2015-11

Robust search policies against an intelligent evader

Lin, Kyle Y.; Singham, Dashi I.

Monterey, California. Naval Postgraduate School

NPS-OR-15-009

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

ROBUST SEARCH POLICIES AGAINST AN INTELLIGENT EVADER

by

Kyle Y. Lin

Dashi I. Singham

November 2015

Approved for public release; distribution is unlimited

The previous version has a signature misplaced on Figure 1, which is corrected in this version.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From-To)*
11-30-2015	Technical Report	11-30-2014 to 11-30-2015

4. TITLE AND SUBTITLE

ROBUST SEARCH POLICIES AGAINST AN INTELLIGENT EVADER

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Kyle Y. Lin, Dashi I. Singham

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) AND ADDRESS(ES)
Operations Research Department
Naval Postgraduate School
Monterey, CA 93943

8. PERFORMING ORGANIZATION REPORT NUMBER

NPS-OR-15-009

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
The views expressed in this report are those of the author(s) and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

14. ABSTRACT

In a classical search model, an object is hidden in one of many cells. Knowing the probability that the object is in each cell, a searcher wishes to find it. Each search in a cell incurs a cost and will discover the object with some probability, with both the cost and discovery probability dependent on the cell. This paper revisits this search problem with an intelligent evader who decides where to hide in order to evade the search. We make two contributions to the literature. First, we show how to compute a randomized policy for the searcher to minimize the expected cost until discovering the evader. Second, if the search has to stop at some point, with the deadline unannounced in advance, we show how the searcher can sequentially allocate each search to simultaneously maximize the probability of discovering the evader by an arbitrary deadline. In the case where the search cost is identical for all cells, our analysis shows that the latter policy is more robust.

15. SUBJECT TERMS

search theory; two-person zero-sum game; robust strategy

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Kyle Y. Lin
Unclassified	Unclassified	Unclassified	UU	34	**19b. TELEPHONE NUMBER** *(include area code)* 831-656-2648

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

THIS PAGE INTENTIONALLY LEFT BLANK

NAVAL POSTGRADUATE SCHOOL
Monterey, California 93943-5000

Ronald A. Route
President

James Newman
Acting Provost

The report entitled "Robust Search Policies Against an Intelligent Evader" was prepared for public release.

Further distribution of all or part of this report is authorized.

This report was prepared by:

Kyle Y. Lin
Associate Professor of
Operations Research

Dashi I. Singham
Research Assistant Professor of
Operations Research

Reviewed by:

Johannes O. Royset
Associate Chair for Research
Department of Operations Research

Released by:

Patricia A. Jacobs
Chair
Department of Operations Research

Jeffrey D. Paduan
Dean of Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

In a classical search model, an object is hidden in one of many cells. Knowing the probability that the object is in each cell, a searcher wishes to find it. Each search in a cell incurs a cost and will discover the object with some probability, with both the cost and discovery probability dependent on the cell. This paper revisits this search problem with an intelligent evader who decides where to hide in order to evade the search. We make two contributions to the literature. First, we show how to compute a randomized policy for the searcher to minimize the expected cost until discovering the evader. Second, if the search has to stop at some point, with the deadline unannounced in advance, we show how the searcher can sequentially allocate each search to simultaneously maximize the probability of discovering the evader by an arbitrary deadline. In the case where the search cost is identical for all cells, our analysis shows that the latter policy is more robust.

THIS PAGE INTENTIONALLY LEFT BLANK

1 INTRODUCTION

In a classical optimal search model, an object is hidden in one of n cells, with the prior probability of being in cell i as p_i, $i = 1, \ldots, n$, with $\sum_{i=1}^{n} p_i = 1$. A search in cell i costs c_i, and will discover the object with probability α_i, if the object is indeed hidden in cell i, $i = 1, \ldots, n$. That is, $1 - \alpha_i$ is the overlook probability in cell i. To minimize the expected cost until the object is found, the optimal policy is to search, at any time, the cell that has the largest present value of $\alpha_i p_i / c_i$, with the hiding probabilities $(p_1, \ldots, p_n)$ being updated in the Bayesian fashion after each unsuccessful search. This result was first attributed to Blackwell in his notes on dynamic programming (see [1], [2]). When $c_i = 1$ for all i, Chew [3] and Kadane [4] showed that the same policy maximizes the probability of discovering the object within t searches, for every $t = 1, 2, \ldots$. For variants of this search problem, please see [5]–[9]. For a general survey on search theory, please see [10]–[13].

In this paper, we study this classical search model with an intelligent evader, who decides where to hide in order to evade the search. We seek to determine a robust search strategy—possibly a randomized one—that performs well regardless of where the evader hides. We consider the two common formulations: (1) discover the evader at minimal expected cost, and (2) discover the evader by a deadline with maximal probability.

In the first formulation, the searcher wants to minimize the expected cost until discovering the evader, regardless of where the evader hides. In the special case where $c_i = 1$ for all i, Roberts and Gittins (see [14] and [15]) studied the evader's problem of choosing the hiding probability in each cell to maximize the expected time until being found. Based on the values $(\alpha_1, \ldots, \alpha_n)$, in some special cases the optimal hiding probability can be analytically determined, while in general it can be approximated by progressively searching through the probability vector space. Their method, however, is restricted to the case where $c_i = 1$ for all i. In this paper, we present an iterative algorithm to compute the evader's optimal policy for arbitrary search costs $(c_1, \ldots, c_n)$. A byproduct of the algorithm is an improving searcher's policy at each iteration, until we find each player's optimal policy.

In the second formulation, letting $c_i = 1$ for all i, the searcher wants to maximize the probability of discovering the evader by an arbitrary deadline, regardless of where the evader hides. For *a given deadline*, namely a positive integer t, Subelman [16] showed how to allocate the t searches to maximize the probability of discovering the evader, regardless of where the evader hides. In this paper, we strengthen this result by showing how to sequentially allocate the searches over time to maximize the probability of discovering the evader by deadline t, *simultaneously for every* $t = 1, 2, \ldots$. In other words, this policy minimizes the evasion probability by *an arbitrary deadline*.

The rest of this paper proceeds as follows. Section 2 discusses how the searcher can mini-

mize the expected cost to detection against an intelligent evader. Section 3 discusses how the searcher can simultaneously maximize the detection probability by an arbitrary deadline against an intelligent evader. Finally, Section 4 compares the two policies in the case where $c_i = 1$ for all i, and offers some concluding remarks.

2 EXPECTED COST TO DETECTION

Consider the classical search model with an intelligent evader. The evader decides where to hide among n cells and the searcher decides the sequence of cells to search. A search in cell i costs c_i and will discover the evader with probability α_i, if the evader is indeed hidden in cell i, $i = 1, \ldots, n$. In this section, we consider a two-person zero-sum game played by the evader and the searcher, where the evader wishes to maximize the expected cost to detection, while the searcher wishes to minimize it.

The evader's pure strategy space is $\{1, \ldots, n\}$, where each pure strategy corresponds to a cell to hide. A mixed strategy for the evader is a probability vector $\mathbf{p} = (p_1, \ldots, p_n)$ such that the evader hides in cell i with probability p_i, where $p_i \geq 0$ for $i = 1, \ldots, n$, and $\sum_{i=1}^{n} p_i = 1$. The searcher's pure strategy space is $\{1, 2, \ldots, n\}^{\infty}$. Each pure strategy corresponds to a *search sequence* followed by the searcher until discovering the evader. A mixed strategy for the searcher is a probability measure on $\{1, 2, \ldots, n\}^{\infty}$. For a search sequence $\xi \in \{1, 2, \ldots, n\}^{\infty}$, write $V_i(\xi)$ for the expected cost to detection if the evader hides in cell i, for $i \in \{1, \ldots, n\}$. In other words, $V_i(\xi)$ is the payoff of the evader-searcher strategy pair (i, ξ). While the evader's pure strategy space is finite, the searcher's pure strategy space is uncountable, because it is the Cartesian product of a countable number of $\{1, \ldots, n\}$.

The evader's objective function is to determine a mixed strategy to maximize the expected cost to detection for any search sequence, namely,

$$\max_{\mathbf{p}} \inf_{\xi \in \{1,\ldots,n\}^{\infty}} \sum_{i=1}^{n} p_i V_i(\xi). \tag{1}$$

The searcher's objective function is to determine a mixed strategy f over the space of pure search strategies ξ to minimize the expected cost to detection for any hiding cell, namely,

$$\inf_{f} \max_{i \in \{1,\ldots,n\}} \int V_i(\xi) f(\xi) d\xi, \tag{2}$$

where we write $f(\xi)$ for the density of the searcher's mixed strategy.

Since the function $h(p_1, \ldots, p_n) \equiv \sum_{i=1}^{n} p_i V_i(\xi)$ in Equation (1) is a hyperplane in n-dimensional space, it follows that

$$\inf_{\xi \in \{1,\ldots,n\}^{\infty}} \sum_{i=1}^{n} p_i V_i(\xi)$$

is the lower envelope of an uncountable set of hyperplanes, which is therefore a concave function in $(p_1, \ldots, p_n)$. The objective function in Equation (1) thus maximizes this con-

cave function. Define

$$V^* \equiv \max_{\mathbf{p}} \inf_{\xi \in \{1,\ldots,n\}^\infty} \sum_{i=1}^{n} p_i V_i(\xi),$$

and write $\mathbf{p}^*$ for the corresponding maximizer, which is the evader's optimal mixed strategy. By using $\mathbf{p}^*$, the evader can guarantee that the expected cost to detection is, at least, V^*. We next present an iterative algorithm that delivers improving bounds for V^* in each iteration.

The main idea of the algorithm goes as follows. Since the searcher's pure strategy space is uncountable, it is difficult to compute V^* directly, except for special cases. However, if we restrict the searcher to a finite set of search sequences, then we can formulate a two-person zero-sum *matrix game* with a finite number of rows and columns, where each row corresponds to the evader choosing which cell to hide, and each column corresponds to a search sequence for the searcher. It is straightforward to solve a two-person zero-sum matrix game via linear programming (see [17]). If the finite set of search sequences includes a variety of effective search sequences, then the searcher can achieve an expected cost to detection close to V^*.

Below we present an algorithm to add new search sequences iteratively to strengthen the searcher's set of pure strategies. In each iteration, the algorithm solves a two-person zero-sum matrix game, and produces an improved upper bound and a lower bound for V^*.

Algorithm 1.

1. Set $U = \infty$ as an upper bound for V^*, and $L = 0$ as a lower bound for V^*, and pick $\varepsilon > 0$, so that the algorithm will stop when $U/L - 1 < \varepsilon$.
2. Pick an arbitrary set of search sequences, denoted by S, such that the evader's optimal policy against the searcher's pure strategy set S is an interior point. That is, if $(p_1, \ldots, p_n)$ is the searcher's optimal policy against the searcher's pure strategy set S, then $p_i > 0$ for all i.
3. Consider a two-person zero-sum matrix game with n rows and $|S|$ columns, where row i corresponds to the evader hiding in cell i and column j a search sequence in S. Populate the values in the matrix with the value at (i, j) being the corresponding expected cost to detection. Solve the matrix game, and write $\mathbf{p} = (p_1, \ldots, p_n)$ for the optimal mixed strategy for the evader, and v the value of the game.
4. Update $U \leftarrow v$.
5. For the evader's mixed strategy $\mathbf{p}$, determine a corresponding optimal search sequence s, and write $C(\mathbf{p})$ for the corresponding expected cost to detection. Update $L \leftarrow \max(L, C(\mathbf{p}))$.
6. If $U/L - 1 < \varepsilon$, stop; otherwise, update $S \leftarrow S \cup \{s\}$ and go to step 3.

In step 3 of Algorithm 1, the value of the matrix game v is an upper bound for V^*, since

the searcher's mixed strategy using only search sequences in S is a feasible strategy for the searcher. Furthermore, if $S \subseteq S'$, then the searcher cannot do worse using S' than by using S. Hence, the upper bound in each iteration will be at least as good as the one from the previous iteration, and thus the best upper bound to date. In fact, the new upper bound will be strictly better than the previous upper bound, unless we have found the optimal solution; see Proposition 2. On the other hand, the evader's strategy $\mathbf{p}$, together with the optimal search sequence against it, produces a lower bound for V^*, since by using $\mathbf{p}$ the evader can guarantee the expected cost to detection to be no less than $C(\mathbf{p})$. It is, however, possible that the lower bound in the current iteration is worse than lower bounds in previous iterations. Hence, in step 5, we keep the best known lower bound to date.

Proposition 2. In Algorithm 1, write U_k for the upper bound in step 4 in iteration k, for $k = 1, 2, \ldots$. If $U_k > V^*$, then $U_{k+1} < U_k$, for $k = 1, 2, \ldots$.

Proof: In iteration k of Algorithm 1, write ξ_k for the new search sequence added, S_k for the set of search sequences, $\mathbf{p}_k$ the evader's optimal mixed strategy in step 3, and L_k for the lower bound in step 5, for $k = 1, 2, \ldots$. We already know that $U_{k+1} \leq U_k$, since $S_k \subseteq S_{k+1}$. To prove the proposition by contradiction, suppose instead that $U_{k+1} = U_k$, and we will show that $U_k = V^*$.

Since $U_{k+1} = U_k$, then in iteration $k+1$, the searcher has an optimal mixed strategy that does not use ξ_{k+1}, so $\mathbf{p}_k$ is also optimal for the evader in iteration $k+1$. It then follows that

$$C(\mathbf{p}_k) \geq U_{k+1}, \tag{3}$$

for otherwise using $\mathbf{p}_k$ the evader cannot guarantee U_{k+1} in iteration $k+1$. On the other hand, by construction, we have that

$$L_k \geq C(\mathbf{p}_k). \tag{4}$$

Equations (3) and (4) imply that $L_k \geq C(\mathbf{p}_k) \geq U_{k+1} = U_k$. Together with $L_k \leq V^* \leq U_k$, we can conclude that $L_k = V^* = U_k$, which completes the proof. □

Algorithm 1 is an implementation of Kelley's convex cutting plan algorithm (see [18]). The rationale of our algorithm can be best understood via an example with $n = 2$ cells, where the evader's mixed strategy can be delineated by $\mathbf{p} = (p, 1-p)$, with $p \in [0,1]$. Figure 1 illustrates this idea. Each straight line represents the expected cost to detection for a search sequence, plotted against the evader's choice $p \in [0,1]$. The function $C(p)$ is the lower envelope of the set of all search sequences; hence, a concave function in p, as indicated by the bold curve in Figure 1. We seek to determine $V^* = \max_{p \in [0,1]} C(p)$. Suppose in the current iteration, S consists of two search sequences represented by the two solid straight lines s_1 and s_2. By mixing these two search sequences, the searcher can guarantee that the

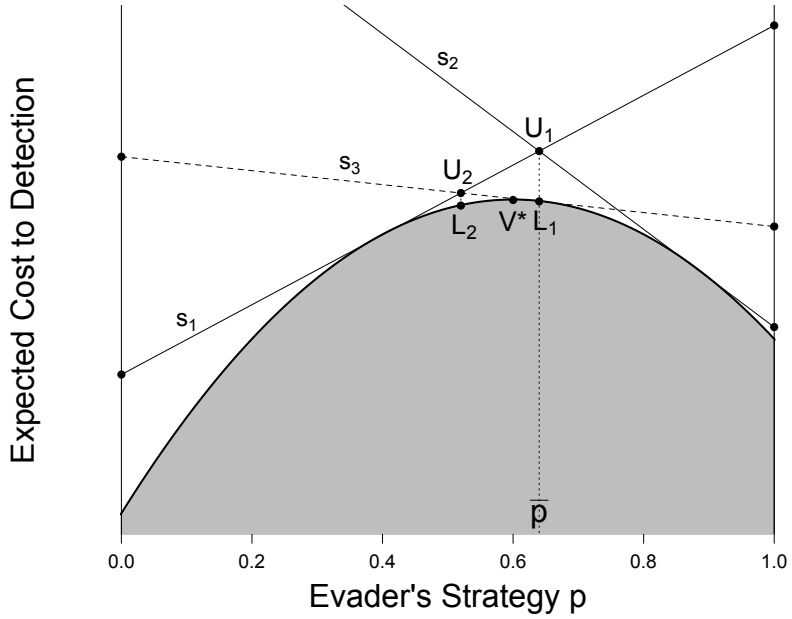

Figure 1: Progressive calculation of upper and lower bounds. The solid tangent lines represent the set of search sequences in the current iteration, and the dashed tangent line is the new search sequence generated to be used in the next iteration.

expected cost to detection is no more than U_1, an upper bound for V^*. The best evader's strategy against S, namely $\bar{p}$, induces a new search sequence (the dashed straight line s_3) and the corresponding expected cost to detection L_1, a lower bound for V^*. Furthermore, by adding this newly generated search sequence into S, in the next iteration we can compute a new upper bound U_2 and a new lower bound L_2, which may or may not be the overall best lower bound to date. In the case $n = 3$, the straight lines are replaced by planes, whose lower envelope becomes a dome.

Conjecture 3. Algorithm 1 will terminate for any $\varepsilon > 0$.

To understand this conjecture intuitively, formulate the evader's problem as a convex opti-

mization problem, with n variables $p_1, p_2, \ldots, p_{n-1}, V$, as follows:

$$\begin{aligned} \max_{p_1,\ldots,p_{n-1},V} \quad & V \\ \text{s.t.} \quad & C\left(p_1,\ldots,p_{n-1},1-\sum_{i=1}^{n-1} p_i\right) - V \geq 0, \\ & p_i \geq 0, \qquad\qquad i = 1,\ldots,n-1, \\ & \sum_{i=1}^{n-1} p_i \leq 1. \end{aligned} \tag{5}$$

In Figure 1, the feasible region of this convex optimization problem is indicated by the gray area. The rationale of Conjecture 3 parallels that of Kelley's convex cutting plane algorithm on page 419 in Luenberger [18]. It has been shown that the sequence produced by Kelley's algorithm will converge to the optimal solution, provided that the objective function and constraints are continuously differentiable. In our problem, the lower envelope of hyperplanes need not be continuously differentiable where the hyperplanes intersect. However, the nondifferentiability does not necessarily hinder the convergence, since finding one of these intersecting hyperplanes as a cut can actually make substantial progress. In addition, if we rewrite constraint (5) as many linear constraints, each of which corresponds to a search sequence, then each of these linear constraints is continuously differentiable. Unfortunately, the number of these linear constraints is infinite, so the convergence does not follow directly from the theorem in Luenberger [18]. Nevertheless, based on these observations, intuitively the algorithm will produce a sequence of policies that converge to the optimal solution. In computational experiments, Algorithm 1 was always able to obtain upper bound U and lower bound L for V^* when we set $U/L - 1 < \varepsilon = 10^{-5}$; see Table 1.

In step 2 of Algorithm 1, the initial set S needs to produce a corresponding optimal evader strategy $\mathbf{p}$ with $p_i > 0$ for all i. Otherwise, if say $p_j = 0$ for some j, then the optimal search sequence against it will never search in cell j, so the jth element in the newly added column will be infinity. Consequently, that newly added column will not be used by the searcher in the next iteration, prohibiting the algorithm from moving forward. In addition, it helps speed up convergence if the initial set S consists of search sequences that are generated from evader's policies that are close to the optimal policy. Gittins [15] demonstrated that in the case $c_i = 1$ for all i, the evader's policy with $\alpha_i p_i$ being a constant for all i often offers a good approximation to the evader's optimal policy. Extending the idea to our case with arbitrary c_i, write $\mathbf{p}'$ for the evader's policy with $\alpha_i p_i / c_i$ being a constant for all i. The optimal search sequence against $\mathbf{p}'$, and those against evader's policies near $\mathbf{p}'$, would be good choices to include in S in step 2. To achieve these two goals, we implement step 2 in Algorithm 1 as follows:

2. (a) Generate the optimal search sequence against evader's policy $\mathbf{p}' = (p'_1, \ldots, p'_n)$, and add it to S. Solve the optimal evader strategy $\mathbf{p} = (p_1, \ldots, p_n)$ against S.
2. (b) If $\mathbf{p}$ is an interior solution, then stop and output S; otherwise, let i be an arbitrary cell with $p_i = 0$. Let $\eta = p'_i$.
2. (c) Obtain a new evader's policy $\mathbf{q}$ by setting

$$\begin{aligned} q_i &\leftarrow \beta\eta, \\ q_j &\leftarrow p_j(1-\beta\eta), \qquad j \neq i, \end{aligned}$$

where $0 < \beta < 1$ is a predetermined scaler.
2. (d) Determine an optimal search sequence s against $\mathbf{q}$. Update $S \leftarrow S \cup \{s\}$.
2. (e) Determine the optimal evader strategy $\mathbf{p}$ against S. If $p_i = 0$, then go to step (c); otherwise, go to step (b).

The rationale of the preceding can be understood as follows. If the evader does not want to hide in cell i, then the searcher needs to spend more time searching the other cells (or less time in cell i), in order to drive down the expected cost to detection. Hence, in each iteration 2. (c)–2. (e), we add a search sequence, which is optimal against an evader who hides in cell i with a smaller probability than the previous iteration, while keeping the ratio of hiding probabilities in the other cells the same as the best up-to-date evader policy. By doing so, the searcher progressively includes stronger search sequences against all cells except cell i, so eventually the evader's optimal policy $\mathbf{p}$ in step 2. (e) will involve hiding in cell i.

The closer β is to 1, the more iterations that are required to come up with a qualified initial set S, but those search sequences will have better quality; hence, fewer iterations are required for steps 3–6 in Algorithm 1. Through numerical tests, we find $\beta \in (0.7, 0.9)$ works well. Table 1 reports the number of iterations required in order to compute V^* to a specified accuracy. We set $\beta = 0.8$, and $c_i = 1$ for all i, and draw α_i independently from the standard uniform distribution.

Table 1: Mean and 90th percentile of the number of search sequences generated in Algorithm 1—in step 2 (initialization) and step 5—until $U/L - 1 < \varepsilon$, based on 1000 independent replications. The values of α_i are drawn from standard uniform random variates, with $\beta = 0.8$, and $c_i = 1$ for all i. The standard error of the sample mean is 0.30%–1.07% of the respective sample mean.

	Number of Cells					
	$n = 2$		$n = 4$		$n = 8$	
ε	Mean	90th	Mean	90th	Mean	90th
10^{-2}	2.6	4	13.4	17	59.0	68
10^{-3}	3.4	4	19.4	23	81.9	92
10^{-4}	4.5	6	24.4	29	102.2	114
10^{-5}	5.2	7	28.3	34	118.9	133

THIS PAGE INTENTIONALLY LEFT BLANK

3 PROBABILITY OF DETECTION BY AN ARBITRARY DEADLINE

Consider a scenario in which the searcher has to give up the search if he has not discovered the evader by a deadline. The total cost of the search is not a concern; instead, the searcher wants to maximize the probability of discovering the evader by the deadline. For a given deadline t, Subelman [16] demonstrates how to allocate the t searches among the n cells to maximize the probability of discovering the evader, regardless of where the evader hides. If the deadline is unannounced at the beginning of the search, then is it possible to allocate the search sequentially in such a way that it maximizes the probability of discovering the evader regardless of when the search has to end? This section presents a search policy to achieve this goal.

At any stage of the search, the state of the search can be delineated by $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ if the searcher has searched x_i times in cell i. While the sum of the components of $\mathbf{x}$ gives t, we sometimes use $\mathbf{x}_t$ to denote the state after t searches. A randomized policy π maps from a state to a probability distribution over n cells, such that in state $\mathbf{x}$ the searcher next searches in cell i with probability $\pi(\mathbf{x}, i) \in [0,1]$, with $\sum_{i=1}^{n} \pi(\mathbf{x}, i) = 1$.

Let $Q_i(\pi, t)$ denote the probability that the searcher will find the evader with policy π within t searches if the evader hides in cell i, for $i = 1, \ldots, n$. If the searcher wants to maximize the probability of discovering the evader within t searches, regardless of where the evader hides, then we need to determine policy π that solves

$$\min_{\pi} \max_{i} Q_i(\pi, t), \tag{6}$$

simultaneously for every $t = 1, 2, \ldots$. Whereas Subelman [16] shows how to solve (6) for a predetermined t, our contribution is to show that it is possible to achieve Subelman's results *simultaneously for every* $t = 1, 2, \ldots$.

3.1 Subelman's Results

For a given t, Subelman [16] solved the problem in (6) by formulating a two-person zero-sum game, where the searcher wants to maximize the probability of discovering the evader within t searches, while the evader chooses where to hide to minimize that probability. The optimal search solution can be described as follows. There exist nonnegative integers $\mu_i(t)$, and fractional numbers $\Psi_i(t) \in [0,1)$, for $i = 1, \ldots, n$, with

$$\sum_{i=1}^{n} \mu_i(t) + \sum_{i=1}^{n} \Psi_i(t) = t,$$

such that cell i will receive $\mu_i(t)$ searches with probability $1-\Psi_i(t)$, or $\mu_i(t)+1$ searches with probability $\Psi_i(t)$, for $i=1,\ldots,n$. In other words, cell i will receive at least $\mu_i(t)$ searches, and will receive an additional search with probability $\Psi_i(t)$. Subleman showed how to compute $\mu_i(t)$ and $\Psi_i(t)$ for any positive integer t. With the optimal search strategy, the probability of discovering the evader within t searches is the same wherever the evader hides.

In what follows, we explain how to construct a policy $\pi(\mathbf{x},i)$ that achieves the probability distribution of allocating t searches according to Subelman's results simultaneously for every $t=1,2,\ldots$. Such a policy maximizes the probability of discovering the evader within t searches simultaneously for every t, so it does not require knowing the search deadline at the beginning of the search.

3.2 Sequential Allocation Problem

By applying a randomized policy π, we can generate a random search sequence, denoted by $\{X(t),t=1,2,\ldots\}$, such that $X(t)=i$ indicates that cell i is searched at time t. According to Subelman [16], for the random search sequence to maximize the probability of discovering the evader by time t, we need $X(1),X(2),\ldots,X(t)$ to meet the following constraints:

$$P\left\{\sum_{s=1}^{t}\mathbf{1}(X(s)=i)=\mu_i(t)\right\}=1-\Psi_i(t), \qquad \text{for } i=1,\ldots,n; \tag{7}$$

$$P\left\{\sum_{s=1}^{t}\mathbf{1}(X(s)=i)=\mu_i(t)+1\right\}=\Psi_i(t), \qquad \text{for } i=1,\ldots,n; \tag{8}$$

where $\mathbf{1}(A)$ is the indicator function, returning 1 or 0 depending on whether event A is true or false. To prove that the searcher has a uniformly optimal policy to maximize the detection probability by an arbitrary deadline, we need $\{X(t),t=1,2,\ldots\}$ to meet constraints (7) and (8) simultaneously for $t=1,2,\ldots$.

For notational convenience, let $\mu_i(0)=0$ and $\Psi_i(0)=0$, for all i. Define

$$\begin{aligned}\Delta_i(t)\equiv P\{X(t)=i\}&=E\left[\sum_{s=1}^{t}\mathbf{1}(X(t)=i)\right]-E\left[\sum_{s=1}^{t-1}\mathbf{1}(X(t)=i)\right]\\&=\mu_i(t)+\Psi_i(t)-(\mu_i(t-1)+\Psi_i(t-1)),\end{aligned} \tag{9}$$

so that $\Delta_i(t)$ is the probability that the tth search is allocated to cell i. In other words, based on how the first $t-1$ searches have been allocated, we need to find a way to allocate the tth search, such that the overall probability of allocating it to cell i is $\Delta_i(t)$, and that constraints

(7) and (8) are met at time t. Note that given $\Delta_i(t)$, $t = 1,2,\ldots$, it is straightforward to compute $\mu_i(t)$ and $\Psi_i(t)$, $i = 1,2,\ldots$.

3.3 The Case with Two Cells

Consider the case with $n = 2$ cells. Given $\Delta_i(t)$, for $i = 1,\ldots,n$ and $t = 1,2,\ldots$, our goal is to determine a state-dependent distribution for which cell to search next, such that Subelman's results hold simultaneously for every $t = 1,2,\ldots$. For each combination of possible states $(\mathbf{x}_t, \mathbf{x}_{t+1})$ at times t and $t+1$, write $P(\mathbf{x}_t, \mathbf{x}_{t+1})$ for their joint probability. The marginal probabilities at times t and $t+1$ are therefore

$$P(\mathbf{x}_t) = \sum_{x_{t+1}} P(\mathbf{x}_t, \mathbf{x}_{t+1}), \qquad P(\mathbf{x}_{t+1}) = \sum_{x_t} P(\mathbf{x}_t, \mathbf{x}_{t+1}).$$

Additionally, write $\mathbf{e}_i$ for a vector of length two with a one at position i and a zero at the other position; namely $\mathbf{e}_1 = (1,0)$ and $\mathbf{e}_2 = (0,1)$. For a given state $\mathbf{x}_t$ at time t, the new state at time $t+1$ becomes $\mathbf{x}_t + \mathbf{e}_i$ if cell i is searched at time $t+1$. Generally speaking, at time t we will determine $P(\mathbf{x}_t)$ and $P(\mathbf{x}_t, \mathbf{x}_t + \mathbf{e}_i)$ that meet constraints (7) and (8), which, in turn, yields the corresponding policy π defined by

$$\pi(\mathbf{x}_t, i) = \frac{P(\mathbf{x}_t, \mathbf{x}_t + \mathbf{e}_i)}{P(\mathbf{x}_t)}. \tag{10}$$

To begin, consider $t = 1$ (the first search). We can simply set

$$\pi((0,0),1) = P\{X(1) = 1\} = \Delta_1(1), \qquad \pi((0,0),2) = P\{X(1) = 2\} = \Delta_2(1)$$

to meet constraints (7)–(8). To show that we can generate $X(t)$ to meet constraints (7)–(8) for $t = 1,2,\ldots$, we use mathematical induction on t. Suppose that we have generated $X(s)$ for $s = 1,\ldots,t$, which meet constraints (7)–(8). We next show how to do so for time $t+1$. For notational convenience, let $\psi_i = \Psi_i(t)$, $\delta_i = \Delta_i(t+1)$, and $\psi_i' = \Psi_i(t+1)$, for $i = 1,2$. Recall that, by definition, in Equation (9), for $i = 1,2$,

$$\psi_i' = \begin{cases} \psi_i + \delta_i, & \text{if } \psi_i + \delta_i < 1, \\ \psi_i + \delta_i - 1, & \text{if } \psi_i + \delta_i \geq 1. \end{cases}$$

In addition, $\psi_i, \psi_i' \in [0,1)$, and $\delta_i \in [0,1]$, for $i = 1,2$. Consider three cases:

1. $\psi_1 + \psi_2 = 0$.
   In this case, we have that $\psi_1 = \psi_2 = 0$. At time $t+1$, search in cell 1 with probability

δ_1 and in cell 2 with probability δ_2. The corresponding policy is

$$\pi((\mu_1(t),\mu_2(t)),1)=\delta_1, \qquad \pi((\mu_1(t),\mu_2(t)),2)=\delta_2.$$

2. $\psi_1+\psi_2=1$, and $\psi_1'+\psi_2'=0$.
Because $\psi_1'=\psi_2'=0$, we have that $\psi_i+\delta_i=1$ for $i=1,2$. If cell 1 has received $\mu_1(t)$ searches through time t ($\mu_2(t)+1$ searches in cell 2), then at time $t+1$ search in cell 1. If cell 1 has received $\mu_1(t)+1$ searches through time t ($\mu_2(t)$ searches in cell 2), then at time $t+1$ search in cell 2. The corresponding policy is

$$\begin{aligned}\pi((\mu_1(t),\mu_2(t)+1),1)&=1, \qquad \pi((\mu_1(t),\mu_2(t)+1),2)=0;\\ \pi((\mu_1(t)+1,\mu_2(t)),1)&=0, \qquad \pi((\mu_1(t)+1,\mu_2(t)),2)=1.\end{aligned}$$

3. $\psi_1+\psi_2=1$, and $\psi_1'+\psi_2'=1$. Through time t, cell i has received at least $\mu_i(t)$ searches, for $i=1,2$, with $\mu_1(t)+\mu_2(t)=t-1$. The additional search is either in cell 1 with probability ψ_1, or in cell 2 with probability ψ_2.
Since $\psi_1'+\psi_2'=1$, either $\psi_1+\delta_1<1$ and $\psi_2+\delta_2>1$, or $\psi_1+\delta_1>1$ and $\psi_2+\delta_2<1$; we assume the former case without loss of generality. It then follows that $\mu_1(t+1)=\mu_1(t)$, $\psi_1'=\psi_1+\delta_1$, and $\mu_2(t+1)=\mu_2(t)+1$, $\psi_2'=\psi_2+\delta_2-1$.
To allocate the search at time $t+1$, we need to ensure that through time $t+1$, cell 1 receives at least $\mu_1(t)$ searches and cell 2 at least $\mu_2(t)+1$ searches, with the additional search either in cell 1 with probability ψ_1', or in cell 2 with probability ψ_2'. It is then straightforward to verify that the joint probability distribution below meets constraints (7)–(8).

	$\mathbf{x}_{t+1}-\boldsymbol{\mu}(t)$		
$\mathbf{x}_t-\boldsymbol{\mu}(t)$	$(1,1)$	$(0,2)$	
$(1,0)$	ψ_1	0	ψ_1
$(0,1)$	$\psi_2-\psi_2'$	ψ_2'	ψ_2
	ψ_1'	ψ_2'	

From the joint probability table, we can use (10) to determine the corresponding policy:

$$\begin{aligned}\pi((\mu_1(t)+1,\mu_2(t)),1)&=0, \quad \pi((\mu_1(t)+1,\mu_2(t)),2)=1;\\ \pi((\mu_1(t),\mu_2(t)+1),1)&=1-\frac{\psi_2'}{\psi_2}, \quad \pi((\mu_1(t),\mu_2(t)+1),2)=\frac{\psi_2'}{\psi_2}.\end{aligned}$$

Remark 4. The method just presented can be extended to a search problem with $n\geq 3$ cells, which will be explained in Section 3.4. In the case of $n=2$ cells, however, there is a rather elegant method that requires only one uniform $(0,1)$ random variable to produce a search policy that achieves Equation (6) simultaneously for every $t=1,2,\ldots$. Suppose

the generated uniform $(0,1)$ random variable is u. At $t=1$, search in cell 1 if $u \le \Delta_1(1)$; otherwise, search in cell 2. For $t \ge 2$, define

$$I(t) = (\Psi_1(t-1), \min(1, \Psi_1(t-1)+\Delta_1(t))] \cup [0, \max(0, \Psi_1(t-1)+\Delta_1(t)-1)]. \quad (11)$$

Figure 2 illustrates this method, where bold line segments indicate $I(t)$, for $t = 1,2,\ldots$. At $t \ge 2$, let the searcher search in cell 1 if $u \in I(t)$; otherwise, search in cell 2.

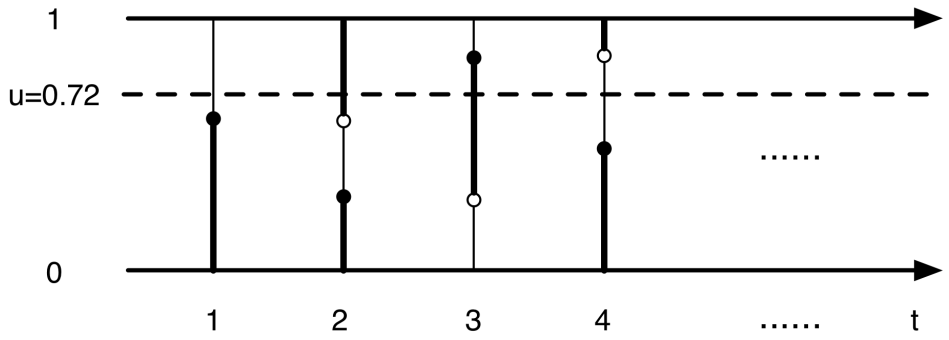

Figure 2: Generating the search policy with one uniform $(0,1)$ random variable when $n=2$. The bold line segments correspond to $I(t)$ defined in Equation (11). Generate a uniform $(0,1)$ random variable and draw the corresponding dashed line. At time t, search in cell 1 if the dashed line crosses a bold line segment or search in cell 2 if it does not. In this example, with $u = 0.72$, the searcher follows the search sequence $2,1,1,2,\ldots$.

As seen in Equation (11), the probability of searching in cell 1 at time t is $|I(t)| = \Delta_1(t)$. In addition, with the construction as illustrated in Figure 2, through time t the searcher will search in cell 1 either $\mu_1(t)$ times or $\mu_1(t)+1$ times. Hence, such a search policy achieves Equation (6) simultaneously for every $t = 1,2,\ldots$.

3.4 The Case with Three Cells

We are given $\Delta_i(t)$, $\Psi_i(t)$, and $\mu_i(t)$, for $i = 1,2,3$, and need to generate $X(t)$ to meet constraints (7)–(8) for $t = 1,2,\ldots$. Recall that, for $i = 1,2,3$,

$$\mu_i(t) = \left\lfloor \sum_{s=1}^{t} \Delta_i(s) \right\rfloor,$$

$$\Psi_i(t) = \sum_{s=1}^{t} \Delta_i(s) - \mu_i(t),$$

where $\mu_i(t)$ is the integral part and $\Psi_i(t)$ is the fractional part of the desired number of searches.

To begin, consolidate cells 1 and 2 into a single cell, referred to as cell A, with

$$\begin{aligned}\Delta_{\mathrm{A}}(t) &= \Delta_1(t) + \Delta_2(t),\\ \mu_{\mathrm{A}}(t) &= \mu_1(t) + \mu_2(t) + \lfloor \Psi_1(t) + \Psi_2(t) \rfloor,\\ \Psi_{\mathrm{A}}(t) &= \Psi_1(t) + \Psi_2(t) - \lfloor \Psi_1(t) + \Psi_2(t) \rfloor,\end{aligned}$$

for $t = 1, 2, \ldots$. Between cells A and 3, we can apply the method in Section 3.3 to generate a search sequence that matches constraints (7)–(8) at $t = 1, 2, \ldots$. If, at time t, the search is allocated to cell 3, then we set $X(t) = 3$. If, at time t, the search is allocated to cell A, then we need to decide whether $X(t) = 1$ or $X(t) = 2$.

To show that we can generate $X(t)$ that meet constraints (7)–(8), for $t = 1, 2, \ldots$, we use mathematical induction on t. At $t = 1$, if the first search goes to cell A, then assign i to $X(t)$ with probability $\Delta_i(1)/(\Delta_1(1) + \Delta_2(1))$, for $i = 1, 2$. Suppose that we have generated $X(s)$ for $s = 1, \ldots, t$, such that the search policy meets constraints (7)–(8). We next show how to do so at time $t+1$. For notational convenience, let $\psi_i = \Psi_i(t)$, $\delta_i = \Delta_i(t+1)$, and $\psi_i' = \Psi_i(t+1)$, for $i = 1, 2$. Recall that, by definition, for $i = 1, 2$,

$$\psi_i' = \begin{cases} \psi_i + \delta_i, & \text{if } \psi_i + \delta_i < 1,\\ \psi_i + \delta_i - 1, & \text{if } \psi_i + \delta_i \geq 1.\end{cases}$$

In addition, $\psi_i, \psi_i' \in [0, 1)$, and $\delta_i \in [0, 1]$, for $i = 1, 2$. Consider two cases based on the value of $\psi_1 + \psi_2$.

3.4.1 The Case $0 \leq \psi_1 + \psi_2 < 1$

In this case, we have $\Psi_{\mathrm{A}}(t) = \psi_1 + \psi_2$ and $\mu_{\mathrm{A}}(t) = \mu_1(t) + \mu_2(t)$. With the induction hypothesis, through time t, cells 1 and 2 have received $(\mu_1(t), \mu_2(t))$ searches with probability $1 - \Psi_{\mathrm{A}}(t) = 1 - \psi_1 - \psi_2$, or $(\mu_1(t) + 1, \mu_2(t))$ searches with probability ψ_1, or $(\mu_1(t), \mu_2(t) + 1)$ searches with probability ψ_2. Denote these three cases at time t by $(0, 0)$, $(1, 0)$, and $(0, 1)$, respectively. Consider three cases based on δ_1 and δ_2.

1. $\psi_1 + \delta_1 < 1$, $\psi_2 + \delta_2 < 1$, and $\psi_1 + \delta_1 + \psi_2 + \delta_2 < 1$.
   In this case, we have $\psi_1' = \psi_1 + \delta_1$, $\psi_2' = \psi_2 + \delta_2$, and $\psi_1' + \psi_2' < 1$. In addition, $\mu_{\mathrm{A}}(t+1) = \mu_{\mathrm{A}}(t) = \mu_1(t) + \mu_2(t)$ and $\Psi_{\mathrm{A}}(t+1) = \psi_1' + \psi_2'$. Through time $t+1$, cell A will receive either $\mu_{\mathrm{A}}(t)$ searches with probability $1 - \Psi_{\mathrm{A}}(t+1) = 1 - \psi_1' - \psi_2'$, or $\mu_{\mathrm{A}}(t) + 1$ searches with probability $\Psi_{\mathrm{A}}(t+1) = \psi_1' + \psi_2'$. At time $t+1$, the search allocation between cells 1 and 2 needs to be either $(\mu_1(t), \mu_2(t))$, $(\mu_1(t) + 1, \mu_2(t))$,

or $(\mu_1(t), \mu_2(t)+1)$. Denote these three cases at time $t+1$ by $(0,0)$, $(1,0)$, and $(0,1)$, respectively. It is then straightforward to verify that the joint probability distribution below between search allocations at time t and time $t+1$ meets all constraints. By allocating the search at time $t+1$ to obtain this joint probability distribution, the resulting search policy meets all constraints at time $t+1$.

	$\mathbf{x}_{t+1} - \boldsymbol{\mu}(t)$			
$\mathbf{x}_t - \boldsymbol{\mu}(t)$	$(0,0)$	$(1,0)$	$(0,1)$	
$(0,0)$	$1-\psi_1'-\psi_2'$	$\psi_1'-\psi_1$	$\psi_2'-\psi_2$	$1-\psi_1-\psi_2$
$(1,0)$	0	ψ_1	0	ψ_1
$(0,1)$	0	0	ψ_2	ψ_2
	$1-\psi_1'-\psi_2'$	ψ_1'	ψ_2'	

2. $\psi_1+\delta_1<1$, $\psi_2+\delta_2<1$, and $\psi_1+\delta_1+\psi_2+\delta_2\geq 1$.
In this case, we have $\psi_1'=\psi_1+\delta_1$, $\psi_2'=\psi_2+\delta_2$, and $\psi_1'+\psi_2'\geq 1$. In addition, $\mu_{\mathrm{A}}(t+1)=\mu_{\mathrm{A}}(t)+1=\mu_1(t)+\mu_2(t)+1$ and $\Psi_{\mathrm{A}}(t+1)=\psi_1'+\psi_2'-1$. Through time $t+1$, cell A will receive either $\mu_{\mathrm{A}}(t)$ searches with probability $1-\Psi_{\mathrm{A}}(t+1)=2-\psi_1'-\psi_2'$, or $\mu_{\mathrm{A}}(t)+1$ searches with probability $\Psi_{\mathrm{A}}(t+1)=\psi_1'+\psi_2'-1$. At time $t+1$, the search allocation between cells 1 and 2 needs to be either $(\mu_1(t)+1,\mu_2(t))$, $(\mu_1(t),\mu_2(t)+1)$, or $(\mu_1(t)+1,\mu_2(t)+1)$. Denote these three cases by $(1,0)$, $(0,1)$, and $(1,1)$, respectively. The joint probability distribution below produces a search policy that meets all constraints at time $t+1$.

	$\mathbf{x}_{t+1} - \boldsymbol{\mu}(t)$			
$\mathbf{x}_t - \boldsymbol{\mu}(t)$	$(1,0)$	$(0,1)$	$(1,1)$	
$(0,0)$	$1-\psi_2'-\psi_1(1-\frac{\psi_1'+\psi_2'-1}{\psi_1+\psi_2})$	$1-\psi_1'-\psi_2(1-\frac{\psi_1'+\psi_2'-1}{\psi_1+\psi_2})$	0	$1-\psi_1-\psi_2$
$(1,0)$	$\psi_1(1-\frac{\psi_1'+\psi_2'-1}{\psi_1+\psi_2})$	0	$\psi_1\frac{\psi_1'+\psi_2'-1}{\psi_1+\psi_2}$	ψ_1
$(0,1)$	0	$\psi_2(1-\frac{\psi_1'+\psi_2'-1}{\psi_1+\psi_2})$	$\psi_2\frac{\psi_1'+\psi_2'-1}{\psi_1+\psi_2}$	ψ_2
	$1-\psi_2'$	$1-\psi_1'$	$\psi_1'+\psi_2'-1$	

3. $\psi_1+\delta_1<1$, $\psi_2+\delta_2\geq 1$ (the case $\psi_1+\delta_1\geq 1$, $\psi_2+\delta_2<1$ can be dealt with in a similar way).
In this case, we have $\psi_1'=\psi_1+\delta_1$, $\psi_2'=\psi_2+\delta_2-1$, and $\psi_1'+\psi_2'<1$. In addition, $\mu_{\mathrm{A}}(t+1)=\mu_{\mathrm{A}}(t)+1=\mu_1(t)+\mu_2(t)+1$ and $\Psi_{\mathrm{A}}(t+1)=\psi_1'+\psi_2'$. Through time $t+1$, cell A will receive either $\mu_{\mathrm{A}}(t)$ searches with probability $1-\Psi_{\mathrm{A}}(t+1)=1-\psi_1'-\psi_2'$, or $\mu_{\mathrm{A}}(t)+1$ searches with probability $\Psi_{\mathrm{A}}(t+1)=\psi_1'+\psi_2'$. At time $t+1$, the search allocation between cells 1 and 2 needs to be either $(\mu_1(t),\mu_2(t)+1)$, $(\mu_1(t)+1,\mu_2(t)+1)$, or $(\mu_1(t),\mu_2(t)+2)$. Denote these three cases by $(0,1)$, $(1,1)$, and $(0,2)$, respectively. The joint probability distribution below produces a search policy that meets all constraints at time $t+1$.

	$\mathbf{x}_{t+1}-\boldsymbol{\mu}(t)$			
$\mathbf{x}_t-\boldsymbol{\mu}(t)$	$(0,1)$	$(1,1)$	$(0,2)$	
$(0,0)$	$1-\psi_1-\psi_2$	0	0	$1-\psi_1-\psi_2$
$(1,0)$	0	ψ_1	0	ψ_1
$(0,1)$	$\psi_1+\psi_2-\psi_1'-\psi_2'$	$\psi_1'-\psi_1$	ψ_2'	ψ_2
	$1-\psi_1'-\psi_2'$	ψ_1'	ψ_2'	

3.4.2 The Case $1 \le \psi_1+\psi_2 < 2$

In this case, $\Psi_{\mathrm{A}}(t)=\psi_1+\psi_2-1$, and $\mu_{\mathrm{A}}(t)=\mu_1(t)+\mu_2(t)+1$. With the induction hypothesis through time t, cells 1 and 2 have received $(\mu_1(t)+1,\mu_2(t))$ searches with probability $1-\psi_2$, or $(\mu_1(t),\mu_2(t)+1)$ searches with probability $1-\psi_1$, or $(\mu_1(t)+1,\mu_2(t)+1)$ searches with probability $\Psi_{\mathrm{A}}(t)=\psi_1+\psi_2-1$. Denote these three cases at time t by $(1,0)$, $(0,1)$, and $(1,1)$, respectively. Consider four cases based on δ_1 and δ_2.

1. $\psi_1+\delta_1<1$, $\psi_2+\delta_2<1$.
   In this case, we have $\psi_1'=\psi_1+\delta_1$, $\psi_2'=\psi_2+\delta_2$, and $1\le\psi_1'+\psi_2'<2$. In addition, $\mu_{\mathrm{A}}(t+1)=\mu_{\mathrm{A}}(t)=\mu_1(t)+\mu_2(t)+1$ and $\Psi_{\mathrm{A}}(t+1)=\psi_1'+\psi_2'-1$. Through time $t+1$, cell A will receive either $\mu_{\mathrm{A}}(t)$ searches with probability $1-\Psi_{\mathrm{A}}(t+1)=2-\psi_1'-\psi_2'$, or $\mu_{\mathrm{A}}(t)+1$ searches with probability $\Psi_{\mathrm{A}}(t+1)=\psi_1'+\psi_2'-1$. At time $t+1$, the search allocation between cells 1 and 2 needs to be either $(\mu_1(t)+1,\mu_2(t))$, $(\mu_1(t),\mu_2(t)+1)$, or $(\mu_1(t)+1,\mu_2(t)+1)$. Denote these three cases at time $t+1$ by $(1,0)$, $(0,1)$, and $(1,1)$, respectively. The joint probability distribution below produces a search policy that meets all constraints at time $t+1$.

	$\mathbf{x}_{t+1}-\boldsymbol{\mu}(t)$			
$\mathbf{x}_t-\boldsymbol{\mu}(t)$	$(1,0)$	$(0,1)$	$(1,1)$	
$(1,0)$	$1-\psi_2'$	0	$\psi_2'-\psi_2$	$1-\psi_2$
$(0,1)$	0	$1-\psi_1'$	$\psi_1'-\psi_1$	$1-\psi_1$
$(1,1)$	0	0	$\psi_1+\psi_2-1$	$\psi_1+\psi_2-1$
	$1-\psi_2'$	$1-\psi_1'$	$\psi_1'+\psi_2'-1$	

2. $\psi_1+\delta_1<1$, $\psi_2+\delta_2\ge 1$, and $\psi_1'+\psi_2'<1$ (the case $\psi_1+\delta_1\ge 1$, $\psi_2+\delta_2<1$ can be dealt with in a similar way).
   In this case, we have $\psi_1'=\psi_1+\delta_1$, $\psi_2'=\psi_2+\delta_2-1$, and $\psi_1'+\psi_2'<1$. In addition, $\mu_{\mathrm{A}}(t+1)=\mu_{\mathrm{A}}(t)=\mu_1(t)+\mu_2(t)+1$ and $\Psi_{\mathrm{A}}(t+1)=\psi_1'+\psi_2'$. Through time $t+1$, cell A will receive either $\mu_{\mathrm{A}}(t)$ searches with probability $1-\Psi_{\mathrm{A}}(t+1)=1-\psi_1'-\psi_2'$, or $\mu_{\mathrm{A}}(t)+1$ searches with probability $\Psi_{\mathrm{A}}(t+1)=\psi_1'+\psi_2'$. At time $t+1$, the search allocation between cells 1 and 2 needs to be either $(\mu_1(t),\mu_2(t)+1)$, $(\mu_1(t)+1,\mu_2(t)+1)$, or $(\mu_1(t),\mu_2(t)+2)$. Denote these three cases at time $t+1$ by $(0,1)$, $(1,1)$, and $(0,2)$, respectively. The joint probability distribution below produces a search policy that meets all constraints at time $t+1$.

	$\mathbf{x}_{t+1}-\boldsymbol{\mu}(t)$			
$\mathbf{x}_t-\boldsymbol{\mu}(t)$	$(0,1)$	$(1,1)$	$(0,2)$	
$(1,0)$	0	$1-\psi_2$	0	$1-\psi_2$
$(0,1)$	$1-\psi_1'-\psi_2'$	$\psi_1'-\psi_1$	ψ_2'	$1-\psi_1$
$(1,1)$	0	$\psi_1+\psi_2-1$	0	$\psi_1+\psi_2-1$
	$1-\psi_1'-\psi_2'$	ψ_1'	ψ_2'	

3. $\psi_1+\delta_1<1$, $\psi_2+\delta_2\geq 1$, and $\psi_1'+\psi_2'\geq 1$ (the case $\psi_1+\delta_1\geq 1$, $\psi_2+\delta_2<1$ can be dealt with in a similar way).
   In this case, we have $\psi_1'=\psi_1+\delta_1$, $\psi_2'=\psi_2+\delta_2-1$, and $1\leq\psi_1'+\psi_2'<2$. In addition, $\mu_{\mathrm{A}}(t+1)=\mu_{\mathrm{A}}(t)+1=\mu_1(t)+\mu_2(t)+2$ and $\Psi_{\mathrm{A}}(t+1)=\psi_1'+\psi_2'-1$. Through time $t+1$, cell A will receive either $\mu_{\mathrm{A}}(t)+1$ searches with probability $1-\Psi_{\mathrm{A}}(t+1)=2-\psi_1'-\psi_2'$, or $\mu_{\mathrm{A}}(t)+2$ searches with probability $\Psi_{\mathrm{A}}(t+1)=\psi_1'+\psi_2'-1$. At time $t+1$, the search allocation between cells 1 and 2 needs to be either $(\mu_1(t)+1,\mu_2(t)+1)$, $(\mu_1(t),\mu_2(t)+2)$, or $(\mu_1(t)+1,\mu_2(t)+2)$. Denote these three cases at time $t+1$ by $(1,1)$, $(0,2)$, and $(1,2)$, respectively. The joint probability distribution below produces a search policy that meets all constraints at time $t+1$.

	$\mathbf{x}_{t+1}-\boldsymbol{\mu}(t)$			
$\mathbf{x}_t-\boldsymbol{\mu}(t)$	$(1,1)$	$(0,2)$	$(1,2)$	
$(1,0)$	$1-\psi_2$	0	0	$1-\psi_2$
$(0,1)$	$\psi_1'-\psi_1$	$1-\psi_1'$	0	$1-\psi_1$
$(1,1)$	$\psi_1+\psi_2-\psi_1'-\psi_2'$	0	$\psi_1'+\psi_2'-1$	$\psi_1+\psi_2-1$
	$1-\psi_2'$	$1-\psi_1'$	$\psi_1'+\psi_2'-1$	

4. $\psi_1+\delta_1\geq 1$, $\psi_2+\delta_2\geq 1$.
   In this case, we have $\psi_1'=\psi_1+\delta_1-1$, $\psi_2'=\psi_2+\delta_2-1$, and $\psi_1'+\psi_2'<1$. In addition, $\mu_{\mathrm{A}}(t+1)=\mu_{\mathrm{A}}(t)+1=\mu_1(t)+\mu_2(t)+2$ and $\Psi_{\mathrm{A}}(t+1)=\psi_1'+\psi_2'$. Through time $t+1$, cell A will receive either $\mu_{\mathrm{A}}(t)+1$ searches with probability $1-\Psi_{\mathrm{A}}(t+1)=1-\psi_1'-\psi_2'$, or $\mu_{\mathrm{A}}(t)+2$ searches with probability $\Psi_{\mathrm{A}}(t+1)=\psi_1'+\psi_2'$. At time $t+1$, the search allocation between cells 1 and 2 needs to be either $(\mu_1(t)+1,\mu_2(t)+1)$, $(\mu_1(t)+2,\mu_2(t)+1)$, or $(\mu_1(t)+1,\mu_2(t)+2)$. Denote these three cases at time $t+1$ by $(1,1)$, $(2,1)$, and $(1,2)$, respectively. The joint probability distribution below produces a search policy that meets all constraints at time $t+1$.

	$\mathbf{x}_{t+1}-\boldsymbol{\mu}(t)$			
$\mathbf{x}_t-\boldsymbol{\mu}(t)$	$(1,1)$	$(2,1)$	$(1,2)$	
$(1,0)$	$1-\psi_2$	0	0	$1-\psi_2$
$(0,1)$	$1-\psi_1$	0	0	$1-\psi_1$
$(1,1)$	$\psi_1+\psi_2-\psi_1'-\psi_2'-1$	ψ_1'	ψ_2'	$\psi_1+\psi_2-1$
	$1-\psi_1'-\psi_2'$	ψ_1'	ψ_2'	

3.5 The Case with Four or More Cells

Denote the consolidation of cells 1 through k as A_k, for $k = 1, 2, 3, \ldots, n-1$. First, consider cell A_{n-1} and cell n, and use the method in Section 3.3 to generate a search sequence between these two cells. Second, use the method in Section 3.4 to split the searches in cell A_{n-1} between cell A_{n-2} and cell $n-1$. Repeat the procedure to split the searches in cell A_k between cell A_{k-1} and cell k, for $k = n-1, n-2, \ldots, 2$, to have a complete search sequence.

4 THE CASE OF IDENTICAL SEARCH COSTS

In this section, we consider the special case where $c_i = 1$ for all i, in order to compare the two objective functions and their respective optimal policies. In the classical search model—where the evader's mixed strategy is known—if $c_i = 1$ for all i, then the search policy that minimizes the expected time to detection also maximizes the probability of discovering the evader by time t, for every $t = 1, 2, \ldots$ (see [3], [4]). In our model, however, where there is an intelligent evader, the searcher's optimal policy that minimizes the expected time to detection (as discussed in Section 2) and the searcher's optimal policy that maximizes the probability of detecting the evader by an arbitrary deadline (as discussed in Section 3) need not be the same, except for the special case when $\alpha_i = \alpha$ for all i.

Proposition 5. If $c_i = 1$ and $\alpha_i = \alpha$ for $i = 1, \ldots, n$, then any search policy that achieves Equation (6) simultaneously for every $t = 1, 2, \ldots$ also minimizes the expected time to detection, regardless of where the evader hides.

Proof: Since $\alpha_i = \alpha$ for $i = 1, \ldots, n$, a search policy that achieves Equation (6) simultaneously for every $t = 1, 2, \ldots$ must have

$$\Delta_i(t) = \frac{1}{n}, \qquad \text{and} \qquad \mu_i(t) = \left\lfloor \frac{t}{n} \right\rfloor,$$

for $i = 1, \ldots, n$ and $t = 1, 2, \ldots$. One implementation of such a policy is to allocate the first n searches based on a random permutation of $\{1, \ldots, n\}$, and then repeat that permutation (or generate a new random permutation) for the next n searches, and so on. The policy is symmetric among the n cells, so the expected time to detection is the same regardless of where the evader hides, thus minimizing it. □

When the detection probabilities α_i differ among the cells, then the searcher's optimal policies against the two objectives need not be the same. Consider an example with $n = 2$ cells and $(\alpha_1, \alpha_2) = (0.5, 0.75)$. When the objective function is the expected time to detection, Roberts and Gittins [14] showed that the evader's optimal hiding probability is $(0.6, 0.4)$. It then follows that the searcher's optimal policy—in the first three searches—is to follow the sequence 1-2-1 with probability 0.8, or 2-1-1 with probability 0.2. (After the first three searches, the searcher's optimal policy is not unique.) When the objective function is to maximize the probability of detecting the evader by an arbitrary deadline, the searcher's optimal policy at $t = 1$, however, is to search in cell 1 with probability 0.6, or in cell 2 with probability 0.4. As seen in Table 2, the policy that maximizes the detection probability by an arbitrary deadline yields an expected time to detection only 1.79% above the optimal value.

If the searcher follows the sequence 1-2-1 with probability 0.8, or 2-1-1 with probability 0.2, in the first three searches, then the detection probability at $t = 1$ is $0.8 \times 0.5 = 0.4$ if the evader hides in cell 1, or $0.2 \times 0.75 = 0.15$ if the evader hides in cell 2. In other words, the searcher can guarantee a detection probability of 0.15, regardless of where the evader hides. As seen in Table 2, this policy does not always produce near-optimal detection probabilities for different deadlines.

Table 2: Performance of the two policies for $(\alpha_1, \alpha_2) = (0.5, 0.75)$. The optimal values are in bold, and the numbers in parentheses indicate suboptimality.

Objective function	Expected time to detection	Probability of not detecting the evader after t searchers		
		$t = 1$	$t = 2$	$t = 3$
Expected time to detection	**2.8**	17/20 (21.43%)	1/2 (14.29%)	**1/4**
Detection probability by an arbitrary deadline	2.85 (1.79%)	**7/10**	**7/16**	**1/4**

We further test (α_1, α_2) on a grid, where we vary α_1 from $0.1, 0.2, 0.3, \ldots, 0.8$, and α_2 from $\alpha_1 + 0.1, \alpha_1 + 0.2, \ldots, 0.9$. Among these 36 combinations, the policy that maximizes the detection probability by an arbitrary deadline, on average, produces expected times to detection that are 1.32% suboptimal, with the maximum being 4.04%. On the other hand, since the optimal policy that minimizes the expected time to detection is not unique, it takes extra effort if we want to determine which of them maximizes the detection probability by a given deadline. If we choose an optimal policy arbitrarily, then based on our numerical tests on the same 36 combinations of (α_1, α_2), for deadlines $t = 1, 2, \ldots, 10$, the corresponding evasion probabilities are 11%–18% suboptimal.

We further take the same problem instances used in Table 1 to compare these two policies. For $n = 2, 4, 8$, the policy that maximizes the detection probability by an arbitrary deadline produces an expected time to detection that is 1.38%, 1.34%, 1.17% suboptimal, respectively. On the other hand, the policy that minimizes the expected time to detection produces evasion probabilities that are typically 10%–20% suboptimal. Based on these observations, we believe that, in practice, the optimal policy that simultaneously maximizes the detection probability by an arbitrary deadline is more robust.

LIST OF REFERENCES

[1] D. Matula, "A periodic optimal search," *The American Mathematical Monthly*, vol. 71, no. 1, pp. 15–21, Jan. 1964.

[2] W. L. Black, "Discrete sequential search," *Information and control*, vol. 8, no. 2, pp. 159–162, 1965.

[3] M. C. Chew, "A sequential search procedure," *The Annals of Mathematical Statistics*, vol. 38, no. 2, pp. 494–502, 1967.

[4] J. B. Kadane, "Discrete search and the Neyman-Pearson Lemma," *Journal of Mathematical Analysis and Applications*, vol. 22, no. 1, pp. 156–171, Jan. 1968.

[5] M. C. Chew, Jr, "Optimal stopping in a discrete search problem," *Operations Research*, vol. 21, no. 3, pp. 741–747, 1973.

[6] J. B. Kadane, "Optimal whereabouts search," *Operations Research*, vol. 19, no. 4, pp. 894–904, 1971.

[7] M. Kress, K. Y. Lin, and R. Szechtman, "Optimal discrete search with imperfect specificity," *Mathematical Methods of Operations Research*, vol. 68, no. 3, pp. 539– 549, 2008.

[8] S. M. Ross, "A problem in optimal search and stop," *Operations Research*, vol. 17, no. 6, pp. 984–992, Dec. 1969.

[9] I. Wegener, "The discrete sequential search problem with nonrandom cost and overlook probabilities," *Mathematics of Operations Research*, vol. 5, no. 3, pp. 373–380, Aug. 1980.

[10] S. Alpern, R. Fokkink, L. Gasieniec, R. Lindelauf, and V. S. Subrahmanian, Eds., *Search Theory*. Springer, Apr. 2013.

[11] S. Alpern and S. Gal, *The Theory of Search Games and Rendezvous*. Kluwer Academic Publishers, 2003.

[12] L. Stone, *Theory of Optimal Search*. INFORMS, 2004.

[13] A. R. Washburn, *Search and Detection*, 4th ed. INFORMS, 2002.

[14] D. Roberts and J. Gittins, "The search for an intelligent evader: Strategies for searcher and evader in the two region problem," *Naval Research Logistics Quarterly*, vol. 25, no. 1, pp. 95–106, 1978.

[15] J. C. Gittins and D. M. Roberts, “The search for an intelligent evader concealed in one of an arbitrary number of regions,” *Naval Research Logistics Quarterly*, vol. 26, no. 4, pp. 651–666, 1979.

[16] E. Subelman, “A hide-search game,” *Journal of Applied Probability*, vol. 18, no. 3, pp. 628–640, Sep. 1981.

[17] A. R. Washburn, *Two-Person Zero-Sum Games*, 3rd ed. Rockville, MD: INFORMS, 2003.

[18] D. G. Luenberger, *Linear and Nonlinear Programming*. Kluwer Academic Publishers, 2003.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Research Sponsored Programs Office, Code 41
   Naval Postgraduate School
   Monterey, California

4. Richard Mastowski (Technical Editor) 1
   Graduate School of Operational and Information Sciences (GSOIS)
   Naval Postgraduate School
   Monterey, California

5. Dr. Kyle Y. Lin ..... electronic copy
   Operations Research Department
   Naval Postgraduate School
   Monterey, California

6. Dr. Dashi I. Singham ..... electronic copy
   Operations Research Department
   Naval Postgraduate School
   Monterey, California