American Journal of Operations Research, 2016, 6, 153-166
Published Online March 2016 in SciRes. http://www.scirp.org/journal/ajor
http://dx.doi.org/10.4236/ajor.2016.62018

Optimal Search for Hidden Targets by Unmanned Aerial Vehicles under Imperfect Inspections

Boris Kriheli[1], Eugene Levner[1,2], Alexander Spivak[2]

[1]Department of Logistics, Ashkelon Academic College, Ashkelon, Israel
[2]Department of Computer Science, Holon Institute of Technology, Holon, Israel
Email: levner@hit.ac.il, borisk@hit.ac.il, spivak@hit.ac.il

Received 20 January 2016; accepted 7 March 2016; published 10 March 2016

Abstract

Assume that a target is hidden or lost in one of several possible locations and is to be found by the unmanned aerial vehicle (UAV). A target can be either a hostile object or missing personnel in remote areas. Prior probabilities of target locations are known. Inspection operations done by the UAVs are imperfect, namely, probabilities of overlooking the hidden target and probabilities of false alarms exist for any possible location. The UAV has to sequentially inspect the locations so that to find the target with the minimum loss or damage incurred by the target before it is detected subject to a required level of confidence of target identification. A fast (polynomial-time) priority-based algorithm for finding an optimal search strategy is developed.

Keywords

Search and Detection, Discrete Search, Imperfect Inspection, Greedy Algorithm

1. Introduction

The need for searching for hidden or lost objects arises in many applications. Common civilian applications are policing, control of air/water/soil pollution, fire fighting, prevention of natural disaster like floods, and others. Common military applications are the battlefield surveillance, detection of enemy intrusion or hidden hostile targets, rescuing lost people or objects in the open sea, and many others. We refer the interested reader to Stone [1] and Benkoski *et al*. [2] for further examples.

Suppose a target is hidden in some region. Until being found, it may cause loss or damage to the environment, the damage scale being dependent on the location of the target and the time needed to detect and neutralize it.

How to cite this paper: Kriheli, B., Levner, E. and Spivak, A. (2016) Optimal Search for Hidden Targets by Unmanned Aerial Vehicles under Imperfect Inspections. *American Journal of Operations Research*, **6**, 153-166.
http://dx.doi.org/10.4236/ajor.2016.62018

The problem is to efficiently detect and remove or neutralize the target with the help of unmanned aerial vehicles (UAV). The UAV (also called *drone*) is an aircraft without a human pilot on board. It is controlled either autonomously by on-board computers or by an operator (or a group of operators) on the ground or in another aircraft. The UAVs are preferred for missions that are dangerous, complicated or just impossible for manned aircraft. Numerous examples of advanced practical applications of UAVs can be found in Valavanis [3].

A typical example arising in military logistics is a discrete-time search optimization problem in which a single automatic searcher (a military robot or an UAV) moves through a discretized 3-dimensional airspace and needs to find a static target located/hidden in a finite set of possible geographical locations. The searching automated device is subject to search-time constraints, fuel consumption and other factors. The typical objective of the search is to maximize the probability of detection or minimize the cost (or time) of the search (see, e.g., Dell *et al*. [4], Kress *et al*. [5], Sato and Royset [6], Kress *et al*. [7]). The resulting optimization problems are quite challenging from the computational point of view. General resource-and time-constrained min-cost and max-probability problem versions are proved to be NP-hard (Wegener [8]; Trummel and Weisinger [9]). In the present paper, we consider a different setting, namely, we intend to minimize the incurred loss or damage. Such problem arises in military search, surveillance, and reconnaissance operations with patrol aircraft and UAVs, where physical, logistic and operational constraints limit the search time.

The choice of a search strategy strongly influences the search costs and losses incurred by the search as well as the probability of finding the target within a specific time limit. In military search-and-detection missions of the UAVs, the problem of finding good strategies has its own specificity because, unlike manned aircraft, these mobile agents need to operate in autonomous regime, without human advice and involvement. Under hostile or sabotage circumstances, like radio interference, two types of errors in search-and-detection operations usually occur: 1) the so-called “false-negative detection error”, wherein a target object is overlooked; and 2) the “false-positive detection error”, also called “a false alarm”, in which a decoy or a false target is wrongly classified as a correct target. Unfortunately, the problem of selecting the best search strategy is fundamentally hard due to its probabilistic nature, combinatorial structure and nonlinearity induced by the detection probability. In particular, looking twice into the same location by a searcher generally does not double the detection probability.

There exist hundreds of reviews, papers and texts providing a deep background and presenting wide applications of the discrete search problems (see, e.g., Benkoski *et al*. [2], Washburn [10], Song and Teneketzis [11], Kress *et al*. [5], Sato and Royset [6], Wilson *et al*. [12], Chung and Burdick [13], Kriheli and Levner [14], Kriheli *et al*. [15] [16], and the numerous citations in those works).

In the present work, we study a search-and-detection process subject to the false-negative and false-positive inspection outcomes. The general resource-constrained problem being NP-hard, we restrict our study to finding optimal strategies in a certain special scenario described below. The search is to minimize the losses incurred during the search. We organize the search process using a greedy strategy in which, at each step, the on-board computer used by the UAV computes a current *search effectiveness* for each location and sequentially searches for a next location with the highest current search effectiveness.

Complementary to the discrete search model developed by Chung and Burdick [13], in which the search objectives are to maximize the probability of detecting the targets or to minimize the search time, in the present study we consider the objective function of the min-loss type; we investigate the different search scenario and stopping rule, develop a new greedy index-based search strategy and, finally, prove its properties. In contrast to the works by Kress *et al*. [5] and Chung and Burdick [13], our model does not require human interference. Specifically, the decision to stop the search is generated automatically by the on-board computer program basing on the input data, sensors’ assessments and the search history.

In the scenario studied in the present paper, the local-optimal (greedy) strategy yields a global optimal. Being attractive due to its simplicity and computational efficiency, such local-search strategy guarantees finding an optimal search sequence with a pre-specified confidence level of target identification. Several earlier known discrete search models, in particular, the search model with only false-negative outcomes (Kadane [17]; Levner [18]) and the search model for perfect inspections (Rabinowitz and Emmons [19]) are special cases of the model developed in the present paper.

The remainder of the paper is organized as follow. The next section overviews the related work. Section 3 introduces the discrete search problem in more detail. Section 4 analyses the problem and introduces its solution algorithm. Section 5 presents numerical examples and computational results. Section 6 concludes the paper.

2. Related Work

The discrete search problem is one of the oldest and most popular problems of Operational Research. Its pioneering study was made by Bernard Koopman and his team during the World War II to provide efficient methods for detecting submarines (Koopman [20]). The search theory has been used by the US Navy in the search for the H-bomb lost in the ocean near Palomares, Spain, in 1966. Adetailed historical survey of search theory and the bibliography of the discrete search literature can be found in Stone [1], Benkoski *et al*. [2] and Washburn [10]. In recent decades, new models and methods for different modifications of the discrete search problems have been introduced. A wide range of general optimization methods are being used for solving basic search problems, among them the Lagrange multipliers, convex programming, dynamic programming, and branch-and-bound enumeration. The extensive bibliography can be found in the excellent survey by Benkoski *et al*. [2]. In the present paper, we complement the latter survey. Notice, however, that our survey is much more selective, being restricted to the overview of the discrete search for a stationary target under imperfect inspections. The review deals with the results related to the efficient local (greedy) solution methods, with an emphasis on those obtained in recent two decades. In such algorithms, the search effectiveness (also called the location attractiveness) is defined, step by step, for each location. The greedy (index-based) algorithm proceeds by searching, at each next step, the location with the highest search effectiveness. Typically it is the ratio of the detection probability to the corresponding cost, but, as we shall show, it may be of a more complicated structure.

For the sake of completeness, we start with a brief overview of related classic results obtained in the 1970s. Sweat [21] suggested an efficient optimization strategy for maximizing the expected total discounted reward. Kadane [17] and Stone [1] considered a more general scenario in which the times and probabilities of fault detections can change during the search. Chew [22], Wegener [8], Assaf and Zamir [23], and Levner [18] considered varying costs of inspecting different locations. Chew [22] and Kadane [17] studied the problem of maximizing the detection probability subject to cost constraints and showed that a locally-optimal strategy is globally optimal when the cost is proportional to the number of searches. Kadane and Simon [24] proposed a general approach to the min-cost and max-reward (finite-horizon) search problems. Notice that in the above flow of early works, the assumption has been imposed that only the false-negative detection tests occur in the system, the false-positive outcomes (the false-alarms) being totally ignored.

Trummel and Weisinger [9] and Wegener [8] have showed that the general min-cost, min-time, max-probability-of-detection and min-cost-with-switches search problems are NP-hard.

When the location inspections are imperfect, such a problem type may need an infinite sequence of steps as its solution. Thus, the problem is to find *an optimal strategy* rather than an optimal sequence itself. The latter sequence depends upon a random instant when the searcher finds the target. A typical stopping rule for the random search process is the following: "*Stop at the step when the target is found for the first time*". Obviously, this rule, being applicable for the case of no-positive-false tests is not valid if the positive-false error occurs. Other criteria for the termination of the search process are treated by Chew [22] and Ross [25]; however, their termination conditions correctly capture only the false-negative search operations, that is, rely on the assumption that false-positive detection probabilities are zeros. More detailed analysis of the early works can be found in the excellent texts by Ahlswede and Wegener [26], Stone [1], Chudnovsky and Chudnovsky [27], and Benkoski *et al*. [2].

In recent decades, the operational research community has made great strides in the problem of sequential search under discrete time and space. Research results in Chung and Burdick [13] and the references therein presented a Bayesian construction for the problem of searching for multiple targets by imperfect autonomous devices such as robots, wherein the probability of detection of these targets is an objective for the optimal search trajectory. Kress *et al*. [5] and Chung and Burdick [13] are the first researchers known to us who address the possibility of false-positive detections. They analyzed several myopic and biology-inspired search strategies minimizing the search time and have shown that a local index-based policy is optimal when each detection by an autonomous search device is followed by a reliable work by a human verification team.

In the present work, we continue and extend the results mentioned above, focusing on studying the search scenario in which the imperfect inspections can be either false-positive, or false-negative, or of the both types. A main contribution of our work is that, complementary to the works of Kress *et al*. [5] and Chung and Burdick [13], we study the practically important scenario where the interference of a human verification team is extremely expensive or practically impossible, in which case the UAV's on-board computer should decide whether to stop or not to stop the search basing on the search history and the sensors' assessment, without the interference of the human team.

3. Problem Formulation

We define *the sequential discrete search problem* as follows. A discrete search area of interest (AOI) is a complex geographical area which contains a set of N possible locations of a stationary target. The objective of the search is to detect and neutralize a hidden or lost hostile target. A target is to be found by a single searcher. Given the stationary nature of the target, it is assumed to be present throughout the entire duration of the search process, *i.e.*, the target does not leave the search area and new targets do not appear. A target location in the search area is uncertain. In other words, an inspection of each location can be imperfect. This means that there given prior probabilities α_i of the false alarm (if the UAV's sensors falsely announce that the target is found when the location i is clean, and, besides, there given prior probabilities β_i of overlooking the target-when actually location i contains a target, $i = 1, \cdots, N$. The imperfect quality of inspections implies that examination of each location can happen more than once. Hence, the search sequence, in general, may be infinite. After the search starts, the single mobile searcher should perform a set of sequential inspections in order to identify the target. The durations and costs of the inspections of all the locations being given, the goal of the search is to determine a search strategy that the searcher should employ in order to locate the target with the minimum cost or maximum detection probability.

A formal description of the problem is as follows. A system contains N modules $1, 2, \cdots, N$. The input data are the following: Each location i, $i = 1, \cdots, N$, is characterized by the following known parameters:

- p_i—prior probability that location i contains a target;
- α_i—prior probability of a "false alarm", or a false-positive outcome; this is a conditional probability that sensors of the autonomous UAV, after inspecting location i, declare that the target is discovered in location i whereas, in fact, it is *not* in this location;
- β_i—prior probability of overlooking, or a false-negative outcome; this is a conditional probability that the sensor of the UAV declares that target is not in location i whereas, in fact, it is exactly there;
- t_i—expected time to inspect location i;
- c_i—search cost rate per unit time when searching location i; this is the amount (in monetary or physical units) of loss or damage incurred during one time unit of search;
- CL_0—a pre-specified confidence level, this is a measure of confidence that the target is identified correctly; this value is close to 1 and will be exactly defined below.

Each sequential inspection strategy is specified by an infinite sequence of steps:

$$\boldsymbol{s} = \left\langle S[0], s[1], s[2], \cdots, s[n], \cdots \right\rangle$$

where $s[n]$ denotes the *location number* (further called *the element number*), more exactly, the number of the location which is inspected at the n-th step of sequence s, all $s[n] \in \{1, 2, \cdots, N\}$, and $S[0]$ is an initial sub-sequence with which strategy $\boldsymbol{s}$ starts, which will be defined below.

In what follows, the terms *strategy* and *sequence* are used interchangeably.

We suppose that the UAV makes its search moving in a certain sequence of steps $\boldsymbol{s}$. In order to exactly define the stopping rule of the search process, we first introduce a notion of the confidence level CL. Without loss of generality, we suppose that the on-board computer, at each step of $\boldsymbol{s}$, say i, may count and memorize the search history, that is, how many times, denoted h_i, the sensor has already observed that the location i contains the target. (Recall that, due to the presence of the false positive outcomes, we are not obliged to immediately stop searching in the case when the sensor claims for the first time that it discovered the target).

Notice that in the scenario studied by Kress *et al*. [5] and Chung and Burdick [13], as soon as the automated device (UAV) detects a target location (for brevity's sake, we shall call such a test *positive*), a ground human team is sent to this location to verify the detection. Thus, the human team is involved to verify the search quality of the imperfect device. In contrast, we consider a search scenario for a *completely autonomous* aircraft. In order to enhance automatically the search quality level, the on-board computer analyses the input data and the overall search history, and then computes the conditional probability $p_{[i,h]}$ that the examined location i indeed contains the target under the condition that, during the previous sequential inspections, the (imperfect) UAV's sensors have already declared h times that the location i contains the target. Naturally, the conditional probability $p_{[i,h]}$ increases with h growing, and we require that the $p_{[i,h]}$ should be sufficiently high. More concretely, the $p_{[i,h]}$

value should not be less than a pre-specified value that we called the *confidence level* and denoted by *CL* (for example, one can take *CL* = 0.95 or 0.99). Further we show that the on-board computer can me programmed to compute how many times, during their sequential inspections, the UAV's sensors has to observe that the cell *i* does contain the target, in order *to guarantee* that $p_{[i,h]} \geq CL$. A specific feature of this scenario is that the computer memorizes the entire history of the sensors' observations during the UAV's search.

The minimum positive integer *h*, such that the probability $a_{[i,h]}$ exceeds or equals the confidence level *CL* is called the *critical height* H_i; this value will be precisely determined below from the non-linear Equation (1) below depending on the values of the input probabilities. Notice that all the values H_i, for each cell *i*, can be computed and memorized before the search process starts. In this scenario, the basic idea is that, instead of using the human involvement at each positive test, like in the Kress's *et al*. [5] scenario, the fully autonomous search stops when for some cell *i*, the outcome of a current inspection will be that, for the H_i-th time, cell *i* is discovered to contain the target. It is clear that in this case the probability of the correct decision will be guaranteed to be sufficiently high, without the human-made additional testing. After arriving to such a case, the UAV carrying an explosive payload may hit a hostile target, or a human involvement, like sending a rescue team, is performed.

The number accumulated up to a certain step *n*, counting that last step, and associated with the location searched at step *n* in ***s*** is called the (current) *height* of the cell $s[n] = i$ and denoted by h_i. As explained above, the conditional probability $p_{[i,h]}$ is defined as "the probability of that the target is correctly identified under condition that the sensors have repeatedly declared a fixed number of times that the target is indeed in location *i*". Clearly, the $p_{[i,h]}$ values can be different for different location *i*, depending upon given probabilities p_i, α_i and β_i. The precise meaning of the term *repeatedly declared a fixed number of times* will be explained below in detail.

Let us summarize the said above. Without loss of generality, we suppose that the on-board computer of the UAV, in each step of the search sequence ***s***, say *n*, counts and memorizes the entire search history, that is, how many times the UAV's sensors had claimed, before the current step *n*, that different locations (indexed $s[n] = i$) contain the target. Such a number, accumulated up to step *n* and associated with any location searched at that step in ***s*** is called the *height* of the location $s[n] = i$ at step *n* and denoted by $h_i = h_i(n)$. For brevity, in the following we shall omit the index *n* in $h_i(n)$ if this does not lead to confusion. Notice that, due to existence of the false alarms the search should not stop when at a step *n* the sensors declare for the first time that they discovered the target. We need to continue the search until the probability $a_{[i,h]}$ reaches the pre-specified value *CL*.

Now we can give the exact definition: for a given *n* value, the *confidence level* CL_i at step *n* is defined as the desired lower bound for the conditional probability $p_{[i,h_i]}$ that location $s[n] = i$ is indeed contains the target under condition that the sensors have totally declared h_i times (that is, in h_i previous steps) that the target is located in location *i*. When the $h_i(n)$ value at each step *n* of ***s*** is known, there is a direct way to compute the corresponding probability $p_{[i,h_i]}$ for any h_i, as the following claim states.

Claim 1. (a) The conditional probability $p_{[i,h_i]}$ relates to the height $h_i = h_i(n)$ as follows:

$$p_{[i,h_i]} = \frac{p_i \cdot (1-\beta_i)^{h_i}}{p_i \cdot (1-\beta_i)^{h_i} + (1-p_i)\alpha_i^{h_i}} \tag{1}$$

(b) Probabilities $p_{[i,h_i]}$ increase with the growth of h_i.

The proof of Claim 1 is given in the next section.

Given the input probabilities, the optimal search scenario is specified by the following conditions:

1) the elements (locations) are inspected sequentially and independently of each other;

2) for any search strategy and any target location, the outcomes of inspections are independent;

3) the *stopping rule* is defined as follows: Given a required (permitted) value of the confidence level CL_0, say $CL_0 = 0.95$, the search process stops when, at some step *n* for some location $s[n] = i$ and the corresponding height h_i, the requested confidence level of identifying the target is achieved, that is:

$$p_{[i,h_i]} \geq CL_0 . \tag{2}$$

Claim 2. If, for any location *i*, the probability $1 - \beta_i$ of "no-overlooking" exceeds the false-alarm probability α_i (that is, $1 - \beta_i \geq \alpha_i$), then the probabilities $p_{[i,h_i]}$ increase with the growth of *n*, for any pair of constant values (α_i, β_i).

The proof follows immediately from two following facts: (a) the height h_i of any location *i* increases with the growth of *n* in any search strategy ***s***, and, (2) the function

$$p_{[i,h_i]} = p_{[i,h_i]}(h_i) = \frac{p_i \cdot (1-\beta_i)^{h_i}}{p_i \cdot (1-\beta_i)^{h_i} + (1-p_i)\alpha_i^{h_i}}$$
$$= \frac{1}{1+\left[(1-p_i)/p_i\right]\left[\alpha_i/(1-\beta_i)\right]^{h_i}}$$

is obviously increasing in h_i, whenever $\alpha_i/(1-\beta_i) \geq 1$. (Index n is omitted in $h_i = h_i(n)$ and $p_{[i,h_i]} = p_{[i,h_i]}(n)$ in the above formulas for the sake of simplicity).

Corollary. Given the probabilities p_i, α_i and β_i, for all i, and the required confidence level CL_0, one can solve the inequality (2) with respect to h_i. By virtue of Claim 2, the minimum integer h_i that satisfies (2) presents the minimum number of positive tests of location i guaranteeing the required confidence level of the search. We denote this minimum h_i-value by H_i and call it a *critical height* of location i.

The CL_0 value being given in advance, all the H_i values can be computed from (2), for all locations, before the UAV starts its search process. Now we are able to formulate the stopping rule.

The stopping rule. The search process stops at the step at which the number of positive tests provided by the sensors for some location i^* reaches, for the first time, its critical height value, H_{i^*}. Thus, our model does not require human operator interference at each positive test for defining the stop moment; indeed, the search is stopped automatically by the on-board computer program as far as the H_i values are computed from (2) for all i and for the given CL_0 values, and the search history is known.

4. Problem Analysis

In the following analysis, for a given sequence $\boldsymbol{s}$, we need the following notation:

- $T_{s[n]} = T(s[n], s)$-(accumulated) time spent by the UAV when moving from s [1] to location $s[n]$ on the n-th step of strategy s; $T_{s[n]} = \sum_{m=1}^{n} t_{s[m]}$,
- $P_{s[n]}$—unconditional probability that the sensor has totally declared in $H_{s[n]}$ steps that the location $s[n]$ contains the target, up to the n-th step of strategy $\boldsymbol{s}$. (The values $H_{s[n]}$ and $P_{s[n]}$ depending on parameters α_i and β_i, and guaranteeing a required confidence level CL_0 will be computed below).
- $F(s)$—the expected total loss incurred by the target before the latter is found and neutralized.

In accordance with the above conditions (i) and (ii) of the considered search scenario, the expected (linear) total loss, $F(\boldsymbol{s})$, is defined as follows:

$$F(\boldsymbol{s}) = \sum_{n=1}^{\infty} P_{s[n]} c_{s[n]} T_{s[n]}.$$

In this notation, the stochastic infinite-horizon search problem is to find an optimal sequence $\boldsymbol{s}^*$ minimizing the expected total loss $F(\boldsymbol{s})$ incurred by the target before the search is stopped.

We add the following notation:

Event B_i = {after a single inspection, the sensor declares that location i contains the target}.

Event C_i = {Location i is really contains the target}.

In terms of the events, the location probabilities p_i and probabilities of the errors of two types are expressed as follows:

$$p_i = P(C_i); \quad \alpha_i = P(B_i / \bar{C}_i); \quad \beta_i = P(\bar{B}_i / C_i).$$

The probability f_i that the sensor declares that location i is detected as containing the target is:

$$f_i = P(B_i) = P(\text{sensor declares that location } i \text{ contains the target})$$
$$= P(C_i)P(B_i / C_i) + P(\bar{C}_i)P(B_i / \bar{C}_i) = p_i \cdot (1-\beta_i) + (1-p_i)\alpha_i$$

The conditional probability of the event that the location i contains the target under condition that the sensor has declared that the location i contains the target in a single inspection, is computed as follows:

$$P(C_i / B_i) = \frac{P(C_i)P(B_i / C_i)}{P(C_i)P(B_i / C_i) + P(\bar{C}_i)P(B_i / \bar{C}_i)} = \frac{p_i \cdot (1-\beta_i)}{p_i \cdot (1-\beta_i) + (1-p_i)\alpha_i}.$$

Let us return to Claim 1 from Section 3. Recall that we need to prove that the conditional probability $p_{[i,h_i]}$ of the event that the location *i* indeed contains the target under condition that the sensor has declared in exactly h_i steps of the sequence ***s*** that the location *I* contains the target, satisfies the following relation:

$$p_{[i,h_i]} = P\left(C_i / B_i^{(1)} \cap B_i^{(2)} \cap \cdots \cap B_i^{(h_i)}\right) = \frac{p_i \cdot (1-\beta_i)^{h_i}}{p_i \cdot (1-\beta_i)^{h_i} + (1-p_i)\alpha_i^{h_i}}$$

Proof. Since the sequential inspections of locations made by the UAV are independent, we have that for any pair of indices $i, j \in \{1, 2, \cdots, N\}$ the following equalities hold:

$$\begin{cases} P\left(B_i^{(k+1)} / C_i \cap B_j^{(k)}\right) = P\left(B_i^{(k+1)} / C_i\right) \\ \text{and} \\ P\left(B_i^{(k+1)} / \overline{C}_i \cap B_j^{(k)}\right) = P\left(B_i^{(k+1)} / \overline{C}_i\right) \end{cases}$$

Then

$$\begin{aligned} & P\left(C_i / B_i^{(1)} \cap B_i^{(2)} \cap \cdots \cap B_i^{(h_i)}\right) \\ & = \frac{P(C_i) \cdot P\left(B_i^{(1)} \cap B_i^{(2)} \cap \cdots \cap B_i^{(h_i)} / C_i\right)}{P(C_i) \cdot P\left(B_i^{(1)} \cap B_i^{(2)} \cap \cdots \cap B_i^{(h_i)} / C_i\right) + P(\overline{C}_i) \cdot P\left(B_i^{(1)} \cap B_i^{(2)} \cap \cdots \cap B_i^{(h_i)} / \overline{C}_i\right)} \end{aligned}$$

Next, using the relation

$$P(A \cap B / C) = P(A / C) P(B / A \cap C),$$

we obtain:

$$\begin{aligned} & P\left(B_i^{(1)} \cap B_i^{(2)} \cap \cdots \cap B_i^{(h_i)} / C_i\right) \\ & = P\left(B_i^{(1)} / C_i\right) P\left(B_i^{(2)} \cap \cdots \cap B_i^{(h_i)} / C_i \cap B_i^{(1)}\right) \\ & = P\left(B_i^{(1)} / C_i\right) P\left(B_i^{(2)} \cap B_i^{(3)} \cdots \cap B_i^{(h_i)} / C_i\right) \\ & = P\left(B_i^{(1)} / C_i\right) P\left(B_i^{(2)} / C_i\right) P\left(B_i^{(3)} \cap \cdots \cap B_i^{(h_i)} / C_i \cap B_i^{(2)}\right) \\ & = P\left(B_i^{(1)} / C_i\right) P\left(B_i^{(2)} / C_i\right) P\left(B_i^{(3)} \cap \cdots \cap B_i^{(h_i)} / C_i\right) = \cdots \\ & = P\left(B_i^{(1)} / C_i\right) P\left(B_i^{(2)} / C_i\right) \cdots P\left(B_i^{(h_i)} / C_i\right) \\ & = P^{h_i}\left(B_i^{(1)} / C_i\right) = (1-\beta_i)^{h_i}. \end{aligned}$$

Similarly,

$$\begin{aligned} & P\left(B_i^{(1)} \cap B_i^{(2)} \cap \ldots \cap B_i^{(h_i)} / \overline{C}_i\right) \\ & = P\left(B_i^{(1)} / \overline{C}_i\right) P\left(B_i^{(2)} \cap \cdots \cap B_i^{(h_i)} / \overline{C}_i \cap B_i^{(1)}\right) \\ & = P\left(B_i^{(1)} / \overline{C}_i\right) P\left(B_i^{(2)} \cap B_i^{(3)} \cdots \cap B_i^{(h_i)} / \overline{C}_i\right) \\ & = P\left(B_i^{(1)} / \overline{C}_i\right) P\left(B_i^{(2)} / \overline{C}_i\right) P\left(B_i^{(3)} \cap \cdots \cap B_i^{(h_i)} / \overline{C}_i \cap B_i^{(2)}\right) \\ & = P\left(B_i^{(1)} / \overline{C}_i\right) P\left(B_i^{(2)} / \overline{C}_i\right) P\left(B_i^{(3)} \cap \cdots \cap B_i^{(h_i)} / \overline{C}_i\right) = \cdots \\ & = P\left(B_i^{(1)} / \overline{C}_i\right) P\left(B_i^{(2)} / \overline{C}_i\right) \cdots P\left(B_i^{(h_i)} / \overline{C}_i\right) \\ & = P^{h_i}\left(B_i^{(1)} / \overline{C}_i\right) = \alpha_i^{h_i}. \end{aligned}$$

Therefore,

$$p_{[i,h_i]} = \frac{P(C_i)\cdot P\left(B_i^{(1)}\cap B_i^{(2)}\cap\cdots\cap B_i^{(h_i)}/C_i\right)}{P\left(B_i^{(1)}\cap B_i^{(2)}\cap\cdots\cap B_i^{(h_i)}\right)}$$

$$= \frac{P(C_i)P^{h_i}\left(B_i^{(1)}/C_i\right)}{P(C_i)P^{h_i}\left(B_i^{(1)}/C_i\right)+P(\bar{C}_i)P^{h_i}\left(B_i^{(1)}/\bar{C}_i\right)} = \frac{p_i\cdot(1-\beta_i)^{h_i}}{p_i\cdot(1-\beta_i)^{h_i}+(1-p_i)\alpha_i^{h_i}}.$$

Claim 1 is proved.

Consider in more detail the search strategy ***s*** which is defined as an infinite-horizon sequence of location numbers, where, at each step *n*, location *s* *s*[*n*] is inspected and tested whether or not it contains the target:

$$s = \langle S[0], s[1], \cdots, s[n], \cdots\rangle.$$

In this sequence, the initial sub-sequence denoted by *S*(0) starts the search process. It is defined in such a way that the search cannot stop during this initial sub-sequence. For instance, we selected *S*(0) in the following way

$$S[0] = \left\{\underbrace{1,1,\cdots,1}_{H_1-1}, \underbrace{2,2,\cdots,2}_{H_2-1}, \cdots, \underbrace{N,N,\cdots,N}_{H_N-1}\right\}$$

Let us compute the problem's objective function *F*(*s*). For this aim, we need, first, to compute the unconditional probability $P_{s[n]}$ of the event that a location *s*[*n*] = *i* is detected by the sensor as a target-containing location exactly H_i times up to the *n*th step of strategy *s* (not necessarily successively). We introduce an auxiliary parameter $s^*[n]$ as follows: For any given sequence ***s*** and the location *s*[*n*] inspected by the sensor on step *n*, let $s^*[n]$ be the total number of inspections of location *s*[*n*] (not necessarily successively) up to the *n*th step of strategy ***s***. Obviously, $s^*[n] \le n$, for all *n*, and $H_{s[n]} \le s^*[n]$, for all *n* in ***s***.

Notice that the on-board computer of the UAV can easily count and store the $s^*[n]$ value in its memory as soon as the sequence ***s*** is known up to the *n*-th step.

Claim 3. The probability $P_{s[n]}$ can be computed as follows:

$$P_{s[n]} = \binom{s^*[n]-1}{H_{s[n]}-1}\left(1-f_{s[n]}\right)^{s^*[n]-H_{s[n]}}\left(f_{s[n]}\right)^{H_{s[n]}}$$

for $n \ge 1$ and $P_{s[n]} = 0$ for $n = 0$.

This claim immediately follows from the above definitions of $H_{s[n]}$ and $s^*[n]$, using the binomial distribution of $H_{s[n]} - 1$ inspections of the location *s*[*n*] in which the sensor has revealed the target in *s*[*n*], within the total number $s^*[n]$ of inspections of the location *s*[*n*] up to the *n*-th step of ***s***.

All the components of the problem's objective function *F*(*s*) can be now computed as follows:

- The time $T_{s[n]}$ spent for the inspection of all the elements $S[0], \cdots, s[n]$ up to the *n*th step of strategy ***s*** is:

$$T_{s[n]} = \sum_{i=1}^{N} t_i(H_i-1) + \sum_{m=1}^{n} t_{s[m]}, \quad n \ge 1.$$

- The unconditional probability $P_{s[n]}$ of the event that a location *s*[*n*] is detected by the sensor as a target-containing location exactly H_i times up to the *n*th step of strategy *s*-as defined in Claim 3.

Our main result is the index-based greedy algorithm below that permits to find the search strategy. The following theorem completely describes the algorithm.

Theorem. Define ratios $Q_{s[n]}$ as follows:

$$Q_{s[n]} = \left(c_{s[n]}/t_{s[n]}\right)\binom{s^*[n]-1}{H_{s[n]}-1}\left(1-f_{s[n]}\right)^{s^*[n]-H_{s[n]}}\cdot f_{s[n]}^{H_{s[n]}}$$

The strategy $\boldsymbol{s}^*$ is optimal iff the ratios $Q_{s[n]}$ are arranged in non-decreasing order, for all $n \ge 1$.

The proofs of Lemma and Theorem are done straightforwardly by the standard interchange argument (the interested reader is referred, for example, to Levner [18], Pinedo [28] and Kress *et al*. [5], for further details of this method).

The search procedure in Theorem is index-based and greedy, so far as at each step the searcher selects for inspection the next minimum-ratio location. When all the $\alpha_i = \beta_i = 0$, for all i, we have a special case of the so-called search scheduling problem with *perfect* inspections well known in scheduling literature (see, for example, Kadane and Simon [24] and Rabinowitz and Emmons [19]. When all $\alpha_i = 0$, but $\beta_i \neq 0$, this is the case of false-negative inspections; if $\alpha_i \neq 0$, but all $\beta_i = 0$, this is the case of the false-positive inspections only

5. Examples: Imperfect Search in a Small Area

The following examples illustrate the validity and applicability of the suggested model.

Example 1. Consider a problem of searching for a lost target by a single automatic device in a stochastic setting. The model and application are motivated by those described in Chung and Burdick [13]. An area of interest is divided into N possible locations in one of which a target object is hidden. The specificity of the considered automatic search is that this autonomous device, at each step of search, has a limited memory and so does not remember the full history of all the outcomes of its previous search steps. The on-board computer only remembers how many times a target has been detected in each location, up to a current step in the search sequence. The search terminate as soon as the number of such outcomes in one of the locations reaches its pre-specified critical height, H_i, which, in turn guarantees the required confidence level in identification of the target. For simplicity, we consider an area with only three locations numbered 1, 2 and 3. The input data are given in **Table 1**; the required confidence level CL is 0.95.

The results of intermediate computations are given in **Table 2**.

The ratios Q_i in decreasing order and corresponding location numbers are presented in **Table 3**.

The search process stops at the ninth step. The optimal strategy obtained up to the ninth step is: $\boldsymbol{s}^* = \langle S[0], 1, 2, 1, 2, 3, 3, 3, 3, 2, \cdots \rangle$; here $S[0] = \langle 1, 2 \rangle$. The minimum value of losses incurred by the obtained solution $F(\boldsymbol{s}^*)$ is 49.398. We compared the obtained minimum losses with the losses incurred by another strategy, $\boldsymbol{s}_1$, in which the locations are visited periodically, being arranged in each period in decreasing order of location probabilities: $s_1 = \langle 3, 2, 1;\ 3, 2, 1;\ 3, 2, 1; \cdots \rangle$. In this case, $F(\boldsymbol{s}_1) = 57.484$, which implies that $F(\boldsymbol{s}^*)$ is essentially better.

Example 2. Consider another area of interest with $N = 6$ possible locations in one of which a target object is

Table 1. Input data for Example 1.

Locations	1	2	3
$p_i = P(C_i)$	0.1	0.15	0.75
$\alpha_i = P(B_i / \bar{C}_i)$	0.04	0.06	0.12
$\beta_i = P(\bar{B}_i / C_i)$	0.1	0.07	0.05
t_i	5	8	10
c_i	1	1	1

Table 2. Intermediate results for Example 1.

Locations	1	2	3
H_i	2	2	1
f_i	0.126	0.190	0.743

Table 3. Locations selected at each step and ratios Q_i.

Step no. in $\boldsymbol{s}$	1	2	3	4	5	6	7	8	9
Ratio Q_i	0.1527	0.0819	0.0384	0.0312	0.0257	0.0191	0.0141	0.0105	0.0089
No. of location	1	2	1	2	3	3	3	3	2

hidden. In this example, we explore experimental data similar to those used in the failure search model developed by Levner [18]. The data are given in **Table 4** and a confidence level $CL = 0.95$.

The results of intermediate computations are given in **Table 5**.

The ratios Q_i in decreasing order and corresponding location numbers are presented in **Table 6**.

The optimal search strategy obtained up to the 13th step is: $\boldsymbol{s}^* = \langle S[0], 1, 2, 3, 4, 5, 3, 4, 4, 4, 3, 4, 4, 4 \rangle$, where $S[0] = \{1, 3, 5, 6\}$. The minimum losses incurred by this solution $F(\boldsymbol{s}^*)$ is 79.912. In the both numerical examples the ratios Q_i, as well as the probability that the search stops in a wrong location caused by a false alarm, quickly decrease as the number of steps in the search sequence grows. We compared the obtained minimum losses with the losses incurred by strategy $\boldsymbol{s}_2$, in which the locations are visited periodically, being arranged in each period in decreasing order of location probabilities: $\boldsymbol{s}_2 = \langle 4, 3, 2, 5, 1, 6; 4, 3, 2, 5, 1, 6; \cdots \rangle$. In this case, $F(\boldsymbol{s}_2) = 85.143$, thus, the optimal search strategy $\boldsymbol{s}^*$ is indeed advantageous.

6. Computational Experiments

In order to find the exact solutions of the considered problem on the random data, the developed algorithm was implemented using MATLAB v. 7.1 and executed on the following hardware: Dell PowerEdge R410 server, CPU-2x Intel Xeon X5660 @ 2.8 GHz (12 cores total), 64 GB RAM. The problem instances were generated randomly as shown in **Table 7**.

A total of 10,000 different instances were tested by the MATLAB solver. For all experiments, the calculation time was limited to 0.01 CPU-hour per problem instance. For each instance, the solver provided a greedy-generated solution and the best solution generated by the full search over 1000 randomly generated feasible strategies. The following phenomenon similar to the "80/20 rule" (also known as *the Pareto rule*, or "*the law of the vital few*", see *Rushton et al.* [29]) has been empirically observed. Namely, in our calculation for approximately 82% of the problem instances, about 20% of the randomly generated strategies provided a better solution than

Table 4. Input data for Example 2.

Location	1	2	3	4	5	6
$p_i = P(C_i)$	0.08	0.05	0.15	0.65	0.05	0.02
$\alpha_i = P(B_i / \bar{C}_i)$	0.05	0.01	0.07	0.1	0.02	0.03
$\beta_i = P(\bar{B}_i / C_i)$	0.06	0.05	0.1	0.01	0.07	0.03
t_i	5	3	7	13	3	2
c_i	1	1	1	1	1	1

Table 5. Intermediate results for Example 2.

	Locations					
	1	2	3	4	5	6
H_i	2	1	2	1	2	2
f_i	0.1212	0.057	0.1945	0.6785	0.0655	0.0484

Table 6. Selected locations at each step and ratios in Example 2.

No. of step	0	1	2	3	4	5	6
No. of location	1, 3, 5, 6	1	2	3	4	5	3
Ratio Q_i		0.3421	0.3154	0.2865	0.2361	0.2101	0.1840

No. of step	7	8	9	10	11	12	13
No. of location	4	4	4	3	4	4	4
Ratio Q_i	0.161	0.123	0.089	0.061	0.032	0.012	0.009

the yielded greedy solution.

The results for different average cost values (from $10 to $500) are presented in **Figure 1**.

In addition, we have observed that for the values of the confidence level *CL* varying in interval [0.8; 0.96], for different number of cells varying in interval [10; 30], the number of instances for which the greedy algorithm behaved better than all other strategies, slowly increased with the CL growing (see **Figure 2**).

In **Figures 3-5**, we present the average number *K* of steps before the search is terminated, as a function of the confidence level *CL* for 10,000 instances and different numbers of cells (*N* = 10; 20; 30).

It is worth noticing that the number of steps of the algorithm becomes approximately the same, for large values of the confidence level.

Table 7. The input data characteristics.

Data type	Range
Number of cells	Unif (10, 30)
Expected inspection time (in Min)	Unif (1, 10)
Location probabilities	Unif (0.04, 0.95)
Probabilities of false alarms	Unif (0.03, 0.5)
Probabilities of overlooking	Unif (0.04, 0.5)
Search cost rate (in relative units)	Unif (10, 40)

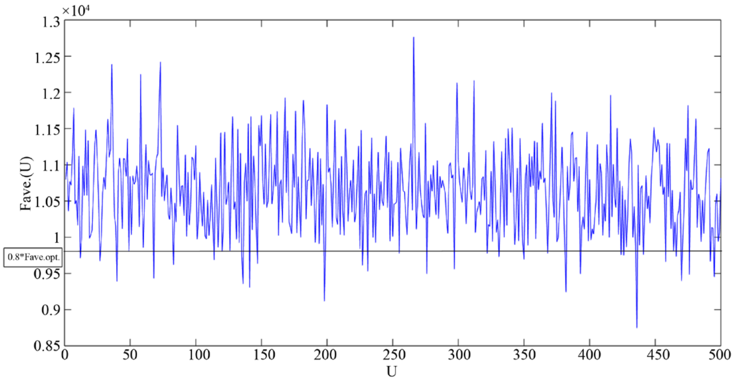

Figure 1. The average number of strategies yielding a better solution than the greedy solution.

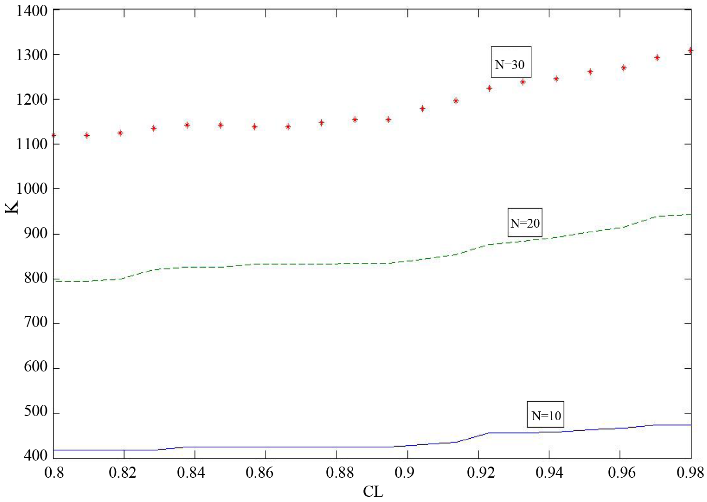

Figure 2. The number of instances for which the greedy algorithm was the best search algorithm, for different *CL*.

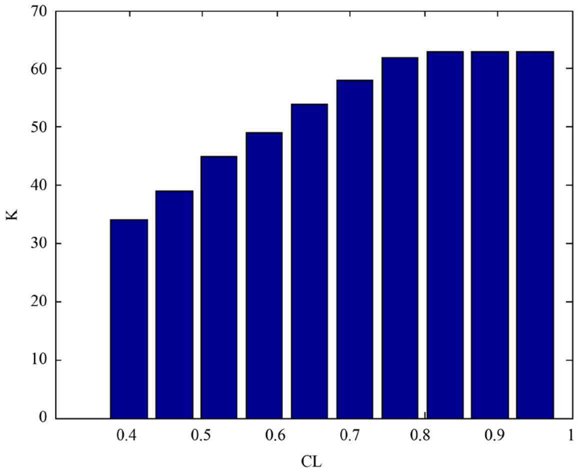

Figure 3. The average number K of steps for the number of cells $N = 10$.

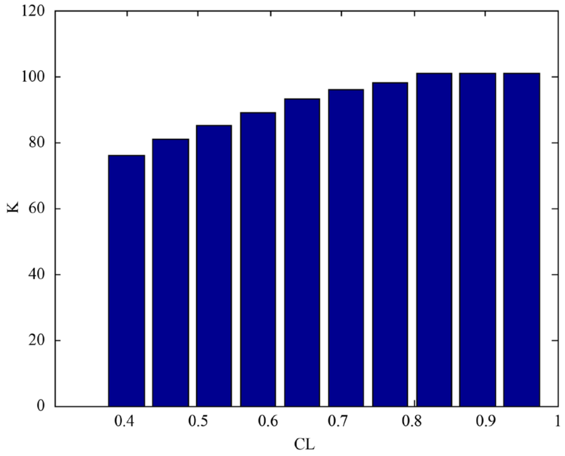

Figure 4. The average number K of steps for the number of cells $N = 20$.

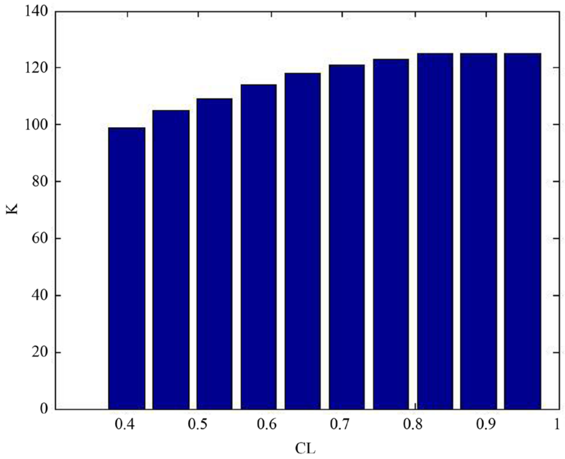

Figure 5. The average number K of steps for the number of cells $N = 30$.

7. Conclusions

We have studied the sequential discrete search problem for a lost or hidden target and considered a practice-oriented case where the automatic device performs imperfect inspections subject to the false-negative and false-positive inspection outcomes. For optimizing the search process minimizing the expected loss or damage, we suggest a greedy strategy. Being attractive due to its computational efficiency and simplicity, such local search strategy guarantees finding an optimal (minimum-loss) search sequence, for a pre-specified confidence level.

We believe that the suggested approach can be applied for a wider range of search scenarios (e.g., with multiple mobile agents, multiple targets, resource—and precedence constraints, agents with—and without memory). Notice that in the considered scenario the on-board computer program uses only partial information about the history of all search outcomes. Incorporating more complete information into the search model (for instance, joint probabilities of several locations) leads to a more complicated dynamic setting, which is an attractive direction for further research. The suggested greedy method could be combined with more general and sophisticated solution methods such as dynamic programming, branch-and-bound, and biology-motivated algorithms.

Another prospective future research is to compare the computational efficiency of different sequential and non-sequential search methods using analytical and simulation tools. It is known that the min-cost and max-probability versions of the discrete search problem with general resource-and time constraints are NP-hard (Wegener [8]; Trummel and Weisinger [9]). We conjecture that the min-loss versions of the considered problem also have this property. This is a challenging topic for future research.

References

[1] Stone, L.D. (1989) Theory of Optimal Search. 2nd Edition, Academic Press, New York.

[2] Benkoski, S.J., Monticino, M.G. and Weisinger, J.R. (1991) A Survey of the Search Theory Literature. *Naval Research Logistics*, **38**, 469-494. http://dx.doi.org/10.1002/1520-6750(199108)38:4<469::AID-NAV3220380404>3.0.CO;2-E

[3] Valavanis, K.P. (2008) Advances in Unmanned Aerial Vehicles, Springer, Berlin.

[4] Dell, R.F., Eagle, J.N., Martins, G.H.A. and Santos, A.G. (1996) Using Multiple Searchers in Constrained-Path, Moving-Target Search Problems. *Naval Research Logistics*, **43**, 463-480. http://dx.doi.org/10.1002/(SICI)1520-6750(199606)43:4<463::AID-NAV1>3.0.CO;2-5

[5] Kress, M., Lin, K.Y. and Szechtman, R. (2008) Optimal Discrete Search with Imperfect Specificity. *Mathematical Methods of Operations Research*, **68**, 539-549. http://dx.doi.org/10.1007/s00186-007-0197-2

[6] Sato, H. and Royset, J.O. (2010) Path Optimization for the Resource-Constrained Searcher. *Naval Research Logistics*, **57**, 422-444. http://dx.doi.org/10.1002/nav.20411

[7] Kress, M., Royset, J.O. and Rozen, N. (2012) The Eye and the Fist: Optimizing Search and Interdiction. *European Journal of Operational Research*, **220**, 550-558. http://dx.doi.org/10.1016/j.ejor.2012.02.016

[8] Wegener, I. (1985) Optimal Search with Positive Switch Cost Is NP-Hard. *Information Processing Letters*, **21**, 49-52. http://dx.doi.org/10.1016/0020-0190(85)90108-5

[9] Trummel, K.E. and Weisinger, J.R. (1986) The Complexity of the Optimal Searcher Path Problem. *Operations Research*, **34**, 324-327. http://dx.doi.org/10.1287/opre.34.2.324

[10] Washburn, A.R. (2002) Search and Detection (Topics in Operations Research Series). 4th Edition, INFORMS, New York.

[11] Song, N.O. and Teneketzis, D. (2004) Discrete Search with Multiple Sensors. *Mathematical Methods of Operations Research*, **60**, 1-13. http://dx.doi.org/10.1007/s001860400360

[12] Wilson, K.E., Szechtman, R. and Atkinson, M.P. (2011) A Sequential Perspective on Searching for Static Targets. *European Journal of Operational Research*, **215**, 219-226. http://dx.doi.org/10.1016/j.ejor.2011.05.045

[13] Chung, T.H. and Burdick, J.W. (2012) Analysis of Search Decision Making Using Probabilistic Search Strategies. *IEEE Transactions on Robotics*, **28**, 132-144. http://dx.doi.org/10.1109/TRO.2011.2170333

[14] Kriheli, B. and Levner, E (2013) Search and Detection of Failed Components in Repairable Complex Systems under Imperfect Inspections. In: Batyrshin, I. and Mendoza, M.G., Eds., *Advances in Computational Intelligence*, Lecture Notes in Artificial Intelligence, Vol. 7630, Springer, Berlin, 399-410. http://dx.doi.org/10.1007/978-3-642-37798-3_35

[15] Kriheli, B., Levner, E. and Spivak, A. (2013) Optimization Model of Routing-and-Searching Process Performed by Unmanned Aerial Vehicles in Special Target-Detection Operations. *Proceedings of the IFAC Conference on Manufacturing Modelling, Management and Control*, Vol. 7, Part 1, Saint Petersburg, 19-21 June 2013, 1838-1842.

[16] Kriheli, B., Levner, E., Bendersky, M. and Yakubov, E. (2015) A Fast Algorithm for Scheduling Detection-and-Rescue Operations Based on Data from Wireless Sensor Networks. *Research in Computing Science*, **104**, 9-21.

[17] Kadane, J.B. (1971) Optimal Whereabouts Search. *Operations Research*, **19**, 894-904. http://dx.doi.org/10.1287/opre.19.4.894

[18] Levner, E. (1994) Infinite-Horizon Scheduling Algorithms for Optimal Search for Hidden Objects. *International Transactions in Operational Research*, **1**, 241-250. http://dx.doi.org/10.1016/0969-6016(94)90023-X

[19] Rabinowitz, G. and Emmons, H. (1997) Optimal and Heuristic Inspection Schedules for Multistage Production Systems. *IIE Transactions*, **29**, 1063-1071. http://dx.doi.org/10.1080/07408179708966425

[20] Koopman, B.O. (1980) Search and Screening: General Principles with Historical Applications. Pergamon, New York.

[21] Sweat, C.W. (1970) Sequential Search with Discounted Income, the Discount a Function of the Cell Searched. *The Annals of Mathematical Statistics*, **41**, 1446-1455. http://dx.doi.org/10.1214/aoms/1177696790

[22] Chew, M.C. (1973) Optimal Stopping in a Discrete Search Problem. *Operation Research*, **21**, 741-747. http://dx.doi.org/10.1287/opre.21.3.741

[23] Assaf, D. and Zamir, S. (1985) Optimal Sequential Search: A Bayesian Approach. *The Annals of Statistics*, **13**, 1213-1221. http://dx.doi.org/10.1214/aos/1176349665

[24] Kadane, J.B. and Simon, H.A. (1977) Optimal Strategies for a Class of Constrained Sequential Problems. *The Annals of Statistics*, **5**, 237-255. http://dx.doi.org/10.1214/aos/1176343791

[25] Ross, S.M. (1969) A Problem in Optimal Search and Stop. *Operations Research*, **17**, 984-992. http://dx.doi.org/10.1287/opre.17.6.984

[26] Ahlswede, R. and Wegener, I. (1987) Search Problems. John Wiley & Sons, New York.

[27] Chudnovsky, D.V. and Chudnovsky, G.V. (1989) Search Theory, Some Recent Developments. Marcel Dekker Inc., New York.

[28] Pinedo, M. (2008) Scheduling: Theory, Algorithms, and Systems. Springer, Berlin**.**

[29] Rushton, A., Oxley, J. and Croucher, P. (2000) The Handbook of Logistics and Distribution Management. 2nd Edition, Kogan Page, London, 107-108.