Target search with an allocation of search effort to overlapping cones of observation

Hugo Vaillaud, Claire Hanen, Emmanuel Hyon, Cyrille Enderli

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 801–811
DOI: 10.15439/2023F7181
ISSN 2300-5963 ACSIS, Vol. 35

Target search with an allocation of search effort to overlapping cones of observation

Hugo Vaillaud
Thales DMS
Sorbonne Université, CNRS, LIP6
Paris, France
hugo.vaillaud@fr.thalesgroup.com

Claire Hanen
Sorbonne Université, CNRS, LIP6
UPL, Université Paris Nanterre
Paris, France
claire.hanen@lip6.fr

Emmanuel Hyon
Sorbonne Université, CNRS, LIP6
UPL, Université Paris Nanterre
Paris, France
emmanuel.hyon@lip6.fr

Cyrille Enderli
Thales DMS
Elancourt, France
cyrille-jean.enderli@fr.thalesgroup.com

***Abstract*—This paper addresses the problem of an aerial moving target search with a radar on an airborne platform. An observation of the radar is modeled as a cone covering a set of regions of the search area. We assume overlapping cones of observation, and we want to find the discrete allocation plan of search effort to the cones in order to optimize target detection. For the stationary target search with overlapping cones, we present a dynamic programming algorithm that computes the optimal allocation. An approximate greedy heuristic, which is more appropriate in a real time context, is also presented and assessed. The moving target search problem is solved with the Forward And Backward (FAB) algorithm coupled with the different stationary search algorithms. In this paper, we use a radar detection model that has been shown to be more realistic than the ones usually considered. Also, several models of movement of the target are considered with different Markovian transition matrices. We compare the performance of the mentioned algorithms on several scenarios.**

***Index Terms*—Moving Target search, Dynamic Programming, FAB algorithm, Overlapping Observation Cones.**

I. Introduction

SEARCHING for a moving target with a sensor is a complex problem, especially when the search area is large. This has been the subject of much research in the past, as evidenced in literature reviews by Stone [1] and by Rapp [2].

In general, one seeks to optimize an objective related to the probability of detecting one or more moving targets. The search area is discretized into identical regions and a sensor plan is computed over a fixed horizon. All regions are associated with prior probabilities of presence of targets. The probability of detecting targets in a region depends on the amount of search effort invested in it, the visibility of the region, and the probability that targets are present in the region.

Depending on the type of the platform (drone, satellite) or sensor (radar, camera), several approaches exist (see the review in [2] and references included therein). One can try to solve a *path constrained search*, where the subset of regions that will be observed and the order of visit of these consecutive regions must be determined (see [3], [4]). One can instead try to find the quantities of effort (discrete or continuous), constrained by a budget, to allocate independently to each region, at each time step, in order to maximize the probability of detecting the targets over the time horizon considered [1], [5], [6]. The search plan can be computed using the Forward-and-Backward algorithm (FAB) [7], [5], [6], Branch and Bound methods [8], or MILP [9] among others (see [2]).

In the problem we address, the search effort cannot be allocated independently to each region. Indeed, each observation of the radar is characterized by a *cone* that covers a set of regions. Moreover we assume that the cones may overlap, so that a region can be covered by several observation cones. This model applies for 2D as well as for 3D dimensional cones. As pointed out in chapter 14 of [10], an important parameter of the search plan design is *beam search spacing*, that defines the maximum number of consecutive overlapping cones. In this paper, we investigate its role in the efficiency of search plans and algorithms. We consider here discrete search effort on each cone which means we deal with the number of observations.

In [11], the authors consider a similar context for a continuous search effort and propose a customized version of the Forward-And-Backward algorithm to solve it, without proving its optimality. Path search have been considered in [9] where several regions are observed at the same time from each position and a region can be observed from several positions. The solution proposed is a mixed integer programming method. In [12], the authors addressed the problem of a moving target search with a radar where the cones of observation are disjoint: each region is covered by a single cone of observation. They show that for continuous effort the FAB algorithm is still optimal, and adapt the FAB algorithm to the discrete effort case. None of these works address the problem of allocating a discrete search effort to overlapping cones.

Another relevant issue in target search is the detection model. Chapter 2 of [13] presents an extensive literature review on detection models and quotes that in most of the papers the visibility of a region depends only on the distance from the radar. However, one of the main characteristics

of radar measurements is that the detection probability also decreases on the borders of an observation cone. Such a model was first proposed in [11]. In this paper we consider a general model where the detection probability of a single observation depends arbitrarily on the region and on the observation cone.

In this work, we address a moving target search with a radar by considering overlapping cones of observation for the allocation of discrete search effort and with a new detection model. Unfortunately, in this case the usual algorithm to optimize the probability of finding a stationary target given by [1] is not optimal. Thus, we face up a combinatorial optimisation problem to compute the optimal quantity of effort to allocate at each time step.

We present a dynamic programming algorithm that finds the optimal allocation of effort for a stationary target search case with overlapping cones of observation. Operational considerations require a short execution time of the algorithm. Therefore, we study different computational heuristics. Among them we propose a greedy algorithm which approximates efficiently the optimal solution. All stationary algorithms are then used in a FAB algorithm to solve the moving target search. We then present numerical experiments to compare the performance of these different algorithms in different scenarios. Among them we consider different models of target movement.

This paper is organized as follows. Section II details the framework of the target search. In Section III we present the dynamic programming algorithm that finds the optimal allocation for a stationary target search while Section IV is devoted to heuristics and an upper bound. We recall the moving target search problem and the FAB algorithm in part V. At last Section VI focuses on the numerical experiments.

II. Framework

The search problem that we consider is the following:

a) the target: An airborne platform faces a search area, which is partitioned into J regions. We assume a single target. The target position is unknown but we assume that we know its movement model. Similarly as in previous works, namely [1], the target moves from one region to another, at each time step $t \in \{0, \dots, T\}$, following a Markovian transition model with some transition function $\pi_t(i, j)$ defined for all pair of regions $i, j \in \{1, \dots, J\}$ and $t = 0, \dots, T$. As soon as we know the prior probability $p_0(j)$ that the target is initially located in a region j, we deduce $p_t(j)$ which is the probability that the target is in region j at time t. A trajectory is a sequence of regions $\omega = (\omega_t)_{t=0,\dots,T}$.

b) the sensor: We also assume a single sensor that observes the search area such that at each time step, it has a budget of observations that it can allocate to one or several angles. An observation made by the sensor in an angle encompasses all the regions covered by the cone defined by the angle and the position of the sensor. We define M_a as the set of regions observed in angle a (see Figure 1) and conversely by Y_j the set of angles covering region j. We say that two angles a, a' overlap if $M_a \cap M_{a'} \neq \emptyset$.

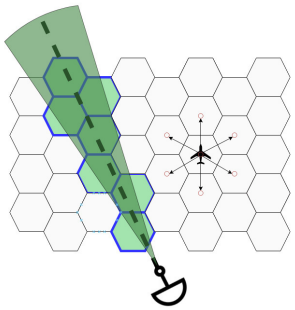

Fig. 1. A single cone of observation and the regions being observed.

We assume that the set of angles A is indexed such that each region of the search area is observable from at least one angle and from at most N consecutive angles. In Figure 2, we present three examples for $N = 1, 2, 3$. The set A is indexed from left to right. In each example, there are four cones of observation and a region indicated by a red circle. The cones that cover the region are represented in green such that an area covered by several green cones appears darker. The other cones of observation are represented in white. For $N = 1$, the search area is partitioned into disjoint cones of observation. For $N = 2$ the red circle is covered by angles 2 and 3, and for $N = 3$, the red circle is covered by angles $1, 2, 3$.

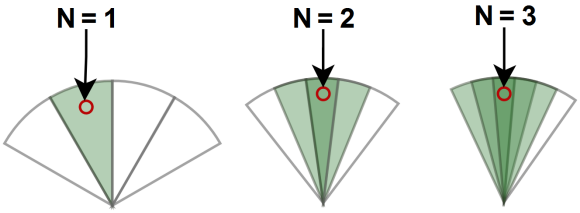

Fig. 2. Example of overlapping cones of observation.

c) the search plan: We define the search plan Z as a matrix of shape $|A| \times (T + 1)$. It defines at each time step $t = 0, \dots, T$, an allocation vector $\zeta(t)$ whose components $\zeta_a(t)$ define the number of observations made in the angle a at time t for $a \in A$.

We define $\kappa(a) \in \mathbb{R}^+, \kappa(a) \geq 1$ the cost of investing an effort to the angle a. The cost of an allocation vector $\zeta(t)$ is

$$K(\zeta(t)) = \sum_{a \in A} \zeta_a(t) \cdot \kappa(a)\,.$$

At each time step, we construct an allocation vector $\zeta(t)$ such that its cost is bounded by a budget C_t.

d) The detection model: The range of a radar is not uniform in an observation cone: it is at its highest in the center of the cone and decreases on the borders. In our model, we assume that a region-angle couple (j, a) such that $a \in Y_j$ is associated with a visibility coefficient $\alpha_{j,a} \in [0, 1]$ representing the conditional probability of detecting the target in the region j when the sensor makes one observation in the

angle a, provided that the target is in the region j. We also assume that the observations are independent.

We call this model "*realistic*": the coefficient depends on the distance of the region to the radar and the angular offset to the center of the observation cone. We also consider in our experiments the usual model where the coefficient only depends on the distance. It is called "*distance*" model. In this case $\alpha_{j,a} = \alpha_j$ for all $a \in Y_j$. Figure 3 illustrates these two models.

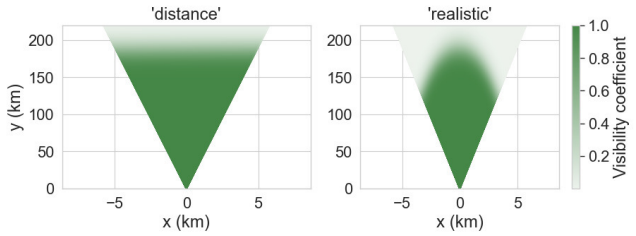

Fig. 3. Two types of detection models in a single cone of observation.

Consider a single time step t of the plan and the allocation of effort vector ζ at this time step (t is omitted in the notation). The detection function $b(j, \zeta)$ is the conditional probability of detecting the target in a region j for an effort vector ζ given that the target is located in j. It can be expressed by the following property:

Property 1 (Detection function)**.**

$$b(j, \zeta) = 1 - \prod_{a \in Y_j} (1 - \alpha_{j,a})^{\zeta_a} . \tag{1}$$

Proof. Let D_j and $\bar{D}_j$ be respectively the events associated with a detection and a non detection of the target in the region j. Let C_j be the event associated with the target located in region j. By definition, The detection function in a region is defined as: $b(j, \zeta) = \mathbb{P}(D_j|C_j) = 1 - \mathbb{P}(\bar{D}_j|C_j)$.

Now, we know that for each observation of j in angle a, the probability of missing the target, given it is present, is $1-\alpha_{j,a}$.

So, if ζ_a independent observations are made in angle a, the conditional probability of missing the target in region j, given C_j is $(1 - \alpha_{j,a})^{\zeta_a}$.

Finally, as region j is covered by all angles $a \in Y_j$, and as all observations are independent, the probability $\mathbb{P}(\bar{D}_j|C_j)$ of missing the target, given it is in region j, is $\prod_{a \in Y_j}(1-\alpha_{j,a})^{\zeta_a}$ so that the property hold.

□

When we use the "distance" model, since the value of $\alpha_{j,a}$ is the same for any a in Y_j, then the formula is simplified to

$$b(j, \zeta(t)) = 1 - (1 - \alpha_j)^{\sum_{a \in Y_j} \zeta_a(t)} . \tag{2}$$

e) the objective function: Let $P_t(Z, \omega)$ be the conditional probability of detecting the target before the time horizon t given the target follows the trajectory ω and the allocation plan is Z. $P_t(Z, \omega)$ is defined as follows:

$$P_t(Z, \omega) = 1 - \prod_{s=0}^{t} \big(1 - b(\omega_s, \zeta(s))\big) . \tag{3}$$

As in [1], we consider a generic objective function composed of any linear combination of such probabilities:

$$\hat{P}_T(Z) = \mathbb{E}\left[\gamma_{T+1}(\omega) + \sum_{t=0}^{T} \gamma_t(\omega) \cdot P_t(Z, \omega)\right] . \tag{4}$$

Two remarkable objectives can be formulated by adjusting the coefficients γ_t:

- The maximization of the probability $P_T(Z)$ of detecting the target before the time horizon T by executing the allocation plan Z: ($\gamma_t = 0$ for $t = 0, \dots, T-1$, $\gamma_T = 1$ and $\gamma_{T+1} = 0$).
- The minimization of the mean completion time $M_T(Z)$ (i.e. the expected number of time steps until the target is detected or the time horizon T is reached). By setting $\gamma_t = 1$ for $t = 0, \dots, T$ and $\gamma_{T+1} = -(T + 1)$, we get the objective $-M_T(Z)$ to be maximized.

The global objective of the planning algorithm is thus to find the feasible plan (satisfying the budget constraint) Z^* that maximizes $\hat{P}_T(Z)$.

III. Optimal stationary target search

The Forward and Backward algorithm decomposes the moving target search problem into successive resolutions of stationary target search problems at each time step. Hence, we first consider the problem of stationary target search. The aim is to minimize the probability of missing the target at a single time step t, given the probabilities of the target presence in every region, and a budget C for the observations.

For the sake of clarity, in this section we do not use the time step t in our notations and $\zeta_a(t)$ is denoted by ζ_a. We present a dynamic programming algorithm to compute the optimal allocation of effort to angles bounded by a budget C for the search of a stationary target when the cones of observation overlap.

The stationary target search problem can be formulated as a convex mathematical program:

$$\begin{aligned} \min \quad & 1 - P(\zeta) = \sum_{j=1}^{J} p(j) \cdot \prod_{a \in Y_j} (1 - \alpha_{j,a})^{\zeta_a} \\ \text{s.t.} \quad & \sum_{a \in A} \kappa(a) \cdot \zeta_a \leq C \ \text{ and } \ \zeta_a \in \mathbb{N}, \forall a \in A . \end{aligned} \tag{5}$$

To our knowledge, the complexity of this problem is unknown, even in the simpler case considered in [1] where the effort applies independently to regions and assuming unit costs. The complexity of the algorithm presented in [1] depends linearly on the maximal number of observations, which is constrained by the budget, and thus is pseudo-polynomial. However from a practical point of view, in the operational instances the maximum number of observations is usually far lower than the number of regions, or cones in our case which makes the algorithm polynomial.

A. Properties

The algorithm uses the fact that at most N consecutive cones of observation overlap (as illustrated in Figure 2) to decompose the problem into elementary subproblems and computes the optimal solution iteratively by considering consecutive angles.

We consider the tuple $(a_k)_{k=1,\dots,|A|}$ of the angles of A in increasing order. For each phase $k \in \{1,\dots,|A|\}$ of the algorithm, we define the set of regions R_k that are observable from angle a_k but not from any angle $a_{k'}$ with $k' > k$.

For example in Figure 2, when $N = 2$, there are 4 cones of observations associated to the angles $(a_k)_{k=1,\dots,4}$. The red circle is covered by the cones a_2 and a_3 but not by a_4. The circle belongs to R_3. Note that in an operational instances the number of angles is far greater than 4.

Property 2. *The sets $(R_k)_{k=1,\dots,|A|}$ form a partition of the set of regions.*

Note that, since the R_k are disjoint, Property 2 implies that the probability of detecting the target $P(\zeta)$ (given by Eq. (5)) can be computed by summing the probabilities of detecting the target in each R_k, for $k = 1,\dots,|A|$.

Let us denote by $n_k = \min(N, k)$ the maximum number of angles that can observe a region in R_k. Notice that for $k = |A|$, R_k contains all regions observable by angle a_k, so there might be a region in R_k covered by N angles.

Property 3. *A cone M_a intersects R_k if and only if $a \in \{a_{k-n_k+1},\dots,a_k\}$.*

Property 3 implies that the probability of missing the target in R_k can be determined using the allocation of effort on $\{a_{k-n_k+1},\dots,a_k\}$, which can be expressed as a list of n_k values: $\zeta_{a_k},\dots,\zeta_{a_{k-n_k+1}}$ for $k = 1,\dots,|A|$.

We define $W_k(\zeta_{a_k},\dots,\zeta_{a_{k-n_k+1}})$ the probability of missing the target in R_k as:

$$W_k(\zeta_{a_k},\dots,\zeta_{a_{k-n_k+1}}) = \sum_{j\in R_k} p(j) \prod_{l=k-n_k+1}^{k} (1-\alpha_{j,a_l})^{\zeta_{a_l}}\,. \tag{6}$$

Let us define the partial objective of order k as:

$$Part_k(\zeta) = \sum_{l=1}^{k} W_l(\zeta_{a_l},\dots,\zeta_{a_{l-n_l+1}})\,. \tag{7}$$

The partial objective $Part_k(\zeta)$ is thus the part of the global objective (5) that excludes the part of the area covered by the cones of the angles $a_{k+1},\dots,a_{|A|}$. Notice that our objective is then:

$$1 - P(\zeta) = Part_{|A|}(\zeta) \tag{8}$$

B. The algorithm

We now define the dynamic programming scheme to minimize $Part_{|A|}(\zeta)$, where each phase k corresponds to the decision concerning the effort ζ_{a_k}. The $n_k - 1$ past decisions at phase k are summarized into a state $E = (B, x)$ where:

- B is an integer between 0 and C representing the remaining budget available for the decisions of the phases k to $|A|$;
- x is a list of $n_k - 1$ elements representing the last decisions made in reverse chronological order (thus concerning the angles $a_{k-1},\dots,a_{k-n_k+1}$).

In this context, the initial state is $(C, null)$ (where $null$ represents an empty list). If x is a list of values we denote by $x[p]$ the p^{th} element of the list, and by $x[p\dots q]$ the sub list of x with elements indexed from p to q. Moreover if u is an element or another list we denote by $x \cdot u$ the concatenation of x and u.

The transitions of the dynamic programming scheme are defined as follows:

If at phase k the system is in state $E = (B, x)$, the decision δ (effort on angle a_k) satisfies $\delta \in \{0,\dots,\left\lfloor \frac{B}{\kappa(a_k)} \right\rfloor\}$. For such a decision δ, we define $x' = \delta \cdot x[1\dots n_k - 2]$, $\delta' = x[n_k - 1]$, and $B' = B - \delta\kappa(a_k)$. The pair (B', x') is the resulting state of the decision δ at phase $k+1$. Notice that $x = x'[2\dots n_k - 1]\cdot\delta'$, and $B = B' + x'[1]\kappa(a_k)$. This will be used in Property 4.

The immediate cost of the decision δ is then $W_k(\delta\cdot x)$. Using the previous notations, we observe that $W_k(\delta\cdot x) = W_k(x'\cdot\delta')$.

Let $F_k(B, x)$ be the minimal value of $Part_{k-1}(\zeta)$ where ζ is subject to the two constraints:

$$\sum_{l=1}^{k-1} \zeta_{a_l} \le C - B\,,$$
$$\zeta_{a_{k-1}},\dots,\zeta_{a_{k-n_k+1}} = x\,.$$

We can now express the optimal objective of the stationary search as follows:

$$\min_x F_{|A|+1}(0, x) \tag{9}$$

Property 4. *The values $F_k(B, x)$ satisfy the following recurrence equation:*

$$F_{k+1}(B', x') = \begin{cases} F_k(B' + x'[1]\kappa(a_k), x = x'[2\dots n_k - 1]) + W_k(x') \text{ if } k < N \\ \min\limits_{0\le\delta'\le C'} \Big(F_k\big(B' + x'[1]\kappa(a_k), x = x'[2\dots n_k - 1]\cdot\delta'\big) + W_k(x'\cdot\delta')\Big) \\ otherwise. \end{cases} \tag{10}$$

where $C' = \left\lfloor \frac{C - B' - \sum_{i=1}^{n_{k+1}-1} x'[i]\kappa(a_{k+1-i})}{\kappa(a_{k-n_{k+1}})} \right\rfloor$.

Proof. Let ζ be the optimal allocation solution of $F_{k+1}(B', x')$. By the decomposition of the values $Part_k$ as a sum of W_l values in (7), we can see that if $k < N$, $n_{k+1} = n_k + 1$ and $x' = \zeta_{a_k},\dots,\zeta_1$ contains the effort of all the first angles that led to a remaining budget B'. So for earlier decisions, all the allocation of effort is know and thus $F_{k+1}(B', x') = Part_k(\zeta) = W_k(x') + F_k(B' + x'[1]\kappa(a_k), x = x'[2\dots n_k - 1])$.

Assume that $k > N$, so that $n_{k+1} = n_k = N$. Setting $\delta' = \zeta_{a_{k-n_k+1}}$, we have $x'\cdot\delta' = \zeta_{a_k},\dots,\zeta_{a_{k-n_k+1}}$. Setting $x = x'[2\dots n_k - 1]\cdot\delta'$, the allocation ζ considered for angles a_1

to a_{k-1} defines a feasible solution of the sub problem whose optimal value is $F_k(B' + \kappa(a_k)x'[1], x)$. So $F_{k+1}(B', x') = Part_{k-1}(\zeta) + W_k(x' \cdot \delta')$ is not less than the right hand side of the equality.

Conversely, consider any $\delta' \leq C'$. Set $x = x'[2 \dots n_k - 1] \cdot \delta'$. Let ζ' be the optimal plan solution of $F_k(B' + \kappa(a_k)x'[1], x)$. We can define the plan ζ'' such that $\zeta''_{a_l} = \zeta'_{a_l}$ if $l < k$, and $\zeta''_{a_k} = x'[1]$, and obtain a feasible solution of the sub problem for which $F_{k+1}(B', x')$ is the optimal value. So we get:

$$\begin{aligned} F_{k+1}(B', x') &\leq Part_k(\zeta'') = Part_{k-1}(\zeta') + W_k(x' \cdot \delta') \\ &= F_k(B' + \kappa(a_k) \cdot x'[1], x) + W_k(x' \cdot \delta'). \quad (11) \end{aligned}$$

Hence the reverse inequality holds. □

The forward dynamic programming algorithm issued from Property 4 computes for each phase k from 1 to $|A|$ and for each possible state (B', x') of this phase the value $F_k(B', x')$. Algorithm 1 shows this first part that computes the minimal target missing probability. The computation of the optimal effort allocation ζ is then done using classic dynamic programming approach by searching among the stored values $F_k(B', x')$ which decisions gave the optimal value, with a lower complexity.

Algorithm 1 DP algorithm

```
for all B ≤ C do
    F_1(B, null) = 1
end for
for k = 2 to |A| + 1 do
    for all possible couples (B', x'), with length(x') = n_k − 1 do
        Compute F_k(B', x') using equation (10)
    end for
end for
return min_x F_{|A|+1}(0, x)
```

We can now analyze the complexity of Algorithm 1.

Property 5. *The time complexity of Algorithm 1 is:*

$$\mathcal{O}\left(|A| \cdot N \cdot J \cdot C^N\right).$$

It is pseudo-polynomial for fixed N.

Proof. Let us denote by $z_{max} = \left\lfloor \frac{C}{\min_{a \in A} \kappa(a)} \right\rfloor$ the maximum number of observations in an angle. For each iteration of the outer loop, $k \in \{2, \dots, |A| + 1\}$ a state (B', x') is composed of the remaining budget and the at most $N - 1$ last decisions composing x'. The space needed to store each value $F_k(B', x')$ is therefore $\mathcal{O}(|A| \cdot C \cdot z_{max}^{N-1})$. Now for each $F_k(B', x')$, the time complexity of the computation of the recurrence equation (10) depends on the number of different δ' values, which can be bounded by z_{max}. For each δ' the computation of $W_k(x'.\delta')$ is in $\mathcal{O}(N \cdot |R_k|)$. As $\min_{a \in A} \kappa(a) \geq 1$, $z_{max} \leq C$. This gives the whole time complexity of the algorithm. □

Notice that in real applications, N is usually quite small and ranges from 2 to 6 (in the 3D case), and the budget and z_{max} are usually far lower than J, which makes this approach tractable in practice.

IV. Heuristics and bounds

This section is devoted to the presentation of some heuristics methods and a relaxation for the stationary target search.

A. Greedy algorithm

We first present an iterative algorithm that approximates the optimal search effort allocation. This algorithm is an adaptation of the optimal greedy algorithm for the computation of the allocation of discrete effort in the case of disjoint cones of observation and unit costs which was proposed in [12]. Due to the overlapping of the observation cones, the conditions that ensured its optimality vanish. However our experiments, in section VI, shown that it provides a fast and high quality approximation of the optimal solution. We first detail the concept of rate of return of angles on which the algorithm is based and we then present the complete algorithm.

1) Rate of return of angles: Let us first define $p^{(ang)}(a)$ as the probability that the target is in the cone associated with the angle a. We have

$$p^{(ang)}(a) = \sum_{j \in M_a} p(j). \quad (12)$$

We define $b'(j, \zeta, a)$ as the *rate of change* of the function $b(j, \zeta)$ for an additional effort in the angle a by:

$$\begin{aligned} b'(j, \zeta, a) &= b(j, \zeta + \mathbf{1_a}) - b(j, \zeta) \\ &= \alpha_{j,a} \prod_{a' \in Y_j} (1 - \alpha_{j,a'})^{\zeta_{a'}}. \quad (13) \end{aligned}$$

Where $b(j, \zeta)$ is defined in Equation (1) and where $\mathbf{1_a}$ is the unit vector of length A such that $\mathbf{1_a}(\mathbf{a}) = \mathbf{1}$ and $\mathbf{1_a}(\mathbf{a'}) = \mathbf{0}$ for $a' \neq a$. As it can be seen in the Equation (13), the value of $b'(j, \zeta, a)$ depends not only on angle a but also on all the angles that cover the region j.

We define the *angular detection function* $\beta(a, \zeta)$ as the conditional probability of detecting the target in one of the regions in the cone associated with an angle a, for an allocation of effort ζ on angles, given the target is located in the cone. The angular detection function $\beta(a, \zeta)$ is:

$$\beta(a, \zeta) = \frac{\sum_{j \in M_a} p(j) \cdot b(j, \zeta)}{p^{(ang)}(a)}.$$

We define β' as the *angular rate of change* of the function β for an additional effort in angle a by:

$$\beta'(a, \zeta) = \frac{\sum_{j \in M_a} p(j) \cdot b'(j, \zeta, a)}{p^{(ang)}(a)}.$$

We then define $\rho(a, \zeta)$ as the *rate of return* function. It represents, for an allocation ζ, the ratio between the increase of the probability of detection for a new increment of effort

in the angle a, and the investment cost generated by the same increment. It equals:

$$\rho(a,\zeta) = \frac{p^{(ang)}(a)\beta'(a,\zeta)}{\kappa(a)} = \frac{\sum_{j\in M_a} p(j)\cdot b'(j,\zeta,a)}{\kappa(a)}. \tag{14}$$

In the overlapping cones case, the value of the *rate of return* $\rho(a,\zeta)$ depends not only on the value of effort ζ_a but also on the allocation of effort ζ to all the angles that cover any region j covered by a. Algorithm 2 summarizes the main steps. Starting from a null allocation of effort, at each iteration an additional unit of effort is added in the angle with the best rate of return until the budget is reached (as we assumed costs not less than 1, there are at most C such iterations). To compute or update the rate of returns, each region j appears at most N times in the sums, and there are at most N terms in the computation of $b'(j,\zeta,a)$. The complexity of this algorithm is thus $\mathcal{O}(C\cdot|A|\cdot J\cdot N^2)$.

Algorithm 2 Greedy algorithm

$\zeta = (0,\dots,0), B = 0$, and Compute $\rho(a,\zeta), \forall a \in A$
while $\exists a \in A$ such that $B+\kappa(a) \le C$ **do**
  Compute $a* = \underset{a\in A, B+\kappa(a)\le C}{argmax}\ (\rho(a,\zeta))$
  $\zeta = \zeta + \mathbf{1}_{a^*}$
  $B = B + \kappa(a^*)$
  Update $\rho(a,\zeta)$ for all a such that $M_a \cap M_{a^*} \neq \emptyset$
end while

In the non-overlapping cones case and unit costs, the optimisation problem can be exactly solved using this algorithm [12]. In our case this result does not hold. Hence, our heuristic is suboptimal.

B. Adapted Random Permutation Scan Method

We now introduce an algorithm adapted from the conventional *random permutation scan* method as described in [14], referred to as *RPSM* in the following. The original strategy entails a comprehensive sweep of the entire search area in the absence of any prior information about the target's position. During each iteration, the radar conducts an observation in each considered direction following a random order. The algorithm stops when the budget of observations runs out.

In this study, we have adapted this method to scenarios where there is prior knowledge about the target's location. The set of considered angles corresponds to those that cover at least one region j such that $p(j) > 0$ thus we consider angles such that $p^{(ang)}(a) > 0$.

a) The algorithm: A buffer H is used to store the angles that have not been observed yet. The buffer is initialized with the set of angles covering at least one region with a positive prior on target presence (*i.e.* $p^{(ang)}(a) > 0$). When the buffer is empty, it is reinitialized with the same set of angles.

The markovian transition matrix is used to compute the probability of presence of the target at each timestep. Hence at time step t, the probability $p_t(j)$ is computed recursively from the prior probability p_0 and the transition matrices π_t of the Markov chain. The set of angles covering at least one region with a positive prior on target presence is updated consequently at each timestep. An allocation plan is initialised to a null plan and allocations are computed sequentially with Algorithm 3. When the buffer is empty, it is updated with $\{a \in A : p_t^{(ang)}(a) > 0\}$.

Algorithm 3 RPSM adapted to target search with priors

$\zeta = (0,\dots,0)$, $B = 0$
$H = \{a \in A : p^{(ang)}(a) > 0\}$
while $\exists a \in H, B+\kappa(a) \le C$ **do**
  Sample a from H without replacement
  if $B+\kappa(a) \le C$ **then**
    $\zeta = \zeta + \mathbf{1}_a$
    $B = B + \kappa(a)$
  end if
  if $H = \emptyset$ **then**
    $H = \{a \in A : p^{(ang)}(a) > 0\}$
  end if
end while

This algorithm is used as a baseline for the search of a stationary and moving target.

C. Relaxation of the problem for the computation of an upper bound

For the moving target search problem, the FAB algorithm is guaranteed to converge to the optimal plan when the search effort is continuous but not when it is discrete (see [1]). As in [12], we use the value of the optimal payoff obtained in the continuous case as an upper bound on the optimal value of the payoff obtainable in the discrete case. The aim is then to measure the efficiency of the discrete approach with respect to the bound given by the continuous optimal solution.

The detection function and the objective functions (stationary and moving target) have the same formulation both in discrete and continuous cases. These functions can thus be used to compute the optimal allocation of effort in the continuous case.

Computing the optimal allocation of continuous effort for a stationary target search is a quite difficult convex optimization problem. Henceforth, we propose here to further relax the problem in order to compute an allocation on disjoint angles that provides a simpler upper bound for the optimal probability of detection on this problem. This relaxation requires that $\kappa(a) = 1\ \forall a \in A$.

The relaxed instance I' is defined by splitting each angle a into N disjoint cones of same area: $u_1(a),\dots,u_N(a)$. The budget of I' is $C' = C\cdot N$. The visibility coefficients are such that for each region $j = 1,\dots,J$, if a' is an angle of instance I', then $\alpha'_{j,a'} = \alpha_j = \max_{a\in Y_j}\alpha_{j,a}$.

Property 6. *A continuous allocation ζ for the original instance I defines a feasible continuous allocation ζ' for the relaxed instance I', by allocating ζ_a effort to each of the N cones issued from angle a. If a' is an angle of I' then: $\zeta'_{a'} = \sum_{a,\exists i, u_i(a)=a'}\zeta_a$. Moreover $P(\zeta') \ge P(\zeta)$.*

Proof. The allocation ζ' is feasible for the relaxed instance I' because

$$\sum_{a' \in A'} {\zeta'}_{a'} = \sum_{a'} \sum_{a, \exists i, u_i(a)=a'} \zeta_a = \sum_a N \cdot \zeta_a \leq N \cdot C = C'.$$

Let us first observe that for a region j, since for any angle a, $\alpha_{j,a} \leq \alpha_j$,

$$\prod_{a \in Y_j} (1-\alpha_{j,a})^{\zeta_a} \geq \prod_{a \in Y_j} (1-\alpha_j)^{\zeta_a}.$$

Thus,

$$1 - \prod_{a \in Y_j} (1-\alpha_{j,a})^{\zeta_a} \leq 1 - \prod_{a \in Y_j} (1-\alpha_j)^{\zeta_a}. \quad (15)$$

Now consider the new instance I'. For each region j, there exists a unique angle $a'(j)$ that covers j (since in I' we have disjoint observation cones). Hence

$$\begin{aligned} P(\zeta') &= \sum_{j \in J} p(j) \left(1 - (1-\alpha_j)^{\zeta'_{a'(j)}}\right) \\ &= \sum_{j \in J} p(j) \left(1 - (1-\alpha_j)^{\sum_{a \in Y_j} \zeta_a}\right) \\ &= \sum_{j \in J} p(j) \Big(1 - \prod_{a \in Y_j} (1-\alpha_j)^{\zeta_a}\Big). \end{aligned}$$

From (15), we deduce that $P(\zeta') \geq P(\zeta)$. □

We can compute the optimal allocation of continuous effort on disjoint cones of observations for the relaxed instance I' with the algorithm presented in [12]. We can then use it to compute an upper bound on the optimal allocation of continuous effort to overlapping cones for the search of a moving target.

V. Moving target search

We now consider the moving target search problem with objective function given by Eq. (4). The mathematical formulation of this problem is:

$$\begin{aligned} \max \quad & \hat{P}_T(Z) \quad (16) \\ \text{s.t.} \quad & \sum_{a \in A} \zeta_a(t) \cdot \kappa(a) \leq C_t \text{ for } t = 0, \ldots, T\,. \\ & \zeta_a(t) \in \mathbb{N}\,, \forall a \in A, \text{ for } t = 0, \ldots, T\,. \end{aligned}$$

Here this problem is solved by using the FAB algorithm [7] which is a generalization of Brown's recursion [15]. Both algorithms adapted to an allocation to disjoint angles have been presented in [12].

Let us recall the principles of the FAB algorithm [1]. Starting from an initial plan Z_0, at each iteration (outer loop) the FAB algorithm computes a new plan, until two successive plans are equal, or the allowed number of iterations is reached.

In an iteration of the outer loop, starting from a plan Z, the computation of a new plan is done iteratively in an inner loop for each time step $t = 0, \ldots, T$. At a time-step t of the inner loop, the algorithm computes the values $q(j, Z, t)$ for each region j. To give an intuition, when the objective is the detection probability before horizon T, $q(j, Z, t)$ represents the probability that the target is in region j at time t and is not detected at any time other than t according to the current plan Z. The plan Z is then updated by computing an optimal stationary allocation for time t, using $q(j, Z, t)$ for $j = 1, \ldots, J$ as a prior probability distribution of target presence in the regions.

The computation of q values is done by using the markovian target movement, by decomposing q into a product of two functions R and S.

$$q(j, Z, t) = R(j, Z, t) \cdot S(j, Z, t)\,. \quad (17)$$

When the objective is the detection probability before horizon T, $R(j, Z, t)$ (resp. $S(j, Z, t)$) represents the probability that the target is in region j at time t and has been missed in time steps $t+1, \ldots, T$ (resp. $0, \ldots, t-1$) withs the plan Z. Those functions can be expressed with a recursive form thanks to the markovian property of the target movement (see details in [1]).

$$\begin{aligned} R(j, Z, t) = {} & E_{jt}[\gamma_t(\omega)] + \\ & \sum_{i=1}^{J} R(i, Z, t+1) \cdot \big(1 - b(i, \zeta(t+1)\big) \cdot \pi_t(j, i)\,. \quad (18) \end{aligned}$$

$$\begin{aligned} S(j, Z, t) = {} & \\ & \sum_{i=1}^{J} S(i, Z, t-1) \cdot \big(1 - b(i, \zeta(t-1)\big) \cdot \pi_{t-1}(i, j)\,. \quad (19) \end{aligned}$$

The value $E_{jt}[\gamma_t(\omega)]$ is the expectation of the coefficient γ_t of the objective function over trajectories such that the target is in region j at time t. Notice that it only depends on the target movement and does not depend on the plan, so that it can be pre-computed.

The initial values, for $j = 1, \ldots, J$, are $S(j, Z, 0) = p_0(j)$ and $R(j, Z, T) = E_{jT}[\gamma_T(\omega)]$.

Notice that in the inner loop of the FAB algorithm, the plan Z is updated iteratively for increasing time steps. So the future of the plan does not change between two consecutive iterations of the innerloop and thus $R(j, Z, t)$ according to (18) can be computed before the inner loop for each region j and each t. At each iteration t of the inner loop, for each region j, $S(j, Z, t)$ can be computed using (19) with the plan updated in the previous iteration.

If we start from a null plan, then the first iteration of FAB computes a myopic plan where the optimal stationary effort allocation of time t is computed from the target move probabilities at time t regardless of the allocation in the previous time steps. But the algorithm can be used with different initial plans.

Unlike the continuous effort case, the algorithm FAB for a discrete effort is not guaranteed to converge to the optimal plan. However, there exists a necessary condition of optimality.

Proposition 1 (From Theorem 3.4 [1])**.** *Assume we have a discrete-effort exponential detection function by region* $b(j, \zeta)$.

If Z is optimal, then for all t the allocation $\zeta(t)$ is an optimal allocation for the stationary target search with prior distribution q.

Therefore, finding, with dynamic programming, the optimal stationary plan at each step t ensures that the plan computed with FAB satisfies the necessary conditions which only guarantees that the returned plan is a local optimum.

In order obtain an optimal plan, we may consider a nonlinear integer program. Indeed, as stated in Section 3.3.2 of [1], when the effort is allocated in each region j independently, one can express the moving target search problem as a nonlinear integer program. However, even in this simple case, this is of a little theoretical and practical use since the objective function is computed by considering all the possible trajectories ω of the target. Hence, this does not scale to our problem with at least 1500 regions.

VI. Experimental results

In this section, we study the efficiency of the algorithms presented in the previous sections, in terms of solution quality and computational effort. We implemented the models and algorithms of the previous sections and designed different realistic scenarios, inducing our experimental parameters, and generated random instances according to these scenarios. The algorithms were implemented in Python and we ran scenarios on an two Intel Xeon E5-2690v3 (24 cores / 2.60 GHz).

Even though the algorithms presented in this paper could be used with a 3D search area, we assumed a stationary airborne platform facing a 2D search area discretized in regions spaced 10km apart, with a total of 1500 regions. The regions are arranged in a honeycomb pattern for an efficient use of 2D space. Using smaller distances between regions would improve the realism of the scenarios. However, it hugely increases the number of regions in the search area (e.g a distance of 1km gives 155000 regions) and then the computation time. In our study, we chose the parameters so that the dynamic programming algorithm could run in a reasonable time. We expect that the behaviour of the algorithms remains consistent for both small and large regions.

The target moves between two adjacent regions at each timestep. Considering the maximum relative speed of fighter jets, we consider a timestep to last roughly 7s in these experiments.

The platform has a radar that can make observations in the angular domain $[-60°; +60°]$. A radar "dwell" (name for an observation) in a direction affects the regions located in a cone of geometric height 250km and angular diameter 3 degrees. We consider set of angles such that they are equally spaced in the angular domain and each associated with an observation cone of angular width $3°$.

The maximum number of cones N covering a region is related to the angular separation between angles and the number of angles. When the angular separation between angles is $3°$ we have $N = 1$ and $|A| = 40$; if the angular separation is $1.5°$, $N = 2$ and $|A| = 79$. The radar is considered to use a single mode.

In the experiments, we assumed a fixed cost for making an observation in an angle a to $\kappa(a) = 1$.

In each scenario, an area of interest is initialized. The area of interest is the set of regions that have a nonzero probability of containing the target. This set is fully contained in the part of the search area that is visible from the radar.

The experiments use six parameters for the tested scenarios:

- **FOV_portion**: Determines the size of the area of interest as its portion of the total observable area. It can be 0.05, 0.5, or 1.0.
- **Horizon**: The number of time steps for the search. It can be 1, 2, 5, or 10.
- **Budget**: The number of observations that can be made during a time step. It can be 1, 2, 5, 10, 20, 40, or 50.
- **N**: The maximum number of consecutive cones covering a region (see Figure 2). It can be 1, 2 or 3. Higher values did not lead to significantly different results from what was observed at $N = 3$.
- **visibility**: Indicates the detection model used (see Figure 3) either "*distance*" or "*realistic*".
- **movement_type**: Defines the movement transition matrix. Can be "drone" or "jet". "drone" defines a uniform distribution over all possible direction. "jet" selects a random direction and then defines a distribution where there is 0.9 chance of going in this direction and 0.1 of remaining in the current region. The probability is 0 in all other directions. It approximates the movement model of a fighter jet that does not change direction.

A dwell is considered to last 100ms, allowing for a maximum of 70 observations per time step in different angles. We limited the number of observations to 50 per time steps and the horizon to 10, as larger values do not lead to significantly different results.

In the following experiments, we first focus on the problem of a stationary target search. We evaluate how using a detection model that depends only on the distance can negatively impact the performance of the allocation computed. Then, we evaluate the error of the probability obtained with the greedy algorithm compared with the optimal probability given by the dynamic programming algorithm. Then, we compare the probability of detection obtained with the dynamic programming algorithm with the baseline and the upper bound.

The second part experiments the moving target search. We show how increasing the maximum number N of observation cones covering a region of the search area affects the mean completion time of the plan. Also, we exhibit the performance of the algorithms FAB + greedy algorithm and FAB + dynamic programming for a fixed value of N in terms of the objective function and computation time. Finally, we show how the target movement model affects the mean completion time of the plan.

A. Stationary target search

1) Error caused by the use of the "distance" detection model: In the literature, the detection model used for the radar is often only a function of the distance to the radar. In this

section, we evaluate the error caused by the use of such a model compared to the "*realistic*" detection model.

To do so, we compute optimal allocations for stationary target search for the allocation of search effort on overlapping cones of observations such that $N = 3$ on different scenarios. We vary the budget and the FOV_portion. We first compute the optimal allocation for the "realistic" detection model and obtain the optimal probability of detection. Then, we compute the optimal allocation for the "distance" detection model and compute the detection probability of this last allocation by using the "realistic" detection model. We compute the mean absolute error of the probability of detection obtained with the "distance" detection model as compared to the optimal probability. We also show values ranging from the 5th to the 95th percentile of the error.

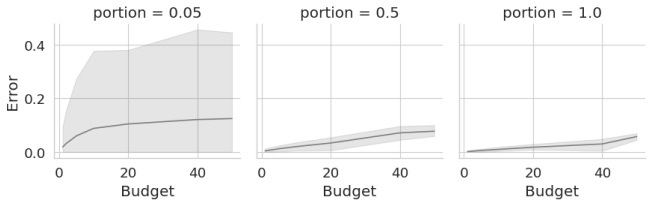

Fig. 4. Error caused by the use of the "distance" detection model.

In Figure 4 we see that the mean error is larger than 0.1 in many cases (especially when the area of interest is small) and the error can even achieve values above 0.4 for some scenarios for large values of budget. The use of the "distance" detection model therefore leads to a significant error in the probability of detection. We will use the more realistic detection model in the remaining experiments.

2) Comparison between the greedy algorithm and the dynamic programming algorithm: Algorithms 1 and 2 are both used as a module of the FAB algorithm, it is therefore interesting to study their behaviour in the simpler context of the stationary target search.

We ran both algorithms on different scenarios. We then computed the mean absolute error of the probability of detection obtained with the greedy algorithm as compared to the optimal probability obtained with dynamic programming.

Figure 5 depicts this error for small (left) to large (right) areas of interest w.r.t the budget of observations and parameter N. We computed the error only for $N = 2$ and 3 because using larger values requires amounts of memory unavailable to us to run the dynamic programming algorithm. The shaded areas around the curves represent the values between the 5th and the 95th percentile of the error.

We observe that the error is small. It is 0 in 81% of the simulations, rarely takes value above 0.02: the maximum error observed was 0.056. There seems to be a relationship between the size of the area of interest, the budget of observations and the error. For small areas of interest, there tends to be a higher error for small budgets. On the contrary for large areas of interest, there tends to be higher error for large budgets.

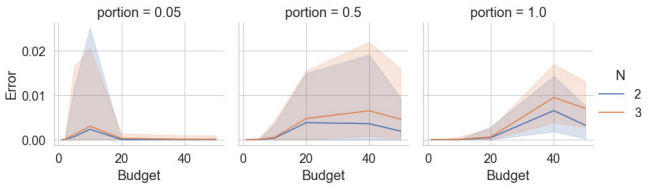

Fig. 5. Error of the greedy algorithm w.r.t. budget for different values of FOV_portion (columns), detection model (rows) and N (color).

Overall, we observed that the greedy algorithm is a good approximation of the optimal plan for the search of a stationary target. It also has a much lower space and time complexity, as shown in Section III-B.

3) Stationary target search: comparison between the optimal solution, RPSM and the relaxation: In Figure 6, we observe the optimal probability of detection computed with the dynamic programming algorithm w.r.t. budget, and we compare it with the one obtained with the random permutation scan method and the upper bound provided by the relaxation of the problem. We show the results for different sizes of the area of interest (columns) and different values of N the maximum number of cones that cover a region (rows). We show the results up to $N = 3$ because larger values do not lead to results significantly different from the ones obtained with $N = 3$.

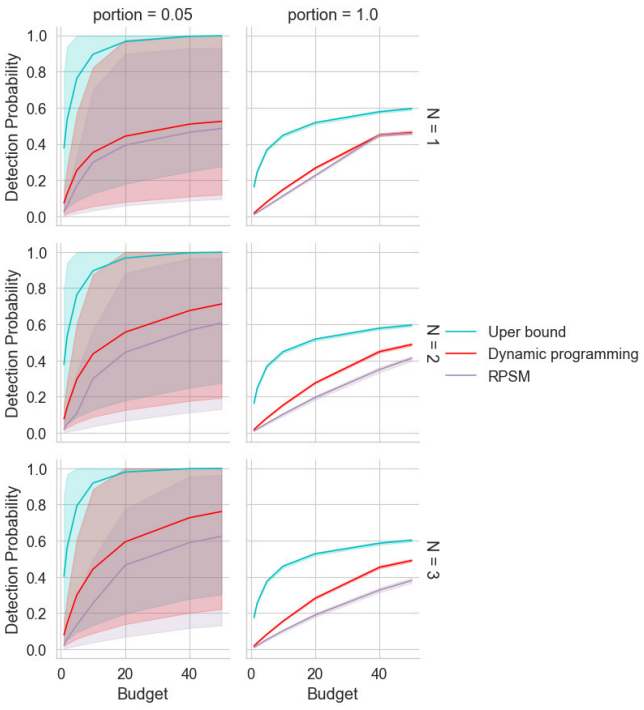

Fig. 6. Probability of detection obtained with three algorithms w.r.t. budget for different values of FOV_portion (columns) maximum number of cones that cover a region (rows).

First, we see that the upper bound provided by the relaxation is far larger than the optimal probability of detection. This

bound is not very informative. To get a more informative bound, we should solve the convex program corresponding to the relaxation with gradient techniques.

Our analysis demonstrates a correlation between the magnitude of N and the disparity in performance between the greedy algorithm and RPSM. When applied to a large area of interest, both algorithms exhibit similar performance for $N = 1$. An increment of N to 3 causes a slight enhancement in the greedy algorithm's performance by 4%, whereas RPSM's performance concurrently diminishes.

Conversely, when the area of interest is smaller, the performance difference between the two algorithms becomes greater. Increasing the N value from 1 to 3 in this context brings mutual benefits: a significant 17% increase for the greedy algorithm and a modest 3% improvement for RPSM.

In the context of the search of a stationary target, the increase in N almost does not benefit RPSM due to its inherent constraint of allocating substantial effort to regions with low probability of detection. Conversely, the greedy algorithm derives significant advantages from such increases in N.

B. Moving target search

In this section, we evaluate the performance of our algorithms in the context of the moving target search.

1) Interest of increasing N for the moving target search: In Figure 7, we first observe the mean completion time (referred by "meantime" in the remaining of this section) obtained with FAB+dynamic programming w.r.t budget for different values of N. We observed comparable results across different values of horizon, and the difference of performance between the algorithms are best observed for larger values of horizon. We therefore isolate the results for a horizon of 10.

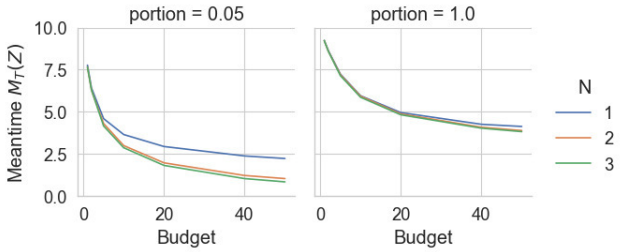

Fig. 7. Meantimes w.r.t. budget obtained with FAB + dynamic programming for N= 1,2,3 for different values of FOV_portion.

In our observations, an increase in the value of N induces a decrease in the meantime. Specifically, for a smaller area of interest, we witness a 6% enhancement in efficiency when N is incremented from 1 to 2, followed by a further 2% improvement as N increases from 2 to 3. This trend shows the diminishing benefits of increasing N.

2) Meantime for different algorithms: In Figure 8, we compare the meantimes obtained with FAB+greedy algorithm, FAB+dynamic programming and RPSM w.r.t. budget, for a fixed horizon = 10. we observe it for different sizes of area of interest.

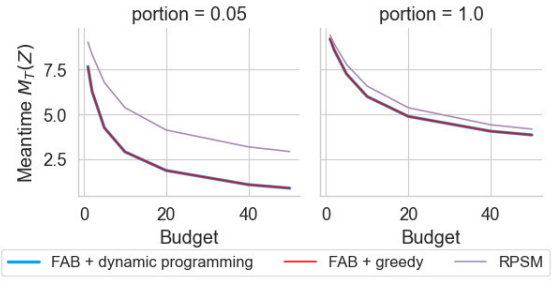

Fig. 8. Comparison of meantimes obtained with different algorithms (the red line is behind the blue line)

Our analysis reveals that the meantimes of plans computed by FAB coupled with dynamic programming and FAB with the greedy algorithm are remarkably similar. Indeed, in 0.45% of the instances, their results are identical.

The mean absolute error in meantime between FAB + greedy algorithm and FAB + dynamic programming is marginal, at 0.002 timesteps (1 timestep = 7 seconds in our experiments, so 0.014 seconds). The peak recorded performance advantage of FAB + dynamic programming over FAB + greedy method amounted to 5.25% of the computed meantime (1.15 seconds gained).

We observed that the adaptation of RPSM to moving target search always performs worse than FAB, especially when the area of interest is small.

3) Comparison of computation times: In Figure 9, we observe the mean computation times of FAB + dynamic programming and FAB + greedy for different values of N, w.r.t. the horizon.

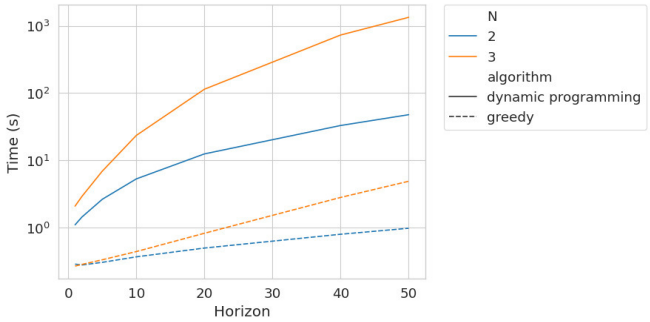

Fig. 9. Computation times for FAB + dynamic programming and FAB + greedy algorithm for different values of N

We observe that FAB + greedy algorithm is several order of magnitudes faster to compute the allocation plan than FAB + dynamic programming. FAB + greedy is often under 1s while FAB + dynamic programming can take several dozens of minutes. Consistently with the worst case time complexity, we also observe that the computation time of the dynamic programming algorithm increases faster with N than the computation time of the greedy algorithm. In a context where allocations must be computed in real time, the greedy algorithm therefore provides a good approximation of the

plan computed with FAB + dynamic programming in a very reasonable amount of time. Furthermore, there is a tradeoff to be made between plan quality and computation time as using higher values of N affects both significantly.

4) Search for targets with different movement models: In Figure 10, we observe the meantime of plans for different target moves w.r.t. the budget. We isolate the experiments where horizon = 10 and the area of interest is small. Indeed, these are cases when the difference between the results are best observed.

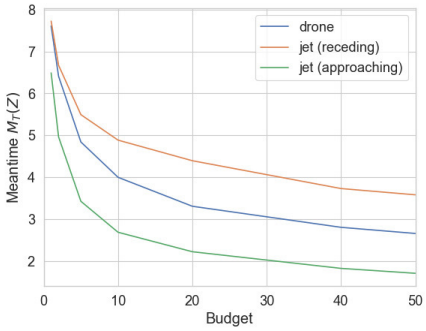

Fig. 10. Meantime for different target types

We can see that when we have a prior on the target direction, it is easier to detect it when it is approaching us and harder when it is receding. When the target moves like a drone and we do not have a prior on its direction, the meantime lies between the meantimes for the two type of jet moves. Note that in the case of a receding target, our model is optimistic because we do not consider ground echoes. Overall, when the target is a jet and we have a prior on its direction, the prior heavily impacts the meantime.

VII. Conclusion

We proposed an optimal algorithm and a fast heuristic for the problem of search of a stationary target with a radar by considering overlapping cones of observation and discrete effort. This allowed us to compute better plans in diverse contexts for moving targets. We also proposed a detection model for the radar which is more realistic than the ones generally used in the literature. Those algorithms were compared in different scenarios for the search of stationary and moving targets. The experiments showed that the realistic detection model gives allocation plans that are significantly different from the one that was previously used in the literature. Using overlapping cones observations led to a notable improvement of the solution obtained. The use of the greedy approximation gave very close to optimal solutions for the stationary search and reduced drastically the computation time for the search of a moving target. Finally, we demonstrated that a prior on the target's direction heavily impacts the mean completion time of the plan.

Future work should consider further theoretical and practical issues. Among them is the complexity of the stationary target search problem. Live experimentations of such algorithms in real conditions remain to be done in order to demonstrate their maturity. Also, other modern important issues are related to the more general problem of searching and tracking multiple targets with multiple assets like e.g. jet fighters and remote carriers embedding different types of active and passive sensors.

References

[1] L. D. Stone, J. O. Royset, and A. R. Washburn, *Optimal search for moving targets*. Springer, 2016.
[2] M. Raap, M. Preuß, and S. Meyer-Nieberg, "Moving target search optimization – a literature review," *Computers and Operations Research*, vol. 105, pp. 132–140, 2019.
[3] F. Delavernhe, P. Jaillet, A. Rossi, and M. Sevaux, "Planning a multi-sensors search for a moving target considering traveling costs," *European Journal of Operational Research*, vol. 292, no. 2, pp. 469–482, 2021.
[4] J. N. Eagle, "The optimal search for a moving target when the search path is constrained." *Operations Research*, vol. 32, pp. 1107–15, 1984.
[5] C. Simonin, J.-P. Le Cadre, and F. Dambreville, "A common framework for multitarget search and cross-cueing optimization," in *International Conference on Information Fusion*, 2008, pp. 1–8.
[6] H. A. L. Thi, D. M. Nguyen, and T. P. Dinh, "A DC programming approach for planning a multisensor multizone search for a target," *Computers and Operations Research*, vol. 41, pp. 231–239, 2014.
[7] A. R. Washburn, "Search for a moving target: The FAB algorithm," *Operations Research*, vol. 31, no. 4, pp. 739–751, 1983.
[8] A. Washburn, "Branch and bound methods for a search problem," *Nav. Res. Logist.*, vol. 45, no. 3, pp. 243–257, 1998.
[9] M. Morin, I. Abi-Zeid, P. Lang, L. Lamontagne, and P. Maupin, "The optimal searcher path problem with a visibility criterion in discrete time and space," in *International Conference on Information Fusion*, 2009, pp. 2217–2224.
[10] S. S. Blackman and R. Popoli, *Design and analysis of modern tracking systems*, ser. Artech House radar library. Boston: Artech House, 1999.
[11] J. L. Williams, "Search theory approaches to radar resource allocation," *7th U.S. / Australia Joint Workshop on Defense Applications of. Signal Processing*, 2011.
[12] H. Vaillaud, C. Hanen, E. Hyon, and C. Enderli, "Target search with a radar on an airborne platform," in *International Conference on Information Fusion*, 2023, to appear in IEEE Xplore.
[13] S. Pérez Carabaza, *Multi-UAS Minimum Time Search in Dynamic and Uncertain Environments*, ser. Springer Theses. Cham: Springer International Publishing, 2021. [Online]. Available: https://link.springer.com/10.1007/978-3-030-76559-0
[14] J. W. Caspers, "Radar random permutation scan method," Patent, oct, 1966.
[15] S. S. Brown, "Optimal search for a moving target in discrete time and space," *Operations Research*, vol. 28, no. 6, pp. 1275–1289, 1980.