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

Jake Clarkson* Kevin D. Glazebrook † Kyle Y. Lin‡

November 14, 2018

Abstract

An object is hidden in one of several discrete locations according to some known probability distribution, and the goal is to discover the object in minimum expected time by successive searches of individual locations. If there is only one way to search each location, this search problem is solved using Gittins indices. Motivated by modern search technology, we extend earlier work to allow 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 object. 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. Our analysis begins by—for each mode—identifying 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 probability distribution of the object's hiding location 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.

Keywords: Bayesian updating, Gittins index, optimal search, speed-accuracy trade-off, stochastic coupling, threshold-type policy.

1 Introduction

An object is hidden in one of N discrete locations and a searcher wishes to find it. For $i = 1, \dots N$, the object is hidden in location i with known *hiding probability* p_i, with $\sum_{i=1}^{N} p_i = 1$. The discrete distribution $(p_1, p_2, \dots, p_N)$ is a prior distribution describing the searcher's knowledge about the object's location in advance of the search. After any (unsuccessful) search of a location is completed, Bayes' rule is applied to update the probability distribution about the object's location to a posterior distribution.

There are two search modes available to look for the object in each location: a *fast mode* and a *slow mode*. For example, a search squad can use a dog (fast mode) or a metal detector (slow

*STOR-i Centre for Doctoral Training, Science and Technology Building, Lancaster University, LA1 4YR, j.clarkson@lancaster.ac.uk

†Department of Management Science, Lancaster University, LA1 4YX, k.glazebrook@lancaster.ac.uk

‡Operations Research Department, Naval Postgraduate School, Monterey, CA 93943, kylin@nps.edu

mode) to locate a hidden bomb. When using an unmanned aerial vehicle to look for survivors in mountains or airplane crash sites, the speed of the unmanned aerial vehicle can be adjusted to be either fast or slow. Each search mode is characterized by its *search time* and *detection probability*, both known in advance by the searcher. A slow (resp. fast) search in location i will take search time $t_{i,s}$ (resp. $t_{i,f}$), and, if the object is hidden there, find the object with detection probability $q_{i,s}$ (resp. $q_{i,f}$), independent of everything else, for $i = 1, \ldots, N$. A slow search has a larger detection probability, but takes longer, so $q_{i,s} > q_{i,f}$ and $t_{i,s} > t_{i,f}$. The goal of the searcher is to minimize the expected time to find the object, or *expected search time*.

Search problems with multiple search modes are of increasing importance due to advanced technologies resulting in several ways to search a location. There may be several choices of search agents such as humans, animals, and robots. 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. Shechter et al. (2015) investigate a search problem involving two search modes. A fast search at a location may damage the object, resulting in a failure. A slow search, on the other hand, may expose the searcher to additional risk such as enemy fire, which is considered a failure too. The searcher's goal is to minimize the probability that the search ends in a failure. Contrary to our model, a search will discover the object with certainty if the object is hidden in the searched location; in other words, there is no possibility of overlook. Alpern and Lidbetter (2015) study a search game on a network, where the searcher moves along arcs to look for a hidden object. The searcher can choose between two speeds, and is guaranteed to find the object when passing it at the slower speed. For readers interested in a general survey on search theory, please see Alpern et al. (2013); Alpern and Gal (2003); Stone (2004); Washburn (2002).

The version of our search problem (outlined in the opening two paragraphs) in which there is only one search mode for each location has been studied extensively in the literature. For this simpler problem, we write t_i for the search time of location i, and q_i for its detection probability. Gittins (1989) gives an account of this problem, founded on a comment by Kelly in Gittins (1979), which exploits the fact that it is equivalent to a multi-armed bandit problem for which Gittins indices provide an optimal solution. The single-mode problem may also be formulated as what Cowan and Katehakis (2015) call a multi-armed bandit under general commitment, in which the period between one decision time and the next depends on the arm played. An optimal policy is to always search a location that has a maximal value of $p_i' q_i / t_i$, where p_i' is the object's current (posterior) hiding probability for location i, $i = 1, \ldots, N$. This result was first attributed to Blackwell in his notes on dynamic programming (Matula, 1964; Black, 1965). When $t_i = 1$ for $i = 1, \ldots N$, Chew (1967) and Kadane (1968) showed that the same policy maximizes the probability of discovering the object within m searches, for every $m = 1, 2, \ldots$. For variants of this search problem, please see Chew (1973); Kadane (1971); Kress et al. (2008); Ross (1969); Wegener (1980); Lin and Singham (2015, 2016).

The search problem becomes significantly more challenging if multiple search modes are available for each location. In addition to deciding where to look next, the searcher needs to choose a search mode. Hence the problem is now equivalent to what Gittins et al. (2011) call a *family of alternative superprocesses*. This is a radical extension of the multi-armed bandit problem in which each arm has its own decision structure. There is a general theory for such problems based on a sufficient condition due to Whittle (1980). For our search problem, we have found that a more direct approach which makes selective use of this general theory gives the most natural account.

In Section 2, we show that the simple, single search mode result can be extended to yield the following conclusion. If each location in our two-mode problem has a pre-specified sequence mandating how successive searches should be conducted (i.e., using which search mode), then a policy optimal under these within-location sequences (which now only needs to specify the order in which locations are searched) is a Gittins index policy. This fact reduces our two-mode problem to the determination of within-location sequences respected by an optimal policy. A natural conjecture is that such a within-location sequence for location i might coincide with one which is optimal when $p_i = 1$, namely always search slow when $q_{i,s}/t_{i,s} \geq q_{i,f}/t_{i,f}$ and always search fast otherwise. It turns out that while it is indeed optimal to always search slow in location i if $q_{i,s}/t_{i,s} \geq q_{i,f}/t_{i,f}$, it is *not* always optimal to search fast there if $q_{i,s}/t_{i,s} < q_{i,f}/t_{i,f}$. In Section 3, we give a sufficient condition for each mode to dominate the other mode in the same location, such that the latter should never be used.

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 4 by the study of some two-location problems in which one location has just one search mode, which has perfect detection. Section 5 presents a range of heuristic policies with suboptimality bounds for the general two-mode problem based on the analyses of Sections 3 and 4. Section 6 demonstrates the performance of these heuristics in an extensive numerical study. Finally, Section 7 concludes and suggests a few future research directions.

2 Model and Preliminaries

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

1. A single object is hidden in one of N discrete locations (henceforth boxes for conciseness) 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.

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

Standard theory indicates that there exists an optimal policy which is *stationary, nonrandomized and Markov*, which here means that prior to discovery the next action will be a deterministic function of the history of actions taken to date (Puterman, 2014). That history can be summarized by the number of unsuccessful slow and fast searches in each box to date. Following any such 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 mandate the search of 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. From any optimal search sequence, we can extract an infinite subsequence determining the search modes (slow or fast) of successive visits to box i, for $i = 1, \ldots, N$. We call this subsequence an optimal *within-box subsequence* for box i, and denote it by $A_i^* := \{a_{i,n}^*, n \in \mathbb{Z}^+\}$, where $a_{i,n}^*$ is the mode at which the nth search of box i is made in the optimal search sequence in question, and $\mathbb{Z}^+$ is the set of positive integers.

A discussion of the single-mode version of the above search problem may be found in Section 8.2 of Gittins (1989). That analysis serves to show that when there is only one search mode for each box, a search sequence that minimizes 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 this problem, where the searcher needs to decide only which location to search next because the choice of search mode is predetermined. Assume that, for $i = 1, \ldots N$, the within-box subsequence $A_i = \{a_{i,n}, n \in \mathbb{Z}^+\}$ is pre-specified, where $a_{i,n}$ is the mode at which the nth search of box i is to be made. How do we then optimally interlace these N within-box subsequences to produce a search sequence that minimizes the expected search time?

Write $\sigma(\mathbf{A})$ for a search sequence that arbitrarily interlaces within-box subsequences $\mathbf{A} = \{A_i, i = 1, \ldots, N\}$, and write $\tau(\sigma(\mathbf{A}))$ for the random search time under $\sigma(\mathbf{A})$. In order to minimize $E[\tau(\sigma(\mathbf{A}))]$, we first try to maximize $E[\beta^{\tau(\sigma(\mathbf{A}))}]$ for some $\beta \in (0, 1)$ and then later take the limit $\beta \to 1$. By conditioning on the location of the object, we have

$$E\left[\beta^{\tau(\sigma(\mathbf{A}))}\right] = \sum_{i=1}^{N} p_i \sum_{n=1}^{\infty} \left\{ \prod_{m=1}^{n-1} (1 - q_{i,a_{i,m}}) \right\} q_{i,a_{i,n}} \beta^{t(i,n)} \tag{1}$$

where $t(i, n)$ is the time of completion of the n^{th} search of box i under $\sigma(\mathbf{A})$ given that the object is yet to be found. It follows simply that the task of choosing $\sigma(\mathbf{A})$ to maximize $E[\beta^{\tau(\sigma(\mathbf{A}))}]$ may be formulated as a semi-Markov multi-armed bandit with the following features:

1. At each decision epoch, one of the N arms of the bandit is pulled.

2. Inspection of (1) shows that the n^{th} pull of arm i takes time $t_{i,a_{i,n}}$ and earns a deterministic reward
$$p_i \left\{ \prod_{m=1}^{n-1} (1 - q_{i,a_{i,m}}) \right\} q_{i,a_{i,n}}.$$
This reward is received at the completion of the n^{th} pull of arm i at time $t(i,n)$ and is discounted by factor β.

3. Decision epochs occur at time 0 and at the completion of successive arm pulls.

4. An optimal policy chooses successive arms to pull to maximize the aggregate reward received.

The analysis in Chapter 2 of Gittins (1989) may be deployed as follows to demonstrate that the above multi-armed bandit may be solved by a Gittins index policy. Consider a situation in which box i has been searched some $n_i \in \mathbb{N}$ times already, $i = 1, \ldots, N$, where $\mathbb{N}$ denotes the set of nonnegative integers. The Gittins index associated with box i, $i = 1, \ldots, N$, is given by

$$G_i(n_i, A_i, \beta) = p_i \left\{ \prod_{m=1}^{n_i} (1 - q_{i,a_{i,m}}) \right\} \left[\max_{r \in \mathbb{Z}^+} \frac{\sum_{u=n_i+1}^{n_i+r} \left\{ \prod_{v=n_i+1}^{u-1} (1 - q_{i,a_{i,v}}) \right\} q_{i,a_{i,u}} \beta^{\sum_{v=n_i+1}^{u} t_{i,a_{i,v}}}}{1 - \beta^{\sum_{u=n_i+1}^{n_i+r} t_{i,a_{i,u}}}} \right].$$

We say a search sequence is *consistent* with the within-box subsequences $\mathbf{A}$ if the search sequence can be obtained by interlacing elements of $\mathbf{A}$. The following theorem solves the multi-armed bandit.

Theorem 1 A search sequence consistent with the within-box subsequences $\mathbf{A}$ which maximizes $E\left[\beta^{\tau(\sigma(\mathbf{A}))}\right]$ is characterized as follows. At any point at which box i has been searched $n_i \in \mathbb{N}$ times, $i = 1, \ldots, N$, the next search will be of any box j satisfying $j = \arg\max_{i=1,\ldots,N} G_i(n_i, A_i, \beta)$, and will use search mode a_{j,n_j+1}.

The Gittins index policy described in the above result plainly minimizes $E\left[1 - \beta^{\tau(\sigma(\mathbf{A}))}\right]/(1 - \beta)$ among all search sequences consistent with the within-box subsequences $\mathbf{A}$. The problem of determining a search sequence consistent with $\mathbf{A}$ to minimize $E\left[\tau(\sigma(\mathbf{A}))\right]$ is now solved using the Gittins indices

$$\begin{aligned} G_i(n_i, A_i) &= \lim_{\beta \to 1} (1 - \beta) G_i(n_i, A_i, \beta) \\ &= p_i \left\{ \prod_{m=1}^{n_i} \left(1 - q_{i,a_{i,m}}\right) \right\} \left[\max_{r \in \mathbb{Z}^+} \frac{\sum_{u=n_i+1}^{n_i+r} \left\{ \prod_{v=n_i+1}^{u-1} (1 - q_{i,a_{i,v}}) \right\} q_{i,a_{i,u}}}{\sum_{u=n_i+1}^{n_i+r} t_{i,a_{i,u}}} \right], n_i \in \mathbb{N} \qquad (2) \end{aligned}$$

for $i = 1, \ldots N$. We state this conclusion explicitly as follows.

Corollary 2 A search sequence consistent with the within-box subsequences $\mathbf{A}$ which minimizes $E\left[\tau(\sigma(\mathbf{A}))\right]$ is characterized as follows. At any point at which box i has been searched $n_i \in \mathbb{N}$ times, $i = 1, \ldots, N$, the next search will be of any box j satisfying $j = \arg\max_{i=1,\ldots,N} G_i(n_i, A_i)$, and will use search mode a_{j,n_j+1}.

Remark 3 An equivalent set of indices (in the sense of determining the same optimal search sequences) can be obtained by dividing all Gittins indices $G_i(n_i, A_i)$, $i = 1, \ldots, N$, in (2) by the quantity

$$\sum_{j=1}^{N} p_j \left\{ \prod_{m=1}^{n_j} (1 - q_{j,a_{j,m}}) \right\}$$

to obtain new indices which take the form

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

where p'_i is the object's current (posterior) hiding probability for box i. The indices $G'_i(n_i, A_i)$, $i = 1, \ldots, N$, 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 summarize, once we know how to conduct successive searches of each box optimally—namely, an optimal within-box subsequence for each box—a suitable collection of Gittins indices will then determine how we should choose optimally which box to search. The next section will identify sufficient conditions for a box such that an optimal within-box subsequence consists of only one search mode.

3 Structural Properties of an Optimal Policy

Generally speaking, an optimal choice of search mode for any box depends on the object's current (posterior) hiding probabilities and the search modes of the other boxes (see the online appendix A for a numerical example). It would be useful, however, to identify boxes where one search mode is so much better than the other that the latter should never be used in an optimal policy, regardless of the search modes of the other boxes. Sections 3.1 and 3.2 present sufficient conditions for such dominance to occur. Based on these findings, Section 3.3 introduces a Monte Carlo method to estimate the optimal expected search time, and Section 3.4 presents a lower bound on it. Section 3.5 extends the sufficient conditions to search problems with three or more search modes per box.

3.1 A Sufficient Condition for the Slow Mode to Dominate

We consider the two-mode search problem described in Section 2. Our first result states that, if the fast mode and the slow mode for some box have the same detection rate (i.e., the ratio between detection probability and search time), then an optimal policy never needs to use the fast mode for that box.

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

Without loss of generality, we prove Theorem 4 for $j = 1$. The proof requires the introduction of a variant of the two-mode search problem, and two lemmas. To begin, suppose that $q_{1,s}/t_{1,s} = q_{1,f}/t_{1,f}$. Further 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 4 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 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 4, we will show that $T_S \leq T_A$.

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. In addition, at any point, the searcher may stop searching box 1 in order to search another box, and, when she returns to box 1, her search is resumed from the place where she abandoned it last. After reaching the end point t_s, the searcher then jumps back to 0 and moves toward t_s again. We write T_W for the minimized expected search time when the searcher uses the within-box subsequences A_i for boxes $i = 2, \ldots, N$ and sweeps continuously for box 1.

Please note that if the searcher searches a random subset of the line segment $[0, t_s]$ with length t_f, then the probability of finding the object, if it is hidden in box 1, is $q_s(t_f/t_s) = q_f$. 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. This advantage is quantified in the next lemma.

Lemma 5 $T_A - T_W \geq p_1 t_s/2$.

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.

Let searcher 1 conduct her optimal search, choosing between boxes using Gittins indices, as detailed in Corollary 2. Below we describe a feasible policy for searcher 2 which mimics searcher 1's optimal policy. Whenever searcher 1 searches box $i \neq 1$, let searcher 2 search the same box using the same mode. When searcher 1 searches box 1 using the slow (resp. fast) mode, 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. With this feasible policy for searcher 2, we can see that if the object is not hidden in box 1, the conditional expected search time is identical for both searchers.

Now, consider the case in which the object is hidden in box 1. First, we examine the expected time spent in box 1 for each searcher. For searcher 1, one can show that this amount is $t_s/q_s = t_f/q_f$, regardless of A—the within-box subsequence for box 1. 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 n^{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 n^{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 n^{th} visit, but her detection probability will depend on the point $x \in [0, t_s]$ at which her $(n-1)^{\text{st}}$ 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

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

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

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

Now suppose that searcher 1 uses the slow mode on her n^{th} visit to box 1, so her relevant detection probability is q_s. Correspondingly, searcher 2's n^{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 n^{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 $i = 2, \dots, N$ is no larger for searcher 2 than for searcher 1.

We conclude from the above calculations that

$$T_A - T_W \geq p_1 \left(\frac{t_s}{q_s} - \left(\frac{t_s}{q_s} - \frac{t_s}{2} \right) \right) = \frac{p_1 t_s}{2},$$

which completes the proof. ■

The next lemma shows that the inequality in Lemma 5 becomes equality when, in the standard problem, box 1 is always searched using the slow mode.

Lemma 6 $T_S - T_W = p_1 t_s / 2$.

Proof. We again consider two searchers. Searcher 2 uses continuous sweeping for box 1 (as in the proof of Lemma 5), 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 $i = 2, \dots, N$. 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 Corollary 2. To study an optimal policy for searcher 2, divide the interval $[0, t_s]$ into m equal-length subintervals $1_r := [(r-1)t_s/m, rt_s/m)$, $r = 1, \dots, m-1$, and $1_m := [(m-1)t_s/m, t_s]$. 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 $r = 1, \dots m-1$, 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 $2, 3, \dots, N$ alongside m identical small boxes $1_1, 1_2, \dots, 1_m$, while searcher 3 has regular boxes $1, 2, \dots, N$. For both searchers, p_i is the object's hiding probability for box i, $i = 2, \dots, N$. For searcher 2, p_1/m is the object's hiding probability for each of the small boxes $1_r, r = 1, \dots, m$, 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 $G_{1_r}(k)$, $r = 1, \dots, m$, 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 $G_{1_1}(k+1) < G_{1_r}(k)$, $r = 2, \dots, m$, and so the small boxes 1_r, $r = 2, \dots, m$, 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 5 and 6, we can conclude that $T_S \le T_A$, which completes the proof of Theorem 4. We next conclude this section with our main result, which extends Theorem 4 as follows.

Theorem 7 In the two-mode search problem, if any box j satisfies

$$\frac{q_{j,s}}{t_{j,s}} \ge \frac{q_{j,f}}{t_{j,f}}, \tag{3}$$

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

Proof. Without loss of generality set $j = 1$. If (3) is an equality, the result is an immediate consequence of Theorem 4. Suppose now that (3) is a strict inequality and let

$$\widehat{t}_{1,f} := \frac{q_{1,f}\cdot t_{1,s}}{q_{1,s}} < t_{1,f}, \quad \text{so} \quad \frac{q_{1,s}}{t_{1,s}} = \frac{q_{1,f}}{\widehat{t}_{1,f}}. \tag{4}$$

Suppose now we fix within-box subsequences to be A for box 1, 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 $\widehat{t}_{1,f} < t_{1,f}$, it is clear that $\widehat{T}_A \le T_A$. In addition, by (4), it follows from Theorem 4 that $\widehat{T}_S \le \widehat{T}_A$, 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 $T_S = \widehat{T}_S \le \widehat{T}_A \le T_A$, completing the proof. ■

3.2 A Sufficient Condition for the Fast Mode to Dominate

This section gives a sufficient condition for a box such that an optimal policy never need to use the slow mode for that box. We first need a lemma.

Lemma 8 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 8 relies on a simple argument featuring a pairwise interchange of consecutive fast and slow searches of box j, and is therefore omitted.

Theorem 9 In the two-mode search problem, if any box j satisfies

$$\frac{q_{j,f}(1-q_{j,s})}{t_{j,f}} \geq \frac{q_{j,s}}{t_{j,s}}, \tag{5}$$

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

Proof. Without 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 $A_1 = \{a_{1,n}, n \in \mathbb{Z}^+\}$, 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 (2) 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 (5), 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 8, 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 Corollary 2, 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 $G\{A_1; A_i^*, i \neq 1\}$ 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 $G\{A_1^{(r\leftrightarrow r+n)}; A_i^*, i \neq 1\}$ is strictly worse than $G\{A_1^{(r\leftrightarrow r+n+1)}; A_i^*, i \neq 1\}$, $n \in \mathbb{N}$. 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

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

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 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. ■

3.3 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 7 or in Theorem 9, then the problem reduces to the single-mode search problem solved in the literature (Matula, 1964; Black, 1965); otherwise, an optimal policy remains unknown. One way to estimate the optimal expected search time is to discretize the state space, and use standard algorithms for the solution of Markov decision processes, such as value iteration. While this approach produces satisfactory results for $N = 2$, 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 7 and 9. If a box's search times and detection probabilities satisfy (3), we say it is a type-S box; if they satisfy (5), 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. According to Theorems 7 and 9, it is optimal to use only the slow mode of boxes in $\mathcal{S}$, and only the fast mode of boxes in $\mathcal{F}$.

Recall from Corollary 2 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. We already know optimal subsequences for boxes in $\mathcal{S}$ and $\mathcal{F}$, and calculation of their respective indices is straightforward since, if only one search mode is used, the maximum in (2) is always obtained at $r = 1$. For boxes in $\mathcal{H}$, given any subsequence, the index calculation is just as easy, as seen in the next lemma, whose proof can be found in the online appendix B.

Lemma 10 If $i \in \mathcal{H}$, then for any within-box subsequence $A_i = \{a_{i,m}, m \in \mathbb{Z}^+\}$ and any $n \in \mathbb{N}$ we have

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

In other words, the maximum in (2) is always obtained at $r = 1$.

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 Corollary 2 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.

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.

3.4 Lower Bounds on the Optimal Expected Search Time

Write V^* for the optimal expected search time. If there is no box 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 reduce the slow search time to

$$\widehat{t}_{i,s} := \frac{t_{i,f}q_{i,s}}{q_{i,f}} < t_{i,s}, \tag{6}$$

and write V' for the optimal expected search time for this modified search problem. It is clear that $V' \le V^*$, 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) in Theorem 7 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.

Another way to compute a lower bound on V^* is to reduce the fast search time for each box $i \in \mathcal{H}$ to

$$\widehat{t}_{i,f} := \frac{q_{i,f}t_{i,s}(1-q_{i,s})}{q_{i,s}} < t_{i,f}, \tag{7}$$

so that the fast mode dominates the slow mode for box i, since (7) meets the sufficient condition (5) in Theorem 9. It is also possible to compute a lower bound by reducing the slow search time for some boxes in $\mathcal{H}$ according to (6), and reducing the fast search time for the other boxes in $\mathcal{H}$ according to (7). There are $2^{|\mathcal{H}|}$ lower bounds of this kind, where $|\mathcal{H}|$ is the number of boxes in $\mathcal{H}$, and one can choose the largest of these to obtain the tightest such lower bound.

In some cases, we can apply a similar idea to get another lower bound on V^* by increasing either the fast or the slow detection probability so that one search mode dominates in each box—provided that the increased detection probability does not exceed 1. A lower bound computed by increasing a detection probability, however, is weaker than one computed by reducing a search time, either by Theorem 7, or by direct comparison of the adjusted search modes.

3.5 Dominance among Multiple Search Modes

While the paper focuses on the case where there are two modes per box, in this subsection we extend Theorems 7 and 9 to boxes with three or more search modes to find conditions for one mode to dominate the others.

To proceed, for any search mode with detection probability q' and search time t', we define a function $g(t)$ for $t > 0$ based on Theorems 7 and 9 as follows:

$$g(t) := \begin{cases} (q'/t')\,t, & \text{if } t \le t', \\ q', & \text{if } t' < t \le t'/(1-q'), \\ t/(t+t'/q'), & \text{if } t > t'/(1-q'). \end{cases} \tag{8}$$

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

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

Figure 1 illustrates $g(t)$ and $D(q', t')$ for $(q', t') = (0.4, 1)$.

Now consider a search problem with N boxes where, for $i = 1, \ldots, N$, box i has some $K_i \in \mathbb{Z}^+$ search modes, namely $(q_{i,k}, t_{i,k})$ for $k = 1, \ldots, K_i$. For any box i, if there exists $j \in \{1, \ldots, K_i\}$ such that $(q_{i,k}, t_{i,k}) \in D(q_{i,j}, t_{i,j})$ for $k = 1, \ldots, K_i$, then 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. The following is an extension of Theorems 7 and 9 to this multiple-mode setting.

Theorem 11 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.

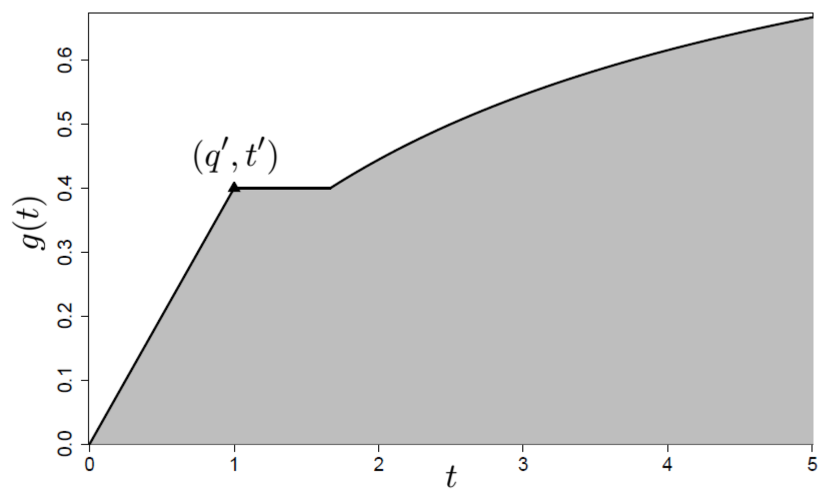


Figure 1: The black line shows the different parts of the function $g(t)$ for $(q', t') = (0.4, 1)$. The set $D(0.4, 1)$ is shown by the line alongside the shaded gray areas.

The proof of Theorem 11 involves similar arguments used to prove Theorems 7 and 9, and is deferred to the online appendix C.

Theorem 11 extends Theorems 7 and 9 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 11, it is reasonable to conjecture a more general result—extending Theorems 7 and 9 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 $(q_{i,k}, t_{i,k}) \in D(q_{i,j}, t_{i,j})$. 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.

4 A Special Case with Two Boxes

This section presents an optimal policy for a particular search problem with 2 boxes. 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.

Neither the condition in Theorem 7 nor the condition in Theorem 9 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 5.

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

$$\underbrace{f, f, \dots f}_{m}, \underbrace{s, s, \dots s}_{n}, 2, f, f \dots, \tag{9}$$

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 m fast searches in box 1, followed by n 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 $m, n \in \mathbb{N}$. We now make further inference on an optimal policy via two propositions, whose proofs are deferred to the online appendices D and E, respectively.

Proposition 12 Define $P_1 := (t_f/q_f)/(t_2 + t_f/q_f)$. An optimal action in state p is to search box 2, if and only if $p \le P_1$.

Now let $V_i(m, n)$ denote the expected search time under (9) if the object is hidden in box i, for $i = 1, 2$ and $m, n \in \mathbb{N}$. Since $q_2 = 1$, we have $V_2(m, n) = mt_f + nt_s + t_2$. If the object is hidden in box 1, then we can compute that

$$\begin{aligned} V_1(m, n) = &\sum_{j=1}^{m} (1 - q_f)^{j-1} q_f \cdot (jt_f) + (1 - q_f)^m \sum_{j=1}^{n} (1 - q_s)^{j-1} q_s \cdot (mt_f + jt_s) \\ &+ (1 - q_f)^m (1 - q_s)^n \left(mt_f + nt_s + t_2 + \frac{t_f}{q_f} \right). \end{aligned}$$

After some algebraic work, we see that

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

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

$$V(m,n,p) = pV_1(m,n) + (1-p)V_2(m,n). \tag{10}$$

The next proposition uses (10) to provide further insight into an optimal policy.

Proposition 13 Define $P_2 := (t_f/q_f)/(t_s/q_s) < 1$. The unique optimal action in state p is to search fast in box 1, if $p > \max(P_1, P_2)$.

In the case $P_1 \geq P_2$, an optimal policy is completely characterized by Propositions 12 and 13. In the case $P_1 < P_2$, however, Propositions 12 and 13 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 $V(m,n,p)$ for all $m, n \in \mathbb{N}$ that are relevant. Define

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

which 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

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

Compute $k' := \min\left\{k : h^{(k)}(P_2) \leq P_1\right\}$. 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 (9) for which $m + n \leq k'$, 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 $V(m,n,p)$ among those with $m + n \leq k'$.

5 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 7 and 9. While Theorem 7 proves that the slow mode is optimally designated for boxes in $\mathcal{S}$ and Theorem 9 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 5.1 and 5.2, and derive corresponding suboptimality bounds in Section 5.3. In Section 5.4, we extend suboptimality bounds for selected heuristics to the multiple-mode search problem.

5.1 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 Corollary 2. 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 10 is equivalent to the search mode with the larger detection probability per unit time 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.

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 7 and 9, 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

$$\theta_i = \log\left(\frac{q_{i,s}/t_{i,s}}{q_{i,f}/t_{i,f}}\right) \times \frac{1}{\log(1-q_{i,s})}, \tag{11}$$

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 2 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 6. 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.

5.2 A Threshold-Type Heuristic Policy

Recall the special search problem with 2 boxes studied in Section 4, 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

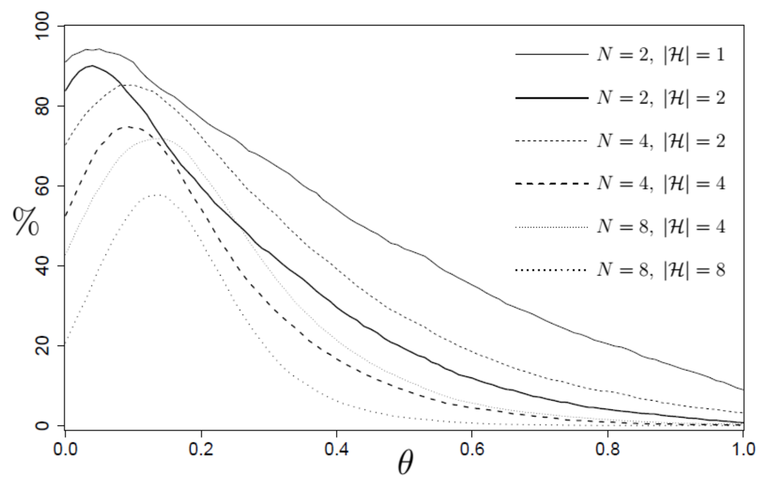


Figure 2: The percentage of search problems generated in Section 6 in which ADR with parameter θ coincides with BADR, for $\theta \in [0, 1]$ and various values of N and $|\mathcal{H}|$.

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 another box. Because, in a type-H box, the fast mode is more effective at finding the object—namely as $q_f/t_f > q_s/t_s$—it is intuitive that these many searches will optimally involve at least one fast search. It then follows from Lemma 8 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 the detection probability per unit time 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

$$\alpha := \frac{q_f/t_f}{q_s/t_s} - 1, \tag{12}$$

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

$$f(x) = \frac{1-p}{p(1-q_f)^{x/t_f} + 1 - p}, \tag{13}$$

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 (13) 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.

Extended to all box types, these notions of immediate and future benefit provide an intuition for the theoretical results of Sections 3.1 and 3.2.

Proposition 14 Both the immediate and future benefit are larger for the fast mode in a type-F box, and larger (or at least the same) for the slow mode in a type-S box.

The proof of this proposition is deferred to the online appendix F.

While in a type-H box, the immediate benefit is clearly always larger for the fast mode, the future benefit may go either way. If both the immediate and future benefit are larger for the fast mode, then it is reasonable to designate fast for that box. Otherwise, the advantage of the slow mode over the fast mode in future benefit can be measured by

$$\beta := \frac{\log(1-q_s)/t_s}{\log(1-q_f)/t_f} - 1, \tag{14}$$

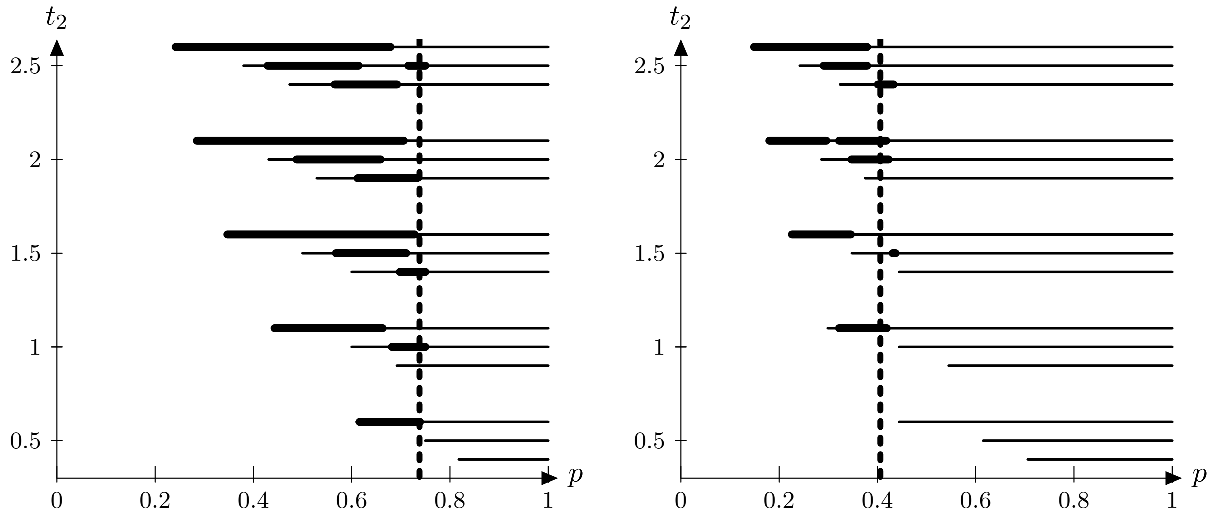
which is positive and does not depend on p. Finally, since the immediate benefit only materializes if the object is in the searched box, while the future benefit only materializes if the object is elsewhere, a natural choice of threshold over which we designate the fast mode is the probability $\widehat{p}$ satisfying $\widehat{p}\alpha = (1-\widehat{p})\beta$, which solves to

$$\widehat{p} = \frac{\beta}{\alpha+\beta}. \tag{15}$$

To demonstrate the threshold, 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), the left-hand side of Figure 3 plots the threshold in (15) against an optimal policy estimated via value iteration, in which the state space $[0, 1]$ is discretized 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. Another example on the right-hand side of Figure 3, and many further choices we made of a type-H box 1, drew a similar conclusion.

Also seen in Figure 3, when box 1 is optimally searched for $p \le \widehat{p}$, an optimal search mode appears to depend heavily on the parameters of box 2. It may be difficult to incorporate such dependence into a heuristic. Therefore, we define a heuristic as follows. For each box in $\mathcal{H}$, if $\beta \le 0$, it can be shown that $\widehat{p} \le 0$, so 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 $p \le \widehat{p}$ and that which designates the slow mode for $p \le \widehat{p}$. 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 the *best threshold* (BT) heuristic.

Figure 3: Optimal actions for $p \in (0, 1)$ for a pair of two-box problems. Box 2 has one search mode with $q_2 = 0.3$ (upper line), 0.6 (middle line), 0.9 (lower line), and t_2 ranging between 0.5 and 2.5. Box 1 is type-H. In the left-hand problem, $q_{1,f} = 0.4$, $q_{1,s} = 0.64$, $t_{1,f} = 1$ and $t_{1,s} = 1.7$, and in the right, $q_{1,f} = 0.3$, $q_{1,s} = 0.5$, $t_{1,f} = 0.4$ and $t_{1,s} = 0.73$. A thick/thin line indicates a slow/fast search in box 1; no line indicates a search in box 2. The dotted line is the threshold $\widehat{p}$ in (15).



5.3 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

$$\delta_{i,s} := \frac{q_{i,f}/t_{i,f}}{q_{i,s}/t_{i,s}} - 1, \qquad \delta_{i,f} := \frac{q_{i,s}/t_{i,s}}{(1 - q_{i,s})q_{i,f}/t_{i,f}} - 1. \tag{16}$$

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 (12).

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 (16). The proofs of the proposition and the corollary below are deferred to the online appendix G.

Proposition 15 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

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

Corollary 16 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.

$$\frac{V_{\mathrm{DR}} - V^*}{V^*} \le \max_{i \in \mathcal{H}} \delta_{i,f}. \tag{17}$$

$$\frac{V_{\mathrm{BADR}} - V^*}{V^*} \le \min\{\max_{i \in \mathcal{H}} \delta_{i,s}, \max_{i \in \mathcal{H}} \delta_{i,f}\}. \tag{18}$$

$$\frac{V_{\mathrm{BSM}} - V^*}{V^*} \le \max_{i \in \mathcal{H}} \ \min\left\{\delta_{i,s}, \delta_{i,f}\right\}. \tag{19}$$

$$\frac{V_{\mathrm{BT}} - V^*}{V^*} \le \max_{i \in \mathcal{H}} \delta_{i,f}. \tag{20}$$

Among those in Corollary 16, 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 16 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 6, the heuristics consistently and substantially outperform these bounds.

5.4 Heuristic Policies and Suboptimality Bounds for Multiple Search Modes

Consider the multiple-mode search problem introduced in Section 3.5, where box i has some $K_i \in \mathbb{Z}^+$ search modes, namely $(q_{i,k}, t_{i,k})$ for $k = 1, \ldots, K_i$, and $i = 1, \ldots, N$. Without loss of generality, label the search modes of box i such that $q_{i,1} < \cdots < q_{i,K_i}$, for $i = 1, \ldots, N$. The single-mode heuristic policies DR and BSM introduced in Section 5.1 can be extended to this setting as follows. 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 5.3 to bound the suboptimality of DR and BSM in the multiple-mode setting. For any box i and any mode k, define

$$\delta_{i,k} := \max\left(\max_{j=1,\ldots k}\left\{\frac{q_{i,j}/t_{i,j}}{q_{i,k}/t_{i,k}}\right\}, \max_{j=k+1,\ldots K_i}\left\{\frac{q_{i,j}/t_{i,j}}{(1-q_{i,j})q_{i,k}/t_{i,k}}\right\}\right) - 1. \tag{21}$$

If mode k for box i satisfies the condition of Theorem 11, so is dominating for box i, then the left-hand inner maximization 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 (21) 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 11 and hence dominating for box i.

We next present a version of Proposition 15 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 (21). The proofs of the proposition and the corollary below are deferred to the online appendix H.

Proposition 17 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

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

Corollary 18 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.

$$\frac{V_{\text{DR}} - V^*}{V^*} \leq \max_{i=1,\dots,N} \delta_{i,m_i}. \tag{22}$$

$$\frac{V_{\text{BSM}} - V^*}{V^*} \leq \max_{i=1,\dots,N} \min_{k=1\dots,K_i} \delta_{i,k}. \tag{23}$$

The bound for BSM in Corollary 18 is at least as strong as that for DR. For each $i = 1, \dots, N$, if we consider some limit under which we have $\delta_{i,k_i} \downarrow 0$ for some $k_i \in \{1, \dots, K_i\}$, then mode k_i becomes dominating for box i and BSM approaches optimality. If $\delta_{i,m_i} \downarrow 0$ for $i = 1, \dots, N$, then both DR and BSM approach optimality. As in Corollary 16, all bounds in Corollary 18 depend only on detection probabilities and search times; they do not depend on the object's hiding probabilities.

6 Numerical Results

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

$$q_s \sim U(0.2, 0.9) \quad t_f \sim U(0.1, 4.5) \quad a \sim U(0.1, 1) \quad b \sim U(0.1, 1), \tag{24}$$

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 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 (24) 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 (12), 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 9 in (5) shows that any draws of q_s and t_f do not preclude a drawn box from being any of the three box types.

6.1 Estimating the Optimal Expected Search Time via Monte Carlo Simulation

To assess the effectiveness of the Monte Carlo (MC) method proposed in Section 3.3, 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, 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 Markov decision process and use value iteration to compute the optimal expected search time in each state. The value iteration algorithm stops when the values in successive iterations are within 10^{-6} for all states.

According to numerical results from this value iteration, it is extremely rare for an optimal policy to involve the slow mode for a type-H box in which the future benefit is larger for the fast mode. 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 (24) 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 1 as average percentages over the optimal values obtained from value iteration (the right-hand side of Table 1 will be explained in Section 6.2). As seen in Table 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.

Table 1: Performance of the MC and ensemble 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			
	$\|\mathcal{H}\| = 1$		$\|\mathcal{H}\| = 2$		$\|\mathcal{H}\| = 1$		$\|\mathcal{H}\| = 2$	
	$p = 0.5$	$p = 0.9$	$p = 0.5$	$p = 0.9$	$p = 0.5$	$p = 0.9$	$p = 0.5$	$p = 0.9$
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

6.2 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 5 are 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 compute the expected search time for a heuristic, we use the formula

$$E[T] = E[T|T \leq b]P(T \leq b) + E[T|T > b]P(T > b), \tag{25}$$

where T is the total search time under that heuristic. For any $b > 0$, we can use (25) to compute an upper bound and a lower bound on $E[T]$, where the difference between the two bounds decreases as b increases. We choose b large enough so that our estimate is within 0.001% of the true value.

To assess our four heuristics, we first fix $|\mathcal{H}| = 1$ and use the same 2000 pairs of boxes generated in Section 6.1. 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 (25) 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 2 displays the results.

Table 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

As seen in Table 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 2, in the problems where DR is suboptimal, the other three heuristics show a remarkable improvement on DR, which can perform poorly.

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 is very small, as seen in Table 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.

Table 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	$(0.7, 0.1, 0.1, 0.1)$	$(0.51, 0.07, \ldots, 0.07)$

Also seen in Table 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 6.1 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 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 $N > 2$.

6.3 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 6.2. 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 (24) 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, 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 (25) 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 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

Table 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.

		$\lvert\mathcal{H}\rvert = 2$				$\lvert\mathcal{H}\rvert = 4$			
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

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 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 5.2, the importance of the immediate benefit, which is larger for the fast mode for all type-H boxes, decreases with the size of the hiding probability.

We next increase the number of boxes to $N = 8$. As seen in Table 5, the BT heuristic again performs the best across all scenarios. Patterns on relative performance between the heuristics in Table 4 are also observed in Table 5. In particular, as $|\mathcal{H}|$ increases, BADR becomes computationally

Table 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.

		$\lvert\mathcal{H}\rvert = 4$				$\lvert\mathcal{H}\rvert = 8$			
Scenario	Metric	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

more efficient relative to BSM ($|\mathcal{H}|+1$ versus $2^{|\mathcal{H}|}$ policies evaluated), while its relative performance degrades only slightly and is still close to optimum. For $N = |\mathcal{H}| = 8$, BADR requires 3.5% (9/256) of the computational effort of BSM, while, in their worst performing scenario, BADR is on average about 0.1% above optimality compared to 0.02% for BSM.

By comparing the results in Tables 4 and 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 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 2 in Section 5.1, 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 will decrease, meaning it is more likely that BSM or BT will

Table 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%

produce a search sequence that is near optimal.

6.4 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 (11). Recall δ_s and δ_f from (16), 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 (14), which measures the advantage of the slow mode over the fast mode in future benefit. The following proposition, whose proof is deferred to the online appendix I, connects θ_H, δ_s, δ_f and β.

Proposition 19 If $\delta_s \geq \delta_f$, or equivalently, $\theta_H \geq 0.5$, then $\beta < 0$.

While Proposition 14 shows that $\beta < 0$ for any type-F box, Proposition 19 identifies some type-H boxes for which $\beta < 0$.

Proposition 19 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 5.1 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 6.1, 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 (24) 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 7 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.

Table 7: 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

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 7. The other two heuristics BT and BSM, however, share a different behavior. Also seen in Table 7, 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 behavior 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 (15). 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.

7 Conclusion

Motivated by advanced search technology, in this paper we extend a search model in the literature to allow a choice between two search modes in each possible location. This extension complicates the problem substantially if one search mode takes less time but the other finds the hidden object with a higher probability. We develop theorems to derive the optimal policy for many cases, and otherwise use them to simplify the search problem in general and design heuristic policies, which consistently deliver near-optimal performance in an extensive numerical study.

A natural extension to our search problem is to allow three or more search modes per location. While Theorem 11 in Section 3.5 extends Theorems 7 and 9 to identify when one mode dominates all the others, further work is required to determine whether Theorems 7 and 9 can be adapted to rule out *inferior* search modes in cases where no single mode dominates. While several of our single-mode heuristics can be extended to this multiple-mode setting, developing more sophisticated heuristics requires a further study beyond the scope of this paper.

Future research directions may include the incorporation of a network structure to our two-mode search problem. After the completion of a search, the searcher can only next choose a location adjacent to their current location. In addition, we may consider our two-mode model as a two-person, zero-sum game between a searcher and a hider, the latter who chooses where to hide the object before the search begins.

Acknowledgements

We are grateful for the support of the EPSRC funded EP/L015692/1 STOR-i Centre for Doctoral Training. We also thank the review team for many helpful comments.

References

Alpern, S., Fokkink, R., Gasieniec, L., Lindelauf, R., and Subrahmanian, V. S., editors (2013). *Search Theory*. Springer.

Alpern, S. and Gal, S. (2003). *The Theory of Search Games and Rendezvous*. Kluwer Academic Publishers.

Alpern, S. and Lidbetter, T. (2015). Optimal trade-off between speed and acuity when searching for a small object. *Operations Research*, 63(1):122–133.

Black, W. L. (1965). Discrete sequential search. *Information and control*, 8(2):159–162.

Chew, 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.

Cowan, W. and Katehakis, M. N. (2015). Multi-armed bandits under general depreciation and commitment. *Probability in the Engineering and Informational Sciences*, 29(1):51–76.

Gittins, J. C. (1979). Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society. Series B (Methodological)*, 41:148–177.

Gittins, J. C. (1989). *Multi-armed bandit allocation indices*. Wiley.

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.

Kress, M., Lin, K. Y., and Szechtman, R. (2008). Optimal discrete search with imperfect specificity. *Mathematical Methods Of Operations Research*, 68(3):539–549.

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. *Operations Research Letters*, 44(1):25–32.

Matula, D. (1964). A periodic optimal search. *The American Mathematical Monthly*, 71(1):15–21.

Puterman, M. L. (2014). Markov decision processes: discrete stochastic dynamic programming.

Ross, S. M. (1969). A Problem in Optimal Search and Stop. *Operations Research*, 17(6):984–992.

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.

Stone, L. (2004). *Theory of Optimal Search*. INFORMS.

Washburn, A. R. (2002). *Search and Detection*. INFORMS, 4th edition.

Wegener, I. (1980). The discrete sequential search problem with nonrandom cost and overlook probabilities. *Mathematics Of Operations Research*, 5(3):373–380.

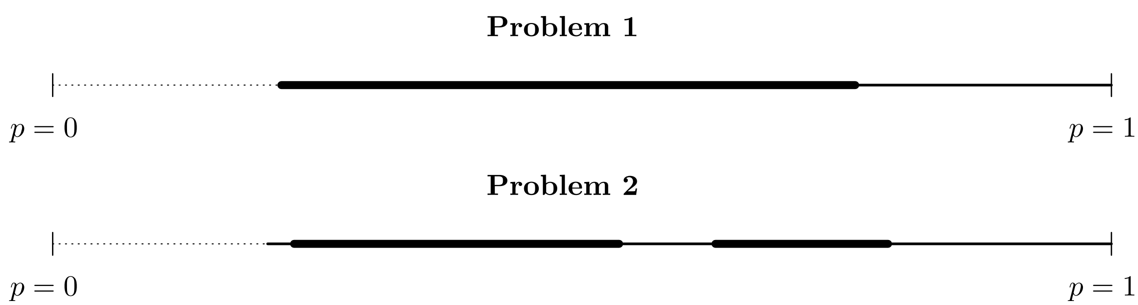
Whittle, P. (1980). Multi-armed bandits and the gittins index. *Journal of the Royal Statistical Society. Series B (Methodological)*, 42:143–149.

Appendices

A Example Showing Dependence of the Optimal Search Mode

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 4 shows an optimal search mode for each value of $p \in (0,1)$, where p is the hiding probability for box 1. 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 is also dependent on the available search mode of box 2.

Figure 4: 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/thin line indicates a slow/fast search in box 1; a dotted line indicates a search in box 2.



B Proof of Lemma 10

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 $A = \{a_n : n \in \mathbb{Z}^+\}$, (2) gives the equation for the Gittins index of the box after some $n \in \mathbb{N}$ searches of the box have been made. Consider the maximization part of (2), namely

$$\max_{r \in \mathbb{Z}^+} \frac{\sum_{u=n+1}^{n+r} \left\{ \prod_{v=n+1}^{u-1} (1 - q_{a_v}) \right\} q_{a_u}}{\sum_{u=n+1}^{n+r} t_{a_u}}. \tag{26}$$

To prove that (26) is maximized by taking $r = 1$, note that taking $r \geq 2$ instead adds

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

to the numerator in (26), and $\sum_{u=n+2}^{n+r} t_{a_u}$ to its denominator. To complete the proof, for each $u = n+2, n+3, \ldots$, it is sufficient to prove the inequality

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

where the right-hand term is the value of (26) with $r = 1$. Consider two cases.

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

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

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

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

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

   If $a_u = f$, then we have

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

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

The proof is completed. ■

C Proof of Theorem 11

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 (8). For $k = 2, 3, \ldots, K$, define $\widehat{q}_k := g(t_k) \ge q_k$, 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. Categorize 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 $t_k \le t_1$, in group B if $t_1 < t_k \le t_1/(1-q_1)$, 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 9, 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 9 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 7 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 object with probability q_1 when she meets it, and has the ability to restart where the previous sweep stopped. The proof of Lemma 5 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 6.

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.
■

D Proof of Proposition 12

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 7 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 10, 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 Corollary 2, 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. ■

E Proof of Proposition 13

Consider two cases. First, suppose $P_1 \geq P_2$, which is equivalent to $t_2 \leq \Delta$. So, for any fixed $m \in \mathbb{N}$ and $p \in (0,1)$, $V(m,n,p)$ in (10) is minimized by taking $n = 0$. Together with Proposition 12, 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

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

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

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

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

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

then for all $n \in \mathbb{N}$, the search sequence with n and $m = 1$ is better than that with n and $m = 0$. 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 12, it follows that the only optimal action is to search box 1 fast for $p > P_2$. ■

F Proof of Proposition 14

The result concerning the immediate benefit is trivial. We prove the result for the future benefit, beginning with the statement for a type-S box. 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

$$\frac{A+B}{C+D} = \frac{-\log(1-q_s)}{t_s} \quad \text{and} \quad \frac{B}{D} = \frac{-\log(1-q_f)}{t_f}. \tag{27}$$

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 $q_s/t_s \geq q_f/t_f$, which implies 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 (27), completes the proof for a type-S box.

We now prove the statement for a type-F box. Again, 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

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

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 (27) completes the proof for a type-F box. ∎

G Proofs of Proposition 15 and Corollary 16

Proof of Proposition 15:

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

$$\widehat{t}_{i,s} := \frac{t_{i,f} q_{i,s}}{q_{i,f}} < t_{i,s}. \tag{28}$$

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

$$\widehat{t}_{i,f} := \frac{q_{i,f} t_{i,s}(1-q_{i,s})}{q_{i,s}} < t_{i,f}. \tag{29}$$

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

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

Suppose we adapt (1) in Section 2 to consider $E[\tau(\sigma(\mathbf{A}))]$. Then we see 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, $i = 1, \dots, N$, is non-zero. Finally, for any box not in $\mathcal{H}$, the search times themselves are also the same. Consequently, we can conclude that

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

where the last line follows from (28) and (29). ■

Proof of Corollary 16:

All bounds are derived from Proposition 15. The bound in (17) follows because DR designates fast for boxes in $\mathcal{H}$. The bound in (18) 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 (19) follows because BSM compares all single-mode policies. Finally, the bound for BT in (20) follows because DR is one of the candidates for BT. ■

H Proofs of Proposition 17 and Corollary 18

Proof of Proposition 17:

For $i = 1, \dots N$, let

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

In the modified search problem where t_{i,k_i} is replaced by $\widehat{t}_{i,k_i}$ for $i = 1, \dots N$, Π is the optimal policy. The same argument used in the proof of Proposition 15 can be applied to the modified and original search problems to show

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

which completes the proof. ■

Proof of Corollary 18:

Because DR uses mode m_i for box i, $i = 1, \dots N$, the bound in (22) follows immediately from Proposition 17. The bound in (23) follows from Proposition 17 because BSM compares all single-mode policies. ■

I Proof of Proposition 19

Before proving the result of Proposition 19, we need a few lemmas.

Lemma 20 For any $a > 1$, we have

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

Proof. Consider the function $h(a) = a - 1 - \log(a)\sqrt{a}$. 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

$$h'(a) = 1 - \left(\frac{\log(a) + 2}{2\sqrt{a}}\right), \tag{30}$$

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

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

so the denominator of the fraction in (30) 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 21 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 20 to obtain the inequality for $\xi = 1/2$. The result follows because $(1 - x)^{-1/2} \leq (1 - x)^{-\xi}$ for all $\xi \in [1/2, 1]$. ■

We are now ready to prove Proposition 19. Using their definitions in (11) and (16), it is straightforward to show that $\delta_s \geq \delta_f$ is equivalent to $\theta_H \geq 0.5$. We show that $\delta_s \geq \delta_f$ 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 $\delta_s \geq \delta_f$ if and only if $\delta_s + 1 \geq (1 - q_s)^{-1/2}$. We shall prove the contrapositive of the result, namely that if $\beta \geq 0$, then $\delta_f > \delta_s$, or equivalently $\delta_s + 1 < (1 - q_s)^{-1/2}$.

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

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

Therefore,

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

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 21 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 21 with $x = q_f$ and $\xi = 1$. Consequently, $f'(q_f) < 0$ for $q_f \in (0, q_s)$, and the proof is completed. ■