

Aerial Vehicle Search-Path Optimization: A Novel Method for Emergency Operations

Manon Raap¹  · Silja Meyer-Nieberg¹ ·
Stefan Pickl¹ · Martin Zsifkovits¹

Received: 12 August 2015 / Accepted: 16 September 2016 / Published online: 26 September 2016
© Springer Science+Business Media New York 2016

Abstract This paper presents a novel search-path optimization method for moving target search by an aerial vehicle, applicable to realistically sized search areas. For such missions, long endurance vehicles are needed, which are usually fixed-winged. The proposed method accounts for flight kinematics of fixed-wing and rotary-wing aerial vehicles. It additionally accounts for movements of the target, considerably increasing complexity of search-path optimization, compared to a static target. The objective is to maximize the probability to detect a conditionally deterministic moving target within a given time period. We propose a first K -step-lookahead planning method that takes flight kinematic constraints into account and in which the target and platform state space are heterogeneous. It consists of a binary integer linear program that yields a physically feasible search-path, while maximizing the probability of detection. It is based on the MAX-K-COVERAGE problem, as it selects K waypoints while maximizing the probability that a target is within the field of view of a platform at one of these waypoints. This K -step-lookahead planning method is embedded in an iterative framework, where the probability of overlooking a target is fed back to the controller

Communicated by Kyriakos G. Vamvoudakis.

✉ Manon Raap
manon.raap@unibw.de
Silja Meyer-Nieberg
silja.meyer-nieberg@unibw.de
Stefan Pickl
stefan.pickl@unibw.de
Martin Zsifkovits
martin.zsifkovits@unibw.de

¹ Fakultät für Informatik, Universität der Bundeswehr München, Werner-Heisenberg-Weg 39, 85577 Neubiberg, Germany

after observations are made. Simulations show the applicability and effectiveness of this method.

Keywords Aerial vehicle · Moving target search · Search-path optimization · Time-dependent routing

Mathematics Subject Classification 90B40 · 90C10 · 90C90

1 Introduction

Consider a disastrous scenario like a plane crash at sea where the only information about the crash site comes from the flight plan, maybe supplemented with a Mayday call or a last blip on the radar. Among the survivors, some may have managed to get in a life raft, which may be drifting as it is subject to winds and currents. In such a case, it is not sufficient to just fully cover the search area, since it is possible for the life raft to move into an area that has already been observed. To increase the chance of survival, efficient planning methods are required. With the ongoing technological development of unmanned aerial vehicles (UAVs), the idea of employing a UAV for search has risen. For large scale search missions, a fixed-winged UAV is needed. Such platforms are more suitable due to their long endurance and increased load capacity, so that high quality sensors can be equipped. In this application, additional restrictions must be set for the planned search-path to be physically feasible. For example, a fixed-wing UAV is not able to hover, turn sharply or fly backwards. Besides accounting for the specific flight kinematics, the estimation of the target position and its movement must be considered. The difficulty here lies in the fact that the probability of detecting the target at a certain time k depends on the estimation of the position of the target at that time, which in turn is a function of *all* previous observations.

1.1 Related Work

To apply search theory to real-life search missions, efficient planning methods are required. Due to this important application, extensive research (initiated by Koopman [1]) has resulted in several optimization approaches. Stewart [2] proposed the path-constrained search effort allocation problem. Several branch and bound methods have been developed, e.g., with the MEAN bound by Martins [3], with the DMEAN bound by Lau et al. [4] and with a static bound by Sato [5]. Morin et al. [6] introduced a mixed-integer linear program with visibility constraint, in which agents are allowed to observe adjacent cells. The first approach explicitly considering aerial vehicle kinematics for search was presented by Bourgault et al. [7]. They proposed myopic control solutions, which tend to yield suboptimal search paths. Foraker et al. [8,9] focused on the continuous counterpart.

1.2 Contributions

To successfully employ fixed-wing platforms for moving target search, we need a method allowing for anticipation of the estimated future position and motion of the target, taking the flight kinematical constraint of such a platform into account. The individual considerations of these aspects can be found in the mentioned related work. However, there is no approach incorporating both in one method so far, as it is done in the paper at hand. Another aspect of our approach is the heterogeneous state space for the target and platform, whereas all published K -step-lookahead methods so far consider both to move on a single shared grid. This concept has two benefits; a target specific grid allows for a more precise estimation of the target position, whereas a platform specific grid allows for modeling more natural flight kinematics. Our approach furthermore incorporates a novel method for efficient relocation of the platform.

1.3 Outline

The remainder of the paper is organized as follows: In Sect. 2, we describe the preliminary problem statement, followed by the presentation of our method for search-path optimization in Sect. 3. In Sect. 4, we describe the iterative framework in which we embed our method. Concrete simulations are presented in Sect. 5. We discuss further research in Sect. 6 and end this paper with the conclusions in Sect. 7.

2 Problem Statement

This section describes the prerequisites, constraints, and objective that form the search-path problem as introduced in [7]. These consist of the following aspects we describe in the subsections: the target model, the sensor model, the platform model, and finally, the objective function.

2.1 Target Model

We consider the search for a moving target in discrete time in the search area $\mathbb{O} \subset \mathbb{R}^2$. The time allocated to a planning stage is defined by a sequence $\mathcal{K} = (0, \dots, K)$ of $K + 1$ time steps. A grid-based probability map partitions the search area uniformly into a finite set of cells \mathcal{C} . The target occupies one unknown cell $C_k \in \mathcal{C}$ at time step $k \in \mathcal{K}$. For the duration of the search mission, a probability map p_{C_k} is maintained, in which the probability of containment $p_{C_k}(c)$ represents the probability of the target occupying cell c at time k without being detected prior to time k . Although the initial position of the target is unknown, it is characterized by a known prior probability distribution p_{C_0} .

A target path can be modeled by a stochastic process (C_0, \dots, C_K) , which is often assumed to be a Markovian process. We, however, consider the special case of a conditionally deterministic target. Here, the path of a target depends merely on its initial position C_0 . If this stochastic variable was known, the path of the target would

be known entirely. Thus, the path of the target is deterministic, conditioned on its initial position. This assumption is made more often for survivor search at sea [10] and also for search problems in general [11]. The probability map evolves due to the target motion model according to

$$pc_{k+1}(c) = \sum_{c' \in \mathcal{C}} d(c', c) pc_k(c'), \tag{1}$$

where the transition function $d(c', c) \in \{0, 1\}$ represents the probability that the target moves from cell c' to cell c and is assumed to be known for all $c' \in \mathcal{C}$.

2.2 Sensor Model

The considered aerial sensor platform is equipped with a stabilized sensor, which is used for making observations. Two results are defined for the observation $Z_{c,k}$ on cell $c \in \mathcal{C}$ at time k :

$$Z_{c,k} := \begin{cases} 1, & \text{if the platform detects the target in cell } c \text{ at time } k, \\ 0, & \text{otherwise.} \end{cases}$$

The glimpse probability $pg_k(o_k, c)$ represents the probability of target detection, given target occupancy within cell c and platform position o_k at time k , i.e.,

$$pg_k(o_k, c) := P(Z_{c,k} = 1 | C_k = c, o_k). \tag{2}$$

The *overlook* probability po_k represents the complement of the glimpse probability, i.e., the probability of overlooking the target, given target occupancy within cell c and platform position o_k at time k , i.e.,

$$po_k(o_k, c) := P(Z_{c,k} = 0 | C_k = c, o_k) = 1 - pg_k(o_k, c). \tag{3}$$

When observations are made, the probability map is updated by the glimpse probability as in Eq. (2). Therefore, Eq. (1) is extended to account for observation results as follows:

$$pc_{k+1}(c) = B \sum_{c' \in \mathcal{C}} d(c', c) pc_k(c') (1 - pg_k(o_k, c')), \tag{4}$$

where the normalization coefficient B is given by

$$B = \left(\sum_{c \in \mathcal{C}} pc_k(c) (1 - pg_k(o_k, c)) \right)^{-1}. \tag{5}$$

2.3 Platform Model

The object to control in this problem is an aerial platform. Its motion model, adopted from [12], is the same for rotary-wing and fixed-wing platforms and is given by

$$\begin{aligned} x_{k+1} &= x_k + s_k \cdot \cos(\theta_k + \alpha_k) \\ y_{k+1} &= y_k + s_k \cdot \sin(\theta_k + \alpha_k) \\ \theta_{k+1} &= \theta_k + \alpha_k, \end{aligned} \tag{6}$$

where the parameter s_k is the speed of the platform, the parameter θ_k is its heading angle and α_k is its change of heading at time k , which are obviously restricted to the laws of physics. The variables x_k and y_k represent the coordinates of the platform on the plane above search area \mathbb{O} .

2.4 Objective

The objective is to determine a search-path $o = (o_1, \dots, o_K)$ maximizing the cumulative probability of detection over time period $\{1, \dots, K\}$, i.e.,

$$\max_o \sum_{k=1}^K \sum_{c \in \mathcal{C}} pd_k(o_k, c), \quad pd_k(o_k, c) = pc_k(c) pg_k(o_k, c) \tag{7}$$

where $pd_k(o_k, c)$ is the probability of detecting the target at time k in cell c . Furthermore, the probability of containment $pc_k(c)$ is calculated by Eq. (4), but with $B = 1$. The normalization from Eq. (5) is omitted, so that the probability map is not normalized. Consequently, the probability of containment $pc_k(\cdot)$ does not represent an actual probability anymore, since it does not sum to unity over the grid cells. It does, however, sum to the probability that the target has not been found up until time k , despite the search effort. Therefore, the objective function in Eq. (7) yields the cumulative probability of detection over \mathcal{K} .

3 Method for Search-Path Optimization

In this section, we introduce our K -step-lookahead planning method for search-path optimization. First, in Sect. 3.1, we use the platform model to construct a discretized and finite platform state space. We then construct a reduced graph, of which the nodes are a subset of the platform state space, in Sect. 3.2. Then, in Sect. 3.3, we describe how we apply certainty equivalence to the overlook probability. We furthermore reformulate the probability map into a set of elements in Sect. 3.4. We then define the search-path problem on the reduced graph in Sect. 3.5, which we formulate as binary integer linear program in Sect. 3.6.

3.1 Platform State Space

The platform motion model in (6) results in an infinite and continuous platform state space. For efficiency purposes, we assume a constant speed $s_k = s$ for all k and

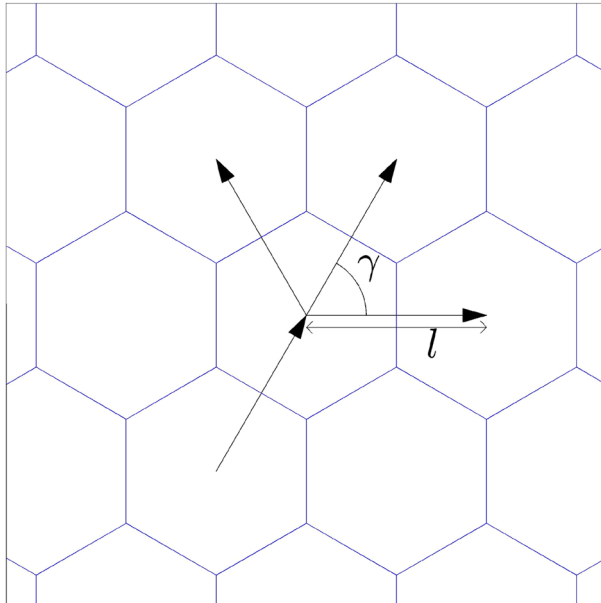


Fig. 1 Three fixed-wing platform control options, resulting in a hexagonal platform space discretization

limit the change of heading α to a predefined set of options. For fixed-wing platforms, we choose $\{-\gamma, 0, \gamma\}$ (in radians), and for rotary-wing platforms, we choose $\{-2\gamma, -\gamma, 0, \gamma, 2\gamma, 3\gamma\}$ (also in radians). We use $\gamma = \frac{\pi}{3}$, such that the resulting platform state space \mathcal{V}' is discretized and finite. The Voronoi diagram induced by this platform state space forms a hexagonal grid. Figures 1 and 2 show a segment of this grid, supplemented by the control options. The Euclidean distance $l := \|o_k - o_{k+1}\|$ represents the relocation distance in one time step when relocating with speed s .

A hexagonal grid for the aerial platform allows for a more natural modeling of its kinematics, when compared to using a square grid. All published K -step-lookahead methods for moving target search optimization (a.o. [2,4–6]) use a single grid for both searchers and targets. However, such an approach compromises the level of detail of the estimation of the target position. The proposed method uses a finer grid for the target, which allows for a more detailed estimation of the target position. A visualization of the heterogeneous grids is shown in Fig. 3.

3.2 Reduced Graph

A reduced graph $G = ((\mathcal{V} \cup v_0), A, R)$ is used for search-path planning and is defined by its nodes \mathcal{V} , adjacency matrix A , reachability matrix R and the initial position $v_0 \in \mathcal{V}'$ of the platform. The complexity of the search-path problem increases exponentially with the number of nodes, so a reduced graph is beneficial to decrease computational costs. The nodes are therefore restricted to those which yield a significant glimpse probability at some time k , i.e.,

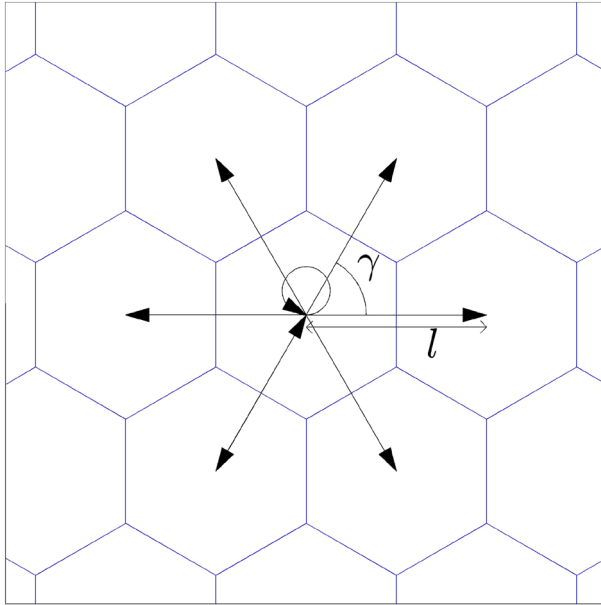


Fig. 2 Seven rotary-wing platform control options, resulting in a hexagonal platform space discretization

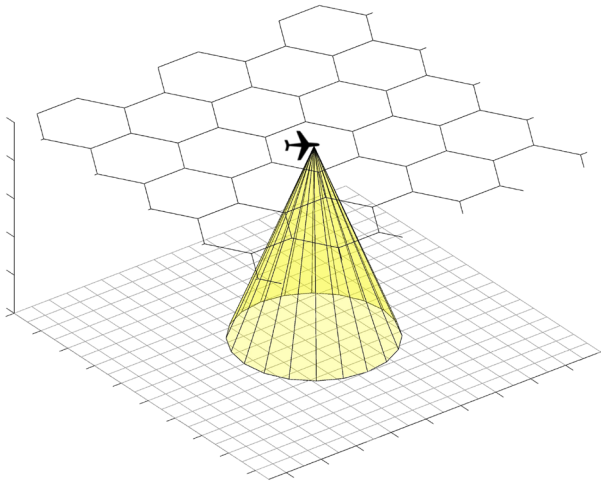


Fig. 3 The heterogeneous state spaces for the target (*square grid*) and the platform (*hexagonal grid*)

$$\mathcal{V} := \left\{ v \in \mathcal{V}' : \exists k \in \mathcal{K} : \sum_{c \in \mathcal{C}} pgk(v, c) \geq \tau \right\},$$

where τ is a threshold value. The binary adjacency matrix A is constructed by adding an arc between each pair of direct neighbors as follows. An entry $a_{v,v'}$ is 1 if node v is adjacent to node v' and 0 otherwise. The parameter $\tilde{a}_{v,v'}$ is its negation. Furthermore,

we define $\mathcal{V}(v)$ to be the set of nodes adjacent to node v , i.e., $\mathcal{V}(v) := \{v' \in \mathcal{V} : a_{v,v'} = 1\}$.

The reachability matrix R is used to direct the platform toward a node $v \in \mathcal{V}$ and is constructed as follows. First, recall v_t to be the platform position at time t . Furthermore, let k' be the minimal number of time steps that the platforms requires to reach any node in \mathcal{V} . A platform relocates with predefined speed $s_{\text{reloc}} > s$. It holds this increased speed until $t + k' - 1$ and starts searching again at time $t + k'$. The set of relocation arcs is represented by the binary reachability matrix R of size $|\mathcal{V}| \times K$, where entry

$$r_{v,k} := \begin{cases} 1, & \text{node } v \text{ is reachable in } k' + k \text{ time steps,} \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

The parameter $\tilde{r}_{v,k}$ is its negation. By using these relocation arcs, there is at least one feasible search-path yielding a reward in terms of probability of detection. Note that the horizon recedes with $k' + K$ time steps at each planning stage.

3.3 Model Simplification

To further simplify the model for planning purposes, we assume the target to be detected with probability one when it is in the field of view (FOV) of the sensor platform. In other words, we set the overlook probability to zero while planning. After an observation is made, the actual overlook probability is fed back to the controller and taken into account in the subsequent planning stage. Let us define the field of view as follows.

Definition 3.1 (*Field Of View (FOV)*) The *field of view* $FOV(v)$ of a platform at node v at time k is given by a subset of \mathcal{C} such that the glimpse probability exceeds a certain threshold $\eta > 0$, i.e.,

$$FOV(v_k) := \{c \in \mathcal{C} : pg_k(v_k, c) \geq \eta\}.$$

We exchange the glimpse function from Eq. (2) by the following, such that a target within field of view $FOV(v_k)$ is detected with probability one and zero otherwise, i.e.,

$$pg_k(v_k, c) := \begin{cases} 1, & \text{if } c \in FOV(v_k), \\ 0, & \text{otherwise.} \end{cases}$$

3.4 Reformulation of the Probability Map

We reformulate the probability map into a set of elements \mathcal{E} . Here, we add one element e to the set \mathcal{E} for each cell within a FOV at a node, i.e., $c \in \bigcup_{v \in \mathcal{V}} FOV(v)$. The initial probability of containment of the cell $pc_0(c)$ is assigned as reward w_e to the corresponding element, i.e., $w_e = pc_0(c)$. Element e is furthermore initially located

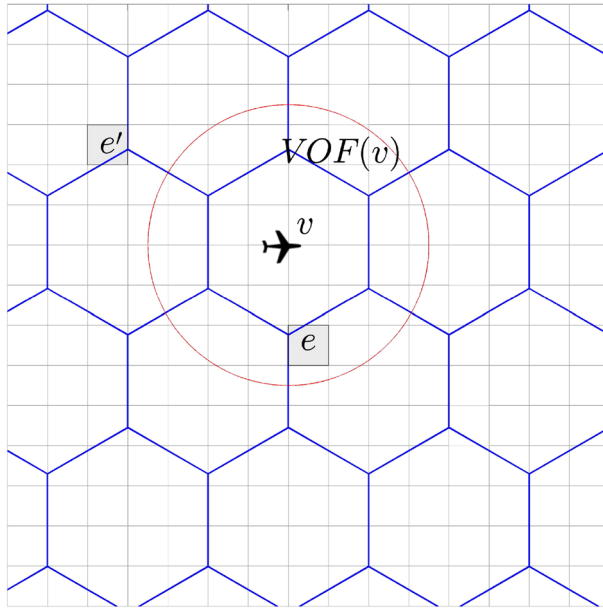


Fig. 4 Element e is within the FOV of node v , whereas element e' is not. Therefore, $l_{e,v}^k = 1$ and $l_{e',v}^k = 0$ for current time k

at cell c , and its location over time is updated according to the transition probability $d(c', c)$. A binary location matrix L specifies the location of the elements with respect to the FOV of a node as follows: An entry $l_{e,v}^k$ is 1 if element e is within the FOV of node v at time k and 0 otherwise. Figure 4 shows this schematically.

3.5 The Search-Path Problem on the Reduced Graph

Now that the graph G , the set of elements \mathcal{E} , and its location matrix L are defined, we can formulate the search-path problem as follows. It consists of finding a physically feasible walk on the graph G , such that the sum over the reward of covered elements is maximized. For a simple graph (which has no multiple arcs), a walk may be specified completely by a sequence of nodes [13]. Formally:

Definition 3.2 (Walk) Let $\mathcal{K}' = \{k_{\text{start}}, k_{\text{start}} + 1, k_{\text{start}} + 2, \dots, k_{\text{end}}\} \subset \mathbb{N}$ be a sequence of time steps. A walk is a sequence of nodes $(v_k)_{k \in \mathcal{K}'}$, where consecutive nodes in the sequence are adjacent nodes in the graph. That is, node v_{k+1} must be in set $\mathcal{V}(v_k)$, for all $k \in \{k_{\text{start}}, \dots, k_{\text{end}} - 1\}$.

The physical feasibility of a walk is inherent for a rotary-wing platform. However, a fixed-wing platform can not hover, make sharp turns, or fly backwards.

Definition 3.3 (Physically Feasible Walk) A walk $(v_k)_{k \in \mathcal{K}'}$ is physically feasible for fixed-wing platforms if and only if node v_{k+1} is in set $\mathcal{V}(v_k) \setminus \mathcal{V}(v_{k-1})$, for all $k \in \{k_{\text{start}} + 1, \dots, k_{\text{end}} - 1\}$.

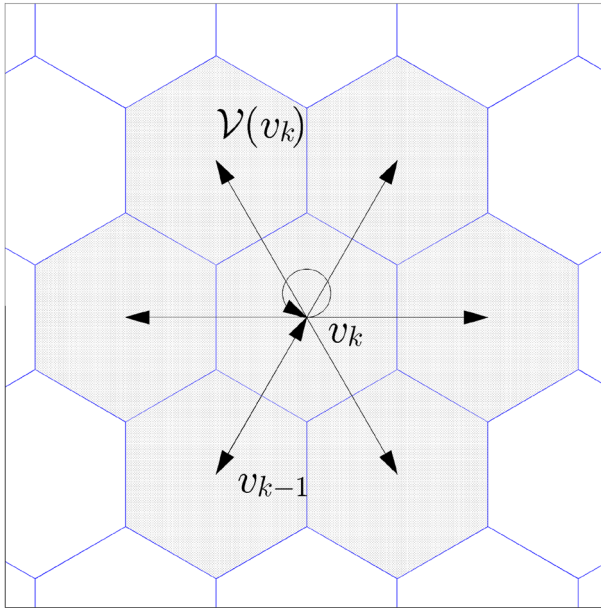


Fig. 5 Kinematical constraints on the walk for rotary-wing platforms. The nodes in set $\mathcal{V}(v_k)$ are adjacent to node v_k . A walk (v_{k-1}, v_k, v_{k+1}) is physically feasible for a rotary-wing platform if and only if node $v_{k+1} \in \mathcal{V}(v_k)$

The sets $\mathcal{V}(v_k)$ and $\mathcal{V}(v_k) \setminus \mathcal{V}(v_{k-1})$ are shown schematically in Figs. 5 and 6, respectively.

3.6 Binary Integer Linear Programming Formulation

To find a physically feasible walk that maximizes the probability of detection, we formulate the following binary integer linear program (BILP). It is based on the MAX-K-COVERAGE problem [14], as it selects K waypoints while maximizing the probability that a target is within the field of view of a platform at one of these waypoints. The constraints restrict the choice of waypoints, so that they define a physically feasible search-path.

Decision Variables

Let $\mathbb{B} := \{0, 1\}$. Decision variable $z_v^k \in \mathbb{B}$ is 1 if the platform is at node v at time k and 0 otherwise. These are the main decision variables and describe a physically feasible walk. The auxiliary decision variable $z_e \in \mathbb{B}$ is 1 if element e is covered and 0 otherwise.

Objective

The objective is to maximize the sum over the rewards of all covered elements, i.e.,

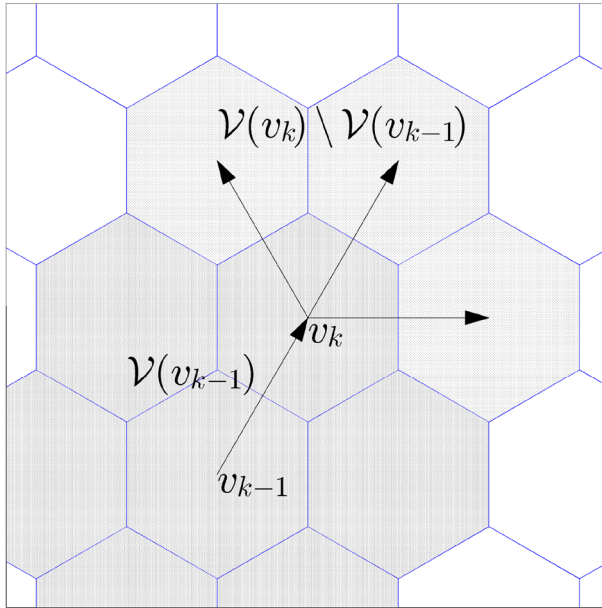


Fig. 6 Kinematical constraints on the walk for fixed-wing platforms. The nodes in set $\mathcal{V}(v_{k-1})$ are adjacent to node v_{k-1} . Analogously, the nodes in set $\mathcal{V}(v_k)$ are adjacent to node v_k . A walk (v_{k-1}, v_k, v_{k+1}) is physically feasible for a fixed-wing platform if and only if node $v_{k+1} \in \mathcal{V}(v_k) \setminus \mathcal{V}(v_{k-1})$

$$\max \sum_{e \in \mathcal{E}} w_e z_e, \tag{9}$$

where an element is covered if there is at least one $k \in \mathcal{K}$ such that the element is located at time k within the FOV of the platform at node v , i.e.,

$$\sum_{k=1}^K \sum_{v \in \mathcal{V}} l_{e,v}^k z_v^k \geq z_e, \quad \forall e \in \mathcal{E}. \tag{10}$$

In this formulation, elements that are observed multiple times, only count once in the objective function. This is analogous to the MAX-K-COVERAGE problem.

Walk Constraints

The following constraints are introduced to ensure the necessary structure of a walk. First of all, the walk must be a sequence of nodes. So for any time k a maximum of one node can be selected, i.e.,

$$\sum_{v \in \mathcal{V}} z_v^k \leq 1, \quad \forall k \in \mathcal{K}. \tag{11}$$

This sum may be zero, because it is possible for a platform to relocate during a number of time steps, say k' , before the search starts. In this case, no nodes are selected for

time steps $k < k'$. However, once the platform is at a node at time $k \geq k'$, it also has to be at a node at time $k + 1$, i.e.,

$$\sum_{v \in \mathcal{V}} z_v^k - \sum_{v \in \mathcal{V}} z_v^{k+1} \leq 0, \quad \forall k \in \{1, \dots, K - 1\}. \tag{12}$$

The next constraint ensures the adjacency of direct successive nodes, i.e.,

$$\sum_{v' \in \mathcal{V}} \tilde{a}_{v,v'} z_{v'}^{k+1} + z_v^k \leq 1, \quad \forall k \in \{1, \dots, K - 1\}, \forall v \in \mathcal{V}. \tag{13}$$

Kinematical Constraints

The flight kinematic constraints ensure that the walk contains no sharp turn, no loop, and no cycle of length two for fixed-wing platforms. To this end, we introduce parameter $\psi \in \mathbb{B}$, which assumes the value zero if the platform is fixed-winged:

$$\sum_{v' \in \mathcal{V}} a_{v,v'} z_{v'}^{k+1} + z_v^{k-1} \leq 1 + \psi, \quad \forall k \in \{2, \dots, K - 1\}, \forall v \in \mathcal{V}. \tag{14}$$

A rotary-wing platform, however, is able to make sharp turns, hover and fly backwards. Therefore, such platforms should not be restricted by this constraint. Parameter ψ assumes the value one if the platform is rotary-winged. This constraint is thereby relaxed for rotary-wing platforms.

At the start of a planning stage, the platform either relocates or keeps searching. These scenarios require different constraints; either the relocation constraint or the connecting constraint.

Relocation Constraint

The relocation constraint ensures the reachability of an assigned node. It prevents the assignment of the platform to a node at a time k , when it is physically out of reach:

$$\sum_{v \in \mathcal{V}} \sum_{k=1}^K z_v^k \tilde{r}_{v,k} = 0. \tag{15}$$

Connecting Constraints

The connecting constraints ensure the physical feasibility over the entire duration of the search mission, by connecting the walks of the consecutive planning stages. The first newly assigned node is restricted by the second last node (denoted by v_{-1}) and by the last node (denoted by v_0). These nodes have been selected in the previous planning stage and are therefore mandatory at the start of the next walk, i.e.,

$$z_{v_0}^0 = 1, \quad z_{v_{-1}}^{-1} = 1. \tag{16}$$

Constraints (12) and (13) for time $k = 0$ and constraint (14) for times $k = -1$ and $k = 0$ are furthermore included to ensure the physical feasibility on the connection between the walks. To be able to fix the last two nodes v_{-1} and v_0 in the next planning stage, at least two nodes are to be selected in the current planning stage. This is ensured by the following constraint:

$$\sum_{v \in \mathcal{V}} \sum_{k=1}^K z_v^k \geq 2. \tag{17}$$

Binary constraints

The binary constraints for all decision variables are

$$z_e, z_v^k \in \mathbb{B}, \quad \forall e \in \mathcal{E}, \forall k \in \mathcal{K}, \forall v \in \mathcal{V}. \tag{18}$$

The number of possible physically feasible walks on the graph G is of order $\mathcal{O}(2^{K|\mathcal{V}|})$.

Lemma 3.1 *There exists an optimal solution to the BILP in (9)–(18).*

Proof We show that all constraints hold for a given physically feasible walk on the graph G , that the objective value in (9) is bounded and that the discrete solution space is nonempty and finite. Let $(v_k)_{k \in \mathcal{K}}$ be a physically feasible walk on the graph G that is reachable by a the platform. Then, for each $k \in \{2, \dots, K - 1\}$, it holds that $v_{k+1} \in \mathcal{V}(v_k) \setminus \mathcal{V}(v_{k-1})$ by Definition (3.3), and for $k = 1$, it holds that $v_{k+1} \in \mathcal{V}(v_k)$ by Definition (3.2), i.e., $a_{k+1,k} = 1$ for each $k \in \{1, \dots, K - 1\}$ and $a_{k+1,k-1} = 0$ for each $k \in \{2, \dots, K - 1\}$. Moreover, $r_{v,k} = 1$ for each $k \in \mathcal{K}$ by construction of the reachability matrix R as in (8). The values of the decision variables $\{z_v^k : k \in \mathcal{K}, v \in \mathcal{V}\}$ corresponding to $(v_k)_{k \in \mathcal{K}}$ are $z_{v'}^{k'} = 1$ if the k' -th entry of $(v_k)_{k \in \mathcal{K}}$ equals v' and 0 otherwise. It is now easy to verify that constraints (11)–(18) are satisfied. Furthermore, we can set all values for the auxiliary variables $\{z_e : e \in \mathcal{E}\}$ at their lower bounds at zero, so that also constraint (10) is satisfied for both a fixed-wing platform ($\psi = 0$) and a rotary-wing ($\psi = 1$) platform. The objective value in (9) is bounded, because it is a finite sum over linear terms in a single bounded variable. Finally, we have $K|\mathcal{V}| + |\mathcal{E}|$ binary decision variables and hence a finite solution space with a size of order $\mathcal{O}(2^{K|\mathcal{V}|+|\mathcal{E}|})$.

We conclude that there exists an optimal solution to the BILP in (9)–(18). □

4 Iterative Framework with Feedback

In this section, we describe the essence of solving the iterative search-path planning method. In each iteration, a K -step-look-ahead planning is solved, where the reward after K time steps is ignored. This iterative framework accounts for sensor imperfections and also for sensor disturbances. We do this by setting the overlook probability to zero. Then, when an observation is made, an estimation of the overlook probability is fed back to the planner and can be taken into account in the consequent planning stage. Feeding back the estimated overlook probability has two major advantages.

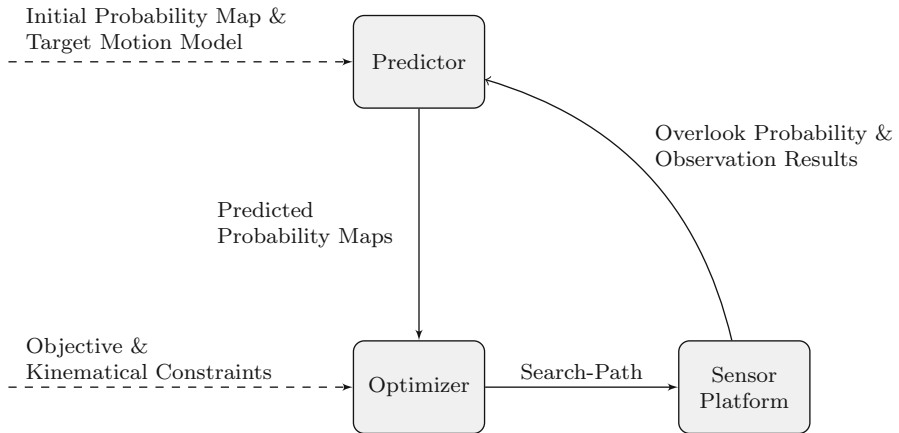


Fig. 7 Schematic overview of the iterative framework for moving target search

First, it significantly simplifies the problem to be deterministic. Secondly, the overlook probability is hard to predict in advance in realistic search scenarios (due to, e.g., fog, smoke or obstacles). By applying this iterative framework, the planning period at each stage remains constant, and the horizon recedes with K time steps until the target is found. Figure 7 shows a schematic overview of search-path planning in an iterative framework.

5 Simulations

In this section, we present the simulation environment and the results. All simulations were performed on an Intel(R) Core™ i7-4810MQ CPU processor with 2.80 GHz and a usable memory of 15.6 GB. The simulation platform is written in MATLAB, using the commercial solver Gurobi to solve the BILPs.

We compared the BILP method with the artificial potential field (APF) method [15], under the exact same kinematical constraints as for the BILP method. We chose this well-established method, because it is often used in control applications including scenarios for autonomous UAV path-planning and, in particular, for UAV search for moving targets [16]. Another reason to use the APF method is its robustness to the challenge of deciding when the platform is in an area where the probability of containment is zero. In this case, the probability of detecting the target may be zero for each possible search-path and no decision can be made by many greedy or even K -step-lookahead methods. Since the APF utilizes a potential field, this problem only occurs when the platform flies exactly centered in a *symmetric* distributed probability of location of the target.

5.1 Experimental Set-Up

To generate random experiments, we developed test instance generators for *two* types of instances. The first type of instances represents situations at the start of the search

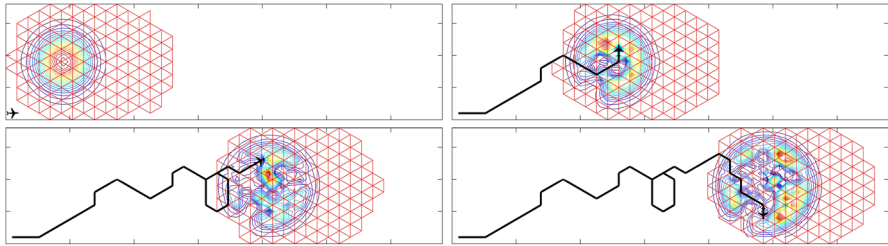


Fig. 8 Snapshots of a simulation of one fixed-wing platform searching for a single target

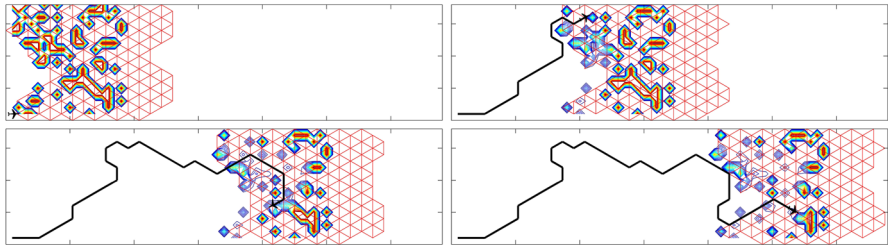


Fig. 9 Snapshots of a simulation of one fixed-wing platform searching for a single target with a scattered location probability function

mission, where the pdf is still *dense*. See, for example, the leftmost snapshot in Fig. 8. The second type of instances represent situations where the pdf is *scattered* of the search area, as shown in Fig. 9. This situation also occurs typically after observations have been made. See, for example, the rightmost snapshot in Fig. 8. We used equal settings for most characteristics for each type of test instance. We will enumerate there first starting with the platform characteristics. We set $s_{reloc} = M$ as the relocation speed, where M is some large value such that each node is reachable within one time step. This way, the disadvantageous greedy character of the APF is canceled out during relocation. The start position is irrelevant due to the large M and the homogeneous environment. We used $s = 4$ as the search speed, resulting in a distance between nodes of $l = 4$. Consequently, the inradius of the hexagons is two (as in Fig. 4). For sensor characteristics, we used the typical glimpse probability function [17, 18]:

$$pg_k(o_k, c) = 1 - \exp^{-\omega(o_k, c, k)},$$

with $\omega(o_k, c, k) \geq 0$ being a measure of search effectiveness for cell c . The search effectiveness decreases with the Euclidean distance $\|o_k - c\|$ between cell c and the platform at o_k at time k , as follows:

$$\omega(o_k, c, k) = W (\|o_k - c\|)^{-1},$$

where W is some sensor quality indicator for which we used $W = 0.735$. We use $\eta = 0.5$ as the detectability threshold, resulting in a FOV with radius 2.5. Since we focus on conditionally deterministic target behavior, the transition matrix is binary.

The target moves one cell in east direction with probability one. So far, we mentioned all characteristics that are uniform for all test instances. The major aspect in which the instances of each type differ, is the randomly generated start state of the search mission. We distinguish between start states with a *dense* pdf and start states with a *scattered* pdf:

- (i) **Dense pdf:** The search missions in these simulations take place on a 100×100 square grid. All initial probability maps were bivariate normal distributed ($\sim \mathcal{N}(\mu, \Sigma)$), with $\mu = \begin{pmatrix} 50 \\ 30 \end{pmatrix}$ and $\Sigma = \begin{pmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{pmatrix}$, where $\sigma_1^2, \sigma_2^2 \sim \mathcal{U}(40, 75)$.
- (ii) **Scattered pdf:** The search missions in these simulations take place on a smaller 28×36 square grid. Whether a cell has a probability of containing the target greater than zero is Bernoulli distributed, with parameter ξ uniformly distributed, i.e., $\xi \sim \mathcal{U}(0.1, 0.5)$. Then, the probability in which of these cells the target is located is uniformly distributed over the positive cells.

We ran a total of 200 test instances on randomized initial probability maps, 20 for each of the time periods $K \in \{6, 8, 10, 12, 14\}$, for both dense and scattered types of tests. The test results show the success of the search, measured by the achieved cumulative probability of detection. These results are presented next.

5.2 Results

Both methods are applicable to search-path planning for moving targets. However, the results are most promising for the proposed BILP method. The simulation results in Fig. 10 show the improved cumulative probability of detection in percentage for the *dense* test instances compared to that achieved by the APF method. The simulation results in Fig. 11 show the results analogously for the *scattered* test instances. On many instances, the BILP yielded 20 % improvement and even an improvement in more than 73 % was achieved on a scattered test instance.

There is a noticeable difference when comparing the results for a dense (Fig. 10) and scattered (Fig. 11) pdf. We notice a decrease in superiority of the BILP with increasing K in the first case (dense), whereas there seems no significant decrease in superiority in the latter case. This can be explained by the fact that when the cumulative probability of detection approaches one, the reward yielded at a time step approaches

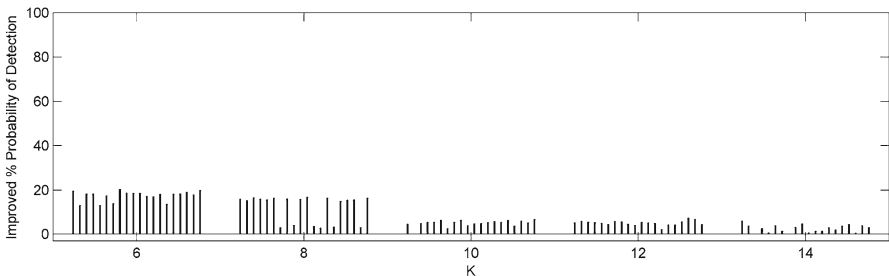


Fig. 10 Simulation results for *dense* pdf: cumulative probability of detection

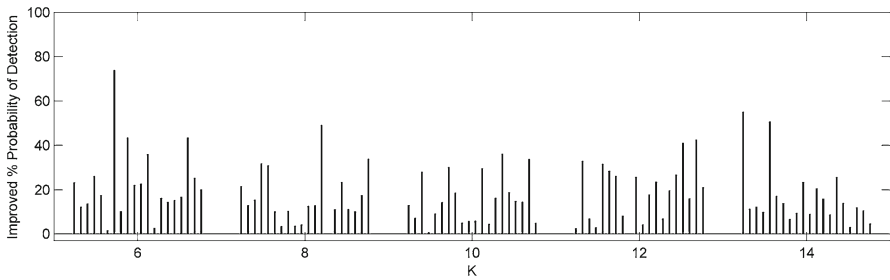


Fig. 11 Simulation results for **scattered** pdf: cumulative probability of detection

Table 1 Simulation results: available, maximum, and average computation times (in seconds) of the BILP and APF methods

K	Available	BILP		APF	
		AVG	MAX	AVG	MAX
6	120.00	0.40	0.69	0.03	0.06
8	160.00	0.94	2.35	0.19	0.22
10	200.00	3.10	8.39	0.23	0.27
12	240.00	14.74	40.14	0.28	0.33
14	280.00	110.18	341.13	0.30	0.32

zero. Indeed, in the first case it is more likely to approach a cumulative probability of detection of one in a dense pdf when compared to a scattered pdf, because in the former the high reward actions are grouped together around the mean of the pdf. The computation times are shown in Table 1.

Within the iterative framework, optimization is performed during the execution of a previously planned search-path. Hence, this duration is the available time for computation and is shown in the second column in Table 1. An estimation on available computation time can be acquired using the standard turn rate in aviation. It takes the platform two minutes to turn 360°. This takes six nodes on the reduced graph. So each time step would approximately have a duration of 20 s in the application. This means that when $K = 12$, the solver has four minutes available for computation of the next stage. The problem was solved well within available computation time for all instances with $K \leq 12$, but the available computation time was exceeded for a few instances with $K = 14$. So for $K \geq 14$, the problem may become intractable on realistically sized search areas for the solver we used. In this case, the APF can be used as backup method to provide a feasible solution at very low computational cost, as it needs less than one second on all instances.

6 Further Research

The aim in further research is to decrease the computation time of the proposed method. Promising efficient algorithms for path-constrained moving target search exist [3,4], which could be adopted to the case in which kinematical constraints must be taken into

account and in which the target- and searcher grids are heterogeneous. The current state-of-the-art approach is the branch and bound method with DMEAN bound by Lau et al. [4]. This method generalizes and tightens the MEAN bound by Martins [3]. Both methods branch on the waypoints and compute the bound by solving a longest path problem on an directed acyclic graph, while ignoring observation results. As the authors point out, this bound is not as strong for cases in which targets have a high probability of staying in their current cell, say c . Over 200 times more bounding attempts are needed on those test instances when compared to the cases with more energetic targets. This can be explained by the fact that the computed longest path in the calculation of the bound is a repetition of cell c , for which observations after the first are clearly redundant. The same argument holds in our case with conditionally moving targets. Therefore, our further research aims for the development of a branch and bound procedure for solving our model that employs a more suitable bound for conditionally moving targets and, moreover, takes kinematical constraints into account and applies to heterogeneous target- and searcher grids.

7 Conclusions

Several recent cases, especially emergencies at sea, have shown the high importance of acquiring knowledge about the location of a target. Natural disasters, extreme cases in air transport and the shipping industry as well as terroristic threats, are the basis for possible scenarios of interest. Many of these scenarios can have disastrous consequences when the target is not found (in time). Effective search-path optimization methods are therefore needed. We presented a novel approach, consisting of a binary integer linear program (BILP) solution, embedded in an iterative framework.

We ran simulations for several search-path lengths and compared the performance of our approach to a well-established method for search-path planning. Results show that our approach yields better results on all test instances, however at increased computational costs. Some time for optimization is available though, since it is performed during the execution of a previously planned search-path.

The main goal of this paper was to introduce and demonstrate the new mathematical formulation. As a consequence, we did not focus yet on improving the computation time. This may be accomplished by the development of a customized algorithm, which is the aim in further research.

Acknowledgements This work is supported by Airbus Defence and Space, Germany. We are grateful to Airbus Defense and Space employees Kai Kistorz and Dr. Axel Luthardt for their valuable feedback. Moreover, our gratitude goes to the anonymous reviewers for their insightful comments.

References

1. Koopman, B.O.: Search and Screening: General Principles with Historical Applications. Pergamon Press, New York (1980)
2. Stewart, T.: Search for a moving target when searcher motion is restricted. *Comput. Oper. Res.* **6**(3), 129–140 (1979)

3. Martins, G.H.A.: A new branch and bound procedure for computing optimal search paths. Master's thesis, Naval Postgraduate School, Monterey (1993)
4. Lau, H., Huang, S., Dissanayake, G.: Discounted MEAN bound for the optimal searcher path problem with non-uniform travel times. *Eur. J. Oper. Res.* **190**(2), 383–397 (2008)
5. Sato, H.: Path optimization for single and multiple searchers: models and algorithms. Ph.D. thesis (2008)
6. Morin, M., Lamontagne, L., Abi-zeid, I., Lang, P., Maupin, P.: The optimal searcher path problem with a visibility criterion in discrete time and space. In: Proceedings of the 12th International Conference on Information Fusion, pp. 2217–2224. (2009)
7. Bourgault, F., Furukawa, T., Durrant-Whyte, H.: Optimal search for a lost target in a bayesian world. In: Yuta, S., Asama, H., Prassler, E., Tsubouchi, T., Thrun, S. (eds). *Field and Service Robotics*. Springer Tracts in Advanced Robotics, vol. 24, pp. 209–222. Springer, Berlin Heidelberg (2006)
8. Foraker, J., Royset, J.O., Kaminer, I.: Search-trajectory optimization: Part I, formulation and theory. *J. Optim. Theory Appl.* **169**(2), 530–549 (2016)
9. Foraker, J., Royset, J.O., Kaminer, I.: Search-trajectory optimization: Part II, algorithms and computations. *J. Optim. Theory Appl.* **169**(2), 550–567 (2016)
10. Iida, K., Hohzaki, R., Inada, K.: Optimal survivor search for a target with conditionally deterministic motion under reward criterion. *J. Oper. Res. Soc. Japan* **41**(2), 246–260 (1998)
11. Stone, L.D., Richardson, H.R.: Search for targets with conditionally deterministic motion. *SIAM J. Appl. Math.* **27**(2), 239–255 (1974)
12. Furukawa, T., Bourgault, F., Lavis, B., Durrant-Whyte, H.: Recursive bayesian search-and-tracking using coordinated UAVs for lost targets. In: Proceedings of the 2006 IEEE International Conference on Robotics and Automation, pp. 2521–2526. (2006)
13. West, D.B., et al.: Introduction to Graph Theory, vol. 2. Prentice Hall, Upper Saddle River (2001)
14. Khuller, S., Moss, A., Naor, J.S.: The budgeted maximum coverage problem. *Inf. Process. Lett.* **70**(1), 39–45 (1999)
15. Baxter, J., Burke, E., Garibaldi, J., Norman, M.: Multi-robot search and rescue: a potential field based approach. In: Mukhopadhyay, S.C., Gupta, G.S. (eds). *Autonomous Robots and Agents, Studies in Computational Intelligence*, vol. 76, chap. 2, pp. 9–16. Springer, Berlin Heidelberg (2007)
16. Xiao, X., Dong, Z., Wu, J., Duan, H.: A cooperative approach to multiple UAVs searching for moving targets based on a hybrid of virtual force and receding horizon. In: INDIN, pp. 1228–1233. IEEE (2012)
17. Dell, R.F., Eagle, J.N., Alves Martins, G.H., Santos, A.G.: Using multiple searchers in constrained-path, moving-target search problems. *Naval Res. Logist.* **43**(4), 463–480 (1996)
18. Eagle, J.N., Yee, J.R.: An optimal branch-and-bound procedure for the constrained path, moving target search problem. *Oper. Res.* **38**(1), 110–114 (1990)