Optimal Search in Discrete Locations: Extensions and New Findings

Jake Clarkson, M.Sci.(Hons.), M.Res

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

July 2020

Abstract

A hidden target needs to be found by a searcher in many real-life situations, some of which involve large costs and significant consequences with failure. Therefore, efficient search methods are paramount. In our search model, the target lies in one of several discrete locations according to some hiding distribution, and the searcher's goal is to discover the target in minimum expected time by making successive searches of individual locations.

In Part I of the thesis, the searcher knows the hiding distribution. Here, if there is only one way to search each location, the solution to the search problem, discovered in the 1960s, is simple; search next any location with a maximal probability per unit time of detecting the target. An equivalent solution is derived by viewing the search problem as a multi-armed bandit and following a Gittins index policy.

Motivated by modern search technology, we introduce two modes—fast and slow—to search each location. The fast mode takes less time, but the slow mode is more likely to find the target. An optimal policy is difficult to obtain in general, because it requires an optimal sequence of search modes for each location, in addition to a set of sequence-dependent Gittins indices for choosing between locations. For each mode, we identify a sufficient condition for a location to use only that search mode in an optimal policy. For locations meeting neither sufficient condition, an optimal choice of search mode is extremely complicated, depending both on the hiding distribution and the search parameters of the other locations. We propose several heuristic policies motivated by our analysis, and demonstrate their near-optimal performance in an extensive numerical study.

In Part II of the thesis, the searcher has only one search mode per location, but does not know the hiding distribution, which is chosen by an intelligent hider who aims to maximise the expected time until the target is discovered. Such a search game, modelled via two-person, zero-sum game theory, is relevant if the target is a bomb, intruder, or, of increasing importance due to advances in technology, a computer hacker. By Part I, if the hiding distribution is known, an optimal counter strategy for the searcher is any corresponding Gittins index policy. To develop an optimal search strategy in the search game, the searcher must account for the hider's motivation to choose an optimal hiding distribution, and consider the set of corresponding Gittins index policies. However, the searcher must choose carefully from this set of Gittins index policies to ensure the same expected time to discover the target regardless of where it is hidden by the hider. As a result, finding an optimal search strategy, or even proving one exists, is difficult.

We extend several results for special cases from the literature to the fully-general search game; in particular, we show an optimal search strategy exists and may take a simple form. Using a novel test, we investigate the frequency of the optimality of a particular hiding strategy that gives the searcher no preference over any location at the beginning of the search.

Acknowledgements

I'll begin by thanking my supervisors, without whom this PhD would not have been possible. First, Kevin. It seems a long time since I mistakenly thought you were called Keith (coincidently, the name of your twin) after my STOR-i welcome day! Since then we've spent many hours discussing challenging problems in your office, but enough about the owners of our respective football clubs; I think we made more progress talking about search problems. Thanks for your invaluable guidance, encouragement and wisdom regarding not only the PhD but also my future career. You were also an excellent tour guide in the Monterey Peninsula, teaching me a lot about, inter alia, wildlife (sorry no whale watching) and Motown. I also encountered some fantastic new fish-based dishes there; who couldn't get swept up in your love of cioppino and, in particular, clam chowder? Monterey brings me on to Kyle, who enriched each NPS visit by being a most-welcoming host. Yet, your influence extended far beyond Monterey trips, thank you for your continual help and meticulous feedback throughout the three years - your input was crucial. In particular, you've helped my writing; now ~~I tend to find that I use less unnecessary words and phrases and, therefore, can be more 'to the point', so to speak, without losing any information that I wish to convey to the reader~~ I'm more succinct. I hope to work more with both of you in the future.

Trips to NPS and the PhD itself would not have been possible without the financial support of ESPRC, of which I am grateful, and, of course, STOR-i itself, a wonderfully supportive, friendly and vibrant atmosphere in which to do research. I'd like to thank everyone involved at STOR-i, staff and students alike, but in particular Jon Tawn and the admin team, Kim Wilson, Jen Bull and Wendy Shimmin, who make everything run so smoothly whilst all being so friendly and approachable. I'd especially like to thank them for dealing with my requests regarding lumbar support and ever-changing thesis hand in date.

I've had so many enjoyable experiences at STOR-i: dressing up for the Christmas meal, dancing until cramp hits at the STOR-i ball, away days (the good and the 'interesting'). I've made lots of great friends who I'd like to thank for enduring my endless tales of buses, bargains and dreams, humouring my latest fad, and supporting me through tougher times. Since I celebrated it recently, in reverse-chronological order of the date of their golden birthday (Paperless Post, 2020), I'd like, in particular, to thank:

Graham: no-one else takes such an interest in my dreams (or my kitchen utensils...); I raise my eyebrows at others!

Sean: at his peak, a fantastic food critic and user of satire, and otherwise still pretty funny.

Lucy: our humour never failed to brighten my day, most of it Terribly Silly, some slightly crude, laughing until we're both crazy like a fool!

Sam: the most superb, astounding, majestic good sir, someone Hoo's always great for a Chuckle no matter Watt the situation; you think of everything! Thank you also for introducing me to the concept of a 'golden birthday'; I'm sure your unbelievable memory for dates will correct me if the order of this list is wrong.

Emily: always there for support whenever I needed it, and always up for a good old natter! Never lose your Twinkling Sparkle!

Kathryn: a regularly-supportive ear over peppermint tea, always helping put the *pieces into place* ...

Luke: a fount of wise words and stimulating conversation, not least about who should open and how many spinners to play in the next test.

Euan: a natural quipster, who kept me from going nuts by turning the seemingly mundane into a zesty mix of trpamolining dogs, fetching emails and much more.

Next come the only four people I've met who come close to making me look sane.

Jack: the man with the best-smelling arms in Devon. Bike-safety gear aside, a fellow seeker of good value; I got the best value with your friendship.

David: a bin-diving, graffiti-dishing, 'fit'-assessing, Spaniard-in-Belgium-spotting, sign-slapping, 'vegetarian' Tasmanian devil. Whilst your antics never failed to enlighten my day, in some tough times you've been a knight in shining armour.

Harry: a frequent partner in crime, always egging eachother on in our latest craze. A refreshingly-free spirit with relentless enthusiasm and très Bon life advice when required. Er ist alles in Butter (as long as it's grass fed).

Harjit: I've mentioned the best Baker, now time for the worst baker, the confident, insightful, whimsical and practical Harjit Hullait. Thanks for all the moments that can only be described as 'CH' (and C doesn't stand for conventional); social events were always better when you hadn't food in the oven.

And that just leaves . . . Rob, my LIC[1] for 8 (eight) glorious years, where do I start? From the Brummie hipster who I Met on an unquestionably-hot day in October 2011, you've changed a fair bit. You've learnt about the washing machine, how to use pots and pans (as opposed to baking trays and more baking trays), and that AM doesn't end when you go to bed. On a serious note, your excellent sense of humour, willingness to play 'guessing games' and continuing interest in my Cricket Captain saves has significantly enriched my time at Lancaster; we've laughed uncontrollably an uncountable number of times, mainly at things an outsider would find most baffling.

Away from STOR-i, I'd like to thank my home friends (who've been bored my tales far more than anyone else) Aidan, Imogen, James and Robyn for many jokes, mainly directed (amicably) at eachother. I'd also like to thank Andrew for his Terrific Sass.

Finally, I'd like to thank my family, especially my parents, grandparents and sister.

[1] live-in-chum

I don't think they were expecting me to have 23 years of education, but they've been a pillar of strength throughout all of it. I consider myself extremely lucky to have them; they've picked me up when I'm down, and we've regularly laughed with (and often at) eachother in better times. Without you guys, I would not have made it this far.

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.

With several adjustments detailed at the start of Chapter 3 of Part I, the content in Part I, Chapter 3 has been accepted for publication as Clarkson, J., Glazebrook, K. D., and Lin, K. Y. (2020) Fast or Slow: Search in Discrete Locations with Two Search Modes. *Operations Research*, 68(2):552-571. Parts of the abstract have also been taken from the abstract of the aforementioned paper.

A version of the content in Part II, Chapter 7 will be submitted for publication by Clarkson, J., Lin, K. Y., and Glazebrook, K. D. in 2020.

The word count for this thesis is approximately 69,000 words.

Jake Clarkson

Contents

List of Figures

List of Tables

Motivation

There are many real-life situations involving a hidden target which needs to be found by a searcher. Examples include a bomb or landmine by a bomb squad, a survivor of a disaster by a rescue team, an intruder by a patroller, and the remains of a ship or plane by a salvage team. A recent, famous example of the latter is the Malaysia Airlines Flight 370 that went missing in the Indian Ocean in early 2014. The search need not be a physical process, however; the target could be a hacker of a computer system, or a bug in some programming code.

In all of these applications, there is a lot at stake. Huge costs may be involved in conducting the searches; for example, the search for Flight 370 is estimated to have cost between \$130 and \$160 million dollars (Australian Transport Safety Bureau, 2017). Also, there may be significant consequences if the search is unsuccessful; valuable equipment could be forfeited or damaged, personal data could be breached or, even worse, human life could be lost. Therefore, efficient search methods are paramount, either to end the search in a minimal amount of time, or to give the searcher the best chance of finding the target before a deadline.

Any mathematical formulation of a problem where a searcher looks for a target hidden in some region, called the *search space*, is called a *search problem*. Search problems can broadly be split into two categories depending on the number of entities able to make decisions, called *decision makers*. In a *one-sided search*, the searcher is the only decision maker, choosing how to traverse the search space to look for the target. Among the aforementioned examples, a salvage team searching for the remains of a ship or a plane and a programmer looking for a bug are one-sided searches. Stone (1975) is a monograph on one-sided search.

In a *two-sided search*, there is a second decision maker who controls the position of the target in the search space. The addition of the second decision maker creates an extra layer of complexity, since their actions affect the best course of action for the searcher. Therefore, game theory, the study of interaction between intelligent decision makers, is often used to model a two-sided search. Of the aforementioned examples, in the case of the intruder and the hacker, the target itself is the second decision maker, whilst in the case of the hidden bomb or landmine, the second decision maker is a third party who hides the target. In both situations above, the second decision maker, called the *hider*, wants the target to remain hidden, either for as long as possible to maximise damage caused in the case of the intruder or the hacker, or until a known detonation deadline in the case of the hider of the bomb or landmine. Such scenarios involving a searcher and hider with directly conflicting objectives are a particular type of two-sided search called a *search game*. Alpern and Gal (2003) is a monograph on search games, in which two-person, zero-sum game theory plays a key part.

Our final example involves a rescue team looking for a survivor. Whilst the survivor is an intelligent entity, their situation affects how the search is modelled. If the survivor cannot move, possibly trapped under rubble after an earthquake, then they are not a decision maker, and the search is one sided. Otherwise, the survivor is a decision maker, and the search is two sided. However, unlike the hider, the survivor does not want to remain hidden; the searcher and survivor have identical objectives. Such problems fall into a class of two-sided search called *rendezvous search*, a topic also covered in Alpern and Gal (2003).

Whilst a survey of the search literature will be provided, this thesis will focus on extensions to the following classic one-sided, discrete search problem. The target is an inanimate object hidden in one of several discrete locations according to some known probability distribution. The searcher aims to discover the object in minimum expected time by making successive searches of individual locations. The searcher must conduct a full search of a location before moving on to another one. Each location has a known, fixed time it takes to search that location, and a known, fixed probability with which the object, if hidden there, is found upon searching that location.

A simple solution is derived and reported in several papers from the early 1960s, some using direct computation and others dynamic programming methods (Norris, 1962; Bram, 1963; Matula, 1964; Black, 1965). A comment by Kelly in Gittins (1979) notes that the problem is equivalent to a tractable version of the well-known multi-armed bandit problem (MAB). First proposed by Thompson (1933), the MAB problem has many applications, including to clinical trials and web design, and takes its name from slot machines in casinos called one-armed bandits. When a player plays a one-armed bandit, they are allocated a reward typically drawn from some unknown distribution; the MAB problem presents the player with several of these one-armed bandits and asks in which order should they be played to maximise the player's total reward. Kelly views each location as a one-armed bandit played whenever the location is searched. The celebrated work Gittins and Jones (1974) finds a remarkably simple solution to a version of the MAB where the reward from playing a bandit depends entirely upon a known state that the bandit is in. Kelly applies the solution of Gittins and Jones (1974) to the classic search problem to corroborate the aforementioned results from the early 1960s.

This PhD has two aims. The first, tackled in Part I of the thesis, is to add a second speed to the classic search problem, so the searcher may search each location either at a fast speed or a slow speed. A fast search takes less time, but the slow search has a higher probability of finding the object, leading to a trade-off between speed and effectiveness. Two or more search speeds for each location are often available in real-life search problems. Firstly, there may be a choice of several search devices; for example, a search for a bomb down a stretch of road could be conducted travelling quickly in a vehicle with sensors, or travelling slowly on foot with trained sniffer dogs.

The vehicle travel, analogous to the fast search, covers the road more quickly, but the chance of missing a hidden bomb will increase. Being on foot with the trained dogs, corresponding to the slow speed, will take more time to cover the same distance, but may well detect a hidden bomb with a larger probability. Secondly, of increasing importance due to advances in technology, for a search device such as a robot, there may be multiple settings on the travel speed or sensor mode.

Despite this increased relevance, two-speed problems have received little attention in the academic literature. To our knowledge, only Shechter et al. (2015) and Alpern and Lidbetter (2015) investigate search problems involving two search modes. However, neither account for the searcher overlooking the object at the slow speed, a key part of our model. The former also permits no overlook of the object at the fast speed, with the objective to avoid damage to both the object and the searcher. Kadane (2015) allows an arbitrary number of search modes per location in the classic search problem, but the simple solution presented in Section 3 of Kadane (2015) is not correct.

In the classic search problem with only one speed per location, the searcher only needs to choose which location to search at which time. In the two-speed extension, the searcher must also choose at which speed to search a selected location. Therefore, the multi-armed bandit framework is no longer directly applicable, with the problem now equivalent to an extension to the MAB coined a *family of alternative superprocesses* by Gittins et al. (2011). For such problems, a simple solution is not guaranteed to exist. A key research question is the following. For a single location, is it always better to search at one speed than another, or does the best speed depend on the wider progress of the search process? Kadane (2015) concludes the former is true, but we disagree, finding an optimal choice of speed to be complicated in some scenarios.

The second aim of this thesis, tackled in Part II, is to develop the existing literature on a two-sided extension of the classic (one-speed) one-sided problem in which the target, rather than an inanimate object, is a hider who aims to remain hidden for a maximum expected time. Such an extension is a search game, which, unlike the one-sided problem, is far from being solved by the existing literature. Most current work is either limited to the case where each location takes the same time to search (Bram, 1963; Gittins and Roberts, 1979), or additionally limited to two locations (Roberts and Gittins, 1978; Gittins, 1989). Norris (1962) makes some preliminary findings in the general search game with arbitrary locations and search times, but few theoretical results have been proven.

Due to advances in technology, intruders, in particular computer hackers, are becoming more and more sophisticated. Further, in this progressively digital age, personal information stored on computer systems is becoming more and more valuable. Therefore, we believe the search game is an increasingly important problem which warrants further study. We aim to extend some of the aforementioned work to the general search game, in particular the existence of an optimal search strategy shown by Bram (1963) and the existence of a simple optimal search strategy derived by Gittins (1989).

Outline of Thesis

This thesis is split into two parts labelled Part I and Part II. The former concerns one-sided search where a searcher looks for an inanimate target; the latter regards two-sided search, specifically search games where a searcher seeks an intelligent hider who actively tries to avoid being found. Whilst both types of search share some common features, their mathematical modelling differs significantly; therefore, each part contains its own literature review, background knowledge, conclusions, further work, and appendices. Some concluding remarks relevant to both parts and a shared bibliography are at the end of the thesis.

The outline of Part I is as follows. Chapter 1 reviews the one-sided search literature with a focus on searches in discrete locations. Particular attention is paid to problems solved by an index policy, where a real number (index) is calculated for each location and a location with a maximal index chosen to search next.

Chapter 2 describes a special type of Markov decision process called a multi-armed bandit, known to be optimally solved by a Gittins index policy of Gittins and Jones (1974). We elaborate on a comment of Kelly in Gittins (1979), which shows that a classic search problem in discrete locations introduced in Chapter 1 can be modelled as a multi-armed bandit and hence solved by a Gittins index policy. Whilst the classic search problem was solved in the literature for special cases before the discovery of a Gittins index policy, the Gittins index approach has the advantage of solving the problem in its full generality, useful for the following novel classic-search-problem extension formulated in Section 2.3.2. The searcher is given a choice of two modes when searching any location, a fast mode and a slow mode. A search with the fast mode takes less time, but is more likely to overlook the target sought by the searcher, leading to a trade-off between speed and effectiveness that substantially complicates the problem.

Chapter 3, an extended version of Clarkson et al. (2020), studies the two-mode extension, determining conditions on a location for only one of its modes to be used in an optimal search policy. Alongside a Gittins index policy, these conditions solve the problem in many cases, otherwise simplifying it. Insight into the different types of benefit to searching a location is used to design heuristic policies, which consistently deliver near-optimal performance in a thorough numerical study.

Chapter 4 summarises Part I and suggests several extensions of the two-mode search, specifically to more than two search modes per location, and the placing of locations onto a network structure.

The outline of Part II is as follows. Since a search game involves two players (searcher and hider) in direct conflict, two-person, zero-sum game theory is relevant; an introduction is given in Chapter 5.

Chapter 6 reviews the search-game literature, which has mainly focused on problems where the searcher always discovers the hider when their paths cross. Models allowing the searcher to overlook the hider are less common; the latter part of Chapter 6 summarises the literature's development of a simple search game with overlook in discrete locations, often limited to special cases.

Chapter 7 studies the discrete-location search game in full generality, proving that an optimal search strategy exists and may take a simple form. A simple test for the optimality of a strategy for the hider is developed and used to investigate the rate of optimality of a particular hiding strategy in an extensive numerical study.

Chapter 8 summarises Part II and proposes the addition of a second search mode to the search game of Chapter 7 in a combination of Parts I and II.

PART I: One-Sided Search

Dreaming about the day when you wake up and find that what you're looking for has been here the whole time.

– Taylor Swift

Chapter 1

Literature Review

Search theory is one of the oldest branches of operational research with origins in World War II. In 1942, the US Navy set up the Anti-Submarine Warfare Operations Research Group to explore techniques for quick location of enemy submarines and ships. The group's work laid the foundations of search theory, with the main contributor being an American mathematician called Bernard Koopman. After the war, Koopman recorded his results in a book called 'Search and Screening' (Koopman, 1946). The book was declassified by the US Navy in 1958, which motivated a boom in research into search theory. Many bibliographies (Benkoski et al., 1991; Dobbie, 1968; Enslow Jr., 1966; Hohzaki, 2016) and surveys (Alpern and Gal, 2003; Stone, 1975, 2004; Stone et al., 2016; Washburn, 2002) have since summarised and collated the extensive literature on the topic.

What follows, aided by the aforementioned bibliographies and surveys, is a literature review of one-sided search, the topic of Part I of this thesis. The literature review of two-sided search is deferred to Part II.

In any one-sided search problem, the sole decision maker is a searcher who seeks a target hidden in some *search space*. The searcher does not know the target's precise location, yet does have some knowledge of the target's whereabouts expressed using a probability distribution, called the *target distribution*, defined on the search space.

A *search policy* guides the searcher through the search space either until the target is found or, if one exists, until a search deadline. The searcher may be free to move instantaneously from one point in the search space to another, or may be constrained in some way by the geography of the search space. The target is found according to a known detection function dependent on the target's location and the time the searcher spends there. The aim of the searcher is to optimise a *search objective*, some measure of the success of the search.

The search space can be discrete or continuous, and the target stationary or moving. This survey will briefly consider continuous search spaces in Section 1.1, before presenting a more thorough exposition of discrete search spaces in Section 1.2. In the latter, there is a particular emphasis on problems solved or partially solved by an index policy, since such policies are key to the work later in this thesis.

1.1 Continuous Search Spaces

Early continuous-search-space work focuses on a stationary target with no constraints on the movement of the searcher. The most common search objective in such problems, maximising the probability of finding the target for a fixed amount of search effort E, is first studied in Koopman (1957), one of three excerpts from Koopman (1946) to be published in the journal *Operations Research*. A search policy is a distribution of the effort E over the search space. If x effort is applied at point y in the search space, a known detection function determines the probability of finding the target at y given it is located there. Koopman (1957) finds an optimal search policy when the detection function is exponential. de Guenin (1961) extends Koopman's work to a more general class of detection functions, and Dobbie (1963) additionally solves the problem of minimising the total search effort to attain a given probability of finding the target.

In the problem of Koopman (1957) and its extensions above, the order that search effort is made does not matter, as the searcher is only interested in maximising the probability of finding the target once all the allocated effort E has been made. However, if E is unknown or uncertain, more important parts of the search space should be searched first. Arkin (1964) lets a search policy be a distribution of E over both the search space and time. With search space $\mathbb{R}^n$, Arkin (1964) derives conditions for the existence of a *uniformly optimal* search policy, a search policy optimal for any amount of effort E. In other words, if a uniformly optimal search policy was stopped after any amount of effort E had been made, the probability of finding the target with E effort available would have been maximised.

A moving target was introduced into the problem of Koopman (1957) in the 1970s. Since the target moves, the searcher's position at a given time is now important, so Arkin's definition of a search policy as a distribution of effort over time and space must be assumed. The target's motion can be modelled as stochastic, or deterministic given some stochastic parameter. The latter is first studied in Stone and Richardson (1974), with a parameter vector in $\mathbb{R}^n$ with known distribution. A detection function maps a parameter value $\boldsymbol{\theta} \in \mathbb{R}^n$ and a search policy to a probability of detection conditional on the target moving according to $\boldsymbol{\theta}$. For a general detection function, Stone and Richardson (1974) finds uniformly optimal search policies. Stone (1977) and Pursiheimo (1977) independently generalise the space of the parameter to allow both the target's initial position and velocity to be random.

When the target moves according to a stochastic process, a sample path of the process replaces the parameter argument $\boldsymbol{\theta}$ in the detection function. All assuming an exponential detection function, Hellman (1970, 1971) consider the target moving via a diffusion process and Saretsalo (1973) via a more general Markov process. Stone (1979) provides necessary and sufficient conditions for an optimal policy, allowing any concave detection function and the target to move according to a wide range of stochastic processes. Stone's results apply to both a target moving in discrete or continuous time and to a discrete or continuous search space.

Constraints are first added to the searcher's motion in Beck (1964, 1965) in the following *linear search*. The target is hidden in $\mathbb{R}$ according to a known distribution, and the searcher, starting at the origin, must follow a continuous path in $\mathbb{R}$, detecting the target with probability 1 when passing it. The objective is to minimise the expected time until the target is found. Early results in Beck (1964, 1965) find conditions on the target distribution required for an optimal policy to exist. Franck (1965) independently derives these conditions, additionally solving the problem in simple cases and providing an algorithm to estimate an optimal policy in the general case. With increasingly inventive titles, Anatole Beck continued to study linear search; for example, Beck and Warren (1973) consider a non-linear objective where searching becomes more expensive over time, and Beck and Beck (1984) consider particular target distributions (e.g uniform).

Foley et al. (1991) studies linear search with Koopman's objective, maximising the probability of detection with a fixed amount of search effort. More recently, Jotshi and Batta (2008) extends linear search to a network structure, a finite collection of points in $\mathbb{R}^2$ (nodes) and lines of various lengths (edges) which join some pairs of nodes. A heuristic search policy is proposed for the case where the target distribution is uniform across the network. Jotshi and Batta (2009) allows two targets to be hidden uniformly on a network, with the objective to minimise the time to find both. Search on a network is far more common in the two-sided search with an intelligent hider discussed in more detail in Part II of this thesis.

In the wider literature, there are many extensions of the groundbreaking continuous-search-space work described above. However, we now proceed to a survey of the focus of the work in this thesis, a discrete search space.

1.2 Discrete Search Spaces

A discrete search space is partitioned into n subareas called 'boxes', 'locations' or 'cells' in the literature; here we use 'boxes'. The following describes the classic search problem in discrete time and space. The position of the target within a box is not important, only which box the target lies in; therefore, the target distribution is $\mathbf{p} \equiv (p_1, \ldots, p_n)$, where p_i is the known probability that the target is located in box i, $i = 1, \ldots, n$. The target is stationary throughout the search. The searcher finds the target by consecutively searching individual boxes, so a search policy is an ordered sequence of boxes to search.

For $i = 1, \ldots, n$ and $k \in \mathbb{Z}^+ \equiv \{1, 2, \ldots\}$, the kth search of box i takes known *search time* $t_i(k) > 0$ to complete, and finds the target, if in box i, with known *detection probability* $q_i(k) \in (0, 1)$. It is assumed that the $t_i(k)$ do not degrade to 0 as $k \to \infty$. For $i = 1, \ldots n$, q_i and t_i may be viewed as functions with domain $\mathbb{Z}^+$ and respective ranges $(0, 1)$ and $\mathbb{R}^+$. If these functions are constant for each box, in other words, we may write $q_i \equiv q_i(k)$ and $t_i \equiv t_i(k)$ for $k \in \mathbb{Z}^+$, $i = 1, \ldots, n$, then the problem is *time independent*; otherwise, the problem is *time dependent*.

A similar model allows searches to be made in continuous time; in other words, the searcher is free to spend any length of time searching a box before moving to another. A search policy is, therefore, an ordered sequence of boxes alongside a corresponding list of search lengths. Detection probabilities are determined by functions $\mathcal{Q}_i : \mathbb{R}^+ \to (0, 1)$, where $\mathcal{Q}_i(x)$ is the probability of having detected the target after a total of x time units of search in box i, given the target is in box i, $i = 1, \ldots, n$. For examples of such models, see Onaga (1971), Wegener (1981), Nakai (2004) and Chapter 2 of Stone (1975).

The following survey, however, will concentrate on the classic search problem in discrete time and its extensions, with a particular focus on problems where an index policy plays a part in the solution. A variety of search objectives have been studied in the literature, split by subsection throughout the rest of this survey.

1.2.1 Minimising the Expected Search Time

Whilst for continuous search spaces the early work maximised the probability of finding the target with a fixed amount of search effort available, in the discrete case, the early work minimised the expected time until the target is found, or *expected search time*. Since all detection probabilities are strictly less than 1, after no number of searches can the searcher guarantee finding the target, so, with this objective, a search policy is an *infinite*, ordered sequence of boxes.

Blachman (1959) solves one of the earliest time-independent search problems found in the literature. However, unlike the classic problem, whilst $\mathbf{p}$ still determines which box will contain the target, the target does not lie in this box from the beginning of the search, rather appearing after a random, uniformly-distributed length of time.

The classic time-independent problem, where the target is present from the start of the search, was solved in the years following. A policy which minimises the expected search time is as follows. If m_i unsuccessful searches have already been made of box i, $i = 1, \ldots, n$, the next search is of any box with a maximal value of

$$\frac{p_i(1-q_i)^{m_i} q_i}{t_i}. \tag{1.2.1}$$

The above is an example of an *index policy*, where an index, (1.2.1) in this case, is calculated for each box, and any box with a maximal index is searched next.

If m_i failed searches have been made of box i, then, by Bayes' theorem, $p'_i \equiv p_i(1-q_i)^{m_i}/k$ is the posterior probability that the target is in box i, $i = 1, \ldots, n$, where $k = \sum_{j=1}^{n} p_j(1-q_j)^{m_j}$. Therefore, an equivalent index policy to (1.2.1) uses

$$\frac{p'_i q_i}{t_i} \tag{1.2.2}$$

as the index of box i, with $\mathbf{p}' \equiv (p'_1, \ldots, p'_n)$ representing the current posterior target distribution. Note that, given the target has not been found, $p'_i q_i$ is the probability of finding the target on the next search of box i, so (1.2.2) is the probability of detection per unit time. Therefore, a policy optimal in the classic time-independent problem picks a box with a maximal probability of detection per unit time on its next search and hence is myopic.

Several authors independently proved the optimality of (1.2.1) (and hence also (1.2.2)). Norris (1962), a technical report at MIT focused on a two-sided search where the target is an intelligent hider, uses a direct approach, showing that, at any point during the search, any arbitrary search policy may be improved by instead choosing a box using (1.2.1). Bram (1963), which also studies a two-sided search, proves the result when $t_i = 1$ for $i = 1, \ldots, n$ (known as *unit search times*) by a short, direct approach exploiting the one-to-one correspondence between the finishing times of individual searches and the positive integers. Matula (1964) reports that Blackwell derived the result using dynamic programming in 1962 in some unpublished notes. Black (1965) provides an intuitive graphical proof and, years later, Ross (1983) a simple argument involving an interchange of consecutive searches.

In addition to reporting Blackwell's proof, Matula (1964) shows that if and only if $\log(1 - q_i)/\log(1 - q_j) \in \mathbb{Q}$ for all $i, j \in \{1, \ldots, n\}$ does there exist an optimal index policy which, after an initial transient period, is periodic, in other words, in which some fixed, finite sequence of boxes is repeated indefinitely. In this case, a closed-form expression can be found for the optimal expected search time.

The remainder of this subsection discusses extensions to the classic time-independent problem, many of which are either solved by some variation of (1.2.1), or have (1.2.1) play a part in their solution.

Small Adjustments to the Objective An extension in Norris (1962) assumes a penalty is incurred for a failed search, and the searcher minimises the total expected penalty incurred until the target is discovered. Unlike a search time, the penalty of a failed search depends on the target's true location as well as the box searched.

Specifically, if box j is searched whilst the target is in box i, the penalty is $\rho_i t_j$ for some known $\rho_i > 0$. Norris (1962) shows that (1.2.1) is optimal if the index of box i is multiplied by ρ_i.

In Sweat (1970), the searcher receives a fixed, known reward upon finding the target, but, if m_i searches are made of box i before the target is found, $i = 1, \ldots, n$, the reward is discounted by $\prod_{i=1}^{n} \beta_i^{m_i}$ for known $\beta_i \in (0, 1)$. Sweat (1970) shows (1.2.1) maximises the expected reward if t_i is replaced by $1 - \beta_i$.

Time Dependence Chapter 4 of the monograph Stone (1975) studies the classic time-dependent search problem. If m_i unsuccessful searches have been made of box i, $i = 1, \ldots, n$, an index analogous to (1.2.1) is

$$\frac{p_i q_i(m_i + 1) \prod_{k=1}^{m_i} (1 - q_i(k))}{t_i(m_i + 1)}. \tag{1.2.3}$$

If (1.2.3) is decreasing in m_i for $i = 1, \ldots, n$, then, by extending the method of Black (1965), Stone (1975) proves the optimality of any index policy with index (1.2.3). Further, under the same condition, by adapting a general result of Degroot (1970), Stone (1975) shows that such index policies are the only optimal search policies.

It is unsurprising that (1.2.3) must be decreasing in m_i for the corresponding index policy to be optimal. Like (1.2.1), the index in (1.2.3) is proportional to the probability of detection per unit time on the next search of box i, so is a measure of how appealing the $(m_i + 1)^{\text{st}}$ search of box i is to the searcher. Therefore, like (1.2.1), any index policy with index (1.2.3) is myopic. Suppose, for some $j \in \{1, \ldots, n\}$, (1.2.3) is not decreasing in m_j, and the $(m_j + 2)^{\text{nd}}$ search of box j is more appealing to the searcher than the $(m_j + 1)^{\text{st}}$. Suppose m_i failed searches have been made of box i, $i = 1, \ldots, n$. In order to make the appealing $(m_j + 2)^{\text{nd}}$ search of box j as soon as possible, it could be optimal to search box j next even if the current index of box j is not maximal. In other words, if searches of a fixed box may become more appealing with time, the searcher's optimal behaviour is no longer myopic. Kadane and Simon (1977) and Wegener (1980) both construct an algorithm to compute an optimal policy in the time-dependent case without assuming (1.2.3) is decreasing in m_i. The former assumes some finite number of searches of each box will guarantee a detection, an assumption relaxed by the latter. A comment by Kelly in Gittins (1979) finds a closed-form solution to the fully-general classic time-dependent problem; Kelly's solution will be discussed in detail in Chapter 2.

Random Detection Probabilities With time-independent search times, Hall (1976, 1977) replace $q_i(k)$ with a random variable $Q_i(k)$ with a known distribution, $i = 1, \dots, n$, $k \in \mathbb{Z}^+$. Immediately after the kth search of box i is made, the realisation of $Q_i(k)$ used in that search is revealed to the searcher. Therefore, it is possible for the searcher to follow the index policy in (1.2.3) with $q_i(m_i + 1)$ replaced by the expectation of $Q_i(m_i + 1)$. Hall (1976) shows that, when the $Q_i(k)$ are independent, such an index policy is optimal under the decreasing condition of Stone (1975). Hall (1977) extends this result to dependent $Q_i(k)$.

With unit search times, Kelly (1982) assumes the detection probability of any search of any box is a realisation from the same random variable Q with known distribution. However, past realisations of Q are not known by the searcher, so, if m_i failed searches have been made of box i, only $p_i E[Q(1 - Q)^{m_i}]$ can be calculated, an index shown to be optimal. Kelly (1982) also investigates the behaviour of an optimal index policy as the number of failed searches tends to infinity. Like Matula (1964), there is periodic behaviour; however, unlike Matula (1964), the nature of the behaviour depends on the target distribution $\mathbf{p}$.

Multiple Targets Smith and Kimeldorf (1975) considers K hidden targets in the classic time-independent problem, where K is a random variable with a known distribution. Each target is hidden according to the target distribution $\mathbf{p}$. The objective is to minimise the expected time until the first target is found. Smith and Kimeldorf (1975) shows an optimal policy exists no matter the distribution of K.

The main result of Smith and Kimeldorf (1975) is the following. For $n \geq 3$, an optimal index policy exists if and only if K has a positive Poisson distribution (Poisson conditional on $K \geq 1$). In this case, the optimal index policy, deduced in the following, is based on the myopic indices in (1.2.1) and (1.2.2) known to be optimal when $P(K = 1) = 1$. Recall the numerator $p_i' q_i$ of (1.2.2) is the probability that the next search of box i ends the search. In Smith and Kimeldorf (1975), the search ends if at least one of the targets is found. If box i is searched, any given target is found with probability $p_i' q_i$; therefore, if there are k targets, the probability of finding at least one target is $1 - (1 - p_i' q_i)^k$. Suppose m_i failed searches have been made of box i so far. Since p_i is the probability that any given target was placed in box i at the start of the search, and each target in box i is found with probability q_i independently of any other targets, by Bayes' theorem, we have $p_i' \propto p_i(1 - q_i)^{m_i}$ as in the classic problem with one target.

Since the number of targets is a random variable K, the searcher can only estimate the true number of targets using $E[K]$. Finding no targets in a box changes the searcher's beliefs not only about $\mathbf{p}$, but also about K; therefore, the expectation is taken over the current posterior for K, also simply updated by Bayes' theorem. It follows that the myopic index of Smith and Kimeldorf (1975) is given by

$$\frac{1 - E_K\left[(1 - p_i(1 - q_i)^{m_i} q_i)^K\right]}{t_i}. \tag{1.2.4}$$

For K with positive Poisson distribution with parameter $\lambda > 0$, an equivalent index policy replaces the numerator of (1.2.4) with $1 - \exp(-\lambda p_i(1 - q_i)^{m_i} q_i)$. Also, as expected, note that if $P(K = 1) = 1$ and the classic time-independent model is recovered, then (1.2.4) reduces to (1.2.1).

Assaf and Zamir (1987) studies the two-box problem with generally-distributed K, showing that (1.2.4) is not always optimal, but $\log(E[K^x])$ being concave for $x \in [0, 1]$ is sufficient for the optimality of (1.2.4), a condition satisfied by any positive binomial or positive Poisson distribution, but not every geometric distribution. Sharlin (1987) presents an algorithm to find an optimal policy for a two-box problem for K with any distribution.

Assaf and Zamir (1985) assumes the number of targets k is known, but the target distribution $\mathbf{P} \equiv (P_1, \ldots, P_n)$ with which each target is hidden is random with a known distribution. A myopic index is similar to (1.2.4), but with expectation taken over the current posterior for $\mathbf{P}$, namely

$$\frac{1 - E_{\mathbf{P}}\left[(1 - P_i(1 - q_i)^{m_i} q_i)^k\right]}{t_i}. \tag{1.2.5}$$

When $k = 1$, (1.2.5) reduces to (1.2.1) with p_i replaced by $E[P_i]$; Assaf and Zamir (1985) shows such an index policy is optimal for $\mathbf{P}$ with any distribution. Further, when $n = 2$, (1.2.5) is shown to be optimal for any k and distribution on $\mathbf{P}$, contrasting Assaf and Zamir (1987) where the number of targets was random but the target distribution known.

Kimeldorf and Smith (1979) studies the model of Smith and Kimeldorf (1975), but minimises the expected time until all targets are found. It is assumed that the searcher is unaware of how many targets have been found until the search is over. The main result of Smith and Kimeldorf (1975) is shown to still hold: for $n \geq 3$, an optimal index policy exists if and only if K has a positive Poisson distribution. The optimal index policy is similarly adapted from (1.2.2) as follows. if m_i searches of box i have been made, $i = 1, \ldots, n$, and there are k targets remaining, then the probability of ending the search by finding all k targets on the next search of box i is proportional to $(p_i(1 - q_i)^{m_i} q_i)^k$. Therefore, the optimal index is

$$\frac{E_{K'}\left[(p_i(1 - q_i)^{m_i} q_i)^{K'}\right]}{t_i}, \tag{1.2.6}$$

where K' is the random variable representing the number of targets remaining. At the start of the search, $K' = K$, with the distribution of K' updated throughout the search by Bayes' theorem. For K with positive Poisson distribution with parameter $\lambda > 0$, an equivalent index policy replaces the numerator of (1.2.6) with $\exp(\lambda p_i(1 - q_i)^{m_i} q_i) - 1$; note the comparison to the result of Smith and Kimeldorf (1975). Also, as expected, note that if $P(K' = 1) = 1$, (1.2.6), like (1.2.4), reduces to (1.2.1).

Kimeldorf and Smith (1979) also draws comparisons with the discounted reward model of Sweat (1970). If C is the time until all targets are found, it is shown that (1.2.6) with t_i replaced by $\exp(t_i) - 1$ is optimal if the objective is to minimise $E[1 - \exp(-C)]$.

False Detections Kress et al. (2008) considers a time-independent search with the usual detection probabilities and search times made by a sensor which may falsely detect the target in box i with probability $r_i < q_i$, $i = 1, \ldots, n$. After a reported detection in box i, a team with perfect detection verifies the sensor's findings, which takes time c_i. Since there are more outcomes to each search, calculating the posterior $\mathbf{p}'$ is more complicated than in the classic search problem, but still possible using Bayes' theorem. In particular, if the faultless verification team are sent to some box i, then either the target is found or box i is removed from consideration. Kress et al. (2008) shows that the myopic index policy in (1.2.2) is optimal if $c_i r_i$, the expected time wasted by the verification team on a false detection in box i, is added to the denominator.

Kress et al. (2008) also considers the case where the verification process is either long or risky. Here, an appropriate alternative objective maximises the probability that the first detection by the sensor is a true detection. In this case, it is shown that (1.2.2) is optimal with t_i replaced by r_i.

Travel Times Gluss (1961) studies the unit-search-time problem with perfect detection ($q_i = 1$ for $i = 1, \ldots, n$), meaning a search policy is some permutation of $\{1, \ldots, n\}$. The boxes lie in a line, and it takes the searcher time $|i - j|$ to travel from box i to box j, for $i, j \in \{1, \ldots, n\}$. Gluss (1961) was the first to consider travel times between n boxes, where the objective is to minimise the combination of (expected) search and travel time. Unlike the previous problems in this section, there is no simple, optimal index policy, and Gluss (1961) only finds a solution for a uniform target distribution.

Lössner and Wegener (1982) generalises Gluss (1961) with an arbitrary travel time from box i to box j; an interpretation is a search on a network structure with the target hidden at one of the nodes. Lössner and Wegener (1982) derives necessary and sufficient conditions for an optimal policy to be periodic as in Matula (1964). Under these conditions, an optimal policy must belong to a finite set of search policies. Solving the travel-time problem without such periodic behaviour was later shown to be NP-hard in Wegener (1985).

1.2.2 Maximising the Probability of Detection by a Deadline

Chew (1967) is the first to study the classic discrete search problem with the objective of Koopman (1957), namely, maximising the probability of finding the target before a fixed deadline, $d > 0$. Since (if it lasts that long) the search will end at time d whether the target has been found or not, a search policy is a finite sequence of boxes which ends when the next desired search would take the cumulative search time over d. Further, since the precise detection time of the target before d does not matter, only the number of searches of each box in a search policy is important; therefore, a search policy can be simplified to a list $a_1, \ldots, a_n$, where a_i is the number of times box i is to be searched, and $\sum_{i=1}^{n} a_i t_i \leq d$. However, due to the interest in uniformly optimal search policies, which, recall, are optimal for any choice of deadline, it is useful to continue to think of a search policy as an ordered sequence of boxes of indeterminate length.

Even though the objective is no longer to minimise the expected search time, the index policies in (1.2.1) and (1.2.3) still play a key part in optimal solutions. Chew (1967) shows that, in the classic time-independent problem with unit search times, (1.2.1) is uniformly optimal. Kadane (1968) extends this result to the classic time-dependent problem with unit search times under the same condition ((1.2.3) is decreasing in m_i) as Chapter 4 of Stone (1975) discussed in Section 1.2.1. Therefore, with unit search times and (1.2.3) decreasing in m_i, using (1.2.3) as an index both minimises the expected search time and maximises the probability of detection before any deadline.

The same cannot be said for general search times. In this case, Chapter 4 of Stone (1975) shows that (1.2.3) maximises the probability of finding the target before any deadline which coincides with the end of a search of some box under (1.2.3). For any other deadline, (1.2.3) may not be optimal. The following simple example demonstrates why this is the case. Suppose $n = 2$ and $t_1(k) = 1$, $t_2(k) = 5$ for all $k \in \mathbb{Z}^+$. Suppose the index in (1.2.3) is larger for box 2 at the start of the search, so any index policy ξ using (1.2.3) as its index will search box 2 first. If the deadline d belongs to $[1, 5)$, the probability of finding the target before d under ξ is 0, as the first search cannot be completed before d. Yet, under any policy which searches box 1 first, the same probability is non zero, showing that ξ is suboptimal. In other words, at the start of the search, a search of box 2 has a greater probability of detection per unit time than box 1, but cannot be finished before the deadline, so is suboptimal. Clearly such conflicts cannot happen in the unit-search-time problems of Chew (1967) and Kadane (1968). Wegener (1982) both proves the existence of and constructs an algorithm to find an optimal search policy for general, time-dependent search times.

The remainder of this subsection discusses the use of index policies in extensions to the time-independent model of Chew (1967).

Multiple Targets and False Detections In addition to minimising the expected search time, two papers discussed in Section 1.2.1 also maximise the probability of detection before a known deadline for unit search times, thus extending the model of Chew (1967). In both, it was shown that (1.2.1) remained uniformly optimal. The first is the model of Kress et al. (2008), where the searcher may falsely detect the target with a known, box-dependent probability. The second is the model of Kimeldorf and Smith (1979), where the number of targets is a random variable. Kimeldorf and Smith (1979) shows that (1.2.1) is uniformly optimal regardless of how many targets need be detected before the deadline and whether or not the searcher is told how many targets have been found throughout the search process.

Multiple Searchers In the classic time-independent problem with unit search times, Song and Teneketzis (2004) allows x searchers who can search for the target simultaneously. Whilst detection probabilities may vary from box to box, each searcher has the same detection probability for a fixed box. Only one searcher is allowed in a box at any time; therefore, it is assumed that $x < n$ or the problem is trivial. A simple extension of the index policy in (1.2.1) is to search the x boxes with the x largest indices in the next time period. However, it is shown by counter example that this myopic policy is not always optimal. An optimal policy does use (1.2.1), but must look ahead until the deadline d, calculating (1.2.1) with $m_i = 0, 1, \ldots d - 1$ for $i = 1, \ldots, n$ and collecting the largest xd indices into a set $\mathcal{L}$. For $i = 1, \ldots, n$, if m of the d indices calculated for box i are in $\mathcal{L}$, then an optimal policy makes m searches of box i before the deadline d. Since only the probability of detection by d is important, the total of xd searches can be made in any order provided multiple searchers do not enter the same box at the same time.

Ding and Castanon (2017) extends the model of Song and Teneketzis (2004) by allowing heterogeneous searchers. Each searcher s has the same detection probability for each box, but has their own budget b_s and their own subset of boxes $\mathcal{B}_s$. Searcher s cannot make more than b_s searches in total, and cannot search any box not in $\mathcal{B}_s$. The objective is to maximise the probability of finding the target once each searcher has used up their budget. An optimal index policy no longer exists; Ding and Castanon (2017) formulates a minimum-cost network flow problem and derives an algorithm to find an optimal policy. An extension where the detection probability in any box depends on the choice of searcher is formulated; however, comparison to the weapon-allocation problem of Ahuja et al. (2007) shows the extension to be NP-hard, and only approximate algorithms are provided.

1.2.3 A Moving Target

As for a continuous search space, study of a moving target began in the 1970s. Both search objectives of the previous two subsections have been studied. Henceforth, for brevity, we refer to these objectives as *MinT* (minimising the expected search time) and *MaxP* (maximising the probability of detection before a known deadline).

Pollock (1970) is the first to add a moving target to the classic time-independent problem, analysing the simple case with two boxes and unit search times. As usual, the target's starting box is determined by the target distribution; yet, at the end of each search of a box, the target, unseen by the searcher, switches boxes with a known probability. The switching probability depends only on the target's current box, so target movement can be modelled using a discrete-time Markov chain with a known Markov transition matrix M.

In the stationary-target problem, the searcher uses their history of failed searches to calculate the posterior probability p' that the target is in box 1. With a moving target, M also provides information about the target's location, making the calculation of p' more complicated and the problem harder to solve. For both *MinT* and *MaxP*, Pollock (1970) shows that the optimal policy is a threshold policy; in other words, for some $p^* \in (0, 1)$, search box 1 if and only if $p' > p^*$. However, the value of p^* depends on the objective, and a closed form is only found in special cases. Schweitzer (1971) investigates recursive procedures to calculate p^* for *MinT*.

Ross (1983) conjectures that, in the two-box model of Pollock (1970) with general search times, there exists an optimal threshold policy for *MinT*. Note that, in Ross' extension, the movement of the target no longer coincides with $\mathbb{Z}^+$, and is dependent on the policy of the searcher. While White (1992) shows Ross' conjecture is true for a wide range of parameters, and Weber (1986) proves the conjecture if searches are made in continuous time, in full generality with discrete-time searches, the conjecture remains unproven to this day.

A recent extension to the unit-search-time, two-box model of Pollock (1970) is studied in Flesch et al. (2009). The searcher has perfect detection, and, at each unit-time step, alongside searching either box, has a third option available: wait and search no box. So the searcher is not penalised for waiting, the objective is to minimise the expected cost to find the target rather than time, with a search of either box incurring cost 1. To avoid waiting forever being optimal, the searcher receives a prize upon finding the target, which is subtracted from the incurred cost. With the prize available, the searcher may wish to make a search when the posterior target distribution contains less uncertainty; i.e., p' is close to 0 or 1. Indeed, Flesch et al. (2009) shows that a two-threshold policy of the following form is either optimal or optimal within any $\epsilon > 0$. For $0 < p_l \leq p_u < 1$, search box 2 for $p' \in [0, p_l]$, wait for $p' \in (p_l, p_u)$, and search box 1 for $p' \in [p_u, 1]$.

Kan (1977) extends the model of Pollock (1970) to n boxes and general search times, but solves the problem only in special cases. For example, the case where every row of the transition matrix M is equal to some fixed $\mathbf{v}$ is solved. Here, after every search, the posterior target distribution $\mathbf{p}'$ is $\mathbf{v}$, so the history of failed searches gives the searcher no information about the target's location. Therefore, for all but the first search, it is optimal to search the same box. For *MinT*, an optimal box j^* is any with a maximal value of (1.2.2) with $p'_i = v_i$, $i = 1, \ldots, n$. An optimal box to search first is determined by a separate index dependent on j^* and the original target distribution $\mathbf{p}$. In the unit-search-time case, for *MaxP*, it is optimal to search box j^* in every search.

Due to the difficulty in solving the models of Pollock (1970) and Kan (1977), the majority of moving-target problems in the literature allow searches to be made in continuous time, as in Weber (1986). Dobbie (1974) considers a target moving via a continuous-time Markov process, but in most work the target moves via a discrete-time Markov chain. Famous examples include Brown (1980) and Washburn (1983); for a recent survey, see Stone et al. (2016).

1.2.4 Information-Based Objectives

All search objectives discussed so far concern finding the target. However, the searcher may instead be interested in learning as much as possible about the target's location. In the classic time-independent problem, Mela (1961) is the first to consider maximising the expected information gained about the target's location before a deadline d (*MaxI*), measuring information gain by the difference in entropy between the original target distribution and the posterior target distribution $\mathbf{p}^d$ at time d. Using simple two-box examples, Mela (1961) shows that optimal policies for *MaxP* and *MaxI* with the same deadline do not always coincide. This section will consider two common search objectives involving some degree of information gain.

Whereabouts Searches The first objective is also considered by Mela (1961), namely, maximising the probability of correctly guessing (at d) the target's location (*MaxG*). *MaxG* is realistic if a major rescue operation for a hostage is planned at time d, but a sensor may make preliminary sweeps of the search space to determine where the hostage is located. If the target is found before the deadline d, then the searcher will guess correctly with probability 1. Otherwise, the searcher clearly optimally guesses the box j with the largest posterior probability in $\mathbf{p}^d \equiv (p_1^d, \ldots, p_n^d)$, and is correct with probability p_j^d. A further three-box example in Mela (1961) shows that an optimal policy for *MaxP* does not always coincide with an optimal policy for *MaxG*. Therefore, for *MaxG*, it is not always optimal to simply maximise the chance of finding the target before d, then make the best guess possible if the target remains undiscovered; some thought must be given to reducing the uncertainty in $\mathbf{p}^d$.

The literature has since studied *MaxG* further; Tognetti (1968) is the first, coining the term *whereabouts search*. A policy for a whereabouts search involves two quantities: a sequence of boxes to search up to time d (of which the order is unimportant) and a box to guess if the search is unsuccessful. Tognetti (1968) consists of a single example, a generalisation of an example of Mela (1961) to n boxes. The target distribution is uniform, and the boxes are identical with search time 1. By the result of Chew (1967), if $d = n$, for *MaxP*, the problem is optimally solved by following (1.2.1) and hence searching each box once. However, if the target is not found, $\mathbf{p}^d$ is uniform, leading to maximum uncertainty and the worst possible situation for guessing. Tognetti (1968) shows that, for general $d \in \mathbb{Z}^+$, any optimal policy for *MaxG* never searches the box it guesses.

Kadane (1971) generalises Tognetti's result to the classic time-dependent problem with *MaxG*. Further, it is shown that, given the optimal box j^* to guess, determining the optimal sequence ξ^* of boxes to search is simply the problem with objective *MaxP* and box j^* removed from consideration. Therefore, by Kadane (1968), if all search times are unit, the index policy (1.2.3) (with box j^* ignored) determines ξ^*.

Kadane (1971) shows that if all boxes are identical, then j^* is the mode of the target distribution, a natural choice, since it leads to the mode of $\mathbf{p}^d$ being maximised. However, such a choice is not always optimal for non-identical boxes, since the box with largest probability in the target distribution may be searched very efficiently, allowing a lot of information to be gained in a short space of time by searching that box. In general, j^* can only be determined by solving n problems with objective *MaxP*, each corresponding to a different guessed box - a result generalised by Stone and Kadane (1981) to a target moving in either discrete or continuous time.

Search and Stop In the second type of problem, search times are considered as search costs with the objective to minimise the expected cost. However, information on the target's location is more valuable than in a *MinT* problem since, after any search of an individual box, for a known penalty cost c, the searcher has the option to stop the search before finding the target. The stop option may be appealing to the searcher if uncertainty about where to optimally search next is high, or if there is a strong chance that the target lies in a box with an inefficient or useless search mode.

Chew (1967) was the first to allow stopping with penalty c, adapting the classic time-independent problem with unit search costs. Chew (1967) assumes that one box, say box n, is *unsearchable* with detection probability 0; therefore, if the target lies in box n it cannot be found, so any optimal policy must stop after a finite number of searches. The index policy in (1.2.1) again is key; Chew (1967) shows that any optimal policy either searches a box with a maximal index or stops the search.

Ross (1969) studies the problem of Chew (1967) with general search times, but neither an unsearchable box nor a cost for stopping the search. Yet, there is a reward r_i for finding the target in box i, and the objective is to minimise the expected cost minus the expected reward. Ross (1969) shows that his objective is equivalent to that of Chew (1967) with $c = r_i$, $i = 1, \ldots, n$.

Ross (1969) shows an optimal policy exists, but only that the result of Chew (1967) (that it is optimal to choose a box using (1.2.1) or stop) holds when r_i is constant over all boxes. In other words, Ross (1969) extends the result of Chew (1967) to general search times, but not to a box-dependent reward structure. However, for box-dependent rewards, Ross (1969) shows that if it is ever suboptimal to search a box i with a maximal index, it is never optimal to search box i at any point in the search process. Unlike Chew (1967), the absence of an unsearchable box means it may never be optimal stop searching; Ross (1969) shows that this is the case and (1.2.1) is optimal if $\sum_{i=1}^{n} t_i/q_i r_i < 1$. Chapter 5 of Stone (1975) extends the results of Ross (1969) to time-dependent detection probabilities and search times under the same assumption as Sections 1.2.1 and 1.2.2, that (1.2.3) is decreasing in m_i for $i = 1, \ldots, n$.

The above shows that, under certain assumptions, a simple, myopic index policy tells the searcher an optimal box to search next. However, it is not so easy to know when to optimally stop searching. Chew (1967) bounds the region of the state space where it is optimal to stop, and Ross (1969) and Chew (1973) both approximate an optimal stopping rule. Further, Chew (1973), which studies the model of Chew (1967) with general search times, shows that if $cq_i \geq \sum_{j=1}^{n} t_j(2 - q_j)$ for $i = 1, \ldots, n$, where

c is the stopping cost, then it can be determined whether it is optimal to stop or not by looking n searches ahead using (1.2.1).

Finally, before this chapter optimally stops, we consider an example of Kan (1974) with two boxes. Let p be the probability that the target is in box 1. Note that equality in (1.2.1), attained by some $p_0 \in (0, 1)$, maximises uncertainty about where to search next. Therefore, similarly to the waiting model in Flesch et al. (2009), it is intuitive that there would exist $p_l \leq p_u$ such that it is optimal to stop for $p \in (p_l, p_u)$, to search box 1 for $p \geq p_u$ and to search box 2 for $p \leq p_l$. However, Kan (1974) shows by counterexample that such p_l, p_u do not always exist. The conjectured reason is that, whilst the uncertainty in which box to search next grows as p moves closer to p_0, so does the information that the searcher gains from carrying on with the search. This idea will be revisited in Chapter 3 of this thesis.

Chapter 2

Search and Multi-Armed Bandits

In this chapter, a multi-armed bandit (MAB) will be introduced, and, in Section 2.1, it will be shown that both the classic time-independent and time-dependent search problems discussed in Chapter 1 can be formulated as an MAB. In Section 2.2, the Gittins index policy which optimally solves the search MAB will be derived and shown to agree with (and generalise in the time-dependent case) the classic-search-problem solutions from the literature from Chapter 1. Finally, Section 2.3 introduces the focus of Chapter 3, a search problem with two search modes, reviewing the relevant literature, and discussing to what extent Gittins index theory still applies. For the latter discussion, the generalised time-dependent solution obtained by a Gittins index policy in Section 2.2 is crucial.

The MAB problem, first studied by Thompson (1933), takes its name from slot machines in casinos called one-armed bandits. When a one-armed bandit is played, the player receives a reward; in the MAB problem, multiple one-armed bandits (called arms) are available, and the player can choose one arm to play at each time period. The aim is to maximise the total reward received over a known period of time.

Several different assumptions can be made about the generation of rewards. As in Thompson (1933), each arm may have its own unknown reward distribution from which its rewards are drawn. Each play provides information about the corresponding arm's reward distribution; therefore, the MAB problem becomes a trade-off between exploring unfamiliar reward distributions and using prior knowledge to exploit more-lucrative arms. There is a vast literature on such bandits, particularly within the last 15 years; for a recent overview, see Slivkins (2019).

Alternatively, each arm may move through a set of states, with its reward dependent entirely upon its current state which is known to the player. The player maximises the total reward over an infinite length of time, with rewards discounted as time progresses to make those received near the start more valuable. Since there is no new information to be gained about each arm, the focus is entirely on exploiting the largest rewards at the earliest time possible. Such bandits, which are a special type of Markov decision process (MDP), will be the focus of this chapter. Puterman (2014) provides a comprehensive study of MDPs; however, for the purposes of this thesis, any background knowledge of MDPs is not required. The solution to any MDP-based MAB, discovered in the 1970s, will be discussed further in Section 2.2. The following formulates an MDP-based MAB with the aid of Gittins et al. (2011), a monograph of such bandit problems.

At any time, a one-armed bandit B is in a known state within a known state space E. Any time $d \in \{0, 1, 2, \dots\}$ is a *decision time* at which the player selects either a freeze or play action. Freezing B at time d leaves its state unchanged until the next decision time $d + 1$ and accrues no reward. Playing B at time d accrues an immediate, known, discounted reward $\theta^d r(x(d))$, where $\theta \in (0, 1)$ is a known discount factor, $x(d)$ is the known state of B at time d, and r is a known function from E to $\mathbb{R}$. Under the play action, the state of B at time $d+1$ is determined by known transition probabilities dependent only on $x(d)$; therefore, only the current state of B affects its future, so the Markov property is satisfied.

An MAB presents the player with n one-armed bandits $B_1, \dots, B_n$, each with the same discount factor θ, but different known state spaces $E_1, \dots, E_n$ and reward functions $r_1, \dots, r_n$. Bandit B_i is referred to as the ith *arm* of the MAB. At each decision time $d \in \{0, 1, 2, \ldots\}$, the player chooses one arm to play and $n-1$ arms to freeze. Write $x_i(d) \in E_i$ for the state of arm i at time $d \in \{0, 1, 2, \ldots\}$, so the state of the MAB at time d is $(x_1(d), \ldots, x_n(d))$, and the state space is $E_1 \times \cdots \times E_n$. If arm i is played at decision time d, the reward is $\theta^d r_i(x_i(d))$, with the aim of the player to maximise the total reward over an infinite horizon.

2.1 Search Problems as Multi-Armed Bandits

An MAB formulation has been used to tackle many problems, such as clinical trials in Thompson (1933), experimental design in Robbins (1952), and web design in many more recent papers. MDP-based bandits have been applied to job-scheduling problems; see Jacko (2010) for a survey. Search was added to the list of MAB applications when a comment by Kelly in Gittins (1979), a celebrated work on MDP-based bandits discussed further in Section 2.2, showed how the classic time-dependent search problem introduced in Chapter 1 can be formulated as an MDP-based MAB. In this section, like Kelly, we derive the MAB for the classic time-independent problem with unit search times, then generalise to the classic time-dependent problem.

Recall, in the classic time-independent problem, the target is located in box i with known probability p_i, with a search of box i taking known search time $t_i > 0$ and finding the target, if in box i, with known detection probability $q_i \in (0, 1)$, $i = 1, \ldots, n$. In the unit-search-time problem, $t_i = 1$ for $i = 1, \ldots, n$. The aim of the searcher is to minimise the expected search time by consecutively searching boxes. A search policy is, therefore, an infinite, ordered sequence of boxes to search. Write $V(\xi)$ for the expected search time under search policy ξ, so an optimal search policy minimises $V(\xi)$.

For $i = 1, \ldots, n$, write $V_i(\xi)$ for the expected search time under ξ conditional on the target being in box i, so

$$V(\xi) = \sum_{i=1}^{n} p_i V_i(\xi).$$

Write $c_{i,\xi}(m)$ for the time at which the mth search of box i ends following ξ, $i = 1, \ldots, n$, $m \in \mathbb{Z}^+ \equiv \{1, 2, \ldots\}$. If the target is in box i, detection must occur at time $c_{i,\xi}(m)$ for some $m \in \mathbb{Z}^+$. To be found at time $c_{i,\xi}(m)$, the first $m-1$ searches of box i must fail and the mth succeed, which occurs with probability $q_i(1-q_i)^{m-1}$; therefore

$$V_i(\xi) = \sum_{m=1}^{\infty} c_{i,\xi}(m) q_i (1-q_i)^{m-1},$$

and hence

$$V(\xi) = \sum_{i=1}^{n} p_i \left(\sum_{m=1}^{\infty} c_{i,\xi}(m) q_i (1-q_i)^{m-1} \right).$$

Since

$$\frac{1-\theta^{c_{i,\xi}(m)}}{1-\theta} = 1 + \theta + \theta^2 + \ldots + \theta^{c_{i,\xi}(m)-1} \to c_{i,\xi}(m) \quad \text{as} \quad \theta \to 1, \tag{2.1.1}$$

for θ close to 1, we have

$$\begin{aligned} V(\xi) &\approx \sum_{i=1}^{n} p_i \sum_{m=1}^{\infty} \frac{1-\theta^{c_{i,\xi}(m)}}{1-\theta} q_i (1-q_i)^{m-1} \\ &= \sum_{i=1}^{n} \frac{p_i}{1-\theta} \sum_{m=1}^{\infty} q_i (1-q_i)^{m-1} - \sum_{i=1}^{n} \sum_{m=1}^{\infty} \frac{\theta^{c_{i,\xi}(m)}}{1-\theta} p_i q_i (1-q_i)^{m-1}. \end{aligned} \tag{2.1.2}$$

Since the first term in (2.1.2) does not depend on ξ, and $1-\theta$ is constant, minimising (2.1.2) over all search policies ξ is equivalent to maximising

$$\sum_{i=1}^{n} \sum_{m=1}^{\infty} \theta^{c_{i,\xi}(m)} p_i q_i (1-q_i)^{m-1}. \tag{2.1.3}$$

We may view (2.1.3) as a discounted sum of rewards from playing the following MAB. Each arm represents a box, with a play of an arm equivalent to a search of its corresponding box. Since search times are unit, the decision times correspond with $\mathbb{N} \equiv \{0, 1, \ldots\}$. The state of each arm is the number of previous searches of its corresponding box. Therefore, the state space is $\mathbb{N}^n$ and state transitions are deterministic, namely, after an unsuccessful search of any box, its arm's state increases by 1. For $i = 1, \ldots, n$, the reward function $r_i : \mathbb{N} \to \mathbb{R}$ satisfies $r_i(m) = p_i q_i (1-q_i)^m$ for all $m \in \mathbb{N}$. In other words, the reward before discounting obtained by searching box i for a $(m+1)^{\text{st}}$ time is the probability of finding the object on the $(m+1)^{\text{st}}$ search of box i. The discount factor θ comes from the approximation in (2.1.1), with the original search problem retained by letting $\theta \to 1$. Since the target's discovery may only occur at the end of a search of some box, unlike the classic formulation of the MDP-based MAB, the reward from the arm played at decision time d is actually attained at decision time $d+1$ when the search of the corresponding box finishes. However, as will be seen in Section 2.2, this adjustment does not affect the relevance of the MAB theory.

Maximising (2.1.3) is also equivalent to minimising the expected search time when search times are arbitrary. However, in the associated MAB, the decision times no longer correspond with $\mathbb{N}$; the gap between successive decision times d and d' depends on the box i searched at d, with $d' = d + t_i$. As a result, the MAB is no longer an MDP but a semi-Markov decision process (SMDP) which, in its full generality, allows the gaps between decision times to be random variables. For a general formulation of a SMDP, see Baykal-Gürsoy (2010), but such knowledge is not required here.

In his comment in Gittins (1979), Kelly also discusses the classic time-dependent problem where, recall, the kth search of box i takes known search time $t_i(k) > 0$ and finds the target, if in box i, with known detection probability $q_i(k) \in (0,1)$, $i = 1, \ldots, n$, $k \in \mathbb{Z}^+$. It is assumed that the $t_i(k)$ do not degrade to 0 as $k \to \infty$. Where $c_{i,\xi}(m)$ is again the time at which the mth search of box i ends following search policy ξ, $i = 1, \ldots, n$, we may express $V(\xi)$, the expected search time under ξ, as

$$V(\xi) = \sum_{i=1}^{n} p_i \left(\sum_{m=1}^{\infty} c_{i,\xi}(m) q_i(m) \prod_{k=1}^{m-1} (1 - q_i(k)) \right). \tag{2.1.4}$$

Repeating the analysis for the time-independent case, it follows that minimising the expected search time in (2.1.4) is equivalent, as $\theta \to 1$, to maximising

$$\sum_{i=1}^{n} \sum_{m=1}^{\infty} \left(\theta^{c_{i,\xi}(m)} p_i q_i(m) \prod_{k=1}^{m-1} (1 - q_i(k)) \right), \tag{2.1.5}$$

the time-dependent version of (2.1.3), which may also be viewed as a discounted sum of rewards from a similar MAB.

2.2 The Gittins Index for a Search Problem

A policy for an MAB is a rule which decides which arm, if any, to play at any decision time, with an optimal policy maximising the total, expected discounted reward over an infinite horizon. In 1974, John Gittins, alongside Jones, found an index policy optimal for any MDP-based MAB, a major breakthrough in the area. An index policy calculates a real number, called an index, for each arm dependent only on the arm's current state. The index summarises the value of playing the arm in that state, so the index policy plays any arm with a maximal index. The formula for the index derived by Gittins, later coined the *Gittins index*, was first presented and proven to be optimal in Gittins and Jones (1974). In the following years, Gittins, Glazebrook, Jones and Nash explored the breadth and significance of the result in solving previously-intractable problems (for example, see Nash (1973), Glazebrook (1976a,b), Gittins and Glazebrook (1977), Gittins and Jones (1979)); the celebrated Gittins (1979) collated these works. Many elegant alternative proofs have since shown the optimality of the Gittins index. Among the most notable are the dynamic-programming approach of Whittle (1980), the fair-charge reasoning of Weber (1992), and the interchange argument of Tsitsiklis (1994).

This section will first derive the Gittins index for the classic time-independent search problem, showing its equality to the index policy in (1.2.1) found independently to be optimal by Blackwell (reported in Matula (1964)), Norris (1962), Bram (1963) and Black (1965). Second, the Gittins index derivation will be adjusted to the classic time-dependent search problem, allowing a generalisation of the index policy of Stone (1975) in (1.2.3) previously restricted to a special case. For a derivation of the Gittins index for a general MAB, see Chapter 2 of Gittins et al. (2011).

Gittins Index for the Time-Independent Search Problem Consider an MAB with two arms and discount factor θ. The first arm, s_λ, is parameterised by $\lambda > 0$

and has just one state. If s_λ is played at decision time d, a discounted reward $\theta^{d+1}\lambda$ is received at the next decision time $d+1$. As described in Section 2.1, the second arm B corresponds to searching a box with detection probability q, search time t and probability p of containing the target. Therefore, B moves through states in $\mathbb{N}$ which represent the number of previous searches, and, if played at decision time d in state $m \in \mathbb{N}$, discounted reward $\theta^{d+t}pq(1-q)^m$ is received at the next decision time $d+t$. The following derives the Gittins index for B in a given state.

As usual, at each decision time, the player must choose one arm to play whilst freezing the other. Since s_λ has only one state, once it becomes optimal to play s_λ and freeze B, it remains optimal to play s_λ forevermore. Therefore, an optimal policy will play arm B until some *retirement time* τ when it first becomes optimal to play s_λ, then play s_λ eternally. For an arbitrary second arm with stochastic changes of state, τ is a random variable. However, since the state changes of our second arm B are deterministic, τ is also deterministic.

Suppose the player has not yet retired and B is in state $m \in \mathbb{N}$; therefore, the current time is mt. If the player retires immediately, the total future sum of rewards is given by

$$\sum_{j=1}^{\infty} \lambda\theta^{mt+j} = \frac{\lambda\theta^{mt+1}}{1-\theta}. \tag{2.2.1}$$

On the other hand, if the player does not retire immediately, and instead retires at the future decision time that will maximise the total reward, the total future sum of rewards is given by

$$\begin{aligned} &\sup_{b\in\mathbb{Z}^+} \left\{ \sum_{j=m+1}^{m+b} \theta^{jt}pq(1-q)^{j-1} + \sum_{j=1}^{\infty} \theta^{(m+b)t+j}\lambda \right\} \\ &= \sup_{b\in\mathbb{Z}^+} \left\{ \frac{\theta^{(m+1)t}pq(1-q)^m(1-\theta^{bt}(1-q)^b)}{1-\theta^t(1-q)} + \frac{\lambda\theta^{(m+b)t+1}}{1-\theta} \right\}. \end{aligned} \tag{2.2.2}$$

The Gittins index for B in state m is the value of the parameter λ such that the player is indifferent between playing B in state m (retiring at some future decision time) and playing s_λ (retiring now), in other words, the value of λ that equates (2.2.1) and (2.2.2), given by

$$\sup_{b\in\mathbb{Z}^+}\left\{\frac{(1-\theta)\theta^{t-1}pq(1-q)^m(1-\theta^{bt}(1-q)^b)}{(1-\theta^{bt})(1-\theta^t(1-q))}\right\}. \tag{2.2.3}$$

To calculate the supremum, we may remove multiplicative terms which do not depend upon b, leaving

$$\sup_{b\in\mathbb{Z}^+}\left\{\frac{1-(\theta^t(1-q))^b}{1-(\theta^t)^b}\right\}. \tag{2.2.4}$$

To evaluate (2.2.4), note that

$$\frac{1-x}{1-y} > \frac{(1-x)(x^{b-1}+x^{b-2}+\ldots+x+1)}{(1-y)(y^{b-1}+y^{b-2}+\ldots+y+1)} = \frac{1-x^b}{1-y^b}$$

for any $b \in \{2,3\ldots\}$ and $0 < x < y < 1$. Taking $y = \theta^t$ and $x = \theta^t(1-q)$ shows that (2.2.4), and hence also (2.2.3), is always attained at $b = 1$ for any $0 < q, \theta < 1$ and $t > 0$. Therefore, the Gittins index for B in state m in (2.2.3), which we will denote $v(B,m)$, satisfies

$$v(B,m) = \frac{(1-\theta)\theta^{t-1}pq(1-q)^m}{1-\theta^t}. \tag{2.2.5}$$

To interpret the Gittins index, we split (2.2.5) into a ratio of two terms. The first term is $\theta^t pq(1-q)^m$, the discounted reward received after the next ($(m+1)^{\text{st}}$) search of the box if the current time mt is reset to 0. For $t \in \mathbb{Z}^+$, the second term satisfies

$$\frac{\theta(1-\theta^t)}{1-\theta} = \sum_{j=1}^{t}\theta^j,$$

interpreted as the discounted time from 0 until a search of the box is completed at time t. Therefore, since both terms in the ratio are discounted, (2.2.5) may be interpreted as the undiscounted, average rate of reward from time 0 to t received by playing B, with such an interpretation still valid for a general $t > 0$. The undiscounted, average rate of reward from s_λ over any time interval is λ, satisfying our interpretation that setting $\lambda = v(B,m)$ makes the player indifferent between playing B in state m and playing s_λ.

Now consider the MAB associated with the classic time-independent search problem with n arms, $B_1,\ldots B_n$, corresponding to n boxes, where box i has detection probability q_i, search time t_i, and contains the target with probability p_i. Suppose B_i is in state $m_i \in \mathbb{N}$, so m_i searches have been made of box i, $i = 1, \dots, n$. Then, first shown to be optimal by Gittins and Jones (1974), a Gittins index policy next plays any arm j satisfying

$$j = \underset{i=1,\dots,n}{\arg\max}\, v(B_i, m_i), \tag{2.2.6}$$

with $v(B_i, m_i)$ the Gittins index of arm B_i, $i = 1, \dots, n$, taking the form in (2.2.5).

Recall the classic time-independent search problem is retained by letting $\theta \to 1$. By (2.1.1), under this limit, the Gittins index policy in (2.2.6) next searches any box j satisfying

$$j = \underset{i=1,\dots,n}{\arg\max} \frac{p_i(1-q_i)^{m_i} q_i}{t_i},$$

equivalent to the optimal index policy in (1.2.1) discovered by multiple authors in the 1960s. As discussed in Chapter 1, the index is myopic as it selects a box with a maximal probability of detection per unit time for its next search, equivalent to a maximal average rate of reward in the associated MAB.

For a general MAB, the Gittins index is not necessarily myopic; in other words, (2.2.3) is not always attained by $b = 1$, so the average rate of reward that forms an arm's Gittins index may look ahead more than one play of the arm into the future. In this instance, Gittins indices can be difficult to calculate. The supremum in (2.2.3) is always attained at $b = 1$ for the search bandit B since the undiscounted reward received from B decreases monotonically (by a multiplicative factor of $1-q$) with each search. Therefore, the average rate of reward looking ahead for any $b \in \{2, 3, \dots\}$ consecutive searches of a box will always be smaller than looking ahead just one search.

Gittins Index for the Time-Dependent Search Problem Recall from Section 2.1 that the classic time-dependent search problem may also be formulated as an MAB and hence is also optimally solved via a Gittins index. If B corresponds to searching a box whose kth search takes time $t(k)$ and has detection probability $q(k)$, $k \in \mathbb{Z}^+$, then the time-independent derivation of the Gittins index is adjusted as follows.

If the player has not yet retired and B is in state $m \in \mathbb{N}$, the current time is $a \equiv \sum_{v=1}^{m} t(v)$. Similarly to (2.2.1), retiring immediately accrues total future reward

$$\frac{\lambda\theta^{a+1}}{1-\theta}. \tag{2.2.7}$$

Retiring at the future decision time that maximises total future reward accrues

$$\sup_{b\in\mathbb{Z}^+}\left\{\frac{\lambda\theta^{a+1+\sum_{u=m+1}^{m+b}t(u)}}{1-\theta}+p\theta^a\sum_{u=m+1}^{m+b}\left(\theta^{\sum_{v=m+1}^{u}t(v)}q(u)\prod_{v=1}^{u-1}(1-q(v))\right)\right\}, \tag{2.2.8}$$

the time-dependent version of (2.2.2). The Gittins index for B in state m is the value of λ equating (2.2.7) and (2.2.8), namely

$$\sup_{b\in\mathbb{Z}^+}\left\{\frac{(1-\theta)p\sum_{u=m+1}^{m+b}\theta^{\sum_{v=m+1}^{u}t(v)}q(u)\prod_{v=1}^{u-1}(1-q(v))}{\theta\left(1-\theta^{\sum_{u=m+1}^{m+b}t(u)}\right)}\right\}.$$

Therefore, in the time-dependent search problem with n boxes retained as $\theta \to 1$, after $m_i \in \mathbb{N}$ searches of box i have been made, its Gittins index is given by

$$p_i\sup_{b\in\mathbb{Z}^+}\left\{\frac{\sum_{u=m_i+1}^{m_i+b}q_i(u)\prod_{v=1}^{u-1}(1-q_i(v))}{\sum_{u=m_i+1}^{m_i+b}t_i(u)}\right\}, \tag{2.2.9}$$

$i = 1, \ldots, n$.

The supremum in (2.2.9) is attained at $b = 1$ if the undiscounted reward from the arm associated with box i decreases monotonically with each search. Unlike the time-independent problem, decreasing rewards are not guaranteed for boxes in the time-dependent problem. If the $(m_i + 2)^{\text{nd}}$ search of box i is more appealing to the searcher than the $(m_i + 1)^{\text{st}}$ because of a superior detection probability and search time combination, then the Gittins index for box i in state m_i may look ahead more than one search. Otherwise, substituting $b = 1$ into (2.2.9) gives

$$\frac{p_i q_i(m_i+1)\prod_{v=1}^{m_i}(1-q_i(v))}{t_i(m_i+1)},$$

precisely the optimal index in (1.2.3) derived by Stone (1975) under the condition that (1.2.3) is decreasing in m_i (equivalent to monotonically-decreasing rewards from box i). Therefore, the Gittins index in (2.2.9) extends the result of Stone (1975) to the fully-general time-dependent search problem.

The Gittins index solution to the time-dependent problem is useful in the next section, where a second search mode is introduced to the classic time-independent problem.

2.3 Search Problems with a Choice of Search Mode

This section extends the classic time-independent search problem to allow the searcher a choice of two search modes, fast or slow, when searching any box. A search with the fast mode takes less time, but the slow mode has a greater chance of finding the target, creating a trade-off between the accuracy and speed of the search.

Search problems with multiple search modes are of increasing importance due to advanced technologies resulting in several ways to conduct searches. There may be several choices of search agents, such as humans, animals, and robots. For example, a search squad can use a dog (fast mode) or a metal detector (slow mode) to locate a hidden bomb. Further, for any one such agent, for example a robot, there may be multiple settings on the travel speed or the sensor mode. Notwithstanding this increased relevance, such problems have received little attention in the academic literature. The limited literature is reviewed in Section 2.3.1, before Section 2.3.2 introduces the two-mode search problem and discusses immediate connections to the classic single-mode problem.

2.3.1 Literature Review

Search problems with adjustable or multiple search modes in the literature are few and far between. The three papers described below model search in continuous time; in other words, the searcher may switch the location of their search effort at any time.

In Posner (1963) and Persinger (1973), the objective is to minimise the expected search time. Posner (1963) studies a search for a satellite in the sky using a radar beam. The sky is discretised into 'cells', and a preliminary search highlights cells most likely to contain the satellite to be prioritised for further investigation. During the preliminary search, the searcher may examine multiple cells at once by adjusting the width of the radar beam at the cost of less information gain per cell. A preliminary search with the narrowest possible beam width is found to be best. In Persinger (1973), the target is hidden in a subset of $\mathbb{R}^n$. The searcher has two sensors available, but can only use one at each point in time. The searcher can switch between sensors instantaneously, with the choice of sensor affecting the speed and detection capabilities of the searcher. An optimal search plan is found for a specific example in $\mathbb{R}^2$. In Nakai (1987), the objective is to maximise the probability of finding a target hidden in n boxes before a known deadline d. However, at any time, the searcher can either allocate search effort, or improve the detection rate in one or more of the boxes. Nakai (1987) shows it is optimal to improve detection rates up until some time $d' < d$ and thereafter concentrate solely on allocating search effort.

More recently, Alpern and Lidbetter (2015) studies a search on a network with a fast and a slow mode, with the latter's perfect detection simplifying the analysis. However, the search is two sided; in other words, the target is an intelligent hider who actively avoids detection. Therefore, the game-theoretic methodology used does not apply to one-sided search, so discussion of Alpern and Lidbetter (2015) is deferred to Part II of this thesis, which focuses on two-sided search.

Shechter et al. (2015) and Kadane (2015) consider a search for a target in n boxes with multiple search modes available. In Shechter et al. (2015), the modes are fast and slow, but both modes have perfect detection and unit search time, being instead differentiated by their probability of a 'failure'. If a fast search finds the target, the target may become damaged, resulting in a failure. When making a slow search, regardless of whether the target is found or not, the searcher may be hit by enemy fire, which also results in a failure. The searcher's goal is to minimise the probability that the search ends in a failure. Shechter et al. (2015) finds an optimal policy begins by searching boxes with a low probability of containing the target using the fast mode, before searching those with a higher probability of containing the target using the slow mode.

Kadane (2015) allows an arbitrary number of modes per box, each mode having its own detection probability q and search time t. Maximising the probability of detection before a known deadline is studied for a single-box problem. To allow the search to continue right up to the deadline, the last search of a box at a mode with parameters q and t is allowed to be curtailed at time st with detection probability $1-(1-q)^s$ for any choice of $s \in (0,1]$. Kadane (2015) shows it is optimal to always search the box using the same mode, that with the largest value of $-\log(1-q)/t$. The detection probability and search time for a mode are also allowed to depend on its number of prior uses, as long as, for each mode, $-\log(1-q)/t$ decreases in the number of uses. Under this assumption, it is optimal to use the mode with the largest $-\log(1-q)/t$ given its current number of uses, continuing until the deadline by curtailing the last search. Kadane (2015) formulates the use-dependent problem with an arbitrary number of boxes as a convex optimisation problem.

Returning to a single box, Kadane (2015) describes a branch-and-bound algorithm to find a solution when the last search cannot be curtailed. As discussed in Section 1.2.2, when the deadline is not guaranteed to coincide with the end of a search of a box, maximising the probability of detection before the deadline becomes significantly harder, since it may be optimal to use an inferior mode to minimise time spent not searching just before the deadline.

The metric $-\log(1-q)/t$, central to Kadane's deadline results, also appears in Chapter 3 of this thesis, where the objective is to minimise the expected search time in Kadane's use-independent model with two modes per box. Section 3.4.4 explains the difference between the roles of $-\log(1-q)/t$ under each objective.

As mentioned in the motivation section of this thesis, Kadane (2015) also considered minimising the expected search time, but we believe his conclusions to be incorrect; more detail is provided in the next subsection, where the two-mode search problem studied in Chapter 3 is introduced.

2.3.2 A Two-Mode Time-Independent Search Problem

We formulate the two-mode search problem as follows. As in the classic time-independent problem, the target lies in box i with probability p_i, $i = 1, \ldots, n$, $\sum_{i=1}^{n} p_i = 1$, and the aim of the searcher is to minimise the expected search time by consecutively searching boxes. Unlike the classic problem, each box has two search modes, fast and slow. For every box, each mode has its own detection probability and search time; let the slow (resp. fast) search time of box i be $t_{i,s}$ (resp. $t_{i,f}$), and the slow (resp. fast) detection probability of box i be $q_{i,s}$ (resp. $q_{i,f}$). The slow mode has a larger detection probability, but takes longer to complete a search, so $q_{i,s} > q_{i,f}$ and $t_{i,s} > t_{i,f}$.

The search problem becomes more challenging if two search modes are available for each box, since, in addition to deciding where to look next, the searcher needs to choose a search mode. Therefore, the search may no longer be modelled as an MAB but instead as a *superprocess*, an extension of an MAB discussed in Chapter 4 of Gittins et al. (2011) with multiple ways (called actions) to play each arm. In the superprocess representing the two-mode search problem, each arm represents a box and has two actions, the fast search and the slow search. However, superprocess theory is not required in this thesis; Chapter 3 studies the two-mode search problem via a direct approach, exploiting only the MAB theory in Section 2.2 as follows.

Suppose, for each box i, a sequence of modes $A_i \equiv (a_{i,1}, a_{i,2}, \ldots)$ is fixed such that the kth search of box i must be made using mode $a_{i,k}$; remaining is a classic time-dependent problem with $q_i(k) = q_{i,a_{i,k}}$ and $t_i(k) = t_{i,a_{i,k}}$, $i = 1, \ldots, n$, $k \in \mathbb{Z}^+$. As shown in Sections 2.1 and 2.2, the time-dependent problem may be formulated as an MAB and solved using the Gittins index in (2.2.9). Therefore, the two-mode problem is reduced to finding an optimal sequence of modes for each box, since, given optimal mode sequences, searches of individual boxes are optimally interlaced using Gittins indices; more detail will be provided in Section 3.2 of Chapter 3. Determining conditions for the optimal mode sequence of a box to contain just one search mode is the main topic of Chapter 3.

In Gittins et al. (2011), a candidate index policy (CI) for a superprocess is proposed, based upon the Gittins index of the MAB recovered if the way to play each arm (i.e., via which actions) is pre-determined. A sufficient condition for the optimality of CI is derived in Whittle (1980) and shown also to be necessary in Glazebrook (1982). In CI, every arm has an index for each of its actions, with CI choosing a maximal index over all arm-action combinations.

For the two-mode search problem, CI reduces to the following policy. Suppose $m_{i,f}$ fast and $m_{i,s}$ slow searches have previously been made of box i, $i = 1, \ldots, n$, so the posterior probability p_i' that the target is in box i satisfies

$$p_i' = \frac{p_i(1-q_{i,f})^{m_{i,f}}(1-q_{i,s})^{m_{i,s}}}{\sum_{j=1}^{n} p_j(1-q_{j,f})^{m_{j,f}}(1-q_{j,s})^{m_{j,s}}}.$$

For each box, calculate

$$\frac{p_i' q_{i,f}}{t_{i,f}} \quad \text{and} \quad \frac{p_i' q_{i,s}}{t_{i,s}} \tag{2.3.1}$$

to create $2n$ indices across all boxes. CI chooses any box-mode combination with an index maximal amongst all $2n$.

Inspection of (2.3.1) shows that the value of p_i' has no bearing on which is the larger of the two indices. Therefore, CI may be implemented such that only one mode is ever used to search box i, fast if $q_{i,f}/t_{i,f} > q_{i,s}/t_{i,s}$, slow if $q_{i,f}/t_{i,f} < q_{i,s}/t_{i,s}$, and either fast or slow if $q_{i,f}/t_{i,f} = q_{i,s}/t_{i,s}$. Kadane (2015), which extends the two-mode problem to an arbitrary number of modes per box, concludes that CI is optimal. If true, for each box, all but one mode can be optimally discarded, and any multiple-mode extension to the classic time-independent search problem is no harder than the single-mode original. However, the analysis in Chapter 3 of this thesis, published in Clarkson et al. (2020), disagrees with Kadane (2015), showing that an optimal search policy may require both modes of a box, with the optimal choice of mode depending on the searcher's current (posterior) beliefs about the target's location.

Kadane's proof that CI is optimal attempts to adapt to multiple modes the argument in Chapter III, Section 5 of Ross (1983), a simple proof of the optimality of the Gittins index in the classic time-independent problem with a single search mode per box. Ross' proof shows that any search sequence beginning with a search of box i immediately followed by a search of box j with $p_i q_i / t_i < p_j q_j / t_j$ can be improved by switching the order of boxes i and j. Therefore, any search sequence can be improved until the box with maximal index is searched next.

However, in the multiple-mode problem, where an element of a search sequence must specify both a box to search and a mode at which to search it, there may be many suboptimal search sequences that cannot be improved by switching consecutive elements. For example, if every optimal search sequence involves both the fast and slow mode of box i, then no search sequence containing only the fast mode for box i can be improved by switching consecutive elements to obtain an optimal sequence. In the following simple counter example, originally constructed by my supervisor Kyle Lin, CI is not optimal, and Kadane's proof shown to fail.

Example 2.3.1 Consider the following two-box problem. Box 1 contains the target with probability $p = 0.6$ and has the usual search parameters $q_f = 0.4$, $q_s = 0.64$, $t_f = 1$ and $t_s = 1.64$. Box 2 has only one search mode with search time $t_2 = 2.5$ and detection probability 1. Clearly any optimal search sequence looks in box 2 only once, since after a failed search of box 2, the target is certain to be in box 1.

Write f (resp. s) for a fast (resp. slow) search of box 1, and 2 for a search of box 2. The search sequence under CI is $(f, 2, f, f, \ldots)$, which uses only the fast mode of box 1 since $q_f / t_f > q_s / t_s$, and, by the single-mode result, cannot be improved by interchanging two consecutive searches. To calculate the expected search time under $(f, 2, f, f, \ldots)$, we condition on the target's location. If the target is in box 1 (resp.

box 2), then the expected search time is

$$t_f + (1 - q_f)\left(t_2 + \frac{t_f}{q_f}\right) = 4 \quad (\text{resp. } t_f + t_2 = 3.5)\,.$$

Therefore, the expected search time is

$$0.6 \times 4 + 0.4 \times 3.5 = 3.8.$$

Now we calculate the expected search time under the search sequence $(s, 2, f, f, \ldots)$. If the target is in box 1 (resp. box 2), the expected search time is

$$t_s + (1 - q_s)\left(t_2 + \frac{t_f}{q_f}\right) = 3.44 \quad (\text{resp. } t_s + t_2 = 4.14)\,,$$

so the expected search time is

$$0.6 \times 3.44 + 0.4 \times 4.14 = 3.72.$$

Therefore, the sequence under CI, namely $(f, 2, f, f, \ldots)$, is suboptimal despite no improvement being possible via an interchange of consecutive elements.

Example 2.3.2 Consider the following two-box problem. Box 1 contains the target with probability $p = 0.6$ and has the usual search parameters $q_f = 0.4$, $q_s = 0.64$, $t_f = 1$ and $t_s = 1.64$. Box 2 has only one search mode with search time $t_2 = 3$ and detection probability 1. Clearly any optimal search sequence looks in box 2 only once, since after a failed search of box 2, the target is certain to be in box 1.

Write f (resp. s) for a fast (resp. slow) search of box 1, and 2 for a search of box 2. The search sequence under CI is $(f, 2, f, f, \ldots)$, which uses only the fast mode of box 1 since $q_f/t_f > q_s/t_s$, and, by the single mode result, cannot be improved by interchanging two consecutive searches. To calculate the expected search time under $(f, 2, f, f, \ldots)$, we condition on the target's location. If the target is in box 1 (resp. box 2), then the expected search time is

$$t_f + (1 - q_f)\left(t_2 + \frac{t_f}{q_f}\right) = 4.3 \quad (\text{resp. } t_f + t_2 = 4)\,.$$

Therefore, the expected search time is

$$0.6 \times 4.3 + 0.4 \times 4 = 4.18.$$

Now we calculate the expected search time under the search sequence $(s, 2, f, f, \ldots)$. If the target is in box 1 (resp. box 2), the expected search time is

$$t_s + (1 - q_s)\left(t_2 + \frac{t_f}{q_f}\right) = 3.62 \quad (\text{resp. } t_s + t_2 = 4.64)\,,$$

so the expected search time is

$$0.6 \times 3.62 + 0.4 \times 4.64 = 4.028.$$

Therefore, the sequence under CI, namely $(f, 2, f, f, \ldots)$, is suboptimal despite no improvement being possible via an interchange of consecutive elements.

Suppose there is only one box containing the target with probability 1. The minimal expected search time μ^* satisfies

$$\mu^* = \min[t_f + \mu^*(1 - q_f), t_s + \mu^*(1 - q_s)],$$

from which it follows that

$$\mu^* = \min\left[\frac{t_f}{q_f}, \frac{t_s}{q_s}\right]. \tag{2.3.2}$$

Modelling the number of searches needed to find the object as a geometric distribution with success probability q_f (resp. q_s) shows that always searching fast (resp. slow) finds the target in expected time t_f/q_f (resp. t_s/q_s); it follows from (2.3.2) that CI is optimal.

However, when the box is one of many, it turns out that while it is indeed optimal to always search the box slow if $q_s/t_s \geq q_f/t_f$, it is *not* always optimal to search the box fast if $q_s/t_s < q_f/t_f$, so CI can be suboptimal. The CI policy (renamed as the detection rate heuristic policy) is studied as a heuristic policy in Chapter 3, where the focus is on the two-mode search problem.

Chapter 3

Fast or Slow: Search in Discrete Locations with Two Search Modes

The content of this chapter is Clarkson et al. (2020) with several changes detailed at the end of this introduction. The aim of this chapter is to study the two-mode search problem introduced in Section 2.3.2. Throughout this chapter, since the hidden target is immobile and unintelligent, it will be referred to as the *object*.

Section 3.1 formulates the two-mode search problem and, with appeal to Chapter 2, shows that Gittins indices reduce the two-mode problem to the determination of optimal sequences of search modes for each box. In Section 3.2, we give a sufficient condition for each mode to dominate the other mode in the same box, such that the latter need never be used in an optimal mode sequence. This analysis both solves the problem in some special cases, and yields insightful bounds on the optimal expected search time in general. Further insight is derived in Section 3.3 by the study of some two-box problems in which one box has just one search mode with perfect detection. Section 3.4 presents a range of heuristic policies with suboptimality bounds for the general two-mode problem based on the analyses of Sections 3.2 and 3.3. Section 3.5 demonstrates the performance of these heuristics in an extensive numerical study.

The changes to Clarkson et al. (2020), aside from minor changes in wording and references to Chapters 1 and 2 of this thesis, are as follows. Section 1 (Introduction) of Clarkson et al. (2020) is omitted, since its material has already been covered in Section 2.3.2 of this thesis. Section 3.5 (Dominance among Multiple Search Modes) and Section 5.4 (Heuristic Policies and Suboptimality Bounds for Multiple Search Modes) of Clarkson et al. (2020) are deferred to Chapter 4 of this thesis. Aside from the proofs in Appendices B.1 and B.2 of this thesis, all work in the online appendices of Clarkson et al. (2020) has been returned to the main body.

Section 3.1 is a shortened version of Section 2 of Clarkson et al. (2020), since most material has been covered in Sections 2.2 and 2.3.2 of this thesis. Figure 3.2.2 is added to support the introduction of the continuous-sweeping search in Section 3.2.1. The proofs of Lemma 3.2.2 and Theorem 3.2.4 are adapted so the latter can specify when the slow mode of a box is uniquely optimal. Example 3.2.6 and its supporting Appendix A are added to demonstrate a case where the slow mode is not uniquely optimal. With no change to its proof required, the statement of Theorem 3.2.8 is strengthened, specifying when the fast mode of a box is uniquely optimal. A more detailed explanation is provided regarding the lower bounds in Section 3.2.4. Figure 3.4.3 is added to Section 3.4.2 to provide a second example of the BT-heuristic threshold in action. Section 3.4.4 is added to link the immediate and future benefit introduced in Section 3.4.2 to the theoretical results of Section 3.2. Aside from Proposition 3.4.3, which is in Section 5.2 of Clarkson et al. (2020), the material in Section 3.4.4 is all new. Section 3.5.1 is added to provide more details on the calculation of expected search time.

3.1 Model and Preliminaries

We formulate the two-mode search problem as a semi-Markov decision process with the following special features:

1. A single object is hidden in one of n discrete boxes labelled $1, \ldots, n$. The

object is hidden in box i with *hiding probability* $p_i > 0$ for $i = 1, \ldots, n$, with $\sum_{i=1}^{n} p_i = 1$.

2. At each decision epoch preceding the object being discovered, a single action is taken, which specifies both the box to be searched next and the search mode to be used.

3. A slow (resp. fast) search in box i takes *search time* $t_{i,s}$ (resp. $t_{i,f}$) to complete, and finds the object—if it is hidden in box i—with *detection probability* $q_{i,s}$ (resp. $q_{i,f}$). The search times satisfy $0 < t_{i,f} < t_{i,s}$ and the detection probabilities $0 < q_{i,f} < q_{i,s} < 1$.

4. Decision epochs occur at time 0, and at the completion of each unsuccessful search, until the object is found. Since the search times vary among boxes and modes, the times between decision epochs are not constant, so the process is semi-Markov.

5. Similarly to the search multi-armed bandits of Chapter 2, the state of the system may be thought of as the number of unsuccessful slow searches and the number of unsuccessful fast searches in each box to date. Given this history of searches, the current (posterior) hiding probability for each box may be calculated from its original hiding probability using Bayes' theorem (see Remark 3.1.2 for details of the calculation). Therefore, we may also consider the state of the system to be the current posterior across all boxes. Note that the former choice of state leads to a discrete state space, while the latter induces a continuous state space.

6. The goal of the analysis is to determine a policy—a rule for choosing actions—to minimise the expected time to find the object, or *expected search time.*

Standard theory indicates that there exists an optimal policy which is *stationary, nonrandomised and Markov* (Puterman, 2014). In this context, Markov means the optimal action depends only on the total number of times each action has been taken to date (sufficient to calculate the posterior hiding probabilities), and not the order in which these actions have been interlaced. Nonrandomised means it is not necessary for the searcher to randomise their action choice in any state to follow an optimal policy. Stationarity means the optimal action is the same if a state is revisited; whilst revisiting a state is not possible with the history-state formulation, it is possible with the posterior-state formulation. Combining the three terms, the conclusion (with either choice of state) is that prior to discovery the next optimal action will be a deterministic function of the current state.

Note that the objective in Puterman (2014) is to minimise the expected total cost over an infinite horizon, whilst our objective is to minimise total cost until the object is found. Yet, by creating an additional, no-cost-incurring state inhabited with probability 1 any time after the object has been discovered, our model can be seen to be equivalent to an infinite-horizon problem, so the theory of Puterman (2014) applies.

Following any optimal policy generates an optimal *search sequence* of actions to be taken prior to the object's discovery. For $i = 1, \ldots, n$, the condition $p_i > 0$ implies that any such optimal search sequence must search box i (via some search action) infinitely often. Any search sequence which does not satisfy this requirement will have a strictly positive probability of failing to find the object, and consequently an expected search time which is infinite.

The single-mode version of the above is the classic time-independent search problem discussed in Chapters 1 and 2 of this thesis. As discussed in Chapter 2, by the comment of Kelly in Gittins (1979), when there is only one search mode for each box, a search sequence that minimises the expected search time can be found by implementing a Gittins index policy. The two-mode search problem is substantially more difficult, as the searcher not only needs to decide where to search next, but also which search mode to use. To begin our analysis, consider a simpler version of the two-mode problem, where the searcher needs to decide only which location to search next because the choice of search mode is predetermined. Where $\mathbb{Z}^+ \equiv \{1, 2, \ldots\}$, assume that, for $i = 1, \ldots, n$, a *within-box subsequence* $A_i \equiv \{a_{i,k}, k \in \mathbb{Z}^+\}$ is pre-specified, where $a_{i,k}$ is the mode at which the kth search of box i is to be made. How do we then optimally interlace within-box subsequences $\mathbf{A} \equiv \{A_i, \ i = 1, \ldots, n\}$ to produce a search sequence that minimises the expected search time?

To answer this question, recall that Chapter 2 also discussed how Kelly's comment shows Gittins indices solve the single-mode problem when the detection probabilities and search times for each box depend upon its number of prior searches, a problem coined the classic time-dependent search problem in Chapters 1 and 2. As noted in Section 2.3.2, the two-mode problem with fixed within-box subsequences $\mathbf{A}$ is a classic time-dependent search problem, and hence is optimally solved using Gittins indices.

Consider a situation in which box i has been searched some $m_i \in \mathbb{N}$ times already, $i = 1, \ldots, n$, where $\mathbb{N} \equiv \{0, 1, \ldots\}$. The Gittins index for box i in the classic time-dependent search problem is derived in Section 2.2 of Chapter 2 and given in (2.2.9). Applying (2.2.9) to the two-mode problem with fixed $\mathbf{A}$, where the kth search of box i must use mode $a_{i,k}$, the Gittins index for box i, $i = 1, \ldots, n$, is given by

$$G_i(m_i, A_i) \equiv p_i \prod_{k=1}^{m_i} \left(1 - q_{i,a_{i,k}}\right) \left[\sup_{b \in \mathbb{Z}^+} \frac{\sum_{u=m_i+1}^{m_i+b} q_{i,a_{i,u}} \left\{ \prod_{v=m_i+1}^{u-1} (1 - q_{i,a_{i,v}}) \right\}}{\sum_{u=m_i+1}^{m_i+b} t_{i,a_{i,u}}} \right]. \quad (3.1.1)$$

The following theorem summarises.

Theorem 3.1.1 Any search sequence which minimises the expected search time in the two-mode search problem with fixed set of within-box subsequences $\mathbf{A}$ is characterised as follows. At any point at which box i has been searched $m_i \in \mathbb{N}$ times, $i = 1, \ldots, n$, the next search will be of any box j satisfying $j = \arg\max_{i=1,\ldots,n} G_i(m_i, A_i)$, and will use search mode a_{j,m_j+1}.

Remark 3.1.2 An equivalent set of indices (in the sense of determining the same optimal search sequences) can be obtained by dividing all Gittins indices $G_i(m_i, A_i)$,

$i = 1, \ldots, n$, in (3.1.1) by the quantity

$$\sum_{j=1}^{n} p_j \left\{ \prod_{k=1}^{m_j} (1 - q_{j,a_{j,k}}) \right\}$$

to obtain new indices which take the form

$$G'_i(m_i, A_i) \equiv p'_i \left[\sup_{b \in \mathbb{Z}^+} \frac{\sum_{u=m_i+1}^{m_i+b} q_{i,a_{i,u}} \left\{ \prod_{v=m_i+1}^{u-1} (1 - q_{i,a_{i,v}}) \right\}}{\sum_{u=m_i+1}^{m_i+b} t_{i,a_{i,u}}} \right],$$

where p'_i is the object's current (posterior) hiding probability for box i. The indices $G'_i(m_i, A_i)$, $i = 1, \ldots, n$, are not Gittins indices in the classical sense, not least since they all change as each (unsuccessful) search is completed and not only the index of the box just searched.

To summarise, any set of within-box subsequences $\mathbf{A}$ is optimally interlaced using a known Gittins index policy. Therefore, the two-mode search problem is solved by finding an optimal set of within-box subsequences (optimal under the assumption that the searcher will use a Gittins index policy).

Suppose $A_2, \ldots, A_n$ — within box subsequences for boxes $2, \ldots, n$ — are fixed. In general, a best within-box subsequence for box 1 to interlace with $A_2, \ldots, A_n$ depends on the parameters of boxes $2, \ldots, n$ and the sequences $A_2, \ldots, A_n$. In the next subsection, we identify sufficient conditions for a box (say box 1) such that its best within-box subsequence is the same (and consists of only one search mode) no matter the parameters of the other boxes and their chosen within-box subsequences. In other words, no matter the other boxes available and the modes we choose to search them, as long as a Gittins index policy is used to choose between boxes, it is always best to search box 1 using the same search mode.

3.2 Structural Properties of an Optimal Policy

We begin this section with a numerical example. Consider a pair of two-box problems. Box 1 has two search modes which are the same in both problems. Box 2 has just one search mode which differs between the two problems. For each problem, Figure 3.2.1 shows an optimal search mode for each value of $p \in (0,1)$, where p is the hiding probability for box 1 (Section 3.2.3 will give details of the solution method.) Note that, in both problems, when it is optimal to search box 1, the optimal choice of mode depends upon the value of p. Further, a comparison of the two problems shows that the optimal choice of mode for box 1 is also dependent on the available search mode of box 2.

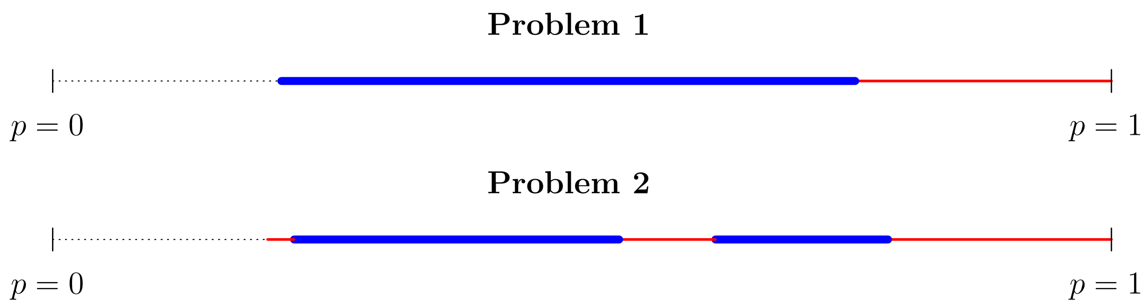

Figure 3.2.1: Optimal actions for $p \in (0,1)$ in a pair of two-box problems. Box 1 is the same in both problems with parameters $q_{1,f} = 0.4$, $q_{1,s} = 0.8$, $t_{1,f} = 1$, $t_{1,s} = 2.2$. Box 2 has a single search mode with parameters q_2 and t_2. In problem 1, $q_2 = 0.2$ and $t_2 = 2$, and in problem 2, $q_2 = 0.9$ and $t_2 = 9$. A thick blue/thin red line indicates a slow/fast search in box 1; a black dotted line indicates a search in box 2.

Figure 3.2.1 shows that, in some search problems, the optimal choice of search mode for a box can be very complicated, depending on the object's current (posterior) hiding probabilities and the search modes of the other boxes. It would be useful, however, to identify boxes where one search mode is so much better than the other that the latter need never be used in an optimal policy, regardless of the search modes of the other boxes. Sections 3.2.1 and 3.2.2 present sufficient conditions for such dominance to occur. Based on these findings, Section 3.2.3 introduces a Monte Carlo method to estimate the optimal expected search time, and Section 3.2.4 presents a lower bound on it.

3.2.1 A Sufficient Condition for the Slow Mode to Dominate

We consider the two-mode search problem described in Section 3.1. For any search mode with detection probability q and search time t, we define the *detection rate* of the search mode to be the ratio q/t. Our first result states that, if the fast mode and the slow mode for some box have the same detection rate, then an optimal policy uses the fast mode for that box at most once and, in fact, never needs to use it at all.

Theorem 3.2.1 In the two-mode search problem, if any box j satisfies

$$\frac{q_{j,s}}{t_{j,s}} = \frac{q_{j,f}}{t_{j,f}},$$

1. There exists an optimal search sequence in which box j is always searched slowly.

2. Any optimal search sequence searches box j fast at most once.

Without loss of generality, we prove Theorem 3.2.1 for $j = 1$, so assume that

$$\frac{q_{1,s}}{t_{1,s}} = \frac{q_{1,f}}{t_{1,f}}.$$

The proof requires the introduction of a variant of the two-mode search problem and two lemmas. To begin, suppose that we fix within-box subsequences $A_2, A_3, \ldots, A_n$—which determine the modes of successive visits to boxes $2, 3, \ldots, n$—and consider competing choices for the within-box subsequence for box 1. Since our focus will be primarily on box 1, we shall, for the remainder of the Theorem 3.2.1 proof, omit the identifying subscript 1 from the notations q_1 and t_1, but it will assist clarity to retain it for p_1. For box 1, we write A for some arbitrary within-box subsequence and S for the specific within-box subsequence consisting entirely of the slow mode. In addition, write T_A for the optimal expected search time under within-box subsequences $A, A_2, A_3, \ldots, A_n$, and T_S for the optimal expected search time under within-box subsequences $S, A_2, A_3, \ldots, A_n$. To prove Theorem 3.2.1, we will show that $T_S \leq T_A$, and when A contains two or more fast modes, $T_S < T_A$.

In order to proceed, we introduce a variant of the two-mode search problem, which will facilitate a comparison between T_S and T_A. In this variant, when searching in box 1, instead of making fast and slow searches in the usual manner, the searcher *sweeps box* 1 *continuously* as described below.

Imagine box 1 is represented by a line segment $[0, t_s]$. If the object is hidden in box 1, then its position is distributed uniformly over $[0, t_s]$. By *sweeping box* 1 *continuously,* the searcher moves on this line segment, starting from 0 toward t_s at constant speed 1, and finds the object with probability q_s when she meets it, independent of everything else. Upon reaching the end point t_s, the searcher jumps immediately back to 0 and moves toward t_s again. In addition, at any point $x \in [0, t_s)$, the searcher may stop searching box 1 in order to search another box. When she returns to box 1, her search is resumed from position x, the place where she abandoned her last search of box 1. See Figure 3.2.2 for a diagram representing this continuous-sweeping search. We write T_W for the optimal expected search time in this continuous-sweeping variant of the two-mode search problem where the searcher uses the within-box subsequences A_i for boxes $i = 2, \ldots, n$ and sweeps continuously for box 1.

In the continuous-sweeping variant, suppose that the searcher makes a sweep of box 1 for time t_f, but, instead of starting the sweep at the place she abandoned box 1 last, selects a random starting point on $[0, t_s)$. If it is hidden in box 1, the searcher passes the object on this sweep with probability t_f/t_s, and hence finds it with probability

$$\frac{t_f}{t_s} \cdot q_s = q_f.$$

So, one way to interpret the standard two-mode search problem is that each time the searcher visits box 1 to conduct a fast search, she sweeps a random subset of $[0, t_s]$ of length t_f, independent of the subsets she has searched before, while to conduct a slow search she does one complete sweep of the interval. In the continuous-sweeping variant of the problem, the searcher has an advantage, since each time she visits box 1 she begins by sweeping the subset which has been searched least hitherto. Therefore,

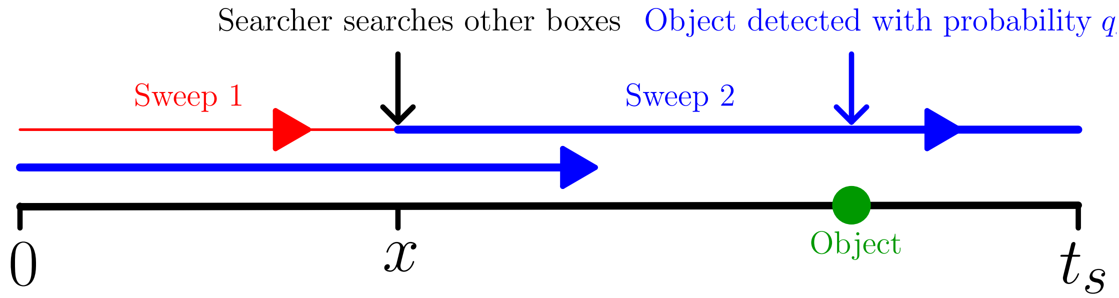

Figure 3.2.2: A diagram showing the continuous-sweeping search. The object, which is represented by a green circle, is hidden uniformly on $[0, t_s]$. The first sweep of box 1 is from 0 to x and marked by a thin red line. Upon reaching x, the searcher leaves box 1 to search other boxes. The second sweep of box 2 starts at x and is marked by a thick blue line. The searcher meets the object on the second sweep, detecting it with probability q_s. During the second sweep, the searcher reaches the endpoint of the interval t_s, at which she jumps back to 0 and moves toward t_s again.

$T_W \leq T_A$; the advantage of the sweeping searcher is quantified in the next lemma.

Lemma 3.2.2 If A contains two or more fast modes, then $T_A - T_W > p_1 t_s/2$. Otherwise, $T_A - T_W \geq p_1 t_s/2$.

Proof. Consider two searchers. Searcher 1 uses within-box subsequence A for box 1, while searcher 2 uses continuous sweeping. Both searchers use within-box subsequence A_i for box i, for $i = 2, \ldots, n$. Searchers 1 and 2 have optimal expected search times equal to T_A and T_W, respectively.

An optimal policy for searcher 1 chooses between boxes using Gittins indices, as detailed in Theorem 3.1.1. Write π_A for an arbitrary optimal policy for searcher 1. Below we describe a feasible policy for searcher 2 which *mimics* π_A. Whenever searcher 1 searches box $i \neq 1$ under π_A, let searcher 2 search the same box using the same mode. When searcher 1 searches box 1 using the slow (resp. fast) mode under π_A, let searcher 2 also search box 1, starting at the place she abandoned last time, moving toward t_s at constant speed 1 for t_s (resp. t_f) time units unless she either finds the object or reaches the endpoint t_s before the allotted time expires. In the former case, the search is over, whilst in the latter case, she jumps back to 0 and moves toward t_s again until the allotted time is exhausted or the object is found. Let the expected search time for searcher 2 under this mimicking policy be $T_W(\pi_A) \geq T_W$.

Suppose searcher 1 follows π_A and searcher 2 mimics π_A. If the object is not hidden in box 1, the conditional expected search time is clearly identical for both searchers. Now suppose the object is hidden in box 1. First, we examine the expected time spent in box 1 for each searcher.

To begin, consider searcher 1, who uses the arbitrary within-box subsequence A to search box 1. By (2.3.2), if searcher 1 were free to choose any within-box subsequence for box 1, the expected time in box 1 given the object is in box 1 is minimised to $\mu^* = t_f/q_f = t_s/q_s$. By an argument similar to the derivation of μ^* in (2.3.2), the same expected time is maximised to $\bar{\mu}$ satisfying

$$\bar{\mu} = \max[t_f + \bar{\mu}(1 - q_f), t_s + \bar{\mu}(1 - q_s)],$$

from which it follows that

$$\bar{\mu} = \max\left[\frac{t_f}{q_f}, \frac{t_s}{q_s}\right].$$

Therefore, $\mu^* = \bar{\mu}$, and hence, given the object is there, the expected time in box 1 for searcher 1 is $t_f/q_f = t_s/q_s$ regardless of the choice of A.

For searcher 2, let Y denote the number of times the searcher needs to meet the object to find it. In other words, searcher 2 sweeps the whole of $[0, t_s]$ a total of $Y - 1$ times in vain, and finds the object on the Y^{th} sweep. Further, it is plain that Y follows a geometric distribution with success probability q_s. Each of the first $Y - 1$ failed complete sweeps takes time t_s while the last successful pass takes an expected time of $t_s/2$, since the object's position is uniformly distributed over $[0, t_s]$. Hence, the expected time spent in box 1 by searcher 2 is

$$t_s\,E[Y - 1] + \frac{t_s}{2} = \frac{t_s}{q_s} - \frac{t_s}{2}.$$

Second, we examine the expected time spent in boxes $i = 2, \ldots, n$ by each searcher if the object is hidden in box 1. By comparing the detection probabilities of each searcher on their k^{th} visit to box 1, we show that this quantity for searcher 2 is no greater than for searcher 1.

Suppose searcher 1 uses the fast mode on her k^{th} visit to box 1, so her relevant detection probability is q_f. Correspondingly, searcher 2 will sweep box 1 for t_f time units on her k^{th} visit, but her detection probability will depend on the point $x \in [0, t_s)$ at which her $(k-1)^{\text{st}}$ unsuccessful visit to box 1 ended. Consider two cases:

1. $x \in [0, t_s - t_f]$. In this case the probability required is given by

$$\begin{aligned} &P(\text{object found in } (x, x+t_f] \mid \text{object was not found in } [0,x]) \\ &= \frac{\frac{t_f}{t_s} \cdot q_s}{1 - \frac{x}{t_s} \cdot q_s} \geq q_f. \end{aligned} \tag{3.2.1}$$

2. $x \in (t_s - t_f, t_s)$. In this case the probability required is given by

$$\begin{aligned} &P(\text{object found in } (x, t_s] \text{ or in } [0, x+t_f-t_s] \mid \text{object was not found in } [0,x]) \\ &= \frac{\left(\frac{t_s - x}{t_s}\right) \cdot q_s + \left(\frac{x+t_f-t_s}{t_s}\right) \cdot (1-q_s) \cdot q_s}{1 - \frac{x}{t_s} \cdot q_s} \\ &= \frac{\left(\frac{t_f}{t_s}\right) \cdot q_s - \left(\frac{x+t_f-t_s}{t_s}\right) \cdot q_s^2}{1 - \frac{x}{t_s} \cdot q_s} \\ &> \frac{\frac{t_f}{t_s} \cdot q_s - \frac{x t_f}{t_s^2} \cdot q_s^2}{1 - \frac{x}{t_s} \cdot q_s} = q_f, \end{aligned} \tag{3.2.2}$$

where the inequality in (3.2.2) follows since

$$x(t_s - t_f) < t_s(t_s - t_f), \quad \text{which implies} \quad t_s(x + t_f - t_s) < x t_f.$$

Note that, whilst (3.2.2) is a strict inequality, (3.2.1) satisfies equality if and only if $x = 0$. Therefore, if searcher 1 uses the fast mode of box 1 for a first time, since searcher 2 will begin her corresponding sweep at $x = 0$, the detection probabilities of both searchers will equal q_f. However, if searcher 1 uses the fast mode for box 1 for a second time, searcher 2 will begin her corresponding sweep at $x = t_f > 0$ with a detection probability strictly greater than q_f.

Now suppose that searcher 1 uses the slow mode on her k^{th} visit to box 1, so her relevant detection probability is q_s. Correspondingly, searcher 2's k^{th} visit to box 1 takes t_s time units, and will discover the object with probability q_s, regardless of where in $[0, t_s)$ this visit begins.

From these calculations, we conclude that the detection probability for searcher 2 on her k^{th} visit to box 1 is no smaller than the corresponding quantity for searcher 1. Consequently, if the object is in box 1, the number of searches of box 1 required to find the object for searcher 2 is stochastically no larger than that for searcher 1. Therefore, if the object is hidden in box 1, the expected time spent in boxes $i = 2, \ldots, n$ is no larger for searcher 2 than for searcher 1. Further, we have equality in this expected search time between the two searchers if and only if searcher 1 makes no more than one fast search of box 1.

We conclude from the above calculations that if A contains two or more fast modes, then

$$T_A - T_W(\pi_A) > p_1 \left(\frac{t_s}{q_s} - \left(\frac{t_s}{q_s} - \frac{t_s}{2} \right) \right) = \frac{p_1 t_s}{2}.$$

Otherwise, $T_A - T_W(\pi_A) = p_1 t_s / 2$. Since $T_W(\pi_A) \geq T_W$, the proof is complete. ■

The next lemma shows that the inequalities in Lemma 3.2.2 become equalities when box 1 is always searched using the slow mode.

Lemma 3.2.3 $T_S - T_W = p_1 t_s / 2$.

Proof. We again consider two searchers. Searcher 2 uses continuous sweeping for box 1 (as in the proof of Lemma 3.2.2), while searcher 3 always searches box 1 slowly, namely using the within-box subsequence S. Both searchers use within-box subsequences A_i for boxes $i = 2, \ldots, n$. Searchers 2 and 3 have optimal expected search times equal to T_W and T_S, respectively.

An optimal policy for searcher 3 chooses which box to search next according to a suitable collection of Gittins indices, as detailed in Theorem 3.1.1. To study an optimal policy for searcher 2, divide the interval $[0, t_s]$ into m equal-length subintervals $1_r \equiv [(r-1)t_s/m, rt_s/m)$, $r = 1, \ldots, m-1$, and $1_m \equiv [(m-1)t_s/m, t_s]$. Think of these subintervals as m small boxes, and enforce the rule for searcher 2 that she must search each small box in its entirety without interruption. Denote the optimal expected search time for searcher 2 under this constraint by T_W^m, noting that $T_W^m \downarrow T_W$ as $m \to \infty$. Note also that, if searcher 2's most recent search among the small boxes was of 1_r (i.e., of the box corresponding to that subinterval of $[0, t_s]$), then her next search of a small box must be of 1_{r+1} if $r = 1, \ldots m-1$, or 1_1 if $r = m$. For each small box, the search time is t_s/m and the detection probability is q_s.

This means that searcher 2 has regular boxes $2, 3, \ldots, n$ alongside m identical small boxes $1_1, 1_2, \ldots, 1_m$, while searcher 3 has regular boxes $1, 2, \ldots, n$. For both searchers, p_i is the object's hiding probability for box i, $i = 2, \ldots, n$. For searcher 2, p_1/m is the object's hiding probability for each of the small boxes $1_r, r = 1, \ldots, m$, while for searcher 3, p_1 is the object's hiding probability for box 1. For searcher 3, a suitable Gittins index policy determines an optimal search sequence. In fact, this is also the case for searcher 2 notwithstanding the ordering constraints among the small boxes, as there exists a Gittins index policy for searcher 2 which guarantees that those constraints are satisfied. To see this, consider a situation in which the object has not been discovered and all of the m small boxes have been visited k times, having corresponding Gittins indices denoted by $G_{1_r}(k)$, $r = 1, \ldots, m$, which are plainly equal. Assume also that these m indices are maximal among those for the $n-1+m$ boxes available to searcher 2. A Gittins index policy is free to break ties in any manner, so we suppose that box 1_1 is searched next by searcher 2. Following this search, assumed unsuccessful, the small boxes now have indices $G_{1_1}(k+1) < G_{1_r}(k)$, $r = 2, \ldots, m$, and so the small boxes 1_r, $r = 2, \ldots, m$, continue to have the maximal index. We suppose that searcher 2's Gittins index policy next chooses box 1_2 for searching and so on. Continuing in this fashion, we see that there is a Gittins index policy for searcher 2 with the property that, in the absence of any discovery of the object, once small box 1_1 is searched, all the remaining small boxes are then searched in the correct order.

Now we stochastically couple the location of the object between the two searchers, such that if the object is in box $i \neq 1$ for searcher 3 then it is in the same box for searcher 2, and if the object is in box 1 for searcher 3, then it is equally likely to be in any of the m small boxes 1_r, $r = 1, \ldots, m$, for searcher 2. In addition, we stochastically couple the search outcomes for the two searchers in boxes $i = 2, \ldots, n$.

At the beginning of the search, it is easy to show that searcher 3's Gittins index for box 1 is $p_1 q_s / t_s$ which is equal to $G_{1_r}(0)$, $r = 1, \ldots, m$, namely searcher 2's Gittins indices for her m small boxes 1_r, $r = 1, \ldots, m$. Hence, the two searchers may follow the same optimal search sequence until one of two things happen:

1. The object is found before searcher 3 searches box 1. Because we stochastically couple the object's location and the search outcomes in boxes $i \neq 1$, searcher 2 will find the object at the exact same time.

2. Searcher 3 searches box 1 before the object is found. When searcher 3 searches box 1, the current Gittins index for box 1 must be maximal among boxes $i = 1, \ldots, n$. Since searcher 2 follows the exact same search sequence, it will follow that the m small boxes 1_r, $r = 1, \ldots, m$, will all be of maximal index for searcher 2 at this point and by the above discussion will now all be searched in order before searcher 2 moves on.

When the object is in box 1, we stochastically couple the search outcomes in box 1 for the two searchers, such that searcher 3 finds the object in box 1 if and only if searcher 2 finds the object in a single sweep through the m small boxes 1_r, $r = 1, \ldots, m$. With probability q_s both searchers find the object on this visit of box 1. In this case, searcher 3's search ends in a further t_s time units, whilst the expected future search time for searcher 2 is

$$\frac{(\sum_{r=1}^{m} r) \cdot t_s}{m^2} = \left(\frac{m+1}{2}\right) \cdot \frac{t_s}{m},$$

since searcher 2 does not need to search small boxes $1_{r+1}, 1_{r+2}, \ldots, 1_m$, should the object be found in 1_r.

With probability $1 - q_s$, neither searcher finds the object on this visit of box 1, and the search continues. At this moment, the current index for searcher 3's box 1 and those for searcher 2's m small boxes are identical. Therefore, some optimal policy for each searcher will henceforth instruct them to follow the same search sequence, until either finding the object in some box i, $i = 2, \ldots, n$, or it again becomes optimal for both searcher 3 to return to box 1 and searcher 2 to return to the boxes 1_r, $r = 1, \ldots, m$. The same argument then repeats.

Consequently, the time spent in boxes $i = 2, \ldots, n$ is identical for the two searchers, and the time spent in box 1 (or boxes 1_r, $r = 1, \ldots, m$, for searcher 2) is identical for the two searchers if the object is not hidden there. The only difference between T_S and T_W^m arises in the time spent in box 1 when the object is hidden in box 1. From the above we conclude that

$$T_S - T_W^m = p_1 \left(t_s - \left(\frac{m+1}{2}\right) \cdot \frac{t_s}{m}\right).$$

Taking $m \to \infty$ in the above yields $T_S - T_W = p_1 t_s / 2$, which completes the proof. ■

From Lemmas 3.2.2 and 3.2.3, we can conclude that $T_S \leq T_A$, with the inequality becoming strict when A contains two or more fast modes. This completes the proof of Theorem 3.2.1.

It is easy to extend the result in Theorem 3.2.1 to boxes where the detection rate of slow is strictly greater than the detection rate of fast; for such boxes it is never optimal to use the fast mode.

Theorem 3.2.4 In the two-mode search problem, if any box j satisfies

$$\frac{q_{j,s}}{t_{j,s}} > \frac{q_{j,f}}{t_{j,f}},$$

then in any optimal search sequence box j is always searched slowly.

Proof. Without loss of generality set $j = 1$. Let

$$\widehat{t}_{1,f} \equiv \frac{q_{1,f} \cdot t_{1,s}}{q_{1,s}} < t_{1,f}, \quad \text{so} \quad \frac{q_{1,s}}{t_{1,s}} = \frac{q_{1,f}}{\widehat{t}_{1,f}}. \tag{3.2.3}$$

Suppose now we fix the within-box subsequences to be A, which contains at least one fast mode, for box 1 and A_i for boxes $i = 2, 3, \ldots, n$, and write T_A for the corresponding optimal expected search time. Fix the same within-box subsequences in a new two-mode search problem in which the fast search time of box 1 is reduced from $t_{1,f}$ to $\widehat{t}_{1,f}$, with all other parameters being unchanged. We write $\widehat{T}_A$ for the corresponding optimal expected search time. Since $\widehat{t}_{1,f} < t_{1,f}$ and A contains at least one fast mode, it is clear that $\widehat{T}_A < T_A$. In addition, by (3.2.3), it follows from Theorem 3.2.1 that $\widehat{T}_S \leq \widehat{T}_A$, where $\widehat{T}_S$ is the optimal expected search time using within-box subsequences $S, A_2, A_3, \ldots, A_n$ for the problem with the new search time $\widehat{t}_{1,f}$. However, under within-box subsequence S, box 1 is never searched fast, so the reduction of $t_{1,f}$ to $\widehat{t}_{1,f}$ is immaterial to the computation of $\widehat{T}_S$. It follows that $T_S = \widehat{T}_S \leq \widehat{T}_A < T_A$, completing the proof. ■

We may combine information from Theorems 3.2.1 and 3.2.4 into one result to identify boxes for which it is optimal to always use the slow mode.

Theorem 3.2.5 In the two-mode search problem, if any box j satisfies

$$\frac{q_{j,s}}{t_{j,s}} \geq \frac{q_{j,f}}{t_{j,f}}, \tag{3.2.4}$$

then there exists an optimal search sequence in which box j is always searched slowly.

We conclude this subsection by discussing the uniqueness of the optimality of the slow mode of a box under (3.2.4). If the slow detection rate is strictly greater than the fast detection rate for some box (without a loss of generality, box 1), then Theorem 3.2.4 tells us that S, which contains only slow modes of box 1, is the *unique* optimal within-box subsequence for box 1. Now suppose the slow and fast detection rates are equal for box 1; then Theorem 3.2.1 tells us that S is an optimal within-box subsequence for box 1 but *not* that S is uniquely optimal. Whilst any within-box subsequence for box 1 containing two or more fast modes is suboptimal, the following example presents a search problem where it is optimal to make just one fast search of such a box 1. Example 3.2.6 is constructed using observations of the proofs of Lemmas 3.2.2 and 3.2.3; the construction of Example 3.2.6 can be found in Appendix A.

Example 3.2.6 Consider a search problem with $n = 2$ boxes. Box 1 contains the object with probability p, and has two search modes with respective search times t_f and t_s and detection probabilities q_f and q_s which satisfy

$$\frac{q_s}{t_s} = \frac{q_f}{t_f}. \tag{3.2.5}$$

Box 2 has just one search mode with detection probability q_2 and search time t_2. Suppose that

$$(1 - q_2) = (1 - q_s)^m \quad \text{for some} \quad m \in \mathbb{Z}^+, \tag{3.2.6}$$

$$\text{and} \quad \frac{pq_s(1 - q_s)^k}{t_s} = \frac{(1 - p)q_2}{t_2} \quad \text{for some } k \in \mathbb{Z}^+, \tag{3.2.7}$$

so we have

$$p = \frac{q_2 t_s}{q_2 t_s + q_s t_2 (1 - q_s)^k}. \tag{3.2.8}$$

For a numerical example, let $q_f = 0.2, q_s = 0.4, t_f = 1, t_s = 2, q_2 = 0.64, t_2 = 2$. Then (3.2.5) clearly holds, and (3.2.6) holds with $m = 2$. Further (3.2.7) holds with $k = 1$ and $p = 8/11$.

Write f (resp. s) for a fast (resp. slow) mode of box 1, and consider the following two within-box subsequences for box 1:

$$A_f \equiv (f, s, s, \ldots) \quad \text{and} \quad S \equiv (s, s, s, \ldots).$$

Write T_{A_f} (resp. T_S) for the optimal expected search time under within-box subsequence A_f (resp. S); we show that $T_{A_f} = T_S$.

First, we derive the unique optimal search sequence under A_f. By (3.2.7), we have

$$\frac{pq_s(1-q_f)(1-q_s)^{k-1}}{t_s} > \frac{(1-p)q_2}{t_2} > \frac{pq_s(1-q_f)(1-q_s)^k}{t_s},$$

so, under within-box subsequence A_f, it is uniquely optimal for the searcher to make one fast then k slow searches of box 1 before searching box 2. By (3.2.6), it is next uniquely optimal for the searcher to search box 1 slowly m times before searching box 2 again and then repeat this cycle indefinitely. To summarise, the unique optimal search sequence π_{A_f} under A_f satisfies

$$\pi_{A_f} = (f, \underbrace{s, \ldots, s}_{k}, 2, \underbrace{s, \ldots, s}_{m}, 2, \underbrace{s, \ldots, s}_{m}, 2, \ldots),$$

where 2 represents a search of box 2.

Now we derive an optimal search sequence π_S under S. By (3.2.7), under within-box subsequence S, it is uniquely optimal for the searcher to begin with k slow searches of box 1, after which the indices of boxes 1 and 2 are tied. Under any optimal policy, the next two searches must be of box 1 and box 2 but with the order unconstrained. Next, by (3.2.6), under S it is uniquely optimal for the searcher to make $m-1$ consecutive slow searches of box 1 before the indices are again tied. The same cycle of $m+1$ searches then repeats indefinitely. Let π_S be the optimal search sequence that always searches box 1 first when there is a tie, so we have

$$\pi_S = (s, \underbrace{s, \ldots, s}_{k}, 2, \underbrace{s, \ldots, s}_{m}, 2, \underbrace{s, \ldots, s}_{m}, 2, \ldots).$$

Note that π_{A_f} and π_S differ only in the mode of their first search. We now use stochastic coupling to show that $T_{A_f} = T_S$.

First, suppose that the object is hidden in box 2, which occurs with probability $1-p$. Under both π_{A_f} and π_S, assuming no prior discovery, the object is found on any search of box 2 with probability q_2. Therefore, if we stochastically couple the outcomes of these searches of box 2 under both π_{A_f} and π_S, the object will be found time $t_s - t_f$ later under π_S than under π_{A_f}.

Now suppose that the object is in box 1, which occurs with probability p. Under π_{A_f}, the object is found with probability q_f on the first search of box 1 and, assuming no prior discovery, with probability q_s on any subsequent search of box 1. Under π_S, assuming no prior discovery, the object is found with probability q_s on any search of box 1. This allows us to stochastically couple the outcome of the first search in box 1 under both π_{A_f} and π_S in the following way:

1. With probability q_f, under both π_{A_f} and π_S, the object is found on the first search of box 1. In this circumstance, the object will be found time $t_s - t_f$ later under π_S than under π_{A_f}.

2. With probability $1-q_s$, under neither π_{A_f} nor π_S is the object found on the first search of box 1. Suppose we further stochastically couple the outcomes of all subsequent searches of box 1 under π_{A_f} and π_S; then the object will be found time $t_s - t_f$ later under π_S than under π_{A_f}.

3. With probability $q_s - q_f$, on the first search of box 1, the object is found under π_S but overlooked under π_{A_f}. Then the object is found in time t_s under π_S, and, using (3.2.6), in expected time $t_f + t_s/q_s + t_2(1-q_s)^k/q_2$ under π_{A_f}.

From the above calculations, $T_S - T_{A_f}$ is given by

$$\begin{aligned} T_S - T_{A_f} &= (1-p)(t_s - t_f) \\ &\quad + p\left[(q_f + (1-q_s))(t_s - t_f) + (q_s - q_f)\left(t_s - t_f - \frac{t_s}{q_s} - \frac{t_2(1-q_s)^k}{q_2}\right)\right] \\ &= t_s - t_f + p(q_f - q_s)\frac{q_2 t_s + q_s t_2 (1-q_s)^k}{q_s q_2}. \end{aligned}$$

Substituting in p from (3.2.8), it follows from (3.2.5) that $T_S - T_{A_f} = 0$.

Note that, as an alternative to the stochastic coupling argument above, it is not difficult to calculate T_S and T_{A_f} directly by conditioning on the location of the object.

3.2.2 A Sufficient Condition for the Fast Mode to Dominate

This subsection gives a sufficient condition for a box such that any optimal policy never uses the slow mode for that box. We first need a lemma.

Lemma 3.2.7 In the two-mode search problem, if any box j satisfies

$$\frac{q_{j,f}}{t_{j,f}} > \frac{q_{j,s}}{t_{j,s}},$$

then a slow search of box j followed immediately by a fast search of the same box j is suboptimal.

The proof of Lemma 3.2.7 relies on a simple argument featuring a pairwise interchange of consecutive fast and slow searches of box j, and is therefore omitted.

Theorem 3.2.8 In the two-mode search problem, if any box j satisfies

$$\frac{q_{j,f}(1-q_{j,s})}{t_{j,f}} \geq \frac{q_{j,s}}{t_{j,s}}, \tag{3.2.9}$$

then in any optimal search sequence box j is always searched fast.

Proof. Without a loss of generality, set $j = 1$. Fix the within-box subsequence for box i to take an optimal value A_i^*, for $i = 2, \ldots, n$. We first suppose that the within-box subsequence for box 1, namely $A_1 = \{a_{1,k}, k \in \mathbb{Z}^+\}$, contains some finite, strictly positive number of slow modes. Thus, for some $\nu \in \mathbb{Z}^+$ we have $A_1 \in \Sigma(\nu)$, the set of within-box subsequences for box 1 with precisely ν slow modes. Write r for the position of the last occurrence of the slow mode within A_1. In the absence of discovery of the object, consider the point in the application of some optimal search sequence at which box 1 is to be searched for the r^{th} time. At this point box 1 has Gittins index $G_1(r-1, A_1)$, which is maximal among all boxes. Since the last slow mode within A_1 occurs at position r, it follows from (3.1.1) that $G_1(r-1, A_1)$ is given by

$$G_1(r-1, A_1) = p_1 \left\{ \prod_{m=1}^{r-1} (1 - q_{1,a_{1,m}}) \right\} \left[\max \left(\frac{q_{1,s}}{t_{1,s}}, \frac{q_{1,s} + q_{1,f}(1-q_{1,s})}{t_{1,s} + t_{1,f}}, G \right) \right],$$

where

$$G = \sup_{l \geq 1} \frac{q_{1,s} + q_{1,f}(1-q_{1,s}) + q_{1,f}(1-q_{1,s}) \cdot \sum_{u=1}^{l}(1-q_{1,f})^u}{t_{1,s} + (l+1)t_{1,f}}.$$

Note that we clearly have

$$\frac{q_{1,f}(1-q_{1,s})}{t_{1,f}} > \frac{q_{1,f}(1-q_{1,s})(1-q_{1,f})^u}{t_{1,f}}$$

for any $u \in \mathbb{Z}^+$. Combining the preceding with (3.2.9), it follows that

$$p_1 \left\{ \prod_{m=1}^{r-1} (1 - q_{1,a_{1,m}}) \right\} \left(\frac{q_{1,f}(1-q_{1,s})}{t_{1,f}} \right) \geq G_1(r-1, A_1).$$

In addition, note that we have

$$\begin{aligned} G_1(r, A_1) &= p_1 \left\{ \prod_{m=1}^{r-1} (1 - q_{1,a_{1,m}}) \right\} (1-q_{1,s}) \sup_{l \geq 0} \frac{\sum_{u=0}^{l} q_{1,f}(1-q_{1,f})^u}{(l+1)t_{1,f}} \\ &= p_1 \left\{ \prod_{m=1}^{r-1} (1 - q_{1,a_{1,m}}) \right\} \left(\frac{q_{1,f}(1-q_{1,s})}{t_{1,f}} \right), \end{aligned}$$

from which it follows that $G_1(r, A_1) \geq G_1(r-1, A_1)$.

Consequently, there exists a search sequence $G\{A_1; A_i^*, i \neq 1\}$, optimal for the fixed within-box subsequences, at which the r^{th} search of box 1 (which is slow) is followed immediately by the $(r+1)^{\text{st}}$ search of box 1 (which is fast). According to Lemma 3.2.7, however, $G\{A_1; A_i^*, i \neq 1\}$ would be strictly improved by reversing the order of these two searches. Denote this new search sequence by $G^{1(r \leftrightarrow r+1)}\{A_1; A_i^*, i \neq 1\}$. Next write $A_1^{(r \leftrightarrow r+1)}$ for the within-box subsequence for box 1 obtained by interchanging the r^{th} and $(r+1)^{\text{st}}$ modes within A_1. According to Theorem 3.1.1, the search sequence $G^{1(r \leftrightarrow r+1)}\{A_1; A_i^*, i \neq 1\}$ is no better than the search sequence $G\{A_1^{(r \leftrightarrow r+1)}; A_i^*, i \neq 1\}$, where the within-box subsequence $A_1^{(r \leftrightarrow r+1)}$ is a member of $\Sigma(\nu)$ in which the last slow mode occurs at position $r+1$.

The foregoing argument that the search sequence $G\{A_1; A_i^*, i \neq 1\}$ is strictly worse than $G\{A_1^{(r \leftrightarrow r+1)}; A_i^*, i \neq 1\}$ can be repeated to show that the latter is strictly worse than $G\{A_1^{(r \leftrightarrow r+2)}; A_i^*, i \neq 1\}$, where by $A_1^{(r \leftrightarrow r+2)}$ we mean the within-box subsequence for box 1 obtained by interchanging the r^{th} mode (slow) and $(r+2)^{nd}$ mode (fast) within A_1. This argument repeats to show that $G\{A_1^{(r\leftrightarrow r+k)}; A_i^*, i \neq 1\}$ is strictly worse than $G\{A_1^{(r\leftrightarrow r+k+1)}; A_i^*, i \neq 1\}$, $k \in \mathbb{N}$. Now write $A_1^{(r:s\to f)}$ for the within-box subsequence for box 1 obtained from A_1 by replacing the slow mode at the r^{th} position with a fast mode. If we write $E[\tau(\pi)]$ for the expected search time under search sequence π, we then have

$$\lim_{k\to\infty} E[\tau(G\{A_1^{(r\leftrightarrow r+k)}; A_i^*, i \neq 1\})] = E[\tau(G\{A_1^{(r:s\to f)}; A_i^*, i \neq 1\})],$$

from which we can further deduce by the foregoing argument that

$$E[\tau(G\{A_1^{(r:s\to f)}; A_i^*, i \neq 1\})] < E[\tau(G\{A_1; A_i^*, i \neq 1\})].$$

We conclude that the within-box subsequence $A_1 \in \Sigma(\nu)$ is dominated by $A_1^{(r:s\to f)} \in \Sigma(\nu-1)$ in the strong sense above. We can repeat this argument a further $\nu - 1$ times to infer that $A_1 \in \Sigma(\nu)$ is dominated in the same strong sense by $F \in \Sigma(0)$, which consists entirely of the fast mode of box 1.

It is clear that any within-box subsequence—including those having infinitely many slow modes—can be arbitrarily well approximated by a within-box subsequence in $\Sigma(\nu)$ for some $\nu \in \mathbb{Z}^+$. Hence, for any $\epsilon > 0$, there exists some $\nu \in \mathbb{Z}^+$ and $A_1 \in \Sigma(\nu)$ such that

$$E[\tau(G\{A_1; A_i^*, i \neq 1\})] - \inf_A E[\tau(G\{A; A_i^*, i \neq 1\})] < \epsilon,$$

where the infimum is over all within-box subsequences for box 1. Because A_1 is dominated by F, we have that

$$E[\tau(G\{F; A_i^*, i \neq 1\})] - \inf_A E[\tau(G\{A; A_i^*, i \neq 1\})] < \epsilon.$$

Finally, since $\epsilon > 0$ is arbitrary, it follows that

$$E[\tau(G\{F; A_i^*, i \neq 1\})] = \inf_A E[\tau(G\{A; A_i^*, i \neq 1\})],$$

which concludes the proof. ■

3.2.3 A Monte Carlo Method to Estimate the Optimal Expected Search Time

In the two-mode search problem, if each box meets either the condition in Theorem 3.2.5 or in Theorem 3.2.8, then the problem reduces to the single-mode search problem solved in the literature discussed in Chapter 1; otherwise, an optimal policy remains unknown. One way to estimate the optimal expected search time is to discretise the state space (considered as the space of posterior hiding probabilities), and use standard algorithms for the solution of Markov decision processes, such as value iteration (see Section 6.3 of Puterman (2014)). While this approach produces satisfactory results for $n = 2$ (being used to solve the two-box problem shown in Figure 3.2.1), it becomes computationally intractable for $n \geq 3$.

In this subsection, we present a method to estimate the optimal expected search time based on Monte Carlo simulation. To begin, we classify each box into one of three types according to Theorems 3.2.5 and 3.2.8. If a box's search times and detection probabilities satisfy (3.2.4), we say it is a type-S box; if they satisfy (3.2.9), we say it is a type-F box; otherwise, we say it is a type-H box. For a two-mode search problem with n boxes labelled $1, 2, \ldots, n$, let $\mathcal{S}$ denote the set of type-S boxes, $\mathcal{F}$ the set of type-F boxes, and $\mathcal{H}$ the set of type-H boxes.

Recall from Theorem 3.1.1 that, if we fix the within-box subsequence A_i for box i, $i = 1, \ldots, n$, then Gittins indices determine optimal ways to interlace these subsequences to produce a search sequence. According to Theorems 3.2.5 and 3.2.8, no matter the modes used to search boxes in $\mathcal{H}$, as long as Gittins indices are used to choose between boxes, then it is optimal to use only the slow mode of boxes in $\mathcal{S}$, and only the fast mode of boxes in $\mathcal{F}$. For boxes in $\mathcal{H}$, how to choose an optimal mode is unknown; the letter 'H' represents the heuristic approach we shall take to choose modes at which to search such boxes.

Calculation of Gittins indices for boxes in $\mathcal{S}$ and $\mathcal{F}$ is straightforward since, if only one search mode is used, the supremum in (3.1.1) is always obtained at $b = 1$. For boxes in $\mathcal{H}$, given any subsequence, the index calculation is just as easy, as seen in the next lemma.

Lemma 3.2.9 If $i \in \mathcal{H}$, then for any within-box subsequence $A_i = \{a_{i,k}, k \in \mathbb{Z}^+\}$ and any $m \in \mathbb{N}$ we have

$$G_i(m, A_i) = p_i \left\{ \prod_{k=1}^{m} (1 - q_{i,a_{i,k}}) \right\} \frac{q_{i,a_{i,m+1}}}{t_{i,a_{i,m+1}}}.$$

In other words, the supremum in (3.1.1) is always obtained at $b = 1$.

Proof. Since the lemma concerns a single box in $\mathcal{H}$, we omit the subscript i in the proof to simplify notation. With its within-box subsequence fixed at an arbitrary $A \equiv \{a_k : k \in \mathbb{Z}^+\}$, (3.1.1) gives the equation for the Gittins index of the box after some $m \in \mathbb{N}$ searches of the box have been made. Consider the supremum in (3.1.1), namely

$$\sup_{b \in \mathbb{Z}^+} \frac{\sum_{u=m+1}^{m+b} q_{a_u} \left\{ \prod_{v=m+1}^{u-1} (1 - q_{a_v}) \right\}}{\sum_{u=m+1}^{m+b} t_{a_u}}. \tag{3.2.10}$$

To prove that (3.2.10) is attained at $b = 1$, note that increasing b from 1 adds

$$\sum_{u=m+2}^{m+b} q_{a_u} \left\{ \prod_{v=m+1}^{u-1} (1 - q_{a_v}) \right\}$$

to the numerator in (3.2.10), and $\sum_{u=m+2}^{m+b} t_{a_u}$ to its denominator. To complete the proof, for each $u = m+2, m+3, \ldots$, it is sufficient to prove the inequality

$$\frac{q_{a_u} \prod_{v=m+1}^{u-1} (1 - q_{a_v})}{t_{a_u}} < \frac{q_{a_{m+1}}}{t_{a_{m+1}}},$$

where the right-hand term is the value of (3.2.10) with $b = 1$. Consider two cases.

1. $a_{m+1} = f$. Regardless of a_u, we have

$$\frac{q_{a_u} \prod_{v=m+1}^{u-1} (1 - q_{a_v})}{t_{a_u}} < \frac{q_{a_u}}{t_{a_u}} \leq \frac{q_f}{t_f},$$

   where the last inequality follows since the box is not in $\mathcal{S}$.

2. $a_{m+1} = s$. If $a_u = s$, then we have

$$\frac{q_{a_u} \prod_{v=m+1}^{u-1}(1 - q_{a_v})}{t_{a_u}} < \frac{q_{a_u}}{t_{a_u}} = \frac{q_s}{t_s}.$$

   If $a_u = f$, then we have

$$\frac{q_{a_u} \prod_{v=m+1}^{u-1}(1 - q_{a_v})}{t_{a_u}} \leq \frac{q_{a_u}(1 - q_{a_{m+1}})}{t_{a_u}} = \frac{q_f(1 - q_s)}{t_f} < \frac{q_s}{t_s},$$

   where the last inequality follows since the box is not in $\mathcal{F}$. ■

If we are lucky and guess the *right* subsequence for each box in $\mathcal{H}$, then we recover an optimal search sequence. This observation motivates a method to estimate the optimal expected search time as follows.

1. Fix the known optimal subsequence for each box in $\mathcal{S}$ and for each box in $\mathcal{F}$.

2. Generate a random subsequence for each box in $\mathcal{H}$.

3. Use Theorem 3.1.1 to interlace all subsequences optimally, and compute the corresponding expected search time.

4. For every subsequence obtained in step 2, replace each element with the other available search mode to obtain an *opposite* subsequence for each box in $\mathcal{H}$. For example, $(f, s, f, \ldots)$ becomes $(s, f, s, \ldots)$. Repeat step 3.

5. Repeat steps 2–4 a large number of times, and return the minimal expected search time.

How to practically implement the above method will be discussed in Section 3.5.1.

Since our goal is to estimate the optimal expected search time, it would be a wasted effort if the within-box subsequences used in one simulation run were identical to those in a previous run. For a fixed number of runs, we are more likely to obtain near-optimal subsequences if each run contains a distinct set of subsequences for boxes in $\mathcal{H}$. Therefore, in step 4, we construct subsequences opposite to those in the previous run to improve the diversity of our simulation runs. For example, to obtain an estimate based on 1000 runs, we generate 500 independent sets of subsequences, and use both these and the corresponding 500 sets of opposite subsequences.

3.2.4 Lower Bounds on the Optimal Expected Search Time

Write V^* for the optimal expected search time. If there are no boxes in $\mathcal{H}$, then V^* can be readily computed. Otherwise, to compute a lower bound on V^*, consider box $i \in \mathcal{H}$. By definition, we have

$$\frac{q_{i,f}(1-q_{i,s})}{t_{i,f}} < \frac{q_{i,s}}{t_{i,s}} < \frac{q_{i,f}}{t_{i,f}}.$$

Suppose that, for each $i \in \mathcal{H}$, we decrease the slow search time to

$$\widehat{t}_{i,s} \equiv \frac{t_{i,f}q_{i,s}}{q_{i,f}} < t_{i,s}, \tag{3.2.11}$$

and write V' for the optimal expected search time for this modified search problem. It is clear that $V' \leq V^*$, since in this modified version, for each box, the search time of each search mode is less than or equal to its counterpart in the original problem. In addition, in this modified problem, each box $i \in \mathcal{H}$ is type-S, as the sufficient condition (3.2.4) in Theorem 3.2.5 is now met. Thus, all boxes in the modification are either type-F or type-S, so an optimal policy is known, and V' can be readily computed.

Decreasing the slow search time as in (3.2.11) for all boxes in $\mathcal{H}$ is not the only way to modify the search problem so that an optimal policy is known. For each box $i \in \mathcal{H}$, we have up to three additional options:

1. If it does not exceed 1, we may increase the slow detection probability to

$$\widehat{q}_{i,s} \equiv \frac{q_{i,f}t_{i,s}}{t_{i,f}} > q_{i,s},$$

so box i becomes type-S as the sufficient condition (3.2.4) in Theorem 3.2.5 is now met.

2. Decrease the fast search time to

$$\widehat{t}_{i,f} \equiv \frac{q_{i,f} t_{i,s}(1 - q_{i,s})}{q_{i,s}} < t_{i,f}, \tag{3.2.12}$$

so box i becomes type-F as the sufficient condition (3.2.9) in Theorem 3.2.8 is now met.

3. Increase the fast detection probability to

$$\widehat{q}_{i,f} \equiv \min\left(\frac{q_{i,s} t_{i,f}}{t_{i,s}(1 - q_{i,s})}, q_{i,s}\right) > q_{i,f},$$

so either box i becomes type-F as the sufficient condition (3.2.9) in Theorem 3.2.8 is now met, or its fast mode dominates by virtue of having the same detection probability as its slow mode.

However, if both modes $(q_{i,s}, \widehat{t}_{i,s})$ and $(\widehat{q}_{i,s}, t_{i,s})$ are available, since $q_{i,s}/\widehat{t}_{i,s} = \widehat{q}_{i,s}/t_{i,s}$, it follows from Theorem 3.2.5 that it is optimal to use $(\widehat{q}_{i,s}, t_{i,s})$. Therefore, a tighter lower bound on V^* is obtained by decreasing the slow search time of box i than by increasing its slow detection probability. A similar argument shows that, if $\widehat{q}_{i,f} \neq q_{i,s}$, the same holds for the fast mode, namely, a tighter bound is obtained decreasing the fast search time of box i than by increasing its fast detection probability. If $\widehat{q}_{i,f} = q_{i,s}$, then it is optimal to use $(\widehat{q}_{i,f}, t_{i,f})$ over $(q_{i,s}, \widehat{t}_{i,s})$ as $t_{i,f} < \widehat{t}_{i,s}$, so a tighter bound is obtained by decreasing the slow search time of box i than by increasing its fast detection probability.

To summarise, one may solely concentrate on lower bounds obtained by decreasing the slow search time for some boxes in $\mathcal{H}$ according to (3.2.11) and decreasing the fast search time for the other boxes in $\mathcal{H}$ according to (3.2.12). There are $2^{|\mathcal{H}|}$ lower bounds of this kind, and one can choose the largest of these to obtain the tightest such lower bound.

3.3 A Special Case with Two Boxes

This section presents an optimal policy for a particular search problem with two boxes, providing insight into the general two-mode problem. Box 1 has the usual two search modes with respective search times t_f and t_s, and detection probabilities q_f and q_s.

We will assume that neither the condition in Theorem 3.2.5 nor the condition in Theorem 3.2.8 applies, so neither search mode dominates and so box 1 is type-H. Box 2 has only one search mode, with search time t_2 and detection probability $q_2 = 1$. Optimal policies for this seemingly simple search problem demonstrate the complexity of optimal policies in general, and provide insight into the design of effective heuristic policies for the general two-mode problem with n boxes in Section 3.4.

With only 2 boxes, the state of the search can be delineated by a single number p, which represents the object's current hiding probability for box 1. Since $q_2 = 1$, after searching box 2 for the first time, the searcher either finds the object, or learns that the object is in box 1. In the latter case, since $q_f/t_f > q_s/t_s$, it is then optimal to use the fast mode in box 1 repeatedly until finding the object, yielding a further expected search time of t_f/q_f.

Together with Lemma 3.2.7, we deduce that in any state $p \in (0, 1)$, it is sufficient to consider search sequences of the type

$$\underbrace{f, f, \dots f}_{x}, \underbrace{s, s, \dots s}_{y}, 2, f, f \dots, \tag{3.3.1}$$

where f and s represent the fast and the slow mode of box 1, respectively, and 2 represents the sole mode of box 2. In other words, any candidate for an optimal search begins with x fast searches in box 1, followed by y slow searches in box 1, then a search in box 2, which is then followed by an infinite sequence of fast searches in box 1, where $x, y \in \mathbb{N}$. We now make further inference on an optimal policy via two propositions.

Proposition 3.3.1 Define

$$P_1 \equiv \frac{t_f/q_f}{t_2 + t_f/q_f}.$$

An optimal action in state p is to search box 2, if and only if $p \leq P_1$.

Proof. First, suppose that $p \leq P_1$, which is equivalent to

$$(1-p)\frac{1}{t_2} \geq p\frac{q_f}{t_f}.$$

In addition, as the condition in Theorem 3.2.5 does not apply to box 1, it follows that

$$(1-p)\frac{1}{t_2} \geq p\frac{q_f}{t_f} > p\frac{q_s}{t_s}.$$

Invoking Lemma 3.2.9, we conclude that, regardless of the choice of within-box subsequence for box 1, the Gittins index of box 2 is greater than that of box 1. By Theorem 3.1.1, it is optimal to search box 2.

Second, suppose that $p > P_1$, which is equivalent to

$$p\frac{q_f}{t_f} > (1-p)\frac{1}{t_2}.$$

By computing the expected search time for the search sequence $(f, 2, f, f, \ldots)$ and that for $(2, f, f, f, \ldots)$, one can see that the former is smaller. Therefore, it is suboptimal to search box 2. ■

Now let $V_i(x, y)$ denote the expected search time under (3.3.1) if the object is hidden in box i, for $i = 1, 2$ and $x, y \in \mathbb{N}$. Since $q_2 = 1$, we have

$$V_2(x, y) = xt_f + yt_s + t_2.$$

If the object is hidden in box 1, then we can compute that

$$\begin{aligned} V_1(x, y) = &\sum_{j=1}^{x}(1-q_f)^{j-1}q_f \cdot (jt_f) + (1-q_f)^x \sum_{j=1}^{y}(1-q_s)^{j-1}q_s \cdot (xt_f + jt_s) \\ &+ (1-q_f)^x(1-q_s)^y \left(xt_f + yt_s + t_2 + \frac{t_f}{q_f}\right). \end{aligned}$$

Note that the above is the sum of three terms. The first concerns the chance of the object being found on one of the first x searches of box 1, which are all fast. The second concerns the chance of the object being found on one of the next y searches of box 1, which are all slow. The third concerns the chance of the object being found in box 1 after box 2 has been searched.

After some algebraic work, we see that

$$V_1(x,y) = \frac{t_f}{q_f} + (1-q_f)^x \Delta + (1-q_f)^x (1-q_s)^y (t_2 - \Delta),$$

where $\Delta = t_s/q_s - t_f/q_f > 0$. In state p, the expected search time is therefore

$$V(x,y,p) = pV_1(x,y) + (1-p)V_2(x,y). \tag{3.3.2}$$

The next proposition uses (3.3.2) to provide further insight into an optimal policy.

Proposition 3.3.2 Define

$$P_2 \equiv \frac{t_f/q_f}{t_s/q_s} < 1.$$

The unique optimal action in state p is to search fast in box 1, if $p > \max(P_1, P_2)$.

Proof. Consider two cases. First, suppose $P_1 \geq P_2$, which is equivalent to $t_2 \leq \Delta$. So, for any fixed $x \in \mathbb{N}$ and $p \in (0,1)$, $V(x,y,p)$ in (3.3.2) is minimised by taking $y = 0$. Together with Proposition 3.3.1, it follows that the only optimal action is to search fast in box 1 if $p > P_1$.

Second, consider the case $P_1 < P_2$, which is equivalent to $t_2 > \Delta$. Compute

$$V(1,y,p) - V(0,y,p) = (1-p)t_f - pq_f(\Delta + (1-q_s)^y(t_2 - \Delta)).$$

If the preceding is negative for some fixed $y \in \mathbb{N}$, or equivalently, if

$$p > \frac{t_f}{t_f + q_f(\Delta + (1-q_s)^y(t_2-\Delta))},$$

then the search sequence with y and $x = 1$ is better than the search sequence with y and $x = 0$. Since the right-hand side of the preceding is increasing in y, if

$$p > \lim_{y\to\infty} \frac{t_f}{t_f + q_f(\Delta + (1-q_s)^y(t_2-\Delta))} = \frac{t_f}{t_f + q_f\Delta} = \frac{t_f/q_f}{t_s/q_s} = P_2,$$

then for all $y \in \mathbb{N}$, the search sequence with y and $x = 1$ is better than that with y and $x = 0$. In other words, if $p > P_2$, then each search sequence that begins with the slow mode of box 1 is inferior to a search sequence that begins with the fast mode of box 1. Therefore, it is suboptimal to search box 1 slowly. Together with Proposition 3.3.1, it follows that the only optimal action is to search box 1 fast for $p > P_2$. ■

In the case $P_1 \geq P_2$, an optimal policy is completely characterised by Propositions 3.3.1 and 3.3.2. In the case $P_1 < P_2$, however, Propositions 3.3.1 and 3.3.2 only specify an optimal action for $p \leq P_1$ and $p > P_2$. To determine an optimal action in state $p \in (P_1, P_2]$, it is sufficient to compare $V(x, y, p)$ for all $x, y \in \mathbb{N}$ that are relevant. Define

$$h(p) \equiv \frac{p(1-q_f)}{p(1-q_f) + (1-p)},$$

which, assuming the posterior-state formulation, is the new state after a fast search in box 1 does not find the object, if the current state is p. Suppose we have $p = P_2$, then after k consecutive, unsuccessful fast searches in box 1, the state becomes

$$h^{(k)}(P_2) \equiv \underbrace{h \circ h \circ \cdots \circ h}_{k}(P_2).$$

Compute

$$k' \equiv \min\left\{k : h^{(k)}(P_2) \leq P_1\right\}.$$

In other words, if $p = P_2$, then after k' consecutive, unsuccessful fast searches in box 1, it is optimal to next search box 2. It then follows that, for $p \in (P_1, P_2]$, it is sufficient to consider search sequences in (3.3.1) for which $x + y \leq k'$, since after k' consecutive, unsuccessful searches in box 1—whether fast or slow—the resulting state will be less than or equal to P_1, where it is optimal to next search box 2. An optimal action in state p is then the first element in the search sequence that yields a smallest value of $V(x, y, p)$ among those with $x + y \leq k'$.

3.4 Heuristic Policies

We now return to considering a general two-mode search problem with n boxes labelled $1, 2, \ldots, n$. Recall that we can partition $\{1, 2, \ldots, n\}$ into three subsets $\mathcal{S}, \mathcal{F}$ and $\mathcal{H}$ using Theorems 3.2.5 and 3.2.8. While Theorem 3.2.5 proves that the slow mode is optimally designated for boxes in $\mathcal{S}$ and Theorem 3.2.8 the same for the fast mode for boxes in $\mathcal{F}$, it is not at all clear which search mode to use when searching a box in $\mathcal{H}$. We propose two types of heuristic policies for the two-mode problem in Sections 3.4.1 and 3.4.2, and derive corresponding suboptimality bounds in Section 3.4.3. In Section 3.4.4, links between the heuristic policies and the theoretical results of Section 3.2 are explored.

3.4.1 Single-Mode Heuristic Policies

A *single-mode heuristic policy* designates one search mode for each box, then chooses between boxes using Gittins indices, as detailed in Theorem 3.1.1. Clearly we should only consider policies that designate the slow mode for boxes in $\mathcal{S}$ and the fast mode for boxes in $\mathcal{F}$, but the best search mode to designate for boxes in $\mathcal{H}$ is unclear. If we simply choose the search mode leading to the larger Gittins index, which by Lemma 3.2.9 is equivalent to the search mode with the larger detection rate q/t, then the designated search mode is fast for any box in $\mathcal{H}$. This heuristic is referred to as the *detection rate* (DR) heuristic. Note that DR is equivalent to the candidate index policy CI for a superprocess discussed in Section 2.3 of Chapter 2.

Although DR is appealing for its simplicity, it is not always the single-mode heuristic with the smallest expected search time. To find the *best single-mode* (BSM) heuristic, one has to test $2^{|\mathcal{H}|}$ different single-mode policies. The computational effort to determine BSM grows exponentially in $|\mathcal{H}|$. To overcome this computational burden, we propose a heuristic based upon the following idea.

From Theorems 3.2.5 and 3.2.8, for each box $i \in \mathcal{H}$, there exists $\theta_i \in (0, 1)$ that satisfies

$$\frac{q_{i,f}}{t_{i,f}} = \frac{q_{i,s}}{t_{i,s}(1-q_{i,s})^{\theta_i}}.$$

Solving the preceding yields

$$\theta_i = \log\left(\frac{q_{i,s}/t_{i,s}}{q_{i,f}/t_{i,f}}\right) \times \frac{1}{\log(1-q_{i,s})}, \tag{3.4.1}$$

which we interpret as box i's *relative resemblance* to a type-F box compared to a type-S box, since under some limit where $\theta_i \to 0$ (resp. 1), box i becomes type-S (resp. type-F).

We propose a heuristic that chooses a parameter $\theta \in [0,1]$, then designates the slow mode for box $i \in \mathcal{H}$ if $\theta_i \leq \theta$, and the fast mode if $\theta_i > \theta$. Call this heuristic the *adjusted detection rate* (ADR) heuristic with parameter θ, and note that setting $\theta = 0$ retains DR, while setting $\theta = 1$ designates the slow mode for all boxes in $\mathcal{H}$.

To determine the best parameter for ADR, first relabel the boxes so that $0 < \theta_1 \leq \theta_2 \leq \cdots \leq \theta_{|\mathcal{H}|} < 1$. For $j = 1, \ldots, |\mathcal{H}| - 1$, the application of any $\theta \in [\theta_j, \theta_{j+1})$ in ADR results in a single-mode heuristic that designates the slow mode for boxes $1, 2, \ldots, j$, and the fast mode for boxes $j+1, \ldots, |\mathcal{H}|$. Applying $\theta \in [0, \theta_1)$ designates the fast mode for every box while applying $\theta \in [\theta_{\mathcal{H}}, 1]$ designates the slow mode for every box. Since there are only $|\mathcal{H}| + 1$ different single-mode heuristics of this type, the computational effort to find the best of them—which we call the *best adjusted detection rate* (BADR) heuristic—grows linearly in $|\mathcal{H}|$.

Figure 3.4.1 shows, for each $\theta \in [0,1]$, the percentage of times that ADR with parameter θ coincides with BADR for the numerical experiments of Section 3.5. For each choice of n and $|\mathcal{H}|$, there is a clear bias toward smaller θ values, showing that it is usually better to designate the fast mode unless a type-H box closely resembles a type-S box. This bias becomes less pronounced as n grows. Throughout all of these numerical experiments, BADR could always be recovered by taking some $\theta \leq 0.5$ within ADR. This observation indicates that the computational effort involved to find BADR can be reduced by always designating the fast mode for any box $i \in \mathcal{H}$ with $\theta_i > 0.5$, with a negligible impact on performance.

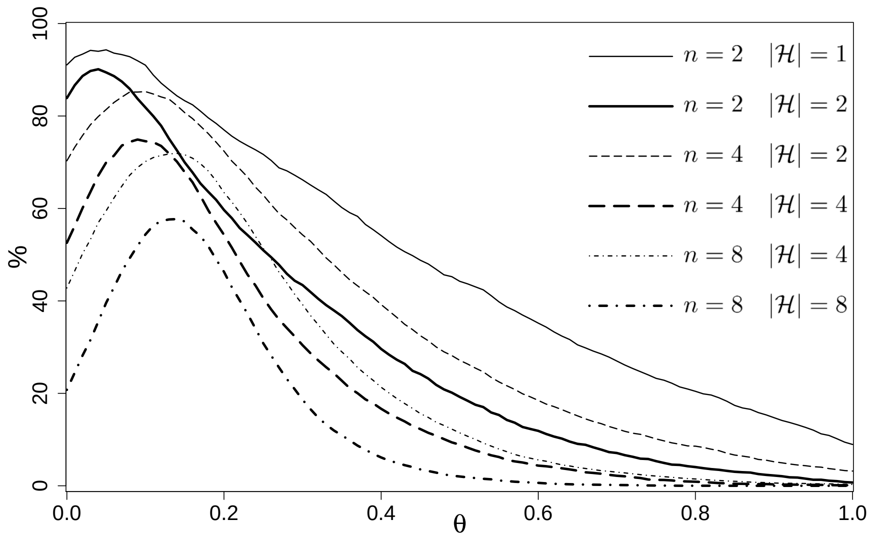

Figure 3.4.1: The percentage of search problems generated in Section 3.5 in which ADR with parameter θ coincides with BADR, for $\theta \in [0, 1]$ and various values of n and $|\mathcal{H}|$.

3.4.2 A Threshold-Type Heuristic Policy

Recall the special search problem with 2 boxes studied in Section 3.3, where any optimal policy uses the fast mode of box 1, a type-H box, if the probability that the object is in that box exceeds a certain threshold. This observation makes intuitive sense, since if it is very likely that the object is hidden in some type-H box, then an optimal policy will likely search that box many times before moving on to any other box. Because, in a type-H box, the fast mode has a larger detection rate than the slow mode, it is intuitive that these many searches will optimally involve at least one fast search. It then follows from Lemma 3.2.7 that it is optimal to make the fast searches first. This argument motivates a heuristic that, for each type-H box, fixes a threshold, then chooses fast if the object's current hiding probability for that box exceeds this threshold.

To come up with a reasonable threshold for this heuristic, consider a type-H box with the usual detection probabilities q_f, q_s and search times t_f, t_s. The *benefit* of searching this box using some mode comes from two sources: the *immediate benefit* and the *future benefit*. The immediate benefit concerns the possibility of finding the object on the search, while the future benefit looks at the information gained about the object's actual location if the search fails. We use detection rates to measure the immediate benefit, namely q_f/t_f for the fast mode and q_s/t_s for the slow mode. For a type-H box, by definition we have $q_f/t_f > q_s/t_s$, so we measure the advantage of the fast mode over the slow mode in immediate benefit by

$$\alpha \equiv \frac{q_f/t_f}{q_s/t_s} - 1, \tag{3.4.2}$$

which is always positive.

To examine the future benefit, we first consider the probability that the object is elsewhere after one or more failed searches. If we search fast for any $x > 0$ time units, then this probability is

$$f(x) = \frac{1-p}{p(1-q_f)^{x/t_f} + 1 - p}, \tag{3.4.3}$$

where p is the object's hiding probability for the type-H box before these failed fast searches. We measure the future benefit by the rate at which the probability in (3.4.3) grows per unit time when the searches begin. Hence, our measure of future benefit for the fast mode is

$$f'(0) = \frac{-p(1-p)\log(1-q_f)}{t_f},$$

and similarly for the slow mode, a term important in the single-box problem of Kadane (2015) discussed in Section 2.3.1, where multiple modes are available and the objective is to maximise the probability of finding the object before a deadline.

While, in a type-H box, the immediate benefit is always larger for the fast mode, the future benefit may go either way. Measure the advantage of the slow mode over the fast mode in future benefit by

$$\beta \equiv \frac{\log(1-q_s)/t_s}{\log(1-q_f)/t_f} - 1, \tag{3.4.4}$$

which does not depend on p. If $\beta \leq 0$ for a type-H box, then both the immediate and future benefit are larger for the fast mode, and it is reasonable to designate fast for that box. Otherwise, the immediate benefit is larger for the fast mode and the future benefit is larger for the slow mode, so there are arguments for using both search modes for the type-H box.

The strength of these respective arguments depends on p. The immediate benefit, which concerns rates of detecting the object, only takes effect if the object is hidden in the searched box. If the object is hidden elsewhere, these detection rates are of no relevance to the searcher. The future benefit concerns the rate at which the searcher's belief that the object is not in the searched box grows, so a larger future benefit leads to the searcher spending a larger proportion of their time searching elsewhere later in the search. Therefore, the future benefit only takes effect if the object is *not* hidden in the searched box. Consequently, for type-H boxes with $\beta > 0$, a natural choice of threshold over which we designate the fast mode is the probability $\widehat{p}$ satisfying $\widehat{p}\alpha = (1 - \widehat{p})\beta$, which solves to

$$\widehat{p} = \frac{\beta}{\alpha + \beta}. \tag{3.4.5}$$

Properties of the threshold will be explored in Section 3.4.4. To demonstrate the practical performance of the threshold for type-H boxes with $\beta > 0$, consider a search problem with 2 boxes, where box 1 is in $\mathcal{H}$ with $q_{1,f} = 0.4$, $q_{1,s} = 0.64$, $t_{1,f} = 1$, and $t_{1,s} = 1.7$. Box 2 has only one search mode, and we vary its detection probability q_2 between 0.3 and 0.9, and its search time t_2 between 0.5 and 2.5. For fifteen choices of (q_2, t_2), Figure 3.4.2 plots the threshold in (3.4.5) against an optimal policy estimated via value iteration, in which the state space $[0, 1]$ is discretised into 10^5 equal-length subintervals. It appears that, when it is optimal to search box 1 for $p > \widehat{p} = 0.738$, the fast mode is mostly optimal, regardless of the values of q_2 and t_2. Hence, our threshold seems to fit well for a wide range of parameters q_2 and t_2. Many further examples with different choices of a type-H box 1 drew a similar conclusion; Figure 3.4.3 presents one such example.

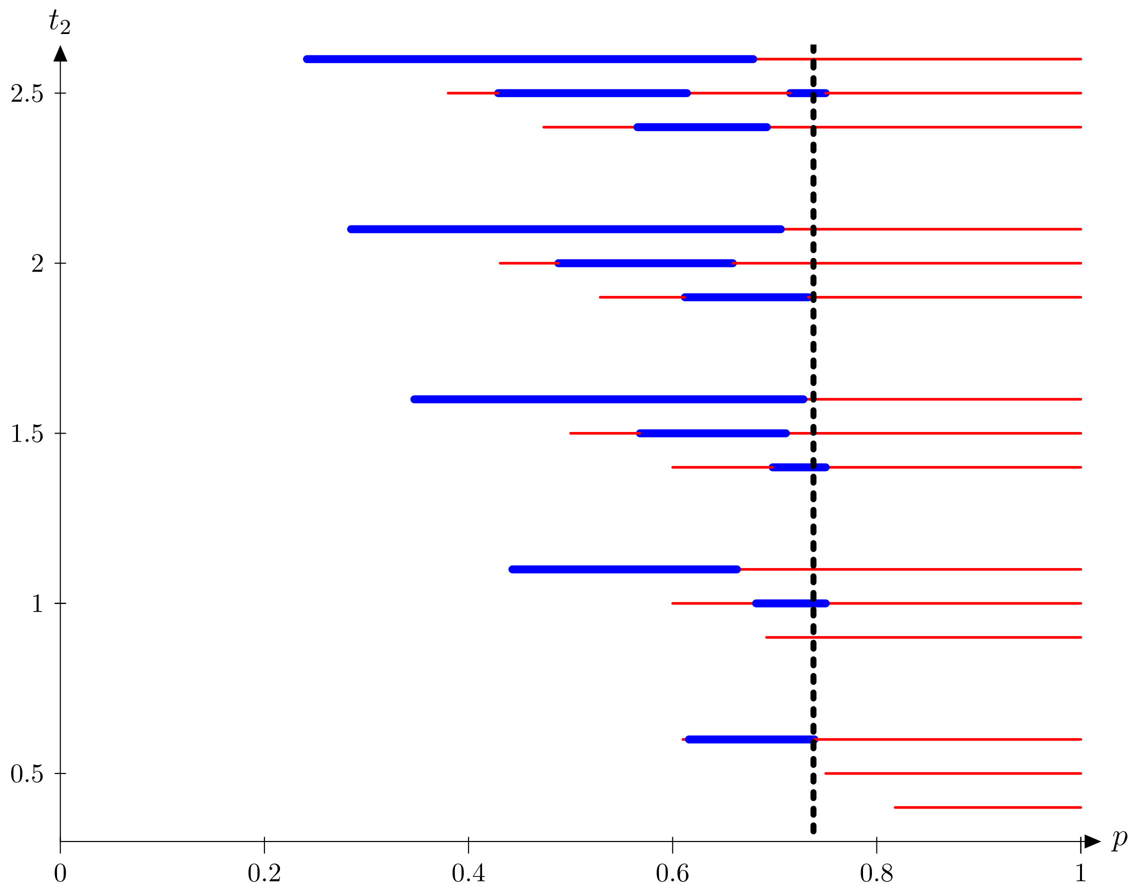

Figure 3.4.2: Optimal actions for $p \in (0, 1)$ in a two-box problem. Box 1 is type-H with $q_{1,f} = 0.4$, $q_{1,s} = 0.64$, $t_{1,f} = 1$ and $t_{1,s} = 1.7$. Box 2 has one search mode with $q_2 = 0.3$ (upper line), 0.6 (middle line), 0.9 (lower line), and $t_2 \in \{0.5, 1, 1.5, 2, 2.5\}$. A blue thick/red thin line indicates a slow/fast search in box 1; no line indicates a search in box 2. The black dotted line is the threshold $\widehat{p}$ in (3.4.5).

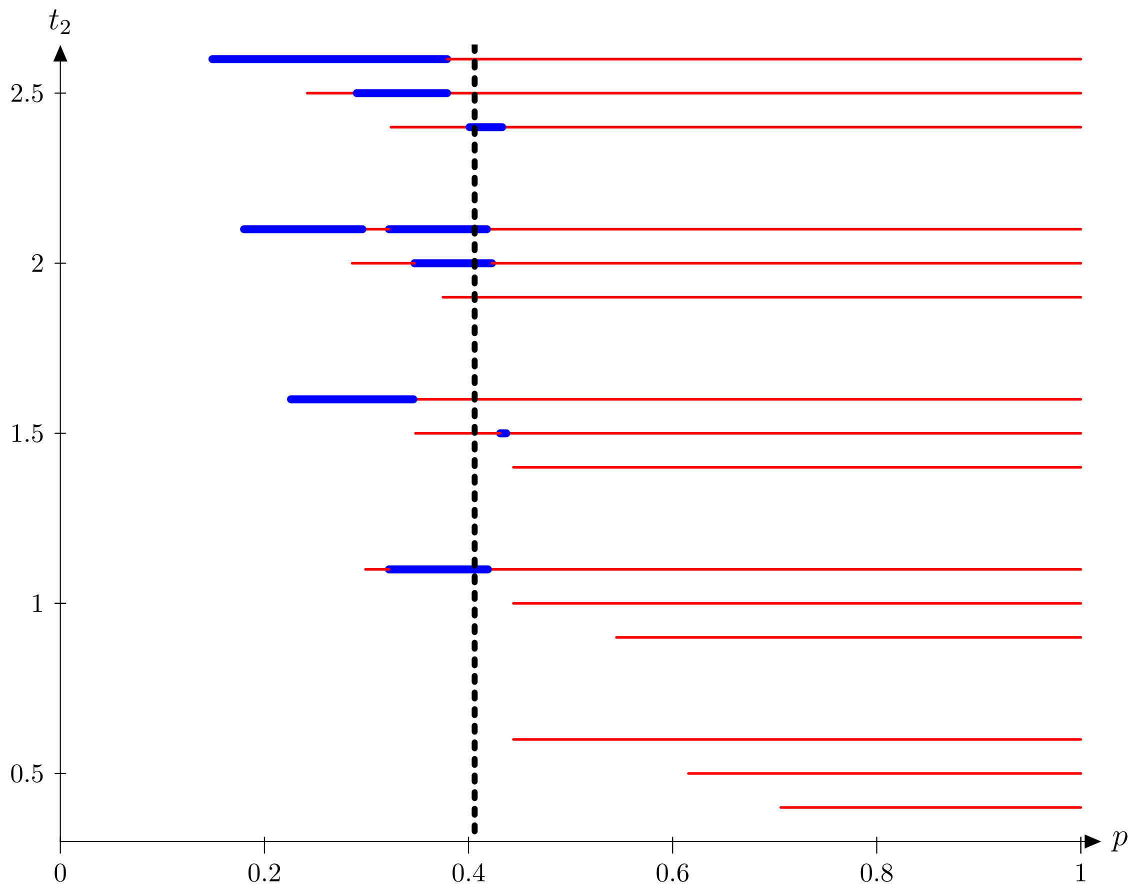

Figure 3.4.3: Optimal actions for $p \in (0,1)$ in a two-box problem. Box 1 is type-H with $q_{1,f} = 0.3$, $q_{1,s} = 0.5$, $t_{1,f} = 0.4$ and $t_{1,s} = 0.73$. Box 2 has one search mode with $q_2 = 0.3$ (upper line), 0.6 (middle line), 0.9 (lower line), and $t_2 \in \{0.5, 1, 1.5, 2, 2.5\}$. A blue thick/red thin line indicates a slow/fast search in box 1; no line indicates a search in box 2. The black dotted line is the threshold $\widehat{p}$ in (3.4.5).

Also seen in Figure 3.4.2, when box 1 is optimally searched for $p \leq \widehat{p}$, an optimal search mode appears to depend heavily on the parameters of box 2. It is more difficult to incorporate such dependence into a heuristic. Therefore, we define a heuristic as follows. For each box in $\mathcal{H}$, if $\beta \leq 0$, we simply designate the fast mode for that box; if $\beta > 0$, we designate the fast mode for $p > \widehat{p}$, and try both the policy which designates the fast mode for $p \leq \widehat{p}$ and that which designates the slow mode for $p \leq \widehat{p}$. For each box, after choosing a search mode, we simply calculate the Gittins index according to the chosen mode, then search a box with a maximal index. This method results in up to $2^{|\mathcal{H}|}$ of these threshold-type policies. We call the one with the smallest expected search time, found only by testing all $2^{|\mathcal{H}|}$, the *best threshold* (BT) heuristic.

3.4.3 Suboptimality Bounds for Heuristic Policies

If $|\mathcal{H}| = 0$, then all of our heuristics are optimal. To bound the suboptimality of our heuristics when $|\mathcal{H}| \geq 1$, we first define a quantity to measure the *distance* of a type-H box from being a type-S box and a type-F box, respectively. For $i \in \mathcal{H}$, let

$$\delta_{i,s} \equiv \frac{q_{i,f}/t_{i,f}}{q_{i,s}/t_{i,s}} - 1, \qquad \delta_{i,f} \equiv \frac{q_{i,s}/t_{i,s}}{(1-q_{i,s})q_{i,f}/t_{i,f}} - 1. \tag{3.4.6}$$

Note that $\delta_{i,s}$ coincides with the measure of the advantage of the fast mode over the slow mode in immediate benefit for box i in (3.4.2).

We now present a proposition, which can be used to bound the suboptimality of our four heuristics in terms of $\delta_{i,s}$ and $\delta_{i,f}$ in (3.4.6).

Proposition 3.4.1 Suppose $|\mathcal{H}| \geq 1$ and write V^* for the optimal expected search time. Write Π for some single-mode policy and V_Π for its corresponding expected search time. For all $i \in \mathcal{H}$, let $\delta_i = \delta_{i,s}$ (resp. $\delta_{i,f}$) if Π designates the slow (resp. fast) mode for box i. We can bound the suboptimality of Π by

$$\frac{V_\Pi - V^*}{V^*} \leq \max_{i \in \mathcal{H}} \delta_i.$$

Proof. Write $\mathcal{H}_S \subseteq \mathcal{H}$ for the subset of $\mathcal{H}$ where the slow mode is designated under policy Π, and $\mathcal{H}_F = \mathcal{H} \setminus \mathcal{H}_S$ for that where the fast mode is designated. To prove the result, we modify the search times for boxes in $\mathcal{H}$. For $i \in \mathcal{H}_S$, we reduce the slow search time of box i to

$$\widehat{t}_{i,s} \equiv \frac{t_{i,f} q_{i,s}}{q_{i,f}} < t_{i,s}. \tag{3.4.7}$$

For $i \in \mathcal{H}_F$, we reduce the fast search time of box i to

$$\widehat{t}_{i,f} \equiv \frac{q_{i,f} t_{i,s}(1 - q_{i,s})}{q_{i,s}} < t_{i,f}. \tag{3.4.8}$$

In this modified search problem, there are no type-H boxes and Π is an optimal policy. Denote the corresponding optimal expected search time in the modified problem by V'_Π. It is clear that $V'_\Pi \leq V^*$, since in the modified problem, for each box, the search time of each search mode is less than or equal to its counterpart in the original problem.

In addition, note that modifying search times changes the set of Gittins indices used by Π to interlace searches of different boxes. So, the search sequence generated by Π in the modified problem is different to that in the original. Suppose we apply the former search sequence to the original problem, and write $V_{\Pi'}$ for the corresponding expected search time. We then have $V_\Pi \leq V_{\Pi'}$, since both values are yielded from the same set of within-box subsequences, but V_Π optimally interlaces these subsequences. Combining $V_\Pi \leq V_{\Pi'}$ with $V'_\Pi \leq V^*$, we have that

$$\frac{V_\Pi - V^*}{V^*} \leq \frac{V_{\Pi'} - V'_\Pi}{V'_\Pi}.$$

It is easily shown that the expected search time under any search sequence can be considered as a linear function of the search times, with positive coefficients that depend only on the prior distribution, the detection probabilities, and the corresponding search sequence. However, when we compute $V_{\Pi'}$ and V'_Π, all of these three things are the same, so the coefficients of the search times are identical. Further, as $V_{\Pi'}$ and V'_Π are single-mode policies, only one coefficient per box, which we denote $c_i > 0$ for box

i, $i = 1, \ldots, n$, is non-zero. Finally, for any box not in $\mathcal{H}$, the search times themselves are also the same. Consequently, we can conclude that

$$\begin{aligned}
\frac{V_{\Pi'} - V'_{\Pi}}{V'_{\Pi}} &= \frac{\sum_{i\in\mathcal{H}_S} c_i(t_{i,s} - \widehat{t}_{i,s}) + \sum_{i\in\mathcal{H}_F} c_i(t_{i,f} - \widehat{t}_{i,f})}{\sum_{i\in\mathcal{H}_S} c_i\widehat{t}_{i,s} + \sum_{i\in\mathcal{H}_F} c_i\widehat{t}_{i,f}} \\
&\leq \max\left[\max_{i\in\mathcal{H}_S}\left(\frac{t_{i,s} - \widehat{t}_{i,s}}{\widehat{t}_{i,s}}\right), \max_{i\in\mathcal{H}_F}\left(\frac{t_{i,f} - \widehat{t}_{i,f}}{\widehat{t}_{i,f}}\right)\right] \\
&= \max_{i\in\mathcal{H}} \delta_i,
\end{aligned}$$

where the last line follows from (3.4.7) and (3.4.8). ■

Corollary 3.4.2 Suppose $|\mathcal{H}| \geq 1$ and write V^* for the optimal expected search time. Write V_{DR}, V_{BADR}, V_{BSM} and V_{BT} for the expected search time for the heuristics DR, BADR, BSM, and BT, respectively. We can bound the suboptimality of these heuristics as follows.

$$\frac{V_{\text{DR}} - V^*}{V^*} \leq \max_{i\in\mathcal{H}} \delta_{i,f}. \tag{3.4.9}$$

$$\frac{V_{\text{BADR}} - V^*}{V^*} \leq \min\{\max_{i\in\mathcal{H}} \delta_{i,s}, \max_{i\in\mathcal{H}} \delta_{i,f}\}. \tag{3.4.10}$$

$$\frac{V_{\text{BSM}} - V^*}{V^*} \leq \max_{i\in\mathcal{H}}\ \min\left\{\delta_{i,s}, \delta_{i,f}\right\}. \tag{3.4.11}$$

$$\frac{V_{\text{BT}} - V^*}{V^*} \leq \max_{i\in\mathcal{H}} \delta_{i,f}. \tag{3.4.12}$$

Proof. All bounds are derived from Proposition 3.4.1. The bound in (3.4.9) follows because DR designates fast for boxes in $\mathcal{H}$. The bound in (3.4.10) follows because BADR compares several single-mode policies, including the one that designates fast for all boxes in $\mathcal{H}$ and the one that designates slow for all boxes in $\mathcal{H}$. The bound in (3.4.11) follows because BSM compares all single-mode policies. Finally, the bound for BT in (3.4.12) follows because DR is one of the candidates for BT. ■

Among those in Corollary 3.4.2, the bounds for V_{DR} and V_{BT} are the weakest, while that for V_{BSM} is the strongest. If we consider some limit in which, for all $i \in \mathcal{H}$, either $\delta_{i,s} \downarrow 0$ or $\delta_{i,f} \downarrow 0$, then BSM approaches optimality. In addition, if $\delta_{i,s} \downarrow 0$ for all $i \in \mathcal{H}$ or $\delta_{i,f} \downarrow 0$ for all $i \in \mathcal{H}$, then BADR also approaches optimality. Finally, if

$\delta_{i,f} \downarrow 0$ for all $i \in \mathcal{H}$, then all four heuristics approach optimality. Note that all of the bounds in Corollary 3.4.2 do not depend on the object's hiding probabilities at all; they depend only on detection probabilities and search times. While these bounds provide analytical insight, they do not necessarily predict heuristic performance well. In the numerical experiments of Section 3.5, the heuristics consistently and substantially outperform these bounds.

3.4.4 Links between Immediate/Future Benefit and Theoretical Results

This subsection explores the links between the notions of immediate and future benefit introduced in Section 3.4.2 and the theoretical results of Sections 3.2.1 and 3.2.2, leading to the discovery of a third source of benefit to searching a box. Further, we explain the link between our use of future benefit and its role in the optimal solution of the single-box, deadline-search model of Kadane (2015).

Firstly, extending the immediate and future benefit definitions to type-S and type-F boxes provides an intuition for Theorems 3.2.5 and 3.2.8, which detailed when one mode dominates for a given box.

Proposition 3.4.3 The following hold for any box.

1. The immediate and future benefit are larger for its slow mode if and only if the box is type-S.

2. The immediate and future benefit are larger for its fast mode if the box is type-F.

Proof. Write q_f and q_s for the usual detection probabilities and t_f and t_s for the usual search times of the box. We prove 1. and 2. separately as follows.

1. The forwards implication is easy to prove; if the immediate benefit is larger for the slow mode of some box, then $q_s/t_s \geq q_f/t_f$, so the sufficient condition (3.2.4) in Theorem 3.2.5 is satisfied and the box is type-S.

To prove the backwards implication, consider a type-S box. By definition, the sufficient condition (3.2.4) in Theorem 3.2.5 is satisfied, so the immediate benefit is larger for the slow mode. To show that the future benefit is also larger for its slow mode, let

$$A = \int_{1-q_s}^{1-q_f} \frac{du}{u}, \quad B = \int_{1-q_f}^{1} \frac{du}{u}, \quad C = t_s - t_f, \quad D = t_f.$$

Then we have

$$\frac{A+B}{C+D} = \frac{-\log(1-q_s)}{t_s} \quad \text{and} \quad \frac{B}{D} = \frac{-\log(1-q_f)}{t_f}. \tag{3.4.13}$$

Further, considering the area beneath the curve $1/u$ for $u \in [1-q_s, 1-q_f]$ and $u \in [1-q_f, 1]$ shows that we have

$$\frac{A}{C} > \frac{(q_s - q_f)}{(t_s - t_f)} \cdot \frac{1}{(1-q_f)} \quad \text{and} \quad \frac{B}{D} < \frac{q_f}{t_f} \cdot \frac{1}{(1-q_f)}.$$

However, as the box is type-S, we have $q_s/t_s \geq q_f/t_f$, from which it follows that $(q_s - q_f)/(t_s - t_f) \geq q_f/t_f$ and hence from the above that $A/C > B/D$. We now infer that

$$\frac{A+B}{C+D} > \frac{B}{D},$$

which, combined with (3.4.13), completes the proof for a type-S box.

2. Consider a type-F box. By definition, the sufficient condition (3.2.9) in Theorem 3.2.8 is satisfied, so the immediate benefit is larger for its fast mode. To show the future benefit is also larger for its fast mode, we again consider the area beneath the curve $1/u$ for $u \in [1-q_s, 1-q_f]$ and $u \in [1-q_f, 1]$, which shows that we have

$$\frac{A}{C} < \frac{(q_s - q_f)}{(t_s - t_f)} \cdot \frac{1}{(1-q_s)} \quad \text{and} \quad \frac{B}{D} > \frac{q_f}{t_f}.$$

Since the box is type-F, it follows that

$$\frac{q_f(1-q_s)}{t_f} \geq \frac{q_s}{t_s} > \frac{(q_s - q_f)}{(t_s - t_f)},$$

which leads to

$$\frac{A}{C} < \frac{B}{D} \Rightarrow \frac{A+B}{C+D} < \frac{B}{D}.$$

Combining the preceding with (3.4.13) completes the proof for a type-F box. ■

Recall α from (3.4.2), the advantage of the fast mode over the slow mode in immediate benefit, and β from (3.4.4), the advantage of the slow mode over the fast mode in future benefit. If we extend its definition to all box types, we may also link the threshold $\widehat{p} \equiv \beta/(\alpha+\beta)$ used in the BT heuristic policy to the immediate benefit, the future benefit, and the theoretical results of Sections 3.2.1 and 3.2.2. These connections, summarised in Table 3.4.1, follow from Proposition 3.4.3 and the following lemma, which shows that, for any type of box, the sign of $\widehat{p}$ depends entirely on the sign of β.

Lemma 3.4.4 For any box, $\alpha + \beta > 0$.

Proof. Consider a box with $q_f = t_f$ and $q_s = t_s$. Any such box is clearly type-S, so by Proposition 3.4.3, its slow mode has the larger future benefit; therefore, for any $0 < q_f < q_s < 1$, we have

$$\frac{q_s}{q_f} < \frac{\log(1-q_s)}{\log(1-q_f)}. \tag{3.4.14}$$

Now consider a general box with search parameters q_f, q_s, t_f and t_s. Let

$$a \equiv \sqrt{\frac{q_s/t_s}{q_f/t_f}}.$$

Then

$$\left(a - \frac{1}{a}\right)^2 = \frac{q_s/t_s}{q_f/t_f} + \frac{q_f/t_f}{q_s/t_s} - 2 \geq 0. \tag{3.4.15}$$

Yet, by (3.4.14), we have

$$\alpha + \beta = \frac{\log(1-q_s)/t_s}{\log(1-q_s)/t_f} + \frac{q_f/t_f}{q_s/t_s} - 2 > \frac{q_s/t_s}{q_f/t_f} + \frac{q_f/t_f}{q_s/t_s} - 2,$$

which, combined with (3.4.15), completes the proof. ■

By Table 3.4.1, a box is type-S if and only if $\widehat{p} \geq 1$, and any type-F box has $\widehat{p} \leq 0$. Recall that, in the BT heuristic, for $p \geq \widehat{p}$ the fast mode is designated and for $p \leq \widehat{p}$

Table 3.4.1: A summary of the links between the theoretical results of Sections 3.2.1 and 3.2.2, the notions of immediate and future benefit, and the threshold $\widehat{p}$ in (3.4.5).

Range of $\widehat{p}$	Larger Immediate Benefit	Larger Future Benefit	Box Type
$\widehat{p} \leq 0$	Fast	Fast	Type-H or Type-F
$0 < \widehat{p} < 1$	Fast	Slow	Type-H
$\widehat{p} \geq 1$	Slow	Slow	Type-S

we try designating both modes. Therefore, if we were to use the threshold for type-S and type-F boxes, they would both be designated their optimal search mode, showing that the threshold choice is consistent with the theoretical results of Sections 3.2.1 and 3.2.2.

Further, Table 3.4.1 shows that boxes where the immediate and future benefit are both larger for the slow mode are exclusively type-S. However, the same cannot be said for the fast mode and type-F; there do exist type-H boxes where the immediate and future benefit are both larger for the fast mode. An obvious question is the following: are all such type-H boxes optimally designated the fast mode? The following example shows that the answer is no.

Example 3.4.5 Consider the two-box problem in Section 3.3, where box 1 has hiding probability p and the usual search parameters q_f, q_s, t_f and t_s, whilst box 2 has only one search mode with search time t_2 and detection probability $q_2 = 1$.

Let $q_f = 0.3$, $q_s = 0.4$, $t_f = 1$ and $t_s = 1.44$. Then box 1 is type-H with $\beta = -0.005$; therefore, the fast mode has the larger immediate and future benefit. Suppose $t_2 = 50$. By Proposition 3.3.1, it is optimal to search box 2 first if and only if $p \leq 1/16$. Further, it is easy to verify that for $p \in [1/16, 2/23]$, after one search of box 1 using either mode, it is optimal to search box 2. Therefore, by the analysis in

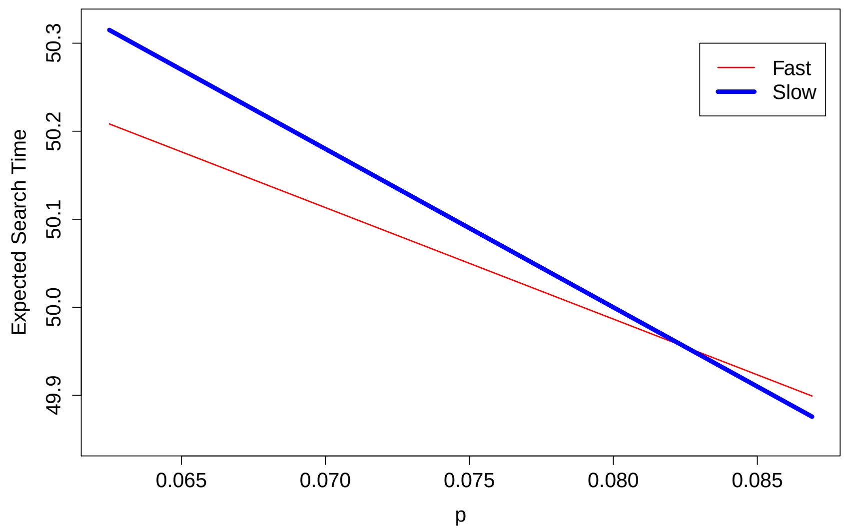

Figure 3.4.4: For Example 3.4.5, the expected search time under $(f, 2, f, f, \ldots)$ (a thin red line, labelled 'Fast') and under $(s, 2, f, f, \ldots)$ (a thick blue line, labelled 'Slow') for $p \in [1/16, 2/23]$. On this range of p, the sequence with the smaller expected search time of these two is the optimal search sequence.

Section 3.3, for $p \in [1/16, 2/23]$, an optimal search sequence is either

$$(f, 2, f, f, \ldots) \quad \text{or} \quad (s, 2, f, f, \ldots), \tag{3.4.16}$$

where f (resp. s) represents a fast (resp. slow) search of box 1 and 2 represents a search of box 2. The expected search time under either sequence in (3.4.16) may be calculated using (3.3.2); both expected search times are shown for $p \in [1/16, 2/23]$ in Figure 3.4.4. Figure 3.4.4 shows it is optimal to search box 1 first using the slow mode for p close to $2/23 \approx 0.087$, despite the slow mode having the smaller immediate and future benefit. However, for $p = 2/23$, where using the slow mode has the biggest advantage, the percentage increase in expected search time searching box 1 fast first instead of slow is less than 0.05%.

A Third Source of Benefit Example 3.4.5 shows there must be a third source of benefit to searching a box, or the slow mode of its box 1 would never be optimal. A few other examples, not limited to the special two-box problem in Section 3.3, were found where it is optimal to use the slow mode of a box despite the fast mode having the larger immediate and future benefit; all share the following three properties present in Example 3.4.5. Firstly, for that box, β is very close to 0, so the advantage of the fast mode over the slow mode in future benefit is very small. Secondly, the range of hiding probabilities where the slow search of the box is optimal is very small. Thirdly, where the slow mode of the box is optimal, the increase in expected search time from the optimum if the slow mode is ignored is close to negligible. Therefore, the effect of this third source of benefit on the expected search time is not very strong compared to the immediate and future benefits.

A fourth observation across the aforementioned examples reveals the nature of the third type of benefit, most efficiently explained for the two-box problem in Section 3.3, where, recall, box 1 has hiding probability p and the usual search parameters q_f, q_s, t_f and t_s, whilst box 2 has only one search mode with search time t_2 and detection probability $q_2 = 1$.

Recall P_1 from Proposition 3.3.1; by direct computation, it is easy to verify that the search sequences $(f, 2, f, f, \ldots)$ and $(2, f, f, f, \ldots)$ have the same expected search time if and only if $p = P_1$. By Proposition 3.3.1, the latter sequence is optimal when $p = P_1$; it follows that P_1 is the unique hiding probability at which it is both optimal to search box 1 fast and search box 2. For $m \in \mathbb{Z}^+$, write

$$P_{m,f} \equiv \frac{t_f/q_f}{t_2(1-q_f)^m + t_f/q_f}; \tag{3.4.17}$$

in other words, after m fast searches of box 1, the posterior hiding probability for box 1 is P_1, and the searcher is indifferent between searching box 1 fast and searching box 2.

Throughout the aforementioned examples where it is optimal to use the slow mode of box 1 despite a smaller immediate and future benefit, the ranges of p where the slow mode is optimal centre on $P_{m,f}$ for some $m \in \mathbb{Z}^+$, particularly for smaller m. Example 3.4.5 demonstrates this effect since $P_{1,f} = 2/23 \approx 0.087$. This observation is explained by no preference of a box to search next being unattractive to the searcher, a feature prominent in several other works in the search literature. For example, in the two-box problems of Flesch et al. (2009) and Kan (1974) (discussed, respectively, in Sections 1.2.3 and 1.2.4 of Chapter 1) where the searcher may stop or delay the search, the stop or delay option is optimal when the searcher is approximately equally attracted to both boxes. Further, consider the two-sided search game in n boxes, the focus of Part II of this thesis, where the object is an intelligent hider who actively tries to avoid detection by choosing the hiding probabilities. Gittins and Roberts (1979) show that equating Gittins indices across all n boxes is often optimal for the hider, and very close to optimal otherwise. The latter example will be discussed much further in Part II of this thesis; the key takeaway here is the following. If $p = P_{m,f}$ for some $m \in \mathbb{Z}^+$, a sequence of m fast searches of box 1 leads to the dreaded $p' = P_1$, where p' is the posterior hiding probability in box 1. Therefore, if the advantages in future and immediate benefit of fast over slow are not too large, it may be optimal for the searcher to search slow to avoid $p' = P_1$. The only reason such a situation can occur is because searches are made in discrete time; in other words, the searcher can only search box 1 fast (resp. slow) for time mt_f (resp. mt_s) for some $m \in \mathbb{Z}^+$.

We only see slow optimal with $p = P_{m,f}$ for small $m \in \mathbb{Z}^+$ for two reasons.

(A.) The earlier in the search a posterior p' is attained, the bigger its effect on the expected search time, since the probability that the object remains undetected decreases as time passes.

(B.) The smaller m, the smaller $P_{m,f}$, so the smaller the influence of the immediate benefit if $p = P_{m,f}$. The immediate benefit is not only larger for the fast mode, but, since $\beta < 0$, by Lemma 3.4.4, we must have $\alpha > |\beta|$; in other words, the immediate benefit favours fast more than the future benefit does. Therefore, the smaller m, the greater the appeal of the slow mode over the fast mode.

Notice from Table 3.4.1 that the same effect never occurs with the modes reversed;

in other words, there are no search problems where slow has the greater immediate and future benefit, but it is optimal to use the fast mode to avoid $p' = P_1$. We offer the following explanation. When fast has the advantage in both benefits, both (A.) and (B.) above strengthen the case for using slow when $p = P_{m,f}$ for small $m \in \mathbb{Z}^+$. Define $P_{m,s}$ as in (3.4.17) but for the slow mode. When slow has the advantage in both benefits, (A.) still applies and strengthens the case for using fast when $p = P_{m,s}$ for small $m \in \mathbb{Z}^+$. However, (B.) is adjusted to the following, which weakens the case for fast for small m.

(B'.) The smaller m, the smaller $P_{m,s}$, so the larger the influence of the future benefit if $p = P_{m,s}$. The future benefit is not only larger for the slow mode, but, since $\alpha < 0$, by Lemma 3.4.4, we must have $\beta > |\alpha|$; in other words, the future benefit favours slow more than the immediate benefit does. Therefore, the smaller m, the lesser the appeal of the fast mode over the slow mode.

We believe that since (A.) and (B'.) are at cross purposes when slow has the advantage in both benefits, it is never optimal to use the fast mode to avoid $p' = P_1$.

The Use of Future Benefit in Kadane (2015) Recall from Section 2.3.1 the following search problem of Kadane (2015). There is a single box, guaranteed to contain the object, with an arbitrary number of search modes available. The objective is to maximise the probability of finding the object before a known deadline d, and, so the search may continue until time d, the last search of a box at a mode with parameters q and t is allowed to be curtailed at time st with detection probability $1 - (1 - q)^s$ for any choice of $s \in (0, 1]$. Kadane (2015) shows it is optimal to always search the box using the same mode, that with the largest value of $-\log(1 - q)/t$, which in Section 3.4.2 we defined as the future benefit of a search mode.

In our search problem with multiple boxes, using a mode with a larger future benefit is beneficial since it provides more information about the object's true location. Yet, this cannot be the reason that future benefit determines an optimal mode in

Kadane's deadline problem, as there is only one box! The following demonstrates the difference between Kadane's and our use of the future benefit.

If the searcher uses the slow (resp. fast) mode up until the deadline d, they fail to find the object with probability $(1-q_s)^{d/t_s}$ (resp. $(1-q_f)^{d/t_f}$). It follows that the searcher has a greater probability of having found the object by d searching slow rather than fast if and only if

$$\frac{-d\log(1-q_s)}{t_s} > \frac{-d\log(1-q_f)}{t_f};$$

in other words, slow is optimal in the deadline search if and only if it has the larger future benefit - as deduced by Kadane.

When the objective is switched to minimising the expected search time, as shown in the final paragraphs of Chapter 2, for a single-box problem, the immediate benefit determines the optimal mode. By Table 3.4.1, if the threshold $\widehat{p}$ from (3.4.5) is between 0 and 1, it is optimal to search the single box fast even when slow has the larger future benefit. The reason for the change in optimal mode from the deadline search is that searching fast gives the searcher a chance of finding the object sooner due to its shorter search time, relevant to the searcher who minimises the expected search time. On the other hand, the deadline searcher does not care how quickly the object is found, as long as it is found before the deadline.

When we add other boxes to the minimising-expected-search-time problem, the future benefit helps gain information about the object's location. We can link this information gain to the optimality of the future benefit in Kadane's single-box deadline search in the following way.

Suppose we search a box for a fixed amount of time using some mode. Kadane shows that, when there is a single box, the larger the future benefit of the mode, the greater the probability of finding the object by the end of the search. This means, in our multiple-box problem, if we use a mode with a larger future benefit on our box, the search is less likely to fail if the object is in our box. Therefore, if the search does fail, the larger the future benefit, the stronger the evidence that the object is elsewhere, which is precisely how we derived the future benefit in Section 3.4.2.

3.5 Numerical Results

This section presents several numerical experiments. To generate search times and detection probabilities for a box, we first draw

$$q_s \sim U(0.2, 0.9) \quad t_f \sim U(0.1, 4.5) \quad a \sim U(0.1, 1) \quad b \sim U(0.1, 1), \tag{3.5.1}$$

and then set $q_f = aq_s$ and $t_s = t_f/b$. For a search problem with n boxes, we control the number of boxes in $\mathcal{H}$. If $|\mathcal{H}| = 0$, then the theoretical results of Section 3.2 provide an optimal solution. As $|\mathcal{H}|$ increases, the extent to which this theory can be applied decreases, so we want to see how our heuristics perform for different values of $|\mathcal{H}|$.

Our sampling plan in (3.5.1) satisfies several desirable properties. First, the measures of immediate benefit q_s/t_s and q_f/t_f are identically distributed and conditionally independent given q_s and t_f. It can be shown that the probabilities that any drawn box is in $\mathcal{S}$, $\mathcal{F}$, and $\mathcal{H}$, are 0.5, 0.1727, and 0.3273, respectively. In addition, since the advantage of the fast mode over the slow mode in immediate benefit, namely α from (3.4.2), satisfies $\alpha = a/b - 1$, it follows that $\alpha + 1$ and $(1 - q_s)^{-1}$ are independent and both have an upper limit of 10. Hence, rearrangement of the condition of Theorem 3.2.8 in (3.2.9) shows that any draws of q_s and t_f do not preclude a drawn box from being any of the three box types.

3.5.1 Estimating an Expected Search Time

In this section, we describe a method for estimating the expected search time either in step 3 of the Monte Carlo (MC) method described in Section 3.2.3, or of a candidate from a heuristic policy (i.e., one of the $2^{|\mathcal{H}|}$ policies evaluated by BT or BSM.)

Theoretically, any heuristic candidate or iteration of MC induces a set of n within-box subsequences and, using Theorem 3.1.1, a search sequence ξ which optimally interlaces them using Gittins indices. Recall that, with fixed within-box subsequences, the two-mode problem reduces to the classic time-dependent search problem discussed in Chapters 1 and 2. Therefore, (2.1.4) may be used to calculate the expected search time, $V(\xi)$, under ξ.

However, in practice, we cannot generate a random subsequence of modes of infinite length in step 2 of MC, and, for a BT candidate, the next mode in the within-box subsequence of a box may depend on its current posterior probability. Even for a single-mode candidate, where each within-box subsequence is known to be $(f, f, \ldots)$ or $(s, s, \ldots)$, we still could not evaluate the infinite sum in (2.1.4) exactly.

We can, however, obtain a lower bound on $V(\xi)$ by, for any $L \in \mathbb{Z}^+$, calculating the sum of the first L terms of (2.1.4), determining the next mode in each box's subsequence on the fly. To obtain an upper bound on $V(\xi)$, let $\bar{\xi}$ be the search sequence identical to ξ for the first L searches but, from the $(L+1)$th search onwards, repeatedly cycles through the boxes $1, \ldots, n$ in that order. To specify the search modes used by $\bar{\xi}$ during these cycles, we split into three cases.

If ξ is a single-mode candidate, let $\bar{\xi}$ cycle the boxes using the same mode for box i as ξ, for $i = 1, \ldots, n$. Therefore, ξ and $\bar{\xi}$ use the same within-box subsequence for each box. Since ξ is a Gittins index policy, $V(\bar{\xi})$ is an upper bound on $V(\xi)$ by Theorem 3.1.1, and, due to the cyclic, single-mode nature of $\bar{\xi}$ from its Lth search onwards, we may calculate $V(\bar{\xi})$ exactly using the infinite sum in (2.1.4).

If ξ is from the MC method, let $\bar{\xi}$ cycle the boxes using only fast modes for boxes in $\mathcal{H}$ and $\mathcal{F}$, and only slow modes for boxes in $\mathcal{S}$. Since the modes in the within-box subsequences for boxes in $\mathcal{H}$ used by ξ from its Lth search onwards are yet to be determined, we can assume that they are all fast. Therefore, ξ and $\bar{\xi}$ use the same within-box subsequence for each box, and we again conclude that $V(\bar{\xi})$ is a calculable upper bound on $V(\xi)$.

Finally, if ξ is a two-mode candidate for BT, let $\bar{\xi}$ cycle the boxes using a mode with detection probability $q_{i,f}$ and search time $t_{i,s}$ for box $i \in \mathcal{H}$, and using the same mode as ξ for boxes in $\mathcal{S}$ and $\mathcal{F}$. Suppose we replace the within-box subsequences of $\bar{\xi}$ with those of ξ, but still cycle the boxes. Then the expected search time will decrease from $V(\bar{\xi})$ to some $\bar{V}$, since the detection probability and search time of each replacement mode is no worse than its predecessor. If we now, instead of cycling the boxes, follow a Gittins index policy, by Theorem 3.1.1, the expected search time will decrease again, this time from $\bar{V}$ to $V(\xi)$. Therefore, $V(\bar{\xi})$ is again an upper bound on $V(\xi)$, again calculated exactly using (2.1.4) due to the cyclic, single-mode nature of $\bar{\xi}$ from its Lth search onwards.

As L increases, the probability that the object remains hidden after the first L searches (when $\bar{\xi}$ and ξ first differ) decreases. Therefore, the percentage increase from the lower to the upper bound decreases as L increases. We choose L large enough so that this increase is less than 0.001%, which ensures that our estimate of $V(\xi)$, which we take to be its lower bound, is within 0.001% of the true value.

3.5.2 Estimating the Optimal Expected Search Time via Monte Carlo Simulation

To assess the effectiveness of the Monte Carlo (MC) method proposed in Section 3.2.3, we compare its output with the optimal expected search time obtained via value iteration for search problems with 2 boxes, where the latter is computationally feasible.

With only 2 boxes, assuming the posterior-state formulation, the state of the search can be delineated by $p \in [0, 1]$, namely the object's current hiding probability for box 1. By dividing the continuous state space $[0, 1]$ into 10^5 equal-length subintervals, we formulate a semi-Markov decision process identical to that described in Section 3.1, but with a finite number of states. At the start of the search, the state is whichever of the 10^5 subintervals contains the initial hiding probability p. When a search is made, we use the midpoint of this subinterval to calculate a posterior p, then take the new state to be whichever of the 10^5 subintervals contains the posterior. We continue to use midpoints to make further posterior calculations. A value iteration algorithm computes the optimal expected search time in each state, stopping when the values in successive iterations are within 10^{-6} for all states.

In Section 3.4.4, it was shown that there do exist search problems where an optimal policy involves the slow mode for a type-H box in which the future benefit is larger for the fast mode. However, according to numerical results from this value iteration, such problems are extremely rare, and in those rare cases, the slow mode is optimal only for a very small subinterval of the state space. Furthermore, if we ignore the slow mode altogether for such boxes, the increase in expected search time from the optimum is close to negligible. For these reasons, we improve the efficiency of our MC method by fixing the within-box subsequence for such boxes to consist of only the fast mode.

To assess our MC method for search problems with 2 boxes, we first use (3.5.1) and rejection sampling to generate the search times and detection probabilities of these 2 boxes such that $|\mathcal{H}| = 1$. We set p equal to 0.5 and run the MC method to estimate the optimal expected search time, then do the same for $p = 0.9$. We repeat the preceding 2000 times to collect data. Finally, we redo the whole procedure with $|\mathcal{H}| = 2$. The results of the MC method for various run lengths are reported in the left-hand side of Table 3.5.1 as average percentages over the optimal values obtained from value iteration (the 'ensemble method' in the right-hand side of Table 3.5.1 will be explained in Section 3.5.3). As seen in Table 3.5.1, for $|\mathcal{H}| = 2$, with 10000 runs (5000 sets of independent subsequences and 5000 sets of opposite subsequences), the MC method estimates the optimal value on average within 0.12%, and the improvement with more runs is small.

3.5.3 Performance of Heuristics with $n = 2$

For search problems with $n = 2$ boxes, we can evaluate our heuristics against optimal values obtained via value iteration. Recall that our four heuristics from Section 3.4 are

Table 3.5.1: Performance of the MC and ensemble (to be explained in Section 3.5.3) methods for search problems with $n = 2$ boxes, reported as average percentage above the optimum calculated via value iteration.

Number of runs	MC method				Ensemble method			
	$\lvert\mathcal{H}\rvert = 1$		$\lvert\mathcal{H}\rvert = 2$		$\lvert\mathcal{H}\rvert = 1$		$\lvert\mathcal{H}\rvert = 2$	
	$p = 0.5$	$p = 0.9$	$p = 0.5$	$p = 0.9$	$p = 0.5$	$p = 0.9$	$p = 0.5$	$p = 0.9$
10K	0.0464	0.0467	0.1198	0.1026	0	0.0002	0.0011	0.0011
100K	0.0310	0.0300	0.0725	0.0627	0	0.0001	0.0009	0.0008
200K	0.0284	0.0267	0.0636	0.0562	0	0.0001	0.0009	0.0007
400K	0.0254	0.0238	0.0568	0.0497	0	0	0.0008	0.0007

the detection rate (DR) heuristic, the best adjusted detection rate (BADR) heuristic, the best single-mode (BSM) heuristic, and the best threshold (BT) heuristic.

To assess our four heuristics, we first fix $|\mathcal{H}| = 1$ and use the same 2000 pairs of boxes generated in Section 3.5.2. However, for each pair, instead of using only $p = 0.5$ and $p = 0.9$, we take the midpoints of the 10^5 subintervals used in the value iteration as our values of p. For each p, the expected search time of each heuristic is computed using the methods of Section 3.5.1 and then expressed as a percentage over the corresponding optimal value obtained via value iteration. We then repeat the procedure for $|\mathcal{H}| = 2$; Table 3.5.2 displays the results.

As seen in Table 3.5.2, all four heuristics are close to optimal on average, although their performance degrades for $|\mathcal{H}| = 2$. The DR heuristic achieves within 0.001% of optimality for 75% of the search problems with $|\mathcal{H}| = 2$, which suggests that a large proportion of boxes in $\mathcal{H}$ are optimally designated the fast mode. Recall that, by definition, the other three heuristics must perform at least as well as DR. As seen in the last two rows of Table 3.5.2, in the problems where DR is suboptimal, the other three heuristics show a remarkable improvement on DR, which can perform poorly.

Table 3.5.2: Performance of heuristics for search problems with $n = 2$ boxes, reported as percentage above the optimum calculated via value iteration.

	$\lvert\mathcal{H}\rvert = 1$				$\lvert\mathcal{H}\rvert = 2$			
Metric	DR	BADR	BSM	BT	DR	BADR	BSM	BT
Mean	0.204	0.017	0.017	0.004	0.403	0.036	0.029	0.007
75th Percentile	0	0	0	0	0.001	0	0	0
95th Percentile	1.11	0.006	0.006	0.002	2.73	0.134	0.096	0.011
99th Percentile	5.23	0.545	0.545	0.108	7.05	1.00	0.839	0.196

Recall that BSM is the best performing among all $2^{|\mathcal{H}|}$ single-mode policies, while BADR is the best performing among a subset of these of size $|\mathcal{H}| + 1$. The two heuristics are identical for $|\mathcal{H}| = 1$, but by definition BSM is stronger for $|\mathcal{H}| \geq 2$. Yet, for $n = |\mathcal{H}| = 2$, the difference in performance is very small, as in Table 3.5.2. The BT heuristic is clearly the best performing heuristic, which, even when $|\mathcal{H}| = 2$, achieves within 0.02% of optimality in 95% of search problems, and within 0.2% of optimality in 99% of problems.

Also seen in Table 3.5.2, for either value of $|\mathcal{H}|$, one or more of our heuristics achieve optimality for more than 75% of the search problems in our numerical study. For these search problems, it is impossible for the MC method of Section 3.5.2 to *beat* the best of these heuristics. In fact, in many search problems, although the MC method gets very close to optimality, at least one of our four heuristics gets even closer. By combining the MC method and our four heuristics, we obtain our best estimate of the optimal value. The right-hand side of Table 3.5.1 shows the performance of this *ensemble* method to estimate the optimal value compared with value iteration for search problems with 2 boxes. Since value iteration is computationally infeasible for search problems with more than 2 boxes, we will use this ensemble method as our benchmark to evaluate our heuristics for $n > 2$.

3.5.4 Performance of Heuristics with $n = 4$ and $n = 8$

We next present numerical results for search problems with $n = 4$ and $n = 8$ boxes. Since value iteration is computationally infeasible, we evaluate our heuristics against estimated optimal values from the ensemble method discussed at the end of Section 3.5.3. For $n = 4$, the ensemble estimate is based on 6×10^5 runs. Since the average improvement in the ensemble estimate is only 0.00015% when the number of runs increases from 3×10^5 to 6×10^5, conducting additional runs beyond 6×10^5 is not likely to improve the accuracy much further. For $n = 8$, the ensemble estimate is based on 10^6 runs for the same reason.

For each n, we first use (3.5.1) and rejection sampling to generate search times and detection probabilities for n boxes with $|\mathcal{H}| = n/2$. To choose prior distributions on the object's location, we consider five different scenarios. Details can be found in Table 3.5.3, in which the scenarios are ordered roughly according to the entropy of the prior. For each prior, the expected search time for each heuristic is evaluated using the methods of Section 3.5.1 and expressed as a percentage over the optimal estimate obtained using the ensemble method. To account for increasing variety in the search problems as n grows, we repeat the preceding $n \times 1000$ times to collect data. The whole process is then repeated for $|\mathcal{H}| = n$.

Table 3.5.4 displays the results for $n = 4$. For each $|\mathcal{H}|$, the best performing heuristic across all five scenarios is BT, which, even with $|\mathcal{H}| = 4$ and in its worst performing scenario, is within 0.3% of optimality in 99% of search problems. The performance of the second best heuristic BSM relative to that of BT depends on the entropy of the prior. Imagine a search problem starting with p close to 1, so the prior has a small entropy. Typically, an optimal search sequence will begin with a few searches of box 1 before moving on to search any other box. After this initial transient period, the posterior probability distribution on the object's location will stay in some envelope that centers at the probability distribution that makes each box

Table 3.5.3: Five scenarios with different prior probability distributions on the object's location.

Scenario	Prior for $n = 4$	Prior for $n = 8$
Uniform	$(0.25, 0.25, 0.25, 0.25)$	$(0.125, \ldots, 0.125)$
Two Dominate	$(0.36, 0.36, 0.14, 0.14)$	$(0.23, 0.23, 0.09, \ldots, 0.09)$
Evenly Spaced	$(0.4, 0.3, 0.2, 0.1)$	$(0.195, 0.175, \ldots, 0.055)$
One Dominates Weakly	$(0.58, 0.14, 0.14, 0.14)$	$(0.37, 0.09, \ldots, 0.09)$
One Dominates Strongly	$(0.7, 0.1, 0.1, 0.1)$	$(0.51, 0.07, \ldots, 0.07)$

equally attractive to search next. Generally, the larger the entropy of the prior, the more likely it is that the posterior stays in this envelope from the very beginning, so the more likely it is that BT and BSM produce similar or even identical search sequences. Consequently, the difference in performance between BT and BSM decreases as the entropy of the prior increases. By similar reasoning, BSM and BT are closer to optimal when the prior has a larger entropy, as a smaller proportion of the state space is explored by the posterior.

Recall that DR designates fast for all type-H boxes. Its performance is much inferior to that of the other three heuristics, becoming worse as the entropy of the prior increases. The latter effect occurs because, as discussed in Section 3.4.2, the importance of the immediate benefit, which is larger for the fast mode for all type-H boxes, increases with the size of the hiding probability, and the lower-entropy priors in Table 3.5.3 give one box a high chance of containing the object.

We next increase the number of boxes to $n = 8$. As seen in Table 3.5.5, the BT heuristic again performs the best across all scenarios. Patterns on relative performance between the heuristics in Table 3.5.4 for $n = 4$ are also observed in Table 3.5.5 for $n = 8$.

By comparing the results in Tables 3.5.4 and 3.5.5, we also see that the performance of DR degrades as n increases. To understand this phenomenon intuitively, first note that as n increases, in general, the entropy of the prior increases, so the future benefit of a search—how a failed search gains information about the object's location—becomes more important. Consequently, the appeal of the slow mode—which was found to almost exclusively have a larger future benefit in boxes where both search modes were used in an optimal policy—increases as n increases. Table 3.5.6 provides empirical evidence of this intuitive argument. Recall that DR corresponds to taking $\theta = 0$ in ADR. A similar phenomenon can also be seen in Figure 3.4.1 in Section 3.4.1, which shows that the best choice of θ for ADR increases with n.

Finally, both BSM and BT see their performance slightly improve as n increases, particularly the former. This phenomenon is again linked to the size of the initial transient period before the posterior settles into some envelope. In general, the entropy of the prior increases as n increases, so the length of this transient period, and the probability that the object is found on it, will decrease, meaning it is more likely that BSM or BT will produce a search sequence that is near optimal.

As n increases, since there are more boxes to search, the expected search time under any search policy takes longer to calculate. For any search problem, DR evaluates just one search policy. However, the number of policies evaluated by the other three heuristics depends on $|\mathcal{H}|$; BADR evaluates $|\mathcal{H}| + 1$, BSM evaluates $2^{|\mathcal{H}|}$ and BT evaluates up to $2^{|\mathcal{H}|}$. Therefore, for both $n = 4$ and $n = 8$, we report the average runtime to compute the expected search time of each heuristic in Table 3.5.7.

As expected, DR is the fastest, and whilst all four heuristics are quick to calculate for our range of problems, the runtime of BSM grows rapidly with $|\mathcal{H}|$ and may soon become intractable as n and $|\mathcal{H}|$ increase further.

As $|\mathcal{H}|$ increases, Table 3.5.7 shows that BADR quickly becomes computationally more efficient relative to BSM, while Tables 3.5.4 and 3.5.5 show its relative performance degrades only slightly and is still close to optimum. In particular, for $n = |\mathcal{H}| = 8$, BADR is more than 20 times faster than BSM, while, in their worst performing scenario, BADR is on average about 0.1% above optimality compared to 0.02% for BSM.

Note that BT is on average quicker to calculate than BADR, despite the worst case scenario of BT calculating $2^{|\mathcal{H}|}$ policies compared to the $|\mathcal{H}|+1$ of BADR. Recall that, if the future benefit of a box in $\mathcal{H}$ is larger for its fast mode, BT heuristically designates the fast mode to that box, and therefore only needs to try one mode sequence for it. Hence, for BT to require the evaluation of $2^{|\mathcal{H}|}$ policies, the future benefit must be larger for the slow mode for every box in $\mathcal{H}$; such boxes are reasonably rare (see Proposition 3.5.1 and Section 3.5.5). Therefore, while it is *possible* to design a search problem where BT must evaluate $2^{|\mathcal{H}|}$ policies, such problems are rare in practice, and BT will more commonly evaluate 2^x policies where x is an integer less than half the size of $|\mathcal{H}|$.

However, the growth in the number of policies to evaluate is still exponential for BT and polynomial for BADR; hence, as $|\mathcal{H}|$ increases further, we expect BADR to become much quicker than BT. Therefore, we believe BADR is the heuristic policy to rely on for both speed and reasonable performance in a search problem with a very large $|\mathcal{H}|$.

3.5.5 Sensitivity Analysis of Heuristics

This section extends the numerical experiments to investigate how the characteristics of a type-H box affect the performance of our heuristics. Consider a type-H box with the usual parameters q_f, q_s, t_f and t_s, and write $\theta_H \in (0, 1)$ for its relative resemblance to a type-F box compared to a type-S box, as defined in (3.4.1). Recall δ_s and δ_f from (3.4.6), which measure the distance of a type-H box from being type-S and type-F, respectively. It is straightforward to show that $\delta_s < \delta_f$ if and only if $\theta_H < 0.5$. In addition, $\theta_H = 0$ coincides with $\delta_s = 0$, and $\theta_H = 1$ coincides with $\delta_f = 0$. Finally, recall β from (3.4.4), which measures the advantage of the slow mode over the fast mode in future benefit. The following proposition connects θ_H, δ_s, δ_f and β. To maintain focus on the numerical results, the proof may be found in Appendix B.1

Proposition 3.5.1 If $\delta_s \geq \delta_f$, or equivalently, $\theta_H \geq 0.5$, then $\beta < 0$.

While Proposition 3.4.3 shows that $\beta < 0$ for any type-F box, Proposition 3.5.1 identifies some type-H boxes for which $\beta < 0$.

Proposition 3.5.1 also tells us that if a type-H box is closer to being type-F than type-S, then both the immediate and future benefit are larger for the fast mode. It further provides an intuition for the observation in Section 3.4.1 that, throughout all the numerical experiments, BADR could always be recovered by taking some $\theta \leq 0.5$ within ADR. If, for some search problem, BADR was only attainable with $\theta > 0.5$, there would be a box in $\mathcal{H}$ designated slow by BADR for which $\theta_H > 0.5$, so with both the immediate and future benefit larger for the fast mode. As discussed in Section 3.5.2, problems where it is optimal in *any* subset of the state space to use the slow mode of such a type-H box are very rare.

To make inference on the effects of type-H boxes with $\theta_H < 0.5$ on the performance of our heuristics, we focus our analysis on search problems with $n = 2$ boxes and one type-H box. We generate 8,000 such search problems using (3.5.1) and rejection sampling. For each we study $p = 0.5$, 0.7, and 0.9, where p is the object's hiding probability for the type-H box. Table 3.5.8 presents the performance of our heuristics as average percentages over optimum, sorted into bins based on the values of p and θ_H. The table reports results only for $\theta_H \leq 0.24$, because for $\theta_H > 0.24$, the difference between any heuristic performance and optimal performance is negligible. Because BADR and BSM are equivalent when $|\mathcal{H}| = 1$, we do not report them separately.

Recall that DR designates the fast mode for the type-H box, and θ_H measures the type-H box's relative resemblance to a type-F box compared to a type-S box. Therefore, it is intuitive that DR's performance improves monotonically as θ_H increases, as seen in Table 3.5.8. The other two heuristics BT and BSM, however, share a different behaviour. Also seen in Table 3.5.8, both heuristics are near optimal when θ_H is close to 0, or when θ_H exceeds 0.2, but their performance degrades as θ_H falls in the range 0.04–0.16. This behaviour is explained by the following.

When θ_H is very small, the type-H box is very close to being type-S, so the single-mode policy Π_s that designates the slow mode for the type-H box will have close-to-optimal performance. As BSM chooses among all single-mode policies, BSM will perform at least as well as Π_s. Recall that BT can choose a policy that, for the type-H box, designates slow for $p \leq \widehat{p}$ from (3.4.5). When θ_H is small, $\widehat{p}$ is close to 1, so BT can choose a policy that is very close to Π_s, so will also perform close to optimally. Finally, recall that BSM and BT can also choose DR, whose performance improves as θ_H increases. For most problems with $\theta_H > 0.2$, DR is optimal, so all three heuristics coincide with optimal performance.

Table 3.5.4: Performance of heuristics for search problems with $n = 4$ boxes in five scenarios, reported as percentage above the estimated optimum from the ensemble method.

		$\vert\mathcal{H}\vert = 2$				$\vert\mathcal{H}\vert = 4$			
Scenario	Metric	DR	BADR	BSM	BT	DR	BADR	BSM	BT
Uniform	Mean	0.738	0.010	0.004	0.003	1.42	0.040	0.007	0.006
	75th Percentile	0.510	0	0	0	2.11	0	0	0
	95th Percentile	4.33	0.018	0.013	0.009	6.59	0.176	0.042	0.034
	99th Percentile	7.98	0.258	0.105	0.087	10.2	1.04	0.168	0.153
Two Dominate	Mean	0.700	0.012	0.007	0.004	1.33	0.041	0.012	0.007
	75th Percentile	0.431	0	0	0	1.83	0	0	0
	95th Percentile	4.10	0.037	0.025	0.011	6.43	0.240	0.067	0.037
	99th Percentile	8.23	0.325	0.196	0.113	10.3	0.868	0.299	0.201
Evenly Spaced	Mean	0.700	0.013	0.008	0.005	1.31	0.039	0.013	0.008
	75th Percentile	0.452	0	0	0	1.72	0	0	0
	95th Percentile	4.20	0.035	0.022	0.010	6.08	0.219	0.062	0.037
	99th Percentile	8.42	0.389	0.241	0.144	10.3	0.951	0.332	0.216
One Dominates Weakly	Mean	0.637	0.019	0.013	0.005	1.25	0.050	0.024	0.009
	75th Percentile	0.388	0	0	0	1.73	0	0	0
	95th Percentile	3.73	0.070	0.052	0.010	5.88	0.325	0.131	0.042
	99th Percentile	7.81	0.546	0.362	0.140	10.4	1.02	0.600	0.237
One Dominates Strongly	Mean	0.569	0.028	0.023	0.005	1.09	0.066	0.043	0.010
	75th Percentile	0.320	0	0	0	1.42	0	0	0
	95th Percentile	3.31	0.097	0.064	0.015	5.01	0.474	0.271	0.043
	99th Percentile	7.38	0.756	0.638	0.141	10.1	1.28	0.995	0.265

Table 3.5.5: Performance of heuristics for search problems with $n = 8$ boxes in five scenarios, reported as percentage above the estimated optimum from the ensemble method.

Scenario	Metric	$\|\mathcal{H}\| = 4$				$\|\mathcal{H}\| = 8$			
		DR	BADR	BSM	BT	DR	BADR	BSM	BT
Uniform	Mean	1.14	0.027	0.003	0.003	2.24	0.104	0.004	0.004
	75th Percentile	1.77	0	0	0	3.45	0.034	0	0
	95th Percentile	3.43	0.020	0	0	5.43	0.376	0.008	0.007
	99th Percentile	4.66	0.140	0.014	0.013	6.70	0.648	0.028	0.026
Two Dominate	Mean	1.11	0.026	0.004	0.003	2.20	0.098	0.006	0.005
	75th Percentile	1.66	0	0	0	3.31	0.034	0	0
	95th Percentile	3.34	0.020	0.001	0.001	5.36	0.352	0.011	0.009
	99th Percentile	4.46	0.136	0.016	0.015	6.86	0.609	0.034	0.029
Evenly Spaced	Mean	1.12	0.025	0.003	0.003	2.21	0.099	0.005	0.005
	75th Percentile	1.61	0	0	0	3.31	0.030	0	0
	95th Percentile	3.44	0.019	0.001	0.001	5.47	0.343	0.010	0.008
	99th Percentile	4.79	0.133	0.016	0.015	6.95	0.625	0.032	0.029
One Dominates Weakly	Mean	1.09	0.027	0.005	0.003	2.15	0.100	0.008	0.005
	75th Percentile	1.6	0	0	0	3.20	0.039	0	0
	95th Percentile	3.23	0.024	0.001	0.001	5.19	0.348	0.012	0.008
	99th Percentile	4.38	0.153	0.020	0.014	6.69	0.619	0.040	0.027
One Dominates Strongly	Mean	1.05	0.031	0.011	0.004	2.06	0.103	0.017	0.006
	75th Percentile	1.46	0	0	0	2.98	0.051	0	0
	95th Percentile	2.98	0.035	0.004	0.002	4.98	0.368	0.023	0.009
	99th Percentile	4.28	0.200	0.031	0.018	6.67	0.619	0.080	0.031

Table 3.5.6: The percentage of type-H boxes to which BSM designates fast in each numerical study.

n	$\lvert\mathcal{H}\rvert = n/2$	$\lvert\mathcal{H}\rvert = n$
2	91.1%	91.4%
4	83.7%	84.6%
8	79.9%	80.3%

Table 3.5.7: The average time (milliseconds) taken to calculate the expected search time under each heuristic in a single search problem for $n = 4$, 8 boxes, and $|\mathcal{H}| = n/2$, n. Results are shown for the Evenly Spaced prior from Table 3.5.3.

n	$\lvert\mathcal{H}\rvert$	DR	BADR	BSM	BT
4	2	0.003	0.008	0.010	0.006
4	4	0.003	0.013	0.038	0.008
8	4	0.009	0.041	0.112	0.023
8	8	0.011	0.074	1.63	0.056

Table 3.5.8: Performance of heuristics for search problems with $n = 2$ and $|\mathcal{H}| = 1$ by value of θ_H, reported as average percentage above the optimum calculated via value iteration.

		θ_H					
Heuristic	p	$(0, 0.04]$	$(0.04, 0.08]$	$(0.08, 0.12]$	$(0.12, 0.16]$	$(0.16, 0.2]$	$(0.2, 0.24]$
DR	0.5	2.76	1.11	0.437	0.151	0.037	0.012
DR	0.7	3.09	1.13	0.383	0.126	0.031	0.010
DR	0.9	2.06	0.594	0.204	0.070	0.019	0.006
BSM	0.5	0.008	0.019	0.019	0.018	0.011	0.011
BSM	0.7	0.019	0.085	0.089	0.091	0.030	0.010
BSM	0.9	0.130	0.393	0.198	0.070	0.019	0.006
BT	0.5	0.007	0.014	0.010	0.008	0.005	0.001
BT	0.7	0.012	0.032	0.023	0.012	0.003	0.001
BT	0.9	0.031	0.053	0.015	0.007	0.002	0.0002

Chapter 4

Conclusions and Further Work

In this chapter, Section 4.1 reviews Part I of this thesis, and Section 4.2 suggests directions for further work.

4.1 Review of Part I

Search theory is a widely-researched area which the literature review in Chapter 1 could only touch upon. In the classic time-independent search problem, sequential searches of discrete boxes for a stationary target are made in discrete time, with the aim to minimise the expected time until detecting the target. The problem was solved by several authors using different methods in the 1960s; the solution is wonderfully simple and intuitive, namely, search the box with the largest probability per unit time of finding the target on the next search. The solution extends to the classic time-dependent search problem (where the search parameters of a box may depend on its number of prior searches) if the probability of a detection on a search of a box decreases with the number of visits.

First noted by a comment of Kelly in Gittins (1979), the multi-armed bandit framework of Chapter 2 provides another way to solve the classic search problem, namely, using the celebrated Gittins indices of Gittins and Jones (1974). The Gittins index solution agrees with earlier solutions in the literature, but has the advantage of solving the classic time-dependent search problem in its full generality.

Chapter 3, based largely on Clarkson et al. (2020), extends the classic time-independent problem to allow a choice between two search modes (fast and slow) in each box. A fast search of a box is quicker, but a slow search is more likely to find the target, leading to a trade-off between speed and effectiveness. Despite advances in search technology increasing the relevance of multiple-mode problems, they are not well studied in the literature. Kadane (2015) studies the extension of the classic time-independent problem to an arbitrary number of search modes per box, but his results are incorrect.

The Gittins index solution to the fully-general classic time-dependent problem in Chapter 2 reduces the two-mode problem to determining an optimal sequence of modes for each box; however, the optimal choice of mode for a box can be very complicated, sometimes depending on the other boxes available to search and the searcher's current beliefs about the target's location. The main results of Chapter 3 are simple sufficient conditions on a box such that its optimal mode sequence contains just one dominating search mode. Under Theorem 3.2.5, the dominating search mode is slow, and under Theorem 3.2.8, the dominating mode is fast.

If all boxes satisfy either Theorem 3.2.5 or Theorem 3.2.8, then the two-mode problem reduces to a single-mode problem easily solved using Gittins indices. Otherwise, for the boxes that satisfy neither theorem, heuristic policies are required. Our main heuristic policy, the threshold policy BT, considers two sources of benefit from searching a box.

The first, the *immediate benefit*, is the probability per unit time of finding the target on a search of a box, well known for its key role in the Gittins index which solves the single-mode problem. Kadane (2015) suggests that the optimal multiple-mode solution is a simple, direct extension of the single-mode solution, but we find the presence of another type of searching benefit when there are multiple modes per box, namely the *future benefit*. In a single-box problem with multiple modes, Kadane (2015) finds that always using the mode with the largest future benefit maximises the probability of finding the target before a deadline. In a problem with multiple boxes and, therefore, uncertainty about the target's location, we find the future benefit takes a new role; the larger the future benefit of a search mode, the more information gained about the target's location after an unsuccessful search using that mode. Such information is pivotal as it allows the searcher to make better box choices later in the search process. Our BT heuristic involves a novel combination of the immediate and future benefit, and consistently delivers near-optimal performance in an extensive numerical study.

In work not included in Clarkson et al. (2020), Chapter 3 also uncovers a third type of benefit which explains why the slow mode can still, in very rare circumstances, be optimal when the fast mode has the larger immediate and future benefit. This third source of benefit, arising from searches being made in discrete time, involves avoidance of the situation where the searcher has no strong preference over which box to search next. The effect in the two-mode problem is negligible; however, in Part II of this thesis, where the target is an intelligent hider actively avoiding detection, the third source of benefit plays a pivotal role in the hider's strategy selection.

4.2 Further Work

The remainder of this chapter will discuss avenues for further work. Since the search literature is so vast but contains little work on multiple-mode problems, there are plenty of existing search models to which a second search mode would be a novel addition.

Aside from the single-box problem of Kadane (2015), there is no work on multiple-mode problems with objective to maximise the probability of finding the target before a deadline. Such an objective is beyond the reach of Gittins index theory, which may complicate the analysis. Since there is no benefit to finding the target in advance of the deadline, the superior information gain of the future benefit may take greater importance than the immediate benefit near the start of the search. However, as the deadline draws closer, the significance of the immediate benefit may grow. Unless the final search of a box may be curtailed as in Kadane (2015), modes with search times finishing as close to the deadline as possible may become desirable even if their benefits are inferior to those of other modes.

The multiple-target model of Smith and Kimeldorf (1975) could allow the searcher two modes per box. When there are many targets left to find, information about the targets' locations may be more valuable, and hence the searcher may prefer a mode with a superior future benefit. When there are less targets left to find, the immediate benefit may take greater importance.

A novel set of models is motivated by the following. Because of either budget, time or demand constraints, search equipment may be of limited use. Therefore, while multiple search modes may exist, all may not be available throughout the duration of the search process. This motivates several formulations where the searcher may choose between multiple modes but only for a limited amount of time.

Firstly, suppose the searcher has a device to aid them; for example, help from an additional searcher, or use of some equipment. The device can be used in any box and boosts detection capabilities, yet it can only be used for a limited amount of time, reflecting real-life either in the scarcity of resources needed for other jobs, or in budget constraints. On which boxes should the device be used and when? Until the device runs out, each box has two modes: search with the device or without it.

Secondly, suppose the searcher needs to hire search equipment. Several pieces are available, yet only one can be hired, accounting for real-life budget constraints. In such a problem, for each box, the searcher chooses a mode for the entire search process. Several variations are of interest. If the price of equipment varies, there could be an initial cost to the searcher to choose a more effective search mode. If the same equipment is used for several boxes, mode choices may be dependent across boxes; e.g., the mode chosen for box 1 may limit the available modes for box 2. Theoretical results in Chapter 3 may immediately eliminate some available modes in the same box yet, in the choice-dependent case, comparison of modes across boxes presents a greater challenge. The single-mode heuristic policies of Chapter 3 may shed light on such problems.

Two additional ideas for further work will be discussed in more detail in the following two subsections.

4.2.1 Multiple Search Modes

A natural extension to the two-mode search problem is to allow three or more search modes per box. As for the two-mode problem, if, for each box, the modes used to search the box are predetermined, we have a classic time-dependent search problem solved by Gittins indices as detailed in Chapter 2. Therefore, the multiple-mode problem is reduced to finding an optimal sequence of modes for each box.

Presented below, a preliminary analysis is made in Clarkson et al. (2020) which extends Theorems 3.2.5 and 3.2.8 to boxes with three or more search modes to find conditions for one mode to dominate the others.

To start, consider a box with only two modes (q, t) and (q', t'). For fixed (q', t'), we consider the values of q and t for which we know that (q', t') is dominant.

First, suppose $t \leq t'$. Then (q', t') is dominant by Theorem 3.2.5 if and only if

$$q \leq \min\left(q', \frac{q't}{t'}\right) = \frac{q't}{t'}. \tag{4.2.1}$$

Now suppose $t > t'$. Then (q', t') is dominant if $q \leq q'$ by virtue of a larger detection probability and smaller search time. Otherwise, (q', t') is dominant by Theorem 3.2.8 if and only if

$$q \leq \frac{tq'(1-q)}{t'}, \quad \text{or equivalently,} \quad q \leq \frac{t}{t + t'/q'}.$$

Therefore, when $t > t'$, mode (q', t') is dominant by either Theorem 3.2.8 or by having a larger detection probability and smaller search time if and only if

$$q \leq \max\left(q', \frac{t}{t + t'/q'}\right). \tag{4.2.2}$$

The maximum in (4.2.2) is attained (uniquely) by the right-hand term if and only if $t > t'/(1 - q')$.

Based on the above, for any search mode with detection probability q' and search time t', we define a function $g(t)$ for $t > 0$ as follows:

$$g(t) \equiv \begin{cases} (q'/t')\, t, & \text{if } t \leq t', \\ q', & \text{if } t' < t \leq t'/(1-q'), \\ t/(t + t'/q'), & \text{if } t > t'/(1-q'). \end{cases} \tag{4.2.3}$$

In addition, define a set of search modes based on (q', t') as follows:

$$D(q', t') \equiv \{(q, t) : t > 0,\ q \leq g(t)\}.$$

By (4.2.1) and (4.2.2), (q', t') dominates (q, t) by either Theorem 3.2.5, Theorem 3.2.8, or by a larger detection probability and smaller search time if and only if $(q, t) \in D(q', t')$. Figure 4.2.1 illustrates $g(t)$ and $D(q', t')$ for $(q', t') = (0.4, 1)$.

Now consider a search problem with n boxes where, for $i = 1, \dots, n$, box i has some $K_i \in \mathbb{Z}^+$ search modes, namely $(q_{i,k}, t_{i,k})$ for $k = 1, \dots, K_i$. For some box i, suppose there exists $j \in \{1, \dots, K_i\}$ such that $(q_{i,k}, t_{i,k}) \in D(q_{i,j}, t_{i,j})$ for $k = 1, \dots, K_i$. Then, since for any $k = 1, \dots, K_i$ the mode $(q_{i,j}, t_{i,j})$ dominates $(q_{i,k}, t_{i,k})$ in a box with just those two search modes, we say mode $(q_{i,j}, t_{i,j})$ is *dominating* for box i. Based on this definition, each box can have at most one dominating mode, which the following shows is always an optimal mode choice.

Theorem 4.2.1 In the above multiple-mode search problem, if some box has a dominating mode, then there exists an optimal search sequence in which that box is always searched using its dominating mode.

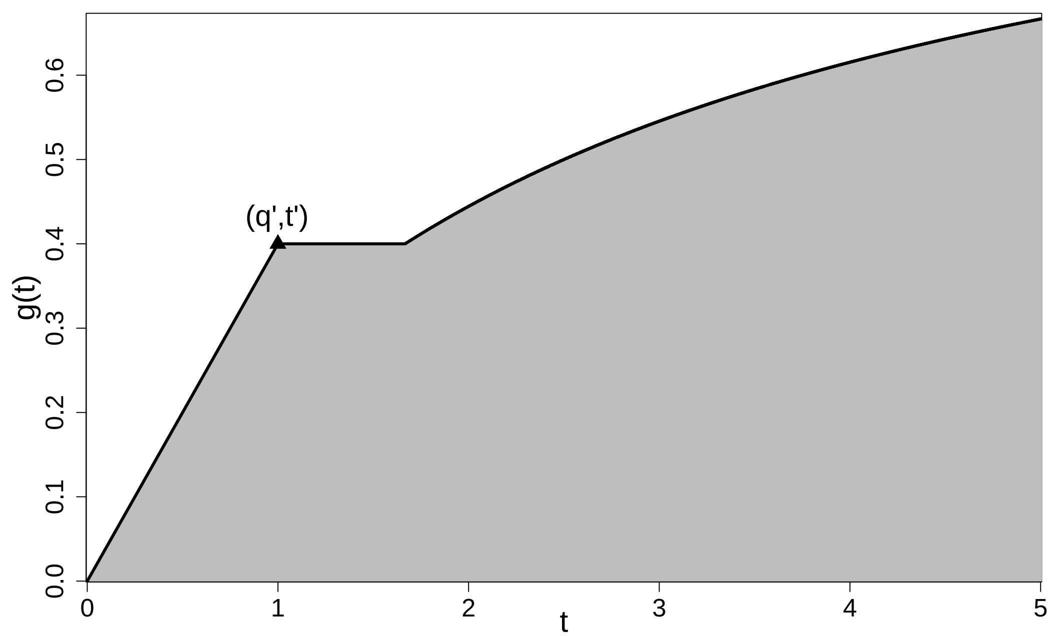

Figure 4.2.1: The black line shows the different parts of the function $g(t)$ for $(q', t') = (0.4, 1)$. The set $D(0.4, 1)$ is shown by the line alongside the shaded gray areas.

The proof of Theorem 4.2.1 involves similar arguments used to prove Theorems 3.2.5 and 3.2.8, and is deferred to Appendix B.2.

Theorem 4.2.1 extends Theorems 3.2.5 and 3.2.8 from the standpoint of a dominating mode and identifies boxes for which only one search mode is needed in an optimal policy. Based on Theorem 4.2.1, it is reasonable to conjecture a more general result—extending Theorems 3.2.5 and 3.2.8 from the standpoint of what we call a *dominated* search mode. We say that mode $(q_{i,k}, t_{i,k})$ is dominated in box i if there exists $j \in \{1, \ldots, K_i\}$, with $j \neq k$, such that $(q_{i,k}, t_{i,k}) \in D(q_{i,j}, t_{i,j})$. In a box with just modes $(q_{i,k}, t_{i,k})$ and $(q_{i,j}, t_{i,j})$ available, we know that the latter dominates. However, if the latter were replaced by some $(q_{i,l}, t_{i,l})$ for some $l \in \{1, \ldots, K_i\} \setminus \{k, j\}$, then neither $(q_{i,k}, t_{i,k})$ nor $(q_{i,l}, t_{i,l})$ necessarily dominates.

We conjecture that, if some box has a dominated mode, then there exists an optimal search sequence in which that box is never searched using the dominated mode. If this conjecture is true, then it can substantially simplify a multiple-mode search problem by enabling the removal of all dominated modes in each box. Whether this conjecture is true, however, remains an open question, and will be left as future research. Several attempts to find a counter example have failed; however, preliminary investigation suggests that the techniques in the proofs of Theorems 3.2.5 and 3.2.8 cannot easily be adjusted to prove the conjecture, so new methods may be required.

The single-mode heuristic policies DR and BSM introduced in Section 3.4.1 are easily extended to this multiple-mode setting as follows. Without loss of generality, label the search modes of box i such that $q_{i,1} \leq \cdots \leq q_{i,K_i}$, for $i = 1, \ldots, n$. For box i, write m_i for the mode with the largest detection probability per unit time q/t. The heuristic DR simply designates mode m_i for box i, $i = 1, \ldots, n$. There are a total of $\prod_{i=1}^{n} K_i$ different single-mode policies, and BSM is the one with the smallest expected search time.

We can also extend the ideas in Section 3.4.3 to bound the suboptimality of DR and BSM in the multiple-mode setting. For any box i and any mode k, define

$$\delta_{i,k} \equiv \max\left(\max_{j=1,\ldots k} \left\{ \frac{q_{i,j}/t_{i,j}}{q_{i,k}/t_{i,k}} \right\}, \max_{j=k+1,\ldots K_i} \left\{ \frac{q_{i,j}/t_{i,j}}{(1-q_{i,j})q_{i,k}/t_{i,k}} \right\} \right) - 1. \tag{4.2.4}$$

If mode k for box i satisfies the condition of Theorem 4.2.1, so is dominating for box i, then the left-hand inner maximisation term is equal to 1, and the right-hand inner term is no greater than 1, so we have $\delta_{i,k} = 0$. Otherwise, for any mode j for which $(q_{i,j}, t_{i,j}) \notin D(q_{i,k}, t_{i,k})$, the corresponding term in (4.2.4) is greater than 1, and can be interpreted as a measure of the distance of mode $(q_{i,j}, t_{i,j})$ from the set $D(q_{i,k}, t_{i,k})$. Thus, we interpret $\delta_{i,k}$ as the distance of mode k from satisfying the conditions of Theorem 4.2.1 and hence dominating for box i.

We next present a version of Proposition 3.4.1 for the multiple-mode search problem, which can be used to bound the suboptimality of DR and BSM in terms of $\delta_{i,k}$ in (4.2.4).

Proposition 4.2.2 Write V^* for the optimal expected search time. Write Π for the single-mode policy that designates mode k_i for box i, and V_Π for its corresponding expected search time. We can bound the suboptimality of Π by

$$\frac{V_\Pi - V^*}{V^*} \leq \max_{i=1,\ldots,n} \delta_{i,k_i}.$$

Proof. For $i = 1, \ldots, n$, let

$$\widehat{t}_{i,k_i} \equiv \frac{t_{i,k_i}}{\delta_{i,k_i} + 1}.$$

In the modified search problem where t_{i,k_i} is replaced by $\widehat{t}_{i,k_i}$ for $i = 1, \ldots, n$, Π is the optimal policy. The same argument used in the proof of Proposition 3.4.1 can be applied to the modified and original search problems to show

$$\frac{V_\Pi - V^*}{V^*} \leq \max_{i=1,\ldots,n} \left[\frac{t_{i,k_i} - \widehat{t}_{i,k_i}}{\widehat{t}_{i,k_i}} \right] = \max_{i=1,\ldots,n} \delta_{i,k_i},$$

which completes the proof. ■

Corollary 4.2.3 Write V^* for the optimal expected search time, and V_{DR} and V_{BSM} for the expected search time for the heuristics DR and BSM, respectively. We can bound the suboptimality of DR and BSM as follows.

$$\frac{V_{\text{DR}} - V^*}{V^*} \leq \max_{i=1,\ldots,n} \delta_{i,m_i}. \tag{4.2.5}$$

$$\frac{V_{\text{BSM}} - V^*}{V^*} \leq \max_{i=1,\ldots,n} \min_{k=1\ldots,K_i} \delta_{i,k}. \tag{4.2.6}$$

Proof. Because DR uses mode m_i for box i, $i = 1, \ldots, n$, the bound in (4.2.5) follows immediately from Proposition 4.2.2. The bound in (4.2.6) follows from Proposition 4.2.2 because BSM compares all single-mode policies. ■

The bound for BSM in Corollary 4.2.3 is at least as strong as that for DR. For each $i = 1, \ldots, n$, if we consider some limit under which we have $\delta_{i,k_i} \downarrow 0$ for some $k_i \in \{1, \ldots, K_i\}$, then mode k_i becomes dominating for box i and BSM approaches optimality. If $\delta_{i,m_i} \downarrow 0$ for $i = 1, \ldots, n$, then both DR and BSM approach optimality. As in Corollary 3.4.2, all bounds in Corollary 4.2.3 depend only on detection probabilities and search times; they do not depend on the target's hiding probabilities.

Extending the BADR and BT heuristics to multiple search modes is not as straightforward, since they both involve a pairwise comparison of search modes. An extension of the threshold policy BT to a problem with multiple search modes could calculate a threshold for each pair of modes. The inference in Table 3.4.1 in Chapter 3 still holds; therefore, if the threshold of two modes is less than 0 (resp. greater than 1), then the faster (resp. slower) mode has the advantage in both immediate and future benefit, and an extended threshold heuristic could discard the slower (resp. faster) mode. If the threshold lies between 0 and 1, the faster mode has the greater immediate benefit and the slower mode the greater future benefit; implementing the threshold in this case may not be as straightforward, as the following demonstrates.

Suppose there are three modes for a box containing the target with probability p, with the third mode being 'medium' with detection probability $q_m \in (q_f, q_s)$ and search time $t_m \in (t_f, t_s)$. Let the three pairwise thresholds be $\widehat{p}_{s,m}$, $\widehat{p}_{m,f}$ and $\widehat{p}_{s,f}$. If any threshold lies outside $[0, 1]$, then one mode can be discarded by an extended BT heuristic since its immediate and future benefits are both less than another mode's. With only two modes remaining, an extended BT heuristic can choose modes in the same manner as in the original. Otherwise, suppose all three thresholds lie in $[0, 1]$ and $\widehat{p}_{s,f}$ is uniquely the largest threshold. For $p \in [\widehat{p}_{s,f}, 1]$, the obvious choice of mode is fast, as the greater immediate benefit of fast is more important for larger p than the greater future benefits of the other two modes. For p below $\widehat{p}_{s,f}$, before another threshold is reached, the choice of mode is not so obvious. We have $p < \widehat{p}_{s,f}$, so slow should be preferred to fast; $p > \widehat{p}_{s,m}$, so medium should be preferred to slow; and $p > \widehat{p}_{m,f}$, so fast should be preferred to medium. In other words, there is intransitivity in the threshold order and a choice of mode for BT is not clear. An alternative approach may, therefore, be required.

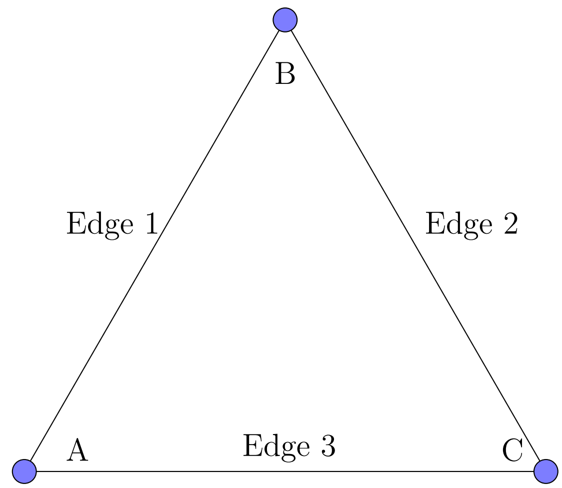

Figure 4.2.2: A simple triangular network with three nodes and edges.

4.2.2 A Two-Mode Search on a Network

Another further work idea is a two-mode search on a network structure (points (nodes) connected by lines (edges) in $\mathbb{R}^2$) rather than in discrete boxes. The target is located on each edge with a known hiding probability. The searcher traverses edges to find the target, with a slow and a fast speed available for each edge. The searcher's path must respect the geography of the network; in other words, the searcher can only next choose an edge connected to the node at which they finished their previous edge search. Such an extension is relevant if the geography of the search space prevents the searcher from moving quickly between any pair of hiding locations, for example, a target hidden on a structure of roads.

The simplest non-trivial network forms a triangle with three nodes and edges, shown in Figure 4.2.2. Unlike the standard two-mode search problem, a fast search may be desirable if the searcher wishes to access a currently-unavailable edge. For example, if the searcher is at node A, edges 1 and 3 are available for the next search, but the searcher may strongly believe the target to be on edge 2. In this instance, even if the fast modes of edges 1 and 3 are inferior to their slow modes, a fast search of one of them may still be optimal to enable a search of edge 2 as soon as possible. Therefore, it is important that any index policy over the currently available edges looks ahead several searches into the future. Such a look-ahead index policy is successfully used on network structure in Lin et al. (2013) and Lin et al. (2014), which study a patroller traversing a network detecting attacks. Unselected indices are viewed as penalties so the searcher does not delay the search of an attractive edge to wait for its appeal (index) to rise even further.

Part I Appendices

Appendix A

Construction of Example 3.2.6

This subsection constructs Example 3.2.6, a search problem in which it is optimal to make one fast search of a box with equal slow and fast detection rates. The construction relies upon several observations from the proofs of Lemmas 3.2.2 and 3.2.3, plus the study of optimal policies in the continuous-sweeping variant of the standard search problem.

Observations from Lemmas 3.2.2 and 3.2.3 Recall the proof of Lemma 3.2.2 considers two searchers; searcher 1 uses arbitrary within-box subsequence A for box 1, while searcher 2 uses continuous sweeping. Box 1 satisfies

$$\frac{q_s}{t_s} = \frac{q_f}{t_f}, \tag{A.0.1}$$

with identifying subscript 1 dropped for ease of notation. Both searchers 1 and 2 use within-box subsequence A_i for box i, $i = 2, \ldots, n$, and have optimal expected search times equal to T_A and T_W, respectively.

Searcher 1 follows an optimal policy by choosing between boxes using Gittins indices. The proof of Lemma 3.2.2 uses π_A to denote an arbitrary optimal policy for searcher 1, and, for any π_A, describes a searcher 2 policy that 'mimics' π_A. It writes $T_W(\pi_A)$ for the expected search time under this mimicking policy, with $T_W(\pi_A) \geq T_W$ for any π_A. In general, $T_W(\pi_A)$ depends upon the choice of π_A. For example, as the sweeping searcher 2 has an advantage over searcher 1 when visiting box 1, optimal policies for searcher 1 that always split ties in favour of box 1 will have a lower expected search time when mimicked by searcher 2 than those that do not.

Let A_f denote a within-box subsequence for box 1 that contains precisely one fast mode. The proof of Lemma 3.2.2 shows that, for any π_{A_f}, we have

$$T_{A_f} - T_W(\pi_{A_f}) = p_1 t_s/2. \tag{A.0.2}$$

So, when searcher 1 uses A_f as the within-box subsequence for box 1, mimicking any searcher 1 optimal policy leads to the same expected search time for searcher 2.

Where S is the within-box subsequence for box 1 containing only the slow mode, Lemma 3.2.3 shows that $T_S - T_W = p_1 t_s/2$. Combined with (A.0.2), we have $T_{A_f} = T_S$ if and only if $T_W(\pi_{A_f}) = T_W$ for any π_{A_f}. By Theorem 3.2.5, T_S is the optimal expected search time for searcher 1. Therefore, it is optimal to make a fast search of a box 1 satisfying (A.0.1) if and only if, in the continuous-sweeping variant, it is optimal for searcher 2 to mimic any searcher 1 optimal policy with $A = A_f$. Hence, we next study optimal policies for searcher 2.

Optimal Policies for the Continuous-Sweeping Searcher The proof of Lemma 3.2.2 also shows that $T_S - T_W(\pi_S) = p_1 t_s/2$ for any π_S. Combined with the result of Lemma 3.2.3, this shows that $T_W = T_W(\pi_S)$ for any π_S, providing a class of optimal policies for searcher 2, namely, index policies where the index of box 1 is given by $p_1' q_s/t_s = p_1' q_f/t_f$ (where p_1' is the current hiding probability for box 1) and the indices of boxes $2, \ldots, n$ are the usual Gittins indices as in (3.1.1). Whenever the index of box 1 is maximal, an optimal action is to make a full sweep of $[0, t_s)$ and update the index of box 1 by multiplying by $(1 - q_s)$. Whenever the index of another box j is maximal, an optimal action is to search box j for $j = 2, \ldots, n$ (and update its index as usual). For any optimal policy in this class, whenever searcher 2 visits box 1, they make a full sweep of the interval $[0, t_s)$, so only ever leave box 1 at position $x = 0$.

Inspection of the proof of Lemma 3.2.3 shows that there are optimal policies for searcher 2 outside of this class. If the index of box 1 is uniquely maximal, then the unique optimal action is to sweep the whole of $[0, t_s)$ before moving on. Suppose instead that the index of box 1 and the index of some other box j are both maximal, and the next mode in the within-box subsequence for box j is a. Then, for the next $t_s + t_{j,a}$ time units, for each $x \in [0, t_s)$ there is a corresponding optimal action given by the following:

1. Sweep the interval $[0, x)$.
2. Search box j using mode a.
3. Sweep the interval $[x, t_s)$.
4. Update the index of box 1 (resp. j) by multiplying by $(1 - q_s)$ (resp. $(1 - q_{j,a})$).

Searcher 2 taking any other action for the next $t_s + t_{j,a}$ time units is suboptimal. Note that if we choose $x = 0$ or $x = t_s$, we fall into the class of optimal searcher 2 policies described in the previous paragraph. Clearly the above situation can be generalised to an i-way tie between indices including box 1, for $i = 3, \dots, n$.

Conditions under which a Mimicking Policy is Optimal Finally, we consider conditions under which a mimicking searcher 2 policy is optimal. If searcher 2 mimics any π_{A_f}, then before the fast mode of box 1 occurs in π_{A_f}, searcher 2 follows an optimal policy and always leaves $[0, t_s)$ at $x = 0$. Yet, after the fast mode of box 1 occurs in π_{A_f}, searcher 2 always leaves $[0, t_s)$ at $x = t_f$. Note that, under any searcher 2 optimal policy, it is optimal for searcher 2 to leave the interval $[0, t_s)$ at some $0 < x < t_s$ to search another box j if and only if, at the moment that the *current pass* of the interval $[0, t_s)$ began from $x = 0$, box j's index was maximal alongside box 1's. So, after the fast mode of box 1 occurs in π_{A_f}, the only manner in which searcher 2 can search another box j without making a suboptimal move is through another box j's index being tied with box 1's index.

After the fast mode of box 1 occurs π_{A_f}, searcher 1 must search each box j, $j = 2, \ldots, n$, an infinite number of times, or the expected search time under π_{A_f} for searcher 1 would be infinite. So, in mimicking π_{A_f}, searcher 2 will also make an infinite number of searches of box j, $j = 2, \ldots, n$. For each of these searches to occur via a tie with box 1's index, there must be some cyclic behaviour between box j's index and box 1's index for $j = 2, \ldots, n$. In other words, for $j = 2, \ldots, n$, if the fast mode of box j is used in π_{A_f} after the fast mode of box 1, we need $(1 - q_s)^{m_{j,f}} = (1 - q_{j,f})$ for some $m_{j,f} \in \mathbb{Z}^+$, since only then, after a further $m_{j,f}$ full sweeps of box 1 and one fast search of box j, will the indices of box 1 and box j be tied again. Similarly, if if the slow mode of box j is used in π_{A_f} after the slow mode of box 1, we need $(1 - q_s)^{m_{j,s}} = (1 - q_{j,s})$ for some $m_{j,s} \in \mathbb{Z}^+$. Further, we need $(p_1, \ldots, p_n)$ to be such that the indices of box 1 and box j tie for the first time before the fast search in π_{A_f}, for $j = 2, \ldots, n$.

Example 3.2.6 presents a search problem with $n = 2$ where these restrictive conditions hold. For simplicity, the fast mode in A_f occurs first and box 2 has just one search mode. By stochastic coupling, Example 3.2.6 proves that $T_{A_f} = T_S$.

Appendix B

Supplementary Proofs for Part I

B.1 Proof of Proposition 3.5.1

Before proving the result of Proposition 3.5.1, we need a few lemmas.

Lemma B.1.1 For any $a > 1$, we have

$$\log(a)\sqrt{a} < a - 1.$$

Proof. Consider the function $h(a) \equiv a - 1 - \log(a)\sqrt{a}$. Since $h(1) = 0$, it is sufficient to show that h is a strictly increasing function for $a > 1$. Taking the derivative of h yields

$$h'(a) = 1 - \left(\frac{\log(a) + 2}{2\sqrt{a}}\right), \tag{B.1.1}$$

so $h'(1) = 0$. For $a > 1$, we have

$$\frac{d}{da}\left(\log(a) + 2\right) = \frac{1}{a} < \frac{1}{\sqrt{a}} = \frac{d}{da}(2\sqrt{a}),$$

so the denominator of the fraction in (B.1.1) grows faster than the numerator, as a increases from $a = 1$. It follows that $h'(a) > 0$ for all $a > 1$, which completes the proof. ■

Lemma B.1.2 For any $0 < x < 1$ and $1/2 \leq \xi \leq 1$, we have

$$-\log(1 - x) < \frac{x}{(1 - x)^{\xi}}.$$

Proof. Take $a = 1/(1-x) > 1$ in Lemma B.1.1 to obtain the inequality for $\xi = 1/2$. The result follows because $(1-x)^{-1/2} \le (1-x)^{-\xi}$ for all $\xi \in [1/2, 1]$. ■

We are now ready to prove Proposition 3.5.1. Using their definitions in (3.4.1) and (3.4.6), it is straightforward to show that $\delta_s \ge \delta_f$ is equivalent to $\theta_H \ge 0.5$. We show that $\delta_s \ge \delta_f$ implies $\beta < 0$.

First, note that we have $\delta_s + 1 > 0$, $\delta_f + 1 > 0$, and $(\delta_s+1)(\delta_f+1) = (1-q_s)^{-1}$. Hence, we have $\delta_s \ge \delta_f$ if and only if $\delta_s + 1 \ge (1-q_s)^{-1/2}$. We shall prove the contrapositive of the result, namely that if $\beta \ge 0$, then $\delta_f > \delta_s$, or equivalently $\delta_s + 1 < (1-q_s)^{-1/2}$.

If $\beta \ge 0$, then we have

$$\frac{t_s}{t_f} \le \frac{\log(1-q_s)}{\log(1-q_f)}.$$

Therefore,

$$\delta_s + 1 = \frac{q_f/t_f}{q_s/t_s} \le \frac{\log(1-q_s)}{\log(1-q_f)} \cdot \frac{q_f}{q_s}.$$

To complete the proof, it remains to show the right-hand term is less than $(1-q_s)^{-1/2}$.

Write the right-hand side of the preceding as a function of q_f, namely

$$f(q_f) = \frac{\log(1-q_s)}{\log(1-q_f)} \cdot \frac{q_f}{q_s}.$$

By L'Hôpital's rule, we can compute

$$\lim_{q_f \downarrow 0} f(q_f) = -\frac{\log(1-q_s)}{q_s} < (1-q_s)^{-1/2},$$

where the inequality follows from Lemma B.1.2 with $x = q_s$ and $\xi = 1/2$. It remains to show that f is a decreasing function for $q_f \in (0, q_s)$. To do so, take the first derivative to obtain

$$f'(q_f) = \frac{\log(1-q_s) \cdot \left(\frac{q_f}{1-q_f} + \log(1-q_f)\right)}{q_s \cdot (\log(1-q_f))^2}.$$

For $q_f \in (0, q_s)$, the denominator is strictly positive, and the numerator is negative by Lemma B.1.2 with $x = q_f$ and $\xi = 1$. Consequently, $f'(q_f) < 0$ for $q_f \in (0, q_s)$, and the proof is completed. ■

B.2 Proof of Theorem 4.2.1

Suppose that mode $(q_{i,j}, t_{i,j})$ is dominating for box i. Without loss of generality, we take $i = j = 1$. Since all the discussion below concerns box 1, we will omit the subscript that identifies box 1 to simplify notation.

Define $g(\cdot)$ for the search mode (q_1, t_1) according to (4.2.3). For $k = 2, 3, \ldots, K$, define $\widehat{q}_k \equiv g(t_k) \geq q_k$, where the inequality follows because $(q_k, t_k) \in D(q_1, t_1)$. Consider a modified search problem in which the $K-1$ search modes $(\widehat{q}_2, t_2), \ldots, (\widehat{q}_K, t_K)$ are also available for box 1, so box 1 now has up to $2K-1$ search modes. We will first show that in this modified search problem, there exists an optimal within-box subsequence for box 1 consisting only of mode (q_1, t_1).

First, it is clear that there exists an optimal within-box subsequence consisting only of modes $(q_1, t_1), (\widehat{q}_2, t_2), \ldots, (\widehat{q}_K, t_K)$, because (q_k, t_k), $k = 2, \ldots, K$, has the same search time as $(\widehat{q}_k, t_k)$, but a smaller (or at best the same) detection probability. Categorise the $K-1$ search modes $(\widehat{q}_2, t_2), \ldots, (\widehat{q}_K, t_K)$ into 3 groups based on each mode's search time; we say search mode $(\widehat{q}_k, t_k)$ is in group A if $t_k \leq t_1$, in group B if $t_1 < t_k \leq t_1/(1-q_1)$, and in group C if $t_k > t_1/(1-q_1)$.

If a within-box subsequence for box 1 contains any search mode in group B, the subsequence can be improved by replacing the search mode in group B with the search mode (q_1, t_1)—which has the same detection probability but a smaller search time. Therefore, there exists an optimal within-box subsequence consisting only of mode (q_1, t_1) and modes in groups A and C.

Next, we show that any search mode in group C is not needed in an optimal within-box subsequence for box 1. The argument is similar to that in Theorem 3.2.8, as it first assumes that the within-box subsequence for box 1 contains a finite number of search modes in group C. The crucial observation is that the last occurrence of a search mode in group C is followed by a subsequence consisting only of (q_1, t_1) and modes in group A—each of which has the same ratio between its detection probability and search time, namely q_1/t_1. This observation allows the interchanging argument used in the proof of Theorem 3.2.8 to go through in a similar fashion. Consequently, we can show that there exists an optimal within-box subsequence consisting only of (q_1, t_1) and modes in group A.

Finally, we adapt the proof of Theorem 3.2.5 to show that even search modes in group A are not needed in an optimal within-box subsequence for box 1. Consider an arbitrary subsequence that consists of only search mode (q_1, t_1) and modes in group A. Again, compare the standard search problem with the variation in which box 1 is replaced with a line segment of length t_1 swept continuously by the searcher. The searcher finds the target with probability q_1 when she meets it, and has the ability to restart where the previous sweep stopped. The proof of Lemma 3.2.2 goes through with obvious modification—because all search modes in group A have the same ratio between their detection probability and search time. The conclusion then follows from Lemma 3.2.3.

We have shown that in the modified search problem with $2K - 1$ search modes for box 1, there exists an optimal within-box subsequence for box 1 consisting only of mode (q_1, t_1). Because such a subsequence is feasible for the original problem where box 1 has K search modes, the result follows. ■

PART II: Two-Sided Search

You never really understand a person until you consider things from his point of view... Until you climb inside of his skin and walk around in it.

– Atticus Finch in “To Kill A Mockingbird” by Harper Lee

Foreward

Part II of this thesis concerns two-sided search, where, in addition to the first decision maker (the searcher), there is a second decision maker who controls the position of the target. There are two main types of two-sided search, that where the second decision maker wants the target to be found, called *rendezvous search*, and that where they do not want the target to be found, called a *search game*. Part II focuses on the latter, relevant to a search for a hidden bomb or landmine, intruder, or computer hacker. Throughout, we assume that the second decision maker is the target itself, whom henceforth we shall call the *hider*.

Unlike the one-sided search of Part I, in a search game, the searcher's best course of action depends on the actions of the hider. Therefore, game theory is commonly used to model search games, in particular, two-person, zero-sum game theory, with the game zero sum since the objectives of the searcher and hider are in direct conflict. An introduction to two-person, zero-sum game theory is given in Chapter 5, before the search-game literature is reviewed in Chapter 6.

In Part I, a classic one-sided search problem in discrete boxes is extended to two search modes per box. In Part II, the same classic model is extended to two players, transforming it into a search game. As will be discussed in Section 6.2.1, such a search-game extension has been studied in the literature, but never in full generality with an arbitrary number of boxes and arbitrary search times. The aim of the work in Chapter 7, to be submitted for publication in 2020, is to provide a comprehensive theory for the fully-general search-game extension. It will be shown that Gittins index policies, optimal in the one-sided problem, still play a key role in an optimal search strategy in the search-game extension, but which box to search when multiple boxes have a maximal index, trivial in the one-sided problem, takes great importance in the search game. Chapter 8 outlines further work.

Recall from Part I that optimal search policies in any one-sided search prefer parts of the search space either likely to contain the target or that are efficient to search. By choosing where to hide, the hider will try to remove any such preference, aiming to leave all areas equally appealing to the searcher. In particular, in the classic time-independent problem where the Gittins index of a box measures its attractiveness to the searcher, the hider will aim to equate all Gittins indices. Having no strong preference of where to search next has already been seen to be unattractive to the searcher in Part I. In the one-sided two-box problems of Flesch et al. (2009) and Kan (1974) (discussed, respectively, in Sections 1.2.3 and 1.2.4 of Part I) where the searcher may stop or delay the search, the stop or delay option is optimal when the searcher is approximately equally attracted to both boxes. Further, in the one-sided, two-mode search of Chapter 3 of Part I, the searcher may optimally avoid a situation of no strong box preference by using a search mode with inferior search parameters. Whilst the effect of this lack of preference is negligible in the one-sided, two-mode search, Part II will show its influence on the search game is of great significance.

Chapter 5

Two-Person Zero-Sum Games

The following introduction to two-person zero-sum games was constructed with the aid of Ferguson (2020), Blackwell and Girshick (1954), Straffin (1993) and Osborne and Rubinstein (1994).

As the name suggests, such a game involves two players with one player winning precisely what the other one loses. Both players have a set of actions which can be thought of as the moves available to that player in the game. Let the set of actions for the first player, P1, be X and for the second player, P2, be Y. Each time the game is played, P1 selects some action $x \in X$ and P2 some $y \in Y$ with both players unaware of the other player's choice. To determine which actions they select, each player employs a *strategy* of which there are two kinds, pure and mixed. The former involves the player taking the same action every time they play the game. The latter involves the player, each time they play the game, choosing an action at random with respect to some probability distribution on their set of actions. The distribution represents the mixed strategy. For example, if $X = \{a, b\}$, a pure strategy for P1 would be to always choose a, and a mixed strategy $\mathbf{x}$ would choose a with probability 0.6 and b with probability 0.4. Since X is finite, we may represent $\mathbf{x}$ as a vector $(0.6, 0.4)$ of length $|X|$. Note that all pure strategies are mixed strategies, yet with a degenerate probability distribution.

Each player's choice of action determines the payoffs received at the end of that iteration of the game. Therefore, we represent payoff via a function A from $X \times Y$ to $\mathbb{R}$. If P1 selects action $x \in X$ and P2 selects $y \in Y$, then $A(x, y)$ is the payoff to P1 and, since the game is zero sum, $-A(x, y)$ is the payoff to P2. Because of this relationship, henceforth, we shall refer to the payoff of P1 as the payoff of the game. Note that, if either P1 or P2 play a mixed strategy, then the payoff is random, and hence we can only calculate the expected payoff. The aim of P1 (resp. P2) is to maximise (resp. minimise) the expected payoff. Therefore, each player is not interested in the level of variance in their actual payoffs; for example, to P1, a guaranteed payoff of 1 is considered as desirable as a 50% chance of a payoff of 2 and a 50% chance of a payoff of 0.

The expected payoff depends on both players' actions; therefore, it is not entirely clear what an optimal strategy for either player is. Clearly, if P1 knew that P2 would play mixed strategy $\mathbf{y}$, then P1 optimally selects the pure strategy $x \in X$ which maximises the expected payoff given P2 plays $\mathbf{y}$. Given this knowledge is lacking, P1 could consider 'playing safe' in the following way. For any P1 strategy $\mathbf{x}$, there exists $y_{\mathbf{x}} \in Y$ which minimises the expected payoff given P1 plays $\mathbf{x}$. In other words, if P1 plays $\mathbf{x}$, then the expected payoff is guaranteed to be no less than if P2 counters with $y_{\mathbf{x}}$. Let a *safe strategy* $\mathbf{x}^*$ for P1 be a strategy with the largest such guaranteed expected payoff, say v_1^*. Similarly for P2, a safe strategy $\mathbf{y}^*$ guarantees P2 the smallest expected payoff, say v_2^*, that could possibly be guaranteed. Clearly $v_1^* \leq v_2^*$, and if P1 plays $\mathbf{x}^*$ and P2 plays $\mathbf{y}^*$, then the expected payoff lies within $[v_1^*, v_2^*]$. The celebrated theorem of von Neumann (1928), called the minimax theorem, states that, when $|X|$ and $|Y|$ are finite (known as a *finite game*), we have $v_1^* = v_2^*$.

The minimax theorem provides extra motivation for P1 to play a safe strategy $\mathbf{x}^*$ in a finite game. Not only does $\mathbf{x}^*$ maximise the expected payoff in the worst-case scenario, it is also the best strategy they can play against any P2 safe strategy $\mathbf{y}^*$. Further, if P1 and P2 play their safe strategies, so the expected payoff is $v_1^* = v_2^*$, then any change in strategy from P1 can only decrease the expected payoff, as $\mathbf{y}^*$ guarantees P2 at most v_2^*. Therefore, if both players play safe strategies, there is no motivation for either player to change strategies, and the game is at an equilibrium. Consequently, safe strategies $\mathbf{x}^*$ and $\mathbf{y}^*$ are known as minimax or optimal strategies, and we define $v^* \equiv v_1^* = v_2^*$ as the *value* of the game, the expected payoff if both players play optimally. Formally, the minimax theorem states that all finite games have a value given by

$$v^* \equiv \max_{\mathbf{x}} \min_{y \in Y} A(\mathbf{x}, y) = \min_{\mathbf{y}} \max_{x \in X} A(x, \mathbf{y}), \tag{5.0.1}$$

where $A(\mathbf{x}, \mathbf{y})$ represents the expected payoff if P1 and P2 play mixed strategies $\mathbf{x}$ and $\mathbf{y}$, respectively.

In a finite game, it is possible to find the optimal strategies for both players and the value of the game using linear programming methods (See Chapter 10 of Ferguson (2020)). It can be difficult to solve games where $|X|$ and $|Y|$ are large, yet there are methods that can reduce the dimensions of such problems. An example of such a method involves removing a pure strategy if it always produces a worse payoff than another pure strategy.

5.1 Infinite Games

Now consider *infinite games*, in which P1 and P2 have an infinite number of pure strategies, and mixed strategies for P1 (resp. P2) are defined by probability measures on X (resp. Y). The minimax theorem of von Neumann (1928) shows all finite games have a value; however, the same is not true for infinite games. Blackwell and Girshick (1954) shows that an infinite game has a value if X and Y are compact in Euclidean space and A is continuous. Several authors have since studied conditions under which an infinite game has a value (Terkelsen, 1972; Gal, 1980; Alpern, 1985; Alpern and Gal, 1988).

The value of an infinite game must be defined slightly differently to the finite-game definition in (5.0.1), as, since the strategy sets are infinite, we must take infimums and supremums. If an infinite game has a value v^*, we have

$$v^* \equiv \sup_f \inf_{y \in Y} \int A(x,y)f(x)dx = \inf_g \sup_{x \in X} \int A(x,y)g(y)dy, \tag{5.1.1}$$

where $f(x)$ (resp. $g(y)$) represents a density of some probability measure on X (resp. Y). Yet, unlike that in (5.0.1), the definition in (5.1.1) does not guarantee the existence of an optimal strategy for either player, since the supremums and infimums may not be attained. Instead, for any $\epsilon > 0$, P1 may only have an ϵ-optimal strategy, which guarantees an expected payoff of at least $v^* + \epsilon$, and, similarly, P2 may only be guaranteed at most $v^* - \epsilon$.

Chapter 13 of Ferguson (2020) shows that in the game of Blackwell and Girshick (1954), with X and Y compact in Euclidean space and A continuous, optimal strategies exist for both players. Alpern and Gal (1988) derives conditions under which an optimal strategy exists for the minimising player (P2).

5.2 Semi-Infinite Games

Blackwell and Girshick (1954) ensures that *semi-infinite games* (also known as semi-finite games), in which one player has an infinite number of pure strategies and the other a finite number, have a much simpler classification than their infinite counterparts.

Without loss of generality, assume that $X = \{1, \ldots, n\}$ and Y is infinite. By Theorem 2.4.2 of Blackwell and Girshick (1954), if the payoff function A is bounded below, then a semi-infinite game has a value, and P1 has an optimal strategy. Since the game has a value, P2 must have an ϵ-optimal strategy for any $\epsilon > 0$. Section 2.4 of Blackwell and Girshick (1954) also determines conditions for P2 to have an optimal strategy by considering the equivalent 'S-game', where P2, instead of choosing a strategy in Y, chooses a vector in

$$S \equiv \{(A(1,y), \ldots, A(n,y)) : y \in Y\} \subset \mathbb{R}^n.$$

Theorem 2.4.1 of Blackwell and Girshick (1954) shows that P2 selecting a mixed strategy is equivalent to choosing a point in S^*, the convex hull of S. Recall a mixed strategy for P2 is a probability density f on Y. If f has finite support given by $\text{supp}(f) \equiv \{y \in Y : f(y) > 0\}$, then we say f is a *mixture* of $\text{supp}(f)$. Theorem 2.4.2 of Blackwell and Girshick (1954) concludes that if S (or equivalently S^*) is closed, then P2 has an optimal strategy which is a mixture of at most n pure strategies in Y.

The intuitive reasoning behind this result is the following, adapted from Chapter 13 of Ferguson (2020). If $\mathbf{s} \equiv (s_1, \ldots, s_n) \in S^*$, then there exists a P2 mixed strategy which, if P1 plays pure strategy i, achieves an expected payoff s_i, $i = 1, \ldots, n$. It follows that the value of the game v^* satisfies

$$v^* = \inf \left\{ \max_{i \in \{1,\ldots,n\}} s_i : \mathbf{s} \in S^* \right\}. \tag{5.2.1}$$

If S^* is closed, then the infimum in (5.2.1) is attained, so there exists $\mathbf{s}^* \in S^*$ on the boundary of S^* with $\max_{i \in \{1,\ldots,n\}} s_i^* = v^*$; it follows that $\mathbf{s}^*$ is an optimal strategy for P2. By Theorem 2.2.2 of Blackwell and Girshick (1954), since S^* is the convex hull of S, any boundary point of S^* can be written as a convex combination of at most n elements of S, with these elements forming $\text{supp}(\mathbf{s}^*)$. On the other hand, if S^* is not closed, then the infimum in (5.2.1) may not be attained, so there may not exist an optimal strategy for P2.

Another approach to studying semi-infinite games is taken by Soyster (1975). The same conclusions are drawn, but via extending the linear program formulation often used to solve finite games to semi-infinite games, and then applying duality theorems over cones.

Chapter 6

Literature Review

This chapter reviews the literature on *search games*, in which a searcher seeks a hider who tries to avoid detection. Since the searcher and hider have directly conflicting objectives, search games are zero-sum, so the theory of Chapter 5 is applicable. Many types of search game have been explored by the literature, yet all models feature the following. There is a search space R known to and traversed by both players from time 0 onwards. A strategy for either player is a path in R to follow, alongside, in some formulations, corresponding speeds at which to travel. Any strategy needs to specify a path assuming that the hider is never detected, so a path continuing either indefinitely or, if one exists, until a search deadline. In most formulations, and unless otherwise specified, the hider chooses their starting point, but the searcher's starting point is fixed.

If the players come within a known *detection distance* of each other, the search ends with some known *detection probability*. Before the search ends, neither player is aware of the other player's position. The payoff is some measure of the success of the search. Common choices are the time until the searcher finds the hider or whether the searcher finds the hider before a deadline. Note that, with the former choice of payoff, the searcher minimises the expected length of the search, whilst with the latter, the hider minimises the probability of detection before the deadline.

The following survey will be split by the detection capabilities of the searcher. The detection probability is 1 in Section 6.1, whilst the searcher may overlook the hider when passing nearby in Section 6.2.

6.1 Perfect-Detection Models

The literature review in this section is aided by Book 1 of Alpern and Gal (2003) and Part I of Alpern et al. (2013). We review models with detection probability 1; in other words, the searcher has perfect detection, and the search ends the first time that the players are within the detection distance of each other. Unless stated otherwise, the payoff for all models reviewed in this section will be the time the search ends. In other words, the expected payoff is the *expected search time* which the searcher aims to minimise and the hider to maximise. For a general search space R, both players have an infinite number of pure strategies, and hence the search game is an infinite two-person zero-sum game. The combined work of Gal (1980) and Alpern and Gal (1988, 2003) shows that any perfect-detection search game discussed in this section has a value, and the minimising player (searcher) has an optimal strategy. The hider, however, may only have ϵ-optimal strategies.

There is a rich literature on perfect-detection models with R unbounded, motivated by the search game on the real line of Beck and Newman (1970), a two-sided extension of the one-sided linear search first studied in Beck (1964), discussed in the literature review of Part I of this thesis. For a survey of unbounded search spaces, see Part Two of Book I of Alpern and Gal (2003). Our focus henceforth will be on R bounded and continuous. Study of such perfect-detection search games was inspired by two problems presented in Isaacs (1965), themselves motivated by combat scenarios. The first allows the hider to be 'mobile' and take any path in R, whilst the second is a simplification of the first with an 'immobile' hider who must remain stationary in R. The next two sections discuss the development in the literature of both problems individually.

6.1.1 The Mobile Hider

Isaacs (1965) calls the problem with a mobile hider and arbitrary, non-zero detection distance the 'Princess and Monster' (PM). The maximum speed of the searcher (Monster) is 1 and of the hider (Princess) is $\omega > 0$. Gal (1979) solves PM for convex R, inspiring many further modifications to PM; for example, Gal (1980) studies a non-convex R, and Garnaev (1992) allows detection to be imperfect and depend on the distance between the two players.

PM is well studied for R a *finite network*, which consists of a finite collection of points in $\mathbb{R}^2$ (nodes or vertices) and lines of various lengths (edges or arcs) which join some pairs of nodes. It is assumed that there is a path (sequence of edges) between every pair of nodes; in other words, R is *connected*. Players may travel down edges and through nodes, and, since edges and nodes are one-dimensional, the detection distance is taken to be 0.

The boundary of a circle may be viewed as a network, a case solved independently in Alpern (1974), Foreman (1974) and Zelikin (1972) under the assumption that $\omega \geq 1$ and, unlike most PM models, both players' starting points are uniformly distributed. Alpern and Gal (2003) solves PM on a network R_k consisting of two nodes connected by some $k \in \mathbb{N}$ unit-length edges. The searcher starts at the origin O, and the hider, as usual, chooses their starting point. The searcher has an optimal strategy related to that of Alpern (1974), but the hider must settle for an ϵ-optimal strategy. Figure 6.1.1 shows R_3 with nodes O and A.

PM becomes harder and remains unsolved on any other network. Even the unit interval (R_1) is hard to solve when the searcher can choose their starting point; Alpern et al. (2009) only conjectures a solution. Yet, Chapter 5 of Alpern et al. (2013) remarks that, if the conjectured solution in Alpern et al. (2009) is true, it may be possible to extend it to a tree network.

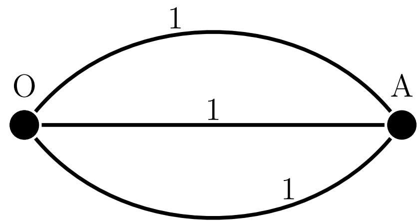

Figure 6.1.1: R_3: Two nodes, O and A, connected by three unit-length edges.

Washburn (1980) studies a version of PM on a *complete* network, where every pair of nodes is connected by an edge. The searcher has unit speed, and the hider can only hide at nodes. However, with knowledge of the searcher's current node, the hider can instantaneously switch nodes just before the searcher leaves their current node to travel to another. Therefore, the hider is motivated to switch to one of the nodes furthest from the searcher's current node. However, the searcher can always reach any node via a single edge, so always choosing one of the furthest nodes puts the hider at risk of being too predictable. Thomas and Washburn (1991) studies the model of Washburn (1980) with the expected payoff the probability that the search ends before a deadline. Detection is generalised in the sense that, for any nodes i, j, the searcher may detect the hider in node i whilst searching node j with some known probability; imperfect-detection models are studied in greater detail in Section 6.2.

The unit interval is the only network where the searcher has a pure strategy in PM guaranteed to eventually detect the hider. On any other network, if the hider knew the searcher's pure strategy, they could choose their path to remain forever undetected, showing that any optimal search strategy must be mixed. The same logic is not true in Isaacs' immobile-hider problem, in which pure search strategies guaranteed to eventually find the hider play a key part.

6.1.2 The Immobile Hider

Isaacs (1965) also presents a simpler 'hide-and-seek' (HS) version of PM in which the hider is immobile. A pure hiding strategy in HS is a single point in R, and a mixed hiding strategy is a probability measure on R. Since the hider does not move, the searcher optimally moves as fast as they can, so no generality is lost by assuming the searcher has fixed, unit speed.

Any search path that passes within the detection distance of each point of R is guaranteed to find the hider; if such a path is closed (starts and ends at the same point in R), we call it a *tour* of R. If there exists a tour of R with length μ which does not cover the same ground twice, then Isaacs (1965) shows that the value of HS is $\mu/2$. An optimal search strategy is to follow such a tour *randomly*; i.e., forwards (resp. backwards) with probability 0.5. An optimal hiding strategy is to hide uniformly in R.

Isaacs' result motivated the study of HS on a finite, connected network R, the focus of the remainder of this section. Write μ for the sum of the lengths of all edges of R, and v for the value of the game. Henceforth, we split our survey of HS, first depending on the starting point of the searcher.

Fixed Searcher Starting Point First, as in Isaacs' original problem, assume a fixed starting point for the searcher, which we take to be the origin node O. Gal (1980) shows that, as in Isaacs' problem, hiding uniformly guarantees an expected payoff of at least $\mu/2$; therefore, $v \geq \mu/2$. A key type of pure search strategy is a *Chinese Postman Tour* (CPT), a tour of shortest possible length $\bar{\mu}$, which it follows must satisfy $\bar{\mu} \leq 2\mu$. Gal (1980) shows that the searcher guarantees an expected payoff of at most $\bar{\mu}/2$ by following a CPT randomly (a RCPT), and hence that $\mu/2 \leq v \leq \bar{\mu}/2 \leq \mu$. A network R is *Eulerian* if $\mu = \bar{\mu}$, in which case $v = \mu/2$, and hiding uniformly (resp. a RCPT) is optimal for the hider (resp. searcher).

For a non-Eulerian network, $\mu < \bar{\mu} \leq 2\mu$. The upper bound is attained only on a (rooted) tree with origin the root, where Gal (1980) shows that a RCPT is optimal for the searcher. It follows that $v = \bar{\mu}/2 = \mu$. Hiding uniformly is not a good strategy on a tree; the hider clearly optimally only hides at leaf nodes, the nodes furthest away from the origin. Gal (1980) derives a distribution on the leaf nodes, later coined the Equal Branch Density (EBD) by Gal (2001), which is optimal for the hider on a tree.

Whilst $v = \mu$ only for trees, the optimality of a RCPT, and hence $v = \bar{\mu}/2$, holds for a wider class of networks. A network is *weakly cyclic* if it can be constructed from a tree by replacing some nodes with cycles; Reijnierse and Potters (1993) shows a RCPT is optimal for such networks. Reijnierse (1995) extends the result further to *weakly Eulerian* networks. These are networks containing disjoint Eulerian networks which, when shrunk to a point, leave a tree. (More precisely, these are networks where the blocks in the bridge-block decomposition of the network are all Eulerian.) In both cases, an extension to the EBD was shown to be optimal for the hider. Gal (2001) strengthens the result of Reijnierse (1995) by proving a converse, namely that *only* on weakly Eulerian networks is a RCPT optimal.

It follows that $v < \bar{\mu}/2$ for non-weakly-Eulerian networks, and hence the searcher can do better than a RCPT. However, optimal strategies for either player can be very complicated and finding them is NP hard (von Stengel and Werchner, 1997). An example, first posed by Gal (1980), is HS on R_3 shown in Figure 6.1.1, a case where the network is simple, but the solution, first found by Pavlović (1993), is not.

Several recent extensions to HS are as follows. Alpern (2010) and Alpern and Lidbetter (2014) study *asymmetric* networks, where, for any two adjacent nodes i and j, it may take longer to travel from i to j than from j to i. The former uses recursion formulae to calculate the value of HS on an asymmetric rooted tree. The latter obtains a closed form for this value which is used to study HS on general asymmetric networks.

In the find-and-fetch model of Alpern (2011), the searcher must return to the origin O 'carrying' the hider before the search completes, and therefore the hider is motivated to hide far away from O. Two scenarios are considered: the searcher's speed carrying the hider may be fast, as they are no longer on the lookout for the hider, or slow, if the hider is heavy.

The expanding search of Alpern and Lidbetter (2013) allows the searcher to 'jump back' immediately to any previously-searched location and resume searching, relevant to non-physical applications such as trying to solve a mathematical problem, or mining for coal, where searching new ground takes a lot longer than travelling back through old ground.

The combinatorial paths of Alpern (2017) only allow the searcher to change direction at nodes, realistic for search in large vehicles. Only non-weakly-Eulerian networks are studied, since, by the result of Gal (2001), a RCPT, which does not involve changing direction on any edge, is optimal on any weakly-Eulerian network. For R_3 shown in Figure 6.1.1, a full solution is obtained.

Searcher Chooses a Starting Point Now suppose that the searcher can choose their starting point to be any point on R, a problem first formulated in Dagan and Gal (2008) which we call arbitrary-start hide-and-seek (AHS) with value $v_a \leq v$, where v is the value of HS on R. As for HS, we deduce $v_a \geq \mu/2$ by hiding uniformly; it follows that $v_a = v = \mu/2$ on a Eulerian network, with identical optimal strategies in HS and AHS.

Now we consider non-Eulerian networks. Of interest in AHS is a *Chinese Postman Path* (CPP), a path of shortest possible length $\bar{\mu}_a$ that traverses all edges. Note that a CPP is similar to a CPT which has length $\bar{\mu}$, yet a path, unlike a tour, does not need to start and finish at the same place; therefore, $\mu \leq \bar{\mu}_a < \bar{\mu}$. As for HS, we deduce $v_a \leq \bar{\mu}_a/2$ by the searcher following a CPP *randomly* (a RCPP). However, for a CPP, we define randomly to be forwards from the start with probability 1/2 and backwards from the end with probability 1/2, only possible since the searcher can choose their starting point.

If $\bar{\mu}_a = \mu$, then clearly $v_a = \mu/2$, with optimal strategies an RCPP and hiding uniformly. An example is R_3 in Figure 6.1.1, which has a CPP starting at O and finishing at A of length $\mu = 3$. However, R_3 has no tour of length 3, so whilst AHS on

R_3 has $v_a = \mu/2$ and is easy to solve, HS on R_3 has $v > \mu/2$ and, by Pavlović (1993), is hard to solve.

Dagan and Gal (2008) shows that a RCPP is optimal for a tree, with, similar to HS, an adaptation of the EBD of Gal (1980) optimal for the hider. Alpern (2008) shows that a RCPP is optimal for a tree with Eulerian graphs attached, a subset of weakly-Eulerian networks. Yet, Alpern et al. (2008) proves it is impossible to characterise the networks for which a RCPP is optimal, so there is no result analogous to that of Gal (2001) for HS and RCPTs.

A Two-Speed Model The model of Alpern and Lidbetter (2015) (T-SHS) extends HS on a network by introducing a second, faster speed (fast) at which the searcher can travel in addition to the original (slow). When passing the hider's location, detection will always occur at the slow speed, but not always at the fast. Two scenarios were considered, both best suited to a small but exposed hider. In the first scenario, the fast speed has no chance of detecting the hider and, therefore, is useful only to save time moving across the network. In the second scenario, solved only on the unit interval, the fast speed has a detection probability in $(0, 1)$.

In the first scenario, the authors prove T-SHS has a value and an optimal searching strategy. Further, it is shown that on a Eulerian network the fast speed is never optimally used, and hence T-SHS reduces to HS. On the other hand, to tour a non-Eulerian network, the same ground must be covered twice. Since the slow speed has perfect detection, it therefore must be optimal to use the fast speed on one of the passes.

A *bimodal CPT* (BCPT) was defined as a tour of shortest possible length $\bar{\mu}$ which traverses each edge at least once at the slow speed, and a *random BCPT* (RBCPT) follows a BCPT in either direction with equal probability. Similarly to HS, where v is the value of T-SHS, we deduce $v \leq \bar{\mu}/2$ by the searcher following a RBCPT. It was shown that $v = \bar{\mu}/2$ for trees and, more generally, for weakly-Eulerian networks.

Therefore, a RBCPT is optimal for the searcher, and a result similar to that of Reijnierse (1995) for HS holds for T-SHS. It is unknown whether the converse result, proved by Gal (2001) for HS, is true for T-SHS, namely that only for weakly Eulerian networks is $v = \bar{\mu}/2$.

The addition of a second speed in Alpern and Lidbetter (2015) has clear links to Chapter 3 of Part I of this thesis, where a second speed was added to the classic one-sided, discrete search problem. Yet, the two-sided nature of T-SHS, its network search space, and the perfect detection at the slow speed result in the methodologies being unconnected; therefore, T-SHS is discussed only here in Part II.

Hiding at Nodes Now suppose that the hider can only hide at the nodes of a network. In this formulation of HS (NHS), the hider has a finite number of pure strategies, so we have a semi-infinite two-person zero-sum game for which, by Alpern and Gal (1988), a value and optimal strategies for both players exist. However, assuming the starting point of the searcher is fixed, NHS is actually more difficult than HS. Of interest is a *Travelling Salesman Problem Tour* (TSPT), a closed path of minimal length $\tilde{\mu}$ that visits every node. Finding a TSPT on a general network is NP hard, whilst a CPT can be found in $O(n^3)$ time where n is the number of nodes. A random TSPT (RTSPT) ensures $v_N \leq \tilde{\mu}/2$, where v_N is the value of NHS; yet, Alpern et al. (2013) proves it is not possible to classify networks where $v_N = \tilde{\mu}/2$.

An extension of NHS (STNHS) assumes that, upon arrival at a node, it takes an additional, known (to both players) *search time* to search the node for the hider. The searcher may choose to pass the node for no extra time cost, but in doing so forgoes the chance to detect the hider. The payoff is the sum of the travelling time between nodes and the search time at nodes before the hider is found.

Kikuta (1990) solves STNHS for equally-spaced nodes with identical search times which lie on a straight line - the two-sided extension of Gluss (1961) discussed in Part I of this thesis. Kikuta (1991) generalises the solution so the distance between adjacent nodes need not be constant. Kikuta and Ruckle (1994) solves STNHS on a rooted tree where every leaf node is connected to the root/origin, and Kikuta (1995) finds a solution for a general rooted tree. Recall that, in HS on a tree, the hider optimally hides only at leaf nodes. Therefore, on a tree, HS and NHS are equivalent; however, STNHS is different because of the addition of a search time at each node. Kikuta (2004) studies STNHS on a cyclic network in which all nodes lie on a circle. The asymmetric networks of Alpern (2010) and Alpern and Lidbetter (2014) are applied to STNHS by Baston and Kikuta (2015); however, in addition to direction-dependent travel times between nodes, edges where travel is only possible in one direction are allowed. Baston and Kikuta (2013) allow the searcher to choose their starting node in STNHS, bounding the value above in the general case and below when all edges are of equal length.

Alpern and Lidbetter (2015) note that STNHS has links with their two-speed model, T-SHS, with passing a node comparable to the fast speed and searching a node comparable to slow. However, recall that T-SHS allows hiding along edges, while STNHS does not.

Hiding in Boxes A model named *BOX* by Hellerstein et al. (2019) is obtained by removing travel time from STNHS, equivalent to a search space of n discrete locations (boxes), all within immediate reach of the searcher, and all with an associated search time. A pure strategy for the hider is a box to hide in, of which there are n, and a pure strategy for the searcher is a permutation of $\{1, \dots, n\}$, of which there are $n!$, detailing the order in which to search the n boxes. Therefore, BOX is a finite game, and its solution, presented in Alpern and Lidbetter (2013), is simple. The unique optimal hiding strategy hides in a box with a probability proportional to its search cost. There are many optimal search strategies, see Alpern and Lidbetter (2013) or Lidbetter (2013).

Several extensions to BOX considered in the literature are as follows.

Dresher (1961) extends BOX with unit search times by, after an unsuccessful search of box i, telling the searcher whether the hider's true box j satisfies $j < i$ or $j > i$. Dresher (1961) solves the problem for $n = 3$, with Johnson (1964) deriving properties of optimal strategies and solving the game for $n \leq 11$. Recently, Fokkink and Stassen (2011) provides an algorithm to solve the game for $n \leq 256$ and a formula for the value of the game as $n \to \infty$.

Efron (1964) studies BOX with unit search times and a mobile hider who can switch boxes between searches. This problem is clearly trivial; all boxes are identical, so, before each search, both players choose box i to hide/search with probability $1/n$. Efron (1964) introduces a deadline $d \in \{1, \ldots, n\}$, with, rather than the search time, the payoff to the hider 0 if detection occurs within the first d searches and 1 otherwise. Further, the searcher is not allowed to search a box more than once. By symmetry, the searcher's optimal strategy is to randomly choose d boxes from $\{1, \ldots, n\}$ and search them in a random order. Despite no restrictions on the hider's movement, Efron (1964) shows that the hider's optimal strategy is identical to the searcher's. Efron (1964) also considers a payoff dependent on the number of searches required to find the hider before the deadline and allows the searcher to search multiple boxes at once.

Both Lidbetter (2013) and Lidbetter and Lin (2019) consider a searcher looking for multiple objects hidden by one hider. The former permits only one object per box and minimises the expected time until all objects are detected. No benefit is found in changing search plan at any point during the search, no matter how many objects have thus far been detected. As well as in boxes, Lidbetter (2013) also considers search on a network structure with the expanding searches of the aforementioned Alpern and Lidbetter (2013) permitted. The game is solved on simple networks, and the value bounded for a general network, where, unlike search in boxes, changing search plan midway may be beneficial. Lidbetter and Lin (2019) considers two extensions to the box search of Lidbetter (2013). In the first, multiple objects may share the same box, but only one can be found at once, so one box may need several searches. In the second, the expected *regret* is minimised, defined as the possible decrease in search time if the searcher knew where all objects were hidden.

Hellerstein et al. (2019) provides general methods for solving finite games where, for any player 1 strategy, an optimal counter strategy for player 2 is easy to obtain. They show how their methods can solve more extensions to BOX, including the expanding search of Alpern and Lidbetter (2013) and the submodular search of Fokkink et al. (2019).

Away from the search-game literature, Anderson and Weber (1990) studies a rendezvous-version of BOX. Recall that in a rendezvous search, the second decision maker is not a hider, but a second searcher who aims to find the first searcher as soon as possible. Therefore, both searchers share the same objective, so the game is no longer zero sum, but rather a cooperative, coordination game. In Anderson and Weber (1990), the boxes are identical, and both searchers move in discrete time. The search ends the first time that both searchers occupy the same box, with both searchers aiming to minimise the expected search time. If each searcher always moves to box i with probability $1/n$, $i = 1, \ldots, n$, the expected search time is n. Anderson and Weber (1990) shows such a random strategy is optimal when $n = 2$, but for $n = 3$, the searchers can do better; the optimal expected search time is shown to be $8/3$ and an strategy optimal for both searchers is found. Anderson and Weber (1990), in fact, initiated work on rendezvous search, 14 years after the problem was posed in Alpern (1976), a seminar in Vienna.

An extension to BOX relevant to this thesis has a searcher with imperfect detection, the subject of the next section.

6.2 Imperfect-Detection Models

In an imperfect-detection model, when the searcher and hider pass within the detection distance, the hider is found with a known location-dependent detection probability strictly less than 1. Models with imperfect detection are less well studied than their perfect-detection counterparts, not least because of their increased difficulty; for an immobile hider, unlike a perfect-detection model, a tour of the search space does not guarantee the hider's discovery. Further, the result of Alpern and Gal (1988) for perfect-detection models guaranteeing the existence of a value and optimal search strategy no longer applies.

The simplest model with *imperfect* detection—referred to as *IBOX*—is the extension of BOX where the searcher may overlook the hider when searching the box containing them. Since development of IBOX is the topic of Chapter 7 of Part II of this thesis, IBOX and its extensions are the focus of remainder of this literature review. Note that IBOX itself is the two-sided extension to the classic one-sided, time-independent search problem studied in Part I of this thesis. As in the classic problem with the target replaced by a hider, for $i = 1, \dots, n$, the hider is located in box i with *hiding probability* p_i, and a search of box i takes known *search time* $t_i > 0$, finding the hider, if in box i, with known *detection probability* $q_i \in (0, 1)$. However, whilst in the classic problem the target distribution $\mathbf{p} \equiv (p_1, \dots, p_n)$ with $\sum_{i=1}^{n} p_i = 1$ is known to the searcher and determined by Nature, in IBOX, the *hiding distribution* $\mathbf{p}$ is chosen by the hider and is unknown to the searcher.

Like BOX, IBOX has n pure hiding strategies, each corresponding to a box to hide in, with a mixed hiding strategy a hiding distribution $\mathbf{p}$. A pure search strategy in IBOX is a *search sequence*, an ordered list of boxes to search (referred to as a search policy in Part I). Since after no number of searches is the hider's detection guaranteed, unlike in BOX, a search sequence has infinite length; therefore, the searcher has an infinite number of pure strategies, and IBOX is a semi-infinite two-person, zero-sum game. Since all possible payoffs are positive, as discussed in Section 5.2, by Blackwell and Girshick (1954), IBOX has a value and the hider has an optimal strategy.

Recall from Part I that, in the classic one-sided, time-independent search problem with target distribution $\mathbf{p}$, the expected search time is minimised by the following Gittins index policy against $\mathbf{p}$. If $m_i \in \mathbb{Z}^+ \equiv \{1, 2, \ldots\}$ searches have already been made of box i, $i = 1, \ldots, n$, the next search is of any box with a maximal value of

$$\frac{p_i(1-q_i)^{m_i} q_i}{t_i}, \tag{6.2.1}$$

with (6.2.1) known as the *Gittins index* of box i. Equivalently, the next search is of any box with maximal

$$\frac{p_i' q_i}{t_i}, \tag{6.2.2}$$

where, by Bayes' theorem, $p_i' \equiv p_i(1-q_i)^{m_i}/k$ is the posterior probability that the target is in box i, $i = 1, \ldots, n$, $k \equiv \sum_{j=1}^{n} p_j(1-q_j)^{m_j}$. Therefore, if the searcher in IBOX knew the hider's mixed strategy $\mathbf{p}$, then any search sequence generated by a Gittins index policy, called a *Gittins index sequence against* $\mathbf{p}$, minimises the expected search time.

As discussed in Part I, several papers from the early 1960s independently prove the optimality of a Gittins index policy in the classic one-sided, time-independent search problem. Among these papers are Norris (1962) and Bram (1963), the first two papers to study IBOX. Section 6.2.1 discusses the analyses of IBOX by Norris (1962), Bram (1963) and others in the literature. Extensions to and variations on IBOX in the literature are discussed in Section 6.2.2.

6.2.1 IBOX: Hiding in Boxes

We are aware of seven papers that have studied IBOX (Norris, 1962; Bram, 1963; Roberts and Gittins, 1978; Gittins and Roberts, 1979; Gittins, 1989; Ruckle, 1991; Lin and Singham, 2015). This subsection discusses any contribution made by each to different aspects of IBOX.

Existence and Form of an Optimal Searching Strategy Bram (1963) concentrates on proving the existence of an optimal search strategy for IBOX with unit search times. The equivalent S-game is considered (see Section 5.2 here), and S^*, the convex hull of S, is shown to be closed. Therefore, by Theorem 2.4.2 of Blackwell and Girshick (1954), IBOX with unit search times has an optimal search strategy which is a mixture of at most n search sequences.

For IBOX with $n = 2$ and unit search times, Chapter 8 of Gittins (1989) finds an optimal mixed search strategy η^* which, no matter which box the hider hides in, attains the same expected search time— an *equalising* strategy. Further, η^* is a mixture of only two search sequences, both Gittins index sequences against an optimal hiding strategy. Recall when there is a tie for the maximal Gittins index in (6.2.1), a Gittins index policy is free to search any box with an index involved in the tie. The first (resp. second) search sequence in the mixture η^* is generated by searching box 1 (resp. 2) whenever there is a tie.

Ruckle (1991) proves that the pure strategy which searches boxes in the order $1, 2, \ldots, n, n-1 \ldots, 1, 2, \ldots$ is optimal for the searcher in the search game with $q_i = 0.5$ and $t_i = 1$, $i = 1, \ldots, n$. For a general n-box search game with unit search times, Ruckle (1991) shows there exists an equalising pure strategy, which, whilst not necessarily optimal, attains near-optimal performance.

While Ruckle's results do not extend to the arbitrary-search-time case, the aim of Chapter 7 is to extend the results of Bram (1963) and Gittins (1989) to IBOX with n boxes and arbitrary search times. Gittins (1989) claims an extension to his method is not difficult for arbitrary n; however, we find such an extension leads to several complications, and extend Gittins' result using a different approach in Chapter 7.

Optimal Strategies in Mobile-Hider IBOX Norris (1962) studies links between IBOX and its extension with a mobile hider who can switch boxes between searches. Unlike the perfect-detection models of Section 6.1, IBOX becomes easier to solve when the hider is mobile since the hider can 'reset' the game by hiding anew after every search. Consequently, mobile-hider IBOX reduces to a succession of identical games in which only one box is searched. Firstly for $n = 2$ and $t_1 = t_2 = 1$ in Chapter 3 and then in full generality in Chapter 8, Norris (1962) solves IBOX with a mobile hider by solving this single-search game, a finite two-person zero-sum game where both players have n pure strategies, each corresponding to a box to hide/search. The solution is as follows. After each search, independent of all previous actions, the hider optimally moves to box i with probability

$$p_{0,i} \equiv \frac{t_i/q_i}{\sum_{j=1}^{n} t_j/q_j}, \tag{6.2.3}$$

and the searcher optimally next searches box i with probability

$$\frac{1/q_i}{\sum_{j=1}^{n} 1/q_j}, \tag{6.2.4}$$

$i = 1, \ldots, n$. The value of the game is

$$\sum_{j=1}^{n} \frac{t_j}{q_j}. \tag{6.2.5}$$

Note that the optimal hiding strategy $\mathbf{p}_0 \equiv (p_{0,1}, \ldots p_{0,n})$ is the unique hiding strategy for which the indices in (6.2.2) across all n boxes are equal. Since (6.2.2) measures the appeal of searching its corresponding box next, when the current hiding distribution is $\mathbf{p}_0$, the searcher has no advantage in searching one box over another.

Further, returning to an immobile hider, if $q_i = 1$ for $i = 1, \ldots, n$, so IBOX becomes BOX, then $\mathbf{p}_0$ hides in box i with probability proportional to t_i, the unique optimal hiding strategy in BOX found by Alpern and Lidbetter (2013). Therefore, it is natural to wonder how widely and how close to optimal $\mathbf{p}_0$ is in IBOX, a question examined by Roberts and Gittins (1978), which studies unit-search-time IBOX for $n = 2$ and makes the following observation. In mobile-hider IBOX, by always hiding anew using $\mathbf{p}_0$, the hider optimally ensures equality in indices before every search. The hider should therefore aim to mimic this behaviour in IBOX, but is restricted to hiding once at the start of the search. Whilst $\mathbf{p}_0$ attains equality in indices before the first search, subsequent posterior hiding distributions are determined by the history of searches, which are out of the hider's control. Intuitively, the best the hider can do is hide with probability $\mathbf{p}^*$ such that, when the searcher optimally counters with a Gittins index sequence against $\mathbf{p}^*$, the generating Gittins indices in (6.2.1) are as close to being as equal as possible throughout the search. Since the chance that the hider remains undetected decreases as time passes, approximating equality in (6.2.1) earlier in the search takes more importance; therefore, $\mathbf{p}_0$, which begins with equality in (6.2.1), should be a good choice of strategy for the hider in IBOX.

Both Norris (1962) and Roberts and Gittins (1978) show that $\mathbf{p}_0$ is optimal when $n = 2$ and the boxes are identical, the latter by simply appealing to symmetry. Gittins and Roberts (1979), which studies unit-search-time IBOX for arbitrary n, extends the result to n identical boxes, and Ruckle (1991) adds an optimal search strategy for this case.

Ruckle (1991) also solves a two-box problem, where $\mathbf{p}_0$ may be represented by a scalar $p_0 \equiv p_{0,1}$. Both boxes have search time 1, and box 2 has perfect detection. It is shown that p_0 is optimal for $q_1 \in [0.618, 1]$; i.e., when box 1's detection is close to perfect. Section 7.4.3 analyses the connection between Ruckle's solution and our numerical study, offering an explanation of Ruckle's result.

Roberts and Gittins (1978) further investigates two-box problems with unit search times where p_0 is optimal. Without loss of generality, assume $q_1 \geq q_2$. Roberts and Gittins (1978) proves that if $(1 - q_1)^m = (1 - q_2)^{m+1}$ for some $m \in \mathbb{Z}^+$, then p_0 is optimal if and only if $m \leq 12$. A numerical study of problems with $(1 - q_1)^{m_1} = (1 - q_2)^{m_2}$ for $m_1, m_2 \in \mathbb{Z}^+$ finds that p_0 is more likely to be optimal for larger q_1 and q_2, explained by equality in (6.2.1) towards the start of the search taking even greater importance for q_1 and q_2 large since the hider is likely to be found sooner.

Over their numerical study, Roberts and Gittins (1978) concludes p_0 is a good heuristic strategy for the hider by examining its suboptimality (the percentage decrease in expected search time if the hider plays p_0 instead of an optimal p^* and the searcher optimally counters using a Gittins index sequence). In particular, in no problem sampled with (q_1, q_2) lying between the curves $q_1 = q_2$ and $1-q_1 = (1-q_2)^m$ is the suboptimality of p_0 more than 0.075% for $m = 2$ and 3.5% for $m = 10$, with the degradation with m explained by moving further away from the identical-boxes case where p_0 is optimal by symmetry.

Also via a numerical study with unit search times, Gittins and Roberts (1979) finds a relationship between two-box games with detection probabilities $q_l < q_u$ and four, eight and twelve-box games with detection probabilities in $[q_l, q_u]$, coined *related* games. In particular, if p_0 is optimal in a two-box game, $\mathbf{p}_0$ is often optimal in any related, larger game; otherwise, the suboptimality of $\mathbf{p}_0$ is than 0.2%. Further, for detection probabilities evenly spread throughout $[q_l, q_u]$, the suboptimality of $\mathbf{p}_0$ in a related game is almost always smaller than that of p_0 in the original. On the other hand, when each detection probability is equal to either q_l or q_u, (bar the identical-boxes case) the suboptimality of $\mathbf{p}_0$ grows with n, but even for $n = 12$, was generally less than twice as large as the suboptimality of p_0. It is concluded that $\mathbf{p}_0$ remains a good heuristic strategy for the hider for $n > 2$, with the aforementioned patterns again explained by the proximity to the identical-boxes case.

Recurrent Regions For $n = 2$, by definition of p_0, if the current posterior hiding probability for box 1 is $p' > p_0$ (resp. $p' < p_0$), then any Gittins index policy next searches box 1 (resp. 2), and, if the search fails, the searcher's belief that the hider is in box 1 will decrease (resp. increase). If $p' = p_0$, a Gittins index policy may search either box 1 or box 2 next; write p^i for the posterior probability that the hider is in box 1 after a single search of box i, $i = 1, 2$. Noted by both Norris (1962) and Roberts and Gittins (1978), it follows that if the hider plays any strategy $p \in (0, 1)$ and the searcher a Gittins index sequence against p, after some finite number of searches, the sequence of posterior hiding probabilities for box 1 will enter and henceforth remain in the interval $[p^1, p^2]$, coined the *recurrent region* or *recurrent interval.* Clearly $p_0 \in [p^1, p^2]$;

Norris (1962) shows that any optimal hiding strategy must also belong to the recurrent region.

By considering $(n-1)$-dimensional space, Norris (1962) generalises the recurrent region to an arbitrary number of boxes n. Note that, by definition of $\mathbf{p}_0$, the first n searches of a Gittins index sequence against $\mathbf{p}_0$ may be any permutation of $\{1, \ldots, n\}$. For any n, Norris (1962) shows that the recurrent region is generated in $(n-1)$-dimensional space by the $2^n - 2$ posterior hiding distributions that could be attained within the first $n-1$ searches if the searcher follows a Gittins index sequence against $\mathbf{p}_0$. For $n = 3$, the recurrent region is an irregular hexagon in 2-dimensional space; several examples are provided in Norris (1962).

By definition, if the hider plays $\mathbf{p}$ outside the recurrent region, then, if the searcher plays a Gittins index sequence against $\mathbf{p}$, after some number of searches a posterior $\mathbf{p}'$ will lie in the recurrent region. Gittins (1989) declares it is easy to show that $\mathbf{p}'$ is a better hiding strategy than $\mathbf{p}$ and hence that any optimal hiding strategy lies in the recurrent region, extending the same result of Norris (1962) for $n = 2$ to arbitrary n.

Recall that the hider wishes the long-term-average posterior hiding distribution to be close to $\mathbf{p}_0$. By following a Gittins index policy, the searcher ensures the sequence of posteriors remains in the recurrent region centred at $\mathbf{p}_0$. Gittins (1989) notes the paradox; despite being in direct conflict with the hider, any optimal counter of the searcher to a hiding strategy provides the hider with exactly what they want!

Calculating an Optimal Hiding Strategy By Blackwell and Girshick (1954), IBOX is guaranteed to have an optimal hiding strategy. Bram (1963) shows that, in unit-search-time IBOX, if $\mathbf{p}^* \equiv (p_1^*, \ldots, p_n^*)$ is an optimal hiding strategy, then $p_i^* > 0$ for $i = 1, \ldots, n$. In other words, every box is optimally hidden in with a non-zero probability; this result is extended to arbitrary search times in this thesis in Chapter 7.

Write $v(\mathbf{p})$ for the expected search time if the hider plays $\mathbf{p}$ and the searcher a

Gittins index sequence against $\mathbf{p}$; an optimal hiding strategy maximises $v(\mathbf{p})$. The expected search time under any search sequence is linear as a function of $\mathbf{p}$, so $v(\mathbf{p})$ is the infimum of an infinite number of linear functions and hence is concave. Therefore, any local maximum must be a global maximum. Bram (1963) shows that $v(\mathbf{p})$ is continuous and attains its maximum value v^*.

Recall from Part I that Matula (1964), studying the one-sided search, shows that if and only if $\log(1-q_i)/\log(1-q_j) \in \mathbb{Q}$ for all $i, j \in \{1, \ldots, n\}$ does there exist an optimal search sequence which, after an initial transient period, is periodic; in other words, some fixed, finite sequence of boxes, or *cycle* is repeated indefinitely. The condition of Matula (1964) is equivalent to

$$(1-q_1)^{x_1} = \cdots = (1-q_n)^{x_n} \tag{6.2.6}$$

for some coprime integers x_i, $i = 1, \ldots, n$, in which case any cycle contains x_i searches of box i for $i = 1, \ldots, n$. Under (6.2.6), Norris (1962) shows that $v(\mathbf{p})$ is piecewise linear on the recurrent region, a fact further discussed in Gittins (1989) for $n = 2$.

For any initial hiding distribution $\mathbf{p}$, after such a cycle of searches is made, the posterior hiding distribution is equal to $\mathbf{p}$. As a result, Matula (1964) claims that under (6.2.6) the optimal expected search time in the one-sided problem has a closed form; therefore, derived by Norris (1962), $v(\mathbf{p})$ has a closed form in IBOX under (6.2.6). Since any irrational number can be approximated arbitrarily well by a rational number, as pointed out by Norris (1962) and Roberts and Gittins (1978), any problem where (6.2.6) does not hold can be approximated arbitrarily well by one where (6.2.6) holds. Therefore, the closed for expression for $v(\mathbf{p})$ valid under (6.2.6) can be used to estimate $v(\mathbf{p})$ when (6.2.6) does not hold. Otherwise, for any problem, Gittins and Roberts (1979) provides an iterative method to calculate $v(\mathbf{p})$ to any desired accuracy. The method is presented for unit search times, but may easily be generalised to arbitrary search times. For two special cases, Gittins and Roberts (1979) studies properties of $v(\mathbf{p})$ which simplify its calculation. In the first case, boxes can be split into equal-sized groups where all boxes in each group are identical. In the second case, each box has either detection probability q or $\bar{q}$, which satisfy $1 - q = (1 - \bar{q})^m$ for some $m \in \mathbb{Z}^+$.

To find an optimal $\mathbf{p}^*$, Gittins and Roberts (1979) suggests, starting at $\mathbf{p}_0$, making incremental adjustments to hiding strategies and evaluating $v(\mathbf{p})$, proceeding in a direction in which $v(\mathbf{p})$ increases until no longer possible. Since $v(\mathbf{p})$ is concave, the local maximum found will be a global maximum. Gittins and Roberts (1979) adjusts the incremental procedure for the searcher to find a good heuristic strategy. Lin and Singham (2015) presents an iterative algorithm which estimates an optimal strategy for each player by successively calculating tighter bounds on v^*. Alongside the work in Chapter 7, the algorithm will be submitted for publication in 2020.

6.2.2 Extensions to IBOX

This section summarises extensions to IBOX in the literature.

A Reward-Cost Payoff To begin, instead of the total search time, the payoff to the hider in Norris (1962) is, in fact, total reward minus total cost, which allows several new features to be added to IBOX. The hider receives reward 1 for every unsuccessful search and can switch boxes between searches for a cost of μ. The cases $\mu = 0$ and $\mu = \infty$ are, respectively, equivalent to the mobile and immobile-hider versions of IBOX with unit search times discussed in Section 6.2.1. Norris (1962) also studies general μ. For two boxes, $\bar{\mu}$ is found such that the hider never switches boxes when $\mu \geq \bar{\mu}$. The value of the game is shown to decrease monotonically as μ increases from 0 (mobile-hider IBOX) to $\bar{\mu}$ (a game equivalent to IBOX). Norris (1962) notes that the n-box game for arbitrary μ becomes extremely complex. A case with three identical boxes is partially solved, and methods to approximate solutions in the general case are discussed.

Norris (1962) studies further generalisations to the reward-cost payoff structure. In addition to some boxes costing the searcher more to search, the hider may see more benefit remaining undetected in some boxes over others. Therefore, the reward obtained by the hider after an unsuccessful search is allowed to depend on both the box searched and the box where the hider is currently hidden. Further, the consequence of being found may vary from box to box; therefore, the hider incurs a box-dependent *detection cost* on being captured. Finally, the reward and cost are allowed to depend on the current length of the search via an exponentially-decaying discount factor. For $\mu = \infty$ (equivalent to an immobile hider) and general n, Chapter 8 of Norris (1962) shows that, given knowledge of the hider's strategy, an index policy similar to (6.2.1) is optimal for the searcher under the aforementioned extensions.

Neuts (1963) also considers a reward-cost payoff structure with exponential discounting. The hider receives a reward for each unsuccessful search, which, like Norris (1962), depends on the box searched but, unlike Norris (1962), not on the hider's location. Like Norris (1962), there is a detection cost to the hider incurred on their discovery but, unlike Norris (1962), the detection cost is not box dependent. Neuts (1963) first studies the case where the searcher is memoryless, in other words, not allowed to use the history of failed searches to guide their next search. In particular, the searcher must choose $\mathbf{y} \equiv (y_1, \ldots, y_n)$ at the start of the search process and always use $\mathbf{y}$ to select a box for the next search, where y_i is the probability of selecting box i, $i = 1, \ldots, n$. Therefore, like mobile-hider IBOX, Neuts' memoryless problem reduces to an easily-solved single-search game where both players have n pure strategies, each corresponding to a box to hide/search. In fact, with no detection cost or discounting, Neuts' problem is identical to mobile-hider IBOX. Neuts (1963) acknowledges that a memoryless searcher is not practical when the hider cannot move, but uses the memoryless solution to study a non-memoryless problem with an immobile hider where the objective is to minimise the Bayes' risk to the searcher.

However, Neuts' memoryless solution is not correct, being corrected in the no-discounting case in Sakaguchi (1973). With zero-detection cost, as expected, Sakaguchi's solution coincides with the mobile-hider IBOX solution in (6.2.3), (6.2.4) and (6.2.5) found by Norris (1962). Further, Sakaguchi (1973) shows the optimal hider and searcher strategies in (6.2.3) and (6.2.4), respectively, retain optimality when the detection cost is non-zero. Hence, the only change to the value in (6.2.5) is the subtraction of the detection cost. Sakaguchi (1973) extends the solution to allow discounting and box-dependent detection costs, showing that adjusted versions of (6.2.3) and (6.2.4) are optimal. In a model similar to Thomas and Washburn (1991) discussed in Section 6.1, Sakaguchi (1973) further introduces travel costs between boxes and a hider aware of the searcher's current box. Thomas and Washburn (1991), however, allows time-dependent search strategies unlike Sakaguchi (1973).

Searches in Continuous Time Roberts and Gittins (1978), Gittins and Roberts (1979) and Gittins (1989) consider a continuous-time version of IBOX. At any time t, the searcher may split their search effort across the n boxes; let $u_i(t)$ represent the effort in box i at time t, $\sum_{i=1}^{n} u_i(t) = 1$. An exponential detection function is assumed; specifically, if the hider is in box i, the searcher finds the hider by time t with probability $1 - \exp(-\lambda_i U_i(t))$, where

$$U_i(t) \equiv \int_{s=0}^{t} u_i(s)ds,$$

and λ_i is a known detection rate of box i. Gittins (1989) proves that the minimax theorem holds, the hider optimally hides in box i with probability

$$\frac{1/\lambda_i}{\sum_{j=1}^{n} 1/\lambda_j}, \tag{6.2.7}$$

and the searcher optimally sets $u_i(t)$ equal to (6.2.7) at any time t.

Since λ_i measures the appeal of box i to the searcher, as noted by Gittins (1989), hiding using (6.2.7) gives the searcher no preference of any box over another and may be thought of as the continuous-time version of the strategy $\mathbf{p}_0$ in (6.2.3). Whilst $\mathbf{p}_0$ is not always optimal in IBOX, (6.2.7) is always optimal in continuous-time IBOX. The reason is the following. The searcher's optimal strategy in continuous-time IBOX (recall also given by (6.2.7)) keeps the posterior hiding distribution at (6.2.7) throughout the whole search, so (6.2.7) never gives the searcher an advantage in searching one box over another. As discussed in Section 6.2.1, this situation is precisely what the hider wants. On the other hand, in IBOX, whilst any optimal search strategy tries to keep posterior hiding distributions as close to $\mathbf{p}_0$ as possible, approximations must suffice because of the discrete nature of the search. Therefore, some other hiding distribution may lead to better posterior approximations of $\mathbf{p}_0$ over the course of the search, and hence be optimal for the hider.

A Deadline Payoff with Unit Search Times IBOX with unit search times has been studied with a deadline $d \in \mathbb{Z}^+$, with the payoff to the hider 0 if detection occurs within the first d searches and 1 otherwise. Therefore, the searcher maximises the probability of finding the hider before d, a common objective in one-sided search problems discussed in Chapter 1 of Part I of this thesis.

Subelman (1981) finds optimal strategies for both players which depend on d, applicable when d is fixed and known to both players. In Lin and Singham (2016), d is not known by the searcher. Therefore, *uniformly optimal* search strategies, which, recall from Part I, are optimal for any choice of deadline, are of importance. Chew (1967) shows that a uniformly optimal search policy exists in the classic one-sided, time-independent problem with unit search times; Lin and Singham (2016) shows the same is true in the two-sided extension whether the hider knows the deadline or not. Further, a uniformly optimal hiding strategy is shown to exist if and only if all boxes are identical. In this case, the uniformly optimal hiding strategy is $\mathbf{p}_0$, namely, hide in each box with probability $1/n$. Recall that, when the boxes are identical, $\mathbf{p}_0$ is also optimal in regular IBOX (with payoff the total search time); Lin and Singham (2016) also shows that, when the boxes are identical, their uniformly optimal search strategy is optimal in regular IBOX.

Berry and Mensch (1986) adds both a deadline objective and imperfect detection to Dresher (1961), discussed in Section 6.1. A detection probability q is known to both players. Suppose the hider is in box j. Upon a search of any box i, with probability

q, the model is identical to Dresher (1961), namely, the hider is found if $i = j$, and otherwise the searcher is told whether $j < i$ or $j > i$. With probability $1 - q$, the searcher is told $j < i$ regardless of the truth. Suppose a search does not find the hider. If told $j > i$, the searcher knows this is the truth and can rule out boxes $\{1, \ldots, i\}$. Otherwise, the searcher is told $j < i$, but cannot rule out boxes $\{i+1, \ldots, n\}$ as $j < i$ could be a lie. Berry and Mensch (1986) solves the game for $n = 2$ and arbitrary deadline $d \in \mathbb{Z}^+$, and partially solves the game for arbitrary n and $d = 2$.

Nakai (1986b) considers a deadline objective in a problem with a mobile hider and infinite number of boxes ordered numerically. The hider starts in an arbitrary box. After each unsuccessful search, the searcher learns the position of the hider. The hider may then, unseen by the searcher, either move to a neighbouring box or not move at all. The detection probability on the next search depends on whether the hider just switched boxes or not, and not on the hider's current location. There is also a box 0 or 'home' which cannot be searched; once the hider reaches box 0 they will optimally stay there until the deadline. Box 0 is clearly attractive to the hider, but always moving towards box 0 is predictable behaviour. Nakai (1986b) solves the game using recursive methods. As expected, the nearer the hider is to box 0, the greater the probability they should head towards it.

Stewart (1981) considers a similar model to Nakai (1986b) with an unsearchable box 0 but only two boxes. The hider starts in box 2 with their position henceforth unknown to the searcher. The detection probability depends both on the box searched and on whether the hider is about to move. The hider receives payoff 1 if they reach box 0 undetected and payoff 0 otherwise. Stewart (1981) considers both a searcher making searches in discrete and in continuous time.

More recently, Gal and Casas (2014) considers a deadline version of IBOX where the hider is 'prey' and the searcher a 'predator'. The deadline $d \in \{1, \ldots, n\}$ is interpreted as the predator becoming tired and needing rest. The predator may not search a box more than once, with the detection probability interpreted as a 'capture probability'; i.e., the predator always sees the prey when searching the prey's box i, but capture only occurs with probability q_i since the prey will attempt to flee.

Without a loss of generality, suppose $q_1 \leq \cdots \leq q_n$. Recall $\mathbf{p}_0$ from (6.2.3), the unique hiding strategy which makes all n boxes equally attractive to the searcher in IBOX. Gal and Casas (2014) shows if $d < 1/p_{0,1}$, then $\mathbf{p}_0$ is optimal for the prey, and the predator optimally searches box i at some point before the deadline with probability $dp_{0,i}$, $i = 1, \ldots, n$. The value of the game, the probability of capture under these optimal strategies, is therefore

$$\sum_{i=1}^{n} p_{0,i} \times dp_{0,i} \times q_i = \frac{d}{\sum_{i=1}^{n} 1/q_i}. \tag{6.2.8}$$

Note that the prey can attain an expected payoff of at most q_i by always hiding in box i. If $d \geq 1/p_{0,1}$, then (6.2.8) exceeds q_1, so $\mathbf{p}_0$ is no longer optimal for the prey. Gal and Casas (2014) shows that, in this instance, the prey cannot do better than hiding in box 1 (the box with the best chance of fleeing successfully) with probability 1, so the value of the game is q_1. Clearly any optimal strategy for the predator always searches box 1.

Gal et al. (2015) introduces a known, persistence probability β to the model of Gal and Casas (2014). If the predator searches the prey's box before the deadline but fails to capture it, the predator either gives up with probability $1 - \beta$ and the game ends with no capture, or the predator persists with probability β. If the predator persists, the game restarts with the prey allowed to relocate to a new box. Gal et al. (2015) solves the game for arbitrary β.

As in Gal and Casas (2014), if the deadline exceeds a threshold, the prey optimally always hides in the box with the best chance of a successful flee, and otherwise optimally makes all boxes equally attractive to the predator using a mixed strategy $\mathbf{p}^*$. However, in Gal et al. (2015), $\mathbf{p}^* \neq \mathbf{p}_0$ and depends upon β. Further, the threshold is an increasing function of β; therefore, the more likely the predator to persist, the more appealing equating the attractiveness of all boxes using $\mathbf{p}_0$ is to the prey. Extensions in Alpern et al. (2019) allow a chance of the prey being caught whilst relocating, and the detection/capture probabilities to be unknown to both players and learnt as the search progresses.

Alternative Roles of the Players Several versions of IBOX have been considered where the second player is not a hider.

Suppose a target is hidden among n boxes according to a target distribution known to both players. In Croucher (1975), the second player is a protector, who wishes the target to remain hidden. Both the searcher and protector allocate effort to each box, with, if the target is located there, the more (resp. less) effort allocated to a box by the searcher (resp. protector), the greater the probability of the detection of the target. Both players have a finite amount of effort to allocate, and the expected payoff is the probability that the target is found after both players have used all their available effort. Baston and Garnaev (2000) extend the model of Croucher (1975) to incorporate a player-dependent cost of allocating effort into each player's objective, which makes the game non zero sum. Further, the addition of such a cost means each player may no longer wish to allocate all their effort.

In Nakai (1986a), the second player is another searcher. Both searchers aim to find the target before the other, and both believe the target to be hidden according to a different target distribution. As in the continuous-time IBOX model of Gittins (1989), search effort is allocated in continuous time with an exponential detection function. In Nakai (1990), each searcher aims to find their own target before the other searcher. Any effort allocated in box by one searcher disrupts the effectiveness of effort allocated in that box by the other searcher.

Chapter 7

A Search Game in Discrete Locations

This chapter concerns IBOX, the search game in n discrete locations (boxes) with an immobile hider. As discussed in Section 6.2.1, there are various studies of IBOX in the literature, but their results are limited to two boxes or unit search times. Here, with novel proof techniques, we extend several such results to IBOX in its full generality, uncovering various new properties along the way. As a result, the previously-fragmented theory of IBOX is both collated and enhanced.

In Section 7.1, we formulate IBOX. As with all search games discussed in Chapter 6, since there are two players in direct conflict, IBOX is a two-person, zero-sum game introduced in Chapter 5. Specifically, since there are an infinite number of pure strategies for the searcher, IBOX is a semi-infinite game discussed in Section 5.2, for which general theory does not guarantee the existence of an optimal search strategy. Bram (1963) proves an optimal search strategy exists in unit-search-time IBOX; Section 7.2 extends Bram's result to general IBOX.

In Section 7.3, several properties of optimal strategies for each player are presented, including the generalisation of a two-box, unit-search-time result of Gittins (1989) showing the existence of a simple optimal search strategy among the many available.

The two-box proof method of Gittins does not adequately extend to an arbitrary number of boxes, so an alternative approach is taken. Further, we develop a practical test determining whether a hiding strategy is optimal or not by solving a known finite two-person, zero-sum game. In a numerical study, Gittins and Roberts (1979) investigates the frequency of the optimality of a particular hiding strategy (introduced as $\mathbf{p}_0$ in Section 6.2.1) that gives the searcher no preference over which box to search first. In Section 7.4, using the aforementioned optimality test, we extend the numerical study of Gittins and Roberts (1979) to a much wider range of problems split into subgroups, with comparisons between subgroups linked to Part I of this thesis. Section 7.5 derives some additional properties of an optimal hiding strategy.

7.1 Model and Preliminaries

Consider a two-person zero-sum *search game* G as follows. A hider decides where to hide among n boxes labelled $1, \ldots, n$, and a searcher decides an ordered sequence of boxes to search. A search in box i takes time t_i and will find the hider with detection probability q_i, $i = 1, \ldots, n$, if the hider is indeed hidden there. These quantities are common knowledge to both players. The payoff to both players is the time at which the hider is detected. Therefore, the expected payoff is the *expected search time* which the searcher aims to minimise and the hider to maximise.

The hider's pure strategy space is $\{1, \ldots, n\}$, where each pure strategy corresponds to a box to hide. A mixed hiding strategy is a probability vector $\mathbf{p} \equiv (p_1, \ldots, p_n)$ such that the hider hides in box i with probability p_i, where $0 \le p_i \le 1$ for $i = 1, \ldots, n$, and $\sum_{i=1}^{n} p_i = 1$. The searcher's pure strategy space is $\mathcal{S} \equiv \{1, 2, \ldots, n\}^\infty$. Each pure strategy is a *search sequence*—an infinite, ordered list of boxes to search until the hider is found. A mixed search strategy is a function η with domain $\mathcal{S}$ which satisfies the following conditions.

1. The set $\{\xi \in \mathcal{S} : \eta(\xi) > 0\}$ is countable.

2. $\sum_{\xi\in\mathcal{S}}\eta(\xi)=1$.

Under strategy η, the searcher plays $\xi\in\mathcal{S}$ with probability $\eta(\xi)$.

For a search sequence $\xi\in\mathcal{S}$, write $V_i(\xi)$ for the expected search time if the hider hides in box i, for $i=1,\ldots,n$. In other words, $V_i(\xi)$ is the expected payoff of the hider-searcher strategy pair (i,ξ). While the hider's pure strategy space is of size n, the searcher's pure strategy space $\mathcal{S}$ is uncountable, because it is the Cartesian product of a countable number of $\{1,\ldots,n\}$. Therefore, G is a two-person zero-sum semi-infinite game discussed in Section 5.2.

The hider seeks a mixed strategy to guarantee the highest possible expected search time regardless of what the searcher does, so he seeks to determine

$$v_1\equiv\max_{\mathbf{p}}\inf_{\xi\in S}\sum_{i=1}^{n}p_iV_i(\xi). \tag{7.1.1}$$

Likewise, the searcher seeks to determine

$$v_2\equiv\inf_{\eta}\max_{i\in\{1,\ldots,n\}}\int V_i(\xi)\eta(\xi)d\xi.$$

It is clear that $v_1\le v_2$. As discussed in Section 5.2, because the payoff function—namely the length of the search—is bounded below by 0, it follows from Theorem 2.4.2 of Blackwell and Girshick (1954) that $v_1=v_2$, which is the value of G, written by v^*. In addition, by the same Theorem 2.4.2, the hider has an optimal strategy that guarantees an expected search time of at least v^*, and the searcher has an ϵ-optimal strategy; that is, for an arbitrarily small $\epsilon>0$, the searcher can find a strategy to guarantee an expected search time of at most $v^*+\epsilon$. The next section makes further appeal to Blackwell and Girshick (1954) to show that the searcher has an optimal strategy.

7.2 Existence of Optimal Search Strategies

The aim of this section is to prove that the searcher has an optimal strategy in the search game G. We begin by reformulating G as an S-game studied in Blackwell and

Girshick (1954) and previously discussed in Section 5.2 of this thesis. In the S-game, instead of choosing a pure strategy in $\mathcal{S}$, the searcher chooses a vector in the set

$$S \equiv \{(V_1(\xi), \ldots, V_n(\xi)) : \xi \in \mathcal{S}\} \subset \mathbb{R}^n.$$

If the hider hides in box $i \in \{1, \ldots, n\}$ and the searcher selects $(V_1(\xi), \ldots, V_n(\xi)) \in S$, then the expected payoff is $V_i(\xi)$.

By Theorem 2.4.1 of Blackwell and Girshick (1954), the searcher selecting a mixed strategy is equivalent to choosing a point in S^*, the convex hull of S. By Theorem 2.4.2, if S, or equivalently S^*, is closed, then there exists an optimal search strategy.

The intuition behind this result is the following, adapted from Chapter 13 of Ferguson (2020). If $\mathbf{s} \equiv (s_1, \ldots, s_n) \in S^*$, then there exists a mixed search strategy which, if the hider hides in box i, achieves an expected payoff s_i, $i = 1, \ldots, n$. It follows that the value of the game v^* satisfies

$$v^* = \inf \left\{ \max_{i \in \{1, \ldots, n\}} s_i : \mathbf{s} \in S^* \right\}. \tag{7.2.1}$$

If S^* is closed, then the infimum in (7.2.1) is attained, so there exists $\mathbf{s}^* \in S^*$ with $\max_{i \in \{1, \ldots, n\}} s_i^* = v^*$; it follows that $\mathbf{s}^*$ is an optimal search strategy. See Ruckle (1991) for more on the geometrical interpretation of optimal strategies in the search game, particularly for $n = 2$.

Bram (1963) concluded that an optimal search strategy exists for the search game with $t_i = 1$ for $i = 1, \ldots, n$ by showing that S^* is closed. In this section, we take a different approach to extend the result to arbitrary $t_i > 0$, $i = 1, \ldots, n$.

7.2.1 Preliminary Properties of Optimal Strategies

Recall that a pure search strategy is a search sequence—an infinite, ordered list of boxes. We begin by defining a particular type of search sequence prominent throughout Part I and also discussed in Chapter 6 of this thesis.

Definition 7.2.1 A *Gittins index sequence* against a mixed hiding strategy $\mathbf{p} \equiv (p_1, \ldots, p_n)$ is a search sequence constructed with the following rule. If $m_i \in \{1, 2, \ldots\}$ searches have already made of box i during the search process, $i = 1, \ldots, n$, then the next search is of any box j satisfying

$$j = \arg\max_{i=1,\ldots,n} \frac{p_i(1-q_i)^{m_i} q_i}{t_i}. \tag{7.2.2}$$

The terms in (7.2.2) are known as Gittins indices, and a Gittins index sequence is a search sequence that always searches a box with a maximal Gittins index. There may be multiple Gittins index sequences against the same hiding strategy $\mathbf{p}$ due to ties for the maximum in (7.2.2), so we write $C(\mathbf{p}) \subset \mathcal{S}$ for the set of Gittins index sequences against $\mathbf{p}$.

As discussed in Part I, if the searcher *knows* that the hider will choose a mixed strategy $\mathbf{p}$, by the work of several authors in the 1960s (Norris (1962), Bram (1963), Blackwell (reported in Matula (1964)), Black (1965)), any Gittins index sequence against $\mathbf{p}$ is an optimal counter strategy for the searcher. This result is also recognised by a comment by Kelly on Gittins (1979), which formulates the search game with fixed $\mathbf{p}$ as a multi-armed bandit problem optimally solved by Gittins indices, from where the sequences in $C(\mathbf{p})$ take their name. Therefore, we say a Gittins index sequence against $\mathbf{p}$ is an *optimal counter* to $\mathbf{p}$, and $C(\mathbf{p})$ is the set of optimal counters to $\mathbf{p}$.

Recall that a mixed search strategy plays a countable number of search sequences with non-zero probability; we next introduce the following terminology.

Definition 7.2.2 We say a mixed search strategy η is a *mixture* of the countable set of pure search strategies $\{\xi \in \mathcal{S} : \eta(\xi) > 0\}$, and write $V_i(\eta)$ for the expected time to detection under η if the hider hides in box i, $i = 1, \ldots, n$.

For any $\xi \in \mathcal{S}$, as a function of $\mathbf{p}$, $\sum_{i=1}^{n} p_i V_i(\xi)$ is a hyperplane in n-dimensional space. Combined with (7.1.1), it follows that v^* is the maximum of the lower envelope of an uncountable set of hyperplanes, which is a concave function of $\mathbf{p}$. Let $\mathcal{P}^*$ be the set of $\mathbf{p}$ obtaining this maximum, so $\mathcal{P}^*$ is the set of optimal hiding strategies.

In most cases, $|\mathcal{P}^*| = 1$ and the optimal hiding strategy is unique; an example with $|\mathcal{P}^*| > 1$ can be found in Example 7.3.2.

Since the hider maximises a lower envelope of hyperplanes, there must exist at least one hyperplane containing the point $(\mathbf{p}^*, v^*)$ for all $\mathbf{p}^* \in \mathcal{P}^*$. In other words, there exists at least one search sequence which optimally counters every optimal hiding strategy. The following proposition states that an optimal search strategy, if it exists, can only mix such search sequences.

Proposition 7.2.3 Any optimal search strategy η^* is a mixture of some subset of $\cap_{\mathbf{p}\in\mathcal{P}^*} C(\mathbf{p})$. In other words, the searcher will use only search sequences that are Gittins index sequences against every $\mathbf{p} \in \mathcal{P}^*$.

Proof. Since the search game G has a value v^*, if the searcher chooses η^* and the hider any $\mathbf{p}^* \equiv (p_1^*, \ldots, p_n^*) \in \mathcal{P}^*$, the expected time until detection is v^*; in other words, we have

$$v^* = \int \eta^*(\xi) \left(\sum_{i=1}^{n} p_i^* V_i(\xi) \right) d\xi. \tag{7.2.3}$$

Suppose $\eta^*(\bar{\xi}) > 0$, and there exists $\bar{\mathbf{p}} = (\bar{p}_1, \ldots, \bar{p}_n) \in \mathcal{P}^*$ for which $\bar{\xi} \notin C(\bar{\mathbf{p}})$. Then $\sum_{i=1}^{n} \bar{p}_i V_i(\bar{\xi}) > v^*$. Since $\bar{\mathbf{p}}$ is optimal for the hider, $\sum_{i=1}^{n} \bar{p}_i V_i(\xi) \geq v^*$ for all $\xi \in \mathcal{S}$; therefore, (7.2.3) cannot hold with $\mathbf{p}^* = \bar{\mathbf{p}}$, contradicting the optimality of η^*. ■

It is intuitive that adding a new box will increase the value of the game, because the hider has one more place to hide, so the searcher needs to cover more ground, as seen in the next proposition.

Proposition 7.2.4 The following are true.

1. If $\mathbf{p}^*$ is optimal for the hider, then $p_i^* > 0$ for $i = 1, \ldots, n$.

2. If η^* is optimal for the searcher, then $V_i(\eta^*) = v^*$ for $i = 1, \ldots, n$.

3. Adding a new box increases the value of the game.

Note that (1.) is also proved by Bram (1963) for unit-search-time G via a different method to the proof below.

Proof. We begin by proving (1.), which concerns the hider. Let $v(\mathbf{p})$ be the expected search time when the hider chooses $\mathbf{p}$, and the searcher chooses any search sequence in $C(\mathbf{p})$; therefore, any $\mathbf{p}^* \in \mathcal{P}^*$ maximises $v(\mathbf{p})$. Let $\xi_{\mathbf{p}}$ be the element of $C(\mathbf{p})$ which, when multiple boxes satisfy (7.2.2), searches the box with the smallest label.

Suppose, aiming for a contradiction, that we have $\bar{\mathbf{p}} = (\bar{p}_1, \ldots, \bar{p}_n) \in \mathcal{P}^*$ with $\bar{p}_k = 0$ for some $k \in \{1, \ldots, n\}$. Without loss of generality, relabel the boxes so that we have $\bar{p}_n = 0$. Then

$$v(\bar{\mathbf{p}}) = \sum_{i=1}^{n-1} \bar{p}_i V_i(\xi_{\bar{\mathbf{p}}}). \tag{7.2.4}$$

Take any $\epsilon > 0$ and set

$$p_i = \bar{p}_i(1-\epsilon) \quad i = 1, \ldots, n-1, \quad p_n = \epsilon, \quad \mathbf{p} = (p_1, \ldots, p_n).$$

Compare $\xi_{\mathbf{p}} \in C(\mathbf{p})$ and $\xi_{\bar{\mathbf{p}}} \in C(\bar{\mathbf{p}})$. Both apply the same rule when multiple boxes satisfy (7.2.2), and for any $i, j \in \{1, \ldots, n-1\}$, we have

$$\frac{p_i}{p_j} = \frac{\bar{p}_i(1-\epsilon)}{\bar{p}_j(1-\epsilon)} = \frac{\bar{p}_i}{\bar{p}_j}.$$

Therefore, the subsequence formed of only searches of boxes i and j will be the same for $\xi_{\mathbf{p}}$ as for $\xi_{\bar{\mathbf{p}}}$ for any $i, j \in \{1, \ldots, n-1\}$. It follows that $\xi_{\mathbf{p}}$ is just $\xi_{\bar{\mathbf{p}}}$ with searches of box n inserted between some searches of the first $n-1$ boxes. Hence, we must have $V_i(\xi_{\mathbf{p}}) > V_i(\xi_{\bar{\mathbf{p}}})$ for any $i \in \{1, \ldots, n-1\}$ with $\bar{p}_i > 0$. Further, we may choose ϵ small enough so that $\xi_{\mathbf{p}}$ does not search box n until at least $v(\bar{\mathbf{p}})$ time units have passed, so $V_n(\xi_{\mathbf{p}}) > v(\bar{\mathbf{p}})$. From these observations and (7.2.4), it follows that

$$\begin{aligned} v(\mathbf{p}) &= \epsilon V_n(\xi_{\mathbf{p}}) + \sum_{i=1}^{n-1} \bar{p}_i(1-\epsilon) V_i(\xi_{\mathbf{p}}) \\ &> \epsilon v(\bar{\mathbf{p}}) + (1-\epsilon) \sum_{i=1}^{n-1} \bar{p}_i V_i(\xi_{\bar{\mathbf{p}}}) = v(\bar{\mathbf{p}}), \end{aligned}$$

contradicting the optimality of $\bar{\mathbf{p}}$, and therefore proving the hider's part of the result.

Next, we prove (2.), concerning the searcher. Suppose η^* is optimal for the searcher and $\mathbf{p}^*$ is optimal for the hider. Then the expected search time under the strategy pair $(\mathbf{p}^*, \eta^*)$ is

$$v^* = \sum_{i=1}^{n} p_i^* V_i(\eta^*). \tag{7.2.5}$$

Suppose that $V_j(\eta^*) < v^*$ for some $j \in \{1, \ldots, n\}$. By (7.2.5), there must either exist $k \in \{1, \ldots, n\}$ such that $V_k(\eta^*) > v^*$, or we must have $p_j^* = 0$. The former cannot happen as η^* guarantees the searcher an expected search time of at most v^*. The latter cannot happen by the hider's part of the result, leading to a contradiction which proves the searcher's part of the result.

Finally, we prove (3.) by showing that $v_{n+1}^* > v_n^*$, where v_n^* is the value of an n-box game, and v_{n+1}^* is the value if a new box is added to the n-box game. In the game with $n+1$ boxes, the hider can guarantee an expected payoff of at least v_n^* by not hiding in the new box, so $v_{n+1}^* \geq v_n^*$. However, any such strategy has $p_{n+1} = 0$, so is not optimal by the hider's part of the result. Therefore, v_n^* is not the value of the $(n+1)$-box game, so $v_{n+1}^* > v_n^*$. ■

7.2.2 An Equivalent Search Game

In this section, we first show that an optimal search strategy exists in a modified version of the search game G. We then draw the same conclusion for G by showing that an optimal search strategy in the modified game is also optimal in G.

Consider a search game $G(\epsilon)$, parametrized by $\epsilon \in (0, 1/n)$, identical to G in all aspects apart from the set of pure search strategies, which are constructed by the following. For $i = 1, \ldots, n$, write

$$M_i(\epsilon) \equiv \inf\{V_i(\xi) : \xi \in C(\mathbf{p}) \text{ with } p_i < \epsilon\}. \tag{7.2.6}$$

In other words, among all Gittins index sequences against hiding strategies with $p_i < \epsilon$, $M_i(\epsilon)$ is the smallest expected search time if the hider is in box i. In $G(\epsilon)$, for $i = 1, \ldots, n$, a pure strategy $\zeta_i(\epsilon)$ is available to the searcher. When elected, for

$i = 1, \ldots, n$, $\zeta_i(\epsilon)$ results in payoff $M_i(\epsilon)$ if the hider is in box i or payoff 0 otherwise. In addition, available to the searcher in $G(\epsilon)$ are Gittins index sequences against hiding strategies where each box is hidden in with a probability of at least ϵ. In other words, hiding strategies in the set

$$\mathcal{P}(\epsilon) \equiv \{\mathbf{p} : p_i \geq \epsilon, i = 1, \ldots, n\}. \tag{7.2.7}$$

To summarise, in $G(\epsilon)$, the searcher has the following pure strategy set:

$$\mathcal{S}(\epsilon) \equiv \{\xi \in C(\mathbf{p}) : \mathbf{p} \in \mathcal{P}(\epsilon)\} \cup \{\zeta_i(\epsilon) : i = 1, \ldots, n\}. \tag{7.2.8}$$

For any $\epsilon \in (0, 1/n)$, since the payoff in $G(\epsilon)$ is bounded below by 0, we may apply Theorem 2.4.2 of Blackwell and Girshick (1954) to conclude that $G(\epsilon)$ has a value and optimal hiding strategy. To prove that $G(\epsilon)$ has an optimal search strategy, we consider its S-game formulation, in which the searcher chooses a vector in

$$S(\epsilon) \equiv \{(V_1(\xi), \ldots, V_n(\xi)) : \xi \in \mathcal{S}(\epsilon)\} \subset \mathbb{R}^n.$$

The following lemma is proven in Appendix C.1 by showing that $S(\epsilon)$ is closed and applying Theorem 2.4.2 of Blackwell and Girshick (1954).

Lemma 7.2.5 For any $\epsilon \in (0, 1/n)$, the game $G(\epsilon)$ has an optimal search strategy which is a mixture of at most n pure search strategies.

The following result draws upon the properties in Section 7.2.1 to conclude that, for small enough ϵ, the games G and $G(\epsilon)$ are almost equivalent.

Lemma 7.2.6 Consider G and its set of optimal hiding strategies $\mathcal{P}^*$. For any $\mathbf{p}^* \in \mathcal{P}^*$, there exists $\epsilon_{\mathbf{p}^*} \in (0, 1/n]$ such that, for all $\epsilon \in (0, \epsilon_{\mathbf{p}^*})$, the two games G and $G(\epsilon)$ share the same value. In addition, $\mathbf{p}^*$ is still optimal in $G(\epsilon)$, and a search strategy is optimal in G if and only if it is optimal in $G(\epsilon)$.

Proof. First, by Theorem 2.4.2 of Blackwell and Girshick (1954), both G and $G(\epsilon)$ have a value and an optimal hiding strategy.

Let $\mathbf{p}^* \equiv (p_1^*, \ldots, p_n^*) \in \mathcal{P}^*$ and $\epsilon_1 \equiv \min_{i \in \{1,\ldots,n\}} p_i^*$; we have $\epsilon_1 > 0$ by the hider's part of Proposition 7.2.4. Further, under any mixed hiding strategy, some box is chosen with at most probability $1/n$, so $\epsilon_1 \leq 1/n$.

As a function of ϵ, M_i in (7.2.6) is clearly decreasing, as the set over which the infimum is taken grows with ϵ. Write $\mathbf{p} \equiv (p_1, \ldots, p_n)$. If $p_i = 0$, then any $\xi \in C(\mathbf{p})$ never searches box i, so $V_i(\xi)$ is infinite, and hence $M_i(\epsilon) \uparrow \infty$ as $\epsilon \downarrow 0$. On the other hand, if $p_i = 1$, then any $\xi \in C(\mathbf{p})$ only searches box i, so $V_i(\xi) = t_i/q_i$, and hence $M_i(\epsilon) \downarrow t_i/q_i \leq v^*$ as $\epsilon \uparrow 1$, where v^* is the value of G. Combining the above information, we may conclude that

$$\epsilon_2 \equiv \sup\{\epsilon : M_i(\epsilon) > v^*/p_i^*,\ i = 1, \ldots, n\}$$

exists, and $M_i(\epsilon) > v^*/p_i^*$ for all $\epsilon \in (0, \epsilon_2)$, $i = 1, \ldots, n$.

Let $\epsilon_{\mathbf{p}^*} \equiv \min(\epsilon_1, \epsilon_2)$; we show that $\epsilon_{\mathbf{p}^*}$ satisfies the conditions of the lemma. Write $\mathcal{Y}$ for a set of pure search strategies, and define

$$v(\mathbf{p}, \mathcal{Y}) = \inf_{\xi \in \mathcal{Y}} v(\mathbf{p}, \xi),$$

where $v(\mathbf{p}, \xi)$ is the expected search time if the hider chooses $\mathbf{p}$ and the searcher chooses ξ. Throughout the following, let $\epsilon \in (0, \epsilon_{\mathbf{p}^*})$.

In G, a hiding strategy $\mathbf{p}$ is optimally countered by any sequence in $C(\mathbf{p})$, leading to an expected search time of $v(\mathbf{p}, \mathcal{S})$. Bearing the above in mind, we show that $v(\mathbf{p}, \mathcal{S}) \geq v(\mathbf{p}, \mathcal{S}(\epsilon))$ for any hiding strategy $\mathbf{p}$ by considering two cases.

1. Suppose that $\mathbf{p} \in \mathcal{P}(\epsilon)$. Then $C(\mathbf{p})$ is contained in $\mathcal{S}(\epsilon)$; therefore, if the hider chooses $\mathbf{p}$, the searcher does no worse when $\mathcal{S}$ is replaced with $\mathcal{S}(\epsilon)$, so $v(\mathbf{p}, \mathcal{S}) \geq v(\mathbf{p}, \mathcal{S}(\epsilon))$.

2. Suppose that $\mathbf{p} \notin \mathcal{P}(\epsilon)$. Then there exists $j \in \{1, \ldots, n\}$ such that $p_j < \epsilon$. By the construction of $\zeta_j(\epsilon)$, we have $V_i(\zeta_j(\epsilon)) \leq V_i(\xi)$ for any $\xi \in C(\mathbf{p})$, $i = 1, \ldots, n$. Therefore, $\zeta_j(\epsilon) \in \mathcal{S}(\epsilon) \setminus \mathcal{S}$ dominates any sequence in $C(\mathbf{p})$. It follows that $v(\mathbf{p}, \mathcal{S}) \geq v(\mathbf{p}, \mathcal{S}(\epsilon))$.

Now consider $v(\mathbf{p}^*, \mathcal{S}(\epsilon))$; by the above, $v^* = v(\mathbf{p}^*, \mathcal{S}) \geq v(\mathbf{p}^*, \mathcal{S}(\epsilon))$. Since $\epsilon < \epsilon_1$, we have $\mathbf{p}^* \in \mathcal{P}(\epsilon)$, and hence $C(\mathbf{p}^*) \subset \mathcal{S}(\epsilon)$. The only pure search strategies in $\mathcal{S}(\epsilon)$ that, when the hider chooses $\mathbf{p}^*$, could achieve a lower expected search time than a sequence in $C(\mathbf{p}^*)$ are those not in $\mathcal{S}$, namely $\{\zeta_i(\epsilon), i = 1, \ldots, n\}$. Therefore, we have

$$\begin{aligned} v(\mathbf{p}^*, \mathcal{S}(\epsilon)) &= \min\left(v^*, \min_{i \in \{1,\ldots,n\}} \left\{\sum_{j=1}^{n} p_j^* V_j(\zeta_i(\epsilon))\right\}\right) \\ &= \min\left(v^*, \min_{i \in \{1,\ldots,n\}} p_i^* M_i(\epsilon)\right) = v^*, \end{aligned}$$

where the final equality holds since $\epsilon < \epsilon_2$.

To conclude, we have $v(\mathbf{p}, \mathcal{S}) \geq v(\mathbf{p}, \mathcal{S}(\epsilon))$ for all hiding strategies $\mathbf{p}$, and $v^* = v(\mathbf{p}^*, \mathcal{S}) = v(\mathbf{p}^*, \mathcal{S}(\epsilon))$. It follows that $\mathbf{p}^*$ is optimal in $G(\epsilon)$ and the value of $G(\epsilon)$ is v^*.

As for the searcher, since the pure hiding strategies are the same for G and $G(\epsilon)$, any optimal search strategy in G is optimal in $G(\epsilon)$ if it is available to the searcher in $G(\epsilon)$ and vice versa. By Proposition 7.2.3, any optimal search strategy in G chooses only search sequences in $C(\mathbf{p}^*)$, available in $G(\epsilon)$ since $C(\mathbf{p}^*) \subset \mathcal{S}(\epsilon)$. Further, for $i = 1, \ldots, n$, we have $p_i^* M_i(\epsilon) > v^*$. Therefore, it is suboptimal in $G(\epsilon)$ for the searcher to choose any strategy in $\mathcal{S}(\epsilon) \setminus \mathcal{S} = \{\zeta_i(\epsilon), i = 1, \ldots, n\}$. It follows that a search strategy is optimal in G if and only if it is optimal in $G(\epsilon)$, completing the proof. ■

We conclude this section with its main result.

Theorem 7.2.7 In the search game G, for any optimal hiding strategy $\mathbf{p}^*$, there exists an optimal search strategy which is a mixture of at most n elements of $C(\mathbf{p}^*)$.

Proof. By Lemma 7.2.5, there exists a search strategy η^*, optimal in $G(\epsilon)$, which is a mixture of at most n pure search strategies. By Lemma 7.2.6, η^* is also optimal in the search game G. By Proposition 7.2.3, for any optimal hiding strategy $\mathbf{p}^*$, the n pure strategies mixed by η^* must all belong to $C(\mathbf{p}^*)$, completing the proof. ■

7.3 Properties of Optimal Strategies

While we have shown that each player has an optimal strategy in the search game G, it turns out that each player's optimal strategy need not be unique. Section 7.3.1 demonstrates how to identify an optimal hider-searcher strategy pair and presents an example where the hider has multiple optimal strategies. Section 7.3.2 extends a two-box result of Gittins (1989) to show that the searcher may always choose a simple optimal strategy among the many available. An alternative proof method to Gittins (1989) is used; we explain why Gittins' method does not adequately extend to n boxes. Section 7.3.3 develops a practical test of the optimality of any hiding strategy.

7.3.1 An Optimal Hider-Searcher Strategy Pair

We begin by combining Propositions 7.2.3 and 7.2.4 to identify simple conditions on a hider-searcher strategy pair which are both necessary and sufficient for optimality. The backwards implication is Theorem 8.3 of Gittins (1989) applied to the search game.

Theorem 7.3.1 Let v be the expected search time when the hider chooses some mixed strategy $\mathbf{p}$ and the searcher some mixed strategy η. Then $\mathbf{p}$ (resp. η) is optimal for the hider (resp. searcher) if and only if

1. The strategy η is a mixture of some subset of $C(\mathbf{p})$.

2. $V_i(\eta) = v$, for $i = 1, \ldots, n$.

Proof. First, we prove the forwards implication. If $\mathbf{p}$ is optimal for the hider and η is optimal for the searcher, then (1.) follows from Proposition 7.2.3. Further, if $\mathbf{p}$ and η are optimal, then v is the value of the game, so (2.) follows from the searcher's part of Proposition 7.2.4.

Second, we prove the backwards implication. By (1.), $\mathbf{p}$ guarantees an expected search time of at least v regardless of what the searcher does. In addition, by (2.), η guarantees an expected search time equal to v regardless of what the hider does. Therefore, neither the hider nor the searcher can obtain a better guarantee than v; it follows that $\mathbf{p}$ and η are an optimal strategy pair. ■

Theorem 7.3.1 shows that any optimal search strategy η^* is an *equalising* strategy; in other words, whenever the searcher plays η^*, the expected search time is the same no matter where the hider hides. Theorem 5.2 of Ruckle (1991) shows that, in the search game with unit search times, there always exists an equalising pure search strategy ξ. However, ξ is not necessarily a Gittins index sequence against any hiding strategy, so Theorem 7.3.1 does not necessarily apply to determine that ξ is optimal for the searcher. Further, the proof of Theorem 5.2 of Ruckle (1991) does not extend to the search game with arbitrary search times, as, for any three boxes i, j and k, it relies on $V_k(\xi)$ being unaffected when the positions of a search of box i and box j in ξ are switched.

We next use Theorem 7.3.1 to demonstrate that it is possible for the hider to have multiple optimal strategies.

Example 7.3.2 Consider a 2-box search game, where box i has search time t_i and detection probability q_i, $i = 1, 2$, with $q_1 < q_2$ and $t_1 > t_2$. Write p for the probability that the hider hides in box 1. Inspection of (7.2.2) shows if

$$p \in \left[\frac{q_2/t_2}{q_2/t_2 + q_1/t_1}, \frac{q_2/t_2}{q_2/t_2 + q_1(1-q_1)/t_1} \right], \tag{7.3.1}$$

then there exists a Gittins index sequence against p that begins by searching box 1, followed by box 2, and then box 1 again.

Suppose that $(1-q_2) = (1-q_1)^2$. For any p, if the searcher makes, in any order, two unsuccessful searches of box 1 and one unsuccessful search of box 2, then the posterior probability that the hider is in box 1 returns to p, and hence the problem has reset itself. Therefore, the sequence ξ that repeats the cycle of boxes $1, 2, 1$ indefinitely is a Gittins index sequence against any p satisfying (7.3.1).

Further, since $(1-q_2) = (1-q_1)^2$, the contribution of the kth cycle to $V_1(\xi)-V_2(\xi)$ is

$$(1-q_2)^{k-1}\left[q_1(t_1+(1-q_1)(2t_1+t_2)) - q_2(t_1+t_2)\right],$$

so we have

$$V_1(\xi)-V_2(\xi) = \left[q_1(t_1+(1-q_1)(2t_1+t_2)) - q_2(t_1+t_2)\right]\sum_{k=1}^{\infty}(1-q_2)^{k-1},$$

from which it follows that $V_1(\xi) = V_2(\xi)$ if and only if

$$q_1 = \frac{t_1-t_2}{t_1}. \tag{7.3.2}$$

Under (7.3.2), by Theorem 7.3.1, it follows that ξ is optimal for the searcher and any p satisfying (7.3.1) is optimal for the hider.

For a numerical example, if $q_1 = 0.4$, $q_2 = 0.64$, $t_1 = 1$ and $t_2 = 0.6$, then any $p \in [8/11, 40/49]$ is optimal for the hider.

7.3.2 Tie-Breaking in an Optimal Search Strategy

By Theorem 7.3.1, it is sufficient for the searcher to consider Gittins index sequences against any optimal hiding strategy. Therefore, if there exists an optimal $\mathbf{p}$ against which there is a unique Gittins index sequence ξ (so $|C(\mathbf{p})| = 1$), then the pure strategy ξ is optimal for the searcher. One instance can be found in Example 7.3.2; aside from the two endpoints, any p satisfying (7.3.1) is both optimal for the hider and has $C(p) = \{\xi\}$. The pure strategy ξ is optimal for the searcher. Further, Ruckle (1991) shows that in the search game with $q_i = 0.5$ and $t_i = 1$, $i = 1, \ldots, n$, the pure strategy $1, 2, \ldots, n, n-1 \ldots, 1, 2, \ldots$ is optimal for the searcher.

We now focus on the more challenging case where all optimal $\mathbf{p}$ satisfy $|C(\mathbf{p})| > 1$, and the searcher's optimal strategy may be a mixture of Gittins index sequences.

By Definition 7.2.1, the next box searched by any Gittins index sequence against a hiding strategy $\mathbf{p}$ must satisfy (7.2.2). If $|C(\mathbf{p})| > 1$, at some point in the search, the searcher must encounter a *tie* where some $k \in \{2, \ldots, n\}$ boxes satisfy (7.2.2). At such a tie, any Gittins index sequence must search the k tied boxes next, but the order the k boxes are searched is unconstrained. Thereafter, any two Gittins index sequences will be identical until another tie is encountered. Therefore, elements of $C(\mathbf{p})$ differ from one another only in how they break ties.

How the searcher chooses to break ties is important, since knowledge of the searcher's tie-breaking preferences could be used by the hider to their advantage. Therefore, a mixed tie-breaking strategy is required by the searcher. To define a tie-breaking strategy, write S_n for the set of permutations of $\{1, \ldots, n\}$. A permutation $\sigma \in S_n$ serves as a preference ordering to choose which box to search next if a tie is encountered. For $\sigma \in S_n$, let $\sigma(i)$ represent the position of i in σ for $i \in \{1, \ldots, n\}$. Suppose, without loss of generality, after a tie between boxes $1, \ldots, k$, the next k searches are made in the order $1, \ldots, k$. Then we say the tie is *broken using* σ if $\sigma(1) < \cdots < \sigma(k)$.

Of interest are Gittins index sequences that break every tie using the same preference ordering. Let $\sigma \in S_n$ and $\mathbf{p}$ be a mixed hiding strategy. Write $\xi_\sigma(\mathbf{p})$ for the Gittins index sequence against $\mathbf{p}$ that breaks every tie encountered using σ. We define the following subset of $C(\mathbf{p})$,

$$\widehat{C}(\mathbf{p}) \equiv \{\xi_\sigma(\mathbf{p}) : \sigma \in S_n\}. \tag{7.3.3}$$

Whilst $C(\mathbf{p})$ could be an infinite set, $|\widehat{C}(\mathbf{p})| \leq n!$ since $|S_n| = n!$. By Theorem 7.2.7, there exists an optimal search strategy a mixture of at most n elements of $C(\mathbf{p}^*)$ for any optimal hiding strategy $\mathbf{p}^*$. The aim of the remainder of the section is to show that the same holds true if we replace $C(\mathbf{p}^*)$ with $\widehat{C}(\mathbf{p}^*)$.

For any search strategy η (pure or mixed), write $V(\eta) \equiv (V_1(\eta), \ldots, V_n(\eta))$. Now define

$$S(\mathbf{p}) \equiv \{V(\xi) : \xi \in C(\mathbf{p})\} \quad \text{and} \quad \widehat{S}(\mathbf{p}) \equiv \{V(\xi) : \xi \in \widehat{C}(\mathbf{p})\},$$

noting that $\widehat{S}(\mathbf{p}) \subset S(\mathbf{p}) \subset \mathbb{R}^n$.

Clearly, if some search strategy η is a mixture of a subset of $C(\mathbf{p})$, then $V(\eta)$ can be written as a convex combination of the elements in $S(\mathbf{p})$. The following lemma, proved in Appendix C.2, shows that the same statement is true replacing $S(\mathbf{p})$ with $\widehat{S}(\mathbf{p})$.

Lemma 7.3.3 For any hiding strategy $\mathbf{p}$ with $p_i > 0$ for $i = 1, \ldots, n$, the convex hull of $S(\mathbf{p})$ is equal to the convex hull of $\widehat{S}(\mathbf{p})$.

Lemma 7.3.3 allows us to present the main theorem of this section, which strengthens Theorem 7.2.7 by replacing $C(\mathbf{p}^*)$ with $\widehat{C}(\mathbf{p}^*)$.

Theorem 7.3.4 In search game G, for any optimal hiding strategy $\mathbf{p}^*$, there exists an optimal search strategy which is a mixture of at most n elements of $\widehat{C}(\mathbf{p}^*)$.

Proof. Let $\mathbf{p}^* \equiv (p_1^*, \ldots, p_n^*)$ be an optimal hiding strategy. By Theorem 7.2.7, there exists an optimal search strategy η^* which is a mixture of elements of $C(\mathbf{p}^*)$. Therefore, $V(\eta^*)$ can be written as a convex combination of elements of $S(\mathbf{p}^*)$ and hence belongs to the convex hull of $S(\mathbf{p}^*)$.

By the hider's part of Proposition 7.2.4, $p_i^* > 0$ for $i \in \{1, \ldots, n\}$. Therefore, by Lemma 7.3.3, the convex hull of $S(\mathbf{p}^*)$ is equal to the convex hull of $\widehat{S}(\mathbf{p}^*)$. It follows that $V(\eta^*)$ also belongs to the convex hull of $\widehat{S}(\mathbf{p}^*)$ and hence, by the definition of a convex hull, $V(\eta^*)$ can be written as a convex combination of elements in $\widehat{S}(\mathbf{p}^*)$.

By Carathéodory's theorem, $V(\eta^*)$ can be written as a convex combination of at most $n+1$ elements in $\widehat{S}(\mathbf{p}^*)$. Yet, for any $\mathbf{s} \equiv (s_1, \ldots, s_n) \in S(\mathbf{p})$, the weighted average $\sum_{i=1}^{n} s_i p_i$ is equal to the expected search time if the hider chooses $\mathbf{p}$ and the searcher any optimal counter $\xi \in C(\mathbf{p})$. Therefore, all elements of $S(\mathbf{p})$ lie on the same hyperplane in $\mathbb{R}^n$, and so do all elements in $\widehat{S}(\mathbf{p})$. Hence, the number of elements in the convex combination of $V(\eta^*)$ can be reduced to at most n. Therefore, η^* can be written as a mixture of at most n strategies in $\widehat{C}(\mathbf{p}^*)$. ∎

Theorem 7.3.4 extends the result in Proposition 8.4 of Gittins (1989), which is restricted to two boxes with unit search times, to an arbitrary number of boxes and arbitrary search times. It shows that it is sufficient for the searcher to consider search sequences which use the same preference ordering to break every tie.

Gittins proves his Proposition 8.4 via a direct argument which he claims is not difficult to extend to arbitrary n. However, upon further study, we found Gittins' method makes limited progress for $n \geq 3$; the following explains.

For any (pure or mixed) search strategy η and $i, j \in \{1, \ldots, n\}$, write $D_{i,j}(\eta) \equiv V_i(\eta) - V_j(\eta)$. If $D_{i,j}(\eta) = 0$ for some $i, j \in \{1, \ldots, n\}$, we say η *equalises* boxes i and j. Let η_1 and η_2 be search strategies, and relabel boxes and sequences if necessary so that $D_{i,j}(\eta_1) \leq 0$. Then, by direct computation, it is easy to show that there exists a mixture of η_1 and η_2 that equalises boxes i and j if any only if $D_{i,j}(\eta_2) \geq 0$.

Let $\mathbf{p}^*$ be an arbitrary optimal hiding strategy. Recall $\xi_\sigma(\mathbf{p}^*)$ is the Gittins index sequence against $\mathbf{p}^*$ that breaks every tie encountered using σ; write $\xi_\sigma \equiv \xi_\sigma(\mathbf{p}^*)$. When $n = 2$, we have $\widehat{C}(\mathbf{p}^*) = \{\xi_{12}, \xi_{21}\}$, where ξ_{12} (resp. ξ_{21}) breaks any tie in favour of box 1 (resp. box 2). Gittins' method shows that $D_{1,2}(\xi_{12}) \leq 0$ and $D_{1,2}(\xi_{21}) \geq 0$. In other words, there exists a mixture η^* of ξ_{12} and ξ_{21} which equalises boxes 1 and 2; by Theorem 7.3.1, η^* is optimal for the searcher.

By extending Gittins' method to an n-box problem, for any $i, j \in \{1, \ldots, n\}$, we can show that $D_{i,j}(\xi_{i\ldots j}) \leq 0$ and $D_{i,j}(\xi_{j\ldots i}) \geq 0$, where $x \ldots y$ is any permutation of S_n with $\sigma(1) = x$ and $\sigma(n) = y$. Since, for any $\xi \in C(\mathbf{p}^*)$, $D_{i,j}(\xi)$ must lie one side of 0, there exists a mixture of ξ and either $\xi_{i\ldots j}$ or $\xi_{j\ldots i}$ that equalises boxes i and j. Therefore, many mixtures of pairs in $\widehat{C}(\mathbf{p}^*)$ can be constructed that equalise boxes i and j.

To obtain an optimal search strategy using Theorem 7.3.1, we need a mixture of elements of $C(\mathbf{p}^*)$ that equalises all n boxes. Yet, problems occur when a third box, say k, is introduced. Suppose η_1 and η_2 both equalise boxes i and j; then, any mixture of η_1 and η_2 that equalises boxes i and k (or j and k) will equalise boxes i, j and k. However, there is no guarantee that $D_{i,k}(\eta_1)$ and $D_{i,k}(\eta_2)$ will have opposing signs, so there is no guarantee that such a mixture exists. Whilst we managed to prove that such η_1 and η_2 with $D_{i,k}(\eta_1) \leq 0 \leq D_{i,k}(\eta_2)$ exist when $n = 3$ (thus finding an optimal search strategy for the three-box case), the proof cannot be generalised to $n \geq 4$.

7.3.3 Testing the Optimality of a Hiding Strategy

From a practical viewpoint, if an optimal hiding strategy $\mathbf{p}^*$ is known, Theorem 7.3.4 allows the searcher to consider at most the $n!$ search sequences in $\widehat{C}(\mathbf{p}^*)$, and hence reduces G to a finite two-person, zero-sum game. This logic may be reversed. Suppose $\mathbf{p}$ is a hiding strategy, and the searcher cannot form a mixture that equalises all boxes from any subset of $\widehat{C}(\mathbf{p})$. Then, by a combination of Theorems 7.3.1 and 7.3.4, $\mathbf{p}$ is not optimal for the hider. This section develops this idea into a practical optimality test for any hiding strategy.

Write G_C for the subgame of G where the searcher's pure strategies are some finite $C \subset \mathcal{S}$. Then G_C is a finite, $n \times |C|$ game easily solved by linear programming (see Washburn (2003), Chapter 10 of Ferguson (2020)), with optimal strategies guaranteed to exist for both players. Write v_C^* for the value of G_C. Since $C \subset \mathcal{S}$, then v_C^* is an upper bound on v^*, the value of G.

By considering cases where the upper bound is attained, we present an optimality test for a hiding strategy. We say a hiding strategy $\mathbf{p} \equiv (p_1, \ldots, p_n)$ is *interior* if $p_i > 0$ for $i = 1, \ldots, n$; otherwise, we say $\mathbf{p}$ is *exterior*. By Proposition 7.2.4, any optimal hiding strategy is interior. The following result applies Theorem 7.3.4 to determine whether a given interior hiding strategy is optimal in G, and, if so, provides an optimal search strategy.

Proposition 7.3.5 Let $C \subseteq \widehat{C}(\mathbf{p})$ for some interior hiding strategy $\mathbf{p}$.

1. If the strategy $\mathbf{p}$ is optimal in G_C, then $\mathbf{p}$ is optimal in G, and any optimal search strategy in G_C is also optimal in G.

2. If the strategy $\mathbf{p}$ is not optimal in G_C, then any optimal hiding strategy in G_C is exterior, and, if $C = \widehat{C}(\mathbf{p})$, then $\mathbf{p}$ is not optimal in G.

Proof. First, we prove (1.). Suppose $\mathbf{p}$ is optimal in G_C. Since G_C is a finite game, there exists an optimal search strategy in G_C, say η, which guarantees the searcher an expected search time of at most v_C^*, no matter which box the hider hides in. Therefore, $V_i(\eta) \leq v_C^*$ for $i = 1, \ldots, n$. By the minimax theorem for finite games, when the hider plays $\mathbf{p}$ and the searcher plays η, the expected search time is v_C^*. In other words, we have

$$\sum_{i=1}^{n} p_i V_i(\eta) = v_C^*.$$

Since $p_i > 0$ for $i = 1, \ldots, n$, we must have $V_i(\eta) = v_C^*$ for $i = 1, \ldots, n$. In addition, η is a mixture of strategies in $C \subset C(\mathbf{p})$. By Theorem 7.3.1, $\mathbf{p}$ and η are an optimal strategy pair in G and $v^* = v_C^*$, proving (1.).

Now we prove (2.), the case in which $\mathbf{p}$ is not optimal in G_C. For any search sequence ξ, write h_ξ for the hyperplane in n-dimensional space satisfying $h_\xi(\mathbf{x}) \equiv \sum_{i=1}^{n} x_i V_i(\xi)$ for all $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$. For any hiding strategy $\mathbf{y}$, write $l(\mathbf{y}) \equiv \min_{\xi \in C} h_\xi(\mathbf{y})$, so any $\mathbf{y}$ that maximises $l(\mathbf{y})$ is optimal for the hider in G_C. Write $H \equiv \{h_\xi : \xi \in C\}$, a set of $|C|$ hyperplanes. Since $C \subset C(\mathbf{p})$, $h_\xi(\mathbf{p})$ is constant for all $\xi \in C$; therefore, the point $(\mathbf{p}, l(\mathbf{p}))$ belongs to every hyperplane in H in n-dimensional space. It follows that, for any line l through $(\mathbf{p}, l(\mathbf{p}))$ and any hyperplane $h \in H$, either h contains l, or h and l intersect at the unique point $(\mathbf{p}, l(\mathbf{p}))$.

Let L be the lower envelope of the hyperplanes in H; in other words, for any hiding strategy $\mathbf{y}$, we have $(\mathbf{y}, l(\mathbf{y})) \in L$. By definition, any point in L belongs to some $k \in \{1, \ldots, |C|\}$ hyperplanes in H. Since $\mathbf{p}$ is not optimal in G_C, there must exist $\bar{\mathbf{p}}$ with $l(\bar{\mathbf{p}}) > l(\mathbf{p})$. Write $\bar{H}$ for the hyperplanes in H containing $(\bar{\mathbf{p}}, l(\bar{\mathbf{p}}))$. It follows that any hyperplane in $\bar{H}$ contains the line passing through $(\mathbf{p}, l(\mathbf{p}))$ and $(\bar{\mathbf{p}}, l(\bar{\mathbf{p}}))$, and this line intersects any hyperplane in $H \setminus \bar{H}$ only at $(\mathbf{p}, l(\mathbf{p}))$. Since both $(\mathbf{p}, l(\mathbf{p}))$ and $(\bar{\mathbf{p}}, l(\bar{\mathbf{p}}))$ lie in L, so does the line segment between these two points, and so does its extension beyond $\bar{\mathbf{p}}$.

If $\bar{\mathbf{p}}$ is interior, we may extend the line connecting $(\mathbf{p}, l(\mathbf{p}))$ and $(\bar{\mathbf{p}}, l(\bar{\mathbf{p}}))$ until we first reach $(\widehat{\mathbf{p}}, l(\widehat{\mathbf{p}}))$ for some $\widehat{\mathbf{p}}$ exterior. Since $l(\bar{\mathbf{p}}) > l(\mathbf{p})$, we must have $l(\widehat{\mathbf{p}}) > l(\bar{\mathbf{p}})$, so any $\bar{\mathbf{p}}$ with $l(\bar{\mathbf{p}}) > l(\mathbf{p})$ is itself dominated by some exterior $\widehat{\mathbf{p}}$. It follows that any optimal hiding strategy in G_C must be exterior.

We now prove the second part of (2.) with $C = \widehat{C}(\mathbf{p})$ by contradiction. Suppose $\mathbf{p}$ is optimal in G. By Theorem 7.3.4, there exists a search strategy η^* which is both

1. Optimal in G, so guarantees the searcher an expected search time of at most v^*.

2. A mixture of strategies in $\widehat{C}(\mathbf{p})$, so is available to the searcher in the game G_C since $C = \widehat{C}(\mathbf{p})$.

In the game G_C, the hider can guarantee at least v^* with $\mathbf{p}$, and the searcher can guarantee at most v^* with η^*, so $\mathbf{p}$ is optimal in G_C, drawing the contradiction, which completes the proof. ■

We can test the optimality in G of any hiding strategy $\mathbf{p}$ using Proposition 7.3.5 by solving G_C with $C = \widehat{C}(\mathbf{p})$. The hiding strategy $\mathbf{p}_0 \equiv (p_{0,1}, \ldots, p_{0,n})$ with

$$p_{0,i} \equiv \frac{t_i/q_i}{\sum_{j=1}^{n} t_j/q_j}, \quad i = 1, \ldots, n, \tag{7.3.4}$$

is of particular interest, since it creates a tie between all n boxes at the start of the search, giving the searcher no preference over which box to search first. As discussed in Section 6.2.1, $\mathbf{p}_0$ is optimal for G (called IBOX in Section 6.2.1) with a mobile hider, and optimal for G with the usual immobile hider when the boxes are identical. Further, in a numerical study, Roberts and Gittins (1978) and Gittins and Roberts (1979) both find, for 2 and n boxes respectively, that $\mathbf{p}_0$ is often optimal for the hider in G with unit search times. In Section 7.4, we extend these studies of the optimality of $\mathbf{p}_0$ using Proposition 7.3.5.

7.4 Numerical Study

This section presents a numerical study which, using Proposition 7.3.5, determines how often the hiding strategy $\mathbf{p}_0$ defined in (7.3.4) is optimal for various parameters of the search game G. Similar investigations with unit search times are carried out in Roberts and Gittins (1978) and Gittins and Roberts (1979). Not only does our study encompass a much wider range of problems divided into comparable subgroups, but Proposition 7.3.5 is more efficient at determining the optimality of $\mathbf{p}_0$ than the method of Gittins and Roberts, which is described in the following. Recall $v(\mathbf{p})$ as the expected search time when the hider chooses $\mathbf{p}$ and the searcher chooses any search sequence in $C(\mathbf{p})$; an optimal hiding strategy maximises $v(\mathbf{p})$. Both papers by Gittins and Roberts determine the optimality of $\mathbf{p}_0$ by direct comparison of $v(\mathbf{p}_0)$ and $v(\mathbf{p}^*)$, where $\mathbf{p}^*$ is an optimal hiding strategy found by, starting at $\mathbf{p}_0$, making incremental adjustments to hiding strategies and evaluating $v(\mathbf{p})$.

Search games are split into two types. In the first, we have

$$(1-q_1)^{x_1} = \cdots = (1-q_n)^{x_n} \tag{7.4.1}$$

for some coprime, positive integers x_i, $i = 1, \ldots, n$. We call such games *cyclic search games*, since x_i searches of box i for $i = 1, \ldots, n$ reset the search problem.

We will also examine *acyclic* search games, where, for any distinct $i, j \in \{1, \ldots, n\}$,

$$(1-q_i)^{x} \neq (1-q_j)^{y} \tag{7.4.2}$$

for any strictly positive integers x and y. The proportion of problems with $\mathbf{p}_0$ optimal across the two types of search game will be compared.

Before introducing the numerical study, we first discuss the methods used to calculate $V_i(\xi)$, $i = 1, \ldots, n$, for search sequences $\xi \in C(\mathbf{p}_0)$.

7.4.1 Calculating Expected Search Times

Let $\mathbf{p}$ be a hiding strategy. For $i = 1, \ldots, n$, for any search sequence ξ, we have

$$V_i(\xi) = \lim_{K\to\infty} \sum_{k=1}^{K} q_i(1-q_i)^{k-1} b_i(k, \xi), \tag{7.4.3}$$

where $b_i(k, \xi)$ is the time at which the kth search of box i is made under ξ. For $\xi \in C(\mathbf{p})$, the terms $b_i(k, \xi)$ are determined by (7.2.2) and the rule ξ uses to break ties in (7.2.2). However, when computing indices in (7.2.2), detection probabilities, search times and $\mathbf{p}$ itself must be encoded as floating-point numbers, which can create problems identifying ties in (7.2.2). To apply Proposition 7.3.5 to $\mathbf{p}_0$, we need to calculate $V_i(\xi)$ for specific $\xi \in \widehat{C}(\mathbf{p}_0)$, so a computational method to accurately identify ties in (7.2.2) is important. Below, for both cyclic and acyclic search games, we present a technique which can reliably calculate $V_i(\xi)$ for a specific $\xi \in \widehat{C}(\mathbf{p}_0)$ by taking advantage of knowing when ties will be encountered.

Suppose we wish to evaluate ξ_σ, the sequence in $\widehat{C}(\mathbf{p}_0)$ that breaks ties using preference ordering σ whose ith element is denoted $\sigma(i)$, $i = 1, \ldots, n$. At the start of the search, there is a tie between all n boxes; therefore, the first n searches of ξ_σ will be of boxes $\sigma(1), \ldots, \sigma(n)$ in that order. In an acyclic search game, since (7.4.2) holds, ξ_σ will never encounter another tie between any pair of boxes. Therefore, from the $(n+1)$th search onwards, ξ_σ can be calculated using floating-point numbers to compute the indices in (7.2.2).

Whilst a finite approximation to (7.4.3) after any $K \in \mathbb{Z}^+$ searches of box i gives a lower bound for $V_i(\xi_\sigma)$, we calculate an upper bound for $V_i(\xi_\sigma)$ using the following method.

Write

$$m \equiv \left\lfloor \max_{i,j \in \{1,\ldots,n\}} \frac{\log(1-q_i)}{\log(1-q_j)} \right\rfloor + 1.$$

For any $i, j \in \{1, \ldots, n\}$, since $m > \log(1-q_i)/\log(1-q_j)$, then $(1-q_j)^m < (1-q_i)$, so between any two successive searches of box i, any Gittins index sequence will make at most m searches of box j, for $i \neq j$. Therefore, for any $i \in \{1, \ldots, n\}$, no more than time $\widehat{m} \equiv \sum_{j=1}^n m t_j$ can elapse between successive searches of box i following the Gittins index sequence ξ_σ.

It follows that, for any $K \in \mathbb{Z}^+$ and $i \in \{1, \ldots, n\}$, following ξ_σ until K searches of box i have been made, then, after that, assuming box i is searched at regular time intervals of length $\widehat{m}$ gives an upper bound on $V_i(\xi_\sigma)$. In other words,

$$\begin{aligned} V_i(\xi_\sigma) &\leq \sum_{k=1}^{K}(1-q_i)^{k-1}[b_i(k,\xi)-b_i(k-1,\xi)] + (1-q_i)^K \sum_{k=1}^{\infty} \widehat{m}(1-q_i)^{k-1} \\ &\leq \sum_{k=1}^{K}(1-q_i)^{k-1}[b_i(k,\xi)-b_i(k-1,\xi)] + \frac{\widehat{m}(1-q_i)^K}{q_i}. \end{aligned} \tag{7.4.4}$$

Note that the first term in (7.4.4) is the lower bound for $V_i(\xi_\sigma)$ obtained via a finite approximation to (7.4.3) after L searches of box i. Therefore, we increase K until the ratio of the second term of (7.4.4) divided by the first term of (7.4.4) is less than 10^{-8}.

In a cyclic search game, further ties will be encountered after the first n searches. In particular, it follows from the rule in (7.2.2) that the first $\widehat{x} \equiv \sum_{i=1}^{n} x_i$ searches will include exactly x_i searches of box i for $i = 1, \ldots, n$, after which all n boxes will again be tied, so the problem has reset itself. Therefore, the same cycle of $\widehat{x}$ searches will be indefinitely repeated. It follows that, for any $a \in \{1, 2, \ldots\}$, we have $b_i(k+ax_i, \xi_\sigma) = b_i(k, \xi_\sigma)+a\widehat{x}$ for $i = 1, \ldots, n$, which allows the following simplification of (7.4.3).

$$\begin{aligned} V_i(\xi_\sigma) &= \sum_{a=0}^{\infty}\left(\sum_{k=1}^{x_i} q_i(1-q_i)^{k-1+ax_i} b_i(k+ax_i,\xi_\sigma)\right) \\ &= \sum_{a=0}^{\infty}(1-q_i)^{ax_i}\left(\sum_{k=1}^{x_i} q_i(1-q_i)^{k-1}\left[b_i(k,\xi_\sigma)+a\widehat{x}\right]\right) \\ &= \sum_{a=0}^{\infty}(1-q_i)^{ax_i}\left[A_i(\xi_\sigma)+a\widehat{x}(1-c)\right] \\ &= \frac{A_i(\xi_\sigma)+\widehat{x}c}{(1-c)}, \end{aligned}$$

where c is equal to $(1-q_i)^{x_i}$, constant over all boxes by (7.4.1), and

$$A_i(\xi) \equiv \sum_{k=1}^{x_i} q_i(1-q_i)^{k-1} b_i(k,\xi). \tag{7.4.5}$$

Therefore, to evaluate $V_i(\xi_\sigma)$ precisely, we only need to calculate $b_i(k, \xi_\sigma)$ for $k \in \{1, \ldots, x_i\}$. Yet, the repeated cycle of $\widehat{x}$ searches may involve more than just the tie between all n boxes at the start of the cycle. Therefore, from the $(n + 1)$th search onwards, using floating-point numbers to compute indices in (7.2.2) (and hence the $b_i(k, \xi_\sigma)$ terms for $k \in \{1, \ldots, x_i\}$) may again lead to issues identifying ties. However, the relationships in (7.4.1) allow an alternative method to calculate $b_i(k, \xi_\sigma)$ for $k \in \{1, \ldots, x_i\}$ involving an alternative set of indices which encodes integers rather than floating-point numbers, described as follows.

Since ξ_σ is a Gittins index policy against $\mathbf{p}_0$, all n indices in (7.2.2) are equal at the start of the search, say to y. For $i = 1, \ldots, n$, suppose $m_i \in \{1, \ldots, x_i\}$ searches of box i have been made, so the current corresponding index in (7.2.2) is $y(1 - q_i)^{m_i}$. Then we have

$$y(1 - q_i)^{m_i} \propto (1 - q_i)^{m_i} = ((1 - q_i)^{x_i})^{m_i/x_i} = c^{m_i/x_i} \propto x_i/m_i.$$

Therefore, the rule in (7.2.2) is equivalent to searching any box j satisfying

$$j = \underset{i=1,\ldots,n}{\arg\max} \frac{x_i}{m_i}. \tag{7.4.6}$$

Yet, both x_i and m_i are integers, so, unlike using (7.2.2), ties will always be detected using (7.4.6). Therefore, for $i = 1, \ldots, n$, $A_i(\xi_\sigma)$, and hence $V_i(\xi_\sigma)$, can be calculated exactly for any preference ordering σ.

7.4.2 Sample Schemes

We now introduce the numerical study, beginning with the generation of boxes, for which acyclic and cyclic search games require different methods. To generate a search time and detection probability for a box in an acyclic search game, we draw

$$q \sim U(q_l, q_u) \quad t \sim U(1, 5) \tag{7.4.7}$$

for pre-specified $0 < q_l < q_u < 1$. The resulting search game with n such boxes will be acyclic, since for any q_i and q_j drawn from a continuous uniform distribution, $\log(1 - q_i)/\log(1 - q_j) \in \mathbb{Q}$ with probability 0, and hence (7.4.2) is satisfied almost surely.

Table 7.4.1: Sample schemes used throughout the numerical study by values of q_l and q_u used in (7.4.7) and (7.4.8).

Sample Scheme	$[q_l, q_u]$
Varied	$[0.1, 0.9]$
Low	$[0.1, 0.5]$
Medium	$[0.3, 0.7]$
High	$[0.5, 0.9]$

To generate search times and detection probabilities for a set of n boxes in a cyclic search game, we draw

$$q_1 \sim U(q_l, q_u) \quad x_i \sim DU(1, 10) \quad t_i \sim U(1, 5), \quad i = 2, \ldots, n, \tag{7.4.8}$$

for pre-specified $0 < q_l < q_u < 1$, with $DU(1, 10)$ representing the discrete uniform distribution where each integer in $\{1, \ldots, 10\}$ is selected with equal probability. Whilst q_1 is drawn directly by (7.4.8), for $i = 2, \ldots, n$, we attain q_i using x_i, q_1, and the relationship in (7.4.1). To allow comparisons of acyclic and cyclic search games generated using the same q_l and q_u, we use (7.4.8) with rejection sampling, rejecting a search game if (7.4.8) leads to $q_i \notin [q_l, q_u]$ for any $i \in \{2, \ldots, n\}$.

To explore a variety of detection probabilities, we set $q_l = 0.1$ and $q_u = 0.9$ in (7.4.7) and (7.4.8). However, to investigate the effect of the size of detection probabilities, other choices of q_l and q_u will also be used and results compared to one another; see Table 7.4.1. Search games with $n = 2$, 3 and 5 boxes will be investigated. To account for increased variation within a search game as n increases, for each value of n and sample scheme, we study both $n \times 1000$ acyclic and $n \times 1000$ cyclic search games. Results are presented in the next section.

7.4.3 Numerical Results

For each generated search game, we test the optimality of $\mathbf{p}_0$ using Proposition 7.3.5, which involves solving the finite game G_C with the searcher restricted to the set of pure strategies $C = \widehat{C}(\mathbf{p}_0)$. By Proposition 7.3.5, $\mathbf{p}_0$ is optimal in G if and only if $\mathbf{p}_0$ is optimal in G_C. We find v_C^*, the value of G_C, by the standard linear programming methods for solving finite games discussed in Washburn (2003) and Chapter 10 of Ferguson (2020). Since G_C may have multiple optimal hiding strategies, to determine the optimality of $\mathbf{p}_0$ in G_C, we compare v_C^* to $v(\mathbf{p}_0)$, the expected search time when the hider plays $\mathbf{p}_0$ and the searcher plays any search sequence in $C(\mathbf{p}_0)$. In principle, $\mathbf{p}_0$ is optimal in G_C if and only if $v(\mathbf{p}_0)$ and v_C^* are equal. However, due to limitations in computational accuracy, we accept equality if $|v(\mathbf{p}_0) - v_C^*|/v_C^* < 10^{-6}$.

For $n = 2$, 3 and 5, Table 7.4.2 shows the percentage of generated search games in which $\mathbf{p}_0$ is optimal for different sample schemes in Table 7.4.1. Since $|\widehat{C}(\mathbf{p}_0)| = n!$, it is computationally infeasible to solve G_C, and hence test the optimality of $\mathbf{p}_0$, for n much larger than 5.

Table 7.4.2: The percentage of search games in which $\mathbf{p}_0$ is optimal by the number of boxes n, search game type, and sample scheme.

	Acyclic				Cyclic			
n	Varied	Low	Medium	High	Varied	Low	Medium	High
2	39.5	30.5	64.4	88.6	61.7	65.4	81.2	94.6
3	21.9	13.4	55.0	91.7	45.6	48.6	76.3	97.1
5	8.5	3.8	42.7	96.9	24.8	26.9	73.8	98.9

Table 7.4.2 shows $\mathbf{p}_0$ often achieves optimality, but with a strong dependence on n, the sample scheme, and whether the search game is cyclic or acyclic. We first explain the patterns in sample scheme and search game type evident for each fixed n.

Recall Norris (1962) shows that if the hider was free to change boxes after every unsuccessful search (called mobile-hider IBOX in Section 6.2.1), it is optimal for the hider to choose a new box according to $\mathbf{p}_0$, independent of previous hiding locations. Note that, if the hider plays $\mathbf{p} \equiv (p_1, \dots, p_n)$ and the searcher first searches box i, then $p_i q_i / t_i$ is the detection probability per unit time of this first search, $i = 1, \dots, n$. Since $\mathbf{p}_0$ equates these terms across all boxes and hence gives the searcher no preference of a box to search, Norris' result is unsurprising.

The following observation is made by Roberts and Gittins (1978) and was discussed briefly in Section 6.2.1 of this thesis. Another way to view Norris' result is that it is optimal for the hider to keep the Gittins indices in (7.2.2) equal throughout the search process. In our search game, the hider hides once at the start of the search, so it is not possible to maintain equality in (7.2.2). Intuitively, the best the hider can do is hide with probability $\mathbf{p}^*$ such that, when the searcher follows a Gittins index sequence against $\mathbf{p}^*$, the Gittins indices in (7.2.2) stay close to each other throughout the search. Such closeness is more important towards the start of the search, since the probability that the hider remains undetected decreases as time passes. Therefore, it is more important to the hider to approximate equality in (7.2.2) earlier in the search rather than later, explaining why $\mathbf{p}_0$ is often optimal for the hider.

The preceding argument explains the following patterns in Table 7.4.2. We see $\mathbf{p}_0$ optimal more often with the high sample scheme compared to the medium sample scheme compared to the low sample scheme since the larger the detection probabilities, the sooner the hider is likely to be detected, and hence equality in (7.2.2) near the start of the search takes even more importance.

Further, recall that in a cyclic search game, where (7.4.1) holds, unlike an acyclic game, if the hider plays $\mathbf{p}_0$ and the searcher adopts any Gittins index sequence against $\mathbf{p}_0$, after $\widehat{x} \equiv \sum_{i=1}^{n} x_i$ searches, equality in (7.2.2) is reattained. Therefore, cyclic games are more suited to approximating equality in (7.2.2) by starting at $\mathbf{p}_0$, explaining why $\mathbf{p}_0$ is optimal more often in cyclic than in acyclic search games.

Note that, from their own numerical study in Gittins and Roberts (1979), the authors conclude that whether a search game is acyclic or cyclic has no effect on their results. In our results in Table 7.4.2, whilst the effect of the sample scheme and number of boxes n on the *optimality rate* of $\mathbf{p}_0$—the percentage of search games in which $\mathbf{p}_0$ is optimal— appears to be unaffected by the search game type, $\mathbf{p}_0$ is clearly optimal less frequently in acyclic search games.

We also see $\mathbf{p}_0$ optimal more often with the medium sample scheme, with its narrow range of detection probabilities, compared to the varied sample scheme. To explain this phenomenon, we make the following connection to Chapter 3 of Part I, where the searcher was assumed to know the strategy of the hider, but had a choice of two search modes when searching any box.

In our search game, the hider chooses $\mathbf{p}$ to make the search last as long as possible, which involves balancing maximising uncertainty about their location and forcing the searcher into boxes with ineffective search modes. Chapter 3 of Part I introduces two measures of the effectiveness of a search mode of a box i. The first, called the *immediate benefit*, is given by q_i/t_i. The larger the immediate benefit of box i, the greater the detection probability per unit time when box i is searched. The second, called the *future benefit*, is given by

$$\frac{-p_i(1-p_i)\log(1-q_i)}{t_i}. \tag{7.4.9}$$

The larger the future benefit of box i, the greater the information per unit time gained about the hider's location on an unsuccessful search of box i. Inspection of its definition in (7.3.4) shows that, whilst $\mathbf{p}_0$ takes the immediate benefit of the n boxes' search modes into account by hiding in box i with probability proportional to t_i/q_i, it ignores the future benefit.

In Norris' game, where the hider may change boxes after every search and $\mathbf{p}_0$ is optimal, since the game resets after every search, information gained by the searcher about the hider's location is useless and hence there is no concept of future benefit. Yet, in our search game, gaining more information about the hider's location enables the searcher to make better box choices later in the search. Therefore, the hider should be dissuaded from hiding in boxes with a large future benefit, as the information-gain advantages of these boxes will benefit the searcher.

Since $\mathbf{p}_0$ does not take future benefit into account, the larger the variation in future benefit between the n boxes, the less appealing is $\mathbf{p}_0$ for the hider. With the varied sample scheme, there is more opportunity for such variation, hence $\mathbf{p}_0$ is optimal less frequently than with the narrower medium sample scheme.

We now examine patterns as n changes. In general, $\mathbf{p}_0$ is optimal less frequently as n increases, since the more boxes there are, the greater the uncertainty in the hider's location and hence, as demonstrated in Chapter 3 of Part I, the more valuable information about the hider's location becomes. Therefore, the future benefit, ignored by $\mathbf{p}_0$, takes more importance as n grows, so $\mathbf{p}_0$ is optimal less often.

We further examine how the sample scheme affects the degradation in the optimality rate of $\mathbf{p}_0$ with n. As previously noted, the smaller the variation in future benefit between the n boxes, the more frequently $\mathbf{p}_0$ is optimal. The narrower the range of detection probabilities in the sample scheme, the quicker the variation in future benefit decreases as we add more boxes. Therefore, the degradation of the optimality rate of $\mathbf{p}_0$ as n increases is steeper with the varied than with the narrower medium sample scheme since, in the latter, this decrease in future benefit variation suppresses the increasing importance of future benefit as n grows.

Further, the smaller the detection probabilities, the longer the search is expected to last, so the more important future box choices become. Hence, the smaller the detection probabilities, the greater the importance of the future benefit and the less frequently $\mathbf{p}_0$ is optimal. Therefore, with the low sample scheme, we see the sharpest decline in the optimality rate of $\mathbf{p}_0$ as n increases. Due to large detection probabilities and a narrow sample scheme, the optimality rate of $\mathbf{p}_0$ actually improves with n with the high sample scheme.

Future Benefit in Roberts and Gittins (1978) and Ruckle (1991) To conclude this section, we discuss several unexplained results of Roberts and Gittins (1978) and Ruckle (1991) which we believe to be caused by future benefit.

As discussed in Section 6.2.1, Roberts and Gittins (1978) studied two-box search games with $q_1 < q_2$ and unit search times. The authors observed that, whenever $\mathbf{p}_0 \equiv (p_0, 1 - p_0)$ was suboptimal, p^*, an optimal hiding probability in box 1, was greater than p_0, but found no reason for this observation. We believe this phenomenon is explained by future benefit. Since $q_1 < q_2$ and $t_1 = t_2 = 1$, the future benefit in (7.4.9) at any $\mathbf{p}$ is greater for box 2 than box 1. Whilst $\mathbf{p}_0$ considers immediate benefit, it ignores future benefit, explaining why the hider, who wants the searcher to spend more time in boxes with inefficient search modes, may prefer to hide in box 1 with a probability greater than p_0.

Also mentioned briefly in Section 6.2.1, Ruckle (1991) solves a two-box game with $t_1 = t_2 = q_2 = 1$ and a sole parameter $q_1 \equiv q \in (0, 1)$. In this problem

$$p_0 = \frac{1/q}{1/q + 1}.$$

Ruckle (1991) shows that the hider optimally hides in box 1 with probability

$$p^* \equiv \frac{1/q}{1/q + (1-q)^{h-1}} \quad \text{where} \quad h \equiv \lfloor \bar{h} \rfloor \quad \text{with} \quad \bar{h} = 1/q + (1-q)^{\bar{h}-1}. \qquad (7.4.10)$$

We analyse this result of Ruckle (1991) to conclude the following. The optimal hiding strategy in (7.4.10) leads to a tie between the Gittins indices of the two boxes for the hth search, and hence p_0 is optimal for the hider if and only if $h = 1$. In general, $p^* \geq p_0$, and as q decreases, h increases, so p^* increases; see Table 7.4.3.

We offer the following explanation. Recall that p_0 ignores future benefit. For any $q \in (0, 1)$, the future benefit in (7.4.9) is greater for box 2 than for box 1, with the size of the difference growing as q decreases. Therefore, for larger q, the hider optimally hides in box 1 with probability p_0, since difference in future benefit between the two boxes is small. As q decreases, the advantage in future benefit of box 2 over box 1

Table 7.4.3: Values of h by q in the two-box game of Ruckle (1991).

Value of h	Range of q
1	$[0.618, 1]$
2	$[0.382, 0.618]$
3	$[0.276, 0.382]$

grows. Wishing the searcher to spend more time in boxes with poor search modes, the hider optimally hides in box 1 with a probability p^* increasing from p_0.

Ruckle (1991) also finds that an optimal search strategy is a mixture of the two search sequences which make their only search of box 2 on the hth (resp. $(h+1)$th) search. Since any Gittins index policy against p^* encounters its only tie on its hth search, Ruckle's optimal search strategy is a mixture of the two elements of $\widehat{C}(p^*)$, so satisfies Theorem 7.3.4.

7.5 Additional Properties of Optimal Hiding Strategies

If $\mathbf{p}_0$ is suboptimal, Proposition 7.3.5 is less useful for finding an optimal hiding strategy, as only one hiding strategy can be tested at once. In this instance, the algorithm of Lin and Singham (2015) discussed in Section 6.2.1 efficiently finds an optimal strategy for each player by successively computing tighter bounds on v^*, the value of the game. The majority of the work in this chapter will be combined with the algorithm of Lin and Singham (2015) and submitted for publication in 2020.

Whilst the algorithm of Lin and Singham (2015) is efficient, it still requires the solution of numerous finite two-person, zero-sum games. With only calculation of expected search times needed, the corollary to the following lemma and its corollary can 'point the hider in the right direction' towards an optimal hiding strategy.

Lemma 7.5.1 Let $\mathbf{p}$ be a hiding strategy. If, for some $k \in \{1, \ldots, n\}$, we have $V_k(\xi) > v^*$ or $V_k(\xi) < v^*$ for all $\xi \in C(\mathbf{p})$, then $\mathbf{p}$ is not optimal for the hider.

Proof. We use proof by contradiction. Let $\mathbf{p}$ be optimal for the hider and, for some $k \in \{1, \ldots, n\}$, suppose $V_k(\xi) > v^*$ for all $\xi \in C(\mathbf{p})$. (The case $V_k(\xi) < v^*$ for all $\xi \in C(\mathbf{p})$ is proved using a similar argument.) Let η^* be an optimal search strategy, which must exist by Theorem 7.2.7. By Proposition 7.2.3, only elements of $C(\mathbf{p})$ are played with a strictly positive probability by η^*, so $V_k(\eta^*) > v^*$. This contradicts the searcher's part of Proposition 7.2.4, so $\mathbf{p}$ cannot be optimal for the hider. ■

Note that, whilst Lemma 7.5.1 is of limited practical use in determining suboptimal hiding strategies since v^* is unknown, the corollary below is useful.

Corollary 7.5.2 Let $\mathbf{p}$ be a hiding strategy and write ξ_k^+ (resp. ξ_k^-) for any element of $C(\mathbf{p})$ that searches box k first (resp. last) when breaking any tie involving box k. If there exist $i, j \in \{1, \ldots, n\}$ with $V_j(\xi_j^-) < V_i(\xi_i^+)$, then $\mathbf{p}$ must be suboptimal.

Proof. First note that $V_k(\xi_k^+)$ and $V_k(\xi_k^-)$ remain the same regardless of the preference ordering of the $n-1$ boxes other than box k. Further, for any $\xi \in C(\mathbf{p})$, we have $V_k(\xi) \in [V_k(\xi_k^+), V_k(\xi_k^-)]$. If there exist $i, j \in \{1, \ldots, n\}$ with $V_j(\xi_j^-) < V_i(\xi_i^+)$, then either $V_j(\xi_j^-) < v^*$ or $v^* < V_i(\xi_i^+)$; therefore, either $V_j(\xi) < v^*$ for all $\xi \in C(\mathbf{p})$, or $v^* < V_i(\xi)$ for all $\xi \in C(\mathbf{p})$. It follows from Lemma 7.5.1 that $\mathbf{p}$ is suboptimal. ■

Corollary 7.5.2 not only provides a sufficient condition to verify that a hiding strategy is suboptimal by calculating only $2n$ expected search times, but also indicates why a hiding strategy is suboptimal if so, as explained in the following. By Theorem 7.3.1, an optimal search strategy η^* must equalise $V_i(\eta^*)$ across all boxes by mixing elements of $C(\mathbf{p})$. If there exist $i, j \in \{1, \ldots, n\}$ with $V_j(\xi_j^-) < V_i(\xi_i^+)$, then such an equalisation is not possible as any Gittins index sequence against $\mathbf{p}$ makes too few searches of box i and too many searches of box j. Therefore, to get closer to the optimal hiding strategy, it is intuitive that the hider should increase p_i and decrease p_j.

To finish this section, we present bounds on the probability with which the hider optimally hides in each box, reducing the size of the strategy space in which an optimal hiding strategy can lie. By the hider's part of Proposition 7.2.4, for any optimal hiding strategy $\mathbf{p}^* \equiv (p_1^*, \ldots, p_n^*)$, the hiding probability p_i^* belongs to the open interval $(0, 1)$ for $i = 1, \ldots, n$. With the aid of Lemma 7.5.1, we can restrict p_i^* to a closed subset of $(0, 1)$. First, we need a bound on v^*.

Lemma 7.5.3 The value v^* of the search game is bounded above by $U \equiv \sum_{i=1}^n t_i/q_i$ and below by $\max_{i \in \{1,\ldots,n\}} t_i/q_i$.

Proof. To derive the upper bound, consider a variation of the search game in which the hider is free to move between boxes after each unsuccessful search, solved by Norris (1962). As discussed in Section 6.2.1, the value of G (called IBOX in Section 6.2.1) with a mobile hider is

$$\sum_{j=1}^{n} \frac{t_j}{q_j}. \tag{7.5.1}$$

A mobile hider has more options, so (7.5.1) is an upper bound on the value v^* of G.

To prove the lower bound, suppose the hider hides in box j with probability 1. The optimal counter for the searcher is to search box j repeatedly until finding the hider, leading to an expected search time of t_j/q_j, which is a lower bound on v^*. Since this is true for any box j, the maximal value of t_i/q_i over all boxes $i \in \{1, \ldots, n\}$ is the best such lower bound on v^*. ■

Let U be the upper bound on v^* from Lemma 7.5.3. For $i = 1, \ldots, n$, write

$$m_i \equiv \left\lfloor \frac{U}{t_i} \right\rfloor + 1; \tag{7.5.2}$$

in other words, making m_i searches in box i requires more than U time units. The following bound is based on the idea that if p_i is small enough, there must be some box j that the searcher needs to search at least m_j times before searching box i for the first time. Therefore, U time units have passed before box i is visited, and, since $U \geq v^*$, we may invoke Lemma 7.5.1.

Proposition 7.5.4 For $i = 1, \dots, n$, write

$$c_i \equiv \sum_{k=1, k \neq i}^{n} \frac{t_k}{q_k(1-q_k)^{m_k-1}} \qquad \text{and} \qquad \delta_i \equiv \frac{t_i/q_i}{t_i/q_i + c_i}, \tag{7.5.3}$$

where m_k is defined in (7.5.2). Any optimal hiding strategy $(p_1^*, \dots, p_n^*)$ must have $p_i^* \in \left[\delta_i, 1 - \sum_{k=1, k\neq i}^{n} \delta_k\right]$, for $i = 1, \dots, n$.

Proof. Without a loss of generality, we prove the result for $i = n$.

We begin with the following observation. Let $\mathbf{p}$ be a hiding strategy. Suppose that, for any $\xi \in C(\mathbf{p})$, there exists $j \in \{1, \dots, n-1\}$ such that ξ makes at least m_j searches of box j before visiting box n. Then the first time box n is visited, more than time U has elapsed, so we must have $V_n(\xi) > U \geq v^*$ for all $\xi \in C(\mathbf{p})$; by Lemma 7.5.1, $\mathbf{p}$ is not optimal for the hider. Observation of (7.2.2) shows that, for such j to exist, we require

$$\frac{p_j q_j (1-q_j)^{m_j-1}}{t_j} > \frac{p_n q_n}{t_n} \quad \text{for some} \quad j \in \{1, \dots, n-1\}. \tag{7.5.4}$$

Now let $(\bar{p}_1, \dots, \bar{p}_{n-1}, \delta_n)$ be the unique solution to the following system of linear equations:

$$\frac{\bar{p}_k q_k (1-q_k)^{m_k-1}}{t_k} = \frac{\delta_n q_n}{t_n}; \quad k = 1, \dots, n-1; \quad \sum_{k=1}^{n-1} \bar{p}_k = 1 - \delta_n. \tag{7.5.5}$$

It is easy to verify that δ_n satisfies (7.5.3) with $i = n$.

We are now in a position to prove the contrapositive of the theorem. First suppose that $p_n < \delta_n$ for some hiding strategy $\mathbf{p} \equiv (p_1, \dots, p_n)$. Then, by (7.5.5), there must exist $j \in \{1, \dots, n-1\}$ such that $p_j > \bar{p}_j$. It follows that (7.5.4) holds, and hence $\mathbf{p}$ is not optimal for the hider. Finally, suppose that $p_n > 1 - \sum_{k=1}^{n-1} \delta_k$. Then there must exist $l \in \{1, \dots, n-1\}$ such that $p_l < \delta_l$. By the above argument applied to box l, $\mathbf{p}$ is not optimal for the hider. ■

The bounds in Proposition 7.5.4 are easy to calculate, but are rarely tight, with δ_i often close to 0.

Chapter 8

Conclusions and Further Work

In this chapter, Part II of this thesis is reviewed, and a direction for further work suggested.

8.1 Review of Part II

Chapter 6 shows that search games in which a searcher seeks a hider who actively avoids detection are well studied in the literature. However, the majority of work assumes the searcher always detects the hider when their paths cross. This assumption is unrealistic if the hider can conceal themselves in a crowded search space, or has planted a bomb or buried landmine. Introducing a chance of overlooking a stationary hider makes a search game much harder, as the searcher must revisit parts of the search space. Even the simplest search game with overlook, IBOX introduced in Section 6.2 with search space n discrete boxes, has an incomplete theory mostly limited to special cases; Chapter 7 extends this theory to a fully-general IBOX.

In IBOX, the hider chooses a box to hide in, and then the searcher visits the boxes sequentially to minimise the expected time needed to find the hider. If the searcher knows the probabilities $\mathbf{p}$ with which the hider hides in each box, we have a classic one-sided search problem which, as discussed in Part I, is solved by a Gittins index policy against $\mathbf{p}$ where the searcher optimally next searches a box with a maximal Gittins index. However, since there are multiple options when there is a tie for the box with the largest Gittins index, there are multiple Gittins index policies against $\mathbf{p}$. Whilst in the one-sided problem all such policies are optimal, in the two-sided IBOX, the searcher must carefully choose a tie-breaking strategy of which an optimally-playing hider cannot take advantage. Theorem 7.2.7 in Section 7.2 shows that such an optimal strategy for the searcher exists, a result previously proven only in special cases. Further, Theorem 7.3.4 in Section 7.3.2 shows there exists an optimal search strategy with a simple tie-breaking rule easily found when an optimal hiding strategy is known and the number of boxes is not too large. The proof of neither theorem is straightforward, showing the complexity of the seemingly simple IBOX.

Chapter 7 also makes progress on the hider's side. Section 7.3.3 develops novel bounds on an optimal hiding probability in each box (Proposition 7.5.4), and a simple test of optimality for any hiding strategy (Proposition 7.3.5). In special cases, the literature has numerically investigated the optimality of a particular hiding strategy, $\mathbf{p}_0$, which gives the searcher no preference of a box at the start of the search, and is known to be optimal when the hider can hide anew after each unsuccessful search. Using the aforementioned optimality test in Proposition 7.3.5, Section 7.4 extends these numerical investigations, making valuable insight by comparing the optimality rate of $\mathbf{p}_0$ for problems with high and low chances of overlooking the hider.

8.2 Further Work: IBOX with Two Search Modes

The remainder of this chapter discusses further work. As previously mentioned, parts of Chapter 7 and the algorithm of Lin and Singham (2015), which finds optimal strategies for either player, will be combined and submitted for publication in 2020. The numerical study of Section 7.4 will be extended to examine the performance of $\mathbf{p}_0$ as a heuristic strategy for the hider.

Section 6.2.2 shows several extensions to IBOX have already been studied in the literature; the remainder of this section discusses an unstudied extension most relevant to Part I of this thesis, namely IBOX with a fast and a slow search mode available for each box. A slow (resp. fast) search of a box takes search time t_s (resp. t_f), and finds the hider, if in the box, with detection probability q_s (resp. q_f). The slow mode has a larger detection probability, but takes longer to finish, so $q_s > q_f$ and $t_s > t_f$.

With the addition of a second search mode, a pure hiding strategy remains a box to hide in, but the set of pure search strategies grows since a search sequence must now specify both a box to search and a mode at which to search it. Since we still have a semi-infinite two-person, zero-sum game, the theory of Blackwell and Girshick (1954) still applies to guarantee the existence of a value and an optimal hiding strategy. However, it remains to be seen whether the proof of Theorem 7.2.7 can be extended to guarantee the existence of an optimal search strategy. Further, an extension of Theorem 7.3.4 may be possible if, for each box, a sequence determining the modes to use to search that box is fixed. Under such fixed mode sequences, Gittins index policies against a hiding strategy may be defined, which, as in single-mode IBOX, differ from one another only in the way ties between indices are broken.

Recall that in the one-sided, two-mode search problem studied in Chapter 3 of Part I, the hiding probabilities for each box are known to the searcher since the 'hider' is an inanimate object. The key one-sided results from Chapter 3 of Part I are Theorems 3.2.5 and 3.2.8, which give sufficient conditions for each mode to dominate the other mode in the same box, so the dominated mode may be optimally discarded. Since these conditions depend only on the detection probabilities and search times of each box and not the hiding probabilities, Theorems 3.2.5 and 3.2.8 still apply to the two-mode search game. In other words, no matter the strategy of the hider, if

$$\frac{q_s}{t_s} \geq \frac{q_f}{t_f},$$

the searcher optimally always uses the slow mode of the box, and, if

$$\frac{q_f(1-q_s)}{t_f} \geq \frac{q_s}{t_s},$$

the searcher optimally always uses the fast mode of the box.

Outside of these conditions, Part I suggests the posterior probability p' with which the searcher believes the hider lies in the box plays a big part in the optimal mode choice. The best-performing heuristic policy from Part I searches the box fast if p' is greater than a threshold $\widehat{p}$, and otherwise tries both modes. Yet, as seen in Part II, the hider's optimal strategy $\mathbf{p}^*$ is likely to roughly equate the attractiveness of all boxes to the searcher, and hence an optimal counter by the searcher will keep posterior hiding probabilities close to the starting point $\mathbf{p}^*$. Therefore, especially if n is large, we are likely to have $p' < \widehat{p}$ throughout the search, the case where both available modes are tried.

Yet, by the above logic, posteriors are likely to only explore a small region of the hiding probability space, so it is likely that the same search mode will always be optimal in the same box. There are 2^n unique allocations assigning a single mode to each box. Chapter 3 of Part I investigated heuristically cutting down the number of allocations to test; this may be the best route to pursue for heuristics in a two-mode search game.

Part II Appendices

Appendix C

Supplementary Proofs for Part II

C.1 Proof of Lemma 7.2.5

As explained in the main body, by Blackwell and Girshick (1954), we may prove Lemma 7.2.5 by showing that $S(\epsilon)$ is closed. Write

$$\bar{S}(\epsilon) \equiv S(\epsilon) \setminus \{(V_1(\zeta_i(\epsilon)), \ldots, V_n(\zeta_i(\epsilon))) : i = 1, \ldots, n\}.$$

Since they differ by a finite subset of $\mathbb{R}^n$, $S(\epsilon)$ is closed if and only if $\bar{S}(\epsilon)$ is closed. Therefore, the proof will be completed by showing that $\bar{S}(\epsilon)$ is closed.

Throughout the proof, write $\mathbb{Z}^+ \equiv \{1, 2, \ldots\}$, and $V(\xi) \equiv (V_1(\xi), \ldots, V_n(\xi))$; therefore, any element of $\bar{S}(\epsilon)$ takes the form $V(\xi)$ where ξ is a Gittins index sequence against some $\mathbf{p} \in \mathcal{P}(\epsilon)$.

By Definition 7.2.1, the next box searched by any Gittins index sequence against a hiding strategy $\mathbf{p}$ must satisfy (7.2.2). If, at some point whilst following a Gittins index sequence against $\mathbf{p}$, multiple boxes satisfy (7.2.2), we say the searcher has encountered a *tie* and $\mathbf{p}$ is a *tie point.* Note that an equivalent definition of a tie point $\mathbf{p}$ is $|C(\mathbf{p})| > 1$. If $|C(\mathbf{p})| = 1$, we say $\mathbf{p}$ is a *non-tie point.*

If $\mathbf{p}$ is a non-tie point, then there is a unique Gittins index sequence against $\mathbf{p}$, whereas, for a tie point $\mathbf{p}$, as justified in Section 7.3.2, a specific Gittins index sequence against $\mathbf{p}$ is determined by how we break ties between boxes.

Let the set of rules for breaking ties be $\mathcal{R}$. We can think of $\mathcal{R}$ as a set of infinite sequences whose elements are permutations of $\{1, \ldots, n\}$. The jth element of $\mathbf{r} \in \mathcal{R}$ is the order of boxes with which the jth encountered tie should be broken. For example, suppose $n = 5$ and the jth tie following a Gittins index sequence against $\mathbf{p}$ involves boxes 2, 3 and 5. Suppose the jth element of $\mathbf{r}$ is 54231. Then, under rule $\mathbf{r}$, tie j is broken by searching boxes 5, 2 and 3 in that order. Note that changing the jth element of $\mathbf{r}$ to 45123 does not affect the Gittins index sequence generated, demonstrating that multiple rules can generate the same Gittins index sequence.

Further, $\mathcal{R}$ can be identified with the interval $[0, 1]$ in the following way. Any term in any $\mathbf{r} \in \mathcal{R}$ is one of the $n!$ elements of S_n, where S_n is the set of permutations of $\{1, \ldots, n\}$. Number the elements of S_n from 0 to $n!-1$, and rewrite $\mathbf{r}$ as $x_1 x_2 \ldots x_j \ldots$, where x_j is the number (from 0 to $n! - 1$) representing the jth element of $\mathbf{r}$. We now associate with $\mathbf{r}$ the real number given by

$$\phi(\mathbf{r}) \equiv \sum_{j=1}^{\infty} \frac{x_j}{(n!)^j}.$$

The mapping $\phi : \mathcal{R} \to [0, 1]$ is a bijection. Therefore, throughout the proof, by a convergent subsequence $\{\mathbf{r}_a : a \in \mathbb{Z}^+\}$ in $\mathcal{R}$, we mean a sequence for which $\{\phi(\mathbf{r}_a) : a \in \mathbb{Z}^+\}$ converges in $[0, 1]$. However, we shall continue to interpret elements of $\mathcal{R}$ as infinite sequences with terms in S_n for the remainder of the proof.

Before showing that $\bar{S}(\epsilon)$ is closed, we first show that a smaller set, concerning only Gittins index sequences against a fixed hiding strategy with $p_i > 0$ for $i = 1, \ldots, n$, is closed.

Lemma C.1.1 For any $\mathbf{p} \in \mathcal{P}$ with $p_i > 0$ for $i = 1, \ldots, n$, the set

$$S(\mathbf{p}) \equiv \{V(\xi) : \xi \in C(\mathbf{p})\} \tag{C.1.1}$$

is closed.

Proof. Since $p_i > 0$ for $i = 1, \ldots, n$, we have $S(\mathbf{p}) \subset \mathbb{R}^n$. If $|S(\mathbf{p})|$ is finite, then $S(\mathbf{p})$ is a finite collection of points in $\mathbb{R}^n$, which is always closed. For the rest of the proof, we concentrate on the case where $|S(\mathbf{p})|$ is infinite, so $\mathbf{p}$ must be a tie point.

Write $\{\mathbf{s}_a : a \in \mathbb{Z}^+\}$ for a convergent sequence in $S(\mathbf{p})$, and write $\mathbf{s}_0 \equiv \lim_{a\to\infty} \mathbf{s}_a$ for its limit. To show that $S(\mathbf{p})$ is closed, we need to show that $\mathbf{s}_0 \in S(\mathbf{p})$.

Write $\xi_\mathbf{r}$ for the Gittins index sequence against $\mathbf{p}$ that breaks ties using rule $\mathbf{r}$. Let f be the function from $\mathcal{R} \to S(\mathbf{p})$ that maps $\mathbf{r}$ to $V(\xi_\mathbf{r})$. Since any element of $S(\mathbf{p})$ corresponds to some rule $\mathbf{r} \in \mathcal{R}$, the image of f is equal to $S(\mathbf{p})$; therefore, for each a, we can choose $\mathbf{r}_a \in \mathcal{R}$ such that $\mathbf{s}_a = f(\mathbf{r}_a)$. Consider the sequence $\{\mathbf{r}_a : a \in \mathbb{Z}^+\}$ and, by identifying $\mathcal{R}$ with the interval $[0, 1]$, choose a convergent subsequence $\{\mathbf{r}_{h(a)} : a \in \mathbb{Z}^+\}$.

Identifying $\mathcal{R}$ with the interval $[0, 1]$ shows $\mathcal{R}$ is closed; therefore, we have $\mathbf{r}_0 \equiv \lim_{a\to\infty} \mathbf{r}_{h(a)} \in \mathcal{R}$, so $f(\mathbf{r}_0) \in S(\mathbf{p})$. Any infinite subsequence of the convergent sequence $\{\mathbf{s}_a\}$ must converge to the same limit as $\{\mathbf{s}_a\}$; therefore, we have $\mathbf{s}_0 = \lim_{a\to\infty} \mathbf{s}_{h(a)}$. To complete the proof, we show that $\mathbf{s}_0 = f(\mathbf{r}_0)$.

Note that we have $\mathbf{s}_{h(a)} = f(\mathbf{r}_{h(a)}) = V(\xi_{\mathbf{r}_{h(a)}})$, and $f(\mathbf{r}_0) = V(\xi_{\mathbf{r}_0})$. To ease notation, for the remainder of the proof, drop the subscript $\mathbf{r}$ in any sequence $\xi_{\mathbf{r}_a}$ (resp. $\xi_{\mathbf{r}_0}$), instead writing ξ_a (resp. ξ_0). Therefore, our aim to show that $\mathbf{s}_0 = f(\mathbf{r}_0)$ is equivalent to showing that $\lim_{a\to\infty} V(\xi_{h(a)}) = V(\xi_0)$.

Since $\mathbf{r}_0 = \lim_{a\to\infty} \mathbf{r}_{h(a)}$, for each $j \in \mathbb{Z}^+$, there must exist a smallest element of $\mathbb{Z}^+$, say a_j, such that the first j elements of $\mathbf{r}_{h(a)}$ are equal to the first j elements of $\mathbf{r}_0$ for all $a \geq a_j$. Further, the $\{a_j : j \in \mathbb{Z}^+\}$ must form an increasing sequence. Therefore, for any $a \geq a_j$, both $\xi_{h(a)}$ and ξ_0 break the first j ties, $j \in \mathbb{Z}^+$, encountered by any Gittins index sequence against $\mathbf{p}$ in the same manner, so as j increases, the first time when $\xi_{h(a_j)}$ and ξ_0 differ becomes increasingly later and later into the search, so the effect on the expected time to detection decreases to 0. It follows that, for $i = 1, \ldots, n$, we have $\lim_{j\to\infty} V_i(\xi_{h(a_j)}) \to V_i(\xi_0)$, and hence $\lim_{j\to\infty} V(\xi_{h(a_j)}) = V(\xi_0)$. Yet, since the $\{a_j : j \in \mathbb{Z}^+\}$ form an increasing sequence in $\mathbb{Z}^+$, we have $\lim_{j\to\infty} V(\xi_{h(a_j)}) = \lim_{a\to\infty} V(\xi_{h(a)})$, completing the proof. ■

Let

$$\mathcal{P} \equiv \left\{ (p_1, \dots, p_n) : p_i \geq 0,\ \sum_{i=1}^{n} p_i = 1 \right\}$$

be the space of mixed hiding strategies. Let $f_{\mathbf{r}}$ be the function from $\mathcal{P} \to \bar{S}(\epsilon)$ that maps $\mathbf{p}$ to $V(\xi_{\mathbf{r}}(\mathbf{p}))$, where $\xi_{\mathbf{r}}(\mathbf{p})$ is the Gittins index sequence against $\mathbf{p}$ that breaks ties using rule $\mathbf{r}$.

First, via two lemmas, we investigate the continuity of the functions $f_{\mathbf{r}}$ in $\mathcal{P}(\epsilon)$. The first lemma deals with the simpler case of continuity at non-tie points in $\mathcal{P}(\epsilon)$.

Lemma C.1.2 If $\mathbf{p} \in \mathcal{P}(\epsilon)$ is a non-tie point, then $f_{\mathbf{r}}$ is continuous at $\mathbf{p}$ for any $\mathbf{r} \in \mathcal{R}$.

Proof. Write ξ for the unique Gittins index sequence against $\mathbf{p}$. For $k \in \mathbb{Z}^+$, let $\mathcal{P}_k \subset \mathcal{P}$ contain precisely those mixed hiding strategies $\mathbf{x}$ for which every Gittins index sequence against $\mathbf{x}$ is identical to ξ for the first k searches. Clearly, for any $k \in \mathbb{Z}^+$, we have $\mathcal{P}_{k+1} \subseteq \mathcal{P}_k$ and $\mathbf{p} \in \mathcal{P}_k$.

For any $\delta > 0$, write $B(\mathbf{p}, \delta)$ for the open ball with radius δ centred at $\mathbf{p}$. Note that, since $\mathbf{p}$ is a non-tie point in $\mathcal{P}(\epsilon)$, for any $k \in \mathbb{Z}^+$, it is possible, in any direction, to move a small enough (Euclidean) distance in $\mathcal{P}$ away from $\mathbf{p}$ and not disrupt the order of the indices in (7.2.2) that generate the first k searches of ξ.

Therefore, we have

$$\delta_k \equiv 0.5 \times \sup \{\delta : B(\mathbf{p}, \delta) \subseteq \mathcal{P}_k\} > 0, \tag{C.1.2}$$

with $\delta_k \geq \delta_{k+1}$ and $B(\mathbf{p}, \delta_k) \subseteq \mathcal{P}_k$ for all $k \in \mathbb{Z}^+$, where 0.5 is an arbitrarily chosen number between 0 and 1. Write $\delta^* \equiv \lim_{k\to\infty} \delta_k$.

There are two cases. First suppose that $\delta^* > 0$. Let $\mathbf{x} \in B(\mathbf{p}, \delta^*)$; then $\mathbf{x} \in \mathcal{P}_k$ for all $k \in \mathbb{Z}^+$, so any Gittins index sequence against $\mathbf{x}$ is identical to ξ. It follows that ξ, the unique Gittins index sequence against $\mathbf{p}$, is also the unique Gittins index sequence against $\mathbf{x}$. Hence, for any $\mathbf{r} \in \mathcal{R}$, $f_{\mathbf{r}}$ is constant on $B(\mathbf{p}, \delta^*)$, so $f_{\mathbf{r}}$ is continuous at $\mathbf{p}$.

Second, suppose that $\delta^* = 0$. Let $\{\mathbf{x}_a\}$ be a sequence in $\mathcal{P}$ with $\lim_{a\to\infty} \mathbf{x}_a = \mathbf{p}$. Since $\{\mathbf{x}_a\}$ has limit $\mathbf{p}$ and, for any $k \in \mathbb{Z}^+$, we have $\delta_k > 0$, there must exist a smallest number $g(k) \in \mathbb{Z}^+$ such that every term in $\{\mathbf{x}_a\}$ after $\mathbf{x}_{g(k)}$ belongs to the ball $B(\mathbf{p}, \delta_k)$. Formally, for any $k \in \mathbb{Z}^+$, we write

$$g(k) \equiv \min\{A : \mathbf{x}_a \in B(\mathbf{p}, \delta_k),\ a \geq A\}. \tag{C.1.3}$$

Further, for any $l \in \mathbb{Z}^+$, since $\delta_k \geq \delta_{k+l}$, we have $B(\mathbf{p}, \delta_{k+l}) \subseteq B(\mathbf{p}, \delta_k)$ and hence $g(k) \leq g(k+l)$, so the sequence $\{g(k) : k \in \mathbb{Z}^+\}$ is increasing weakly.

Consider the sequence $\{\mathbf{x}_{g(k)} : k \in \mathbb{Z}^+\}$. We have $\lim_{a\to\infty} \mathbf{x}_a = \mathbf{p}$ by assumption; our next aim is to show that $\lim_{k\to\infty} \mathbf{x}_{g(k)} = \mathbf{p}$ also. To do this, we show that, for any $\epsilon > 0$, we can choose K such that $\mathbf{x}_{g(k)} \in B(\mathbf{p}, \epsilon)$ for all $k \geq K$. Choose $\epsilon > 0$. Since $\lim_{k\to\infty} \delta_k = 0$, there exists K such that $\delta_K < \epsilon$. By the definition of g in (C.1.3), we have $\mathbf{x}_a \in B(\mathbf{p}, \delta_K)$ for all $a \geq g(K)$. Since g is increasing, we have $\mathbf{x}_{g(k)} \in B(\mathbf{p}, \delta_K) \subset B(\mathbf{p}, \epsilon)$ for all $k \geq K$, showing that $\{\mathbf{x}_{g(k)}\}$ has limit $\mathbf{p}$.

By the definitions in (C.1.2) and (C.1.3), we have $\mathbf{x}_{g(k)} \in B(\mathbf{p}, \delta_k) \subset \mathcal{P}_k$. Recall ξ as the unique Gittins index sequence against $\mathbf{p}$, and, for $a \in \mathbb{Z}^+$, let ξ_a be any Gittins index sequence against $\mathbf{x}_a$. Since g is increasing and $\mathbf{x}_{g(k)} \in \mathcal{P}_k$, as k increases, the first time when $\xi_{g(k)}$ and ξ differ becomes increasingly later and later into the search, so the effect on the expected time to detection decreases to 0. Therefore, $V_i(\xi_{g(k)}) \to V_i(\xi)$ as $k \to \infty$, $i = 1, \ldots, n$. Since ξ_a is an arbitrary Gittins index sequence against $\mathbf{x}_a$, we draw the following conclusion for any rule $\mathbf{r} \in \mathcal{R}$:

$$\lim_{k\to\infty} f_{\mathbf{r}}(\mathbf{x}_{g(k)}) = \lim_{a\to\infty} f_{\mathbf{r}}(\mathbf{x}_a) = f_{\mathbf{r}}(\mathbf{p}),$$

and hence $f_{\mathbf{r}}$ is continuous at $\mathbf{p}$. ■

Now we consider the continuity of the $f_{\mathbf{r}}$ at tie points in $\mathcal{P}(\epsilon)$, the more challenging case. Informally, if $\mathbf{p} \in \mathcal{P}(\epsilon)$ is a tie point, then $f_{\mathbf{r}}$ is only continuous at $\mathbf{p}$ for certain $\mathbf{r} \in \mathcal{R}$, and only approaching $\mathbf{p}$ via certain paths in $\mathcal{P}$.

To state the continuity conditions precisely, we first need a few definitions. Let $\mathbf{p} = (p_1, \ldots, p_n)$ be a tie point in $\mathcal{P}(\epsilon)$. Let S_n denote the set of permutations of $\{1, \ldots, n\}$, and $\Sigma \subseteq S_n$. For $\sigma \in S_n$, write $\sigma(i)$ for the number in the ith position of

σ and write

$$\mathcal{P}(\mathbf{p}, \Sigma) \equiv \left\{\mathbf{x} \in \mathcal{P} : x_{\sigma(i)}/p_{\sigma(i)} \geq x_{\sigma(i+1)}/p_{\sigma(i+1)},\ i = 1, \ldots, n-1,\ \sigma \in \Sigma\right\}. \quad \text{(C.1.4)}$$

The set $\mathcal{P}(\mathbf{p}, \Sigma)$ may be interpreted as follows. Suppose $\mathbf{x} \in \mathcal{P}(\mathbf{p}, \Sigma)$. Then, for any $\sigma \in \Sigma$, if any tie encountered in a Gittins index sequence against $\mathbf{p}$ is broken using σ, the order that the tied boxes are searched remains the same if we replace $\mathbf{p}$ with $\mathbf{x}$ when calculating the terms in (7.2.2) and still use σ to break any remaining ties. Clearly $\mathbf{p} \in \mathcal{P}(\mathbf{p}, \Sigma)$ for any subset Σ of S_n. Further, note that if $\mathbf{x}, \mathbf{y} \in \mathcal{P}(\mathbf{p}, \Sigma)$, then for any $\lambda \in [0, 1]$, we must have $\lambda\mathbf{x} + (1-\lambda)\mathbf{y} \in \mathcal{P}(\mathbf{p}, \Sigma)$. Therefore, $\mathcal{P}(\mathbf{p}, \Sigma)$ is a convex set containing $\mathbf{p}$.

Informally, the following lemma says that, if $\mathbf{r} \in \mathcal{R}$ contains only elements of $\Sigma \subset S_n$, then $f_{\mathbf{r}}$ is continuous at $\mathbf{p}$ approaching from any path in $\mathcal{P}(\mathbf{p}, \Sigma)$.

Lemma C.1.3 Suppose $\mathbf{p} \in \mathcal{P}(\epsilon)$ is a tie point and $\Sigma \subset S_n$. Let $\{\mathbf{x}_a : a \in \mathbb{Z}^+\}$ be a sequence in $\mathcal{P}(\mathbf{p}, \Sigma)$ with $\lim_{a\to\infty} \mathbf{x}_a = \mathbf{p}$. Then, for any $\mathbf{r} \in \mathcal{R}$ whose elements all belong to Σ, we have $\lim_{a\to\infty} f_{\mathbf{r}}(\mathbf{x}_a) = f_{\mathbf{r}}(\mathbf{p})$.

Proof. First, note that if $\mathcal{P}(\mathbf{p}, \Sigma) = \{\mathbf{p}\}$, then any sequence in $\mathcal{P}(\mathbf{p}, \Sigma)$ is constant, and the result is trivially true. The rest of the argument, similar to the proof of Lemma C.1.2, deals with the case where $\mathcal{P}(\mathbf{p}, \Sigma)$ contains elements in addition to $\mathbf{p}$. Let $\mathbf{r} \in \mathcal{R}$ contain only elements from $\Sigma \subset S_n$. Let $\xi_{\mathbf{r}}$ be the Gittins index sequence against $\mathbf{p}$ that breaks ties using rule $\mathbf{r}$. For $k \in \mathbb{Z}^+$, let $\mathcal{P}_{k,\mathbf{r}} \subset \mathcal{P}$ contain precisely those mixed hiding strategies $\mathbf{x}$ for which the Gittins index sequence against $\mathbf{x}$ under rule $\mathbf{r}$ is identical to $\xi_{\mathbf{r}}$ for the first k searches. Clearly, for any $k \in \mathbb{Z}^+$, we have $\mathcal{P}_{k+1,\mathbf{r}} \subseteq \mathcal{P}_{k,\mathbf{r}}$ and $\mathbf{p} \in \mathcal{P}_{k,\mathbf{r}}$.

For any $\delta > 0$, write $B(\mathbf{p}, \delta)$ for the open ball with radius δ centred at $\mathbf{p}$. Note that any two points in $\mathcal{P}$ must be within Euclidean distance $\sqrt{n}$ of eachother. Therefore, for any $\mathbf{p} \in \mathcal{P}$, we must have $B(\mathbf{p}, \sqrt{n}) = \mathcal{P}$. For $\delta \in [0, \sqrt{n}]$, write $\mathcal{P}(\mathbf{p}, \delta, \Sigma) \equiv B(\mathbf{p}, \delta) \cap \mathcal{P}(\mathbf{p}, \Sigma)$. In other words, $\mathcal{P}(\mathbf{p}, \delta, \Sigma)$ is the subset of mixed hiding strategies in $\mathcal{P}(\mathbf{p}, \Sigma)$ strictly less than (Euclidean) distance δ from $\mathbf{p}$.

Write

$$\delta_{k,\mathbf{r}} \equiv 0.5 \times \sup \{\delta : \mathcal{P}(\mathbf{p}, \delta, \Sigma) \subseteq \mathcal{P}_{k,\mathbf{r}}\}, \tag{C.1.5}$$

so $\mathcal{P}(\mathbf{p}, \delta_{k,\mathbf{r}}, \Sigma) \subseteq \mathcal{P}_{k,\mathbf{r}}$. Note that $\delta_{k,\mathbf{r}} \geq 0$ for all $k \in \mathbb{Z}^+$ since $\mathcal{P}(\mathbf{p}, 0, \Sigma) = \{\mathbf{p}\} \in \mathcal{P}_{k,\mathbf{r}}$. The aim of the following is to show that $\delta_{k,\mathbf{r}} > 0$ for all $k \in \mathbb{Z}^+$.

Let $k \in \mathbb{Z}^+$. We examine two cases. First, suppose that, in the first k searches of $\xi_\mathbf{r}$, no ties are encountered. Then, it is possible, in any direction, to move a small enough (Euclidean) distance in $\mathcal{P}$ away from $\mathbf{p}$ and not disrupt the order of the terms in (7.2.2) that generate the first k searches of $\xi_\mathbf{r}$. Therefore, we may choose $\delta > 0$ such that $B(\mathbf{p}, \delta) \subset \mathcal{P}_{k,\mathbf{r}}$. It follows that $\mathcal{P}(\mathbf{p}, \delta, \Sigma) \subset \mathcal{P}_{k,\mathbf{r}}$, and hence that $\delta_{k,\mathbf{r}} \geq \delta/2 > 0$.

Second, suppose that, in the first k searches of $\xi_\mathbf{r}$, we do encounter ties between boxes. Suppose such a tie involves b boxes. By (7.2.2), whilst the order of the next b boxes searched may depend on the tie-breaking rule, the set of b boxes searched will not. Therefore, after the tie has been broken, the terms in (7.2.2) will be the same no matter how the tie was broken. Hence, it is possible, in any direction, to move a small enough (Euclidean) distance away in $\mathcal{P}$ from $\mathbf{p}$ and not disrupt the order of the terms in (7.2.2) that generate the first k searches of $\xi_\mathbf{r}$ at any point where there is not a tie between boxes. It follows that we may choose $\delta > 0$ such that, for any $\mathbf{x} \in B(\mathbf{p}, \delta)$, any Gittins index sequence against $\mathbf{x}$ differs only in the first k searches to $\xi_\mathbf{r}$ for those searches where $\xi_\mathbf{r}$ is in the process of breaking a tie. Now suppose additionally that $\mathbf{x} \in \mathcal{P}(\mathbf{p}, \Sigma)$, so $\mathbf{x} \in \mathcal{P}(\mathbf{p}, \delta, \Sigma)$. Since $\mathbf{x} \in \mathcal{P}(\mathbf{p}, \Sigma)$, when a tie is reached by $\xi_\mathbf{r}$, if we instead were following a Gittins index sequence against $\mathbf{x}$, any boxes involved in the tie will either still be tied, or lie in the ordering determined by σ for all $\sigma \in \Sigma$. Therefore, since $\mathbf{r}$ contains only elements of Σ, the Gittins index sequence against $\mathbf{x}$ that breaks ties using $\mathbf{r}$ will break the tie the same way as $\xi_\mathbf{r}$, so will be identical to $\xi_\mathbf{r}$ for the first k searches. In other words, $\mathcal{P}(\mathbf{p}, \delta, \Sigma) \subset \mathcal{P}_{k,\mathbf{r}}$, and hence $\delta_{k,\mathbf{r}} \geq \delta/2 > 0$.

Now we have shown $\delta_{k,\mathbf{r}} > 0$ for all $k \in \mathbb{Z}^+$, we are in a position to finish the proof in a similar style to Lemma C.1.2. Write $\delta^*_\mathbf{r} \equiv \lim_{k\to\infty} \delta_{k,\mathbf{r}}$, and let $\{\mathbf{x}_a : a \in \mathbb{Z}^+\}$ be a sequence in $\mathcal{P}(\mathbf{p}, \Sigma)$ with $\lim_{a\to\infty} \mathbf{x}_a = \mathbf{p}$. There are two cases.

First suppose that $\delta^*_{\mathbf{r}} > 0$. Let $\mathbf{x} \in \mathcal{P}(\mathbf{p}, \delta^*_{\mathbf{r}}, \Sigma)$; then $\mathbf{x} \in \mathcal{P}_{k,\mathbf{r}}$ for all $k \in \mathbb{Z}^+$, so the Gittins index sequence against $\mathbf{x}$ which breaks ties using rule $\mathbf{r}$ is identical to $\xi_{\mathbf{r}}$. It follows that $f_{\mathbf{r}}$ is constant on $\mathcal{P}(\mathbf{p}, \delta^*_{\mathbf{r}}, \Sigma)$. Furthermore, since $\lim_{a\to\infty} \mathbf{x}_a = \mathbf{p}$, there must exist A such that $\mathbf{x}_a \in B(\mathbf{p}, \delta^*_{\mathbf{r}})$ for all $a \geq A$. Yet, since $\{\mathbf{x}_a\}$ is a sequence in $\mathcal{P}(\mathbf{p}, \Sigma)$, we also, for all $a \geq A$, have $\mathbf{x}_a \in \mathcal{P}(\mathbf{p}, \delta^*_{\mathbf{r}}, \Sigma)$ and hence $f_{\mathbf{r}}(\mathbf{x}_a) = f_{\mathbf{r}}(\mathbf{p})$; proving the result for the first case.

Second, suppose that $\delta^*_{\mathbf{r}} = 0$. As in the proof of Lemma C.1.2, since $\{\mathbf{x}_a\}$ has limit $\mathbf{p}$ and, for any $k \in \mathbb{Z}^+$, $\delta_{k,\mathbf{r}} > 0$, there must exist a smallest number $g_{\mathbf{r}}(k) \in \mathbb{Z}^+$ such that every term in $\{\mathbf{x}_a\}$ after $\mathbf{x}_{g(k)}$ belongs to the ball $B(\mathbf{p}, \delta_{k,\mathbf{r}})$. Further, since $\{\mathbf{x}_a\}$ is a sequence in $\mathcal{P}(\mathbf{p}, \Sigma)$, every term in $\{\mathbf{x}_a\}$ after $\mathbf{x}_{g(k)}$ also belongs to $\mathcal{P}(\mathbf{p}, \delta_{k,\mathbf{r}}, \Sigma)$. Formally, for any $k \in \mathbb{Z}^+$, we write

$$g_{\mathbf{r}}(k) \equiv \min\{A : \mathbf{x}_a \in \mathcal{P}(\mathbf{p}, \delta_{k,\mathbf{r}}, \Sigma),\ a \geq A\}. \tag{C.1.6}$$

Note from (C.1.5) that, for any $l \in \mathbb{Z}^+$, since $\mathcal{P}_{k+l,\mathbf{r}} \subseteq \mathcal{P}_{k,\mathbf{r}}$, we have $\delta_{k,\mathbf{r}} \geq \delta_{k+l,\mathbf{r}}$. It follows that $B(\mathbf{p}, \delta_{k+l}) \subseteq B(\mathbf{p}, \delta_k)$, and hence $g_{\mathbf{r}}(k) \leq g_{\mathbf{r}}(k+l)$ for all $k \in \mathbb{Z}^+$, so the sequence $\{g_{\mathbf{r}}(k) : k \in \mathbb{Z}^+\}$ is increasing.

Consider the sequence $\{\mathbf{x}_{g_{\mathbf{r}}(k)} : k \in \mathbb{Z}^+\}$. We have $\lim_{a\to\infty} \mathbf{x}_a = \mathbf{p}$ by assumption; our next aim is to show that $\lim_{k\to\infty} \mathbf{x}_{g_{\mathbf{r}}(k)} = \mathbf{p}$ also. To do this, we show that, for any $\epsilon > 0$, we can choose K such that $\mathbf{x}_{g_{\mathbf{r}}(k)} \in B(\mathbf{p}, \epsilon)$ for all $k \geq K$. Choose $\epsilon > 0$. Since $\lim_{k\to\infty} \delta_{k,\mathbf{r}} = 0$, there exists K such that $\delta_{K,\mathbf{r}} < \epsilon$. By the definition of $g_{\mathbf{r}}(k)$ in (C.1.6), we have $\mathbf{x}_a \in B(\mathbf{p}, \delta_{K,\mathbf{r}})$ for all $a \geq g_{\mathbf{r}}(K)$. Since $g_{\mathbf{r}}$ is increasing, we have $\mathbf{x}_{g_{\mathbf{r}}(k)} \in B(\mathbf{p}, \delta_{K,\mathbf{r}}) \subset B(\mathbf{p}, \epsilon)$ for all $k \geq K$, showing that $\{\mathbf{x}_{g_{\mathbf{r}}(k)}\}$ has limit $\mathbf{p}$.

By the definitions in (C.1.5) and (C.1.6), we have $\mathbf{x}_{g_{\mathbf{r}}(k)} \in \mathcal{P}(\mathbf{p}, \delta_{k,\mathbf{r}}, \Sigma) \subseteq \mathcal{P}_{k,\mathbf{r}}$. Recall $\xi_{\mathbf{r}}$ as the Gittins index sequence against $\mathbf{p}$ that breaks ties using $\mathbf{r}$, and, for $a \in \mathbb{Z}^+$, let $\xi_{a,\mathbf{r}}$ be the Gittins index sequence against $\mathbf{x}_a$ which breaks ties using $\mathbf{r}$. Since $g_{\mathbf{r}}$ is increasing and $\mathbf{x}_{g_{\mathbf{r}}(k)} \in \mathcal{P}_{k,\mathbf{r}}$, as k increases, the first time when $\xi_{g_{\mathbf{r}}(k),\mathbf{r}}$ and $\xi_{\mathbf{r}}$ differ becomes increasingly later and later into the search, so the effect on the expected time to detection decreases to 0. Therefore, $V_i(\xi_{g_{\mathbf{r}}(k),\mathbf{r}}) \to V_i(\xi_{\mathbf{r}})$ as $k \to \infty$,

$i = 1, \ldots, n$, so we have

$$\lim_{k\to\infty} f_{\mathbf{r}}(\mathbf{x}_{g(k)}) = \lim_{a\to\infty} f_{\mathbf{r}}(\mathbf{x}_a) = f_{\mathbf{r}}(\mathbf{p}),$$

proving the result for the second case. ■

The continuity of the functions $f_{\mathbf{r}}$ is key in proving the main result: that $\bar{S}(\epsilon)$ is closed.

Proof that $\bar{S}(\epsilon)$ is closed. Write $\{\mathbf{s}_a : a \in \mathbb{Z}^+\}$ for a convergent sequence in $\bar{S}(\epsilon)$, and write $\mathbf{s}_0 \equiv \lim_{a\to\infty} \mathbf{s}_a$ for its limit. To show that $\bar{S}(\epsilon)$ is closed, we need to show that $\mathbf{s}_0 \in \bar{S}(\epsilon)$.

Since each element of $\bar{S}(\epsilon)$ corresponds to some mixed hiding strategy $\mathbf{p} \in \mathcal{P}(\epsilon)$ and tie-breaking rule $\mathbf{r} \in \mathcal{R}$, an equivalent definition of $\bar{S}(\epsilon)$ is written by

$$\bar{S}(\epsilon) \equiv \bigcup_{\mathbf{r}\in\mathcal{R}} \left\{ f_{\mathbf{r}}(\mathcal{P}(\epsilon)) \right\}, \tag{C.1.7}$$

where $f_{\mathbf{r}}(\mathcal{P}(\epsilon))$ is the image of $\mathcal{P}(\epsilon)$ under $f_{\mathbf{r}}$. The images under different $\mathbf{r}$ are not disjoint, however. In particular, if $\mathbf{p}$ is a non-tie point, then $f_{\mathbf{r}}(\mathbf{p})$ are equivalent for all $\mathbf{r} \in \mathcal{R}$. For a tie point $\mathbf{p}$, however, some of the $f_{\mathbf{r}}(\mathbf{p})$ will differ.

By (C.1.7), for all $a \in \mathbb{Z}^+$, we may choose $\mathbf{x}_a \in \mathcal{P}(\epsilon)$ and $\mathbf{r}_a \in \mathcal{R}$ such that $\mathbf{s}_a = f_{\mathbf{r}_a}(\mathbf{x}_a)$. Further, since $\mathcal{P}(\epsilon)$ is bounded, the sequence $\{\mathbf{x}_a : a \in \mathbb{Z}^+\}$ has a convergent subsequence $\{\mathbf{x}_{h(a)} : a \in \mathbb{Z}^+\}$, and, since $\mathcal{P}(\epsilon)$ is closed, $\mathbf{x}_0 \equiv \lim_{a\to\infty} \mathbf{x}_{h(a)} \in \mathcal{P}(\epsilon)$. Any infinite subsequence of the convergent sequence $\{\mathbf{s}_a\}$ must converge to the same limit as $\{\mathbf{s}_a\}$; therefore, we have $\mathbf{s}_0 = \lim_{a\to\infty} \mathbf{s}_{h(a)}$.

We consider two cases. First, suppose that $\{\mathbf{x}_{h(a)}\}$ attains its limit $\mathbf{x}_0$. In other words, there exists $A \in \mathbb{Z}^+$ such that $\mathbf{x}_{h(a)} = \mathbf{x}_0$ for all $a \geq A$. In this instance, the sequence $\{\mathbf{s}_{h(a)} : a \geq A\}$, which has limit $\mathbf{s}_0$, is a sequence in the set $S(\mathbf{x}_0)$, defined in (C.1.1), shown to be closed by Lemma C.1.1. Therefore, $\mathbf{s}_0 \in S(\mathbf{x}_0) \subset \bar{S}(\epsilon)$, completing the proof for the first case.

Second, suppose that $\{\mathbf{x}_{h(a)}\}$ does not attain its limit $\mathbf{x}_0$. Since $\mathbf{x}_0 \in \mathcal{P}(\epsilon)$, we have $f_{\mathbf{r}}(\mathbf{x}_0) \in \bar{S}(\epsilon)$ for any $\mathbf{r} \in \mathcal{R}$. To complete the proof for the second case, we show that $\mathbf{s}_0 = f_{\mathbf{r}}(\mathbf{x}_0)$ for some $\mathbf{r} \in \mathcal{R}$. To do this, we split the second case into two subcases.

First, consider the easier subcase in which $\mathbf{x}_0$ is a non-tie point. Then, for any $\mathbf{r} \in \mathcal{R}$, we have

$$\mathbf{s}_0 = \lim_{a\to\infty} \mathbf{s}_{h(a)} = \lim_{a\to\infty} f_{\mathbf{r}_{h(a)}}(\mathbf{x}_{h(a)}) = f_{\mathbf{r}}(\mathbf{x}_0),$$

where the last equality follows by Lemma C.1.2 and the fact that the $f_{\mathbf{r}}(\mathbf{x}_0)$ are equal for all $\mathbf{r} \in \mathcal{R}$.

The rest of the proof concerns the more challenging subcase, in which $\mathbf{x}_0 = (x_{0,1}, \ldots, x_{0,n})$ is a tie point. Note that, for any $\mathbf{x} \in \mathcal{P}$, there exists a subset $\Sigma_{\mathbf{x}} \subseteq S_n$ for which $\sigma \in \Sigma_{\mathbf{x}}$ if and only if

$$x_{\sigma(i)}/x_{0,\sigma(i)} \geq x_{\sigma(i+1)}/x_{0,\sigma(i+1)}, \; i = 1, 2, \ldots, n-1.$$

Recalling the definition in (C.1.4), we have $\mathbf{x} \in \mathcal{P}(\mathbf{x}_0, \Sigma_{\mathbf{x}})$, and further, $\Sigma_{\mathbf{x}}$ is the unique subset of maximal size such that $\mathbf{x} \in \mathcal{P}(\mathbf{x}_0, \Sigma_{\mathbf{x}})$. Since the elements $\{x_i/x_{0,i} : i \in \{1, \ldots n\}\}$ must lie in some order, $\Sigma_{\mathbf{x}}$ is non-empty for any $\mathbf{x} \in \mathcal{P}(\epsilon)$.

Since there a finite number of subsets of S_n, there must exist $\Sigma^* \subset S_n$ and a convergent subsequence of $\{\mathbf{x}_{h(a)}\}$, say $\{\mathbf{x}_m : m \in \mathbb{Z}^+\}$, such that $\Sigma_{\mathbf{x}_m} = \Sigma^*$ for all $m \in \mathbb{Z}^+$. In other words, $\{\mathbf{x}_m\}$ is a sequence in $\mathcal{P}(\mathbf{x}_0, \Sigma^*)$.

Since $\{\mathbf{x}_m\}$ is a convergent subsequence of $\{\mathbf{x}_{h(a)}\}$, we have $\lim_{m\to\infty} \mathbf{x}_m = \mathbf{x}_0$, and

$$\lim_{m\to\infty} f_{\mathbf{r}_m}(\mathbf{x}_m) = \lim_{m\to\infty} \mathbf{s}_m = \mathbf{s}_0.$$

To finish the proof, we split into further subcases, numbered below.

1. First, we consider the easier subcase in which there are finitely many tie points in $\{\mathbf{x}_m\}$. In this case, we may choose M such that there are no tie points in the sequence $\{\mathbf{x}_m : m \geq M\}$. Therefore, for all $m \geq M$, we have $f_{\mathbf{r}_m}(\mathbf{x}_m) = f_{\mathbf{r}}(\mathbf{x}_m)$ for all $\mathbf{r} \in \mathcal{R}$. Let $\mathbf{r}^* \in \mathcal{R}$ contain only elements in Σ^*. Then we have

$$\mathbf{s}_0 = \lim_{m\to\infty} f_{\mathbf{r}_m}(\mathbf{x}_m) = \lim_{m\to\infty} f_{\mathbf{r}^*}(\mathbf{x}_m) = f_{\mathbf{r}^*}(\mathbf{x}_0),$$

   where the last equality follows by Lemma C.1.3.

2. Now suppose that there are infinitely many tie points in $\{\mathbf{x}_m\}$. We begin with two observations for any $\mathbf{x} \in \mathcal{P}(\epsilon)$.

   Observation 1: First, note that the position of any equalities in the ordering of the terms $\{x_i/x_{0,i} : i \in \{1,\ldots,n\}\}$ completely determines $\Sigma_{\mathbf{x}}$. In particular, for any pair of boxes $i, j \in \{1,\ldots,n\}$, we have $x_i/x_{0,i} = x_j/x_{0,j}$ if and only if there exists $\sigma_1, \sigma_2 \in \Sigma_{\mathbf{x}}$ with $\sigma_1(i) > \sigma_1(j)$ and $\sigma_2(j) > \sigma_2(i)$. Also, we have $x_i/x_{0,i} > x_j/x_{0,j}$ if and only if $\sigma(i) > \sigma(j)$ for all $\sigma \in \Sigma_{\mathbf{x}}$.

   Observation 2: Second, for any pair of boxes $i, j \in \{1,\ldots,n\}$ and any $y, z \in \mathbb{N} \equiv \{0,1,\ldots\}$, write
   $$k_{i,j}(y,z) \equiv \frac{q_j(1-q_j)^z t_i}{q_i(1-q_i)^y t_j},$$
   recalling that q_i (resp. t_i) is the detection probability (resp. search time) of box i, $i = 1,\ldots,n$. Then, following a Gittins index sequence against $\mathbf{x}$, there is a tie between boxes i and j after y (resp. z) searches of box i (resp. j) have been made if and only if $x_i/x_j = k_{i,j}(y,z)$.

   Now consider the sequence $\{\mathbf{x}_m\}$. Since $\{\mathbf{x}_m\}$ has limit $\mathbf{x}_0$, for any two boxes $i, j \in \{1,\ldots,n\}$, we have
   $$\frac{x_{m,i}}{x_{m,j}} \to \frac{x_{0,i}}{x_{0,j}} \quad \text{as} \quad m \to \infty. \tag{C.1.8}$$
   Now choose $\mathbf{x} \in \{\mathbf{x}_m\}$ and suppose that $c, d \in \{1,\ldots,n\}$ satisfy $x_c/x_{0,c} \neq x_d/x_{0,d}$. Then, by Observation 1, the same must be true for every element of $\{\mathbf{x}_m\}$ since $\Sigma_{\mathbf{x}_m} = \Sigma^*$ for all $m \in \mathbb{Z}^+$. Hence, the limit in (C.1.8) is never attained for $i, j = c, d$. In other words, $x_{m,c}/x_{m,d}$ approaches but never reaches $x_{0,c}/x_{0,d}$ as $m \to \infty$.

   Let $y, z \in \mathbb{N}$ and consider $k_{c,d}(y,z)$ defined in Observation 2. There are two scenarios. First, we may have $k_{c,d}(y,z) = x_{0,c}/x_{0,d}$; in this scenario, since the limit in (C.1.8) is never attained, in no Gittins index sequence against any element of $\{\mathbf{x}_m\}$ is there a tie between boxes c and d after y (resp. z) searches of box c (resp. d) have been made. Second, if $k_{c,d}(y,z) \neq x_{0,c}/x_{0,d}$, by the limit in (C.1.8), there exists a finite smallest natural number, say $M_{c,d}(y,z)$, such that the same statement holds after the $M_{c,d}(y,z)$th term of $\{\mathbf{x}_m\}$; in other words, in no Gittins index sequence against any element of $\{\mathbf{x}_m : M_{c,d}(y,z) \geq m\}$ is there a tie between boxes c and d after y (resp. z) searches of box c (resp. d) have been made.

For any $b \in \mathbb{Z}^+$, write

$$M_{c,d}^b \equiv \max\{M_{c,d}(y,z) : y,z \in \mathbb{N} \text{ with } y+z \leq b\}.$$

It follows that, whilst the total number of searches of boxes c and d is no larger than b, no ties involving *both* boxes c and d are encountered in a Gittins index sequence against any element of $\{\mathbf{x}_m : m \geq M_{c,d}^b\}$. Clearly $\{M_{c,d}^b : b \in \mathbb{Z}^+\}$ forms an increasing sequence; therefore, as $m \to \infty$, any tie involving both boxes c and d occurs increasingly later and later into the search, so the effect on the expected time to detection of how such a tie is broken decreases to 0.

Therefore, as $m \to \infty$, only the manner in which ties involving *only* boxes i and j satisfying $x_i/x_{0,i} = x_j/x_{0,j}$ are broken has any effect on the expected time to detection under a Gittins index sequence against an element of $\{\mathbf{x}_m\}$. Without a loss of generality, suppose such a tie is between boxes $1,\ldots,y$, for some $y \in \{2,\ldots,n\}$. Suppose the tie is broken by $\sigma \in S_n$. By Observation 1, since $x_1/x_{0,1} = \cdots = x_y/x_{0,y}$, there exists $\sigma^* \in \Sigma_{\mathbf{x}}$ which ranks boxes $1,\ldots,y$ in the same order as σ. Therefore, breaking the tie using σ^* leads to boxes $1,\ldots,y$ being searched in the same order as breaking the tie using σ. It follows that there exists $\mathbf{r}^*$ with only elements in Σ^* such that, if, for all $m \in \mathbb{Z}^+$, we replace $\mathbf{r}_m$ with $\mathbf{r}^*$, as $m \to \infty$, the effect on the expected time to detection tends to 0. In other words,

$$\mathbf{s}_0 = \lim_{m\to\infty} f_{\mathbf{r}_m}(\mathbf{x}_m) = \lim_{m\to\infty} f_{\mathbf{r}^*}(\mathbf{x}_m) = f_{\mathbf{r}^*}(\mathbf{x}_0),$$

where the last equality follows by Lemma C.1.3. ■

C.2 Proof of Lemma 7.3.3

Proof. If $|C(\mathbf{p})| = 1$ then $S(\mathbf{p}) = \widehat{S}(\mathbf{p})$ and the result is trivially true. For the rest of the proof, assume $|C(\mathbf{p})| > 1$.

First, we show that $S(\mathbf{p})$ is compact. Since $S(\mathbf{p}) \subset \mathbb{R}^n$, by the Heine-Borel theorem, $S(\mathbf{p})$ is compact if and only if it is both closed and bounded. By Lemma C.1.1 in Appendix C.1, $S(\mathbf{p})$ is closed. To show $S(\mathbf{p})$ is bounded, let

$$v(\mathbf{p}, \xi) \equiv \sum_{i=1}^{n} p_i V_i(\xi)$$

be the expected time to detection if the hider uses $\mathbf{p}$ and the searcher uses ξ. Consider the search sequence ξ_1 that repeats the cycle of searches $(1, 2, \ldots, n)$ indefinitely. Clearly $V_i(\xi_1)$ is finite for $i = 1, \ldots, n$, so $v(\mathbf{p}, \xi_1)$ is also finite. Since $C(\mathbf{p})$ is the set of optimal counters to $\mathbf{p}$, we must have $v(\mathbf{p}, \xi) \leq v(\mathbf{p}, \xi_1)$ for any $\xi \in C(\mathbf{p})$. Since $p_i > 0$, we must have $V_i(\xi)$ finite for $i = 1, \ldots, n$ and any $\xi \in C(\mathbf{p})$; it follows that $S(\mathbf{p})$ is bounded and hence compact.

Let A be the convex hull of $S(\mathbf{p})$. By definition A is convex, and, since $S(\mathbf{p})$ is compact in the finite-dimensional vector space $\mathbb{R}_n$, A is also compact. (See Corollary 5.33 of Charalambos and Aliprantis (2013).) Therefore, we may apply the Krein-Milman theorem to deduce that A is the convex hull of its extreme points. To prove Lemma 7.3.3, we show that $\widehat{S}(\mathbf{p})$ is the set of extreme points of A. We first note the following useful facts.

- By definition, a point $\mathbf{x} \in A$ is extreme if and only if, for any $\mathbf{y}, \mathbf{z} \in A$ and $\lambda \in [0, 1]$ satisfying $\mathbf{x} = \lambda\mathbf{y} + (1 - \lambda)\mathbf{z}$, we have $\mathbf{x} = \mathbf{y} = \mathbf{z}$. In other words, the only way we can express $\mathbf{x}$ as a convex combination of elements of A is by $\mathbf{x}$ itself.

- For any $\mathbf{s} \equiv (s_1, \ldots, s_n) \in S(\mathbf{p})$, the weighted average $\sum_{i=1}^n s_i p_i$ is equal to the expected time to detection if the hider chooses $\mathbf{p}$ and the searcher any optimal counter $\xi \in C(\mathbf{p})$. Therefore, all elements of $S(\mathbf{p})$ lie on the same hyperplane, say H, in $\mathbb{R}^n$.

- By the definition of $S(\mathbf{p})$, we have

  $$S(\mathbf{p}) \subset B \equiv \left\{ (v_1, \ldots, v_n) \in \mathbb{R}^n : \min_{\xi \in C(\mathbf{p})} V_i(\xi) \leq v_i \leq \max_{\xi \in C(\mathbf{p})} V_i(\xi),\ i = 1, \ldots, n \right\}.$$

  Further, since A is the smallest convex set containing $S(\mathbf{p})$, and B is also a convex set containing $S(\mathbf{p})$, we have $A \subseteq B$.

The proof will be done by double inclusion. In the first half of the double inclusion proof, we show that any point in $\widehat{S}(\mathbf{p})$ is an extreme point of A. Write $\mathbf{x} \equiv (x_1, \ldots, x_n) \in \widehat{S}(\mathbf{p})$. Then $\mathbf{x}$ corresponds to a search sequence $\xi_\sigma \in \widehat{C}(\mathbf{p})$ which breaks all ties using some $\sigma \in S_n$. Without a loss of generality, let $\sigma \equiv (1, 2, \ldots, n)$. Suppose that

$$\mathbf{x} = \lambda \mathbf{y} + (1 - \lambda)\mathbf{z} \tag{C.2.1}$$

for some $\mathbf{y}, \mathbf{z} \in A$ and $\lambda \in [0, 1]$. To prove that $\mathbf{x}$ is extreme, we show that we must have $\mathbf{x} = \mathbf{y} = \mathbf{z}$.

Since, when breaking any tie, ξ_σ gives preference to box 1 over any other box, no other sequence in $C(\mathbf{p})$ makes the jth search of box 1 any sooner than ξ_σ, for $j \in \{1, 2, \ldots\}$; therefore, we have $x_1 = \min_{\xi \in C(\mathbf{p})} V_1(\xi)$. For any $\mathbf{v} \equiv (v_1, \ldots, v_n) \in A$, since $A \subset B$, we must have $v_1 \geq x_1$. It follows that, for (C.2.1) to hold, we must have $y_1 = z_1 = x_1$.

We now demonstrate how this argument may be repeated to show that $y_2 = z_2 = x_2$. Let $C_1(\mathbf{p})$ be the elements of $C(\mathbf{p})$ which, when breaking any tie involving box 1, give preference to box 1; therefore, for any $\xi \in C(\mathbf{p})$, we have $V_1(\xi) = x_1$ if and only if $\xi \in C_1(\mathbf{p})$. Write $V(\xi) \equiv (V_1(\xi), \ldots, V_n(\xi))$ and $S_1(\mathbf{p}) \equiv \{V(\xi) : \xi \in C_1(\mathbf{p})\}$; therefore, for any $\mathbf{v} = (v_1, \ldots, v_n) \in S(\mathbf{p})$, we have $v_1 = x_1$ if and only if $\mathbf{v} \in S_1(\mathbf{p})$.

Let A_1 be the convex hull of $S_1(\mathbf{p})$; since $S_1(\mathbf{p}) \subset S(\mathbf{p})$, we have $A_1 \subset A$. Note that $\mathbf{y}, \mathbf{z} \in A_1$ since $y_1 = z_1 = x_1$.

Write

$$B_1 \equiv \left\{ (x_1, v_2, \ldots, v_n) \in \mathbb{R}^n : \min_{\xi \in C_1(\mathbf{p})} V_i(\xi) \leq v_i \leq \max_{\xi \in C_1(\mathbf{p})} V_i(\xi),\ i = 2, \ldots, n \right\}.$$

Then B_1 is a convex set containing $S_1(\mathbf{p})$, so $A_1 \subset B_1$. Since, when breaking any tie, ξ_σ gives preference to box 1 over box 2, but then to box 2 over any other box i, $i = 3, \ldots, n$, no other sequence in $C_1(\mathbf{p})$ makes the jth search of box 2 sooner than ξ_σ, $j \in \{1, 2, \ldots\}$; therefore, we have $x_2 = \min_{\xi \in C_1(\mathbf{p})} V_2(\xi)$. Since any $\mathbf{v} \in A_1$ also belongs to B_1, we must have $v_2 \geq x_2$. It follows that, for (C.2.1) to hold, we must have $y_2 = z_2 = x_2$.

We may repeat the above argument a further $n - 3$ times to conclude that $y_i = z_i = x_i$ for $i = 1, \ldots, n - 1$. Finally, since $\mathbf{y}, \mathbf{z}$ and $\mathbf{x}$ all lie in the same hyperplane H in $\mathbb{R}^n$, we must have $y_n = z_n = x_n$, so $\mathbf{y} = \mathbf{z} = \mathbf{x}$, and $\mathbf{x}$ is an extreme point of A.

In the second half of the double inclusion proof, we show that any extreme point of A is in $\widehat{S}(\mathbf{p})$. We will prove the contrapositive of this statement; i.e., we show that any $\mathbf{a} \in A \setminus \widehat{S}(\mathbf{p})$ is not an extreme point of A.

To begin, note that, by definition, any element of A can be written as a convex combination of some $m \in \{1, 2, \ldots\}$ elements of $S(\mathbf{p})$. If $\mathbf{b} \in A \setminus S(\mathbf{p})$, then any such convex combination must contain at least $m \geq 2$ elements of $S(\mathbf{p})$, so $\mathbf{b}$ is not an extreme point of A. Hence, our task is reduced to showing that any $\mathbf{a} \in S(\mathbf{p}) \setminus \widehat{S}(\mathbf{p})$ is not an extreme point of A.

Consider $\mathbf{a} \in S(\mathbf{p}) \setminus \widehat{S}(\mathbf{p})$, and write $\xi_\mathbf{a}$ for the corresponding search sequence in $C(\mathbf{p}) \setminus \widehat{C}(\mathbf{p})$. Since $\xi_\mathbf{a} \notin \widehat{C}(\mathbf{p})$, there exists no permutation in S_n with which $\xi_\mathbf{a}$ breaks every tie it encounters. It follows that there must exist some $k \in \{2, \ldots, n\}$ boxes and k ties encountered by $\xi_\mathbf{a}$ such that the preference ordering among these k boxes used to collectively break the k ties constitutes a cycle. Without loss of generality, assume boxes $1, 2, \ldots, k$ are involved in this cycle. Suppose tie m involves boxes m and $m + 1$, with box m searched before box $m + 1$ by $\xi_\mathbf{a}$, $m = 1, \ldots, k - 1$, and tie

k involves boxes k and 1, with box k is searched before box 1 by $\xi_{\mathbf{a}}$. Note that ties $1, \ldots, k$ are not necessarily consecutive nor in chronological order.

Consider tie k, which uses the preference ordering

$$j_1, \ldots, j_{\alpha-1}, k, j_{\alpha+1}, \ldots, j_{\beta-1}, 1, j_{\beta+1}, \ldots, j_n,$$

bearing in mind that box j_i is not necessarily involved in tie k, $i \in \{1, \ldots, n\} \setminus \{\alpha, \beta\}$. In other words, box k ranks in the αth position, and box 1 ranks in the βth position, for some $1 \leq \alpha < \beta \leq n$.

Let $\xi_\alpha \in C(\mathbf{p})$ break tie k using preference ordering

$$j_1, \ldots, j_{\alpha-1}, 1, k, j_{\alpha+1}, \ldots, j_{\beta-1}, j_{\beta+1}, \ldots, j_n,$$

and all other ties in the same order as $\xi_{\mathbf{a}}$. Similarly, let $\xi_\beta \in C(\mathbf{p})$ break tie k using preference ordering

$$j_1, \ldots, j_{\alpha-1}, j_{\alpha+1}, \ldots, j_{\beta-1}, 1, k, j_{\beta+1}, \ldots, j_n,$$

and all other ties in the same order as $\xi_{\mathbf{a}}$. In other words, at tie k, both ξ_α and ξ_β switch the order boxes 1 and k are searched by $\xi_{\mathbf{a}}$ to prefer box 1, and all boxes searched between boxes 1 and k by $\xi_{\mathbf{a}}$ when breaking tie k have (retaining their order) been shifted after boxes 1 and k are searched in ξ_α, and before boxes 1 and k are searched in ξ_β. Note that if $\beta = \alpha + 1$ (so there are no boxes searched between boxes k and 1 when $\xi_{\mathbf{a}}$ breaks tie k), then $\xi_\alpha = \xi_\beta$, but the following argument is still valid.

Note that $V_i(\xi_\alpha) = V_i(\xi_\beta) = V_i(\xi_{\mathbf{a}})$ for any box $i \in \{j_1, \ldots, j_{\alpha-1}, j_{\beta+1}, \ldots, j_n\}$. Recall that t_i stands for the search time of box i, and let

$$\eta_k \equiv \frac{t_k}{t_1 + t_k} \xi_\alpha \oplus \frac{t_1}{t_1 + t_k} \xi_\beta.$$

Clearly $V_i(\eta_k) = V_i(\xi_{\mathbf{a}})$ for $i \in \{j_1, \ldots, j_{\alpha-1}, j_{\beta+1}, \ldots, j_n\}$. For $i \in \{j_{\alpha+1}, \ldots, j_{\beta-1}\}$, let w_i be the probability that the hider is found on the first search of box i after the kth tie is reached, conditional on the hider being in box i. Then we have $V_i(\xi_\alpha) = V_i(\xi_{\mathbf{a}}) + w_i t_1$ and $V_i(\xi_\beta) = V_i(\xi_{\mathbf{a}}) - w_i t_k$ for $i \in \{j_{\alpha+1}, \ldots, j_{\beta-1}\}$. It follows that

$$V_i(\eta_k) = \frac{t_k}{t_1 + t_k}\left(V_i(\xi_{\mathbf{a}}) + w_i t_1\right) + \frac{t_1}{t_1 + t_k}\left(V_i(\xi_{\mathbf{a}}) - w_i t_k\right) = V_i(\xi_{\mathbf{a}})$$

for $i \in \{j_{\alpha+1}, \ldots, j_{\beta-1}\}$.

Because $V_k(\xi_\beta) > V_k(\xi_\alpha) > V_k(\xi_{\mathbf{a}})$, we have that $V_k(\eta_k) > V_k(\xi_{\mathbf{a}})$. Since $V(\eta_k)$ and $V(\xi_{\mathbf{a}})$ lie in the same hyperplane, H, in $\mathbb{R}^n$, we must have $V_1(\eta_k) < V_1(\xi_{\mathbf{a}})$. To summarise, we have

$$V_k(\eta_k) > V_k(\xi_{\mathbf{a}}), \quad V_1(\eta_k) < V_1(\xi_{\mathbf{a}}) \quad \text{and} \quad V_i(\eta_k) = V_i(\xi_{\mathbf{a}}) \text{ for } i \neq 1, k. \tag{C.2.2}$$

Now, for $m \in \{1, \ldots, k-1\}$, we repeat the same procedure with tie m for boxes m and $m+1$ to create η_m which satisfies:

$$V_m(\eta_m) > V_m(\xi_{\mathbf{a}}), \quad V_{m+1}(\eta_m) < V_{m+1}(\xi_{\mathbf{a}}) \quad \text{and} \quad V_i(\eta_m) = V_i(\xi_{\mathbf{a}}) \text{ for } i \neq m, m+1. \tag{C.2.3}$$

To complete the proof, we show how we may express $\xi_{\mathbf{a}}$ as a mixture of $\{\eta_1, \ldots, \eta_k\}$. To begin, by (C.2.3) with $m = 1, 2$, for any $\lambda \in [0, 1]$, the mixture

$$\eta_{1,2}(\lambda) \equiv \lambda\eta_1 \oplus (1-\lambda)\eta_2 \tag{C.2.4}$$

satisfies

$$V_1(\eta_{1,2}(\lambda)) > V_1(\xi_{\mathbf{a}}), \quad V_3(\eta_{1,2}(\lambda)) < V_3(\xi_{\mathbf{a}}) \quad \text{and} \quad V_i(\eta_{1,2}(\lambda)) = V_i(\xi_{\mathbf{a}}) \text{ for } i = 4, \ldots, n.$$

Also by (C.2.3), there exists $\lambda^* \in [0, 1]$ such that $\eta_{1,2} \equiv \eta_{1,2}(\lambda^*)$ satisfies $V_2(\eta_{1,2}) = V_2(\xi_{\mathbf{a}})$. Therefore, we have

$$V_1(\eta_{1,2}) > V_1(\xi_{\mathbf{a}}), \quad V_3(\eta_{1,2}) < V_3(\xi_{\mathbf{a}}) \quad \text{and} \quad V_i(\eta_{1,2}) = V_i(\xi_{\mathbf{a}}) \text{ for } i \neq 1, 3. \tag{C.2.5}$$

By (C.2.3) with $m = 3$ and (C.2.5), there exists a mixture, $\eta_{1,2,3}$, of η_3 and $\eta_{1,2}$ satisfying

$$V_1(\eta_{1,2,3}) > V_1(\xi_{\mathbf{a}}), \quad V_4(\eta_{1,2,3}) < V_4(\xi_{\mathbf{a}}) \quad \text{and} \quad V_i(\eta_{1,2,3}) = V_i(\xi_{\mathbf{a}}) \text{ for } i \neq 1, 4. \tag{C.2.6}$$

We may repeat this process of mixing η_m and $\eta_{1,\ldots,m-1}$ to create $\eta_{1,\ldots,m}$ for $m = 4, \ldots k-1$, with the resulting $\eta_{1,\ldots,k-1}$ satisfying

$$V_1(\eta_{1,\ldots,k-1}) > V_1(\xi_{\mathbf{a}}), \quad V_k(\eta_{1,\ldots,k-1}) < V_k(\xi_{\mathbf{a}}) \quad \text{and} \quad V_i(\eta_{1,\ldots,k-1}) = V_i(\xi_{\mathbf{a}}) \text{ for } i \neq 1, k. \tag{C.2.7}$$

Finally, by (C.2.2) and (C.2.7), we may mix η_k and $\eta_{1,\ldots,k-1}$ to create $\eta_{1,\ldots,k}$ satisfying $V_i(\eta_{1,\ldots,k}) = V_i(\xi_{\mathbf{a}})$ for $i = 2, \ldots, n$. Yet, since $V(\eta_{1,\ldots,k})$ and $V(\eta_k)$ both lie in H, we must also have $V_1(\eta_{1,\ldots,k}) = V_1(\xi_{\mathbf{a}})$. It follows that, for some $\bar{\lambda} \in [0,1]$, we have

$$\mathbf{a} = V(\xi_{\mathbf{a}}) = V(\eta_{1,\ldots,k}) = \bar{\lambda} V(\eta_{1,\ldots,k}) + (1-\bar{\lambda}) V(\eta_k).$$

By the construction of the η_m for $m = 1, \ldots, k$ as mixtures of elements of $S(\mathbf{p})$, $V(\eta_{1,\ldots,k})$ and $V(\eta_k)$ are two distinct elements in A different from $\mathbf{a}$, showing that $\mathbf{a}$ is not an extreme point of A and completing the proof. ■

Concluding Remarks for Parts I and II

The thesis has studied two important extensions to the classic search problem in discrete locations, both motivated by advances in technology: Part I on the searcher's side, with more ways to search each location, and Part II on the hider's side, with increasingly frequent and sophisticated cyber attacks placing more importance on the search game. Whilst Chapters 4 and 8 respectively draw individual conclusions for Parts I and II, here, some concluding remarks are made on the thesis as a whole.

While Gittins index policies play a key part in both Parts I and II, the focus of the searcher differs greatly between the two parts. In Part I, the searcher exploits areas of the search space most attractive to them. In Part II, the hider all but takes away the notion of one area being more attractive than another, with the searcher's focus on randomising their strategy to guard against being taken advantage of by the hider. However, the two parts do share some common elements. The future benefit, concerning the information gained about the hidden target's location from an unsuccessful search, is crucial in Part I and explains several observations in the numerical study of Part II. The searcher's dislike of having no preference of where to search next, crucial in optimal hiding strategies in Part II, explains why a seemingly dominated search mode is sometimes optimal in Part I.

It is hoped that the key theoretical results in both parts, in addition to the insights explaining the theory and its shortfalls, provide a platform for further study.

Bibliography

Ahuja, R., Kumar, A., Jha, K., and Orlin, J. (2007). Exact and heuristic algorithms for the weapon-target assignment problem. *Operations Research*, 55:1136–1146.

Alpern, S. (1974). The search game with mobile hider on the circle. *Differential games and control theory*, pages 181–200.

Alpern, S. (1976). Hide and seek games. In *Seminar, Institut für höhere Studien, Wien*, volume 26.

Alpern, S. (1985). Search for point in interval, with high–low feedback. *Mathematical Proceedings of the Cambridge Philosophical Society*, 98(3):569–578.

Alpern, S. (2008). Hide-and-seek games on a tree to which eulerian networks are attached. *Networks*, 52(3):162–166.

Alpern, S. (2010). Search games on trees with asymmetric travel times. *SIAM Journal on Control and Optimization*, 48(8):5547–5563.

Alpern, S. (2011). Find-and-fetch search on a tree. *Operations Research*, 59(5):1258–1268.

Alpern, S. (2017). Hide-and-seek games on a network, using combinatorial search paths. *Operations Research*, 65(5):1207–1214.

Alpern, S., Baston, V., and Gal, S. (2008). Network search games with immobile hider, without a designated searcher starting point. *International Journal of Game Theory*, 37(2):281–302.

Alpern, S., Fokkink, R., Gasieniec, L., Lindelauf, R., and Subrahmanian, V. (2013). *Search Theory: A Game Theoretic Perspective*. Springer Publishing Company, Incorporated.

Alpern, S., Fokkink, R., Lindelauf, R., and Olsder, G. J. (2009). *A Numerical Approach to the 'Princess and Monster' Game on an Interval*, pages 1–9. Birkhäuser Boston, Boston.

Alpern, S. and Gal, S. (1988). A mixed-strategy minimax theorem without compactness. *SIAM J. Control Optim.*, 26(6):1357–1361.

Alpern, S. and Gal, S. (2003). *The Theory of Search Games and Rendezvous*. International Series in Operations Research and Managment Science. Kluwer Academic Publishers, Boston, Dordrecht, London.

Alpern, S., Gal, S., Lee, V., and Casas, J. (2019). A stochastic game model of searching predators and hiding prey. *Journal of The Royal Society Interface*, 16(153):20190087.

Alpern, S. and Lidbetter, T. (2013). Mining coal or finding terrorists: The expanding search paradigm. *Operations Research*, 61(2):265–279.

Alpern, S. and Lidbetter, T. (2014). Searching a variable speed network. *Mathematics of Operations Research*, 39(3):697–711.

Alpern, S. and Lidbetter, T. (2015). Optimal trade-off between speed and acuity when searching for a small object. *Oper. Res.*, 63(1):122–133.

Anderson, E. J. and Weber, R. R. (1990). The rendezvous problem on discrete locations. *Journal of Applied Probability*, 27(4):839–851.

Arkin, V. I. (1964). Uniformly optimal strategies in search problems. *Theory of Probability and its Applications*, 9(4):674–7.

Assaf, D. and Zamir, S. (1985). Optimal sequential search: A Bayesian approach. *The Annals of Statistics*, 13(3):1213–1221.

Assaf, D. and Zamir, S. (1987). Continuous and discrete search for one of many objects. *Operations Research Letters*, 6(5):205 – 209.

Australian Transport Safety Bureau (2017). The operational search for mh370. `https://www.atsb.gov.au/media/5773565/operational-search-for-mh370_final_3oct2017.pdf`.

Baston, V. and Garnaev, A. (2000). A search game with a protector. *Naval Research Logistics*, 47(2):85–96.

Baston, V. and Kikuta, K. (2013). Search games on networks with travelling and search costs and with arbitrary searcher starting points. *Networks*, 62(1):72–79.

Baston, V. and Kikuta, K. (2015). Search games on a network with travelling and search costs. *International Journal of Game Theory*, 44(2):347–365.

Baykal-Gürsoy, M. (2010). Semi-markov decision processes. *Wiley Encyclopedia of Operations Research and Management Science*.

Beck, A. (1964). On the linear search problem. *Israel J. Math*, 2:221–228.

Beck, A. (1965). More on the linear search problem. *Israel J. Math*, 3:61–70.

Beck, A. and Beck, M. (1984). Son of the linear search problem. *Israel J. Math*, 48:109–122.

Beck, A. and Newman, D. J. (1970). Yet more on the linear search problem. *Israel Journal of Mathematics*, 8(4):419–429.

Beck, A. and Warren, P. (1973). The return of the linear search problem. *Israel J. Math*, 14:169–183.

Benkoski, S. J., Monticino, M. G., and Weisinger, J. R. (1991). A survey of the search theory literature. *Naval Research Logistics (NRL)*, 38(4):469–494.

Berry, D. A. and Mensch, R. F. (1986). Discrete search with directional information. *Operations Research*, 34(3):470–477.

Blachman, N. M. (1959). Prolegomena to optimum discrete search procedures. *Naval Research Logistics Quarterly*, 6(4):273–281.

Black, W. L. (1965). Discrete sequential search. *Information and Control*, 8(2):159–162.

Blackwell, D. and Girshick, M. A. (1954). *Theory of Games and Statistical Decisions*. John Wiley & Sons, New York.

Bram, J. (1963). A 2-player n-region search game. IRM-31, Operations Evaluation Group, Center for Naval Analysis.

Brown, S. S. (1980). Optimal search for a moving target in discrete time and space. *Operations Research*, 28(6):1275–1289.

Charalambos, D. and Aliprantis, B. (2013). *Infinite Dimensional Analysis: A Hitchhiker's Guide*. Springer-Verlag Berlin and Heidelberg GmbH & Company KG.

Chew, Jr, M. C. (1967). A sequential search procedure. *The Annals of Mathematical Statistics*, 38(2):494–502.

Chew, Jr, M. C. (1973). Optimal stopping in a discrete search problem. *Operations Research*, 21(3):741–747.

Clarkson, J., Glazebrook, K. D., and Lin, K. Y. (2020). Fast or slow: Search in discrete locations with two search modes. *Operations Research*, 68(2):552–571.

Croucher, J. S. (1975). Application of the fundamental theorem of games to an example concerning antiballistic missile defense. *Naval Research Logistics Quarterly*, 22(1):197–203.

Dagan, A. and Gal, S. (2008). Network search games, with arbitrary searcher starting point. *Networks*, 52(3):156–161.

de Guenin, J. (1961). Optimum distribution of effort: An extension of the koopman basic theory. *Operations Research*, 9(1):1–7.

Degroot, M. H. (1970). *Optimal Statistical Decisions*. McGraw-Hill New York.

Ding, H. and Castanon, D. (2017). Multi-agent discrete search with limited visibility.

Dobbie, J. M. (1963). Search theory: A sequential approach. *Naval Research Logistics Quarterly*, 10(1):323–334.

Dobbie, J. M. (1968). A survey of search theory. *Operations Research*, 16(3):525–537.

Dobbie, J. M. (1974). A two-cell model of search for a moving target. *Operations Research*, 22(1):79–92.

Dresher, M. (1961). *Games of strategy: theory and applications*. Prentice Hall.

Efron, B. (1964). Optimum evasion versus systematic search. *Journal of the Society for Industrial and Applied Mathematics*, 12(2):450–457.

Enslow Jr., P. H. (1966). A bibliography of search theory and reconnaissance theory literature. *Naval Research Logistics Quarterly*, 13(2):177–202.

Ferguson, T. S. (2020). *A Course in Game Theory*. WORLD SCIENTIFIC.

Flesch, J., Karagözoğlu, E., and Perea, A. (2009). Optimal search for a moving target with the option to wait. *Naval Research Logistics (NRL)*, 56(6):526–539.

Fokkink, R., Lidbetter, T., and Végh, L. A. (2019). On submodular search and machine scheduling. *Mathematics of Operations Research*, 44(4):1431–1449.

Fokkink, R. and Stassen, M. (2011). An asymptotic solution of dresher's guessing game. In *Decision and Game Theory for Security*, pages 104–116, Berlin, Heidelberg. Springer Berlin Heidelberg.

Foley, R. D., Hill, T. P., and Spruill, M. C. (1991). Linear search with bounded resources. *Naval Research Logistics*, 38(4):555–565.

Foreman, J. (1974). The princess and monster on the circle. *Differential games and control theory*, pages 231–240.

Franck, W. (1965). An optimal search problem. *SIAM Review*, 7(4):503–512.

Gal, S. (1979). Search games with mobile and immobile hider. *SIAM Journal on Control and Optimization*, 17(1):99–122.

Gal, S. (1980). *Search Games*. Mathematics in science and engineering : a series of monographs and textbooks. Academic Press.

Gal, S. (2001). On the optimality of a simple strategy for searching graphs. *International Journal of Game Theory*, 29(4):533–542.

Gal, S., Alpern, S., and Casas, J. (2015). Prey should hide more randomly when a predator attacks more persistently. *Journal of The Royal Society Interface*, 12(113):20150861.

Gal, S. and Casas, J. (2014). Succession of hide-seek and pursuit-evasion at heterogeneous locations. *Journal of The Royal Society Interface*, 11(94):20140062.

Garnaev, A. Y. (1992). A remark on the princess and monster search game. *International Journal of Game Theory*, 20(3):269–276.

Gittins, J. C. (1979). Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society. Series B (Methodological)*, 41(2):148–177.

Gittins, J. C. (1989). *Multi-armed bandit allocation indices*. Wiley.

Gittins, J. C. and Glazebrook, K. (1977). On bayesian models in stochastic scheduling. *Journal of Applied Probability*, 14(3):556–565.

Gittins, J. C., Glazebrook, K. D., and Weber, R. (2011). *Multi-Armed Bandit Allocation Indices*. John Wiley and Sons, Ltd.

Gittins, J. C. and Jones, D. M. (1974). A dynamic allocation index for the sequential design of experiments. *In Progress in Statistics*, pages 241–266.

Gittins, J. C. and Jones, D. M. (1979). A dynamic allocation index for the discounted multiarmed bandit problem. *Biometrika*, 66(3):561–565.

Gittins, J. C. and Roberts, D. M. (1979). The search for an intelligent evader concealed in one of an arbitrary number of regions. *Naval Research Logistics Quarterly*, 26(4):651–666.

Glazebrook, K. (1976a). A profitability index for alternative research projects. *Omega*, 4(1):79–83.

Glazebrook, K. (1976b). A profitability index for alternative research projects. *Omega*, 4(1):79–83.

Glazebrook, K. D. (1982). On a sufficient condition for superprocesses due to whittle. *Journal of Applied Probability*, 19(1):99–110.

Gluss, B. (1961). Approximately optimal one-dimensional search policies in which search costs vary through time. *Naval Research Logistics Quarterly*, 8:277–283.

Hall, G. J. (1976). Sequential search with random overlook probabilities. *The Annals of Statistics*, 4(4):807–816.

Hall, G. J. (1977). Strongly optimal policies in sequential search with random overlook probabilities. *The Annals of Statistics*, 5(1):124–135.

Hellerstein, L., Lidbetter, T., and Pirutinsky, D. (2019). Solving zero-sum games using best-response oracles with applications to search games. *Operations Research*, 67(3):731–743.

Hellman, O. (1970). On the effect of a search upon the probability distribution of a target whose motion is a diffusion process. *The Annals of Mathematical Statistics*, 41(5):1717–1724.

Hellman, O. (1971). Optimal search for a randomly moving object in a special case. *Journal of Applied Probability*, 8(3):606–611.

Hohzaki, R. (2016). Search games: Literature and survey. *Journal of the Operations Research Society of Japan*, 59(1):1–34.

Isaacs, R. (1965). *Differential Games*. Wiley, New York.

Jacko, P. (2010). Restless bandits approach to the job scheduling problem and its extensions. *Modern trends in controlled stochastic processes: theory and applications*, pages 248–267.

Johnson, S. M. (1964). A search game. *Advances in Game Theory*, 52:39.

Jotshi, A. and Batta, R. (2008). Search for an immobile entity on a network. *European Journal of Operational Research*, 191(2):347 – 359.

Jotshi, A. and Batta, R. (2009). Investigating the benefits of re-optimisation while searching for two immobile entities on a network. *International Journal of Mathematics in Operational Research*, 1(1-2):37–75.

Kadane, J. (2015). Optimal discrete search with technological choice. *Mathematical Methods of Operations Research*, 81:317–336.

Kadane, J. B. (1968). Discrete search and the Neyman-Pearson Lemma. *Journal of Mathematical Analysis and Applications*, 22(1):156–171.

Kadane, J. B. (1971). Optimal whereabouts search. *Operations Research*, 19(4):894–904.

Kadane, J. B. and Simon, H. A. (1977). Optimal strategies for a class of constrained sequential problems. *The Annals of Statistics*, 5(2):237–255.

Kan, Y. C. (1974). A counterexample for an optimal search-and-stop model. *Operations Research*, 22(4):889–892.

Kan, Y. C. (1977). Optimal search of a moving target. *Operations Research*, 25(5):864–870.

Kelly, F. (1982). On optimal search with unknown detection probabilities. *Journal of Mathematical Analysis and Applications*, 88(2):422 – 432.

Kikuta, K. (1990). A hide and seek game with traveling cost. *Journal of the Operations Research Society of Japan*, 33(2):168–187.

Kikuta, K. (1991). A search game with traveling cost. *Journal of the Operations Research Society of Japan*, 34(4):365–382.

Kikuta, K. (1995). A search game with traveling cost on a tree. *Journal of the Operations Research Society of Japan*, 38(1):70–88.

Kikuta, K. (2004). A search game on a cyclic graph. *Naval Research Logistics*, 51(7):977–993.

Kikuta, K. and Ruckle, W. H. (1994). Initial point search on wieghted trees. *Naval Research Logistics*, 41(6):821–831.

Kimeldorf, G. and Smith, F. H. (1979). Binomial searching for a random number of multinomially hidden objects. *Management Science*, 25(11):1115–1126.

Koopman, B. (1946). *Search and Screening, OEG Report No. 56*. Operations Evaluation Group, Washington D.C.

Koopman, B. O. (1957). The theory of search. iii. the optimum distribution of searching effort. *Operations Research*, 5(5):613–626.

Kress, M., Lin, K. Y., and Szechtman, R. (2008). Optimal discrete search with imperfect specificity. *Mathematical Methods Of Operations Research*, 68(3):539–549.

Lidbetter, T. (2013). Search games with multiple hidden objects. *SIAM Journal on Control and Optimization*, 51(4):3056–3074.

Lidbetter, T. and Lin, K. (2019). Searching for multiple objects in multiple locations. *European Journal of Operational Research*.

Lin, K., Atkinson, M., Chung, T., and Glazebrook, K. (2013). A graph patrol problem with random attack times. *Operations Research*, 61(3):694–710.

Lin, K., Atkinson, M., and Glazebrook, K. (2014). Optimal patrol to uncover threats in time when detection is imperfect. *Naval Research Logistics*, 61(8):557–576.

Lin, K. Y. and Singham, D. I. (2015). Robust search policies against an intelligent evader. Technical report, Monterey, California. Naval Postgraduate School.

Lin, K. Y. and Singham, D. I. (2016). Finding a hider by an unknown deadline. *Oper. Res. Lett.*, 44(1):25–32.

Lössner, U. and Wegener, I. (1982). Discrete sequential search with positive switch cost. *Mathematics of Operations Research*, 7(3):426–440.

Matula, D. (1964). A periodic optimal search. *The American Mathematical Monthly*, 71(1):15–21.

Mela, D. F. (1961). Information theory and search theory as special cases of decision theory. *Operations Research*, 9(6):907–909.

Nakai, T. (1986a). A search game with one object and two searchers. *Journal of Applied Probability*, 23(3):696–707.

Nakai, T. (1986b). A sequential evasion-search game with a goal. *Journal of the Operations Research Society of Japan*, 29(2):113–122.

Nakai, T. (1987). Continuous-time search model with improvements of detection rates. *Journal of the Operations Research Society of Japan*, 30(3):316–334.

Nakai, T. (1990). A preemptive detection game. *Journal of Information and Optimization Sciences*, 11(1):1–15.

Nakai, T. (2004). Optimal search with time-dependent detection rate functions. *Journal of Statistics and Management Systems*, 7(1):121–129.

Nash, P. (1973). *Optimal allocation of Resources Between Research Projects.* PhD thesis, University of Cambridge.

Neuts, M. F. (1963). A multistage search game. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):502–507.

Norris, R. C. (1962). Studies in search for a conscious evader. *MIT Lincoln Laboratory Technical Report Number 279.*

Onaga, K. (1971). Optimal search for detecting a hidden object. *SIAM Journal on Applied Mathematics*, 20(2):298–318.

Osborne, M. J. and Rubinstein, A. (1994). *A Course in Game Theory*, volume 1 of *MIT Press Books*. The MIT Press.

Paperless Post (2020). Golden birthday: what is it and how to celebrate yours in style. `https://www.paperlesspost.com/blog/golden-birthday/`.

Pavlović, L. (1993). Search game on an odd number of arcs with immobile hider. *The Yugoslav Journal of Operations Research*, 3(5):11–19.

Persinger, C. A. (1973). Optimal search using two nonconcurrent sensors. *Naval Research Logistics Quarterly*, 20(2):277–288.

Pollock, S. M. (1970). A simple model of search for a moving target. *Operations Research*, 18(5):883–903.

Posner, E. (1963). Optimal search procedures. *IEEE Transactions on Information Theory*, 9(3):157–160.

Pursiheimo, U. (1977). On the optimal search for a target whose motion is conditionally deterministic with stochastic initial conditions on location and parameters. *SIAM Journal on Applied Mathematics*, 32(1):105–114.

Puterman, M. L. (2014). *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Reijnierse, J. H. (1995). *Games, graphs and algorithms*. PhD thesis, University of Nijmegen.

Reijnierse, J. H. and Potters, J. A. M. (1993). Search games with immobile hider. *International Journal of Game Theory*, 21:385–394.

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

Roberts, D. M. and Gittins, J. C. (1978). The search for an intelligent evader: Strategies for searcher and evader in the two-region problem. *Naval Research Logistics Quarterly*, 25(1):95–106.

Ross, S. M. (1969). A Problem in Optimal Search and Stop. *Operations Research*, 17(6):984–992.

Ross, S. M. (1983). *Introduction to Stochastic Dynamic Programming: Probability and Mathematical*. Academic Press, Inc., USA.

Ruckle, W. H. (1991). A discrete search game. In *Stochastic Games And Related Topics*, pages 29–43. Springer.

Sakaguchi, M. (1973). Two-sided search games. *Journal of the Operations Research Society of Japan*, 16(4):207–225.

Saretsalo, L. (1973). On the optimal search for a target whose motion is a markov process. *Journal of Applied Probability*, 10(4):847–856.

Schweitzer, P. J. (1971). Threshold probabilities when searching for a moving target. *Operations Research*, 19(3):707–709.

Sharlin, A. (1987). Optimal search for one of many objects hidden in two boxes. *European Journal of Operational Research*, 32(2):251 – 259.

Shechter, S. M., Ghassemi, F., Gocgun, Y., and Puterman, M. L. (2015). Technical note: Trading off quick versus slow actions in optimal search. *Operations Research*, 63(2):353–362.

Slivkins, A. (2019). Introduction to multi-armed bandits. *Foundations and Trends in Machine Learning*, 12(1-2):1–286.

Smith, F. H. and Kimeldorf, G. (1975). Discrete sequential search for one of many objects. *The Annals of Statistics*, 3(4):906–915.

Song, N.-O. and Teneketzis, D. (2004). Discrete search with multiple sensors. *Mathematical Methods of Operations Research*, 60(1):1–13.

Soyster, A. L. (1975). A semi-infinite game. *Management Science*, 21(7):806–812.

Stewart, T. J. (1981). A two-cell model of search for an evading target. *European Journal of Operational Research*, 8(4):369 – 378.

Stone, L. D. (1975). *Theory of Optimal Search*. Academic Press New York.

Stone, L. D. (1977). Search for targets with generalized conditionally deterministic motion. *SIAM Journal on Applied Mathematics*, 33(3):456–468.

Stone, L. D. (1979). Necessary and sufficient conditions for optimal search plans for moving targets. *Mathematics of Operations Research*, 4(4):431–440.

Stone, L. D. (2004). *Theory of Optimal Search*. INFORMS, 2 edition.

Stone, L. D. and Kadane, J. B. (1981). Optimal whereabouts search for a moving target. *Operations Research*, 29(6):1154–1166.

Stone, L. D. and Richardson, H. R. (1974). Search for targets with conditionally deterministic motion. *SIAM Journal on Applied Mathematics*, 27(2):239–255.

Stone, L. D., Royset, J., and Washburn, A. (2016). *Optimal Search for Moving Targets*. Springer International Publishing.

Straffin, P. D. (1993). *Game Theory and Strategy*, volume 36. Mathematical Association of America, 1 edition.

Subelman, E. J. (1981). A hide–search game. *Journal of Applied Probability*, 18:628–640.

Sweat, C. W. (1970). Sequential search with discounted income, the discount a function of the cell searched. *The Annals of Mathematical Statistics*, 41(5):1446–1455.

Terkelsen, F. (1972). Some minimax theorems. *MATHEMATICA SCANDINAVICA*, 31:405–413.

Thomas, L. C. and Washburn, A. R. (1991). Dynamic search games. *Operations Research*, 39(3):415–422.

Thompson, W. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3–4):285—294.

Tognetti, K. P. (1968). An optimal strategy for a whereabouts search. *Operations Research*, 16(1):209–211.

Tsitsiklis, J. N. (1994). A short proof of the gittins index theorem. *The Annals of Applied Probability*, 4(1):194–199.

von Neumann, J. (1928). Zur theorie der gesellschaftsspiele. *Mathematische Annalen*, 100:295–320.

von Stengel, B. and Werchner, R. (1997). Complexity of searching an immobile hider in a graph. *Discrete Applied Mathematics*, 78(1):235 – 249.

Washburn, A. R. (1980). Search-evasion game in a fixed region. *Operations Research*, 28(6):1290–1298.

Washburn, A. R. (1983). Search for a moving target: The fab algorithm. *Operations Research*, 31(4):739–751.

Washburn, A. R. (2002). *Search and Detection*. INFORMS, 4th edition.

Washburn, A. R. (2003). *Two-Person Zero-Sum Games*. INFORMS, Rockville, MD, 3rd edition.

Weber, R. (1992). On the gittins index for multiarmed bandits. *The Annals of Applied Probability*, 2(4):1024–1033.

Weber, R. R. (1986). Optimal search for a randomly moving object. *Journal of Applied Probability*, 23(3):708–717.

Wegener, I. (1980). The discrete sequential search problem with nonrandom cost and overlook probabilities. *Mathematics of Operations Research*, 5(3):373–380.

Wegener, I. (1981). The construction of an optimal distribution of search effort. *Naval Research Logistics Quarterly*, 28(4):533–543.

Wegener, I. (1982). The discrete search problem and the construction of optimal allocations. *Naval Research Logistics Quarterly*, 29(2):203–212.

Wegener, I. (1985). Optimal search with positive switch cost is np-hard. *Information Processing Letters*, 21(1):49 – 52.

White, D. (1992). A conjecture of ross for a search problem. *Journal of Mathematical Analysis and Applications*, 166(1):263 – 271.

Whittle, P. (1980). Multi-armed bandits and the gittins index. *Journal of the Royal Statistical Society. Series B (Methodological)*, 42(2):143–149.

Zelikin, M. (1972). On a differential game with incomplete information. *Soviet Math. Dokl*, 13:228–231.