Computers & Operations Research 36 (2009) 2179 – 2192

Contents lists available at ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/cor

A hierarchical approach for planning a multisensor multizone search for a moving target

Cécile Simonin[a,*], Jean-Pierre Le Cadre[a], Frédéric Dambreville[b]

[a]IRISA/CNRS, Campus de Beaulieu, 35042 Rennes (cedex), France
[b]DGA, 16 bis avenue prieur de la Côte d'or, 94114 Arcueil, France

ARTICLE INFO

Available online 22 August 2008

Keywords:
Search theory
Combinatorial optimization
Simulation

ABSTRACT

This paper deals with a well-known problem in the general area of search theory: optimize the search resources sharing so as to maximize the probability of detection of a (moving) target. However, the problem we consider here considerably differs from the classical one. First, there is a bilevel search planning and we have to consider jointly discrete and continuous optimization problems. To this perspective original methods are proposed within a common framework. Furthermore, this framework is sufficiently general and versatile so as to be easily and successfully extended to the difficult problem of the multizone multisensor search planning for a Markovian target.

1. Introduction

The optimization of search problems is a matter of concern for many real world applications, especially for the Intelligence Community. Moreover, the dimension of the space of search is often very large compared with the amount of resources available to proceed the search. Such kinds of problems can inherently be viewed in a hierarchical manner: determine search zones for the sensors and then optimize their search plans over these zones. However, hierarchical problems are often very hard to solve. Indeed, they are difficult to model in such a way that they could be optimized by classical means: at the upper level, the objective functional can be non-convex or implicitly defined as the result of an algorithm (lower level). To overcome this difficulty, an adaptive solving approach is needed. Thus, we introduce a hierarchical method that allows to solve hierarchical search problems.

In this paper we consider a space of search partitioned into zones of reasonable size. A unique sensor must be able to explore efficiently a whole zone. Each zone is itself partitioned into cells. A cell is an area in which every points have the same properties, according to the difficulty of detection (altitude, vegetation, etc.). An example of search space is given in Fig. 4 in Section 6. Furthermore, each sensor has its own coefficient of visibility over a cell. The visibility coefficients depend also on the kind of target that is searched. Here, there is a unique target to detect. Target location and motion are defined by a given prior within a Bayesian setup. Thus, the hierarchical search problem to solve is made of two completely entangled optimization levels:

- at upper level: find the best allotment of sensors to search zones (a sensor is allotted to a unique zone);
- at lower level: for every sensor, find the best resource sharing in order to have an optimal surveillance over the allotted zone.

This problem is easy to solve when the allotment of sensors to search zones is injective (1:1). Indeed, it can be solved by employing linear programming (LP) at upper level.

But in the general case where the (1:1) hypothesis is abandoned, the complexity of the problem increases dramatically. As a consequence, classical methods become awkward to employ. In order to overcome this difficulty, the upper level is optimized via a rare event simulation method developed by Rubinstein [1]: the cross-entropy (CE) algorithm. At lower level, several algorithms can be employed. We first consider an optimization method based on the algorithm of de Guenin [2] for detecting a stationary target. In a second time, we consider the stationary two-sided search (both target location and sensors resource sharing have to be optimized). This part of our work relies on results of Nakai [3]. Then we see how the CE can be coupled with a forward and backward algorithm, introduced by Brown [4,12], in order to optimize detection when the target has a Markovian motion. At last we study the optimization of a cross-cueing functional where we try to detect and confirm a target with two different means of search in a multiperiod framework.

* Corresponding author. Tel.: +33 299 842 546; fax: +33 299 847 171.
E-mail addresses: csimonin@irisa.fr, simonin.cecile@neuf.fr (C. Simonin), lecadre@irisa.fr (J.-P. Le Cadre), frederic.dambreville@etca.fr (F. Dambreville).

0305-0548/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2008.08.007

2. A model for the hierarchical problem

We first introduce the model and the notations employed in the remainder of the paper. As said before, the problem aims to maximize the probability of locating a target hidden in a large space of search when available resources are scarce.

2.1. Main notations

E	space of search
z	zone index
i	cell index
s	sensor index
t	time index
α	prior on the initial location of the target
$\varphi_s(c_{z,i})$	quantity of resource of sensor s allotted to cell $c_{z,i}$
Φ_s	quantity of resource available for sensor s to search the space
$w^s_{z,i}$	coefficient that characterizes the acuity of sensor s over cell $c_{z,i}$ (visibility coefficient)
$x_{z,s}$	allocation variable. $x_{z,s}=1$ if sensor s is allotted to zone E_z, else $x_{z,s}=0$
m	sensors-to-zones allotment mapping
(P_μ)	family of probability laws for the CE algorithm
$p^M(z\|s)$	probability of allotting sensor s to zone E_z according to M
β^t	"prior" at time t
$U^t(z,i)$	forward functional for cell $c_{z,i}$ at time t
$D^t(z,i)$	backward functional for cell $c_{z,i}$ at time t

2.2. The searching framework

The space of search: The search space, named E, is considered to be the theater of operations. It is thus a large space with spatially variable search characteristics. This variability will be described below, up to an elementary level (the cell level or lower level) where search homogeneity will be assumed. Precisely, the search space E is divided into Z search zones, denoted $E_z, z\in\{1,\dots,Z\}$, each of them partitioned into C_z cells, denoted $\{c_{z,i}\}_{i=1}^{C_z}$, so that:

$$\begin{cases} E=\bigcup\limits_{z=1}^{Z} E_z \quad \text{with } z_1\neq z_2 \Rightarrow E_{z_1}\cap E_{z_2}=\emptyset,\\ E_z=\bigcup\limits_{i=1}^{C_z} c_{z,i} \quad \text{with } i_1\neq i_2 \rightarrow c_{z,i_1}\cap c_{z,i_2}=\emptyset. \end{cases} \tag{1}$$

A cell $c_{z,i}$ represents the smallest search area in which the search parameters are constant. For example, it can be a part of land with constant characteristics (altitude, landscape). Each zone must have a reasonable size in order to be explored by a sensor within a fixed time interval.

The target: The target is hidden in one unit of the search space. Its location is characterized by a prior $\alpha_{z,i}$. Thus, we have

$$\sum_{z=1}^{Z}\sum_{i=1}^{C_z}\alpha_{z,i}=1. \tag{2}$$

This prior may be relatively informative and results (in general) from operational considerations, previous searches, intelligence, etc. We will see also (see Section 3.1) how it can be relaxed.

The means of search: Means of search can be passive (e.g. IRST, ESM) or active sensors (radars). We will consider that searching the target will be carried out by S sensors. Due to operational constraints, each sensor $s\in S$ must be allotted to a *unique* search zone. For example, it could be the exploration time to share between units of a zone. At the lower level the amount of search resource allocated to the cell $c_{z,i}$ for the sensor s —if sensor s is allotted to zone E_z – is denoted $\varphi_s(c_{z,i})$. It can represent the time spent on searching the cell $c_{z,i}$ (passive sensor), the intensity of emissions or the number of pulses (active sensors), etc. Furthermore, each sensor s has a search amount Φ_s, i.e.:

$$s\rightarrow E_z \;\Rightarrow\; \sum_{i=1}^{C_z}\varphi_s(c_{z,i})\leqslant\Phi_s. \tag{3}$$

To characterize the effectiveness of the search at the cell level, we consider the conditional *non-detection* probability $\bar{P}_s(\varphi_s(c_{z,i}))$ which represents the probability of not detecting the target given that the target is hidden in $c_{z,i}$, and that we apply an elementary search effort $\varphi_s(c_{z,i})$ on $c_{z,i}$. Some hypotheses are made to model $\bar{P}_s(\varphi_s(c_{z,i}))$. For all sensors, $\varphi_s(c_{z,i})\longmapsto\bar{P}_s(\varphi_s(c_{z,i}))$ is convex and non-increasing (law of diminishing return). Assuming independence of elementary detections, a usual model [5] is

$$\bar{P}_s(\varphi_s(c_{z,i}))=\exp(-w^s_{z,i}\varphi_s(c_{z,i})), \tag{4}$$

where $-w^s_{z,i}$ is a (visibility) coefficient which characterizes the reward for the search effort put in $c_{z,i}$ by sensor s.

An additional assumption is that sensors act independently at the cell level which means that if S sensors are allotted to $c_{z,i}$ the probability of not detecting a target hidden in $c_{z,i}$ is simply

$$\prod_{s=1}^{S}\bar{P}_s(\varphi_s(c_{z,i})). \tag{5}$$

2.3. The optimization problem

Let us give now a general presentation of the problem. Let $m: s\longrightarrow z$ be a mapping allotting sensors to search zones. Our aim is to find both the optimal mapping m (i.e. the best sensor-to-zone allotment) and the optimal local distributions φ_s in order to minimize the non-detection probability. The objective functional is then

$$F(m,\{\varphi_s,s\in\{1,\dots,S\}\})=\sum_{z=1}^{Z}\left(\sum_{i=1}^{C_z}\alpha_{z,i}\prod_{s\in m^{-1}(z)}\bar{P}_s[\varphi_s(c_{z,i})]\right), \tag{6}$$

which leads to solve the following constrained problem:

$$\left|\begin{array}{ll} \min\limits_{m,\{\varphi_s,s\in\{1,\dots,S\}\}} & F(m,\{\varphi_s,s\in\{1,\dots,S\}\})\\ \text{s.t.} & \forall z,\forall s\in m^{-1}(z),\ \sum\limits_{i=1}^{C_z}\varphi_s(c_{z,i})\leqslant\Phi_s,\\ & \forall i\in z,\ \varphi_s(c_{z,i})\geqslant 0,\\ & m \text{ mapping}: s\in S\rightarrow z\in Z. \end{array}\right. \tag{7}$$

The structure of this problem is clearly hierarchical. As said before, we have to consider two completely interconnected problems:

- A discrete optimization problem, i.e. sensors allotment to search zones (the upper level).
- A continuous optimization problem, i.e. conditionally to a sensor-to-zone allotment, find the best placement of search efforts (the lower level).

3. Optimization of the hierarchical problem

These two levels (see Fig. 1) are completely entangled: a change in the sensor-to-zone allotment results in a change of the distribution

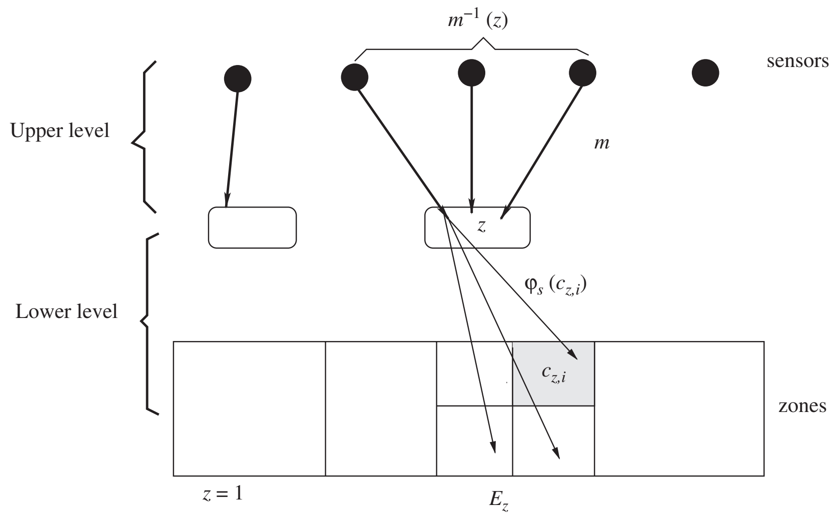

Fig. 1. An overview of the hierarchical optimization scheme.

of search efforts at the lower level, while the sensor-to-zone allotment depends on the effect of the search distribution at the lower level.

At the lower level, the continuous optimization problem can be efficiently solved by classical methods, whereas at the upper level it has to be considered from two points of view, according to the assumptions made about the mapping m.

- The mapping m is injective: the two-level optimization problem (see Eq. (7)) can be viewed as a classical 2D-assignment problem.
- No assumption about m: the general optimization problem can be solved by a learning approach, coupled with a local optimization.

This section aims to describe the solving of the aforementioned problems.

3.1. The optimal distribution of search resources at the lower level

First, we present the methods employed for solving the optimization problem at the lower level. It is assumed that the search efforts are indefinitely divisible. Thus we have to deal with simple and classical continuous optimization problems. Solving the optimization problem widely differs accordingly that m is injective (at most one sensor per zone) or not (more than one sensor per zone).

At most one sensor per search zone, one-sided optimization: In the particular case where at most one sensor (say s) is allotted to a search zone (say z), we have to solve the classical optimization problem:

$$\left| \begin{array}{ll} \min\limits_{\varphi_s} & \sum\limits_{i \in z} f_i[\varphi_s(c_{z,i})] \\ \text{s.t.} & \sum\limits_{i \in z} \varphi_s(c_{z,i}) - \Phi_s \leqslant 0, \quad \varphi_s(c_{z,i}) \geqslant 0, \end{array} \right. \tag{8}$$

where $f_i[\varphi_s(c_{z,i})]$ is a convex and differentiable function of $\varphi_s(c_{z,i})$. A classical choice is $f_i[\varphi_s(c_{z,i})] = \alpha_{z,i} \exp[-w^s_{z,i}\varphi_s(c_{z,i})]$, where $w^s_{z,i}$ is the visibility coefficient in the cell $c_{z,i}$ for the sensor s and $\alpha_{z,i}$ the prior we have about the target location. So, we have to minimize a convex differentiable functional over a convex domain. Necessary and sufficient conditions are given by the Karush–Kuhn–Tucker conditions (denoted KKT for short), yielding (see Appendix A)

$$\forall\, i \in z, \quad \varphi_s(c_{z,i}) = \max\left[0 \left(\frac{1}{w^s_{z,i}}\right) \ln\left(\frac{w^s_{z,i}\alpha_{z,i}}{\lambda}\right)\right] \triangleq \left[\left(\frac{1}{w^s_{z,i}}\right) \ln\left(\frac{w^s_{z,i}\alpha_{z,i}}{\lambda}\right)\right]^+. \tag{9}$$

It is worth stressing that the primal problem (see Eq. (8)) can be efficiently solved via duality (see Appendix A), essentially due to the separability of the objective functional. It remains to find the optimal value of the dual parameter λ, which is obtained by *maximizing* the dual functional $\psi(\lambda)$ (see Appendix A). As $\psi(\lambda)$ is a *monodimensional* concave functional, the maximum is unique and can be found by any classical procedure. This problem (Koopman [5], de Guenin [2]) is quite classical and traces to the origins of search theory.

More than one sensor per search zone, one-sided optimization: In the general case, a subset $m^{-1}(z)$ of sensors is allotted to the zone z. The following continuous optimization problem must be solved for each zone:

$$\left| \begin{array}{ll} \min\limits_{\varphi} & \sum\limits_{i \in z} \alpha_{z,i} \prod\limits_{s \in m^{-1}(z)} \exp(-w^s_{z,i}\varphi_s(c_{z,i})) \\ \text{s.t.} & \forall\, s \in m^{-1}(z),\ \sum\limits_{i \in z} \varphi_s(c_{z,i}) - \Phi_s \leqslant 0, \\ & \quad \forall i \in z, \varphi_s(c_{z,i}) \geqslant 0. \end{array} \right. \tag{10}$$

This problem is more complicated than the classical one (see Eq. (8)), due to the product, but the functional remains (weakly) convex. A simple method for optimizing the sensor use is to optimize iteratively with respect to each continuous variable φ_s (for $s \in m^{-1}(z)$), the other being kept fixed. The algorithm takes then the following form (see also Fig. 1):

- Initialize $\varphi_s(c_{z,i}) = 0\ \forall s \in m^{-1}(z)$,
- *Iteration*: For s_0 going from 1 to $\#m^{-1}(z)$, optimize the resource distribution φ_{s_0}, φ_s being kept fixed for $s \neq s_0$, (11)
- Repeat the above procedure up to convergence.

The two-sided optimization framework: Previously, it was assumed that the prior $\alpha_{z,i}$ was known. This can be a rather demanding assumption, so we try to relax it. To that aim we consider that the only prior we have is the distribution α_z, which is the probability that the target is in zone z. Naturally, it is assumed that these priors sum to 1 (i.e. $\sum_z \alpha_z = 1$). We denote $\alpha_{i|z}$ the conditional probability of the target being in the cell $c_{i,z}$ given that it is in the zone z. Classical definitions yield

$$\sum_{i \in z} \alpha_{i|z} = 1, \quad \alpha_{z,i} = \alpha_z \alpha_{i|z}. \tag{12}$$

Since we no longer assume that the $\{\alpha_{i|z}\}$ are known, we consider that they are determined as the worst distribution i.e. a solution of the following minimax problem:

$$\left|\begin{array}{ll} \min\limits_{m} & \sum\limits_{z}\alpha_z\left(\min\limits_{\varphi_s}\max\limits_{\alpha_{.|z}}\sum\limits_{i\in z}\alpha_{i|z}\prod\limits_{s|s\in m^{-1}(z)}\exp(-w_i^s\varphi_s(c_{i,z}))\right) \\ \text{s.t.} & \forall z,\ \sum\limits_{i\in z}\alpha_{i|z}=1, \\ & \forall z, \forall s\in m^{-1}(z),\ \sum\limits_{i\in z}\varphi_s(c_{z,i})-\Phi_s\leqslant 0, \\ & \forall i\in z,\ \varphi_s(c_{z,i})\geqslant 0, \\ & m \text{ mapping}: s\in S\to z\in Z. \end{array}\right. \tag{13}$$

Here again we can split the problem into two interconnected levels:

- an upper level, i.e. optimization of allotment of sensors to search zones;
- a lower level, i.e. optimization of both prior of the target $\alpha_{i|z}$ and of the resource distribution for the sensors allotted to a search zone.

Solving the local optimization problem for a given zone seems rather formidable at first glance. However, it is shown (see Appendix B) that KKT conditions simply result in the following linear system:

$$\begin{cases} \sum\limits_{s\in m^{-1}(z)} w_{i,z}^s\varphi_s^*(c_{z,i})=\text{cst}, & \forall i\in z, \\ \sum\limits_{i\in z}\varphi_s^*(c_{z,i})=\Phi_s, & \forall s\in m^{-1}(z), \end{cases} \tag{14}$$

where $\varphi_s^*(c_{z,i})$ are solutions of the minimax problem (see Eq. (13)) and the constant cst is unknown. Even if Eq. (14) is a classical linear system, it is worth stressing that it is generally (highly) undetermined.

Actually, there are $C_z\times\#m^{-1}(z)+1$ unknown variables and $C_z+\#m^{-1}(z)$ equations. It is perfectly determined only for $\#m^{-1}(z)=1$ (a classical result of Nakai [3]). In the general case we can represent Eq. (14) as a linear system (see Eq. (15)).

$$\begin{cases} AX=C, \\ X=(\ldots,\varphi_s^*(c_{z,i}),\ldots,\text{cst})^{\mathrm{T}}. \end{cases} \tag{15}$$

Every solution X of Eq. (15) can be written as $X=X_0+\widetilde{A}V$, where X_0 is a particular solution of the following linear system and where the columns of $\widetilde{A}$ form a basis of ker A.[1] Practically, undeterminacy can be overcome by imposing an additional objective functional in the resource placement, e.g.:

$$\sum_{s\in\#m^{-1}(z)}\left[\sum_{i\neq i'}(\varphi_s^*(c_{z,i'})-\varphi_s^*(c_{z,i}))^2\right]. \tag{16}$$

The optimal values of the target distribution (i.e. $\alpha_{i|z}^*$) are deduced from the $\{\varphi_s^*(c_{z,i})\}$ and the KKT conditions (see Appendix B, Eq. (52)).

3.2. Optimizing the sensor-to-zone allotment

We are now considering the optimization of the sensor-to-zone allotment. This will be divided into two parts according to the assumption we make about m.

3.2.1. The LP (linear Programming) solution

Under the assumption that the mapping m is injective, the hierarchical problem is greatly simplified since at the upper level it becomes a 2D-assignment problem. Let $x_{z,s}\in\{0,1\}$ be the assignment variables. We have to solve a 0–1 integer programming problem:

$$\left|\begin{array}{ll} \min\limits_{x_{z,s}} & \sum\limits_{z=1}^{Z}\sum\limits_{s=1}^{S}\text{Cost}_{z,s}x_{z,s} \\ \text{s.t.} & \forall s,\ \sum\limits_{z=1}^{Z}x_{z,s}=1, \\ & \forall z,\ \sum\limits_{s=1}^{S}x_{z,s}=1, \\ & x_{z,s}\in\{0,1\}. \end{array}\right. \tag{17}$$

Furthermore, we know that $\text{Cost}_{z,s}\geqslant 0$. It is quite remarkable that the integrity constraint (i.e. $x_{z,s}\in\{0,1\}$) can be relaxed to positivity constraint ($x_{z,s}\geqslant 0$). The set of solutions remains the same. To show this, it is sufficient to rewrite Eq. (17) as a general LP problem:

$$\left|\begin{array}{ll} \min\limits_{X} & \mathscr{C}^{\mathrm{T}}X \\ \text{s.t.} & AX=1, \\ & X^{\mathrm{T}}=(\ldots,x_{z,s},\ldots) \quad\text{and}\quad \mathscr{C}=(\ldots,\text{Cost}_{z,s},\ldots). \end{array}\right. \tag{18}$$

It is obvious that the constraint matrix A is totally unimodular (i.e. every square submatrix has a determinant $+1,-1$ or 0). Then the LP relaxation solves exactly the integer programming problem. For instance, the simplex algorithm involves $Z\times S$ continuous $x_{z,s}$ variables which is usually quite feasible. The assignment costs $C_{z,s}$ are computed off-line via the low-level optimization procedure which means that

$$\text{Cost}_{z,s}=\sum_{i=1}^{C_z}\alpha_{z,i}\exp(-w_{z,i}^s\varphi_s^*(c_{z,i})), \tag{19}$$

where φ_s^* is the optimal local distribution of the search effort for sensor s. So we have $Z\times S$ continuous low-level optimization problems to solve. The simplicity of the solution is due to the basic assumption we have made (injectivity of m) which allows to write the high-level optimization problem as a 2D-assignment problem. This is no longer possible if this assumption is relaxed since we have to consider (see Eq. (6)): $\alpha_{z,i}\prod_{s\in m^{-1}(z)}\bar{P}_s[\varphi_s(c_{z,i})]$, for each zone of the space of search. So there is no hope to benefit from the "comfort" of LP when the hypothesis of injectivity of m is abandoned.

3.2.2. The solution-learning method

In the general case, the use of LP is not possible: a reason is that the objective functional $F(m,\{\varphi_s,s\in\{1,\ldots,S\}\})$ as given by Eq. (6) cannot be written as a linear functional of the allotment variable $x_{z,s}$, because of the product over sensors. As an exhaustive optimization is clearly unrealistic, we turn toward a simulation approach, namely the CE method. After a brief general presentation of the CE method, we shall consider its ability to solve the upper level problem.

The CE method: The CE method has been developed by Rubinstein [1] initially for evaluating rare events probabilities, for which a direct computation by usual methods would be unreliable. The only way to evaluate them is then to resort to a simulation method based on importance sampling. The CE method allows to tilt proposal densities in order to favor sampling of rare events. In fact, it has been demonstrated that this method is particularly relevant for solving "hard" optimization problems like combinatorial ones. Indeed, when deterministic methods failed to find the optimal solution within a reasonable amount of computation, in most cases the CE method allows to find a fairly good one more quickly. In order to use the CE method to deal with a deterministic optimization problem, this problem must

[1] Such a basis can be obtained by Gaussian elimination.

be first translated into a stochastic one. The set of feasible solutions is then regarded as a set of events subjected to an importance density. Finding the optimum of the problem is considered to be a rare event. A general presentation of the CE algorithm is provided below:

$$
\left|\begin{array}{l}
\text{Choose a family of probability laws } (P_\mu) \text{ representative} \\
\quad \text{of the problem,} \\
\text{Choose a function } B \text{ in order to evaluate the draws,} \\
\text{Initialize the law: } \mu = \mu_0, \\
\textbf{Perform the following iteration until convergence:} \\
\bullet \text{ Draw } N \text{ allotments } m_1, m_2, \ldots, m_N \text{ according to } P_\mu, \\
\bullet \text{ Select the } \theta \times N, 0<\theta<1, \text{ best draws } \{\tilde{m}_1, \tilde{m}_2, \ldots, \tilde{m}_T\}, \\
\quad \text{according to } B(m_j), \\
\bullet \text{ Find } \mu' \text{ minimizing the Kullback distance, i.e.} \\
\quad \text{maximizing: } \sum_{t=1}^{T} \ln(P_{\mu'}(\tilde{m}_t)), \\
\bullet \mu = \mu'.
\end{array}\right. \tag{20}
$$

For convergence analysis of the CE method we refer to [1,6,7]. To overcome the combinatorial problem we had to face at the upper level we will combine the CE algorithm with a classical continuous optimization algorithm at the lower level (see Section 3.1). The CE algorithm draws particular allotments of sensors to search zones that will be evaluated and then selected, in order to obtain a drawing law which will converge toward the optimal allotment. First, we must choose a family of probability laws, (P_μ), describing a probability choice of m. The aim is to find:

$$
\begin{cases}
\mu^* \in \underset{\mu}{\operatorname{argmin}} \sum_m P_\mu(m) B(m), \\
\text{with} \\
B(m) = \min_{\{\varphi_s, s \in \{1,\ldots,S\}\}, m} F[m, \{\varphi_s, s \in \{1, \ldots, S\}\}].
\end{cases} \tag{21}
$$

A discrete probability law $p(z|s)$ is associated to each sensor s. It represents the probability to assign sensor s to zone z. These probabilities are summarized by a matrix M:

$$
\mathrm{M} = \begin{pmatrix}
p^M(1|1) & p^M(2|1) & \ldots & p^M(Z|1) \\
\vdots & \vdots & & \vdots \\
p^M(1|s) & p^M(2|s) & \ldots & p^M(Z|s) \\
\vdots & \vdots & & \vdots \\
p^M(1|S) & p^M(2|S) & \ldots & p^M(Z|S)
\end{pmatrix}, \tag{22}
$$

where $p^M(z|s)$ is the probability to assign sensor s to zone z according to the M matrix. Since m is a mapping, these laws allot at most one search zone per sensor; but 0, 1 or more sensors can be allotted to the same search zone. Of course, we must have

$$
\sum_{z=1}^{Z} p^M(z|s) = 1. \tag{23}
$$

Updating the matrix M: Let $X^k = (x_1^k, x_2^k, \ldots, x_s^k, \ldots, x_S^k)$ be the vector of sensors to zones assignment. Thus, x_s^k is the search zone assigned to sensor s for the draw X^k. The index k corresponds to the k-th iteration of the algorithm. The probability of drawing the vector X^k, according to M is

$$
P_{\mathrm{M}}(X^k) = \prod_{s=1}^{S} p_s^M(x_s^k|s), \tag{24}
$$

where $p_s^M(x_s^k|s)$ is the coefficient z of the row s in matrix M. As explained in Section 3, N draws X^k, $k \in \{1, \ldots, N\}$ will be done. The T best draws, according to F, will be selected to update M. Denoting $\{X^1, \ldots, X^T\}$ as the T "best" vectors among the draws $\{X^1, \ldots, X^N\}$, updating M will be done by calculating the M′ minimizing the Kullback distance $\mathscr{K}$:

$$
\mathscr{K} = \sum_{k=1}^{T} \ln\left[\prod_s p_s^{M'}(x_s^k)\right]. \tag{25}
$$

This problem is addressed in Appendix C. Considering the T best vectors, the updating formula is then

$$
p^{M'}(z|s) = \frac{\mathrm{card}\{X^k \text{ such that } x_s^k = z\}}{T}. \tag{26}
$$

We have introduced upper and lower level algorithms for a stationary target. Now, we will see how our method (upper algorithm coupled with lower algorithm) can be extended to search for a target with Markovian move.

4. An extension to a multiperiod search for a moving target

In the previous sections it was assumed that the target was stationary. So this study is extended to a multiperiod search of a moving target. This means that information about the sensors and the target will be now indexed by time (the period index). The target prior is now trajectorial and we shall consider here a Markovian (target) prior. Furthermore, assuming independence of detections at each period, a target is said undetected for this multiperiod search if it has not been detected at any period of the search. Due to the number of possible target trajectories, the combinatorial complexity of the problem increases dramatically. For a unique sensor the problem has been theoretically solved in [8,9]; while extensions to double layered constraints have been considered in [10]. Practically, all feasible algorithms are based on a forward–backward split introduced by Brown [4]. Similar procedures are also much employed in order to estimate Hidden Markov Models parameters (see e.g. [11]). Here we have to couple our multisensor multizone optimization method with the multiperiod Markovian search. Let us give now a more formal presentation of the problem.

4.1. Forward–backward split

We consider that the search periods are indexed by $t \in \{1, \ldots, T\}$. At each time period t, let $m_t : s \longrightarrow z$ be a mapping allotting sensors to search zones. Our aim is to find both the optimal mappings m_t and the optimal local distributions φ_s^t in order to minimize the non-detection probability, i.e.:

$$
F(\{m_t, \varphi_s^t\}_{t=1}^T) = \sum_{\vec{\omega} \in \Omega} \sum_{z=1}^{Z} \left(\sum_{i \in z} \alpha(\vec{\omega}) \prod_{t=1}^{T} \prod_{s \in m_t^{-1}(z)} \bar{P}_s[\varphi_s^t(c_{z,i})] \right), \tag{27}
$$

where Ω denotes the set of target trajectories, and $\vec{\omega}$ a target trajectory in Ω. This leads to consider the following constrained problem:

$$
\left|\begin{array}{ll}
\min_{\{\forall t, m_t, \varphi_s^t\}} & F(\{m_t, \varphi_s^t\}_{t=1}^T) \\
\text{s.t.} & \forall\, t, \forall z, \forall s \in m_t^{-1}(z),\ \sum_{i \in z} \varphi_s^t(c_{z,i}) \leqslant \Phi_s^t, \\
& \quad \forall i \in z,\ \varphi_s^t(c_{z,i}) \geqslant 0, \\
& \forall\, t,\ m_t \text{ mapping} : s \in S \rightarrow z \in Z.
\end{array}\right. \tag{28}
$$

Fig. 2. The multisensor multizone moving target algorithm.

The problem seems formidable, but it is considerably simplified if we rewrite the objective functional F as follows:

$$
\begin{aligned}
&F(\{m_t,\varphi_s^t\}_{t=1}^T) = \sum_{z=1}^{Z}\left[\sum_{i\in z}\beta_{z,i}^{\tau}\prod_{s\in m_\tau^{-1}(z)}\bar{P}_s(\varphi_s^{\tau}(c_{z,i}))\right],\\
&\text{where}\\
&\beta_{z,i}^{\tau} = \sum_{\vec{\omega}\in\vec{\omega}_{\tau,z,i}}\alpha(\vec{\omega})\prod_{1\leqslant t\leqslant T}^{t\neq\tau}\left(\prod_{s\in m_t^{-1}(z)}\bar{P}_s[\varphi_s^t(c_{z_t,i_t})]\right),\\
&\vec{\omega}_{\tau,z,i} = \{\vec{\omega}\in\Omega \mid \vec{\omega}(\tau) = c_{z,i}\},\\
&\vec{\omega} = (c_{z_1,i_1},\dots,c_{z_\tau,i_\tau},\dots,c_{z_t,i_t},\dots),\\
&\alpha(\vec{\omega}) = \prod_{t=1}^{T-1}\alpha_{t,t+1}(c_{z_t,i_t},c_{z_{t+1},i_{t+1}}).
\end{aligned}
\tag{29}
$$

For a given τ, and considering that the $\beta_{z,i}^{\tau}$ are known (the new "priors"), the multiperiod search problem is put in the standard form (see Section 3.1). Thus, it can be solved by the hierarchical optimization scheme (CE + classical nonlinear optimization) we developed for the monoperiod search, where the "variables" are the $\varphi_s^{\tau}(c_{z,i})$ and the mapping m_τ.

It remains to have a mean to calculate efficiently the $\beta_{z,i}^{\tau}$. To that aim, the trajectorial Markov hypothesis is instrumental and we consider the following splitting of the $\beta_{z,i}^{\tau}$:

$\beta_{z,i}^{\tau} = U^{\tau}(z,i)D^{\tau}(z,i)$ where U and D are recursively defined by:

$$
U^{\tau}(z,i) = \sum_{j\in\tilde{z}}\alpha_{\tau-1,\tau}(j,i)\left(\prod_{s\in m_{\tau-1}^{-1}(\tilde{z})}\bar{P}_s[\varphi_s^{\tau-1}(c_{\tilde{z},i})]\right)U^{\tau-1}(j,\tilde{z}),
$$

$$D^{\tau}(z,i)=\sum_{j\in\tilde{z}}\alpha_{\tau,\tau+1}(i,j)\left(\prod_{s\in m_{\tau+1}^{-1}(\tilde{z})}\bar{P}_s[\varphi_s^{\tau+1}(c_{\tilde{z},i})]\right)D^{\tau+1}(j,\tilde{z}). \tag{30}$$

In the above equation, we denote by $\tilde{z}$, the zones which can be attained conditionally to the hypothesis that the target is in the cell i of the zone z at the period τ and that it has a Markovian prior α. Such a forward–backward split was introduced by Brown [4].

In the following the general algorithm employed for planning a multisensor multizone search for a moving target is described.

4.2. Multisensor multizone moving target algorithm

The algorithm we introduce consists of three optimization levels. A global level, namely the forward–backward level, containing two entangled sublevels a sensor-to-zone allocation optimization level -referred before as the upper level- and a local optimization algorithm allowing to find best resource sharing -referred before as the lower level. At forward–backward level, the goal is to obtain the optimal search plan, i.e. for each time best sensor-to-zone allotment and best resource sharing, by refining successively an initial search plan. It is known as the myopic search plan and defined by not anticipating the motion of the target. It implies that the D (backward) functional is not employed in computing the myopic search plan. Let k be an iteration index such that $D_k^1(1,1)$ is the value of D at first time in the first cell of the first zone for iteration k.

The multisensor multizone moving target algorithm takes the following form:

- **Initialization**:
  - $\forall\tau,\forall z,\forall i,\ D_1^{\tau}(z,i)=1$;
  - $\forall k,\forall z,\forall i,\ F_k^1(z,i)=\alpha_{z,i}$;
- **Iteration** (k index):
  - **Iteration** (τ index):
    - $\forall z,\forall i$, compute the optimal allotment and resource sharing with prior $\beta_{z,i}^{\tau}=U_k^{\tau}(z,i)D_k^{\tau}(z,i)$;
    - $\forall z,\forall i$, compute $U_k^{\tau+1}(z,i)$;
  - $\forall z,\forall i$, compute $D_{k+1}^{\tau}(z,i)$;
  - **Stop**: when the search plan is no more improved.

Fig. 2 illustrates this algorithm. We will see further that the optimal search plan can be far more effective than the myopic search plan.

5. Cooperating search resources and cross-cueing

Up to now, the search efficiency has been only evaluated via a conditional detection functional. Thus, it is a binary (0–1) functional which can appear as rather limitative, in particular if we want to evaluate the benefit of cooperating search resources. This is the case if we consider that a detection by a given search resource can be viewed as a clue and if we want to define an evaluation functional crossing clues coming from various sources.

5.1. A cross-cueing functional

For introducing the cross-cueing functional, we restrict to a given zone z, at a given time-period and we assume that two distinct search resources (1 and 2) are available. Then, the target is said to be "*cross-cued* " if it has been detected by both search resources. Assuming that detections are independent and that the target is actually in cell $c_{z,i}$, the conditional probability $P_{cc}(\varphi_1(c_{z,i}),\varphi_2(c_{z,i}))$ of such event for the cell $c_{z,i}$ is given by

$$P_{cc}(\varphi_1(c_{z,i}),\varphi_2(c_{z,i}))=P(\varphi_1(c_{z,i}))P(\varphi_2(c_{z,i})), \tag{31}$$

where $P(\varphi_1(c_{z,i}))$ represents the probability of detecting the target by means of sensor 1 if it is hidden in cell $c_{z,i}$, while a search effort $\varphi_1(c_{z,i})$ is put on $c_{z,i}$. So, the basic problem we have to solve is

$$\max_{\varphi_1(.),\varphi_2(.)}\ \sum_{i\in z}\alpha_{z,i}P_{cc}(\varphi_1(c_{z,i}),\varphi_2(c_{z,i}))$$
$$\text{s.t.}\quad \sum_{i\in z}\varphi_1(c_{z,i})\leqslant\Phi_1,\quad \sum_{i\in z}\varphi_2(c_{z,i})\leqslant\Phi_2. \tag{32}$$

Notice that the objective functional does not have general concavity properties. However, it is separable. As it is shown in Appendix D, this allows an easy calculation of the dual functional. Thus, the above problem can be efficiently solved via dualization (see Appendix D).

Assume now that k search resources are affected to the zone z, i.e. $m^{-1}(z)=\{l_1,\dots,l_k\}$, then we consider the following objective functional:

$$F(m,\{\varphi_s\, s\in\{1,\dots,S\}\})$$
$$=-\sum_z\sum_{i\in z}\sum_{(s,s'),s\neq s'\in m^{-1}(z)}\alpha_{z,i}c_{s,s',i}P(\varphi_s(c_{z,i}))P(\varphi_{s'}(c_{z,i})). \tag{33}$$

In Eq. (33), the factor $c_{s,s',i}$ stands for the "confirmation" effect gained by a simultaneous detection by search resources s and s'. Let us now introduce multiperiod multizone cross-cueing optimization.

5.2. Multiperiod, multizone optimization

We consider the following cost functional:

$$F(\{m_t,\varphi_s^t\}_{t=1}^T)=-\sum_{\vec{\omega}\in\Omega}\sum_{z=1}^{Z}\left[\sum_{i=1}^{C_z}\alpha_{\vec{\omega}}\prod_{t=1}^{T}\right.$$
$$\left.\times\left(\sum_{(s,s')\in m_t^{-1}(z)}P_{cc}(\varphi_s^t(c_{z,i}),\varphi_{s'}^t(c_{z,i}))\right)\right],$$

with

$$P_{cc}(\varphi_s^t(c_{z,i}),\varphi_{s'}^t(c_{z,i}))=P(\varphi_s^t(c_{z,i}))P(\varphi_{s'}^t(c_{z,i})), \tag{34}$$

where Ω denotes the set of target trajectories, and $\vec{\omega}$ is a target trajectory in Ω. In Eq. (34), the summation is done over all the search resource couples allotted to zone z ($(l,l')\in m_t^{-1}(z)$, $l\neq l'$). Thus, we have to consider the following problem:

$$\left|\begin{array}{ll}\min\limits_{\{\forall t,m_t,\varphi_s^t\}} & F(\{m_t,\varphi_s^t\}_{t=1}^{T}),\\ \text{s.t.} & \forall\, t,\forall z,\forall s\in m_t^{-1}(z),\ \sum\limits_{i=1}^{C_z}\varphi_s^t(c_{z,i})\leqslant\Phi_s^t,\\ & \forall\, i\in z,\ \varphi_s^t(c_{z,i})\geqslant 0,\forall s\in S,\\ & m_t\text{mapping}: s\in S\rightarrow z\in Z.\end{array}\right. \tag{35}$$

Again, the problem is greatly simplified if the objective functional F is rewritten as follows:

$$\left|\begin{array}{l}F(\{m_t,\varphi_s^t\}_{t=1}^{T})=-\sum\limits_{z=1}^{Z}\left[\sum\limits_{i\in z}\beta_{z,i}^{\tau}\sum\limits_{(s,s')\in m_{\tau}^{-1}(z)}P_{cc}(\varphi_s^t(c_{z,i}),\varphi_{s'}^t(c_{z,i}))\right],\\ \text{where }\beta_{z,i}^{\tau}=\sum\limits_{\vec{\omega}\in\vec{\Omega}_{\tau,z,i}}\alpha(\vec{\omega})\prod\limits_{1\leqslant t\leqslant T}^{t\neq\tau}\left(\sum\limits_{(s,s')\in m^{-1}(z_t)}P_{cc}(\varphi_{s_l}^t(c_{z,i}),\varphi_{s_{l'}}^t(c_{z,i}))\right),\\ \vec{\omega}_{\tau,z,i}=\{\vec{\omega}\in\Omega|\vec{\omega}(\tau)=c_{z,i}\},\\ \vec{\omega}=(c_{z_1,i_1},\dots,c_{z_\tau,i_\tau},\dots,c_{z_t,i_t},\dots),\\ \alpha(\vec{\omega})=\prod\limits_{t=1}^{T-1}\alpha_{t,t+1}(c_{z_t,i_t},c_{z_{t+1},i_{t+1}}).\end{array}\right. \tag{36}$$

For a given τ, and considering that the $\beta^{\tau}_{z,i}$ are known (the new "priors"), the multiperiod search problem is put in the standard form. So, the problem is again to calculate the $\beta^{\tau}_{z,i}$. We have recourse to the same splitting, i.e.:

$\beta^{\tau}_{z,i} = U^{\tau}(z,i)D^{\tau}(z,i)$,
where U and D are recursively defined by:

$$U^{\tau}(z,i) = \sum_{j\in\tilde{z}} \alpha_{\tau-1,\tau}(j,i) \times \left(\sum_{(s,s')\in m^{-1}_{\tau-1}(\tilde{z})} P_{\mathrm{cc}}(\varphi^{\tau-1}_{s}(c_{z,i}), \varphi^{\tau-1}_{s'}(c_{z,i})) \right) U^{\tau-1}(j,\tilde{z}),$$

$$D^{\tau}(z,i) = \sum_{j\in\tilde{z}} \alpha_{\tau,\tau+1}(j,i) \times \left(\sum_{(s,s')\in m^{-1}_{\tau+1}(\tilde{z})} P_{\mathrm{cc}}(\varphi^{\tau-1}_{s}(c_{z,i}), \varphi^{\tau+1}_{s'}(c_{z,i})) \right) D^{\tau+1}(j,\tilde{z}). \quad (37)$$

In the above equation, we denote by $\tilde{z}$ the zones which can be attained conditionally to the hypothesis that the target is in the cell i of the zone z at the period τ and that it has a Markovian prior α.

6. Results

This section is devoted to show how our method can be employed practically. All algorithms have been encoded with the MATLAB language and tested on a 1.06 GHz computer.

6.1. Search models

Our goal is to detect a target hidden into the neighborhood of the lake of Laouzas, more precisely near Rieumontagné, in the south of France (see Fig. 3).

First we discretize a map of this region into cells according to ground characteristics (see Fig. 4). Five classes to which cells can belong are identified: forest, water, town, rough land, very rough land or flat land. Cells are then aggregated into four zones ($Z=4$) of equal area ($\forall z$, $C_z = 30$).

Let us now consider search resources. It is assumed that at most six sensors are available for conducting the search. For each sensor s the non-detection probability $\bar{P}_s[\varphi_s(c_{z,i})]$ is given by $\bar{P}_s[\varphi_s(c_{z,i})] = \exp(-w^{s}_{z,i}\varphi_s(c_{z,i}))$ where $w^{s}_{z,i}$ are visibility coefficients. These coefficients depend on the kind of target that is searched and on the kind of land over which the search is conducted (see Table 1).

In the following, parameters of the CE algorithm are constant. The parameter θ is fixed to 0.4 and the initial probability laws for drawing the allotment of sensors to search zones are uniform ($\forall s$, p^{M}_{s} are uniform probability laws). Let us now study optimization of resources for searching a stationary target.

6.2. Stationary target

In this section we restrict the problem to a stationary target. The search model has been given in Section 3.

6.2.1. Detection functional

It is assumed that the target is probably hidden near Rieumontagné (see Fig. 5). The target is searched in a first case by the four first sensors ($S=4$) of Table 1 and in a second case by all the six sensors ($S=6$). Each sensor has an amount of resource equal to 5. In both cases, sensors try to allot a high amount of resource where the target has most chances to be detected due to the prior of the target or to their visibility coefficients. Due to the low amount of resource

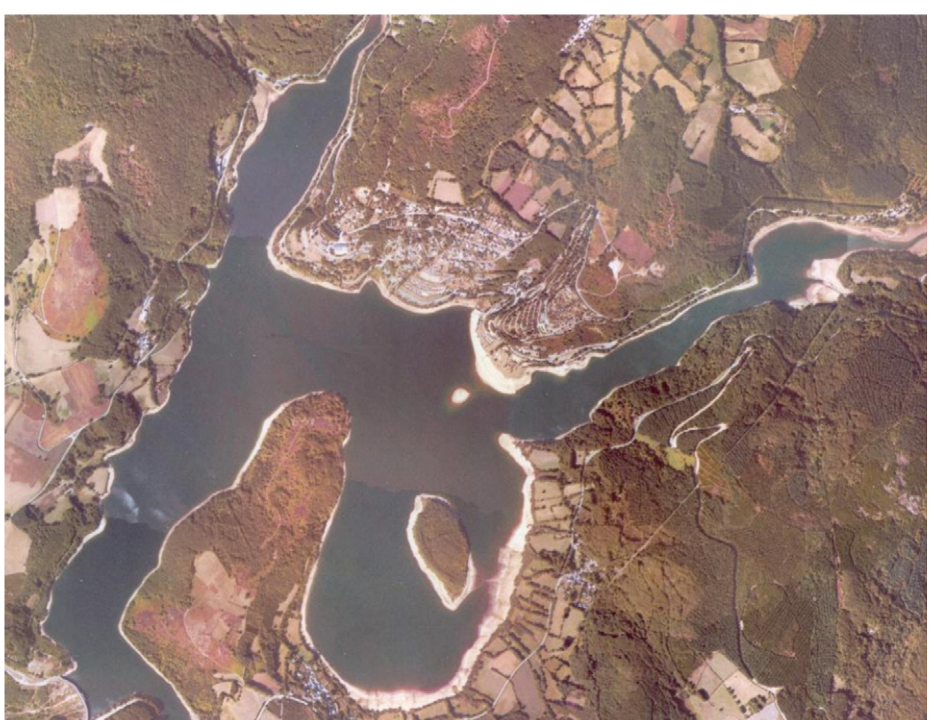

Fig. 3. An aerial photograph of the lake of Laouzas.

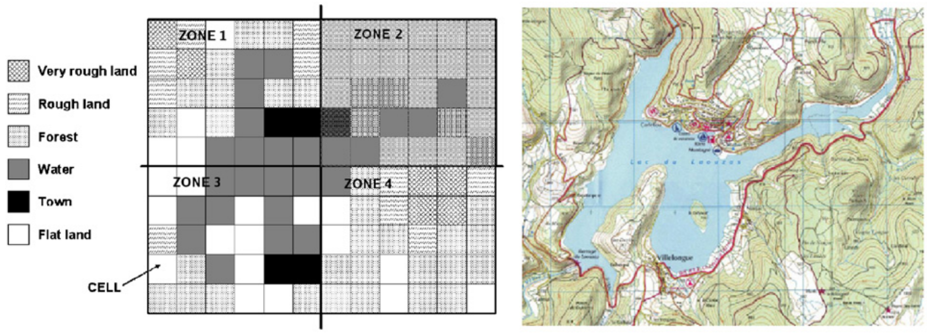

Fig. 4. The space of search: discretized map and map of the lake of Laouzas.

Table 1
Sensor visibility.

	Sensor 1	Sensor 2	Sensor 3	Sensor 4	Sensor 5	Sensor 6
Forest	0.4	0.5	0.6	0.8	0.5	0.1
Water	0.9	0.1	0.1	0.1	0.3	0.5
Town	0.3	0.1	0.4	0.6	0.5	0.2
Rugged land	0.2	0.7	0.8	0.2	0.4	0.6
Very rugged land	0.1	0.6	0.7	0.1	0.3	0.5
Flat land	0.8	0.9	0.1	0.7	0.6	0.2

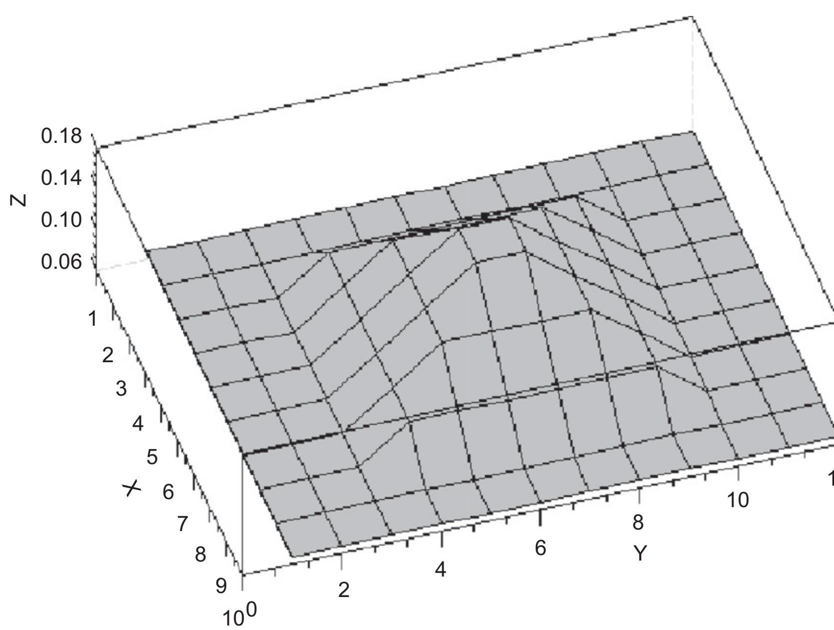

Fig. 5. The target prior.

Table 2
Allotments for stationary target and detection functional.

	Allotments $S=4$				Allotments $S=6$			
Sensor 1	0	0	1	0	0	0	1	0
Sensor 2	0	0	1	0	0	0	1	0
Sensor 3	0	1	0	0	0	1	0	0
Sensor 4	0	0	0	1	0	0	0	1
Sensor 5					1	0	0	0
Sensor 6					1	0	0	0
Non-detection probability	0.8673881				0.8198461			

Table 3
Allotments for stationary target andtwo-sided optimization.

	Allotments $S=4$				Allotments $S=6$			
Sensor 1	0	0	1	0	0	0	1	0
Sensor 2	0	0	0	1	0	0	0	1
Sensor 3	0	0	0	1	0	0	0	1
Sensor 4	0	0	1	0	0	0	1	0
Sensor 5					0	0	0	1
Sensor 6					0	0	1	0
Non-detection probability	0.9218031				0.8892900			

available, each sensor is allotted to the search zone where it has most chances to detect a target. Into its allotment zone, each sensor tries to share resources in order to obtain a uniform probability of detection. Non-detection probability is thus 0.8673881 in the case where four sensors are searching the target, and 0.8198461 in the case where six sensors are employed. Table 2 presents allocation results for both optimizations. Consider for instance the case $S=4$. The i-th row stands for the zone allotment to the i-th sensor. The two-levels hierarchical optimization converges within a few steps of CE. We can notice that the optimal allotment is not injective even if $S=Z$.

6.2.2. Two-sided optimization

Now, prior on the location of the target is considered to be imprecise: the prior over the cells is no longer known. However, general information (at zone level) is known. It is thus assumed that the target has probability 0.25 to hide in each of the four search zones. As before, the target is searched in a first case by the four first sensors of Table 1 and in a second case by all sensors. Each sensor has an amount of resource equal to 5. In both cases, the target tries to hide where sensors have a low visibility. In return, sensors will allot a high quantity of resources at these places. Non-detection probability is then 0.9218031 in the first case and 0.8892900 in the second case. Table 3 presents allocation results for both optimizations.

Optimization of these two stationary problems request between 1 and 2 min according to the size of the search problem.

6.3. Moving target, detection functional

We assume now that the target is Markovian and moves south east from the surroundings of Rieumontagné. The transition matrix describing the target motion is given in Fig. 6 and is assumed to be constant over time. The search is carried out over four time periods by means of the four sensors in first case and six sensors in second case, as presented before and the initial probability distribution on the location of the target (at $t=1$) is the same as before (see Fig. 5). The amount of resources available for each sensor is 5, 6, 7 and 8 for time periods 1, 2, 3 and 4, respectively. Results of the optimization are given in Tables 4 and 5. We can point out that the myopic search plan is less efficient than the optimal search plan (see Tables 4 and 5). However, if the gain in probability is slight, it needs around 20% more resources at each time period in order to obtain the same results with myopic search plans. The enhancement in effectiveness between myopic and optimal search plans is due to the ability of the optimal search plan to anticipate target movement.

Now, let us study how the enhancement of effectiveness is obtained by the multisensor multizone moving target algorithm. We consider an experiment where the six sensors of Table 1 are employed to carry out the search. Each of these sensors has an amount of resource equal to 1 at each time step. The target has the same

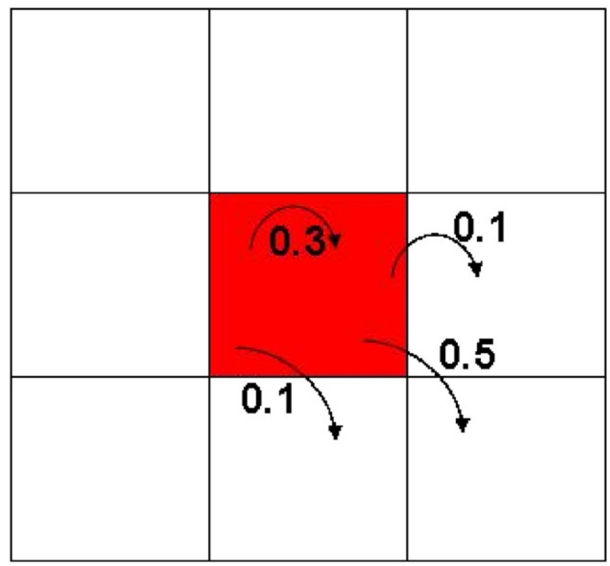

Fig. 6. The target transition probability.

characteristics as before. According to Table 6, we can see that the more important increase in the effectiveness of the search plan is made between iterations 1 and 2 of the multisensor multizone moving target algorithm. It is linked to a change in the allotment of sensors to search zones. After the second iteration, the algorithm gives slighter enhancements thanks to more accurate resource sharing. In order to obtain the same non-detection probability for the myopic search plan than for the optimal one, amount of resources available for each sensor at each time must be increased by around a factor 3. Optimization time for moving target search problems with detection functional is around 20–30 min. This increase of computational time is due to the iteration of CE steps inferred by the FAB algorithm.

6.4. Moving target, cross-cueing functional

In the following we present result of optimization for the cross-cueing problem. The CE algorithm has been adapted in order to handle the double allotment of sensors to search zones: draws with an odd number of sensors allotted to a given zone are rejected. Furthermore we consider that the costs $c_{s,s',i}$ are constant and equal to 1. Here, four sensors are searching the target. Search amount for sensors 1 and 2 are the same: 60, 84, 108, respectively, for time 1, 2, 3. Search amount for sensors 3 and 4 are also equal: 72, 96, 120, respectively, for time 1, 2, 3. Visibility coefficients are the same as before. Results of the optimization are given by Table 7. We can see that here again the gain in probability given by the FAB algorithm is low. However, it allows to save around 15% of resources. In

Table 4
Myopic search plan.

Time	Sensors	Allotment $S=4$				Non-detection probability $S=4$	Allotment $S=6$				Non-detection probability $S=6$
Time 1	Sensor 1	0	0	1	0	0.8673881	0	0	1	0	0.8198461
	Sensor 2	0	0	1	0		0	0	1	0	
	Sensor 3	0	1	0	0		0	1	0	0	
	Sensor 4	0	0	0	1		0	0	0	1	
	Sensor 5						1	0	0	0	
	Sensor 6						1	0	0	0	
Time 2	Sensor 1	0	1	0	0	0.7236255	0	1	0	0	0.6379034
	Sensor 2	0	0	0	1		0	0	0	1	
	Sensor 3	0	1	0	0		0	1	0	0	
	Sensor 4	0	0	0	1		0	0	0	1	
	Sensor 5						0	0	1	0	
	Sensor 6						0	0	0	1	
Time 3	Sensor 1	0	0	1	0	0.5787760	0	0	1	0	0.4727772
	Sensor 2	0	0	1	0		0	0	1	0	
	Sensor 3	0	0	0	1		0	0	0	1	
	Sensor 4	0	0	0	1		0	0	0	1	
	Sensor 5						0	0	0	1	
	Sensor 6						0	1	0	0	
Time 4	Sensor 1	1	0	0	0	**0.4423658**	1	0	0	0	**0.3325033**
	Sensor 2	0	0	0	1		0	0	0	1	
	Sensor 3	0	1	0	0		0	1	0	0	
	Sensor 4	0	0	0	1		0	0	0	1	
	Sensor 5						0	0	1	0	
	Sensor 6						0	1	0	0	

Bold numbers are here to point out the final results of each search plan.

Table 5
Optimal search plan.

Time	Sensors	Allotment $S=4$				Non-detection probability $S=4$	Allotment $S=6$				Non-detection probability $S=6$
Time 1	Sensor 1	1	0	0	0	0.4056058	1	0	0	0	0.2962617
	Sensor 2	0	0	1	0		0	0	1	0	
	Sensor 3	1	0	0	0		1	0	0	0	
	Sensor 4	1	0	0	0		1	0	0	0	
	Sensor 5						1	0	0	0	
	Sensor 6						1	0	0	0	
Time 2	Sensor 1	0	1	0	0	0.4055168	0	1	0	0	0.2958389
	Sensor 2	0	0	0	1		0	0	1	0	
	Sensor 3	0	1	0	0		0	1	0	0	
	Sensor 4	0	0	0	1		0	0	0	1	
	Sensor 5						1	0	0	0	
	Sensor 6						0	0	0	1	
Time 3	Sensor 1	0	0	1	0	0.4053246	0	0	1	0	0.2954212
	Sensor 2	0	0	1	0		0	0	1	0	
	Sensor 3	0	0	0	1		0	0	0	1	
	Sensor 4	0	0	0	1		0	0	0	1	
	Sensor 5						0	0	0	1	
	Sensor 6						0	1	0	0	
Time 4	Sensor 1	0	0	1	0	**0.4052149**	0	0	1	0	**0.2950101**
	Sensor 2	0	0	0	1		0	0	0	1	
	Sensor 3	0	1	0	0		0	1	0	0	
	Sensor 4	0	0	0	1		0	0	0	1	
	Sensor 5						0	0	0	1	
	Sensor 6						0	1	0	0	

Bold numbers are here to point out the final results of each search plan.

order to obtain the same results with a myopic search plan, search resources have to be increased by around 15% at each time period. Optimization time for moving target search problems with cross-cueing functional is around an hour. This increase of computational time compared with moving target search problem with detection functional is due to the complexity of the cross-cueing functional: it requests more time to compute the resource sharing at lower level.

7. Conclusion

The problem we have considered here is the search for a (moving) target within a hierarchical framework. It is difficult because we have to deal with two completely entangled optimization levels involving continuous and discrete optimization in a hierarchical setup. The approach we have proposed is both original and feasible. It is also sufficiently versatile to handle a variety of problems.

Table 6
Evolution of search plans.

Iteration	Time 1				Time 2				Time 3				Time 4				Non-detection probability
Myopic search plan (iteration 1)	0	0	1	0	0	1	0	0	0	0	1	0	1	0	0	0	**0.4423658**
	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	1	
	0	1	0	0	0	1	0	0	0	0	0	1	0	1	0	0	
	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	
Second search plan (iteration 2)	**1**	**0**	**0**	**0**	0	1	0	0	0	0	1	0	**0**	**0**	**1**	**0**	0.4086939
	0	**0**	**1**	**0**	0	0	0	1	0	0	1	0	**0**	**0**	**0**	**1**	
	1	**0**	**0**	**0**	0	1	0	0	0	0	0	1	**0**	**1**	**0**	**0**	
	1	**0**	**0**	**0**	0	0	0	1	0	0	0	1	**0**	**0**	**0**	**1**	
Third search plan (iteration 3)	1	0	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0.4056265
	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	1	
	1	0	0	0	0	1	0	0	0	0	0	1	0	1	0	0	
	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	
Optimal search plan (iteration 4)	1	0	0	0	0	1	0	0	0	0	1	0	0	0	1	0	**0.4052149**
	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	1	
	1	0	0	0	0	1	0	0	0	0	0	1	0	1	0	0	
	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	

Bold numbers are here to point out the final results of each search plan.

Table 7
Myopic and optimal search plans for the cross-cueing problem.

Time	Sensors	Myopic search plan Allotment				Non-detection probability	Optimal search plan Allotment				Non-detection probability
Time 1	Sensor 1	1	0	0	0	0.7359	1	0	0	0	0.4497
	Sensor 2	1	0	0	0		0	1	0	0	
	Sensor 3	0	1	0	0		1	0	0	0	
	Sensor 4	0	1	0	0		0	1	0	0	
Time 2	Sensor 1	1	0	0	0	0.5723	1	0	0	0	0.4487
	Sensor 2	1	0	0	0		0	1	0	0	
	Sensor 3	0	1	0	0		1	0	0	0	
	Sensor 4	0	1	0	0		0	1	0	0	
Time 3	Sensor 1	1	0	0	0	**0.4737**	1	0	0	0	**0.4478**
	Sensor 2	1	0	0	0		0	1	0	0	
	Sensor 3	0	1	0	0		1	0	0	0	
	Sensor 4	0	1	0	0		0	1	0	0	

Bold numbers are here to point out the final results of each search plan.

Thus, extending this hierarchical discrete-continuous optimization framework to the moving target case is rather straightforward. The effectiveness of our approach has been proved on realistic simulations.

Appendix A. Solving the basic search problem

Let f be a separable functional i.e. $\mathbf{X} \in \mathbb{R}^n \to f(\mathbf{X}) = \sum_{i=1}^n f_i(x_i)$ and the constrained (primal) optimization problem:

$$\mathscr{P} \begin{cases} \min\limits_{\mathbf{X}} \quad f(\mathbf{X}) \\ \text{s.t.} \quad x_i \geqslant 0 \quad \text{and} \quad \sum\limits_{i=1}^n x_i = 1. \end{cases} \tag{38}$$

From KKT conditions, we know that there exists Lagrangian multipliers $\{\underline{\mu}_1, \ldots, \underline{\mu}_n\} \in (\mathbb{R}^+)^n$ and $\underline{\lambda} \in \mathbb{R}$ such that the following equalities are satisfied at the optimum of the primal problem $\mathscr{P}$:

$$KKT \begin{cases} f_i'(\underline{x}_i) - \underline{\mu}_i + \underline{\lambda} = 0, \\ \underline{\mu}_i \underline{x}_i = 0, \quad \forall i = 1, \ldots, n. \end{cases} \tag{39}$$

The key for solving the primal problem is to consider the dual functional $\psi(\lambda)$ defined by

$$\begin{cases} \psi(\lambda) = \inf\limits_{\mathbf{X} \in (\mathbb{R}^+)^n} \mathscr{L}(\mathbf{X}, \lambda) \quad (\lambda \in \mathbb{R}), \\ \mathscr{L}(\mathbf{X}, \lambda) = \sum\limits_{i=1}^n f_i(x_i) + \lambda \left(\sum\limits_{i=1}^n x_i - 1 \right). \end{cases} \tag{40}$$

If, furthermore, we assume that a -1 term has been added to the classical non-detection functional—only for simplifying the expression of the dual function $\psi(\lambda)$, i.e. $f_i(x_i) = \alpha_i (\mathrm{e}^{-w_i x_i} - 1)$—KKT conditions yield

$$\begin{cases} -\alpha_i w_i \mathrm{e}^{-w_i \underline{x}_i(\lambda)} + \lambda = 0, \quad \forall i \text{ s.t. } \underline{x}_i(\lambda) > 0, \\ -\alpha_i w_i + \lambda \geqslant 0, \quad \forall i \text{ s.t. } \underline{x}_i(\lambda) = 0 \end{cases} \tag{41}$$

and the dual functional stands as follows:

$$\psi(\lambda) = -\sum_{i=1}^n \gamma_i \left(1 - \frac{\lambda}{\alpha_i w_i}\right)^+ + \lambda \left(\sum_{i=1}^n \frac{1}{w_i} \left[\ln\left(\frac{p_i w_i}{\lambda}\right) \right]^+ - 1 \right). \tag{42}$$

Let λ^* be the unique value of λ maximizing $\psi(\lambda)$, then the solution to the primal problem is

$$\underline{x}_i^* = \frac{1}{w_i} \left[\ln\left(\frac{\alpha_i w_i}{\lambda^*}\right) \right]^+. \tag{43}$$

Appendix B. The local search game

The problem we have to solve is

$$\max_{\{\alpha_{i|z}\}} \min_{\varphi_s} D(\varphi_s, \{\alpha_{i|z}\}),$$

where

$$D(\varphi_s, \{\alpha_{i|z}\}) = \sum_{i \in z} \alpha_{i|z} \prod_{s \in m^{-1}(z)} \exp(-w^s_{z,i}\varphi_s(c_{z,i})),$$

$$\text{s.t.} \begin{cases} \sum_{i \in z} \varphi_s(c_{z,i}) = \Phi_s, \\ \sum_i \alpha_{i|z} = 1. \end{cases} \tag{44}$$

Thus, the above maximin optimization problem can be splitted into two subproblems:

$$\begin{cases} \text{for the target:} \quad \max_{\alpha_{i|z}} \min_{\varphi_s} D(\varphi_s, \{\alpha_{i|z}\}), \\ \text{for the sensors:} \quad \min_{\varphi_s} \max_{\alpha_{i|z}} D(\varphi_s, \{\alpha_{i|z}\}). \end{cases} \tag{45}$$

This is a maximin problem and it is known that if there is a saddle point $(\varphi^*_s, \{\alpha_{i|z}\}^*)$, such that

$$D(\varphi^*_s, \{\alpha_{i|z}\}) \leqslant D(\varphi^*_s, \{\alpha_{i|z}\}^*) \leqslant D(\varphi_s, \{\alpha_{i|z}\}^*), \tag{46}$$

then

$$\max_{\alpha_{i|z}} \min_{\varphi_s} D(\varphi_s, \{\alpha_{i|z}\}) = \min_{\varphi_s} \max_{\alpha_{i|z}} D(\varphi_s, \{\alpha_{i|z}\}) D(\varphi^*_s, \alpha^*_{u|z}) \\ = D(\varphi^*_s, \{\alpha_{i|z}\}^*). \tag{47}$$

In the case where only one sensor is allotted to the search zone, the optimal strategies of the target and of the sensors are simply given by a classical result [3]:

$$\varphi^*_s(c_{z,i}) = \frac{\Phi_s}{w^s_{z,i}} \frac{1}{\sum_{i \in z} {w^s_{z,i}}^{-1}}, \quad \alpha^*_{i|z} = \frac{1}{w^s_{z,i}} \frac{1}{\sum_{i \in z} {w^s_{z,i}}^{-1}}. \tag{48}$$

However, the problem is somewhat complicated when multiple sensors can be allotted to the same search zone. Let us now examine the problems we have to solve (see Eq. (45)).

Optimization of the target strategy: For a given zone z, the optimization problem for the target strategy is

$$\min_{\alpha_{i|z}} \quad -D(\varphi^*_s, \alpha_{i|z}) \\ \text{s.t.} \quad \sum_i \alpha_{i|z} = 1 \quad \text{and} \quad \alpha_{i|z} \geqslant 0. \tag{49}$$

KKT optimality conditions then yield

$$\begin{cases} \text{If } \alpha^*_{i|z} > 0 \text{ then } \prod_s \exp(-w^s_{i,z}\varphi^*_s(c_{z,i})) = \lambda, \\ \text{If } \alpha^*_{i|z} = 0 \text{ then } \prod_s \exp(-w^s_{i,z}\varphi^*_s(c_{z,i})) < \lambda, \end{cases} \tag{50}$$

Optimization of the sensor searching strategy: The problem we have to solve is

$$\min_{\varphi_s} \quad D(\varphi_s, \alpha^*_{i|z}) \\ \text{s.t.} \quad \sum_i \varphi_s(c_{z,i}) = 1 \quad \text{and} \quad \varphi_s(c_{z,i}) \geqslant 0. \tag{51}$$

KKT optimality conditions then yield

$$\begin{cases} \text{If } \varphi^*_s(c_{z,i}) > 0 \text{ then } w^s_{z,i}\alpha^*_{i|z} \exp(-w^s_{z,i}\varphi^*_s(c_{z,i}))R_s(c_{z,i}) = -\nu_s, \\ \text{If } \varphi^*_s(c_{z,i}) = 0 \text{ then } w^s_{z,i}\alpha^*_{i|z} \exp(-w^s_{z,i}\varphi^*_s(c_{z,i}))R_s(c_{z,i}) \\ \quad = -\nu_s - \tau_{i,s} \ (\tau_i \geqslant 0), \end{cases}$$

with

$$R_s(c_{z,i}) = \prod_{s' \neq s} \exp(-w^{s'}_u \varphi^*_{s'}(c_{z,i})). \tag{52}$$

Thus, for a given sensor s, the dual functional Ψ is

$$\Psi(\varphi^*_s, \alpha_{.|z}) = \sum_{i|\varphi^*_s(c_{z,i})>0} -\frac{\nu_s}{w^s_{z,i}} + \sum_{i|\varphi^*_s(c_{z,i})=0} -\left(\frac{\nu_s + \tau_{i,s}}{w^s_{z,i}}\right) \\ + \nu_s \left(\Phi_s - \sum_i \varphi^*_s(c_{z,i})\right). \tag{53}$$

The maximum of Ψ w.r.t. the $\{\tau_{i,s} \geqslant 0\}$ multipliers is obviously achieved for: $\forall i,\ \tau_{i,s} = 0$. So, at the optimum we have (see Eq. (52))

$$w^s_{z,i}\alpha^*_{i|z} \exp(-w^s_{z,i}\varphi^*_s(c_{z,i}))R_s(c_{z,i}) = -\nu_s, \quad \forall s \in m^{-1}(z). \tag{54}$$

Now the multiplier ν_s is necessarily strictly negative, since otherwise we would have the equality $\alpha^*_{i|z} w^s_{z,i} = 0$ whatever the cell i. Now, it assumed that the visibility coefficients $w_{z,i}$ cannot be null, otherwise the target would hide for sure in these cells. This means that $\alpha^*_{i|z}$ is strictly positive whatever the cell $c_{i,z}$. Thus, from Eqs. (50), we have

$$\prod_s \exp(-w^s_{i,z}\varphi^*_s(c_{z,i})) = \lambda, \quad \forall c_{z,i}. \tag{55}$$

Appendix C. Minimizing the Kullback information

Minimizing the Kullback information $\mathscr{K}$ leads to consider the following continuous optimization problem:

$$\max_{p^{M'}} \mathscr{K} \triangleq \sum_{k=1}^{T} \ln\left[\prod_s p^{M'}_s(x^k_s)\right] \quad \text{s.t.} \ \sum_z p^{M'}_s(z) = 1. \tag{56}$$

It is worth making the objective functional $\mathscr{K}$ more explicit i.e.:

$$\mathscr{K} = \sum_z \left(\sum_{s|x^k_s = z}\left[\ln \prod_{s=1}^{S} p^{M'}(z|s)\right]\right) \\ = \sum_z \sum_s [\text{card } \{X^k | x^k_s = z\} \ln(p^{M'}(z|s))]. \tag{57}$$

Denoting $a_{z,s} \triangleq p^{M'}(z|s)$ and $b^k_{z,s} \triangleq$ card $(X^k | x^k_s = z)$, the continuous optimization we have to solve now stands as follows:

$$\max_{a_{z,s}} \quad \sum_z \sum_s b^k_{z,s} \ln(a_{z,s}) \\ \text{s.t.} \quad \sum_z a_{z,s} = 1, s = 1, \ldots, S, 0 \leqslant a_{z,s} \leqslant 1. \tag{58}$$

It is easier to consider that inequality constraints be implicitly taken into account. Denoting Λ an S-dimensional vector made of scalar Lagrange multiplier, we have to consider the maximization of the following Lagrangian functional:

$$\mathscr{L} = \sum_z \sum_s b^k_{z,s} \ln(a_{z,s}) + (\Lambda^{\mathrm{T}} A - 1), \\ \text{where } A^{\mathrm{T}} = \left(\ldots, \sum_z a_{z,s}, \ldots\right), \quad \Lambda^{\mathrm{T}} = (\ldots, \lambda_s, \ldots). \tag{59}$$

By differentiating the Lagrangian $\mathscr{L}$, we obtain the following necessary condition:

$$b^k_{z,s} \frac{1}{a_{z,s}} + \lambda_s = 0, \quad \forall z. \tag{60}$$

Thus we have $a_{z,s} = c_s b^k_{z,s}$, where c_s is a constant. The positivity constraints (of the $a_{z,s}$) are taken into account via the positivity of the constant c_s, while the equality constraint:

$$\sum_z a_{z,s} = 1, \tag{61}$$

leads to

$$p^{M'}(z|s)=\frac{b^k_{z,s}}{T}. \tag{62}$$

Appendix D. Maximizing the cross-cueing functional

We restrict here to a single zone z, divided into cells i. Two search resources can be deployed on this zone and we consider that the target detection is "*confirmed*" if it is detected by the searchers. So, we consider now the following optimization problem $\mathscr{P}$:

$$\mathscr{P}\left|\begin{array}{l}\min \quad -P_{cc} \quad \text{with } P_{cc}=\sum_{i\in z}\alpha_i p(\varphi_1(c_{z,i}))p(\varphi_2(c_{z,i})),\\ \text{s.t.} \quad \sum_{i\in z}\varphi_1(c_{z,i})\leqslant \Phi_1, \quad \sum_{i\in z}\varphi_2(c_{z,i})\leqslant \Phi_2,\\ \qquad \varphi_1(c_{z,i})\geqslant 0, \quad \varphi_2(c_{2,i})\geqslant 0.\end{array}\right. \tag{63}$$

In Eq. (63), $\varphi_1(c_{z,i})$ (resp. $\varphi_2(c_{z,i})$) is the search effort put on the cell i by searcher 1 (resp. searcher 2). As previously, it is assumed that $p(\varphi_1(c_{z,i}))$ (resp. $p(\varphi_2(c_{z,i}))$) is the conditional probability that searcher 1 (resp. searcher 2) detects a target in the cell i, and thus:

$$\begin{cases} p(\varphi_1(c_{z,i}))=1-\exp(-w_{1,i}\varphi_1(c_{z,i})),\\ p(\varphi_2(c_{z,i}))=1-\exp(-w_{2,i}\varphi_2(c_{z,i})).\end{cases} \tag{64}$$

We call P_{cc} (see Eq. (63)), the cross-cueing functional. Generally, this cross-cueing functional is not concave everywhere. However, it has the great advantage to be separable. Considering the primal problem (Eq. (63)) leads to consider the following Lagrangian functional:

$$\mathscr{L}(\lambda,\mu)=-\sum_{i\in z}\alpha_i p(\varphi_1(c_{z,i}))p(\varphi_2(c_{z,i}))+\lambda\left(\sum_{i\in z}\varphi_1(c_{z,i})-\Phi_1\right) + \mu\left(\sum_{i\in z}\varphi_2(c_{z,i})-\Phi_2\right). \tag{65}$$

The Lagrange multipliers λ and μ are positive, and the dual functional $\Psi(\lambda,\mu)$ is defined by

$$\Psi(\lambda,\mu)=\min_{\boldsymbol{\varphi}_1,\boldsymbol{\varphi}_2\in\mathbb{R}^{2n}_+}\mathscr{L}(\lambda,\mu)(\boldsymbol{\varphi}_1,\boldsymbol{\varphi}_2),$$

$$\boldsymbol{\varphi}_1=(\varphi_1(c_{z,1}),\ldots,\varphi_1(c_{z,i}),\ldots,\varphi_1(c_{z,n})),$$
$$\boldsymbol{\varphi}_2=(\varphi_2(c_{z,1}),\ldots,\varphi_2(c_{z,i}),\ldots,\varphi_2(c_{z,n})). \tag{66}$$

For given values of λ and μ, denote $\boldsymbol{\varphi}^*_1$ and $\boldsymbol{\varphi}^*_2$ the search vectors which minimize $\mathscr{L}(\lambda,\mu)$. Assume furthermore that $\varphi^*_1(c_{z,i})$ is strictly positive. What are the consequences? Considering the minimization of $\mathscr{L}(\lambda,\mu)$ on the convex domain $\mathbb{R}^{2n}_+$, a necessary condition for the vector $\boldsymbol{\varphi}^*\triangleq(\boldsymbol{\varphi}^*_1,\boldsymbol{\varphi}^*_2)$ to be a (local) minimum of $\mathscr{L}(\lambda,\mu)$ on the convex set $S\triangleq\mathbb{R}^{2n}_+$ is

$$\begin{cases}-\nabla\mathscr{L}(\lambda,\mu)(\boldsymbol{\varphi}^*)\in\mathscr{N}(S,\boldsymbol{\varphi}^*),\\ \text{where}\\ \mathscr{N}(S,\boldsymbol{\varphi}^*)\triangleq\{\mathbf{s}\in S|\langle\mathbf{s},\mathbf{s}'-\boldsymbol{\varphi}^*\rangle\leqslant 0,\forall\mathbf{s}'\in S\}.\end{cases} \tag{67}$$

$\mathscr{N}(S,\boldsymbol{\varphi}^*)$ is the normal cone to S, in $\boldsymbol{\varphi}^*$. Considering Eq. (67), the following (necessary) condition holds if we assume that $\varphi^*_1(c_{z,i})>0$:

$$\frac{\partial}{\partial\varphi_1(c_{z,i})}\mathscr{L}=-\alpha_i\frac{\partial}{\partial\varphi_1(c_{z,i})}p(\varphi_1(c_{z,i}))p(\varphi_2(c_{z,i}))+\lambda=0 \quad \text{for } \varphi_1(c_{z,i})=\varphi^*_1(c_{z,i}),$$

or

$$\alpha_i w_{1,i}(1-p(\varphi^*_1(c_{z,i})))p(\varphi^*_2(c_{z,i}))=\lambda. \tag{68}$$

The multiplier λ being non-negative, the following implications result straightforwardly from Eq. (67):

$$\begin{cases}\varphi^*_1(c_{z,i})>0\Rightarrow\varphi^*_2(c_{z,i})>0,\\ \alpha_i\,w_{1,i}>0.\end{cases} \tag{69}$$

Moreover, since μ is also positive, we have $\varphi^*_1(c_{z,i})>0 \Leftarrow\Rightarrow \varphi^*_2(c_{z,i})>0$. Denoting $X_{1,i}\triangleq p(\varphi^*_1(c_{z,i}))$, if $\varphi^*_1(c_{z,i})>0$, then $X_{1,i}$ and $X_{2,i}$ are the solutions of the following non-linear system:

$$\begin{cases}\alpha_i w_{1,i}(1-X_{1,i})X_{2,i}=\lambda,\\ \alpha_i w_{2,i}(1-X_{2,i})X_{1,i}=\mu,\\ X_{1,i},X_{2,i}\in[0,1].\end{cases} \tag{70}$$

Subtracting the second row to the first in Eq. (70), we obtain

$$X_{2,i}=X_{1,i}+\left(\frac{\lambda}{\alpha_i w_{1,i}}-\frac{\mu}{\alpha_i w_{2,i}}\right),$$

and the second order equation:

$$-\alpha_i w_{1,i}X^2_{1,i}+\left(\alpha_i w_{1,i}-\lambda+\mu\frac{w_{1,i}}{w_{2,i}}\right)X_{1,i}-\mu\frac{w_{1,i}}{w_{2,i}}=0. \tag{71}$$

So, the problem is reduced to searching the roots of the (above) second order equation lying in the $[0,1]$ interval. It is easily shown that there is at most one root in this interval. It simply remains to test which couple $(X_{1,i},X_{2,i})$ or $(0,0)$ provides the lower value of $\mathscr{L}(\lambda,\mu)$, for every cell i of the zone z. Thus, we see that the separability of the cross-cueing functional P_{cc} greatly simplifies the optimization problem.

The dual functional $\Psi(\lambda,\mu)$ can now be calculated via Eq. (71). This is simply a two-dimensional functional, which is furthermore concave. The following ascent algorithm is considered:

$$\begin{pmatrix}\lambda_{k+1}\\ \mu_{k+1}\end{pmatrix}=\begin{pmatrix}\lambda_k\\ \mu_k\end{pmatrix}+\rho_k g(\boldsymbol{\varphi}^*_1(\lambda_k,\mu_k),\boldsymbol{\varphi}^*_2(\lambda_k,\mu_k)),$$

where

$$g(\boldsymbol{\varphi}^*_1(\lambda_k,\mu_k),\boldsymbol{\varphi}^*_2(\lambda_k,\mu_k))=\begin{pmatrix}\sum_{i\in z}\varphi^*_1(c_{z,i})(\lambda_k,\mu_k)-\Phi_1\\ \sum_{i\in z}\varphi^*_2(c_{z,i})(\lambda_k,\mu_k)-\Phi_2\end{pmatrix}. \tag{72}$$

References

[1] Rubinstein RY, Kroese D. The cross-entropy method: a unified approach to combinatorial optimization, Monté Carlo simulation, and machine learning. Berlin: Springer; 2004.
[2] de Guenin J. Optimum distribution of effort: an extension of the Koopman theory. Operations Research 1961;9(1):1–7.
[3] Nakai T. Search models with continuous effort under various criteria. Journal of Operations Research Society of Japan 1987;31(3):335–51.
[4] Brown SS. Optimal search for a moving target in discrete time and space. Operations Research 1979;32(5):1107–15.
[5] Koopman BO. Search and its optimization. Mathematical Monthly 1979;7: 527–40.
[6] Costa A, Dafydd O, Kroese D. Convergence properties of the cross-entropy method for discrete optimization. Operations Research Letters 2007;35(5): 573–80.
[7] Dambreville F. Cross-entropy method: convergence issues for extended implementation. In: Proceedings of the rare event simulation conference (RESIM 2006), Bamberg, Germany, 2006.
[8] Stone LD. Necessary and sufficient conditions for optimal search plans for moving targets. Mathematics of Operations Research 1979;4(4):431–40.
[9] Stromquist WR, Stone LD. Constrained optimization of functionals with search theory applications. Mathematics of Operations Research 1981;6(4):518–27.
[10] Hohzaki R, Iida K. A concave minimization problem with double layers of constraints on the total amount of resources. Journal of the Operations Research Society of Japan 2000;43(1):109–27.
[11] Ephraim Y, Merhav N. Hidden Markov processes. IEEE Transactions on Information Theory 2002;48:1518–69.
[12] Le Cadre J-P, Souris G. Searching tracks. IEEE Transactions on Aerospace and Electronic Systems 2000;36(4):1149–66.