devise heuristic policies using the point estimates of Q^a_m that are analogous to some of the policies in Section 4.2.2. As in previous sections, the flowchart in Figure 4.4.1 is used to help describe how these policies are applied to the intelligence puzzle with unknown encounter probabilities and to further illustrate how the decision process moves through the different states.

Dynamic Programming Approach

Following on from Section 4.4.2 we recursively construct a policy for the intelligence puzzle with unknown encounter probabilities. The infinite size of the state space leads to the same problem present in Section 4.3.3, that there is no natural end state to use in a backwards recursive solution. This is resolved by first constructing the infinite recursion sequence, which is then truncated by limiting the number of observations allowed of a single source.

To start the infinite recursion sequence we need the immediate expected cost incurred, which is the cost of taking action a, namely $r(\boldsymbol{s}, \boldsymbol{\phi}, a)$. As a consequence of action a, two types of transitions are possible, from state $(\boldsymbol{s}, \boldsymbol{\phi})$ to $(\boldsymbol{s}_{m^+}, \boldsymbol{\phi}^a_{m^+})$ with probability $\mathbb{P}(\boldsymbol{s}_{m^+}, \boldsymbol{\phi}^a_{m^+} \mid \boldsymbol{s}, \boldsymbol{\phi}, a)$ or to $(\boldsymbol{s}, \boldsymbol{\phi}^a_{m^+})$ with probability $\mathbb{P}(\boldsymbol{s}, \boldsymbol{\phi}^a_{m^+} \mid \boldsymbol{s}, \boldsymbol{\phi}, a)$. These are given given in (4.4.3) and (4.4.4) respectively. The value function of action a in state $(\boldsymbol{s}, \boldsymbol{\phi})$ is denoted as $\mathbb{V}^a(\boldsymbol{s}, \boldsymbol{\phi})$. The only natural end states of the decision process are at $(\boldsymbol{s} = \mathbf{0}_{\mathcal{M}}, \boldsymbol{\phi})$ which has no costs associated with it and from where the system cannot transition to states with non-zero costs. Therefore the values of such states are

$$\mathbb{V}(\mathbf{0}_{\mathcal{M}}, \boldsymbol{\phi}) = 0. \tag{4.4.9}$$

In any other state the value function is the sum of the immediate cost of action $r(\boldsymbol{s}, \boldsymbol{\phi}, a)$ and the expected future cost due to transitioning to a different state, which are $\mathbb{P}(\boldsymbol{s}_{m^+}, \boldsymbol{\phi}^a_{m^+} \mid \boldsymbol{s}, \boldsymbol{\phi}, a)\mathbb{V}(\boldsymbol{s}_{m^+}, \boldsymbol{\phi}^a_{m^+})$ and $\mathbb{P}(\boldsymbol{s}, \boldsymbol{\phi}^a_{m^+} \mid \boldsymbol{s}, \boldsymbol{\phi}, a)\mathbb{V}(\boldsymbol{s}, \boldsymbol{\phi}^a_{m^+})$. Then the value

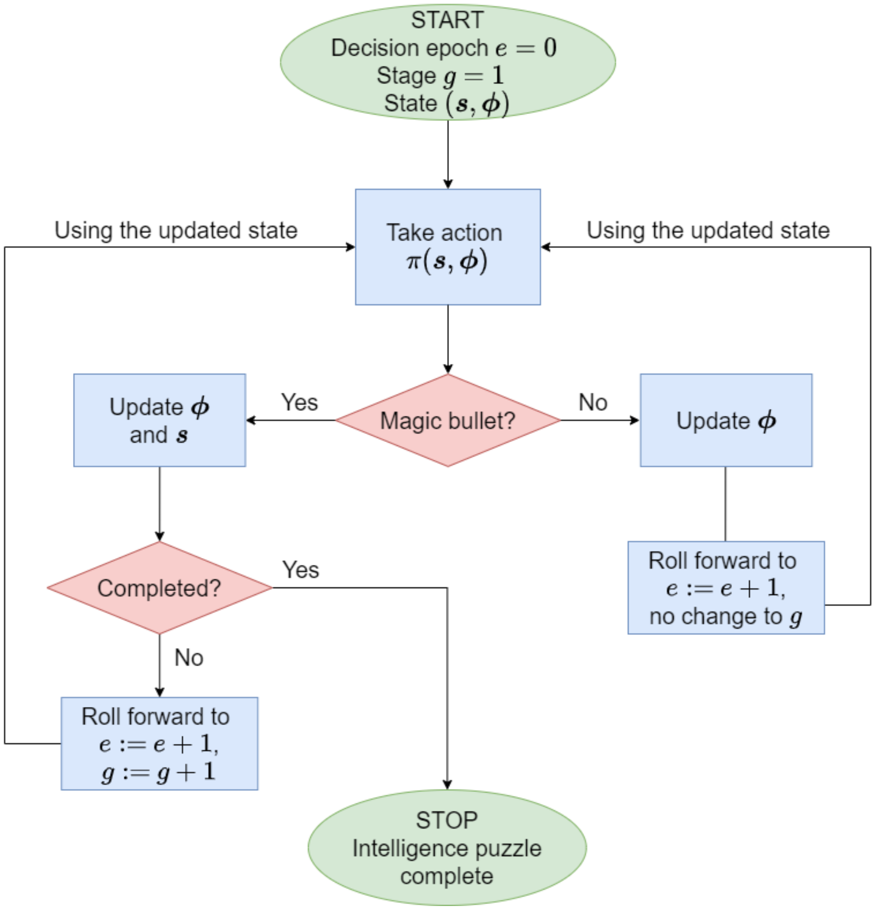

START
Decision epoch $e = 0$
Stage $g = 1$
State $(\boldsymbol{s}, \boldsymbol{\phi})$
Take action
$\pi(\boldsymbol{s}, \boldsymbol{\phi})$
Using the updated state
Using the updated state
Magic bullet?
Yes
No
Update $\boldsymbol{\phi}$ and $\boldsymbol{s}$
Update $\boldsymbol{\phi}$
Completed?
Yes
No
Roll forward to $e := e + 1$, no change to g
Roll forward to $e := e + 1$, $g := g + 1$
STOP
Intelligence puzzle complete

Figure 4.4.1: The process of applying policy $\pi(\boldsymbol{s}, \boldsymbol{\phi})$ to the intelligence puzzle with unknown conditional success probabilities. The updating process of physical state $\boldsymbol{s}$ has been described in detail in Section 4.1.2. The process of updating $\boldsymbol{\phi}$ takes place through updating the knowledge state of the source corresponding to the prescribed action, and is discussed in Section 4.4.1. The policy π may be any policy in Section 4.4.3.

$\mathbb{V}(\boldsymbol{s},\boldsymbol{\phi})$ of every state can be recursively obtained via

$$\begin{aligned}\mathbb{V}(\boldsymbol{s},\boldsymbol{\phi}) &= \min_{a\in\mathcal{S}_a}\left\{\mathbb{V}^a(\boldsymbol{s},\boldsymbol{\phi})\right\}\\ &= \min_{a\in\mathcal{S}_a}\Bigg\{r(\boldsymbol{s},\boldsymbol{\phi},a)+\sum_{m=1}^{\mathcal{M}}\mathbb{P}(\boldsymbol{s}_{m^+},\boldsymbol{\phi}_{m^+}\mid\boldsymbol{s},\boldsymbol{\phi},a)\mathbb{V}(\boldsymbol{s}_{m^+},\boldsymbol{\phi}^a_{m^+})\\ &\qquad+\sum_{m=1}^{\mathcal{M}}\mathbb{P}(\boldsymbol{s},\boldsymbol{\phi}_{m^+}\mid\boldsymbol{s},\boldsymbol{\phi},a)\mathbb{V}(\boldsymbol{s},\boldsymbol{\phi}^a_{m^+})\Bigg\}\\ &= \min_{a\in\mathcal{S}_a}\Bigg\{w(\boldsymbol{s})\frac{\sum_{m=1}^{\mathcal{M}}\phi^a_m t^a_m(\boldsymbol{s})}{\sum_{m=1}^{\mathcal{M}}\phi^a_m}+\frac{\sum_{m=1}^{\mathcal{M}}\phi^a_m p(\boldsymbol{s})\mathbb{V}(\boldsymbol{s}_{m^+},\boldsymbol{\phi}^a_{m^+})}{\sum_{m=1}^{\mathcal{M}}\phi^a_m}\\ &\qquad+\frac{\sum_{m=1}^{\mathcal{M}}\phi^a_m\left(1-p^a_m(\boldsymbol{s})\right)\mathbb{V}(\boldsymbol{s},\boldsymbol{\phi}^a_{m^+})}{\sum_{m=1}^{\mathcal{M}}\phi^a_m}\Bigg\}.\end{aligned}\tag{4.4.10}$$

Since the knowledge state space is infinite, there are no natural values of $\boldsymbol{\phi}$ which the backwards recursion could be started from. Instead, to allow for recursive calculation we cap the values of the posterior parameters at $\phi_{\max}$ so that

$$\phi_{\max}=\sum_{m=1}^{\mathcal{M}}\phi^a_m.\tag{4.4.11}$$

In effect, we limit the number of observations allowed from each source. Under this scheme $\boldsymbol{\phi}^a$ is only updated while $\sum_{m=1}^{\mathcal{M}}\phi^a_m<\phi_{max}$, and after that no further learning of the encounter probability vector takes place. Assuming that all prior parameter vectors and parameter caps are identical so that $\boldsymbol{\phi}^{(0),a}=\boldsymbol{\phi}^{(0)}$ and $\phi^a_{\max}=\phi_{\max}$ the size of the knowledge state is

$$\mid\mathcal{S}_{\boldsymbol{\phi}}\mid=\binom{\phi_{\max}-\sum_{m=1}^{\mathcal{M}}\phi^{(0)}_m-1}{\mathcal{M}-1}^{\mathcal{A}},\tag{4.4.12}$$

and consequently the size of the full state is

$$\mid\mathcal{S}_{\boldsymbol{s},\boldsymbol{\phi}}\mid=2^{\mathcal{M}}\binom{\phi_{\max}-\sum_{m=1}^{\mathcal{M}}\phi^{(0)}_m-1}{\mathcal{M}-1}^{\mathcal{A}}.\tag{4.4.13}$$

While the state space is no longer infinite it is still large and grows quickly with increasing $\mathcal{A}$ and $\mathcal{M}$.

In the truncated version of the problem a source has either reached $\phi_{\max}$ or we are still able to learn about them. Let us denote by $\lambda(\boldsymbol{\phi})$ the set of actions where learning has been constrained

$$\lambda(\boldsymbol{\phi}) = \left\{ a; \sum_{m=1}^{\mathcal{M}} \phi_m^a = \phi_{\max} \right\}. \tag{4.4.14}$$

When learning is possible the value function of the action associated with the source is given as in the non-truncated version of the problem;

$$\mathbb{V}^{a\notin\lambda(\boldsymbol{\phi})}(\boldsymbol{s},\boldsymbol{\phi}) = w(\boldsymbol{s})\frac{\sum_{m=1}^{\mathcal{M}} \phi_m^a t_m^a(\boldsymbol{s})}{\sum_{m=1}^{\mathcal{M}} \phi_m^a} + \frac{\sum_{m=1}^{\mathcal{M}} \phi_m^a p(\boldsymbol{s})\mathbb{V}(\boldsymbol{s}_{m^+},\boldsymbol{\phi}_{m^+}^a)}{\sum_{m=1}^{\mathcal{M}} \phi_m^a} + \frac{\sum_{m=1}^{\mathcal{M}} \phi_m^a \left(1 - p_m^a(\boldsymbol{s})\right) \mathbb{V}(\boldsymbol{s},\boldsymbol{\phi}_{m^+}^a)}{\sum_{m=1}^{\mathcal{M}} \phi_m^a}.$$

However, when $a \in \lambda(\boldsymbol{\phi})$ the knowledge state $\boldsymbol{\phi}_{m^+}^a$ does not exist. In that case the process transitions to state $(\boldsymbol{s}_{m^+},\boldsymbol{\phi})$ or $(\boldsymbol{s},\boldsymbol{\phi})$ with the same probability it would have transitioned to $(\boldsymbol{s}_{m^+},\boldsymbol{\phi}_{m^+}^a)$ and $(\boldsymbol{s},\boldsymbol{\phi}_{m^+}^a)$ and the value function takes the form

$$\mathbb{V}^{a\in\lambda(\boldsymbol{\phi})}(\boldsymbol{s},\boldsymbol{\phi}) = w(\boldsymbol{s})\frac{\sum_{m=1}^{\mathcal{M}} \phi_m^a t_m^a(\boldsymbol{s})}{\sum_{m=1}^{\mathcal{M}} \phi_m^a} + \frac{\sum_{m=1}^{\mathcal{M}} \phi_m^a p(\boldsymbol{s})\mathbb{V}(\boldsymbol{s}_{m^+},\boldsymbol{\phi})}{\sum_{m=1}^{\mathcal{M}} \phi_m^a} + \frac{\sum_{m=1}^{\mathcal{M}} \phi_m^a \left(1 - p_m^a(\boldsymbol{s})\right) \mathbb{V}(\boldsymbol{s},\boldsymbol{\phi})}{\sum_{m=1}^{\mathcal{M}} \phi_m^a}, \tag{4.4.15}$$

which can be rearranged to get

$$\mathbb{V}^{a\in\lambda(\boldsymbol{\phi})}(\boldsymbol{s},\boldsymbol{\phi}) \leq w(\boldsymbol{s})\frac{\frac{\sum_{m=1}^{\mathcal{M}} \phi_m^a t_m^a(\boldsymbol{s})}{\sum_{m=1}^{\mathcal{M}} \phi_m^a}}{\frac{\sum_{m=1}^{\mathcal{M}} \phi_m^a p_m^a(\boldsymbol{s})}{\sum_{m=1}^{\mathcal{M}} \phi_m^a}} + \frac{\frac{\sum_{m=1}^{\mathcal{M}} \phi_m^a p(\boldsymbol{s})\mathbb{V}(\boldsymbol{s}_{m^+},\boldsymbol{\phi})}{\sum_{m=1}^{\mathcal{M}} \phi_m^a}}{\frac{\sum_{m=1}^{\mathcal{M}} \phi_m^a p_m^a(\boldsymbol{s})}{\sum_{m=1}^{\mathcal{M}} \phi_m^a}} \tag{4.4.16}$$

$$= w(\boldsymbol{s})\frac{\sum_{m=1}^{\mathcal{M}} \phi_m^a t_m^a(\boldsymbol{s})}{\sum_{m=1}^{\mathcal{M}} \phi_m^a p_m^a(\boldsymbol{s})} + \frac{\sum_{m=1}^{\mathcal{M}} \phi_m^a p(\boldsymbol{s})\mathbb{V}(\boldsymbol{s}_{m^+},\boldsymbol{\phi})}{\sum_{m=1}^{\mathcal{M}} \phi_m^a p_m^a(\boldsymbol{s})}, \tag{4.4.17}$$

and denote the quantity in (4.4.17) as $\mathbb{V}'^a(\boldsymbol{s},\boldsymbol{\phi})$. The steps to arrive at (4.4.16) from (4.4.15) can be found in Appendix B.2.2. In essence, the value functions of actions in a given state provide an upper bound to the value function of the state, which is the value function of the minimising action. Consequently, when in order to simplify calculations the value of the state is replaced by the value of the action, the resulting quantity, $\mathbb{V}'^a(\boldsymbol{s},\boldsymbol{\phi})$, serves as an upper bound to the value of the state $\mathbb{V}^a(\boldsymbol{s},\boldsymbol{\phi})$. Since (4.4.16) evaluates to equality whenever a is the minimising action we can freely use the expression in (4.4.17) in the recursion. To capture which category an action belongs to we define the indicator function $\mathbb{I}(\boldsymbol{\phi}^a)$ as

$$\mathbb{I}(\boldsymbol{\phi}^a) = \begin{cases} 1 & \text{if } \sum_{m=1}^{\mathcal{M}} \phi_m^a = \phi_{\max}, \\ 0 & \text{if } \sum_{m=1}^{\mathcal{M}} \phi_m^a < \phi_{\max}, \end{cases}$$

and can write the truncated recursion as

$$\mathbb{V}(\boldsymbol{s}, \boldsymbol{\phi}) = \min_{a \in \mathcal{S}_a} \left\{ \mathbb{I}(\boldsymbol{\phi}^a) \mathbb{V}'^a(\boldsymbol{s}, \boldsymbol{\phi}) + \left(1 - \mathbb{I}(\boldsymbol{\phi}^a)\right) \mathbb{V}^a(\boldsymbol{s}, \boldsymbol{\phi}) \right\}. \tag{4.4.18}$$

Learning of $\boldsymbol{Q}^a$ in the setting of the intelligence process is well behaved, with none of the issues seen in Section 4.3.3 regarding the learning of $\boldsymbol{P}^a$ and the convergence of $\mathbb{V}(\boldsymbol{s}, \boldsymbol{\beta})$ present here. We expect that even a moderately large $\phi_{\max}$ would facilitate a good level of learning and result in a close to optimal policy, which we define as follows. A policy based on the recursion in (4.4.18) takes action a^* in state $(\boldsymbol{s}, \boldsymbol{\phi})$ so that

$$a^* := \arg\min_{a \in \mathcal{S}_a} \left\{ \mathbb{I}(\boldsymbol{\phi}^a) \mathbb{V}'^a(\boldsymbol{s}, \boldsymbol{\phi}) + \left(1 - \mathbb{I}(\boldsymbol{\phi}^a)\right) \mathbb{V}^a(\boldsymbol{s}, \boldsymbol{\phi}) \right\}. \tag{4.4.19}$$

While the above policy is expected to have significant merit, its implementation is beyond the scope of this work. Instead we focus on policies where the knowledge state only provides point estimates to the distribution of $\boldsymbol{Q}^a$. These policies directly correspond to the ones considered in Section 4.3.3, and therefore they are discussed in less detail.

Expected Completion Cost Policy (ECC)

The expected completion cost of the puzzle in any given state is provided by $\mathbb{V}(\boldsymbol{s}, \boldsymbol{\phi})$ as per (4.4.18). Having no interest in how the knowledge state may evolve as the process progresses puts us in the same regime as if every source has reached their respective $\phi_{\max}$'s, obtaining estimates of the $\boldsymbol{Q}^a$'s based only on past observations. To signal that the expected completion cost is only dependent on the knowledge state through its point estimate and is treated as a constant in calculating this cost, we denote it as $\mathbb{V}(\boldsymbol{s} \mid \boldsymbol{\phi})$.

Since the value of any state in which the physical state has reached $\mathbf{0}_{\mathcal{M}}$ is 0 independent of the knowledge state, all $(\mathbf{0}_{\mathcal{M}} \mid \boldsymbol{\phi})$ serve as end states with

$$\mathbb{V}(\mathbf{0}_{\mathcal{M}} \mid \boldsymbol{\phi}) = 0. \tag{4.4.20}$$

The value of all other states can be recursively calculated from the end state. Since learning is limited in the same way as in the artificial end states of the truncated recursion $\min_{a\in\mathcal{S}_a}\{\mathbb{V}^a(\boldsymbol{s} \mid \boldsymbol{\phi}^a)\} = \min_{a\in\mathcal{S}_a}\{\mathbb{V}'^a(\boldsymbol{s}, \boldsymbol{\phi}^a)\}$, and the expected completion cost given the current knowledge state can be written as

$$\begin{aligned}\mathbb{V}(\boldsymbol{s} \mid \boldsymbol{\phi}) &= \min_{a\in\mathcal{S}_a}\{\mathbb{V}^a(\boldsymbol{s} \mid \boldsymbol{\phi}^a)\}\\ &= \min_{a\in\mathcal{S}_a}\left\{ w(\boldsymbol{s})\frac{\sum_{m=1}^{\mathcal{M}}\phi_m^a t_m^a(\boldsymbol{s})}{\sum_{m=1}^{\mathcal{M}}\phi_m^a p_m^a(\boldsymbol{s})} + \frac{\sum_{m=1}^{\mathcal{M}}\phi_m^a p(\boldsymbol{s})\mathbb{V}(\boldsymbol{s}_{m^+}, \boldsymbol{\phi})}{\sum_{m=1}^{\mathcal{M}}\phi_m^a p_m^a(\boldsymbol{s})}\right\}.\end{aligned} \tag{4.4.21}$$

The above is analogous to the recursion equation of the basic intelligence puzzle found in Section 4.2.2. If we reintroduced the $\sum_{m=1}^{\mathcal{M}}\phi_m^a$ terms by dividing both the numerators and denominators of (4.4.21) by $\sum_{m=1}^{\mathcal{M}}\phi_m^a$, it would become apparent that (4.4.21) is a deterministically equivalent approximation to the completion cost of the intelligence puzzle as it replaces the unknown quantity Q_m^a with its point estimate, a known quantity. Then for every state $(\boldsymbol{s}, \boldsymbol{\phi})$ the expected completion cost policy is defined as

$$a^* = \pi^{\mathrm{ECC}}(\boldsymbol{s}, \boldsymbol{\phi}) := \arg\min_{a\in\mathcal{S}_a}\{\mathbb{V}^a(\boldsymbol{s} \mid \boldsymbol{\phi})\}. \tag{4.4.22}$$

Note that while the expected completion cost is calculated under the assumption that the information state remains constant till the end of the decision process, in reality it is updated every decision epoch. This is reflected in how the policy is applied, for which we refer to Figure 4.4.1. At every decision epoch e the policy in (4.4.22) is used to determine the action to take. If it does not yield a magic bullet, the knowledge state $\boldsymbol{\phi}$ is updated, and we roll forward to the next decision epoch $e+1$. If a magic bullet is discovered following the action in decision epoch e, both aspects of the state, namely $\boldsymbol{s}$ and $\boldsymbol{\phi}$ are updated, and when we roll forward to the next decision epoch $e+1$ the decision process also enters the next stage $g+1$. Regardless of which aspects of the state were updated following decision epoch e, when (4.4.22) determines the action to take in decision epoch $e+1$ the updated state is used. Consequently, the recursion from the end state to the current state has to be computed for every decision epoch and the states encountered throughout to determine the action dictated by such a policy, which requires significant computational effort. We refer the reader back to Table 4.2.1 and Table 4.2.2 for an indication of the time required to obtain the prescribed action at any one decision time, as the dynamic program we are required to solve here is in essence the same as that in Equation 4.2.7. This limits the use of the ECC to moderate-sized problems with $\mathcal{M} \lesssim 5$.

Myopic policy

We consciously mirror the policies developed for $\boldsymbol{P}^a$ in Section 4.3.3, and therefore move on to consider a policy which does not consider transitions with respect to either state types in reaction to an action taken. As before, we refer to this policy as myopic due to its short sighted nature.

While its aim is to minimise the expected cost of completing the current stage from its current state, the action-independent cost rate of the intelligence puzzle leads to the myopic policy prescribing the action a^* that minimises the expected duration of the stage. As seen previously, we use an approximation based on deterministic equivalence, therefore the myopic policy takes the form

$$a^* = \pi^{\mathrm{MYO}}(\boldsymbol{s}, \boldsymbol{\phi}) := \arg\min_{a \in \mathcal{S}_a} \left\{ \frac{\sum_{m=1}^{\mathcal{M}} \phi_m^a t_m^a(\boldsymbol{s})}{\sum_{m=1}^{\mathcal{M}} \phi_m^a p_m^a(\boldsymbol{s})} \right\},$$

which is calculated for every decision epoch e using the updated state of the puzzle in accordance with Figure 4.4.1.

A different measure of the expected duration of the stage may also be used, which utilises the distribution of $\boldsymbol{Q}^a$ instead of its point estimates. This measure, written as

$$\mathbb{E}\left[\frac{\sum_{m=1}^{\mathcal{M}} Q_m^a(\boldsymbol{\phi}^a) t_m^a(\boldsymbol{s})}{\sum_{m=1}^{\mathcal{M}} Q_m^a(\boldsymbol{\phi}^a) p_m^a(\boldsymbol{s})} \right] = \int_{\Delta_{\mathcal{M}-1}} \frac{\sum_{m=1}^{\mathcal{M}} Q_m^a(\boldsymbol{\phi}^a) t_m^a(\boldsymbol{s})}{\sum_{m=1}^{\mathcal{M}} Q_m^a(\boldsymbol{\phi}^a) p_m^a(\boldsymbol{s})} \prod_{m=1}^{\mathcal{M}} \pi(\boldsymbol{Q}^a) \mathrm{d}\boldsymbol{Q}^a,$$

is difficult to obtain however, as it requires numerical integration over the $\mathcal{M} - 1$ dimensional simplex. In addition, certain parameter combinations require $\phi_m^a > 1$ for all m for the integral to exist. Given the above, we do not consider this measure further.

One-Stage Look-Ahead Policy

To complete the set of policies that will be examined for the intelligence puzzle with unknown encounter probabilities, we state the one-stage look ahead policy as

$$\begin{aligned} a^* &= \pi^{\mathrm{LA}}(\boldsymbol{s}, \boldsymbol{\phi}) \\ &:= \operatorname*{arg\,min}_{a \in \mathcal{S}_a} \left\{ w(\boldsymbol{s}) \frac{\sum_{m=1}^{\mathcal{M}} \phi_m^a t_m^a(\boldsymbol{s})}{\sum_{m=1}^{\mathcal{M}} \phi_m^a p_m^a(\boldsymbol{s})} \right. \\ &\quad \left. + \sum_{m=1}^{\mathcal{M}} \frac{\phi_m^a p_m^a(\boldsymbol{s})}{\sum_{m'=1}^{\mathcal{M}} \phi_{m'}^a p_{m'}^a(\boldsymbol{s})} \operatorname*{arg\,min}_{a' \in \mathcal{S}_a} \left\{ w(\boldsymbol{s}_{m^+}) \frac{\sum_{m''=1}^{\mathcal{M}} \phi_{m''}^{a'} t_{m''}^{a'}(\boldsymbol{s}_{m^+})}{\sum_{m''=1}^{\mathcal{M}} \phi_{m''}^{a'} p_{m''}^{a'}(\boldsymbol{s}_{m^+})} \right\} \right\}. \end{aligned}$$

which utilises a deterministically equivalent approximation of the one-stage look-ahead cost. While it considers the physical state that immediately follows the current one, it disregards any costs accrued from decisions that follow after. Just like the ECC and the myopic policy, it treats the knowledge state as constant when determining the action to prescribe. However, the states are updated following every decision epoch, and at every decision epoch the policy determines which action to prescribe based on the current state of the process, as shown in Figure 4.4.1. This one-stage look-ahead policy is analogous to and is based on the same ideas as the one-step look-ahead policy in Section 4.2.2 and the one-stage look ahead policy in Section 4.3.3. Note that for $\mathcal{M} = 2$ the one-step look-ahead policy is identical to the ECC policy.

Super-Optimal Policy

Much like in Section 4.3.3, we can define a so-called super-optimal policy that utilises the optimal solution of the basic intelligence puzzle in Section 4.2.2 to obtain a lower bound on the expected completion cost. There is no difference between how the super-optimal policy is applied to an intelligence puzzle with unknown conditional success probabilities and unknown encounter probabilities. The only difference is which parameter type's value is unknown to the non-omniscient decision makers.

The super-optimal policy is then applied as follows. For every realisation of the intelligence puzzle, the underlying values of the encounter probabilities are immediately made known to the decision maker. Now all parameter vectors are known quantities to them. When all parameter vectors are known, the intelligence puzzle is as described in Section 4.2, for which an optimal policy can be obtained using the recursion in (4.2.7). As the $\boldsymbol{Q}^a$'s are treated like known quantities, information states are unnecessary and the optimal action only depends on the physical state of the puzzle. Based on the above, the Bayes cost of the super-optimal policy is then obtained by repeatedly applying it to independently sampled instances of the puzzle and averaging over the observed completion costs.

4.4.4 Numerical Experiments

This section examines the performance of the expected completion cost, myopic and one step look-ahead policies, which have been described throughout Section 4.4.3. All are compared to the super-optimal policy set out in the same section. Just as the policies mirror those developed for the puzzle with unknown success probabilities, so do the measures used to analyse their performance.

Just as in Section 4.3.4 we simplify the intelligence puzzle by setting the cost rate of all states with $\boldsymbol{s} \neq \mathbf{0}_{\mathcal{M}}$ to 1, and all base evaluation times to $t_m^a = 1$. Then minimising the expected completion time of the intelligence puzzle is an objective equivalent to minimising its expected cost of completion. For every realisation of the intelligence puzzle the underlying value of the encounter vector of source a, $\boldsymbol{Q}^a$ is sampled from its Dirichlet($\mathbf{1}_{\mathcal{M}}$) prior distribution, and all p_m^a are independently sampled from a Beta(1,1) distribution. We examine the intelligence puzzle for $\mathcal{A} = 2$ and $\mathcal{A} = 5$ combined with $\mathcal{M}$ increasing from 2 to 6. Three values of c are considered, namely 1, 5 and ∞. For every combination of these three parameters we consider 10^4 puzzles, defined by the combination of the sampled p_m^a's and the realisations if $\boldsymbol{Q}^a$'s. Each such puzzle is simulated for and averaged over 10 instances.

Note that the realisations of $\boldsymbol{Q}^a$ and the p_m^a are obtained from the same distributions as the $\boldsymbol{q}^a$ and the realisations of P_m^a are in the unknown success probability case. In addition, we are interested in the same combinations of $\mathcal{A}$, $\mathcal{M}$ and c, which contain the same number of realisations of the intelligence puzzle. This allows us to use the same dataset as Section 4.3.4, only that instead of the values of $P_m^a = p_m^a$ here the values of $\boldsymbol{Q}^a = \boldsymbol{q}^a$ are hidden from the decision maker.

In the figures found in this section abbreviations are used to identify the policies; ECC-Q stands for the expected completion cost, LA-Q the one step look-ahead and MYO-Q the myopic policy. The suffix "Q" is to signal that these are the versions of the policy in which the distribution of $\boldsymbol{Q}^a$ are initially unknown.

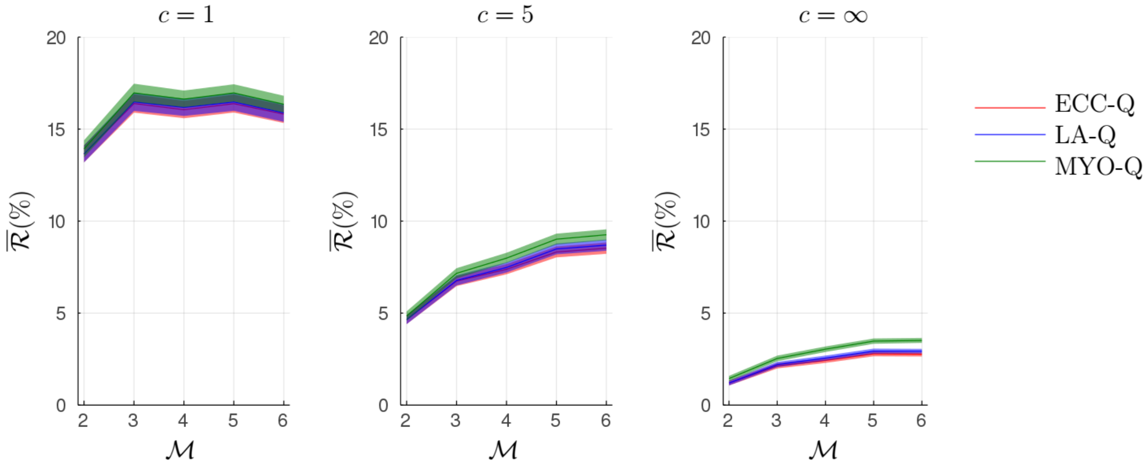


Figure 4.4.2: Mean relative regret $\overline{\mathcal{R}}$ for $\mathcal{A} = 2$ as a function of $\mathcal{M}$ with unit evaluation times.

Following the same structure as Section 4.3.4, we start by examining the mean regret of the policies, in which the regret of every realisation of the intelligence puzzle is averaged over the 10^4 realisations which we base this study on. These individual regrets are calculated using (4.3.20). Since we do not have access to the corresponding optimal completion costs, the super-optimal policy is used as a benchmark in our investigations. Figure 4.4.2 and Figure 4.4.3 show the mean regret as defined above for

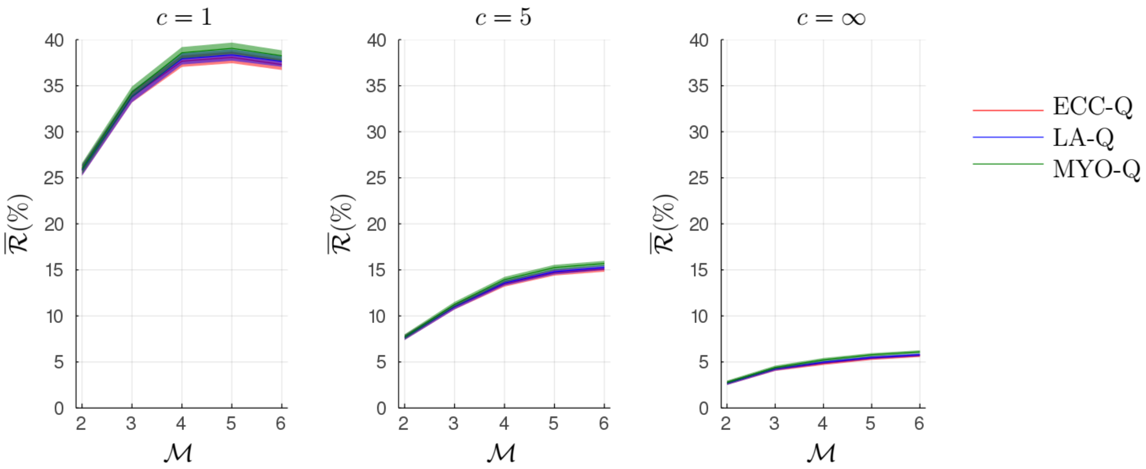


Figure 4.4.3: Mean relative regret $\overline{\mathcal{R}}$ for $\mathcal{A} = 5$ as a function of $\mathcal{M}$ with unit evaluation times.

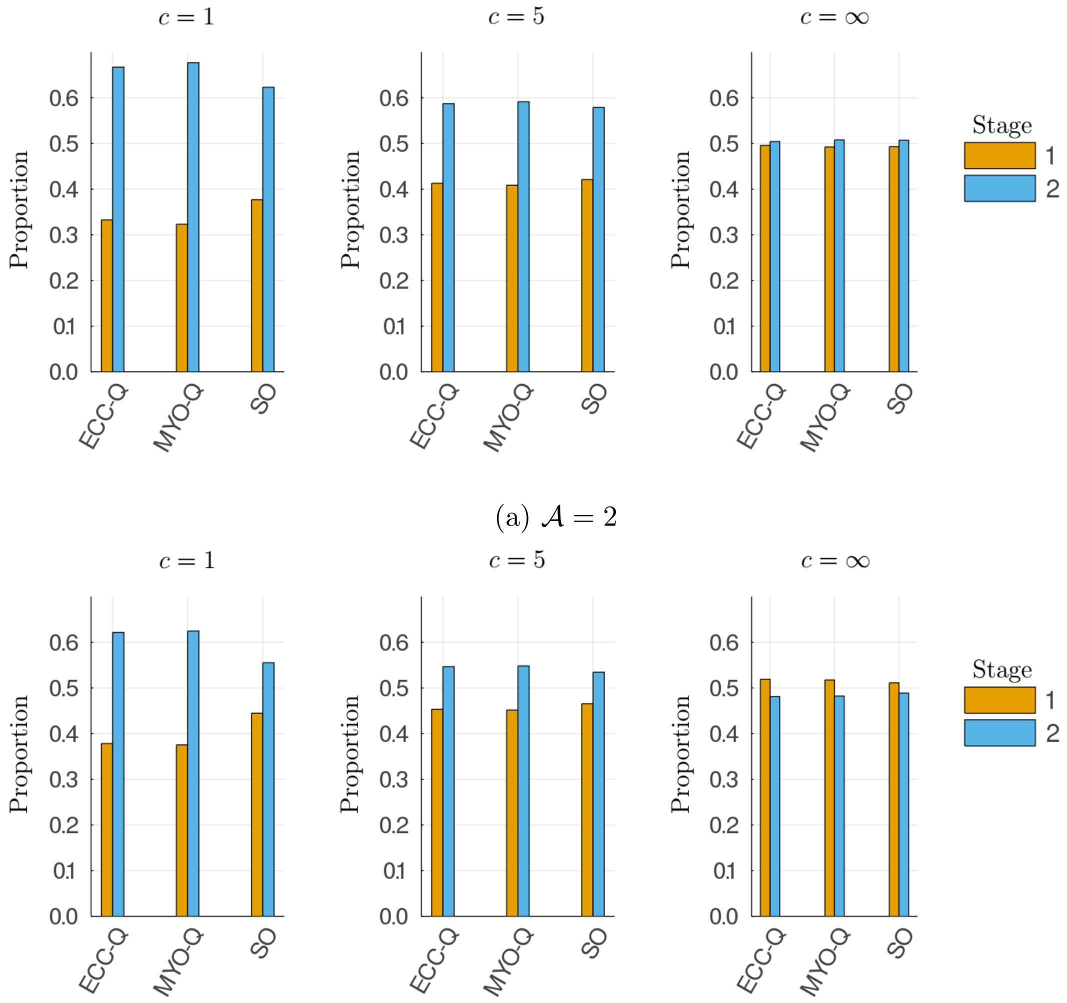
$\mathcal{A} = 2$ and $\mathcal{A} = 5$ respectively with increasing values of $\mathcal{M}$. We note that for some of the combinations of $\mathcal{A}$, $\mathcal{M}$ and c the mean regrets are quite high. However, when the numerical experiments were repeated using less vague Dirichlet($\mathbf{2}_{\mathcal{M}}$) and Dirichlet($\mathbf{5}_{\mathcal{M}}$) priors, we found that the mean regret of all policies decreased. The figures showing these findings can be found in Appendix B.3.

All in all, the behaviour of the three methods were very similar, and the expected ordering in which the ECC had the lowest mean regrets while the myopic policy had the largest is realised. Any differences in the performances of the policies were dominated by common behaviours that can be explained by the need to learn about the $\boldsymbol{Q}^a$'s and how learning of these parameters is carried out.

In stark contrast to our findings in Section 4.3.4, the mean regret increases with $\mathcal{M}$, but decreases with c, which is a behaviour that more closely resembles that of the basic intelligence puzzle than the intelligence puzzle with unknown success probabilities. While the observed ranges of the mean regrets for $c = 1$ are similar to what is seen when it is the P_m^a's that need to be learnt, the mean regrets for $c = \infty$ are more comparable to the regrets observed when the encounter probabilities were known. The regret achieved by the three policies here are even lower than the regret of the order-based policies considered in the no-learning case. The increase in regrets observed with the increase of $\mathcal{M}$ is similar to what is seen in Section 4.2.3; a puzzle with more pieces takes more epochs to solve and therefore there are more opportunities to take sub-optimal actions.

A reduction in regret as c is increased is caused by a combination of many factors. This is a behaviour which was already observed in the basic intelligence puzzle for the policies analogous to those we consider here. However, the original effect has been made more prominent by an array of factors. First of all, we need to note that due to the continued sequential learning of these parameters the best estimates of the $\boldsymbol{Q}^a$ are available at the end of the problem, and therefore that is when the decision maker can make the best, most informed decisions. Furthermore, the estimates of the $\boldsymbol{Q}^a$ improve with every single observation even if the encountered type can no longer provide a magic bullet as the Q_m^a's belonging to the same source are not independent. The more types are discovered, a higher proportion of the encountered tips will be nuisance tips. When $c = 1$ these contribute towards the completion cost in full, but as c increases these contribute less and less. Ultimately, this leads to an increase in the number of observations being made in order to complete the puzzle and consequently increased accuracy of our beliefs regarding the $\boldsymbol{Q}^a$'s. This in turn helps the decision maker exploit the reduced evaluation costs of the discovered types and drastically reduce the cost of completing the latter stages of the puzzle which require the most number of observations. Therefore, the reduction in the regrets as c increases is due to being able to obtain more accurate estimates of $\boldsymbol{Q}$ in the presence of a large c, but also due to the high regrets observed with $c = 1$ due to the less accurate estimation of $\boldsymbol{Q}^a$.

The above reasoning is supported both by examining the proportion the different stages contribute to the completion cost as shown in Figure 4.4.4 and Figure 4.4.5, and the stage-wise mean regrets shown in Figure 4.4.6 and Figure 4.4.7. Looking at the proportion of cost each stage is contributing, we see that the three policies behave very similarly and that the super-optimal is somewhat more balanced with a higher pro-

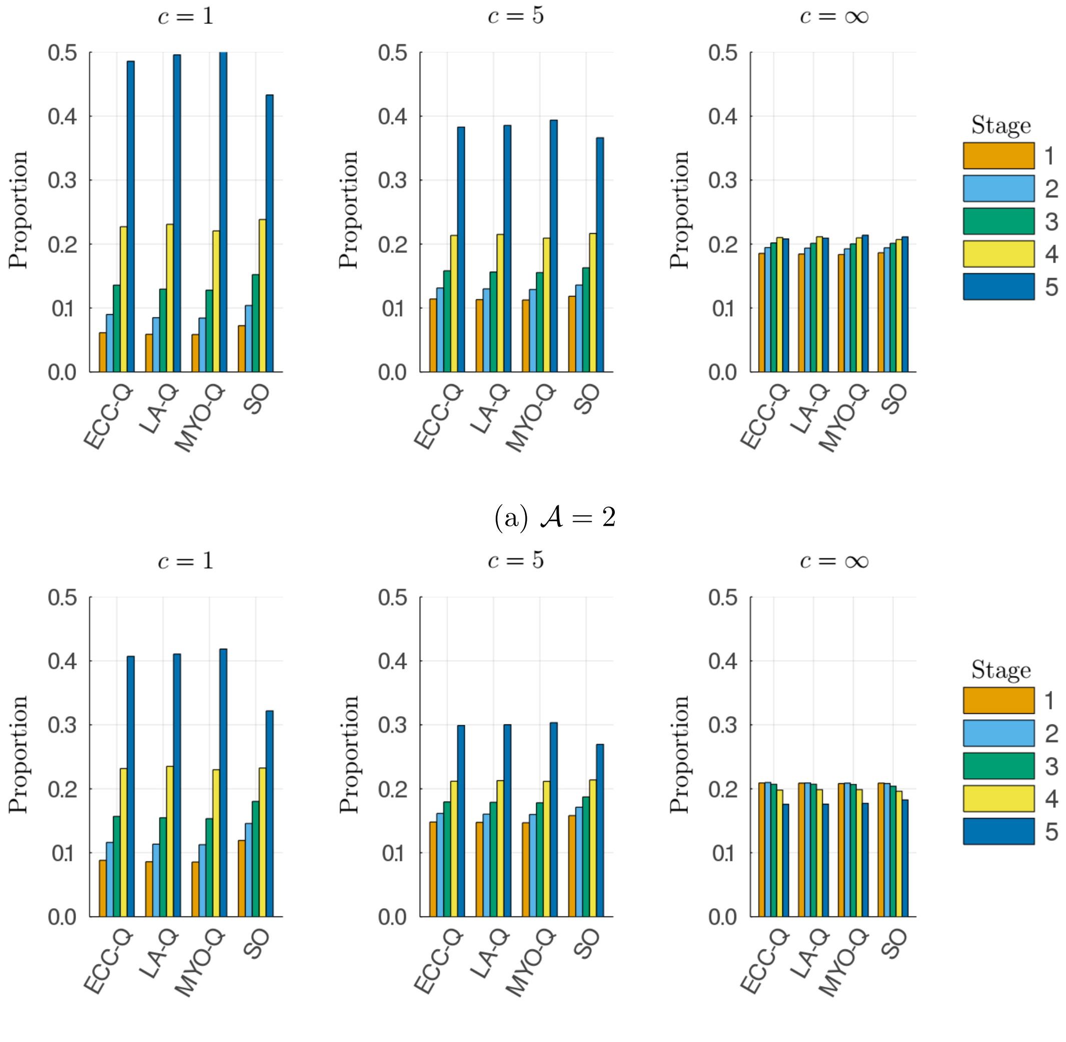


(a) $\mathcal{A} = 2$

(b) $\mathcal{A} = 5$

Figure 4.4.4: Proportion of completion time spent on each stage for $\mathcal{M} = 2$. The one-stage look-ahead policy does not feature in this figure as it is equivalent to the ECC policy when $\mathcal{M} = 2$.

portion of cost being attributed to the first stage and a lower proportion to the last stage of the problem. The most striking feature of Figure 4.4.4 and Figure 4.4.5 is the clear increase in the proportion of costs for early stages and a just as clear decrease for the latter stages with the increase of c. However, similar features have been seen in both other versions of the intelligence puzzle, and therefore these figures need to be interpreted in conjunction with the stage-wise regret figures.



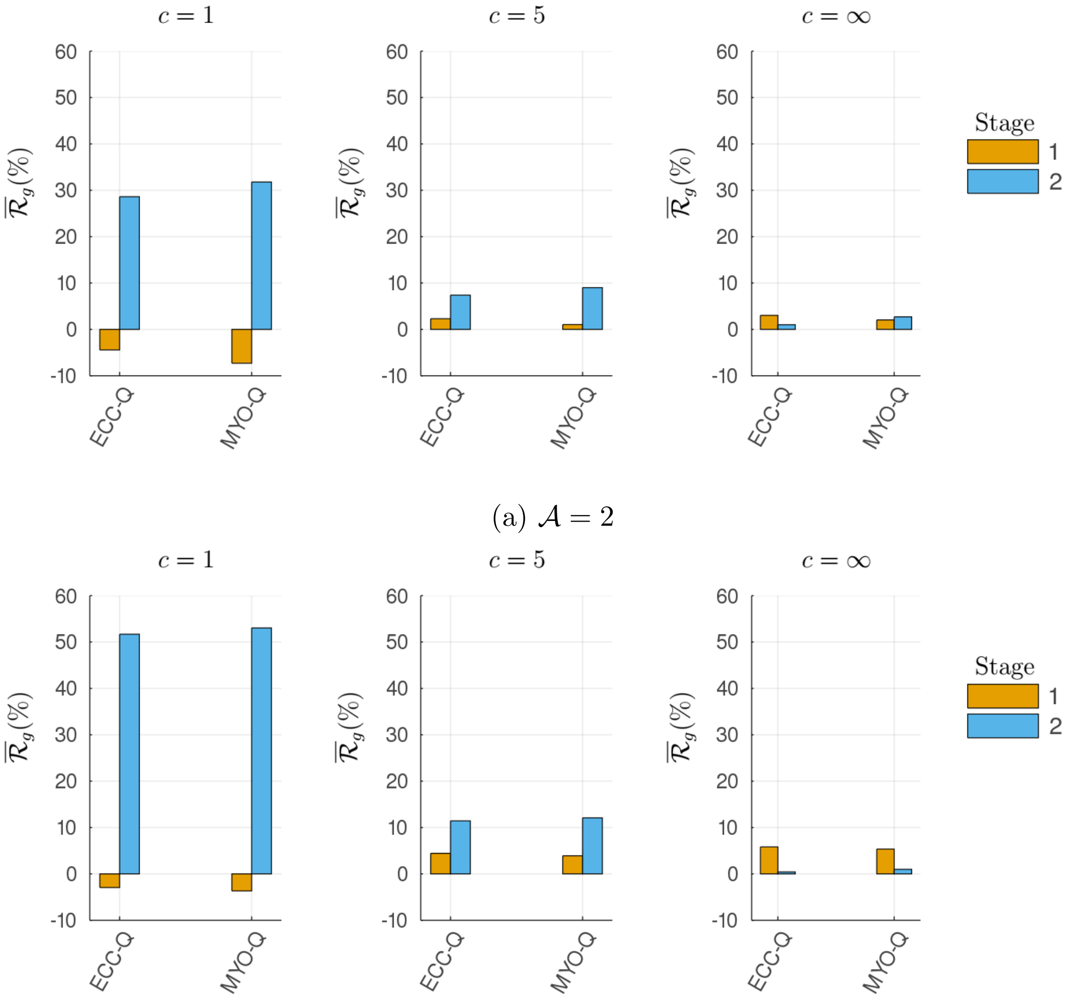
(a) $\mathcal{A} = 2$

(b) $\mathcal{A} = 5$

Figure 4.4.5: Proportion of completion time spent on each stage for $\mathcal{M} = 5$.

The stage-wise mean regrets used here are defined identically as before. The original mean regret measure is adapted to the individual stages to compare the costs accrued by the policies in a given stage g. The stage-wise-regret of a policy π for stage g is denoted $\mathcal{R}_g^\pi$ and is calculated using (4.3.21) for every realisation of the intelligence puzzle considered. Similarly to the mean total regret the super-optimal policy is used instead of the optimal policy as the basis of comparison. To get the mean stage-wise regret of a policy π for stage g, the stage-wise regrets of the puzzles are averaged over.

As expected, Figure 4.4.6 and Figure 4.4.7 show the stage-wise mean regrets to be

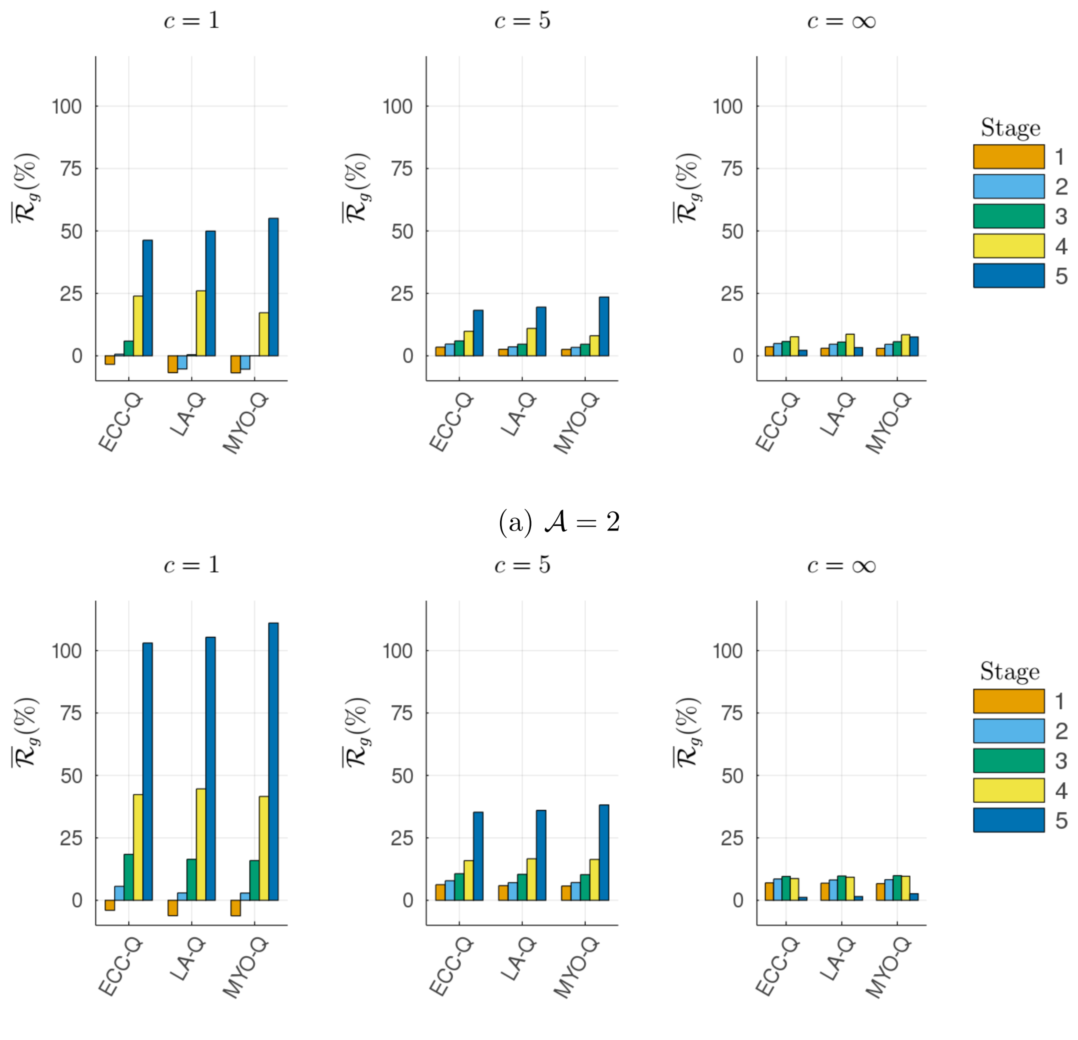


(a) $\mathcal{A} = 2$

(b) $\mathcal{A} = 5$

Figure 4.4.6: Stage-wise mean regret $\overline{\mathcal{R}}_m$ for $\mathcal{M} = 2$. The one-stage look-ahead policy does not feature in this figure as it is equivalent to the ECC policy when $\mathcal{M} = 2$.

much higher than they are in the basic intelligence puzzle. While small negative stage-wise regrets can be seen in the $c = 1$ case for the first stages of the puzzle for all four combinations of $\mathcal{A}$ and $\mathcal{M}$, this is followed by very large mean stage-wise regrets in later stages. Even when all parameters are known large last stage regrets may exist due to a myopic outlook regarding physical state transitions of a policy, but the ECC presenting with such large values suggests that the cause is sub-optimal actions in the early stages. This supports our conjecture that in the $c = 1$ case not enough observations are made to effectively learn about the $\boldsymbol{Q}^a$'s. On the other extreme when $c = \infty$,



(a) $\mathcal{A} = 2$

(b) $\mathcal{A} = 5$

Figure 4.4.7: Stage-wise mean regret $\overline{\mathcal{R}}_m$ for $\mathcal{M} = 5$.

all mean stage-wise regrets are positive, but comparably small. Most noticeably the mean stage-wise regret of the last stage has been much reduced. These results support the earlier discussion that $c = \infty$ provides an environment in which over the course of the intelligence puzzle an accurate estimate of the $\boldsymbol{Q}$ may be obtained and used to exploit the reduced evaluation times.

4.5 Conclusions

In the course of examining the intelligence puzzle, we have considered multiple variants of the problem. We have modelled each of them as a semi-Markov decision problem with the goal of bringing the investigation to its conclusion as quickly as possible by means of finding the pieces of the intelligence puzzle.

To start, we examined the situation where all elements of the intelligence puzzle (encounter probabilities, success probabilities, evaluation time) were known. In this case, we were able to optimally solve the problem via dynamic programming. We studied a variant of the problem in which the pieces of the intelligence puzzle must be discovered in a set order, and demonstrated that simple decision rules are optimal under these circumstances. Ultimately, we had to conclude that heuristic approaches based on those ordered decision rules do not transfer over well. By comparing the time the myopic policy spent on each stage of the problem with those of the optimal policy, we were able to ascertain that the sub-optimality of short sighted policies are due to reaching unfavourable states, from which discovery of the last magic bullet is time-consuming.

We found that when the vectors of conditional success probabilities were unknown, active learning approaches were impractical. The structure of the intelligence puzzle meant that only negative feedback could be used when updating our beliefs, leading to an increasingly pessimistic outlook concerning the completion of the puzzle. However, combining passive Bayesian learning with dynamic programming focused on the discovery of the intelligence pieces led to a heuristic which achieved lower regret than the myopic policy. On the other hand, when the encounter probabilities were unknown, we were able to incorporate learning about future information states in a dynamic program. We formulated a dynamic program that considered establishing good estimates of the encounter probabilities alongside the discovery of the intelligence pieces. We implemented approaches with passive learning only, analogous to those applied to the intelligence puzzle with unknown conditional success probabilities. Again, combining a dynamic program for the discovery of intelligence pieces with passive learning of the encounter probabilities improved on the myopic outcomes.

Chapter 5

Conclusions and Further Research

In this chapter we conclude the thesis by providing a summarised overview of achievements and directions for potential further research.

5.1 Summary and Conclusions

In this section we summarise the research presented in this thesis, and highlight its achievements. In both areas we considered, the focus was on efficient collection of pieces of intelligence called tips, which originated from a variety of intelligence sources.

5.1.1 The Intelligence Problem

In Chapter 3 we examined the intelligence problem using a Bayesian multi-armed bandit framework. The novelty of our approach was to explicitly model the random time required to evaluate (i.e. collect, process and analyse) a tip, and to delay receipt of the associated reward until the evaluation is completed. The value of intelligence was treated as binary in nature, a tip was either relevant or not. After investigating a series of modelling assumptions in Section 3.2, we chose an undiscounted infinite horizon formulation where each arm of the MAB is characterised by their expected reward rate. For this definition of the intelligence problem we considered four models, two in which the probability that a tip is relevant is independent of its evaluation time, and two in which a logistic link function is used to model the time dependence of the relevance probability.

We studied the two independent models, IM1 and IM2 in Section 3.3. To direct an intelligence team in selecting sources we devised 5 policies. These include a simple myopic rule that continues the arm with the highest expected reward rate, the Bather index, which is a randomised index that adds diminishing perturbations to the myopic rule, and the EGGI, a heuristic based on a novel generalisation of the Gittins index, which allows for deterministic non-unit inter-decision times. The development of the generalised Gittins index used in the EGGI is an important methodological contribution. While we use it to construct a heuristic rule, it provides an optimal policy for a simplified intelligence problem with deterministic evaluation times. We have also adapted the knowledge gradient approach to the specifics of the intelligence problem, and is used to form the basis of the CKGI, a new innovative index policy. It marries knowledge gradient type expected rewards and calibrating with regard to a standard bandit, an approach it shares with the Gittins index. We examined the performance of all policies using a regret-type measure, and found that the CKGI consistently performs well with respect to that measure. While in general the EGGI did not perform exceptionally, in the presence of large uncertainties around the evaluation times it achieved the best results. Even though in our numerical experiments the Bather index performed well, the parameters governing the random perturbations required extensive tuning and knowledge of the dataset, which made it impractical.

In Section 3.4 we explored the consequences of time dependent relevance probabilities under two models, DM1 and DM2. We found the expected reward rates of the arms were only possible to obtain numerically, making even the myopic and the Bather index policies challenging to apply. To reduce the computational burden of tuning the parameters of the Bather index we devised an approximate tuning method, which we found to perform similarly to the fully tuned Bather index. While the EGGI could not be applied due to incompatible assumptions, the limitations on applying the knowledge gradient and the CKGI were due to the expected reward rate having a non-monotonic distribution under both DM1 and DM2. In the absence of closed form formulas this posed a computational challenge, and ultimately prevented us from applying these policies to the dependent models.

Overall, in Chapter 3 we have developed promising solution approaches to accommodate the novel features of the intelligence problem with time independent relevance probabilities, but had to conclude that in models that include dependencies devising effective and practicable solution strategies poses a challenge.

5.1.2 The Intelligence Puzzle

In Chapter 4, we defined for the first time a new intelligence collection model called the intelligence puzzle, which focused on investigating a single intelligence question. It takes a novel approach to intelligence by allowing some tips to be fundamentally different to others, assigning an inherent type to all tips. In such case, for every source the probabilities of encountering different types of tips had to be specified, as well as for each type and source combination the probability that they evaluate to magic bullets. The objective of the intelligence puzzle was defined as observing a single relevant tip (which we termed a magic bullet) of each available type and in doing so acquire the least amount of cost. In our investigations, costs were defined in a way that minimising them was equivalent to minimising the time taken to complete the intelligence puzzle. We have found that in such a model the state of the decision problem is not simply the collection of states associated with the sources, and therefore the intelligence puzzle cannot be treated as a bandit problem. Instead, we modelled all variants of the intelligence puzzle as a semi-Markov decision problem.

In Section 4.2 we considered the intelligence puzzle in its most basic form, where all parameters that characterise a source are known. The optimal policy was found using dynamic programming. We have shown that when magic bullets must be discovered in a set order, a myopic approach is optimal, and that under some limited circumstances the order itself does not influence the completion time. However, in general a myopic policy was found not to be optimal, apart from a few specific cases. The policies we studied were the optimal policy, a myopic policy that selects the source with the lowest expected cost of discovering a magic bullet, and as compromise between these two a one-step look-ahead policy. We have also included two decision rules based on our findings for ordered discoveries. When examining the relative sub-optimality of these policies, we have found that policies based on ordered discoveries performed poorly. Through examining the the time taken to discover each magic bullet, we traced back the sub-optimality of the myopic and to a lesser extent the one-step look-ahead policy to the time taken to discover the last magic bullet. While short sighted policies were quick to discover magic bullets early on, lack of forward planning often left a type of magic bullet which was hard to find to discover last, which lead to a long search.

In Section 4.3 we extended the intelligence puzzle to allow the probability that a tip evaluates to a magic bullet to be learnt through repeated observations. We found that only requiring a single magic bullet per intelligence type meant learning could only take place through negative feedback, which made dynamic programming unsuitable. On the other hand, in Section 4.4 we found the distribution of encounter probabilities to be straightforward to learn, and we expect the dynamic programming approach developed to provide a good approximation of the optimal policy. For both extensions of the intelligence puzzle we devised deterministically equivalent analogues of the myopic, one-step look-ahead and optimal policies developed for the basic intelligence puzzle, and compared their performance to that of a super-optimal policy.

Our most important accomplishment in Chapter 4 was to define the intelligence puzzle and for the first time develop insights into the mechanics of such an intelligence collection model. We have also demonstrated the challenges that come with allowing some aspects of the problem to be initially unknown, especially when learning is only possible through negative feedback.

5.2 Further Research

In this section we discuss some avenues of future research that may be of interest, which we organise into three sections. First we discuss topics relating to Chapter 3 in Section 5.2.1, followed by Chapter 4 in Section 5.2.2.

5.2.1 Intelligence Problem

An immediate avenue of further research would be to reinvestigate the intelligence problem in its current form, but in terms of the alternative metric, $\mathbb{E}_s\left[P\right]/\mathbb{E}_s\left[T\right]$. Section 3.5.1 briefly discusses our expectations regarding this metric. One may also expand the range of methodologies considered for the intelligence problem; both Thompson sampling (Thompson, 1933) and UCB (Auer et al., 2002) are well established for multi-armed bandits. A similarly conspicuous possibility is to investigate a different, perhaps better behaved dependence structure than that of Section 3.4. Indeed, there is no inherent reason in the intelligence collection and analysis process why the relevance probability should depend on the evaluation times and not the other way around. While these are only a minor modifications to our work, there are multiple, more ambitious extensions.

The Intelligence Problem with Discounted Rewards

As intelligence is usually collected with a purpose, be it in a police, military, anti-terror or any other setting, preference towards receiving relevant tips sooner rather than later is a plausible assumption. Such urgency can be modelled by discounting future rewards. We have previously mentioned such a model in Section 3.2 as IP2. The presence of discounting puts further focus on the semi-Markov nature of the problem. However, in exchange for the increased difficulty the model would better represent the environment in which intelligence collection takes place.

Interrupted Evaluations

One of the main assumptions in our model is that the intelligence team may not renege on a tip: once it is selected for evaluation the team must complete the evaluation before being able to move on to evaluating the next tip. We propose to investigate the impact of undoing this assumption. In that case the intelligence team is able to switch sources by abandoning an evaluation process before it concludes, and forgoing any potential reward from it. The intelligence team must weigh its options carefully: by reneging on a tip they can avoid spending a long time on a single piece of intelligence, yet, if an evaluation is interrupted, the knowledge gained from the partial observation is limited. The outcome of the interrupted evaluation is not observed, and the distributions of the evaluation times are only updated with right-censored observations. While similar mechanisms have been studied in the context of knapsack constrained bandits (Abernethy et al., 2016; Cayci et al., 2020b), and threshold bandits (Cayci et al., 2020a), these take place over a finite horizon.

Non-Binary Value of Intelligence

Another assumption used in constructing the intelligence problem is that of binary outcomes: a tip is either relevant or not. While much of the research on intelligence treats them as such, binary outcomes can not reflect nuanced differences between the values of tips. While Marshall (2016) assigned non-binary values to tips at the processing step, in the end had to make a binary judgement on the tip being relevant or not. Modelling the value of a tip as a continuous variable would also open a wider range of possibilities to construct dependence between the evaluation time and the value of the tip.

Multiple Intelligence Teams

Models with multiple analysts are present in the intelligence literature, as well as multi-armed bandits where more than one arm may be selected at every decision epoch. While we believe in terms of the intelligence problem this is not a trivial extension, we include some initial thoughts on the effects of multiple intelligence teams working in parallel. The presence of multiple intelligence teams in the intelligence problem modifies when decision epochs take place. Furthermore, the sojourn times following a given decision epoch would depend on actions chosen at previous sojourn times: tips which are still under evaluation, chosen by a different intelligence team may complete before the one whose evaluation starts at the current decision epoch. Consequently, due to the presence of delayed feedback the intelligence teams must make their selections on incomplete information.

Changing Availability of Sources

Sources of intelligence are not necessarily permanent. There are many reasons a source can stop producing intelligence: satellites getting decommissioned, monitored communications moving to secure channels, the cover of informants getting blown, just to name a few. Sources may also be whittled down due to the intelligence team needing to focus on a decreasing number of sources as time goes on. However, new sources can also become available over time. Examples of such are a decommissioned piece of equipment getting replaced by more advanced technologies, placing wire-taps, or a new agent going under-cover. As these changes may occur both over strategic and tactical time-scales, extending both the intelligence problem and the intelligence puzzle is well motivated. Changes in the availability of the sources are equivalent to the multi-armed bandit loosing or acquiring new arms. One difficulty we foresee in such a model, depending on how fast arms are acquired and dropped, is that knowledge of the sources can be limited.

5.2.2 Intelligence Puzzle

As Chapter 4 demonstrates, the intelligence puzzle is a complex problem that we have only scratched the surface of, and therefore there are many avenues of meaningful further research even within the current context of the intelligence puzzle. In general, more could be done to see the effects of restricting the encounter and success probabilities to specific values or ranges and how such constraints change the interactions between them.

Learning Policies for the Intelligence Puzzle with Unknown Encounter Probabilities

While the intelligence puzzle with unknown encounter probabilities is not limited in learning the same way the intelligence puzzle is with unknown success probabilities, the only policies we implemented only learnt about the distribution of the encounter probabilities in a passive manner through Bayesian updating. The natural next step for the intelligence puzzle in Section 4.4 is to develop and implement policies in which learning takes center-stage. In Section 4.4.3 a dynamic programming policy was constructed based on both the physical and the truncated knowledge state of the decision process. With a sufficiently permissive truncation point such a policy is a reasonable approximation of the optimal policy. While feasible for the smallest instances, due to the large state space of the intelligence puzzle such a dynamic programming approach is not practical and approximate approaches should be explored.

Source Dependent Conditional Success Probabilities

As we have seen in Section 4.3.3, the fact that only one magic bullet is needed of any one type severely limits our ability to learn the distribution of the conditional success probabilities when they are unknown to start with. To mitigate that, we propose a change to the intelligence puzzle. We require the conditional success probabilities to not differ between the different types from the same source, i.e. making it a parameter associated with the source instead of the source-type combination. When the success probabilities are characteristics of the sources, there is a limited but present opportunity to learn from the successes as well. Furthermore, this gives the decision maker more direct control of the learning process. When every source-type combination has a different conditional success probability, the decision maker can choose the source to evaluate a tip from, but the encountered type and therefore the success probability they learn about is random. With common success probabilities across all types, picking a source to evaluate a tip from directly picks the success probability to learn about. We believe it to be an interesting avenue for further research to examine the intelligence puzzle that is structured in the above manner as it results in an improved ability to learn, with the benefit most apparent in puzzles where $\mathcal{A}$ is small in comparison to $\mathcal{M}$.

The most important consequence of such a common conditional success probability is the increased ability to learn in the intelligence puzzle with unknown conditional success probabilities. In the case of either the basic intelligence puzzle or the intelligence puzzle with unknown encounter probabilities opting for common p^a's is expected to have limited impact.

Modifying the Cost Rates

While all three versions of the intelligence puzzle was described with a general physical state dependent cost rate $w(\boldsymbol{s})$, in the numerical experiments it was set to $w(\boldsymbol{s}) = 1$ for all $\boldsymbol{s} \neq \mathbf{0}_{\mathcal{M}}$. While this is a very intuitive cost rate and results in an intelligence puzzle which aims to minimise the time to completion, it results in a loss of generality.

Indeed, we can imagine some scenarios in which different states have different cost rates. For example, if some types of information are considered more important then others, states in which those remain undiscovered might accrue cost at a higher rate than those in which they have been discovered. If not the status of a specific type, the number of magic bullets that have been discovered can have an influence on the reward rates. A cost rate that decreases as more types are discovered can be used to capture an intelligence puzzle where the emphasis is on obtaining as many magic bullets as early as possible instead of completing the puzzle.

While it may initially read like a change in the objective, modified cost rates can also be used to model an intelligence puzzle where not all available types of magic bullets are required for completion. Where completion of the puzzle requires any subset of size $\mathcal{M}'$ out of the $\mathcal{M}$ available types, modelling it is as straightforward as setting the $w(\boldsymbol{s})$ of all states associated with stages $g > \mathcal{M}'$ to 0.

As we can see, taking advantage of the flexibility of the model and moving away from unit cost rates can result in many new flavours to the intelligence puzzle, which would be interesting to explore.

A Different View of Intelligence Types

Our contribution in Chapter 4 is not simply developing the intelligence puzzle, but also introducing the idea that pieces of information may have an inherent type. If we use this type to represent an intelligence question, or a terror-plot queuing for interdiction, one can come up with a model that straddles the microscopic and macroscopic viewpoint of the intelligence cycle.

Imagine a series of intelligence questions, to which we refer to as plots queueing to be investigated. To complete the investigation of a plot, a known number of relevant tips must be found for it. However, the sources produce intelligence regarding all plots in the queue, meanwhile relevant tips can only be found for those which are attended by an intelligence team. This assumption reflects that the intelligence team(s) can only find the answer to questions they know of. Over time the intelligence team(s) process all plots waiting at the start.

If every plot only needs one tip to solve it, the formulation of the above problem is identical to that of the intelligence puzzle. In the form described above, even with multiple tips required for interdiction, we may view the problem as an extension of the intelligence puzzle. We can extend both the intelligence puzzle and the queuing plots interpretation with multiple intelligence teams or analysts. However, moving away from the concept of the intelligence puzzle may motivate features unrealistic in the puzzle setting. For example, one might consider a model where plots can not only leave the queue, but also enter it. Furthermore, many features of Kaplan (2010), such as plots expiring could be added to enrich the model. Some fascinating research questions arise from this, including ”Under which conditions will the intelligence team(s) be overwhelmed?”, ”How should the intelligence team(s) select sources to avoid that?”, and ”How should sources be selected to maximise the interdiction rate?”.

Game Theoretical Considerations

As discussed in Section 5.2.1, sources may become available and unavailable for a variety of reasons. In terms of the intelligence puzzle, we find the option of sources dropping out more likely than new ones appearing. One scenario we find fitting with the problem is where upon discovery of a magic bullet, a source stops producing intelligence. By giving control of which source is eliminated to the adversary that is surveilled during the course of the puzzle, we can define a game theoretical version of the basic intelligence puzzle. The goal of the intelligence team is to counter the adversary by completing the intelligence puzzle as quickly as possible, while the adversary eliminates sources one-by one to maximise the time they are able to operate.

Appendix A

Appendix to Chapter 3

A.1 Proving of Theorem 3.3.1

In this appendix we prove Theorem 3.3.1, given in Section 3.3.2, which was as follows. *Given that P^a and T^a are independent, $\mathcal{T}_x^a$ is defined by a threshold t_x^a;*

$$\mathcal{T}_x^a = \{t; 0 < t \leq t_x^a\},$$

so that $t \in \mathcal{T}_x^a$ if and only if $0 < t \leq t_x^a$.

The above can be proven using stochastic dominance. As the theory required to prove Theorem 3.3.1 are not otherwise relevant to this thesis, we introduce them as part of the proof. We follow it up by construct the proof of Theorem 3.3.1 in a general setting.

A.1.1 Required theorems and definitions

We use a general Bayesian model where X is a sample space and $P = \{P_\theta\}_{\theta\in\Theta}$ is a family of probability distributions on X. Let P be a prior distribution on sample space Θ. Then X is a random sample with the values $x \in X$, $X|Z = \theta \sim P_\theta$, where Z is a Θ-valued random variable, $Z \sim P$.

Denote by $F(\cdot|\theta)$ the cdf of P_θ, and by $f(\cdot|\theta)$ the pdf of that distribution. Similarly, denote the cdf of the prior and posterior distributions of θ by $P(\cdot)$ and $P(\cdot|X = x)$, and their pdf by $\pi(\cdot)$ and $\pi(\cdot|X = x)$ respectively, where $X \sim P_\theta$. The posterior predictive cdf and pdf of the next observation are denoted as $F_P^\pi(\cdot|z_n)$ and $f_P^\pi(\cdot|z_n)$, where z_n denotes the observed data, and π as seen in the superscript indicates what the predictive distribution is with respect to.

Theorem A.1.1. *FSD-theorem (Wolfstetter, 1996; Müller and Stoyan, 2002)*
X is unanimously preferred to Y by all agents with monotone increasing utility functions iff $X \geq_{FSD} Y$.

$$X \geq_{\,FSD} Y \Longleftrightarrow \mathbb{E}\left[U(X)\right] \geq \mathbb{E}\left[U(Y)\right] \qquad \forall U \in \mathcal{U},$$

where $\mathcal{U}$ is the set of utility functions that are monotone increasing.
The preference ranking is reversed if the utility functions are decreasing.

Definition A.1.2. *(Müller and Stoyan, 2002; Meczarski, 2015)*
The random variable X is said to be larger than Y in likelihood ratio order ($X \geq_{LR} Y$) if X and Y have densities $f_X(.)$ and $f_Y(.)$ such that

$$\frac{f_X(t)}{f_Y(t)} \leq \frac{f_X(t')}{f_Y(t')} \qquad \forall t \leq t'.$$

Equivalently, the ratio $\frac{f_X(t)}{f_Y(t)}$ is an increasing function of t.

Theorem A.1.3. *(Müller and Stoyan, 2002)*
If $X \geq_{\,LR} Y$ then $X \geq_{\,FSD} Y$.
The likelihood ratio order is sufficient but not necessary condition of first order stochastic dominance.(The likelihood ratio order the stronger ordering.)

Theorem A.1.4. *(Müller and Stoyan, 2002)*
Assume that $X \geq_{\,LR} Y$ and g is an increasing function. Then $g(X) \geq_{\,LR} g(Y)$.
Proof found in Müller and Stoyan (2002) can also be used to show that the ordering is reversed when g is a decreasing function.

Theorem A.1.5. *LR order of posteriors (Meczarski, 2015)*
If the distribution of X is increasing with respect to the likelihood ratio order in θ, then

the conditional distribution of θ under $X = x$ is increasing in x with respect to the likelihood ratio order.
Equivalently,

$$\mathrm{P}_\theta \leq_{LR} \mathrm{P}_{\theta'} \text{ for } \theta \leq \theta' \Longrightarrow P(\cdot|X = x) \leq_{LR} P(\cdot|X = x') \text{ for } x \leq x',$$

where P_θ is a probability distribution of family P with parameter θ and $P(\cdot|X = x)$ is the posterior distribution after observing $X = x$.

Theorem A.1.6. *LR order of predictive distributions (Meczarski, 2015)*
If for distributions of the predicted variable Y we have

$$F(\cdot \mid \theta, z_n) \leq_{LR} F(\cdot \mid \theta', z_n), \qquad \forall z_n, \theta \leq \theta', \theta, \theta \in \Theta$$

and if $P_1 \leq_{LR} P_2$ then the likelihood ratio order transfers to the predictive distributions so that

$$F_P^{\pi_1}(\cdot \mid z_n) \leq_{LR} F_P^{\pi_2}(\cdot \mid z_n),$$

where $F_P^{\pi_i}(\cdot \mid z_n)$ is the cdf of the predictive density function.

A.1.2 Proof of Theorem 3.3.1

We wish to show that $\mathbb{E}\left[\frac{1}{T} \mid \boldsymbol{t}, t\right]$ is decreasing in t.

Proof. Let us assume that the distribution of T is increasing with respect to the likelihood ratio order in θ, so that

$$P_\theta \leq_{\mathrm{LR}} P_{\theta'} \qquad \forall \theta \leq \theta'.$$

This assumption can be checked for different families of probability distributions using Definition A.1.2. Theorem A.1.5 states that the likelihood ratio order of sample distributions are preserved by posterior distributions, therefore the posterior distribution that results from observation of $T = t'$ is larger than the posterior resulting from $T = t$ with respect to the likelihood ratio order for all $t' \geq t$.

$$P_\theta \leq_{\mathrm{LR}} P_{\theta'} \;\; \forall \theta \leq \theta' \quad \Longrightarrow \quad P(\cdot|t) \leq_{\mathrm{LR}} P(\cdot|t') \;\; \forall t \leq t'$$

Since the outcomes of T are IID and only depend on the past observations $\boldsymbol{t}$ through θ we have

$$F(\cdot|\theta,\boldsymbol{t}) \leq_{\mathrm{LR}} F(\cdot|\theta',\boldsymbol{t}) \equiv F(\cdot|\theta) \leq_{\mathrm{LR}} F(\cdot|\theta') \equiv P_\theta \leq_{\mathrm{LR}} P_{\theta'} \qquad \forall \theta \leq \theta', \theta, \theta' \in \Theta.$$

Using Theorem A.1.6 we can transfer the likelihood ratio order from the posterior to the predictive distributions.

$$\begin{array}{lcr} P(\cdot|t) \leq_{\mathrm{LR}} P(\cdot|t') \quad \forall t \leq t' & & \\ \text{together with} & \Longrightarrow & F_P^{\pi_t}(\cdot|\boldsymbol{t}) \leq_{\mathrm{LR}} F_P^{\pi_{t'}}(\cdot|\boldsymbol{t}). \\ P_\theta \leq_{\mathrm{LR}} P_{\theta'} \quad \forall \theta \leq \theta' & & \end{array}$$

Likelihood ratio ordering implies first order stochastic dominance as per Theorem A.1.3.

$$F_P^{\pi_t}(\cdot|\boldsymbol{t}) \leq_{\mathrm{LR}} F_P^{\pi_{t'}}(\cdot|\boldsymbol{t}) \Longrightarrow F_P^{\pi_t}(\cdot|\boldsymbol{t}) \leq_{\mathrm{FSD}} F_P^{\pi_{t'}}(\cdot|\boldsymbol{t})$$

The utility function $U(T) = 1/T$ is a monotone decreasing utility function, therefore making use of Theorem A.1.1 we get

$$\mathbb{E}\left[\frac{1}{T} \mid \boldsymbol{t}, t\right] \geq \mathbb{E}\left[\frac{1}{T} \mid \boldsymbol{t}, t'\right] \qquad \forall t \leq t',$$

meaning that $\mathbb{E}\left[\frac{1}{T} \mid \boldsymbol{t}, t\right]$ is decreasing in t, where the expectation is taken with respect to the posterior predictive distribution of T.

Since $\mathbb{E}\left[\frac{1}{T} \mid \boldsymbol{t}, t\right]$ is decreasing in t, there must exists a $t^{\text{threshold}}$ for which this expectation equals a given positive constant denoted as C.

Then

$$\begin{aligned} \mathbb{E}\left[\frac{1}{T} \mid \boldsymbol{t}, t\right] &\geq C && \forall t \leq t^{\text{threshold}} \\ \mathbb{E}\left[\frac{1}{T} \mid \boldsymbol{t}, t\right] &< C && \forall t > t^{\text{threshold}}. \end{aligned}$$

The value of t where the equality holds is the threshold. □

In the context of the knowledge gradient that threshold t_x^a is defined as

$$\mathbb{E}\left[\frac{1}{T^a} \mid \boldsymbol{t}^a, t\right] = C = \frac{1}{\mathbb{E}\left[P^a \mid \boldsymbol{x}^a, x\right]} \max_{a' \neq a} \mathbb{E}\left[\frac{P^{a'}}{T^{a'}} \mid \boldsymbol{x}^{a'}, \boldsymbol{t}^{a'}\right].$$

A.2 Exploration parameters of the Bather index

$\mathcal{A}$	2		5	
$\mathcal{H}$	b_1	b_2	b_1	b_2
100	0.8	0.6	0.9	0.7
200	0.8	0.7	0.8	0.6
500	0.8	0.8	0.9	0.9
1000	0.8	0.8	0.9	0.9
2000	0.8	0.9	0.8	0.9

Table A.2.1: Exploration parameters of the Bather index for IM1, with $t^{\min} = 0.1$.

s^2	0.1		0.2		0.5		1.0		2.0	
$\mathcal{H}$	b_1	b_2	b_1	b_2	b_1	b_2	b_1	b_2	b_1	b_2
100	2.0	0.2	1.6	0.2	1.8	0.3	1.3	0.4	1.4	0.5
200	1.8	0.3	2.0	0.3	1.7	0.4	1.4	0.5	2.0	0.4
500	0.8	0.5	2.0	0.4	1.7	0.5	1.1	0.6	1.6	0.6
1000	1.3	0.5	0.9	0.6	0.6	0.7	1.0	0.7	1.8	0.7

Table A.2.2: Exploration parameters of the Bather index for IM2, with $\mathcal{A} = 2$.

s^2	0.1		0.2		0.5		1.0		2.0	
$\mathcal{H}$	b_1	b_2	b_1	b_2	b_1	b_2	b_1	b_2	b_1	b_2
100	1.2	0.3	1.5	0.3	1.7	0.3	2.0	0.3	2.0	0.3
200	1.7	0.3	1.9	0.3	1.5	0.4	1.0	0.5	1.9	0.5
500	1.4	0.4	1.6	0.4	1.1	0.5	1.7	0.5	1.6	0.6
1000	0.9	0.6	0.9	0.6	0.8	0.7	1.1	0.7	1.3	0.7

Table A.2.3: Exploration parameters of the Bather index for IM2, with $\mathcal{A} = 5$.

A.3 The Adjusted Relative Regret

The adjusted relative regret is a now defunct regret measure, which we only mention here for completeness of description of work carried out. It was defined as a second measure in addition to the relative regret in (3.3.40). In addition tot he super-optimal policy, it takes into account a worst-case policy, which is the result of an omniscient decision maker identifying and playing the worst arm in all instances:

$$\mathcal{R}^{\pi}_{\text{AR}} = \frac{\text{Reward}^{\pi^{\text{SO}}} - \text{Reward}^{\pi}}{\left(\text{Reward}^{\pi^{\text{SO}}} - \text{Reward}^{\pi^{\text{WC}}}\right)}, \tag{A.3.1}$$

where π^{WC} denotes the worst-case policy.

Regrets calculated this way proved to be very similar to those calculated via (3.3.40) without providing additional insight, and was therefore deemed unnecessary to discuss in the main body of the thesis.

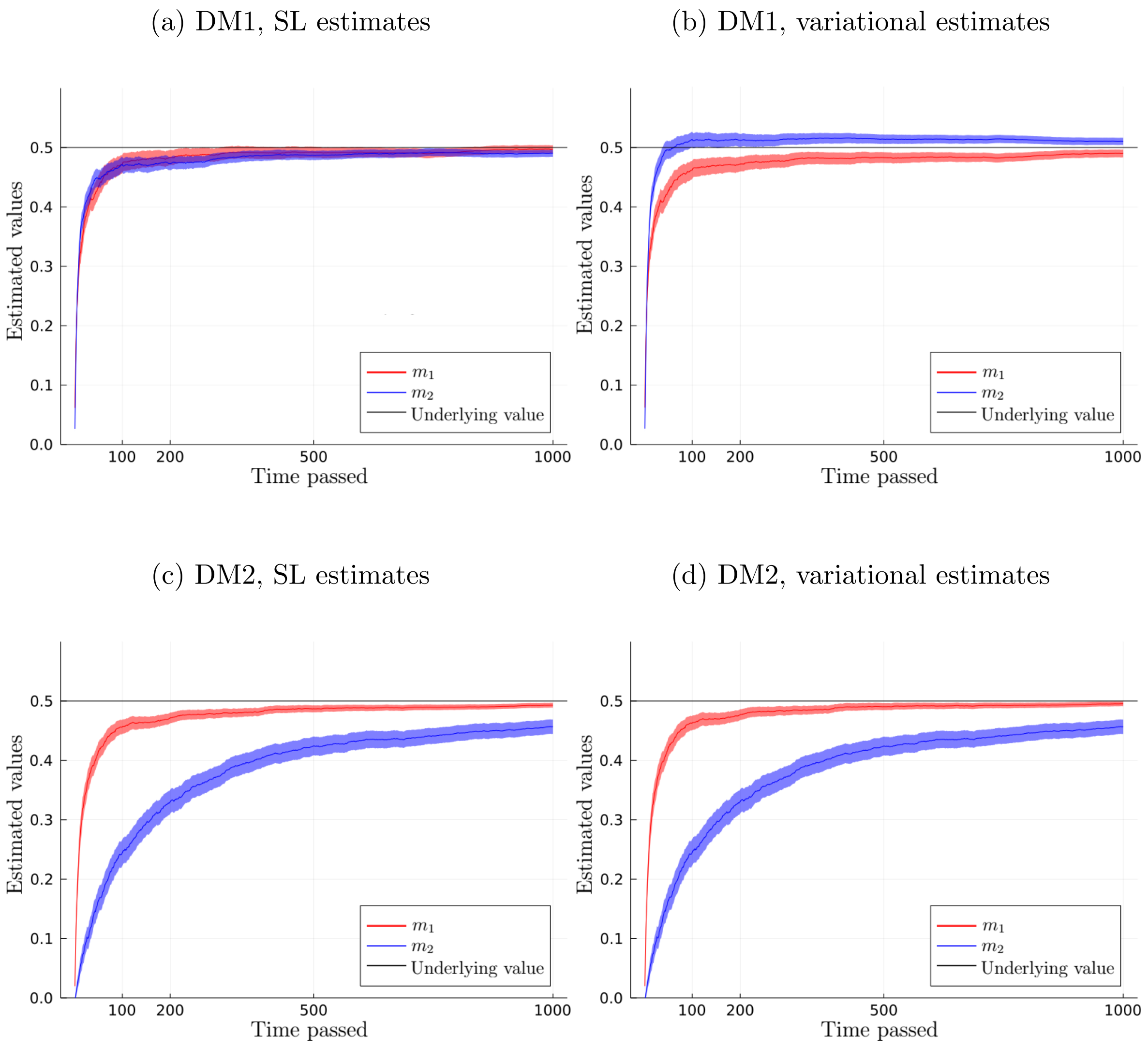
A.4 Issues with the Approximate Bayesian Inference

As discussed in Section 3.4.4, both types of regrets were seen to initially decrease then increase again as the horizon was increased. This is contrary to expectations, as a longer horizon means more observations and a more accurate estimate of the parameters we wish to learn about.

We investigated how the estimates of B_1 and B_2, given by m_1 and m_2 (components of $\boldsymbol{m}$) evolve as time passes. To be able to focus on learning, the decision making element of the problem was removed with only a single arm at a time played till the horizon. 1000 such arms, all with $B_1 = 0.5$ and $B_2 = 0.5$ were simulated. The mean estimated values of B_1 and B_2, along with bands representing the 95% confidence interval can be seen in Figure A.4.1. For both models, the average estimates show a quick initial shift towards the true value, explaining the initially decreasing regret. However, even the average estimates stop short the true values, potentially leading to mis-identification of the best arm and an increased regrets. While the mean of the m_1s is similar across the two models, the mean of the m_2s is much further from the true value of B_2 under DM2 compared to DM1. This could explain why the regret increased earlier and at a faster rate for DM2.

A much smaller number of these instances were also individually plotted. The individual series of estimates in Figure A.4.2 confirm what we suspected from Figure A.4.1; the estimated values of parameters B_1 and B_2 are not accurate and might not converge to their true values in the time-frame/number of observations we are interested in. Even more, the estimates of B_1 and B_2 cover a wide range of values.

Without accurate estimates of the parameters, the expected reward rate of the arms will not be accurate either. This increases the chance of picking, and sticking to a sub-optimal arm, leading to high regrets.

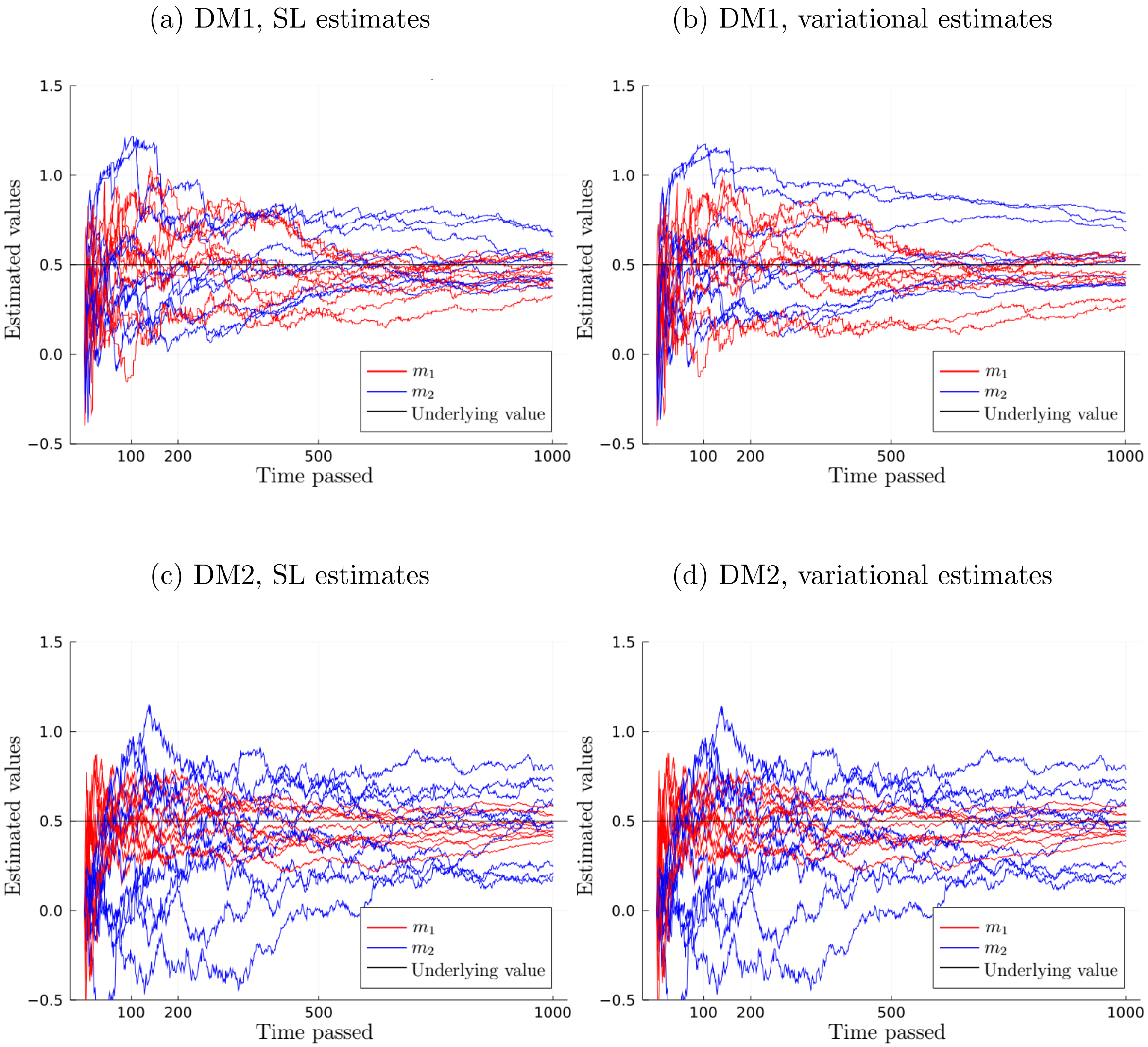

(a) DM1, SL estimates

(b) DM1, variational estimates

(c) DM2, SL estimates

(d) DM2, variational estimates

Figure A.4.1: Mean and 95% confidence interval of 1000 estimated values of B_1 and B_2. It is clear that the estimates of B_2 are further away from the underlying value for DM2 than they are for DM1.


(a) DM1, SL estimates

(b) DM1, variational estimates

(c) DM2, SL estimates

(d) DM2, variational estimates

Figure A.4.2: Estimated values of B_1 and B_2 for 10 instances, where the underlying value of both is 0.5. There are large differences between individual estimates, especially those of B_2.

Appendix B

Appendix to Chapter 4

B.1 Calculations Regarding the Optimality of the Myopic Policy

B.1.1 General condition for $\mathcal{A} = 2$, $\mathcal{M} = 2$

As discussed in Section 4.2.2, the myopic policy is optimal if at every decision epoch it assigns the same action as the optimal policy. Here we derive the necessary condition of myopic optimality of the $\mathcal{A} = 2$, $\mathcal{M} = 2$ case with $w(\boldsymbol{s} \neq \mathbf{1}_{\mathcal{M}}) = 1$, $w(\mathbf{1}_{\mathcal{M}}) = 0$.

The optimal policy assigns action 1 if the value function associated with said action is lower than that of action 2. It may also pick action 1 if the value functions are equal and the optimal policy is indifferent between the options. It has been established in Section 4.2.2 that the myopic policy is optimal in the last stage of the puzzle which includes states $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$, and therefore here we focus on state $\boldsymbol{s}_3$. The condition for the optimal policy choosing source 1 in state $\boldsymbol{s}_3$ is (as it would be in any other state)

$$\mathbb{V}^1(\boldsymbol{s}_3) \leq \mathbb{V}^2(\boldsymbol{s}_3).$$

Using the expanded form of $\mathbb{V}^a(\boldsymbol{s}_3)$ from (4.2.21) we can rewrite the above as

$$\begin{aligned}&\mathbb{E}\left[\mathcal{T}^1(\boldsymbol{s}_3)\right]+\mathbb{P}(\boldsymbol{s}_1 \mid \boldsymbol{s}_3,1)\,\mathbb{V}(\boldsymbol{s}_1)+(1-\mathbb{P}(\boldsymbol{s}_1 \mid \boldsymbol{s}_3,1))\,\mathbb{V}(\boldsymbol{s}_2)\\ &\leq \mathbb{E}\left[\mathcal{T}^2(\boldsymbol{s}_3)\right]+\mathbb{P}(\boldsymbol{s}_1 \mid \boldsymbol{s}_3,2)\,\mathbb{V}(\boldsymbol{s}_1)+(1-\mathbb{P}(\boldsymbol{s}_1 \mid \boldsymbol{s}_3,2))\,\mathbb{V}(\boldsymbol{s}_2)\end{aligned}$$

$$\begin{aligned}\mathbb{E}\left[\mathcal{T}^1(\boldsymbol{s}_3)\right]-\mathbb{E}\left[\mathcal{T}^2(\boldsymbol{s}_3)\right] &\leq \mathbb{P}(\boldsymbol{s}_1 \mid \boldsymbol{s}_3,2)\,\mathbb{V}(\boldsymbol{s}_1)+(1-\mathbb{P}(\boldsymbol{s}_1 \mid \boldsymbol{s}_3,2))\,\mathbb{V}(\boldsymbol{s}_2)\\ &\quad-\mathbb{P}(\boldsymbol{s}_1 \mid \boldsymbol{s}_3,1)\,\mathbb{V}(\boldsymbol{s}_1)-(1-\mathbb{P}(\boldsymbol{s}_1 \mid \boldsymbol{s}_3,1))\,\mathbb{V}(\boldsymbol{s}_2)\\ &\leq \mathbb{V}(\boldsymbol{s}_1)\,(\mathbb{P}(\boldsymbol{s}_1 \mid \boldsymbol{s}_3,2)-\mathbb{P}(\boldsymbol{s}_1 \mid \boldsymbol{s}_3,1))\\ &\quad+\mathbb{V}(\boldsymbol{s}_2)\,(\mathbb{P}(\boldsymbol{s}_1 \mid \boldsymbol{s}_3,1)-\mathbb{P}(\boldsymbol{s}_1 \mid \boldsymbol{s}_3,2)).\end{aligned}$$

By letting $\Delta\mathbb{P}(\boldsymbol{s}_1 \mid \boldsymbol{s}_3)$ denote $\mathbb{P}(\boldsymbol{s}_1 \mid \boldsymbol{s}_3,1)-\mathbb{P}(\boldsymbol{s}_1 \mid \boldsymbol{s}_3,2)$ we get

$$\mathbb{E}\left[\mathcal{T}^1(\boldsymbol{s}_3)\right]-\mathbb{E}\left[\mathcal{T}^2(\boldsymbol{s}_3)\right] \leq \Delta\mathbb{P}(\boldsymbol{s}_1 \mid \boldsymbol{s}_3)\,(\mathbb{V}(\boldsymbol{s}_2)-\mathbb{V}(\boldsymbol{s}_1)). \tag{B.1.1}$$

The myopic policy assigns action 1 if

$$\mathbb{E}\left[\mathcal{T}^1(\boldsymbol{s}_3)\right] < \mathbb{E}\left[\mathcal{T}^2(\boldsymbol{s}_3)\right],$$

which means that

$$\mathbb{E}\left[\mathcal{T}^1(\boldsymbol{s}_3)\right]-\mathbb{E}\left[\mathcal{T}^2(\boldsymbol{s}_3)\right] < 0.$$

It may also assign action 1 when it is indifferent between the actions, however that is only an optimal assignment if the optimal policy is also indifferent, and therefore is a special case we will touch on later.

Combining the two conditions, we can divide by the left-hand-side of (B.1.1) to get

$$1 \geq \Delta\mathbb{P}(\boldsymbol{s}_1 \mid \boldsymbol{s}_3)\,\frac{(\mathbb{V}(\boldsymbol{s}_2)-\mathbb{V}(\boldsymbol{s}_1))}{\mathbb{E}\left[\mathcal{T}^1(\boldsymbol{s}_3)\right]-\mathbb{E}\left[\mathcal{T}^2(\boldsymbol{s}_3)\right]},$$

which, after expanding $\mathbb{V}(\boldsymbol{s}_1)$ and $\mathbb{V}(\boldsymbol{s}_2)$ according to (4.2.18) and (4.2.19), is identical to the condition proposed in (4.2.27)

$$1 \geq \ \Delta\mathbb{P}(\boldsymbol{s}_1 \mid \boldsymbol{s}_3)\ \ \frac{\min\limits_{a\in\mathcal{S}_\mathcal{A}}\mathbb{E}\left[\mathcal{T}^a(\boldsymbol{s}_1)\right]-\min\limits_{a\in\mathcal{S}_\mathcal{A}}\mathbb{E}\left[\mathcal{T}^a(\boldsymbol{s}_2)\right]}{\mathbb{E}\left[\mathcal{T}^2(\boldsymbol{s}_3)\right]-\mathbb{E}\left[\mathcal{T}^1(\boldsymbol{s}_3)\right]}.$$

The same may be obtained through considering the conditions required for both the optimal and the myopic policy to suggest action 2. In that case the relation in (B.1.1) is reversed, and the myopic policy only picks action 2 unambiguously if $\mathbb{E}\left[\mathcal{T}^1\left(\boldsymbol{s}_3\right)\right]-\mathbb{E}\left[\mathcal{T}^2\left(\boldsymbol{s}_3\right)\right]>0$, leading to the same condition for optimality.

As mentioned before, we have not yet considered the possibility that the myopic policy is indifferent to the action taken, in which case $\mathbb{E}\left[\mathcal{T}^1\left(\boldsymbol{s}_3\right)\right]=\mathbb{E}\left[\mathcal{T}^2\left(\boldsymbol{s}_3\right)\right]$ and would lead to a 0 in the denominator of the optimality condition. However, an indifferent myopic policy is only optimal if the optimal policy is also indifferent, requiring that both

$$\begin{aligned}\mathbb{E}\left[\mathcal{T}^1\left(\boldsymbol{s}_3\right)\right]-\mathbb{E}\left[\mathcal{T}^2\left(\boldsymbol{s}_3\right)\right]&=\Delta\mathbb{P}\left(\boldsymbol{s}_1 \mid \boldsymbol{s}_3\right)\left(\mathbb{V}(\boldsymbol{s}_2)-\mathbb{V}(\boldsymbol{s}_1)\right)\\ \mathbb{E}\left[\mathcal{T}^1\left(\boldsymbol{s}_3\right)\right]&=\mathbb{E}\left[\mathcal{T}^2\left(\boldsymbol{s}_3\right)\right]\end{aligned}$$

are satisfied, which combines to

$$0=\Delta\mathbb{P}\left(\boldsymbol{s}_1 \mid \boldsymbol{s}_3\right)\left(\mathbb{V}(\boldsymbol{s}_2)-\mathbb{V}(\boldsymbol{s}_1)\right).$$

Therefore an indifferent myopic policy is only optimal if the transition probabilities associated with the two actions are identical, or the states to which the process may transition to have equal values.

B.1.2 Simplified case with $p_m^a = p_m$

This section contains the calculations leading to the optimality conditions of the myopic policy for the first simplified case of the $\mathcal{A}=2$, $\mathcal{M}=2$ basic intelligence puzzle with $w(\boldsymbol{s}\neq\mathbf{1}_{\mathcal{M}})=1$, as shown in (4.2.29) and (4.2.30). Here the success probabilities of a given type is the same regardless of the source so that $p_1^1=p_1^2=p_1$ and $p_2^1=p_2^2=p_2$. The other restrictions on the parameters are $t_1^1=t_2^1=t_1^2=t_2^2=1$ and $c=1$. We start by separately computing the terms of (4.2.27). Using the definitions of $\mathbb{P}\left(\boldsymbol{s}_1 \mid \boldsymbol{s}_3, a\right)$ and $\mathbb{E}\left[\mathcal{T}^a\left(\boldsymbol{s}\right)\right]$ found in Section 4.2.1 and denoting q_1^1 as q^1 and q_1^2 as q^2 they simplify to

$$\begin{aligned}\Delta\mathbb{P}\left(\boldsymbol{s}_1 \mid \boldsymbol{s}_3\right) &= \mathbb{P}\left(\boldsymbol{s}_1 \mid \boldsymbol{s}_3, 1\right) - \mathbb{P}\left(\boldsymbol{s}_1 \mid \boldsymbol{s}_3, 2\right) \\ &= \frac{q^1 p_1^1}{q^1 p_1^1 + \left(1-q^1\right) p_2^1} - \frac{q^2 p_1^2}{q^2 p_1^2 + \left(1-q^2\right) p_2^2} \\ &= \frac{q^1 p_1}{q^1 p_1 + \left(1-q^1\right) p_2} - \frac{q^2 p_1}{q^2 p_1 + \left(1-q^2\right) p_2} \\ &= \frac{\left(q^1 p_1\right)\left(q^2 p_1 + \left(1-q^2\right) p_2\right) - \left(q^2 p_1\right)\left(q^1 p_1 + \left(1-q^1\right) p_2\right)}{\left(q^1 p_1 + \left(1-q^1\right) p_2\right)\left(q^2 p_1 + \left(1-q^2\right) p_2\right)} \\ &= \frac{p_1 p_2 \left(q^1 - q^2\right)}{\left(q^1 p_1 + \left(1-q^1\right) p_2\right)\left(q^2 p_1 + \left(1-q^2\right) p_2\right)},\end{aligned}$$

$$\begin{aligned}\mathbb{E}\left[\mathcal{T}^2\left(\boldsymbol{s}_3\right)\right] - \mathbb{E}\left[\mathcal{T}^1\left(\boldsymbol{s}_3\right)\right] &= \frac{q^2 t_1^2 + \left(1-q^2\right) t_2^2}{q^2 p_1^2 + \left(1-q^2\right) p_2^2} - \frac{q^1 t_1^1 + \left(1-q^1\right) t_2^1}{q^1 p_1^1 + \left(1-q^1\right) p_2^1} \\ &= \frac{1}{q^2 p_1 + \left(1-q^2\right) p_2} - \frac{1}{q^1 p_1 + \left(1-q^1\right) p_2} \\ &= \frac{q^1 p_1 + \left(1-q^1\right) p_2 - q^2 p_1 - \left(1-q^2\right) p_2}{\left(q^1 p_1 + \left(1-q^1\right) p_2\right)\left(q^2 p_1 + \left(1-q^2\right) p_2\right)} \\ &= \frac{\left(q^1 - q^2\right)\left(p_1 - p_2\right)}{\left(q^1 p_1 + \left(1-q^1\right) p_2\right)\left(q^2 p_1 + \left(1-q^2\right) p_2\right)},\end{aligned}$$

$$\begin{aligned}\min_a \mathbb{E}\left[\mathcal{T}^a(\boldsymbol{s}_1)\right] - \min_a \mathbb{E}\left[\mathcal{T}^a(\boldsymbol{s}_2)\right] &= \min_a \frac{q^a t_0^a / c + \left(1-q^a\right) t_1^a}{\left(1-q^a\right) p_2} - \min_a \frac{q^a t_0^a + \left(1-q^a\right) t_1^a / c}{q^a p_1} \\ &= \min_a \frac{1}{\left(1-q^a\right) p_2} - \min_a \frac{1}{q^a p_1}.\end{aligned}$$

Th expression for $\min_a \mathbb{E}\left[\mathcal{T}^a(\boldsymbol{s}_1)\right] - \min_a \mathbb{E}\left[\mathcal{T}^a(\boldsymbol{s}_2)\right]$ can be further simplified by considering two scenarios,

$$\min_a \mathbb{E}\left[\mathcal{T}^a(\boldsymbol{s}_1)\right] - \min_a \mathbb{E}\left[\mathcal{T}^a(\boldsymbol{s}_2)\right] = \begin{cases} \dfrac{q^1 p_1 - \left(1-q^2\right) p_2}{q^1 \left(1-q^2\right)\left(p_1 - p_2\right)} & \text{if } q^1 > q^2, \\[2ex] \dfrac{q^2 p_1 - \left(1-q^1\right) p_2}{q^2 \left(1-q^1\right)\left(p_1 - p_2\right)} & \text{if } q^1 < q^2, \end{cases}$$

Substituting back into (4.2.27), for $q^1 > q^2$ we get

$$\begin{aligned}
1 &\geq \frac{p_1 p_2 \left(q^1 - q^2\right)}{\left(q^1 p_1 + \left(1 - q^1\right) p_2\right)\left(q^2 p_1 + \left(1 - q^2\right) p_2\right)} \\
&\quad \times \frac{\left(q^1 p_1 + \left(1 - q^1\right) p_2\right)\left(q^2 p_1 + \left(1 - q^2\right) p_2\right)}{\left(q^1 - q^2\right)\left(p_1 - p_2\right)} \frac{q^1 p_1 - \left(1 - q^2\right) p_2}{q^1 \left(1 - q^2\right) p_1 p_2} \\
1 &> \frac{q^1 p_1 - \left(1 - q^2\right) p_2}{q^1 \left(1 - q^2\right)\left(p_1 - p_2\right)}.
\end{aligned}$$

From here we need to consider the cases $p_1 < p_2$ and $p_1 > p_2$ separately. For $p_1 < p_2$ the optimality condition is

$$\begin{aligned}
q^1 \left(1 - q^2\right)\left(p_1 - p_2\right) &< q^1 p_1 - \left(1 - q^2\right) p_2, \\
\frac{\left(1 - q^1\right)}{q^1} \frac{\left(1 - q^2\right)}{q^2} &< \frac{p_1}{p_2},
\end{aligned}$$

while for $p_1 > p_2$ it is

$$\begin{aligned}
q^1 \left(1 - q^2\right)\left(p_1 - p_2\right) &> q^1 p_1 - \left(1 - q^2\right) p_2 \\
\frac{\left(1 - q^1\right)}{q^1} \frac{\left(1 - q^2\right)}{q^2} &> \frac{p_1}{p_2}.
\end{aligned}$$

Similarly, for $q^1 < q^2$ we get

$$\begin{aligned}
1 &> \frac{p_1 p_2 \left(q^1 - q^2\right)}{\left(q^1 p_1 + \left(1 - q^1\right) p_2\right)\left(q^2 p_1 + \left(1 - q^2\right) p_2\right)} \\
&\quad \times \frac{\left(q^1 p_1 + \left(1 - q^1\right) p_2\right)\left(q^2 p_1 + \left(1 - q^2\right) p_2\right)}{\left(q^1 - q^2\right)\left(p_1 - p_2\right)} \frac{q^2 p_1 - \left(1 - q^1\right) p_2}{q^2 \left(1 - q^1\right) p_1 p_2} \\
1 &> \frac{q^2 p_1 - \left(1 - q^1\right) p_2}{q^2 \left(1 - q^1\right)\left(p_1 - p_2\right)},
\end{aligned}$$

which is split into the cases $p_1 < p_2$

$$\begin{aligned}
q^2 \left(1 - q^1\right)\left(p_1 - p_2\right) &< q^2 p_1 - \left(1 - q^1\right) p_2 \\
\frac{\left(1 - q^1\right)}{q^1} \frac{\left(1 - q^2\right)}{q^2} &< \frac{p_1}{p_2},
\end{aligned}$$

and $p_1 > p_2$

$$\begin{aligned}
q^2 \left(1 - q^1\right)\left(p_1 - p_2\right) &> q^2 p_1 - \left(1 - q^1\right) p_2 \\
\frac{\left(1 - q^1\right)}{q^1} \frac{\left(1 - q^2\right)}{q^2} &> \frac{p_1}{p_2}.
\end{aligned}$$

Therefore the conditions for optimality emerge based only on the value of p_1/p_2. It is

$$\frac{\left(1-q^1\right)}{q^1}\frac{\left(1-q^2\right)}{q^2} < \frac{p_1}{p_2} \qquad \text{if } \frac{p_1}{p_2} < 1, \tag{B.1.2}$$

$$\frac{\left(1-q^1\right)}{q^1}\frac{\left(1-q^2\right)}{q^2} > \frac{p_1}{p_2} \qquad \text{if } \frac{p_1}{p_2} > 1. \tag{B.1.3}$$

B.1.3 Simplified case with $p^a_m = p^a$

This section contains the calculations leading to the optimality conditions of the myopic policy for the first further simplified case of the $\mathcal{A} = 2$, $\mathcal{M} = 2$ basic intelligence puzzle with $w(\boldsymbol{s} \neq \mathbf{1}_{\mathcal{M}}) = 1$, as shown in (4.2.29) and (4.2.30). Here the success probabilities of a given type is the same regardless of the source so that $p^1_1 = p^1_2 = p^1$ and $p^2_1 = p^2_2 = p^2$. The other restrictions on the parameters are $t^1_1 = t^1_2 = t^2_1 = t^2_2 = 1$ and $c = 1$. We start by separately computing the terms of (4.2.27). Using the definitions of $\mathbb{P}\left(\boldsymbol{s}_1 \mid \boldsymbol{s}_3, a\right)$ and $\mathbb{E}\left[\mathcal{T}^a\left(\boldsymbol{s}\right)\right]$ found in Section 4.2.1 and denoting q^1_1 as q^1 and q^2_1 as q^2 they simplify to

$$\begin{aligned}
\Delta\mathbb{P}\left(\boldsymbol{s}_1 \mid \boldsymbol{s}_3\right) &= \mathbb{P}\left(\boldsymbol{s}_1 \mid \boldsymbol{s}_3, 1\right) - \mathbb{P}\left(\boldsymbol{s}_1 \mid \boldsymbol{s}_3, 2\right) \\
&= \frac{q^1 p^1_1}{q^1 p^1_1 + \left(1-q^1\right) p^1_2} - \frac{q^2 p^2_1}{q^2 p^2_1 + \left(1-q^2\right) p^2_2} \\
&= \frac{q^1 p^1}{q^1 p^1 + \left(1-q^1\right) p^1} - \frac{q^2 p^2}{q^2 p^2 + \left(1-q^2\right) p^2} = q^1 - q^2.
\end{aligned}$$

$$\begin{aligned}
\mathbb{E}\left[\mathcal{T}^2\left(\boldsymbol{s}_3\right)\right] - \mathbb{E}\left[\mathcal{T}^1\left(\boldsymbol{s}_3\right)\right] &= \frac{q^2 t^2_1 + \left(1-q^2\right) t^2_2}{q^2 p^2_1 + \left(1-q^2\right) p^2_2} - \frac{q^1 t^1_1 + \left(1-q^1\right) t^1_2}{q^1 p^1_1 + \left(1-q^1\right) p^1_2} \\
&= \frac{q^2 + \left(1-q^2\right)}{q^2 p^2 + \left(1-q^2\right) p^2} - \frac{q^1 + \left(1-q^1\right)}{q^1 p^1 + \left(1-q^1\right) p^1} \\
&= \frac{1}{p^2} - \frac{1}{p^1} = \frac{p^1 - p^2}{p^1 p^2}.
\end{aligned}$$

$$\begin{aligned}
\min_a \mathbb{E}\left[\mathcal{T}^a(\boldsymbol{s}_1)\right] - \min_a \mathbb{E}\left[\mathcal{T}^a(\boldsymbol{s}_2)\right] &= \min_a \frac{q^a t^a_0/c + \left(1-q^a\right) t^a_1}{\left(1-q^a\right) p^a_2} - \min_a \frac{q^a t^a_0 + \left(1-q^a\right) t^a_1/c}{q^a p^a_1} \\
&= \min_a \frac{1}{\left(1-q^a\right) p^a} - \min_a \frac{1}{q^a p^a}.
\end{aligned}$$

The expression for $\min_a \mathbb{E}\left[\mathcal{T}^a(\boldsymbol{s}_1)\right] - \min_a \mathbb{E}\left[\mathcal{T}^a(\boldsymbol{s}_2)\right]$ can be further simplified if we were to consider the 4 scenarios given by which actions are better in which states. In the first two the same action is optimal both in $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$. Then

$$\min_a \mathbb{E}\left[T_a(\boldsymbol{s}^{(1)})\right] - \min_a \mathbb{E}\left[T_a(\boldsymbol{s}^{(2)})\right] = \begin{cases} \dfrac{2q^1-1}{q^1\left(1-q^1\right)p^1} & \text{if } a=1 \text{ is always better,} \\ \dfrac{2q^2-1}{q^2\left(1-q^2\right)p^2} & \text{if } a=2 \text{ is always better.} \end{cases}$$

Note that for action 1 to always be better both $\left(1-q^1\right)p^1 > \left(1-q^2\right)p^2$ and $q^1p^1 > q^2p^2$ must be satisfied, in which case $p^1 > p^2$ is always true. Similarly, if action 2 a better choice the $p^1 < p^2$. Then the optimality condition in the case where action 1 is always optimal can be derived by substituting the relevant expressions into (4.2.27) as follows.

$$\begin{aligned} 1 &\geq \left(q^1-q^2\right)\frac{2q^1-1}{q^1\left(1-q^1\right)p^1}\frac{p^1p^2}{p^1-p^2}, \\ 1 &\geq \frac{\left(q^1-q^2\right)\left(2q^1-1\right)p^2}{q^1\left(1-q^1\right)\left(p^1-p^2\right)}, \\ q^1\left(1-q^1\right)\left(p^1-p^2\right) &\geq \left(q^1-q^2\right)\left(2q^1-1\right)p^2, \\ q^1\left(1-q^1\right)\left(\frac{p^1}{p^2}-1\right) &\geq \left(q^1-q^2\right)\left(2q^1-1\right), \\ \frac{p^1}{p^2} &\geq \frac{\left(q^1-q^2\right)\left(2q^1-1\right)+q^1\left(1-q^1\right)}{q^1\left(1-q^1\right)}, \\ \frac{p^1}{p^2} &\geq \frac{\left(q^1\right)^2+q^2-2q^1q^2}{q^1\left(1-q^1\right)}. \end{aligned}$$

Following similar steps with the case where action 2 is always better and noting that $p^1 - p^2 < 0$ we get

$$\begin{aligned}
1 &\geq \left(q^1 - q^2\right) \frac{2q^2 - 1}{q^2\left(1-q^2\right)p^2} \frac{p^1 p^2}{p^1 - p^2}, \\
1 &\geq \frac{\left(q^1 - q^2\right)\left(2q^1 - 1\right)p^1}{q^2\left(1-q^2\right)\left(p^1 - p^2\right)}, \\
q^2\left(1-q^2\right)\left(p^1 - p^2\right) &\leq \left(q^1 - q^2\right)\left(2q^2 - 1\right)p^1, \\
q^2\left(1-q^2\right)\left(1 - \frac{p^2}{p^1}\right) &\leq \left(q^1 - q^2\right)\left(2q^2 - 1\right), \\
-\frac{p^2}{p^1} &\leq \frac{\left(q^1 - q^2\right)\left(2q^2 - 1\right) - q^2\left(1-q^2\right)}{q^2\left(1-q^2\right)}, \\
\frac{p^2}{p^1} &\geq \frac{q^1 + \left(q^2\right)^2 - 2q^1 q^2}{q^2\left(1-q^2\right)}, \\
\frac{p^1}{p^2} &\leq \frac{q^2\left(1-q^2\right)}{q^1 + \left(q^2\right)^2 - 2q^1 q^2}.
\end{aligned}$$

The other set of two scenarios emerge when one action is better in one state and the other in the other state. Then

$$\min_a \mathbb{E}\left[T_a(\boldsymbol{s}^{(1)})\right] - \min_a \mathbb{E}\left[T_a(\boldsymbol{s}^{(2)})\right] = \begin{cases} \dfrac{q^2 p^2 - \left(1-q^1\right)p^1}{\left(1-q^1\right)p^1 q^2 p^2} & \text{if } a = 1 \text{ is better in } \boldsymbol{s}_1, \\ \dfrac{q^1 p^1 - \left(1-q^2\right)p^2}{\left(1-q^2\right)p^2 q^1 p^1} & \text{if } a = 2 \text{ is better in } \boldsymbol{s}_1, \end{cases}$$

Using (4.2.27) the case where $a = 1$ is preferred in $\boldsymbol{s}_1$ but not in $\boldsymbol{s}_2$, the optimality condition if $p^1 > p^2$ is

$$1 \geq \left(q^1 - q^2\right) \frac{q^2p^2 - \left(1 - q^1\right)p^1}{\left(1 - q^1\right)p^1q^2p^2} \frac{p^1p^2}{p^1 - p^2},$$

$$1 \geq \left(q^1 - q^2\right) \frac{q^2p^2 - \left(1 - q^1\right)p^1}{\left(1 - q^1\right)q^2\left(p^1 - p^2\right)},$$

$$\left(1 - q^1\right)q^2\left(p^1 - p^2\right) \geq \left(q^1 - q^2\right)q^2p^2 - \left(1 - q^1\right)p^1,$$

$$\left(1 - q^1\right)q^2p^1 - \left(1 - q^1\right)q^2p^2 \geq \left(q^1 - q^2\right)q^2p^2 - \left(q^1 - q^2\right)\left(1 - q^1\right)p^1,$$

$$\left(1 - q^1\right)q^2p^1 + \left(q^1 - q^2\right)\left(1 - q^1\right)p^1 \geq \left(q^1 - q^2\right)q^2p^2 + \left(1 - q^1\right)q^2p^2,$$

$$q^1\left(1 - q^1\right)p^1 \geq q^2\left(1 - q^2\right)p^2,$$

$$\frac{p^1}{p^2}\frac{q^2\left(1 - q^2\right)}{q^1\left(1 - q^1\right)}.$$

If $p^1 < p^2$, the same steps apply but the relation is reversed. Considering the case where $a = 2$ is preferred in $\boldsymbol{s}_1$ but not in $\boldsymbol{s}_2$ results in the same conditions for optimality. Therefore we have two sets of conditions; two for $p^1 > p^2$

$$\frac{p^1}{p^2} \geq \frac{\left(q^1\right)^2 + q^2 - 2q^1q^2}{q^1\left(1 - q^1\right)}, \tag{B.1.4}$$

$$\frac{p^1}{p^2} \geq \frac{q^2\left(1 - q^2\right)}{q^1\left(1 - q^1\right)}, \tag{B.1.5}$$

and two for $p^1 < p^2$

$$\frac{p^1}{p^2} \leq \frac{q^2\left(1 - q^2\right)}{q^1 + \left(q^2\right)^2 - 2q^1q^2}, \tag{B.1.6}$$

$$\frac{p^1}{p^2} \leq \frac{q^2\left(1 - q^2\right)}{q^1\left(1 - q^1\right)}. \tag{B.1.7}$$

Note that in each pair the second condition is more permissive condition. For (B.1.4) and (B.1.5) we can show this by comparing the numerators of the right hand sides. The condition in (B.1.5) is more permissive if

$$\begin{aligned}q^2\left(1-q^2\right) &\leq \left(q^1\right)^2 + q^2 - 2q^1q^2 \\ 0 &\leq \left(q^1\right)^2 + \left(q^2\right)^2 - 2q^1q^2 \\ 0 &\leq \frac{q^1}{q^2} + \frac{q^2}{q^1} - 2 \\ 2 &\leq \frac{q^1}{q^2} + \frac{q^2}{q^1}.\end{aligned}$$

Since the right hand side of the above expression has a minima at 2, we have shown that (B.1.5) is more permissive than (B.1.4). The same could be done for (B.1.6) and (B.1.7) by comparing the denominators on their right hand side.

We recognise that both (B.1.5) and (B.1.7) are quadratic inequalities and may be rewritten as

$$0 \leq \left(q^2\right)^2 - q^2 + \frac{p^1}{p^2}q^1\left(1-q^1\right) \qquad \text{if } p^1 > p^2$$

$$0 \geq \left(q^2\right)^2 - q^2 + \frac{p^1}{p^2}q^1\left(1-q^1\right) \qquad \text{if } p^1 < p^2$$

which is for $p^1 > p^2$ solved by

$$\frac{1}{2}\left(1-\sqrt{1-4\frac{p^1}{p^2}q^1\left(1-q^1\right)}\right) \leq q^2 \leq \frac{1}{2}\left(1+\sqrt{1-4\frac{p^1}{p^2}q^1\left(1-q^1\right)}\right)$$

and for $p^1 < p^2$ solved by

$$q^2 \leq \frac{1}{2}\left(1-\sqrt{1-4\frac{p^1}{p^2}q^1\left(1-q^1\right)}\right) \qquad \text{or} \qquad \frac{1}{2}\left(1+\sqrt{1-4\frac{p^1}{p^2}q^1\left(1-q^1\right)}\right) \leq q^2,$$

as stated in (4.2.31) and (4.2.32).

B.2 Calculations Regarding the Dynamic Programming Policies

B.2.1 Obtaining the minimising action and associated value function with unknown conditional success probabilities

This section of the appendix contains the steps required to obtain the value function of a state $(\boldsymbol{s}, \boldsymbol{\beta})$ when the state space is limited to $\beta^a_m \leq \beta_{\max}$ $\forall m, a$, namely (4.3.13), from the expression in (4.3.12). Let us start by reminding the reader that the value function of state $(\boldsymbol{s}, \boldsymbol{\beta})$ is

$$\mathbb{V}(\boldsymbol{s}, \boldsymbol{\beta}) = \min_{a \in \mathcal{S}_a} \{\mathbb{V}^a(\boldsymbol{s}, \boldsymbol{\beta})\} \tag{B.2.1}$$

where $\mathbb{V}^a(\boldsymbol{s}, \boldsymbol{\beta})$ is stating the value of action a in state $(\boldsymbol{s}, \boldsymbol{\beta})$, calculated as

$$\begin{aligned}\mathbb{V}^a(\boldsymbol{s}, \boldsymbol{\beta}) = {} & w(\boldsymbol{s}) \sum_{m=1}^{\mathcal{M}} q^a_m t^a_m(\boldsymbol{s}) + \sum_{m=1}^{\mathcal{M}} q^a_m \frac{\alpha^a_m s_m}{\alpha^a_m + \beta^a_m} \mathbb{V}(\boldsymbol{s}_{m^+}, \boldsymbol{\beta}) \\ & + \sum_{m \notin \mu(\boldsymbol{\beta}^a)} q^a_m \left(1 - \frac{\alpha^a_m s_m}{\alpha^a_m + \beta^a_m}\right) \mathbb{V}(\boldsymbol{s}, \boldsymbol{\beta}^a_{m^-}) + \sum_{m \in \mu(\boldsymbol{\beta}^a)} q^a_m \left(1 - \frac{\alpha^a_m s_m}{\alpha^a_m + \beta^a_{\max}}\right) \mathbb{V}(\boldsymbol{s}, \boldsymbol{\beta}).\end{aligned}$$

Let $\mathcal{S}_{a^*}(\boldsymbol{s}, \boldsymbol{\beta})$ denote the set of minimising actions in state $(\boldsymbol{s}, \boldsymbol{\beta})$ so that

$$\mathcal{S}_{a^*}(\boldsymbol{s}, \boldsymbol{\beta}) = \left\{ a; \arg\min_{a \in \mathcal{S}_a} \{\mathbb{V}^a(\boldsymbol{s}, \boldsymbol{\beta})\} \right\}.$$

Observe that as a consequence of (B.2.1), the relation

$$\mathbb{V}^{a \notin \mathcal{S}_{a^*}(\boldsymbol{s}, \boldsymbol{\beta})} > \mathbb{V}^{a \in \mathcal{S}_{a^*}(\boldsymbol{s}, \boldsymbol{\beta})} = \mathbb{V}(\boldsymbol{s}, \boldsymbol{\beta})$$

is true. Based on the above,

$$\begin{aligned}\mathbb{V}^a(\boldsymbol{s}, \boldsymbol{\beta}) \leq {} & w(\boldsymbol{s}) \sum_{m=1}^{\mathcal{M}} q^a_m t^a_m(\boldsymbol{s}) + \sum_{m=1}^{\mathcal{M}} q^a_m \frac{\alpha^a_m s_m}{\alpha^a_m + \beta^a_m} \mathbb{V}(\boldsymbol{s}_{m^+}, \boldsymbol{\beta}) \\ & + \sum_{m \notin \mu(\boldsymbol{\beta}^a)} q^a_m \left(1 - \frac{\alpha^a_m s_m}{\alpha^a_m + \beta^a_m}\right) \mathbb{V}(\boldsymbol{s}, \boldsymbol{\beta}^a_{m^-}) + \sum_{m \in \mu(\boldsymbol{\beta}^a)} q^a_m \left(1 - \frac{\alpha^a_m s_m}{\alpha^a_m + \beta^a_{\max}}\right) \mathbb{V}^a(\boldsymbol{s}, \boldsymbol{\beta}),\end{aligned}$$

as $\mathbb{V}(\boldsymbol{s},\boldsymbol{\beta})$ in the last term of $\mathbb{V}^a(\boldsymbol{s},\boldsymbol{\beta})$ has been replaced with with $\mathbb{V}^a(\boldsymbol{s},\boldsymbol{\beta})$, and can be rearranged to get

$$\mathbb{V}^a(\boldsymbol{s},\boldsymbol{\beta}) \leq \frac{w(\boldsymbol{s})\sum_{m=1}^{\mathcal{M}} q_m^a t_m^a(\boldsymbol{s}) + \sum_{m=1}^{\mathcal{M}} q_m^a \frac{\alpha_m^a s_m}{\alpha_m^a + \beta_m^a} \mathbb{V}(\boldsymbol{s}_{m^+},\boldsymbol{\beta})}{\sum_{m \notin \mu(\boldsymbol{\beta}^a)} q_m^a + \sum_{m \in \mu(\boldsymbol{\beta}^a)} q_m^a \frac{\alpha_m^a s_m}{\alpha_m^a + \beta_{\max}^a}} + \frac{\sum_{m \notin \mu(\boldsymbol{\beta}^a)} q_m^a \left(1 - \frac{\alpha_m^a s_m}{\alpha_m^a + \beta_m^a}\right) \mathbb{V}(\boldsymbol{s},\boldsymbol{\beta}_{m^-}^a)}{\sum_{m \notin \mu(\boldsymbol{\beta}^a)} q_m^a + \sum_{m \in \mu(\boldsymbol{\beta}^a)} q_m^a \frac{\alpha_m^a s_m}{\alpha_m^a + \beta_{\max}^a}}.$$

Let us denote the quantity on the right as $\mathbb{V}'^a(\boldsymbol{s},\boldsymbol{\beta})$. Then we know that

$$\mathbb{V}'^a(\boldsymbol{s},\boldsymbol{\beta}) = \mathbb{V}^a(\boldsymbol{s},\boldsymbol{\beta}) \qquad \forall a \in \mathcal{S}_{a^*},$$
$$\mathbb{V}'^a(\boldsymbol{s},\boldsymbol{\beta}) > \mathbb{V}^a(\boldsymbol{s},\boldsymbol{\beta}) \qquad \forall a \notin \mathcal{S}_{a^*},$$

and therefore the minimising action set and minimising value of $\mathbb{V}^a(\boldsymbol{s},\boldsymbol{\beta})$ and $\mathbb{V}'^a(\boldsymbol{s},\boldsymbol{\beta})$ are identical:

$$\left\{a; \arg\min_{a \in \mathcal{S}_a} \{\mathbb{V}^a(\boldsymbol{s},\boldsymbol{\beta})\}\right\} = \left\{a; \arg\min_{a \in \mathcal{S}_a} \{\mathbb{V}'^a(\boldsymbol{s},\boldsymbol{\beta})\}\right\} = \mathcal{S}_{a^*}(\boldsymbol{s},\boldsymbol{\beta}),$$

$$\min_{a \in \mathcal{S}_a} \{\mathbb{V}^a(\boldsymbol{s},\boldsymbol{\beta})\} = \min_{a \in \mathcal{S}_a} \{\mathbb{V}'^a(\boldsymbol{s},\boldsymbol{\beta})\} = \mathbb{V}(\boldsymbol{s},\boldsymbol{\beta}).$$

Therefore we may write

$$\mathbb{V}(\boldsymbol{s},\boldsymbol{\beta}) = \min_{a \in \mathcal{S}_a} \left\{ \frac{w(\boldsymbol{s})\sum_{m=1}^{\mathcal{M}} q_m^a t_m^a(\boldsymbol{s}) + \sum_{m=1}^{\mathcal{M}} q_m^a \frac{\alpha_m^a s_m}{\alpha_m^a + \beta_m^a} \mathbb{V}(\boldsymbol{s}_{m^+},\boldsymbol{\beta})}{\sum_{m \notin \mu(\boldsymbol{\beta}^a)} q_m^a + \sum_{m \in \mu(\boldsymbol{\beta}^a)} q_m^a \frac{\alpha_m^a s_m}{\alpha_m^a + \beta_{\max}^a}} + \frac{\sum_{m \notin \mu(\boldsymbol{\beta}^a)} q_m^a \left(1 - \frac{\alpha_m^a s_m}{\alpha_m^a + \beta_m^a}\right) \mathbb{V}(\boldsymbol{s},\boldsymbol{\beta}_{m^-}^a)}{\sum_{m \notin \mu(\boldsymbol{\beta}^a)} q_m^a + \sum_{m \in \mu(\boldsymbol{\beta}^a)} q_m^a \frac{\alpha_m^a s_m}{\alpha_m^a + \beta_{\max}^a}} \right\}$$

as given in (4.3.13).

B.2.2 Obtaining the minimising action and associated value function with unknown encounter probabilities

This section of the appendix contains the steps required to obtain the value function of a state $(\boldsymbol{s}, \boldsymbol{\phi})$ when the state space is limited to $\sum_{m=1}^{\mathcal{M}} \phi^a_m \leq \phi_{\max}\ \forall a$, namely (4.4.16), from the expression in (4.4.15). Let us start by reminding the reader that the value of action $a \in \lambda(\boldsymbol{\phi})$ in state $(\boldsymbol{s}, \boldsymbol{\phi})$, calculated as

$$\begin{aligned}\mathbb{V}^{a\in\lambda(\boldsymbol{\phi})}(\boldsymbol{s}, \boldsymbol{\phi}^a) = w(\boldsymbol{s})\frac{\sum_{m=1}^{\mathcal{M}} \phi^a_m t_m(\boldsymbol{s})}{\sum_{m=1}^{\mathcal{M}} \phi^a_m} + \frac{\sum_{m=1}^{\mathcal{M}} \phi^a_m p(\boldsymbol{s})\mathbb{V}(\boldsymbol{s}_{m^+}, \boldsymbol{\phi})}{\sum_{m=1}^{\mathcal{M}} \phi^a_m}\\ + \frac{\sum_{m=1}^{\mathcal{M}} \phi^a_m \left(1 - p^a_m(\boldsymbol{s})\right)\mathbb{V}(\boldsymbol{s}, \boldsymbol{\phi})}{\sum_{m=1}^{\mathcal{M}} \phi^a_m},\end{aligned}$$

where $\lambda(\boldsymbol{\phi})$ is a set of actions for which $\sum_{m=1}^{\mathcal{M}} \phi^a_m = \phi_{\max}$, and that the value function of state $(\boldsymbol{s}, \boldsymbol{\phi})$ is given as

$$\mathbb{V}(\boldsymbol{s}, \boldsymbol{\phi}) = \min_{a\in\mathcal{S}_a}\left\{\mathbb{V}^a(\boldsymbol{s}, \boldsymbol{\phi})\right\},$$

irrespective of a belonging to $\lambda(\boldsymbol{\phi})$ or not. Let $\mathcal{S}_{a^*}(\boldsymbol{s}, \boldsymbol{\phi})$ denote the set of minimising actions in state $(\boldsymbol{s}, \boldsymbol{\phi})$ so that

$$\mathcal{S}_{a^*}(\boldsymbol{s}, \boldsymbol{\phi}) = \left\{a; \operatorname*{arg\,min}_{a\in\mathcal{S}_a}\left\{\mathbb{V}^a(\boldsymbol{s}, \boldsymbol{\phi})\right\}\right\}.$$

Observe that consequently, the relation

$$\mathbb{V}^{a\notin\mathcal{S}_{a^*}(\boldsymbol{s},\boldsymbol{\phi})} > \mathbb{V}^{a\in\mathcal{S}_{a^*}(\boldsymbol{s},\boldsymbol{\phi})} = \mathbb{V}(\boldsymbol{s}, \boldsymbol{\phi})$$

is true. Based on the above we know that

$$\begin{aligned}\mathbb{V}^{a\in\lambda(\boldsymbol{\phi})}(\boldsymbol{s}, \boldsymbol{\phi}^a) \leq w(\boldsymbol{s})\frac{\sum_{m=1}^{\mathcal{M}} \phi^a_m t_m(\boldsymbol{s})}{\sum_{m=1}^{\mathcal{M}} \phi^a_m} + \frac{\sum_{m=1}^{\mathcal{M}} \phi^a_m p(\boldsymbol{s})\mathbb{V}(\boldsymbol{s}_{m^+}, \boldsymbol{\phi})}{\sum_{m=1}^{\mathcal{M}} \phi^a_m}\\ + \frac{\sum_{m=1}^{\mathcal{M}} \phi^a_m \left(1 - p^a_m(\boldsymbol{s})\right)\mathbb{V}^{a\in\lambda(\boldsymbol{\phi})}(\boldsymbol{s}, \boldsymbol{\phi})}{\sum_{m=1}^{\mathcal{M}} \phi^a_m},\end{aligned}$$

which is straightforward to rearrange to get

$$\mathbb{V}^{a\in\lambda(\boldsymbol{\phi})}(\boldsymbol{s}, \boldsymbol{\phi}) \leq w(\boldsymbol{s})\frac{\frac{\sum_{m=1}^{\mathcal{M}} \phi^a_m t^a_m(\boldsymbol{s})}{\sum_{m=1}^{\mathcal{M}} \phi^a_m}}{\frac{\sum_{m=1}^{\mathcal{M}} \phi^a_m p^a_m(\boldsymbol{s})}{\sum_{m=1}^{\mathcal{M}} \phi^a_m}} + \frac{\frac{\sum_{m=1}^{\mathcal{M}} \phi^a_m p(\boldsymbol{s})\mathbb{V}(\boldsymbol{s}_{m^+}, \boldsymbol{\phi})}{\sum_{m=1}^{\mathcal{M}} \phi^a_m}}{\frac{\sum_{m=1}^{\mathcal{M}} \phi^a_m p^a_m(\boldsymbol{s})}{\sum_{m=1}^{\mathcal{M}} \phi^a_m}},$$

and that is the expression in (4.4.16).

B.3 Numerical Study with More Informative Priors

B.3.1 Unknown conditional success probabilities

The numerical experiments in Section 4.3.4 that study $\overline{\mathcal{R}}^{\pi}$ were repeated to evaluate the effects of a more informative prior. Apart from the priors, the same experimental set-up was kept identical. The outcome of this study can be seen in Figure B.3.1 to Figure B.3.4.

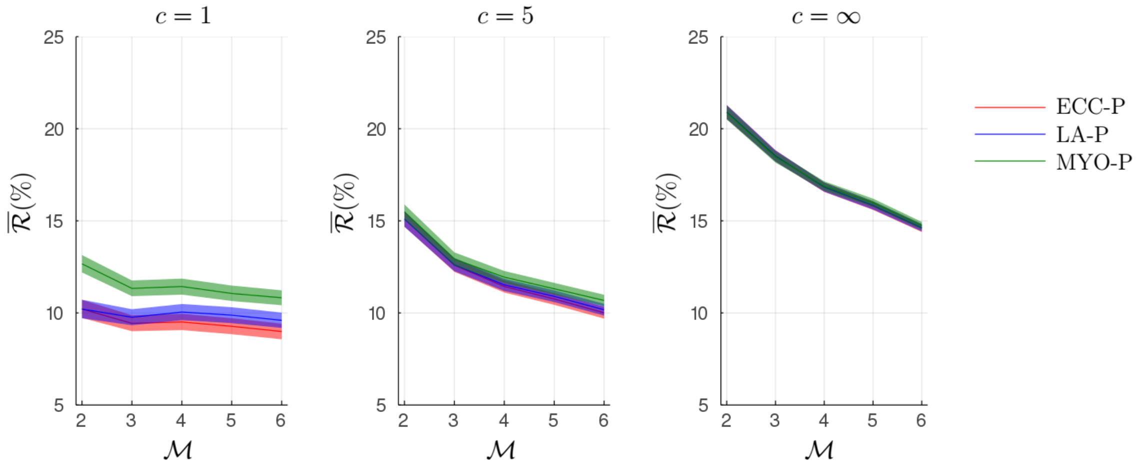


Figure B.3.1: Mean relative regrets of the ECC-P, LA-P and MYO-P policies with more informative priors. Intelligence puzzle with unknown $\boldsymbol{P}^a$, $\mathcal{A} = 2$ and Beta(2,2) priors.

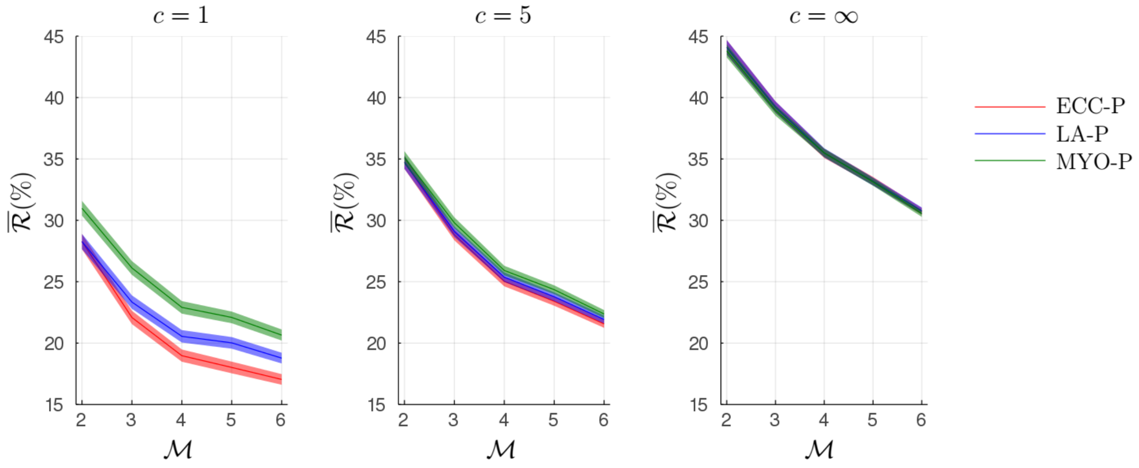


Figure B.3.2: Mean relative regrets of the ECC-P, LA-P and MYO-P policies with more informative priors. Intelligence puzzle with unknown $\boldsymbol{P}^a$, $\mathcal{A} = 5$ and Beta(2,2) priors.

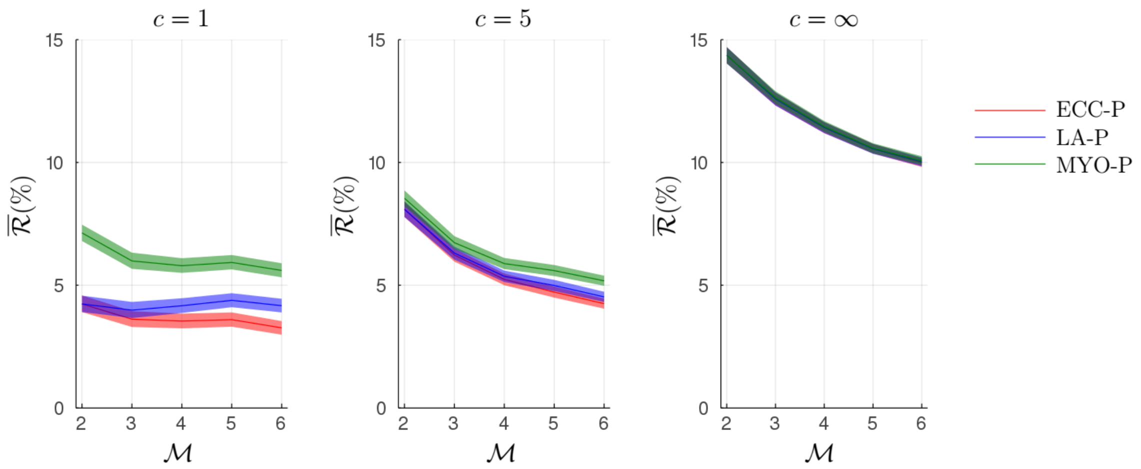


Figure B.3.3: Mean relative regrets of the ECC-P, LA-P and MYO-P policies with more informative priors. Intelligence puzzle with unknown $\boldsymbol{P}^a$, $\mathcal{A} = 2$ and Beta(5,5) priors.

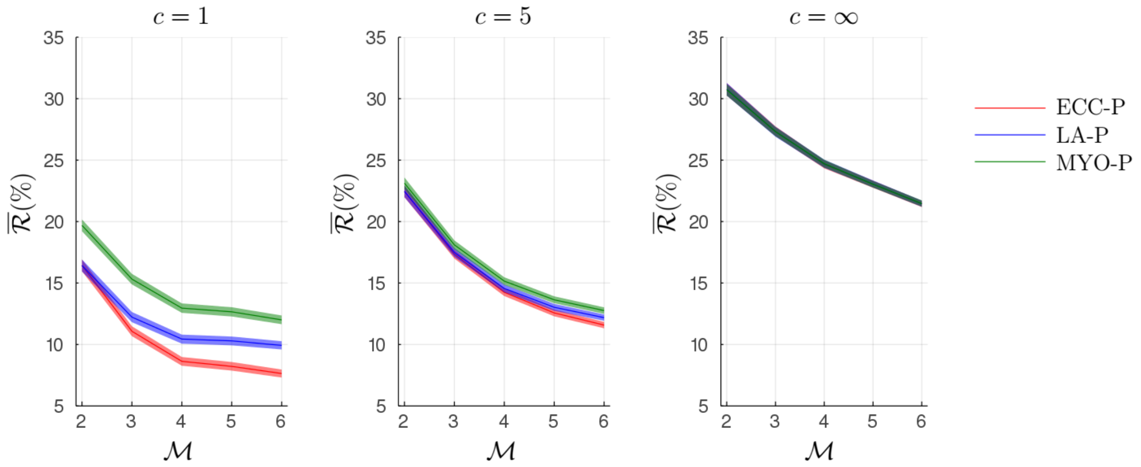


Figure B.3.4: Mean relative regrets of the ECC-P, LA-P and MYO-P policies with more informative priors. Intelligence puzzle with unknown $\boldsymbol{P}^a$, $\mathcal{A} = 5$ and Beta(5,5) priors.

It is clear from the above figures that the conclusions of Section 4.3.4 apply to this numerical study as well. Notice that the more informative the chosen prior is, the lower the observed mean relative regret is.

B.3.2 Unknown encounter probabilities

The numerical experiments in Section 4.4.4 that study $\overline{\mathcal{R}}^{\pi}$ were repeated to evaluate the effects of a more informative prior. Apart from the priors, the same experimental set-up was kept identical. The outcome of this study can be seen in Figure B.3.5 to Figure B.3.8.

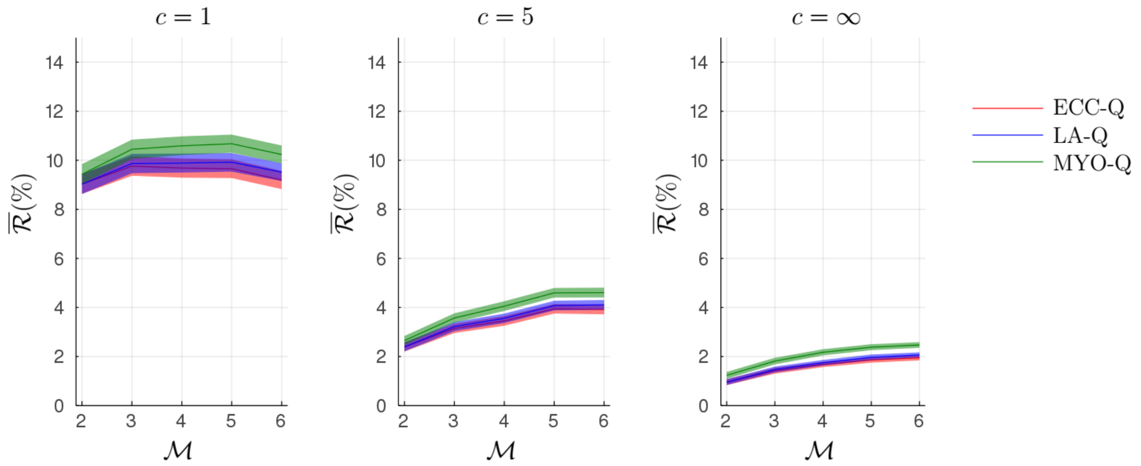


Figure B.3.5: Mean relative regrets of the ECC-Q, LA-Q and MYO-Q policies with more informative priors. Intelligence puzzle with unknown $\boldsymbol{Q}^a$, $\mathcal{A} = 2$ and Dirichlet($\mathbf{2}$) priors.

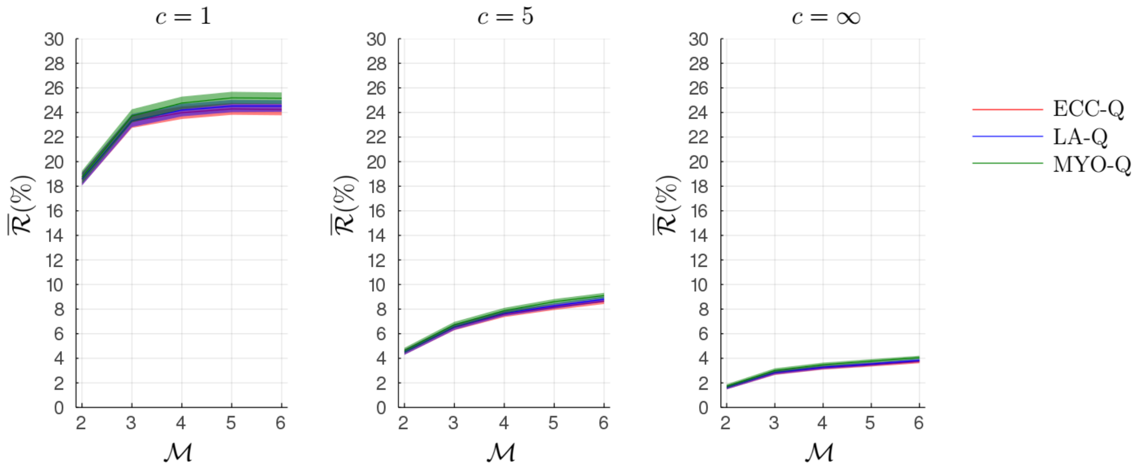


Figure B.3.6: Mean relative regrets of the ECC-Q, LA-Q and MYO-Q policies with more informative priors. Intelligence puzzle with unknown $\boldsymbol{Q}^a$, $\mathcal{A} = 5$ and Dirichlet($\mathbf{2}$) priors.

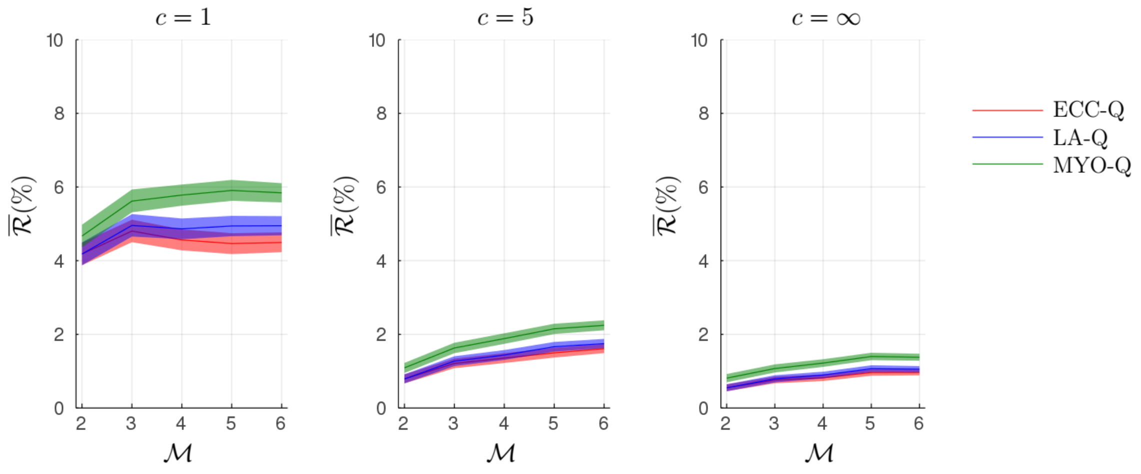


Figure B.3.7: Mean relative regrets of the ECC-Q, LA-Q and MYO-Q policies with more informative priors. Intelligence puzzle with unknown $\boldsymbol{Q}^a$, $\mathcal{A} = 2$ and Dirichlet($\mathbf{5}$) priors.

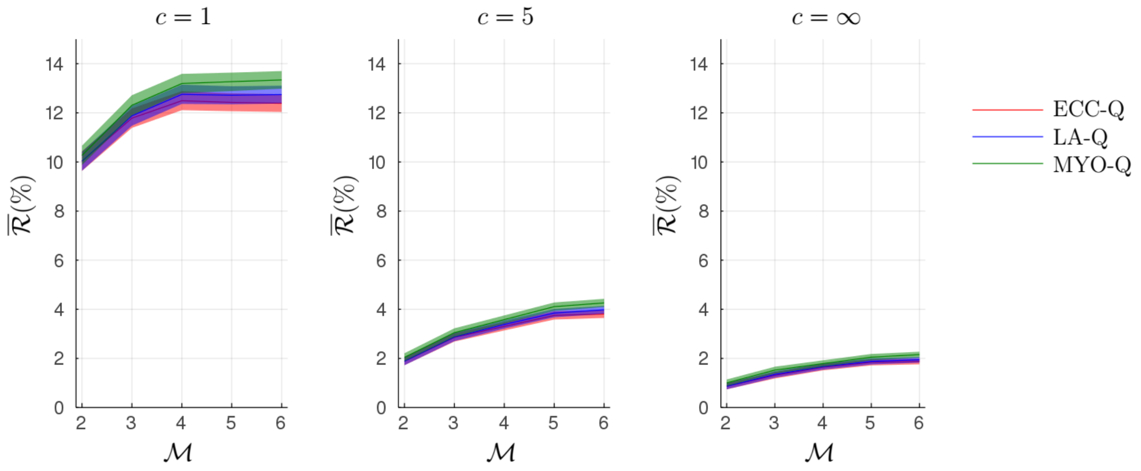


Figure B.3.8: Mean relative regrets of the ECC-Q, LA-Q and MYO-Q policies with more informative priors. Intelligence puzzle with unknown $\boldsymbol{Q}^a$, $\mathcal{A} = 5$ and Dirichlet($\mathbf{2}$) priors.

It is clear from the above figures that the conclusions of Section 4.4.4 apply to this numerical study as well. Notice that the more informative the chosen prior is, the lower the observed mean relative regret is.

Bibliography

Abernethy, J. D., Amin, K., and Zhu, R. (2016). Threshold bandits, with and without censored feedback. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Afanasyeva, L., Bashtova, E., and Bulinskaya, E. (2012). Limit theorems for semi-markov queues and their applications. *Communications in Statistics - Simulation and Computation*, 41(6):688–709.

Atkinson, M., Kress, M., and Lange, R.-J. (2016). When is information sufficient for action? search with unreliable yet informative intelligence. *Operations Research*, 64(2):315–328.

Atkinson, M. P. and Kress, M. (2018). Operating with an incomplete checklist. *IISE Transactions*, 50(4):307–315.

Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E. (2002). The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77.

Bartle, R. G. (1995). *The Elements of Integration and Lebesgue Measure.* John Wiley & Sons, Ltd.

Bather, J. (1980). Randomised allocation of treatments in sequential trials. *Advances in Applied Probability*, 12(1):174–182.

Baykal-Gűrsoy, M. (2011). *Semi-Markov Decision Processes.* John Wiley & Sons, Ltd.

Bellman, R. E. (1957). *Dynamic Programming.* Princeton University Press.

Carmeli, A. (2021). An online learning framework for intelligence collection in uncertain environments. Master's thesis, Naval Postgraduate School.

Cayci, S., Eryilmaz, A., and Srikant, R. (2019). Learning to control renewal processes with bandit feedback. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 3(2):1–32.

Cayci, S., Eryilmaz, A., and Srikant, R. (2020a). Budget-constrained bandits over general cost and reward distributions. In Chiappa, S. and Calandra, R., editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 4388–4398. PMLR.

Cayci, S., Eryilmaz, A., and Srikant, R. (2020b). Continuous-time multi-armed bandits with controlled restarts. *ArXiv*, abs/2007.00081.

Costica, Y. (2010). Optimizing classification in intelligence processing. Master's thesis, Naval Postgraduate School.

Dimitrov, N. B., Kress, M., and Nevo, Y. (2016). Finding the needles in the haystack: efficient intelligence processing. *Journal of the Operational Research Society*, 67(6):801–812.

Dizon-Ross, R. and Ross, S. M. (2020). Dynamic search models with multiple items. *Annals of Operations Research*, 288(1):223–245.

Drugowitsch, J. (2013). Variational bayesian inference for linear and logistic regression. *arXiv: Machine Learning*.

Edwards, J., Fearnhead, P., and Glazebrook, K. (2017). On the identification and mitigation of weaknesses in the knowledge gragient policy for multi-armed bandits. *Probability in the Engineering and Informational Sciences*, 31(2):239–263.

Frazier, P. I., Powell, W. B., and Dayanik, S. (2008). A knowledge-gradient policy for sequential information collection. *SIAM Journal on Control and Optimization*, 47(5):2410–2439.

Gittins, J., Glazebrook, K., and Weber, R. (2011). *Multi-armed bandit allocation indices*. John Wiley & Sons.

Gittins, J. and Jones, D. (1974). A dynamic allocation index for the sequential allocation of experiments. pages 287–298, read at the 1972 European Meeting of Statisticians, Budapest. North Holland.

Gittins, J. C. (1979). Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society: Series B (Methodological)*, 41(2):148–164.

Graczová, D. and Jacko, P. (2014). Generalized restless bandits and the knapsack problem for perishable inventories. *Operations Research*, 62(3):696–711.

Gupta, S. S. and Miescke, K. J. (1996). Bayesian look ahead one-stage sampling allocations for selection of the best population. *Journal of Statistical Planning and Inference*, 54(2):229–244. 40 Years of Statistical Selection Theory, Part I.

Huang, Y. and Guo, X. (2011). Finite horizon semi-markov decision processes with application to maintenance systems. *European Journal of Operational Research*, 212(1):131–140.

Jaakkola, T. S. and Jordan, M. I. (2000). Bayesian parameter estimation via variational methods. *Statistics and Computing*, 10(1):25–37.

Jacko, P. (2019). The finite-horizon two-armed bandit problem with binary responses: A multidisciplinary survey of the history, state of the art, and myths. *ArXiv*, abs/1906.10173.

Janssen, J. and Manca, R. (2007). *Semi-Markov risk models for finance, insurance and reliability*. Springer Science & Business Media.

Kaplan, E. H. (2010). Terror queues. *Operations Research*, 58(4):773–784.

Kaplan, E. H. (2012). Or forum–intelligence operations research: the 2010 philip mccord morse lecture. *Operations Research*, 60(6):1297–1309.

Kaplan, E. H. and Feinstein, J. S. (2012). Intel queues. *Military Operations Research*, pages 17–30.

Kelly, F. P. (1981). Multi-armed bandits with discount factor near one: The Bernoulli case. *The Annals of Statistics*, 9(5):987 – 1001.

Kress, M. and MacKay, N. J. (2014). Bits or shots in combat? the generalized deitchman model of guerrilla warfare. *Operations Research Letters*, 42(1):102–108.

Kronzilber, D. (2017). Multi-armed bandit models of network intrusion in the cyber domain. Master's thesis, Naval Postgraduate School.

Lattimore, T. and Szepesvári, C. (2020). *Bandit algorithms*. Cambridge University Press.

Mamer, J. W. (1986). Successive approximations for finite horizon, semi-markov decision processes with application to asset liquidation. *Operations Research*, 34(4):638–644.

Marshall, J. (2016). *Models of intelligence operations*. PhD thesis, Lancaster University.

Meczarski, M. (2015). Stochastic orders in the bayesian framework. *Collegium of Economic Analysis Annals*, (37):339–360.

Müller, A. and Stoyan, D. (2002). Comparison methods for stochastic models and risks.

Powell, W. B. and Ryzhov, I. O. (2012). *Optimal learning*, volume 841. John Wiley & Sons.

Puterman, M. L. (1994). *Markov decision processes*. Wiley Series in Probability & Mathematical Statistics: Applied Probability & Statistics. John Wiley & Sons, Nashville, TN.

Ross, S. and Seshadri, S. (2021). An optimal stocking problem to minimize the expected time to sellout. *Operations Research Letters*, 49(1):69–75.

Ross, S. M., Weiss, G., and Zhang, Z. (2021). Technical note–a stochastic assignment problem with unknown eligibility probabilities. *Operations Research*, 69(1):266–272.

Ross, S. M. and Wu, D. T. (2013). A generalized coupon collecting model as a parsimonious optimal stochastic assignment model. *Annals of Operations Research*, 208(1):133–146.

Ryzhov, I. O., Powell, W. B., and Frazier, P. I. (2012). The knowledge gradient algorithm for a general class of online learning problems. *Operations Research*, 60(1):180–195.

Seidl, A., Kaplan, E. H., Caulkins, J. P., Wrzaczek, S., and Feichtinger, G. (2016). Optimal control of a terror queue. *European Journal of Operational Research*, 248(1):246–256.

Sherman, J. and Morrison, W. J. (1950). Adjustment of an Inverse Matrix Corresponding to a Change in One Element of a Given Matrix. *The Annals of Mathematical Statistics*, 21(1):124–127.

Spiegelhalter, D. J. and Lauritzen, S. L. (1990). Sequential updating of conditional probabilities on directed graphical structures. *Networks*, 20(5):579–605.

Tekin, M. (2016). New perspectives on intelligence collection and processing. Master's thesis, Naval Postgraduate School.

Thompson, W. R. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3-4):285–294.

Tijms, H. C. (2003). *A first course in stochastic models*. Wiley, Chichester, UK.

Tran-Thanh, L., Chapman, A., Rogers, A., and Jennings, N. (2012). Knapsack based optimal policies for budget–limited multi–armed bandits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 26, pages 1134–1140.

White, D. J. (1993). A survey of applications of markov decision processes. *Journal of the Operational Research Society*, 44(11):1073–1096.

Wolfstetter, E. (1996). Stochastic dominance: Theory and applications.

Wrzaczek, S., Kaplan, E. H., Caulkins, J. P., Seidl, A., and Feichtinger, G. (2017). Differential terror queue games. *Dynamic Games and Applications*, 7(4):578–593.