ORIGINAL ARTICLE

Optimal discrete search with imperfect specificity

Moshe Kress · Kyle Y. Lin · Roberto Szechtman

Abstract   A target is hidden in one of several possible locations, and the objective is to find the target as fast as possible. One common measure of effectiveness for the search process is the expected time of the search. This type of search optimization problem has been addressed and solved in the literature for the case where the searcher has imperfect sensitivity (possible false negative results), but perfect specificity (no false positive detections). In this paper, which is motivated by recent military and homeland security search situations, we extend the results to the case where the search is subject to false positive detections.

Keywords   Discrete search · Imperfect specificity · uniformly optimal

1 Introduction

Discrete search problems have been out of vogue for over two decades. However, recent defense problems, such as searching for a hostage hidden in a city (e.g., relatively recent events in the Gaza strip) or detecting improvised explosive devices (IED) in Iraq, have underscored the need for efficient and effective search methods for detecting targets of various types.

We consider a surveillance system, the purpose of which is to find a target that is hidden in one out of n possible locations. The target location is uncertain but there is some prior information that is quantified in a prior probability distribution. The surveillance system comprises a sensor and a verification team. The sensor, which searches sequentially the n locations, is imperfect and therefore its cues are subject to

M. Kress (✉) · K. Y. Lin · R. Szechtman
Operations Research Department,
Naval Postgraduate School, Monterey,
CA 93943, USA
e-mail: mkress@nps.edu

errors. The verification team, which makes no errors, investigates positive detections by the imperfect sensor and verifies if they are true or false. Such a search process takes time, and the objective is to find a search policy that minimizes the expected search time until the target is found or optimizes some other measures of effectiveness (MOEs) such as the probability of detection.

Discrete search problems of the type mentioned above are not new. Optimal whereabout is studied in Ahlswede and Wegener (1987) and Kadane (1971). Chew (1967) considers an optimal search with stopping rule where all search outcomes are independent, conditional on the location of the searched object and the search policy. In another paper Chew (1973) considers a discrete search, where the target may not be present in one of the searched cells and the problem is when to stop the search. Similar problems are discussed in Pollock (1971). Wegener (1980) investigates a search process where the search time of a cell depends on the number of searches so far. A minimum cost search problem–similar to the one presented above–is discussed in Ross (1983), where only one search mode is considered and the sensor has perfect specificity. Other discrete search problems are studied in Black (2004), Matula (1964), Wegener (1980) and Song and Teneketzis (2004). Stone (1975) gives a comprehensive and detailed analysis of both maximum probability and minimum cost search models. All of the aforementioned references assume that the sensor has perfect specificity, that is, if it records a detection, the target is found. Our model relaxes this assumption and extends classical discrete search theory to incorporate false-positive errors, which are realistic phenomena in many defense and homeland security situations. While some authors (e.g., Danskin 1962) have considered the effect of false positive detections in the presence of decoys, to the best of our knowledge, our model is the first direct generalization of the classical discrete search problem. Incorporating imperfect specificity necessitates the introduction of an investigation stage following a detection.

The specific contributions of this paper are:

- We show that a greedy policy is optimal when each positive detection by the search sensor is followed by an investigation by the verification team.
- We derive the expected time of the search under the above conditions.
- If the verification time is significantly longer than the search time, an alternative MOE is the probability that the *first* detection is a true one. We show that a greedy policy is optimal for this MOE, too.
- Under certain realistic search conditions, we show that the greedy search rule is also uniformly optimal.

The paper is organized as follows. In Sect. 2, we introduce notation and formulate the problem. In Sect. 3, we prove the optimality of a certain greedy rule for the minimum-cost search problem. In Sect. 4, we examine two special cases of the model and show that a greedy rule is optimal also for probability oriented objectives. Sect. 5 gives a summary of the results and briefly discusses future research.

2 Operational motivation, statement of the problem and notation

The operational setting of our model can be demonstrated by the following scenario. A hostage is hidden in a place (e.g., house) located in one of n possible area cells
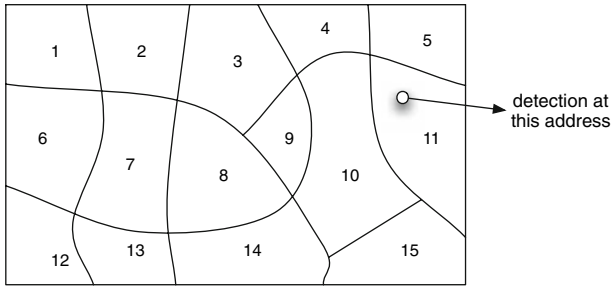
Fig. 1 A city is divided into 15 ACs, and a search in AC 11 yields a detection at a specific address

(AC) (e.g., city blocks). See Fig. 1. The objective is to locate the hostage as quickly as possible. An imperfect sensor searches the ACs one at a time. Following a detection, which identifies a place in the AC (e.g., an address of a house) where the hostage may be, a ground verification and rescue team is sent to that place to verify the detection and, if positive, rescue the hostage. There are three possible types of detection.

1. *Perfect Detection*: The sensor identifies correctly the place (address) where the hostage is kept.
2. *Partial Detection*: The sensor correctly identifies the AC where the hostage is held, but incorrectly identifies the specific place of captivity.
3. *False Detection*: The hostage is not hidden in the AC where the sensor has recorded a detection.

If the ground team is sent to a wrong address in a certain AC, it continues searching the rest of the AC and if the hostage is hidden there (the case of partial detection) he will be found by the team. If the hostage is not hidden in that AC (the case of false detection), the AC is declared to be cleared and therefore removed from further search by the sensor.

Let θ be the parameter that describes the AC where the target is hidden; that is, $\theta = i$ when AC i contains the target. Given the prior probability mass function (p.m.f.) of θ, $\pi : \{1, \ldots, n\} \to [0, 1]$, we write $\pi_i = P(\theta = i)$.

Recall that the sensor is imperfect. Let $p_i = P($ Sensor indicates *detection* in AC $i \mid \theta = i)$; that is, p_i is the probability that the sensor identifies correctly the AC where the hostage is hidden. Let $r_i = P($Sensor indicates detection and identifies correctly the place in AC $i \mid \theta = i)$. Clearly, $r_i \leq p_i$, and $p_i - r_i$ is the probability that the sensor identified the wrong place in the AC where the hostage is hidden. Finally, let $q_i = P($Sensor indicates detection in AC $i \mid \theta \neq i)$; $1 - q_i$ is the specificity of the sensor in AC i. We assume throughout that $p_i \geq q_i$ without loss of generality, because we can reverse the cue of the sensor if $q_i > p_i$.

Given a prior p.m.f. π, we select an action $a(\pi) \in \{1, \ldots, n\}$ that indicates the AC to be searched next. An action $a(\pi) \neq \theta$ results in one of two possible outcomes: a no detection or a false detection. Following either of these outcomes, posterior probabilities are obtained and the prior p.m.f. of θ is updated. In the case of a no detection,

the posterior p.m.f., $\Pi_a^-(\pi) = (\Pi_{a,1}^-, \ldots, \Pi_{a,n}^-)(\pi)$, is given by

$$
\Pi_{a,j}^-(\pi) = \begin{cases} \dfrac{(1-p_a)\pi_a}{1 - p_a\pi_a - q_a(1-\pi_a)}, & \text{if } j = a; \\[2ex] \dfrac{(1-q_a)\pi_j}{1 - p_a\pi_a - q_a(1-\pi_a)}, & \text{if } j \neq a. \end{cases} \tag{1}
$$

Considering the case of a false detection, the posterior p.m.f., denoted by $\Pi_a^+(\pi) = (\Pi_{a,1}^+, \ldots, \Pi_{a,n}^+)(\pi)$, is

$$
\Pi_{a,j}^+(\pi) = \begin{cases} 0 & \text{if } j = a; \\[2ex] \dfrac{\pi_j}{1-\pi_a} & \text{if } j \neq a. \end{cases} \tag{2}
$$

In either case, no detection or false detection, we update the prior π by $\Pi_a^-(\pi)$ and $\Pi_a^+(\pi)$, respectively. This way a sequence of priors is obtained until a true detection occurs.

The time it takes the sensor to search AC i is c_i. In case of perfect detection, the rescue team completes the rescue mission in $C_i^{(1)}$ time units. In case of partial detection, the length of the rescue mission is $C_i^{(2)} > C_i^{(1)}$. In case of false detection, the comprehensive verification time by the ground team in AC i before it is declared to be clear is $C_i^{(3)}$. The *objective function* of the searcher is to minimize the expected total time it takes to rescue the hostage.

To formulate the optimal search policy, first note that the total time it takes to rescue the hostage can be broken into two parts:

1. *Search Time*: The time spent to locate the AC in which the hostage is hidden, including the comprehensive verification time $C_i^{(3)}$ spent in a wrong AC following a false detection.
2. *Rescue Time*: The time it takes to find and rescue the hostage after locating the AC in which the hostage is hidden, either by perfect or partial detection.

Recall that θ denotes the AC that contains the hostage. Conditional on $\theta = i$, the rescue time takes on values $C_i^{(1)}$ or $C_i^{(2)}$ depending on whether the detection in AC i is perfect or partial. Therefore, the conditional expected rescue time is equal to

$$
\frac{r_i}{p_i} C_i^{(1)} + \frac{p_i - r_i}{p_i} C_i^{(2)}.
$$

Because at the beginning of the search, there is probability π_i that the hostage is hidden in AC i, the expected total rescue time is

$$
\sum_{i=1}^{n} \pi_i \left(\frac{r_i}{p_i} C_i^{(1)} + \frac{p_i - r_i}{p_i} C_i^{(2)} \right), \tag{3}
$$

which is a constant, invariant to the search policy. It follows that we can formulate an equivalent objective function, which is to *minimize the expected search time until*

correctly detecting the AC in which the hostage is hidden, either by a perfect or partial detection.

Note that in this equivalent objective function, the parameters r_i, $C_i^{(1)}$, and $C_i^{(2)}$, which only concern the rescue operation, but do not affect the search time, do not appear. In other words, the optimal policy is invariant with respect to these parameters. Throughout the rest of the paper, we use the equivalent modified objective function, and let $C_i = C_i^{(3)}$ to simplify the notation. In addition, hereafter we use the terms *search* and *look* interchangeably and the hostage is called *target*.

3 The optimal search policy

A search policy σ is a sequence of actions adapted to the sequence of priors; in our case each action depends only on the latest prior p.m.f. π. Let $T_\sigma(\pi)$ be the expected search time until detecting the correct AC, if the prior p.m.f. is π and the searcher follows search policy σ. Given two policies σ_1 and σ_2, we write $\sigma_1 \succeq_\pi \sigma_2$ when $T_{\sigma_1}(\pi) \leq T_{\sigma_2}(\pi)$.

The main result of this paper is presented in the next theorem.

Theorem 3.1 *Given a prior p.m.f. π for θ, the optimal search policy follows a greedy rule where the AC to search next is one having the maximal value of*

$$\frac{p_i \pi_i}{c_i + q_i C_i}, \quad i = 1, 2, \ldots, n.$$

We call the search policy in Theorem 3.1 the *greedy rule*, because each time we search in the AC that has the maximal ratio between the probability of finding the hostage and the expected (wasted) cost due to a false detection. Theorem 3.1 generalizes the case of perfect specificity, where the q_i's all equal zero; see, for example, (Ross 1983; Stone 1975).

To facilitate the proof of Theorem 3.1, we introduce two alternatives to express a feasible policy. First, let

$$\begin{pmatrix} a_1, \ a_2, \ a_3, \ a_4, \ \ldots \\ G, \ \ G, \ \ G, \ \ G, \ \ \ldots \end{pmatrix}$$

denote a feasible policy such that the searcher first follows the order $a_1, a_2, \ldots$, until the first detection takes place. If the first detection correctly locates the target (either perfect detection or partial detection), then the problem ends. If the first detection is a false detection, then the searcher switches to the greedy rule thereafter. Second, let

$$\begin{pmatrix} a_1, \ a_2, \ a_3, \ a_4, \ \ldots \\ b, \ \ G, \ \ G, \ \ G, \ \ \ldots \end{pmatrix}$$

denote a policy similar to the previous one, with the exception that if the first search in AC a_1 results in a false detection, then the searcher is required to search in AC b immediately before switching to the greedy rule.

Lemma 3.1 *Consider two policies*

$$\delta_1 = \begin{pmatrix} i, & j, & a_3, & a_4, & \dots \\ j, & G, & G, & G, & \dots \end{pmatrix}$$

and

$$\delta_2 = \begin{pmatrix} j, & i, & a_3, & a_4, & \dots \\ i, & G, & G, & G, & \dots \end{pmatrix}.$$

For any π, $\delta_1 \succeq_\pi \delta_2$, that is, the expected search time with policy δ_1 is shorter than with policy δ_1, if and only if

$$\frac{p_i \pi_i}{c_i + q_i C_i} \geq \frac{p_j \pi_j}{c_j + q_j C_j}.$$

Proof Let

$$\hat{\delta} = \begin{pmatrix} a_3, & a_4, & \dots \\ G, & G, & \dots \end{pmatrix}.$$

By conditioning on the location of the target, we can write

$$
\begin{aligned}
T_{\delta_1}(\pi) = c_i + \pi_i & \left[(1 - p_i) \left(c_j + q_j \left(C_j + T_G(\Pi_j^+ \Pi_i^-(\pi)) \right) \right) \right. \\
& \left. + (1 - q_j) T_{\hat{\delta}}(\Pi_j^- \Pi_i^-(\pi)) \right) \right] \\
& + \pi_j \left[q_i \left(C_i + c_j + (1 - p_j) T_G(\Pi_j^- \Pi_i^+(\pi)) \right) \right. \\
& \left. + (1 - q_i) \left(c_j + (1 - p_j) T_{\hat{\delta}}(\Pi_j^- \Pi_i^-(\pi)) \right) \right] \\
& + (1 - \pi_i - \pi_j) \left[q_i \left(C_i + c_j + q_j(C_j + T_G(\Pi_j^+ \Pi_i^+(\pi))) \right. \right. \\
& \left. + (1 - q_j) T_G(\Pi_j^- \Pi_i^+(\pi)) \right) \\
& \left. + (1 - q_i) \left(c_j + q_j(C_j + T_G(\Pi_j^+ \Pi_i^-(\pi))) + (1 - q_j) T_{\hat{\delta}}(\Pi_j^- \Pi_i^-(\pi)) \right) \right],
\end{aligned}
$$

where $T_G(\cdot)$ denotes the expected search time with the greedy rule. Interchanging the indices i and j we get an expression for $T_{\delta_2}(\pi)$. Because $\Pi_j^- \Pi_i^-(\pi) = \Pi_i^- \Pi_j^-(\pi)$, $\Pi_j^+ \Pi_i^-(\pi) = \Pi_i^- \Pi_j^+(\pi)$, and $\Pi_j^- \Pi_i^+(\pi) = \Pi_i^+ \Pi_j^-(\pi)$, taking the difference we have that

$$T_{\delta_1}(\pi) - T_{\delta_2}(\pi) = -\pi_i p_i(c_j + q_j C_j) + \pi_j p_j(c_i + q_i C_i).$$

From the last equation the result follows immediately. □

We next present the proof of Theorem 3.1.

Proof of Theorem 3.1 The proof is based on induction on the number of ACs. The theorem is trivially true for $n = 1$.

Suppose that the greedy rule is optimal if there are $n - 1$ or fewer ACs. Next we show that it is also optimal when there are n ACs. Without loss of generality let

$$\frac{p_1 \pi_1}{c_1 + q_1 C_1} = \max_{i=1,\dots,n} \frac{p_i \pi_i}{c_i + q_i C_i}. \tag{4}$$

We consider a class of search policies in which AC 1 is searched only following $\tau - 1$ no-detection searches elsewhere; that is, $a_i \neq 1$ for $i = 1, \dots, \tau - 1$ and $a_\tau = 1$. Let Δ_τ denote the set of these policies. We first deal with the case $\tau < \infty$. Let

$$\zeta_1 = \begin{pmatrix} 1, & a_1, & a_2, & \dots \\ G, & G, & G, & \dots \end{pmatrix},$$

$$\zeta_2 = \begin{pmatrix} a_1, & 1, & a_2, & \dots \\ G, & G, & G, & \dots \end{pmatrix},$$

$$\vdots$$

$$\zeta_\tau = \begin{pmatrix} a_1, & a_2, & \dots, & a_{\tau-1}, & 1, & a_{\tau+1}, & \dots \\ G, & G, & \dots, & G, & G, & G, & \dots \end{pmatrix}.$$

From Eq. (4), the induction hypothesis, and Lemma 3.1 we have

$$\zeta_1 \succeq_\pi \begin{pmatrix} 1, & a_1, & a_2, & \dots \\ a_1, & G, & G, & \dots \end{pmatrix} \succeq_\pi \zeta_2.$$

Hence, $\zeta_1 \succeq_\pi \zeta_2$. Repeating this argument we can see that $\zeta_1 \succeq_\pi \zeta_2 \succeq_\pi \cdots \succeq_\pi \zeta_\tau$. In particular, this implies that $\zeta_1 \succeq_\pi \zeta_\tau$. In other words, we show that for any policy in Δ_τ, with $\tau < \infty$, we can find a better policy that starts with AC 1.

Our previous argument shows that (T_{ζ_τ}) is a nondecreasing real sequence, so that $T_{\zeta_1} \leq T_{\zeta_\infty}$. Hence, for any policy in Δ_∞ the expected search time does not increase by starting the search on AC 1. Consequently, it is optimal to first search in AC 1. $\square$

To carry out the optimal policy in practice, we first use the following algorithm to generate the search order if all the searches thus far resulted in no detection.

1. Set $m = 1$.
2. Choose a such that

$$\frac{p_a \pi_a}{c_a + q_a C_a} = \max_j \frac{p_j \pi_j}{c_j + q_j C_j},$$

   and let $e_m = a$.

3.　Update π as follows:

$$\pi_a \leftarrow \frac{(1 - p_a)\pi_a}{1 - p_a\pi_a - q_a(1 - \pi_a)},$$
$$\pi_j \leftarrow \frac{(1 - q_a)\pi_j}{1 - p_a\pi_a - q_a(1 - \pi_a)}, \quad j \neq a.$$

4.　Let $m \leftarrow m + 1$, and go to 2.

Let $\mathbf{e} = \{e_m\}_{m=1}^{\infty}$ denote the search order generated by this algorithm. If following the optimal policy the first m searches all resulted in no detection, then it is optimal to next search AC e_{m+1}. Now suppose that the first $m - 1$ searches resulted in no detection, and the mth search in AC a ($e_m = a$) results in a false detection. To see how we can use $\mathbf{e}$ to find which AC to search next, note that the ratio of the posterior probabilities $\Pi_{a,j}^-(\pi)/\Pi_{a,k}^-(\pi)$, $j \neq a$ and $k \neq a$, remains unchanged, regardless of the search outcome in AC a. Therefore, according to Theorem 3.1, whether the mth search in AC a results in no detection or in a false detection, the relative positions for all the ACs other than a in $\mathbf{e}$ remain unchanged. Consequently, with the optimal policy, we simply continue to follow $\mathbf{e}$ by skipping those ACs that have gone through a comprehensive verification due to a false detection.

4 Special cases

In this section, we examine two special cases of the search model, which represent two extreme cases of the search scenario: one where the investigation process is risky and extremely long compared to the search time, and the other when the effect of the investigation time is negligible.

4.1 Risky and very long investigation process

Suppose that sending out ground units to investigate a detection is a risky and complex operation that may take a very long time compared to the search time, that is, $C_i \gg c_i$, $i = 1, \ldots, n$. In this case, the objective would be to minimize the chance of false positive detections, and consequently, the MOE would be the probability that the first detection is a true one. The optimal policy in this case is greedy, too, as shown in the following theorem.

Theorem 4.1 *Given a prior p.m.f. π for θ, the optimal search policy that maximizes the probability that the first detection is a true one, follows a greedy rule where the AC to search next is one having the maximal value of*

$$\frac{p_i \pi_i}{q_i}, \quad i = 1, 2, \ldots, n.$$

Proof Because the objective is to maximize the probability that the first detection is a true detection, a feasible policy is a sequence of ACs, such that the searcher follows

this sequence until a detection occurs. Consider two policies $\delta_1 = (i, j, a_3, a_4, \ldots)$ and $\delta_2 = (j, i, a_3, a_4, \ldots)$, and let $\hat{\delta} = (a_3, a_4, \ldots)$.

Let $V_\delta(\pi)$ denote the probability that the first detection is a true detection, if the prior p.m.f. is π and the searcher follows search policy δ. By conditioning on the location of the target, we can write

$$V_{\delta_1}(\pi) = \pi_i \left(p_i + (1 - p_i)(1 - q_j) V_{\hat{\delta}}(\Pi_j^- \Pi_i^-(\pi)) \right)$$
$$+ \pi_j \left((1 - q_i) p_j + (1 - q_i)(1 - p_j) V_{\hat{\delta}}(\Pi_j^- \Pi_i^-(\pi)) \right)$$
$$+ (1 - \pi_i - \pi_j)(1 - q_i)(1 - q_j) V_{\hat{\delta}}(\Pi_j^- \Pi_i^-(\pi)).$$

Interchanging the indices i and j we get an expression for $V_{\delta_2}(\pi)$. Because $\Pi_j^- \Pi_i^- (\pi) = \Pi_i^- \Pi_j^-(\pi)$, taking the difference we have that

$$V_{\delta_1}(\pi) - V_{\delta_2}(\pi) = \pi_i p_i q_j - \pi_j p_j q_i.$$

Therefore, $V_{\delta_1}(\pi) \geq V_{\delta_2}(\pi)$ if and only if $p_i \pi_i / q_i \geq p_j \pi_j / q_j$. The rest of the proof follows the steps as in Theorem 3.1 because we can always find a better policy than δ, if δ does not start with the AC that has the maximal value of $p_i \pi_i / q_i, i = 1, \ldots, n$. □

Note that the greedy rule of Theorem 4.1 is also a special case of Theorem 3.1 when $C_i = C, i = 1, \ldots, n$, and $C \to \infty$. In other words, if $C_i = C \gg c_i, i = 1, \ldots, n$, then the greedy rule of Theorem 4.1 also minimizes the expected time to detection.

4.2 Uniform search time, no effect of investigation time

Suppose that while the investigative sensor performs its investigation, the search sensor can proceed in searching other ACs. However, we assume that due to operational constraints and the high risk associated with the investigation operation, a verification team can be sent out to investigate an AC only upon a detection cue by the sensor. Given an infinite capacity of investigating resources, the problem is to find the search policy that minimizes the number of searches until the correct AC is detected by the search sensor (which will be immediately followed by an investigation that will confirm the detection). This scenario can be represented by our model by letting $c_i = 1$ and $C_i = 0$ for $i = 1, \ldots, n$.

In this situation, the greedy search policy given in Sect. 3 not only minimizes the expected number of looks until the target is detected, but it is also uniformly optimal. A discrete search policy is said to be *uniformly optimal* (see e.g., Stone 1975, p. 104) if it maximizes the probability of detecting the correct AC for any given number of available looks. Stone (1975) showed this result for the case when the specificity of the sensor is perfect, that is, $q_i = 0, i = 1, \ldots, n$. We extend this result to the case where $q_i \geq 0$.

Theorem 4.2 *If $c_i = 1$ and $C_i = 0$ for $i = 1, \ldots, n$, then the greedy rule in Theorem 3.1 is uniformly optimal.*

Proof Let $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$ denote the prior p.m.f. of the target's location, where $\pi_i = P(\theta = i), i = 1, \ldots, n$. Without loss of generality, suppose

$$\pi_1 p_1 = \max_{j=1,\ldots,n} \pi_j p_j. \tag{5}$$

First note that the theorem is trivially true if the searcher is allowed only one look.

Suppose there are $k \geq 2$ looks available and consider two searchers A and B with the same prior p.m.f. $\pi = (\pi_1, \ldots, \pi_n)$. Recall that A's policy can be represented by a sequence of actions $\sigma = (a_1, a_2, \ldots, a_k)$, where $a_i(\cdot)$ maps the updated p.m.f. of the target's location to the AC for A's ith look. Suppose A does not start in AC 1; that is, $a_1(\pi) \neq 1$. To prove the theorem, we will show that B can do at least as well as A by first searching in AC 1. The theorem then follows due to Eq. (5).

To do so, consider the following policy for B: First search in AC 1. If the search results in a true detection, then the search ends; otherwise, instead of updating the p.m.f. of the target's location, let B keep the original prior p.m.f. π. Starting from the second look, continue the search with $a_1, a_2, \ldots$, and update the p.m.f. of the target's location according to Eqs. (1) and (2) along the way. The search under $a_1, a_2, \ldots$ continues until B is instructed by $a_1, a_2, \ldots$ to search in AC 1 for the first time (besides the very first look in AC 1). At that point, do not search in AC 1; instead, update the p.m.f. of the target's location according to the outcome from the very first search in AC 1. Say a_m instructs B to search in AC 1 for the first time, then starting in the $(m+1)$st look let B follow $a_{m+1}, a_{m+2}, \ldots, a_k$ throughout the rest of the search.

In order to show that the probability of B finding the target in k looks is no less than that of A, we couple the location of the target θ and the search outcomes for the two searchers, such that A's lth look in AC i yields the same outcome as B's lth look in AC i, for $l = 1, 2, \ldots$, and $i = 1, \ldots, n$. By doing so, we can see that with probability $\pi_1 p_1$, B finds the target in his first look. If B continues after the first look, then in each sample path, A's ith look and its outcome will be identical to B's $(i+1)$st look and its outcome, $i = 1, 2, \ldots$, as long as A has not yet searched in AC 1. When A searches in AC 1 for the first time, either A finds the target (in which case B finds the target in his very first search because of stochastic coupling), or thereafter both searchers will look at the same ACs throughout the search process.

To compare the probability that each searcher can find the target within k looks, we consider two cases:

1. *Searcher A searches in AC 1 on or before his kth look*: In this case, B finds the target within k looks if and only if A does, so the probability of finding the target within k looks is identical for both searchers.
2. *Searcher A never searches in AC 1 during his first k looks*: In this case, with probability $\pi_1 p_1$ B finds the target, and A does not. The only situation where A finds the target within k looks and B does not is if A finds it on his kth look in, say, AC i. Because AC i may have been searched a few times during A's first $k-1$ looks, the probability that A finds the target on the kth look is bounded by $\pi_i p_i$. Therefore, the probability that A finds the target, but not B, is bounded by

$$\max_{i=2,\ldots,n} \pi_i p_i \leq \pi_1 p_1,$$

where the inequality follows from Eq. (5). Therefore, the probability of B finding the target within k looks is at least as large as that of A.

The preceding discussion shows that for $k \geq 1$, there exists a feasible policy that starts with AC 1 and maximizes the probability that the target will be found within k looks. Hence, the greedy rule is uniformly optimal. □

5 Summary and conclusions

In this note we extend previous results concerning discrete searches to the case where the searcher has imperfect specificity. In that case, the imperfect searcher is coupled with a perfect, time-consuming, investigating agent that verifies detection cues by the searcher. A simple greedy rule is developed, which is proven to be optimal when the objective is to minimize the expected time to detection. The expected search time of this greedy search is calculated and some numerical analysis is provided. For certain situations, it is shown that the greedy rule maximizes a probability objective and is uniformly optimal. Note, however, that we assume, as in previous works on this topic, that the time of a transition from one AC to another is zero. In many situations (e.g., unmanned aerial vehicle searching a road for IEDs) this may not be the case and travel time must be accounted for explicitly. Incorporating travel time in this search model leads to more complex dynamic programming settings that will be explored in future research.

References

Ahlswede R, Wegener I (1987) Search problems. Wiley, New York
Black WL (2004) Discrete sequential search. Inf Control 8:159–162
Chew MC (1967) A sequential search procedure. Ann Math Stat 38:494–502
Chew MC (1973) Optimal stopping in a discrete search problem. Oper Res 21(3):741–747
Danskin JM (1962) A theory of reconnaissance: I, ii. Oper Res 10(3):285–309
Kadane JB (1971) Optimal whereabout search. Oper Res 19:894–904
Matula D (1964) A periodic optimal search. Amer Math Mon 71:15–21
Pollock SM (1971) Serch detection and subsequent action: some problems on the interfaces. Oper Res 19:559–586
Ross SM (1983) Introduction to stochastic dynamic programming. Academic Press, Amsterdam
Song N-O, Teneketzis D (2004) Discrete search with multiple sensors. Math Meth Oper Res 60:1–13
Stone LD (1975) Theory of optimal search. Academic Press, Amsterdam
Wegener I (1980) The discrete sequential search problem with nonrandom cost and overlook probabilities. Math Oper Res 5:373–380
Wegener I (1980) Optimal search with positive switch cost is np-hard. Inf Process Lett 21:49–52