Search Theory Report

Chuanquan (Charles) Shu

Advisor: Patrick Jaillet

May 2018

1 Problem Statement and Project Scope

Search optimization entails optimally allocating multiple sensors' search resources to locate a moving target in a multi-zone setting. Two papers (A Hierarchical Approach for Planning a Multisensor Multizone Search for a Moving Target [Hierarchical paper][1] and A DC programming approach for planning a multisensor multizone search for a target [DCA paper][2]) are surveyed and their algorithms are implemented in this project. The project also quantifies the impact of various assumptions in the former paper and in particular explores methods to relax the specific assumption of target transition matrix.

2 Problem Description and Works Surveyed

This report first gives a more specific definition of the problem statement and then elaborates on the work surveyed.

2.1 Problem Description

The problem is to use sensors to detect a moving target in a landscape. The notations are as following: [1].

- z: Zone index; the landscape is divided into multiple zones
- i: Cell index; each zone is divided into multiple cells and each cell is of a different type of terrain (rough land, smooth land, water, town, etc)
- s: Sensor index; a sensor can only be placed in one zone in a given time
- Φ_s: the quantity of resources available for sensor s
- $w_{i,z,s}$: the visibility coefficient; it characterizes the acuity of sensor s over cell i of the zone z and is usually controlled by the type of terrain of cell i
- $\alpha_{z,i}$: the prior probability of target's initial location
- T: the 3-by-3 transition matrix specifying the movement probability of the target from a center cell to the 8 neighboring cells.

The problem is, at every time step, to first optimally allocate each sensor into the zones and then each sensor's resources into different cells in the zone (that contains the sensor). The objective is to minimize the non-detection probability of the target.

[1] The notations in this report follow the those in the DCA paper

2.2 Hierarchical paper [2]

The Hierarchical paper proposes a way to solve the sensor allocation problem for both non-moving and moving target.

Non-Moving Target

This subsection focuses on the setting of non-moving target. The variables to be solved are:

- $u_{z,s}$: the binary indicator of whether sensor s is allotted to the zone z
- $x_{i,z,s}$: the amount of resources sensor s allocated in zone z and cell i; $\sum_{i=1}^{C_z} x_{i,z,s} \leq \Phi_s$

The assumption on nondetection probability by a sensor s in zone z and cell i is:

$$\bar{P}_s(x_{i,z,s}) = exp(-w_{i,z,s} \cdot x_{i,z,s} \cdot u_{z,s}) \tag{1}$$

The assumption on multiple sensors collaboration is that they work independently and hence the nondetection probability by all sensors in zone z and cell i is $\prod_{s=1}^{S} \bar{P}_s(x_{i,z,s})$. The optimization problem is then as follows:

$$\begin{aligned}
\min_{u,x} \quad & \sum_{z=1}^{Z}(\sum_{i=1}^{C_z} \alpha_{z,i} \prod_{s=1}^{S} \bar{P}_s(x_{i,z,s})) \\
s.t. \quad & \sum_{i=1}^{C_z} x_{i,z,s} \leq \Phi_s, \quad z = 1, ..., Z, s = 1, ..., S, \\
& \sum_{z=1}^{Z} u_{z,s} = 1, \quad s = 1, ..., S, \\
& u_{z,s} \in \{0, 1\}, \quad x_{i,z,s} \geq 0, s = 1, ..., S, z = 1, ..., Z, i = 1, ..., C_z
\end{aligned} \tag{2}$$

The term $(x_{i,z,s} \cdot u_{z,s})$ in (1) creates non-linearity that makes (2) hard to solve. But if u is given, (2) becomes easy to solve with standard solver (like Ipopt). Note that the space of u is discrete and finite; it is easy to explore.

The Cross-Entropy (CE) method in the Hierarchical paper thus treats (2) as a hierarchical problem by first solving for u and then solving for x. More specifically, the method randomly generates N u according to a distribution G (uniform at first: a sensor is equally likely to be placed in any zone) and then solve (2) with these u. The K draws of u corresponding to the first K optimal solutions of (2) are then used to update G. This process repeats until convergence and G will become a degenerate distribution (which sensor for which zone is determined; u is solved). At last, with the final u, solve once again (2) for x.

Moving Target

This subsection focuses on the setting of moving target, where at each timestep, the sensors can be completely reallocated and the target will make a move according to transition matrix T. Thus, the objective, the overall nondetection probability, becomes :

$$\sum_{\vec{\omega} \in \Omega} \sum_{z=1}^{Z}(\sum_{i=1}^{C_z} \alpha(\vec{\omega}) \prod_{t=1}^{T} \prod_{s=1}^{S} \bar{P}_s^t(x_{i,z,s}^t)) \tag{3}$$

where Ω is the set of all possible target trajectories and overall nondetection probability is the product of nondetection probability at each step. By reformulation, (3) can be expressed as:

$$\sum_{z=1}^{Z}(\sum_{i=1}^{C_z} \beta_{z,i}^{\tau} \prod_{s=1}^{S} \bar{P}_s^{\tau}(x_{i,z,s}^{\tau})) \tag{4}$$

[2] Implementation in Hierarchical.jl

where $\beta^\tau_{z,i} = \sum_{\vec{\omega}\in\vec{\omega}_{\tau,z,i}} \alpha(\vec{\omega}) \prod_{1\le t\le T}^{t\neq\tau}(\prod_{s=1}^{S} \bar{P}^t_s(x^t_{i,z,s})))$. By this formulation, (4) for a given timestep τ can be solved in the same way as (2) provided $\beta^\tau_{z,i}$ is known. Lastly, to solve for $\beta^\tau_{z,i}$, the paper uses forward-backward split:
$\beta^\tau_{z,i} = U^\tau(z,i)D^\tau(z,i)$ where U and D are recursively defined by:

$$U^\tau(z,i) = \sum_{j\in\{\text{all cells}\}} \alpha_{\tau-1,\tau}(j,i)(\prod_{s=1}^{S} \bar{P}^{\tau-1}_s(x^{\tau-1}_{i,z,s}))U^{\tau-1}(j,i) \tag{5}$$

$$D^\tau(z,i) = \sum_{j\in\{\text{all cells}\}} \alpha_{\tau,\tau+1}(i,j)(\prod_{s=1}^{S} \bar{P}^{\tau+1}_s(x^{\tau+1}_{i,z,s}))D^{\tau+1}(i,j) \tag{6}$$

$\alpha_{\tau-1,\tau}(j,i)$ [the probability of moving from cell j to cell i; not restricted to any zone] and $\alpha_{\tau,\tau+1}(i,j)$ are determined by transition matrix T.

2.3 DCA paper [3]

The DCA paper does not consider the moving target setting; it focuses on reformulating (2) by DC programming, allowing the optimization problem to be solved at once rather than in a hierarchical fashion. In particular, define:
$D = \{x = (x_{i,z,s}) \in \mathbb{R}^d_+ : \sum_{i=1}^{C_z} x_{i,z,s} \le \Phi_s,\ \ z = 1,...,Z, s = 1,...,S\}$
$p(u) = \sum_{z=1}^{Z}\sum_{s=1}^{S} u_{z,s}(1-u_{z,s})$
$K = \{u = (u_{z,s}) \in [0,1] : \sum_{z=1}^{Z} u_{z,s} = 1,\ \ s = 1,...,S\}$ and
$f(x,u) = \sum_{z=1}^{Z}(\sum_{i=1}^{C_z} \alpha_{z,i} \prod_{s=1}^{S} \bar{P}_s(x_{i,z,s}))$
Then (2) is equivalent to:

$$\begin{aligned} &\underset{u,x}{min}\ f(x,u) \\ &s.t.\ \ x \in D, u \in K, p(u) \le 0 \end{aligned} \tag{7}$$

With some proof, we can define a number ρ such that (7) is equivalent to:

$$min\{g(x,u) - h(x,u) : (x,u) \in D \times K\}, \tag{8}$$

where $g(x,u) := \frac{\rho}{2}||(x,u)||^2$ and $h(x,u) := \frac{\rho}{2}||(x,u)||^2 - f(x,u) - tp(u)$.
Solving (8) using DC programming consists of "computing, at each iteration k, the two sequences $\{(y^k,v^k)\}$ and $\{(x^k,u^K)\}$ such that $(y^k,v^k) \in \partial h(x^k,u^k)$ and (x^{k+1},y^{k+1}) is an optimal solution of the following convex problem:"

$$min\{\frac{\rho}{2}||(x,u)||^2 - \langle (x,u),(y^k,v^k)\rangle : (x,u) \in D \times K\}$$

which can be decomposed into:

$$min\{\frac{\rho}{2}||x||^2 - \langle x, y^k\rangle : x \in D\} \tag{9}$$

and

$$min\{\frac{\rho}{2}||u||^2 - \langle u, v^k\rangle : u \in K\} \tag{10}$$

In summary, the steps in DCA algorithm are:

1. Randomly initialize (x^0,u^0) within the constraints $D \times K$
2. Compute $(y^k,v^k) = \nabla h(x^k,u^k)$
3. Compute (x^{k+1},u^{k+1}) by solving (9) and (10)
4. Iterate steps 2 and 3 until convergence

[3] Implementation in DCA.jl

3 Impact of Used Assumptions

There are three assumptions: the probability of initial location (α), the visibility coefficients (w) and the transition matrix of the target (T). Both the Hierarchical and DCA methods requires exact knowledge of the three assumptions to compute the nondetection probability. In this section, the report quantifies the impacts on nondetection probability if the assumptions are inaccurate to various degrees.

3.1 Adding Perturbation to Assumptions

This project begins by solving (4), using the statistics on α, w and T given in the DCA paper, for the sensor allocation strategy and the resulting nondetection probability. Then, randomly apply 1,000 small perturbations on α to recalculate the nondetection probability given the previously solved sensor allocation strategy. The range of these 1,000 outcomes show the variability in the nondetection probaility if the sensor allocation strategy is determined based on inaccurate assumption in α. The above process is repeated for medium, large perturbations and for w, T.

Perturbation is applied by modifying each **non-zero** number v in a given assumption, as follows:

$$v_{perturb} = v_{original} + Normal(0, v_{original}) \times perturb\_magnitude \tag{11}$$

If the given assumption is α or T, the perturbed α or T are also normalized. This analysis holds true that the initial given assumptions on α, w and T are not biased (hence the perturbation has zero mean) and the initial given assumptions are roughly proportionally correct (hence the normal perturbation has varying variance parameters). The *perturb_magnitude* is an artificial number multiplied to the perturbation (small perturbation uses 0.1; medium, 0.3, large, 0.5).

3.2 Results

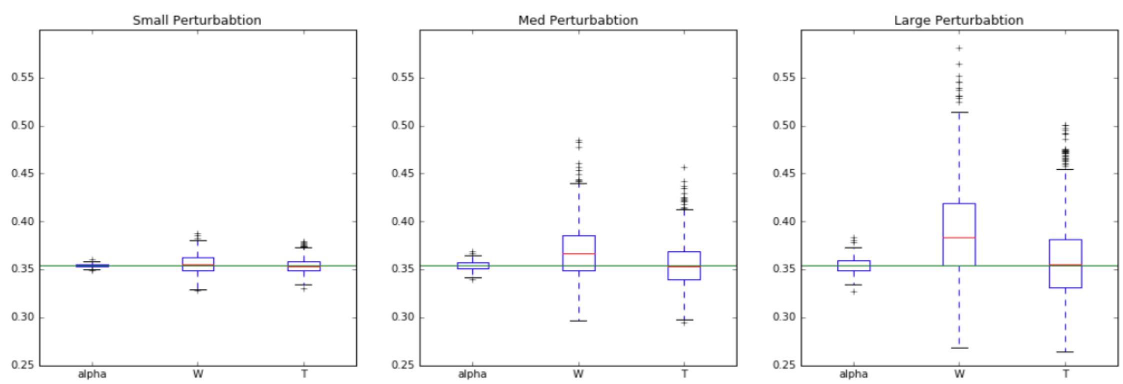

Figure 1: Comparison of different scale of perturbation on different assumptions

The impact of inaccurate assumptions in α, w and T are given in Figure 1. The green horizontal line represents the nondetection probability solved using the statistics (or assumptions) on α, w and T given in the DCA paper. The probability is **0.3542**.

From the figure, the probability of the target's initial location (α) has little impact on the final nondetection; even with large perturbation, the interquartile-range (IQR) stays closely to 0.3542. On the other hand, inaccuracy in sensor visibility coefficients w and and target transition matrix T can lead to diverse results. In particular, under med perturbation for T, roughly half of the results are favorable while the other half are not. The most adverse case in the 1,000 draws is 0.4564. This means while we expect 0.3542 nondetection probability by following the sensor allocation strategy (solved using the given assumptions), the actual nondetection probability can be as high as 0.4564 using inaccurate T assumption. Beyond inaccuracy in T assumptions, the inaccuracy in w could lead to even more adverse outcomes. Specifically, Figure 1 shows that under med perturbation for w, the median of the 1,000 results (0.3670) appears higher than 0.3542. This difference represents a measure of the bias in the solved sensor allocation strategy using inaccurate sensor visibility coefficients.

The insignificance of α is intuitive as the search is multi-timestep. Hence, even without good knowledge of the target's initial position, the algorithm can deduce the target's whereabouts by sequentially allotting sensors at each timestep. The lack of bias when perturbing the transition matrix T might be due to how perturbation is applied. Recall that only the **non-zero** elements in T are perturbed; this means the target's moving direction is not challenged in the 1,000 draws. In the experiment, the transition matrix used has a clear direction as show in Figure 2. As a result, bias is not significant.

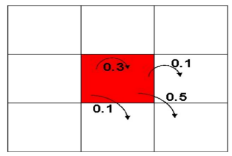

Figure 2: Target's Transition Matrix T

4 Relaxation of the Transition Matrix Assumption (Exploration)

In this section, the report discusses some exploratory works done to relax the requirement of knowing exactly the target's transition matrix (T) which is often too demanding. In particular, the concept of adaptively uncovering the transition matrix is probed.

Starting with a completely unknown T, the key idea is to allot the sensors in a way that the first few timesteps are used for exploration (uncovering T) and the rest can be used for exploitation. The intuition is as follows. Assume there is a powerful sensor. Assume in a simple 2x2 landscape where the target is known to be at the top left cell when t=1. As in Figure 3, if the sensor is placed at the top left cell when t = 1, and bottom right cell when t = 2 and if the target is not located in either time, then T is likely not strong in the SE direction. Hence, the previous timesteps' sensor allotment and nondetections should reveal information about the True T.

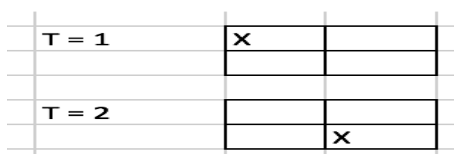

Figure 3: Illustrating the Intuition. We place the sensor at X.

The challenge is to design feedbacks so that previous timesteps' sensor allotment and nondetections can be used to update the guesses of T. To do so, the project makes one simplification: at the end of each timestep, the actual nondetection probability can be observed. To clarify, at the beginning of a timestep, sensors allotment plan is solved under the current transition assumption T_{assume}, yielding an expected nondetection probability. If T_{assume} is different from T_{true}, the actual nondetection probability should be different from the expected one and this discrepancy can be used to update T_{assume}. The specific update steps are:

1. At the beginning of every timestep, use the current T_{assume} and solve for sensors allotment plan for that timestep.
2. Starting from timestep 2, at the end of every timestep, randomly generate 1,000 transition matrices $T_{candidates}$, 250 for each of the 4 directions as show in Figure 4.
3. Given the fixed sensors allotment plans for ALL previous timesteps, use each $T_{candidates}$ and the T_{true} to calculate the expected nondetection probabilities and actual nondetection probabilities for ALL previous timesteps. The mean absolute error (MAE) between the two vectors is calculated.
4. The 10 $T_{candidate}$ corresponding to the first 10 smallest MAE of the best direction (also determined by MAE) are averaged to result in a new T_{assume} which is used in the next timestep.

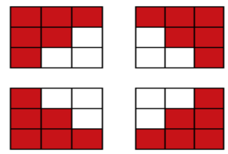

Figure 4: The 4 types of directions: NW, NE, SE, SW. The white squares are restricted to be zero; the red squares are allowed to be non-zero.

To conduct experiments, the configurations of α, w and T_{true} need to be modified so the search problem becomes more difficult and thus allowing more timesteps than just 4. In particular, the T_{true} now has NW as its dominant direct in as in Figure 5 (left).

0.25	0.125	
0.125	0.5	

[0.2 - 0.3]	[0.12 - 0.127]	
[0.11-0.135]	[0.35-0.55]	

Figure 5: T_{True} (left) and Relaxed T (right; for the discussion in section 5.2)

The results of adaptively learning T are summarized in Figure 6. The blue lines show the nondetection probability when the sensor allotment plans are made using uniform T across all timesteps. The expected nondetection decreases slowly compared to planning using T_{True} because the exploration phase has no direction. The associated actual nondetection plateaus probably also because the resources are wasted in the wrong directions. Likewise, the green lines show the nondetection probability when the sensor allotment plans are made adaptively. This adaption apparently nudges more resources to be allotted to the right direction (rather than uniformly spread) and hence green actual nondetection is slightly better than blue's.

The red lines show the nondetection probability when the sensor allotment plans are made using the exact opposite transition matrix from T_{True} across all timesteps. The algorithm expects to spot the target easily because its search has a focused direction. But in reality the focused search is in the completely wrong direction (SE rather than NW). Hence, the actual nondetection plateaus at a very high number. By adapting the T_{assume} at every timestep, the algorithm performs better in the actual nondetection. This is perhaps because the algorithm is able to do focused search in the wrong direction and quickly realize the mistakes (from feedbacks). Then, it adjusts the search to the correct direction. Figure 7 summarizes all actual nondetections.

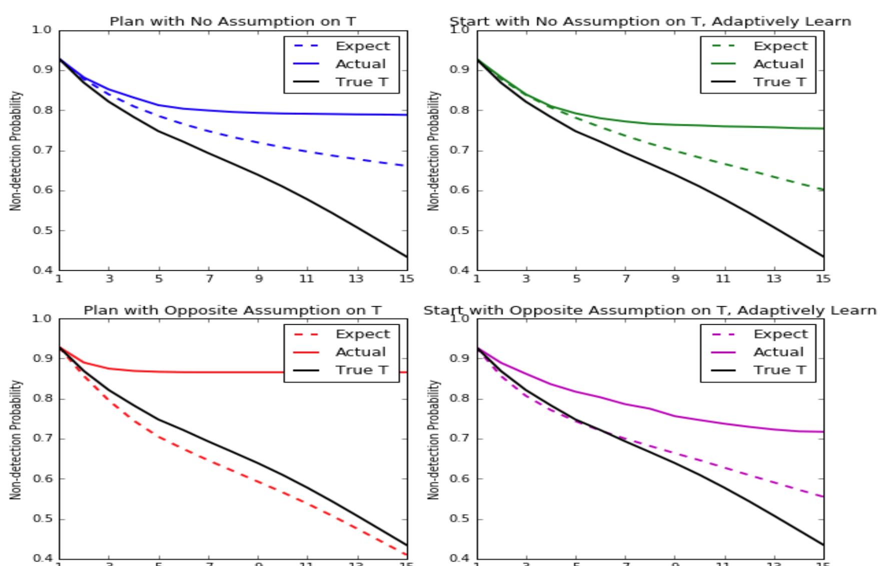

Figure 6: Comparison of Adaptively Uncovering the T

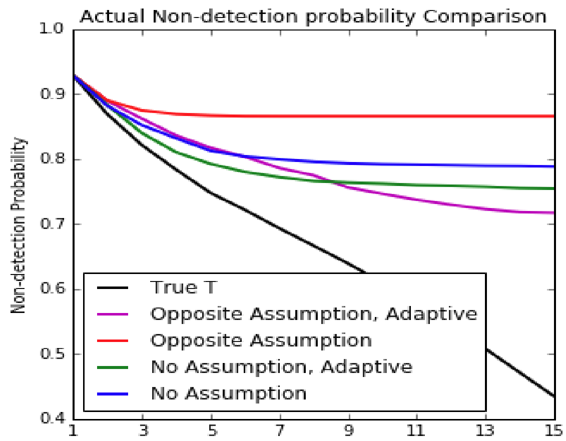

Figure 7: Comparison of Adaptively Uncovering the T

5 Future Work

5.1 Improvements in the Adaptive Solution

The key simplification in the presented adaptive solution is that the actual nondetection probability at the end of each timestep is observable. This is not true in reality. Only the binary outcome of detecting and not detecting the target can be observed. More specifically, the observed outcomes are a sequence of 0's which may be followed by a 1. This type of outcomes does not work with the update procedure proposed in section 4. Hence, a custom update procedure needs to be designed and perhaps more timesteps are need to be used for exploration.

Although Figures 6 and 7 are, to some extent, indicative of adaptive solution's potential, the plotted results are based on 1 run. To formally substantiate its potential, the project needs to do many more runs to construct the standard deviation of the plotted curves. Additionally, the experiment also should be repeated for different configurations of α, w, the initial T and landscape to demonstrate universality.

5.2 Min-Max Approach

An alternative way to relax the Transitional Matrix assumption is to assume that all the numbers in the initially given T are bounded in some intervals, as shown in Figure 5 (right). Consider $T \in \Delta$ where Δ represents the space of T (according to the bounding intervals). Then, the objective in (4) becomes:

$$\min_{u,x} \sum_{z=1}^{Z} \Big(\sum_{i=1}^{C_z} \max_{T \in \Delta} \, (\beta^{\tau}_{z,i} \prod_{s=1}^{S} \bar{P}^{\tau}_{s}(x^{\tau}_{i,z,s})) \Big) \tag{12}$$

Recall $\beta^{\tau}_{z,i}$ is determined by (5) and (6), which contains $\alpha_{\tau-1,\tau}(j,i)$ and $\alpha_{\tau,\tau+1}(i,j)$. These two quantities are controlled by the constraint $T \in \Delta$. Intuitively, the optimization needs to minimize the nondetection when the environment chooses the most adverse T. Further investigation is needed to solve this optimization problem. A preliminary idea is to alternatively fix the *min* variables u, x to update the max variable T and to fix T to update u, x (similar to how Generative Adversarial Network solves its min-max problem).

References

[1] Cécile Simonin, Jean-Pierre Le Cadre, and Frédéric Dambreville. A hierarchical approach for planning a multisensor multizone search for a moving target. *Comput. Oper. Res.*, 36(7):2179–2192, July 2009.

[2] Hoai An Le Thi, Duc Nguyen, and Tao Pham Dinh. A dc programming approach for planning a multisensor multizone search for a target. 41:231–239, 01 2014.