

Continuous Management of Modes and Resources in Detection of a Moving Target

Frédéric Dambreville, Jean-Pierre Le Cadre

► **To cite this version:**

Frédéric Dambreville, Jean-Pierre Le Cadre. Continuous Management of Modes and Resources in Detection of a Moving Target. [Research Report] RR-3952, INRIA. 2000. <inria-00072697>

HAL Id: inria-00072697

<https://hal.inria.fr/inria-00072697>

Submitted on 24 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

***Continuous Management of Modes and
Resources in Detection of a Moving Target***

Frédéric Dambreville, Jean-Pierre Le Cadre

N°3952

Juin 2000

THÈME 3



*Rapport
de recherche*



Continuous Management of Modes and Resources in Detection of a Moving Target

Frédéric Dambreville, Jean-Pierre Le Cadre

Thème 3 — Interaction homme-machine,
images, données, connaissances
Projet Vista

Rapport de recherche n° 3952 — Juin 2000 — 35 pages

Abstract: This paper deals with optimization of divisible resources aimed to the detection of a moving target following a Markovian movement or a conditionally deterministic motion. Our work extends Brown's spatial optimization method. By use of a generalized linear formalism, we develop a method for optimizing the search efforts both spatially and temporally (modeling resource renew), with management of multiple resource types or multi-modes resources. Such optimization involves also the fusion of several detection devices, in order to make them work together efficiently.

Key-words: Search theory, Detection, Optimization, resource allocation, Sensor management

(Résumé : tsvp)

Gestion continue de modes et de ressources pour la détection de cibles mouvantes

Résumé : Ce rapport traite de l'optimisation de ressources sécables appliquées à la détection d'une cible soumise à un mouvement Markovien ou conditionnellement déterministe. Notre recherche étend la méthode d'optimisation spatiale de Brown. En utilisant un formalisme linéaire généralisé, nous avons développé une méthode d'optimisation tant spatiale que temporelle (modélisation de ressources renouvelables) de l'effort de recherche, avec gestion de plusieurs types de ressources ou de plusieurs modes de fonctionnement. Ainsi se trouvent impliqués de nombreux dispositifs de détection qu'il est nécessaire de faire coopérer efficacement.

Mots-clé : Théorie de la recherche, Détection, Optimisation, allocation de ressources, Gestion de capteurs

1 Introduction

This paper deals with the management of modes and resource for detecting a moving target. This means that the searcher has multiple detection devices (e.g. radar, IR, sonar) at its disposal. Each elementary device can also work on various modes. These modes can be related to visibility factors (e.g. radar range, sector size for IR, etc.) and/or to resource constraints (e.g. resource renew, discretion constraints).

Since the problem consists in optimizing detection resources, let us briefly recall the major statement of search theory. Search theory is the discipline that treats the problem of how best to search for an object (the target) when the amount of searching effort is limited and only probabilities of the object's possible location are given. An important literature has been devoted to the subject; the interested reader may consult various extensive surveys [10][3], introductory texts [4] and specialized books.

In this set-up a detection (or search) problem is characterized by three pieces of data: (i) the probabilities of the searched target being in various possible positions, (ii) the local detection probability that a particular amount of local search effort could detect the target, (iii) the total amount of searching effort available. The problem is to find the optimal distribution of the detection (search) effort that maximizes the probability of detection.

The rapid growth of the search theory literature is chronicled in reference [3]. For moving targets problems, decisive progress have been made in developing search strategies for a moving (Markovian) target. In particular, Brown [7] and Washburn [8] have proposed an iterative algorithm (FAB algorithm) in which the motion space and the time frame have been discretized. In this approach, the search effort available in each period is bounded above by a constant that does not depend on the allocations made during any other periods¹.

Here, although the general formalism of search theory will be used subsequently, our problem strongly departs from the original setup. For a multi-period search, the final result is tightly related to the various detectors and modes. Thus, optimizing this sequence is quite challenging. This optimization must be performed

¹Practically, this constraint strongly reduces the interest of the FAB algorithm and limits its application to simple detection systems

both temporally and spatially.

Here we shall develop a versatile formalism capable of handling the management of a very general detection system. This is the aim of the formalism of Lateral Splitting (L.S.) which, jointly with the resource operator definitions, constitute the backbone of this paper. Using it, it is possible to manage a wide variety of resource allocation problems like multi-mode, multi-resource scheduling, resource dependency and to deal with any combination of them. The resource modelings are presented in section 2 as well as the various ways to combine them. The elementary operators (\oplus , \otimes , scalar product, \succ) are then presented in the lateral splitting framework. In this area, the operator \succ deserves a special mention since its aim is to represent conditional dependency between two LSs. The general properties of these operators are then stated. Using the LS and operator formalism, we are in position to develop the general form of an algorithm optimizing the search efforts² (see section 3). Finally, the utility of the general formalism and the related optimized algorithm is illustrated on several examples involving complex detection systems (see section 4).

Definitions: A target moving in a search space E is to be detected. The detection is achieved during T periods, each period being brief enough to satisfy the stationarity hypothesis of the target for the period. The trajectory $\vec{x} = (x_1, \dots, x_T) \in E^T$ of the target is described by a probabilistic prior about \vec{x} , given by a probabilistic density $\alpha(\vec{x})$. A Markovian assumption will be made for algorithmic reasons; α is taken as a product of elementary densities:

$$\alpha(\vec{x}) = \alpha_{1,2}(x_1, x_2) \times \dots \times \alpha_{T-1,T}(x_{T-1}, x_T). \quad (1)$$

A given amount of search effort ϕ_k is available at each period k . Those efforts are constants of the problem. At each period k , the effort ϕ_k may be distributed along E to accomplish the search. The local search effort, applied to the point $x_k \in E$ at time k , is denoted $\varphi_k(x_k)$. Constraints (2) are considered:

$$\forall k \in \{1, \dots, T\}, \varphi_k \geq 0 \text{ and } \int_E \varphi_k(x) dx \leq \phi_k. \quad (2)$$

²The aim of this algorithm is to combine the simplicity and efficiency of the FAB algorithm with the versatility of the LS formalism

or equivalently:

$$\forall k \in \{1, \dots, T\}, \varphi_k \geq 0 \text{ and } \int_E \varphi_k(x) dx = \phi_k . \quad (3)$$

The local efforts φ directly command the local detection probability. We call $p_{k,x_k}(\varphi_k(x_k))$ the probability not to detect within the period k , when target location is x_k . The law of diminishing return is assumed. Thus for x_k fixed, $p'_{k,x_k} < 0$ and $p''_{k,x_k} > 0$. The problem is to find φ so as to minimize the global probability of non detection $\mathbf{P}_{nd}(\varphi)$ under the constraint (3). Independence hypothesis on elementary detections yields:

$$\mathbf{P}_{nd}(\varphi) = \int_{E^T} \alpha(\vec{x}) \prod_{k=1}^{k=T} p_{k,x_k}(\varphi_k(x_k)) \prod_{k=1}^{k=T} dx_k . \quad (4)$$

Brown's method: Brown obtained a convergence toward the optimal solution, by successively optimizing alone each period with the other fixed. More precisely, for a particular period κ , $\mathbf{P}_{nd}(\varphi)$ can also be written:

$$\begin{aligned} \mathbf{P}_{nd}(\varphi) &= \int_E \beta_\kappa^\varphi(x_\kappa) p_{\kappa,x_\kappa}(\varphi_\kappa(x_\kappa)) dx_\kappa , \text{ where:} \\ \beta_\kappa^\varphi(x_\kappa) &= \int_{E^{T-1}} \alpha(\vec{x}) \prod_{\substack{k \neq \kappa \\ 1 \leq k \leq T}} (p_{k,x_k}(\varphi_k(x_k)) dx_k) . \end{aligned} \quad (5)$$

This shows that, when the search efforts are fixed for all periods, except for a period κ , the optimization problem becomes the following 1-period problem:

$$\begin{aligned} \text{Minimize } P_{nd}(\varphi_\kappa) &= \int_E \beta_\kappa^\varphi(x) p_{\kappa,x}(\varphi_\kappa(x)) dx , \\ \text{subject to } \int_E \varphi_\kappa(x) dx &= \phi_\kappa \text{ and } \varphi_\kappa \geq 0 . \end{aligned} \quad (6)$$

The following optimality conditions scaled by parameter η (equations of de Guenin [6]) are obtained:

$$\begin{cases} \beta_\kappa^\varphi(x) p'_{\kappa,x}(\varphi_\kappa(x)) = \eta_\kappa , & \text{if } \beta_\kappa^\varphi(x) > \eta_\kappa / p'_{\kappa,x}(0) \\ \varphi_\kappa(x) = 0 , & \text{else .} \end{cases} \quad (7)$$

Since $p'_{\kappa,x}$ is strictly monotonic, the above equations are invertible, while η is not specified. An uniformly increasing function φ_κ^η is obtained. The increaseness of

φ_κ^η makes $\int_E \varphi_\kappa^\eta(x) dx$ increasing. The good value for η_κ is then obtained by a dichotomic process on φ_κ^η . The whole process requires only a few iterations. It uses basically the Markovian assumption relative to α , so as to drastically reduce the computation requirements for the integral (5), (**F**orward **A**nd **B**ackward algorithm [7][8]). This idea will be instrumental again in our algorithm.

2 Generalized models

Brown's algorithm assumes ϕ to be known and deals only about resources spatial sharing. Optimization on ϕ_k is however a worthwhile issue. Time splitting of a global amount of non renewable resources, or allocation problems on renewable resources are examples. A more general problem, called lateral splitting, will be presented. This problem intends to involve multi-resource and multi-mode scheduling together with spatio-temporal optimization of resources. To do that, it combines spatial optimizations (as usually) and global optimizations, which act at the index level (some indexes may not be temporal). Before defining a formalism for lateral splitting, some examples are given. They should enlighten on how lateral splitting constitutes a common canvas for modeling both the nature of particular resources and the combinations of different resources. For our examples, we will restrict to a search on $T = 4$ periods.

Modeling resources: A resource is certainly characterized by its non detection functions p , but it is not sufficient. The relations between global resources ϕ_k have to be set. In Brown's formalism, this setting is given only by the constant vector ϕ . It is a poor formalism, since constantness forbids interaction between periods. For example, if we want to define a non renewable resource, which total amount, say 100, has to be shared between the 4 periods, it is necessary to handle ϕ as a variable and not as a constant. With such non renewable resource, resource balance decreases of ϕ_k at each period k . Since balance has to be positive and resources are limited by 100, variables ϕ obey to the 4 constraints:

$$\begin{cases} \phi_1 \leq 100, \\ \phi_1 + \phi_2 \leq 100, \\ \phi_1 + \phi_2 + \phi_3 \leq 100, \\ \phi_1 + \phi_2 + \phi_3 + \phi_4 \leq 100, \end{cases}$$

which reads in a matrix formalism:

$$A\phi \leq \psi \text{ with } A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix} \text{ and } \psi = \begin{pmatrix} 100 \\ 100 \\ 100 \\ 100 \end{pmatrix}$$

It is noteworthy that, since $\phi \geq 0$, ϕ is sufficiently characterized by the only constraint $\phi_1 + \phi_2 + \phi_3 + \phi_4 \leq 100$. Nevertheless, in such form the information on ϕ is hidden. We say then, that the constraint is *informationally incomplete*. Such form will not be compatible (formally and semantically) with our modeling of multi-mode scheduling. So, it is not used in the sequel.

It is just as easy to define resource renewable after 2 periods. Assume total amount be 50. Since resource renew after 2 period, resource balance decrease of ϕ_k and possibly increase of ϕ_{k-2} at each period k . Thus, variables ϕ obey to constraints:

$$\begin{cases} \phi_1 \leq 50, \\ \phi_1 + \phi_2 \leq 50, \\ \phi_2 + \phi_3 \leq 50, \\ \phi_3 + \phi_4 \leq 50, \end{cases}$$

yielding again in a matrix formalism:

$$A\phi \leq \psi \text{ with } A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \text{ and } \psi = \begin{pmatrix} 50 \\ 50 \\ 50 \\ 50 \end{pmatrix}$$

Combining resources: The two preceding examples will be now indexed so as to distinguish them. The non renewable resource is indexed R^∞ (i.e. ϕ^{R^∞} , A^{R^∞} , ...) and the other R^2 (i.e. ϕ^{R^2} , A^{R^2} , ...). Suppose now, that these two kinds of resources are used simultaneously. The probability $\mathbf{P}_{nd}(\varphi)$ becomes:

$$\mathbf{P}_{nd} = \int_{\vec{x} \in E^T} \alpha(\vec{x}) \prod_{k=1}^{k=T} p_{k,x_k}^{R^\infty}(\varphi_k^{R^\infty}(x_k)) p_{k,x_k}^{R^2}(\varphi_k^{R^2}(x_k)) dx_k .$$

Moreover, since resources are parallelized, constraints on ϕ^{R^∞} and ϕ^{R^2} are simply keep unchanged and put together. This operation, generalized in next section as

operator \otimes , yields the following constraint:

$$\begin{pmatrix} A^{R\infty} & 0 \\ 0 & A^{R2} \end{pmatrix} \begin{pmatrix} \phi^{R\infty} \\ \phi^{R2} \end{pmatrix} \leq \begin{pmatrix} \psi^{R\infty} \\ \psi^{R2} \end{pmatrix}$$

Now, assume that resources work in two modes, i.e. exactly like resource $R\infty$ in mode 1 and exactly like resource $R2$ in mode 2. Probability \mathbf{P}_{nd} remains unchanged although in this new context. Assume also that running mode can be changed only when resources are unused or renewed, and that 100 resources $R\infty$ in mode 1 is equivalent to 50 resources $R2$ in mode 2. Then, the balance of resources comes from the concatenation of the weighted balance of the two pure resources. Thus, variables $\phi^{R\infty}$ and ϕ^{R2} obey to new constraints:

$$\begin{cases} \phi_1^{R\infty} + 2\phi_1^{R2} \leq 100, \\ \sum_{k=1}^{\kappa} \phi_k^{R\infty} + 2\phi_{\kappa-1}^{R2} + 2\phi_{\kappa}^{R2} \leq 100 \text{ for } \kappa = 2, 3, 4. \end{cases}$$

This operation, generalized in next section as operator \oplus , yields the following constraint:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 2 & 2 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 2 & 2 \end{pmatrix} \begin{pmatrix} \phi^{R\infty} \\ \phi^{R2} \end{pmatrix} \leq \begin{pmatrix} 100 \\ 100 \\ 100 \\ 100 \end{pmatrix}$$

All these examples rely on similar formalism, which will be unified into the lateral splitting, explained in next section. The structure employed for explaining this formalism may appear a bit abstract. In fact, it simply relies on a definition of non ordered matrix, associated with a labeling of each variable (columns) and of each constraints (rows) by a temporal (or non temporal) index. The labeling is useful for semantic reason. For example it gives us a criterion for concatenating two resources in multimode: constraints on first resource for period k is concatenated with constraints on second resource for same period. To illustrate this formalism, suppose a resource running with the two following modes:

- a complex mode, parallelizing $R\infty$ and $R2$,
- a pure mode, where resources renew after 3 periods, $R3$.

Such resource will run in first mode like two parallelized resources, some non renewable, $R\infty$, and some renewable after 2 periods, $R2$. In second mode, it will run like a resource renewable after 3 periods, $R3$. For sake of clarity, we will put no weight on modes. And for this example, the total amount of resources is fixed to 75. In other

word, we consider a problem of the form $(R_\infty \otimes R_2) \oplus R_3$ (refer to next section). Then, 1 resource running in mode R_3 can, after renew, be changed into 1+1 resources in first mode, say 1 running in mode R_2 and 1 in mode R_∞ . Reversely, if 1 resource in mode R_2 and 1 resource in mode R_∞ are renewed (second case never happen), they could be mixed so as to run in second mode like 1 resource in mode R_3 . Notice that the two resources of first mode have then to be released. Here is given the two matrices for each mode. The corresponding periods, or temporal indices, are given for each constraint, or row, of the matrices:

$$\begin{pmatrix} A^{R_\infty} & 0 \\ 0 & A^{R_2} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 \end{pmatrix} \begin{matrix} \leftarrow 1 \\ \leftarrow 2 \\ \leftarrow 3 \\ \leftarrow 4 \\ \leftarrow 1 \\ \leftarrow 2 \\ \leftarrow 3 \\ \leftarrow 4 \end{matrix} \text{ and } A^{R_3} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix} \begin{matrix} \leftarrow 1 \\ \leftarrow 2 \\ \leftarrow 3 \\ \leftarrow 4 \end{matrix}$$

These matrices operate over variables $(\phi^{R_\infty t}, \phi^{R_2 t})^t$ and ϕ^{R_3} respectively. The concatenated constraints are then represented by the following matrix, where each constraints of same index are combined. This matrix operates over the whole resource vector $(\phi^{R_\infty t}, \phi^{R_2 t}, \phi^{R_3 t})^t$, and takes the following form:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{matrix} \leftarrow 1 \\ \leftarrow 2 \\ \leftarrow 3 \\ \leftarrow 4 \\ \leftarrow 1 \\ \leftarrow 2 \\ \leftarrow 3 \\ \leftarrow 4 \end{matrix} \begin{pmatrix} \phi^{R_\infty} \\ \phi^{R_2} \\ \phi^{R_3} \end{pmatrix} \leq \begin{pmatrix} 75 \\ 75 \\ 75 \\ 75 \\ 75 \\ 75 \\ 75 \\ 75 \end{pmatrix}$$

Non ordered matrix structure is useful to represent our operators, since it is in accordance with their obvious properties (commutativity, distributivity...). Moreover, this structure avoids useless inflation of the constraint representation induced by the operators. For instance, it is easy to verify that $(R_\infty \otimes R_2) \oplus R_3$ gives conditions equivalent to $(R_\infty \oplus R_3) \otimes (R_2 \oplus R_3)$. But written in the classical matrix formalism, there is also inflation on the variables and on the number of columns for the

second writting. The corresponding matrix operates over $(\phi^{R\infty t}, \phi^{R3t}, \phi^{R2t}, \phi^{R3t})^t$. It is obvious that the associated constraint is equivalent to the previous case, despite its different formulation:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix} \begin{matrix} \leftarrow 1 \\ \leftarrow 2 \\ \leftarrow 3 \\ \leftarrow 4 \\ \leftarrow 1 \\ \leftarrow 2 \\ \leftarrow 3 \\ \leftarrow 4 \end{matrix} \begin{pmatrix} \phi^{R\infty} \\ \phi^{R3} \\ \phi^{R2} \\ \phi^{R3} \end{pmatrix} \leq \begin{pmatrix} 75 \\ 75 \\ 75 \\ 75 \\ 75 \\ 75 \\ 75 \\ 75 \end{pmatrix}$$

2.1 Formalism for lateral splitting

Let $\mathcal{T} = \{1, \dots, T\}$ be a set of temporal indices, and $\mathbf{S} \notin \mathcal{T}$ a generic non temporal index. A Lateral Splitting, **LS**, is a quintuplet $\Lambda = (U, \tau, p, \mathcal{A}, m)$, where:

- U is a set of variables indices not including 1. Index 1 will be used for indexing constant data,
- $\tau : U \mapsto \mathcal{T} \cup \{\mathbf{S}\}$ is a mapping, for time labeling of the variables,
- $p = (p_\Upsilon)_{\Upsilon \in \tau^{-1}(\mathcal{T})}$ are non detection functions verifying $p'_{\Upsilon,x} < 0$ and $p''_{\Upsilon,x} > 0$ (diminishing return),
- $\mathcal{A} \subset \mathbb{R}^{U \sqcup \{1\}}$ is a set of weighting for variables and constant³. Constraints are defined by this object. Note that \mathcal{A} is a kind of non ordered matrix,
- $m : \mathcal{A} \rightarrow \mathcal{P}(\mathcal{T}) \setminus \{\emptyset\}$ is a mapping from \mathcal{A} , which defines the temporal context of each constraint.

For example, consider 4 variables ϕ_a, ϕ_b, ϕ_c and ϕ_d , representing resources consumed at period 1 for a, b, c and at period 2 for d . Suppose these variables subjected to constraints associated with temporal context as follows:

$$\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 2 & 2 & 2 \\ 1 & 1 & 0 & 0 \end{pmatrix} \begin{matrix} \leftarrow 1 \\ \leftarrow 1 \\ \leftarrow 2 \end{matrix} \begin{pmatrix} \phi_a \\ \phi_b \\ \phi_c \\ \phi_d \end{pmatrix} \leq \begin{pmatrix} 50 \\ 10 \\ 50 \end{pmatrix}, \quad (8)$$

³Symbol \sqcup represents a Disjoint Union.

where the (period) label on each constraint is given by the left arrows.
 The LS associated to this example is then given by:

$$\left\{ \begin{array}{l} \mathcal{T} = \{1, 2\}, \\ U = \{a, b, c, d\}, \\ \tau(a) = \tau(b) = \tau(c) = 1 \text{ and } \tau(d) = 2, \\ \mathcal{A} = \{\alpha, \beta\}, \text{ where } \begin{cases} \alpha(a) = \alpha(b) = 1, \alpha(c) = \alpha(d) = 0, \alpha(1) = -50, \\ \beta(a) = 0, \beta(b) = \beta(c) = \beta(d) = 2, \beta(1) = -10, \end{cases} \\ m(\alpha) = \{1, 2\} \text{ and } m(\beta) = \{1\}. \end{array} \right. \quad (9)$$

To complete the definition of LS, are defined $\phi = (\phi_{\Upsilon})_{\Upsilon \in U}$, a family of global effort variables and $\varphi = (\varphi_{\Upsilon})_{\Upsilon \in \tau^{-1}(\mathcal{T})}$, a family of spatial sharing functions. Concretely, property $\tau(\Upsilon) = k \in \mathcal{T}$ means that resources ϕ_{Υ} are used for search at period k . Otherwise, when $\tau(\Upsilon) = \mathbf{S}$, resources ϕ_{Υ} are not used. Constraints (10) condition the problem of spatial resource distribution:

$$\left\{ \begin{array}{l} \forall \Upsilon \in U, \phi_{\Upsilon} \geq 0 \text{ and } \forall \Upsilon \in \tau^{-1}(\mathcal{T}), \varphi_{\Upsilon} \geq 0, \\ \forall \Upsilon \in \tau^{-1}(\mathcal{T}), \int_E \varphi_{\Upsilon}(x) dx = \phi_{\Upsilon}, \end{array} \right. \quad (10)$$

Now, in addition to these basic constraints (10), other equations on global variables ϕ_{Υ} are evolved from \mathcal{A} :

$$\forall a \in \mathcal{A}, \sum_{\Upsilon \in U} a(\Upsilon) \phi_{\Upsilon} + a(1) \leq 0. \quad (11)$$

The preceding equation has of course a matrix form:

$$A \phi \leq \psi, \quad (12)$$

where *constraint vector* $\psi = (-a(1))_{a \in \mathcal{A}}$ and *constraint matrix* $A = (a(\Upsilon))_{\Upsilon \in U, a \in \mathcal{A}}$ are constants of the problem. In such model, it appears that the sign of $a(\Upsilon)$ (or $a(1)$) determines whether resources are consumed, positive case, or generated, negative case. The problem constraints being defined, the objective functional is the non-detection probability $\mathbf{P}_{nd}(\varphi)$ given below:

$$\mathbf{P}_{nd}(\varphi) = \int_{E^T} \alpha(\vec{x}) \prod_{k=1}^{k=T} \left(\prod_{\Upsilon \in \tau^{-1}(k)} p_{\Upsilon, x_k}(\varphi_{\Upsilon}(x_k)) \right) dx_k \quad (13)$$

LS-problem with inequality constraints, **ICLS**, consists in **optimizing both ϕ and φ so as to minimize $\mathbf{P}_{nd}(\varphi)$ under constraints (10) and (11-12).**

Equality constraint: LS-problem with equality constraints, **ECLS**, is just obtained by replacing (11) or (12) by either $\forall a \in \mathcal{A}, \sum_{\Upsilon \in U} a(\Upsilon) \phi_{\Upsilon} + a(1) = 0$ or $A\phi = \psi$. ECLS is too restrictive and is inadequate with the underlying semantic of our model. All coming formalism will thus rather refer to ICLS. Nevertheless, our algorithm is built for ECLS, and ICLS is defined as a corollary of ECLS by simply adding slack variables (mapping to \mathbf{S}). More precisely, we will add as much new slack variables as cardinality of \mathcal{A} . One variable only is activated (weighted with a non zero value) for each $a \in \mathcal{A}$. Since there is a positivity constraints on variables, it suffices to put a positive weight, say 1, on needed variable, to handle lower inequality. ICLS $(U, \tau, p, \mathcal{A}, m)$ is simply changed into an equivalent ECLS $(U_{\leq}, \tau_{\leq}, p, \mathcal{A}_{\leq}, m_{\leq})$, in the following way:

$$\begin{cases} U_{\leq} = U \sqcup \mathcal{A} \text{ (disjoint union: } U \cap \mathcal{A} = \emptyset \text{)}, \\ \forall \Upsilon \in U, \tau_{\leq}(\Upsilon) = \tau(\Upsilon) \text{ and } \tau_{\leq}(\mathcal{A}) = \mathbf{S}. \end{cases} \quad (14)$$

Define a_{\leq} for each $a \in \mathcal{A}$, by completion inside \mathcal{A} :

$$\begin{cases} \forall \Upsilon \in U \cup \{1\}, a_{\leq}(\Upsilon) = a(\Upsilon), \\ a_{\leq}(a) = 1 \text{ and } \forall \Upsilon \in \mathcal{A} \setminus \{a\}, a_{\leq}(\Upsilon) = 0. \end{cases}$$

Objects \mathcal{A}_{\leq} and m_{\leq} are then defined by:

$$\begin{cases} \mathcal{A}_{\leq} = \{a_{\leq}/a \in \mathcal{A}\}, \\ \forall a_{\leq} \in \mathcal{A}_{\leq}, m_{\leq}(a_{\leq}) = m(a). \end{cases} \quad (15)$$

Taking back our example (9), its equivalent ECLS form is given by:

$$\begin{cases} U_{\leq} = \{a, b, c, d, \alpha, \beta\}, \\ \tau_{\leq}(a) = \tau_{\leq}(b) = \tau_{\leq}(c) = 1, \tau_{\leq}(d) = 2 \text{ and } \tau_{\leq}(\alpha) = \tau_{\leq}(\beta) = \mathbf{S}, \\ \mathcal{A}_{\leq} = \{\alpha_{\leq}, \beta_{\leq}\}, \text{ where } \begin{cases} \alpha_{\leq}(a) = \alpha_{\leq}(b) = 1, \alpha_{\leq}(c) = \alpha_{\leq}(d) = 0, \\ \alpha_{\leq}(\alpha) = 1, \alpha_{\leq}(\beta) = 0, \alpha_{\leq}(1) = -50, \\ \beta_{\leq}(a) = 0, \beta_{\leq}(b) = \beta_{\leq}(c) = \beta_{\leq}(d) = 2, \\ \beta_{\leq}(\alpha) = 0, \beta_{\leq}(\beta) = 1, \beta_{\leq}(1) = -10, \end{cases} \\ m_{\leq}(\alpha_{\leq}) = \{1, 2\} \text{ and } m_{\leq}(\beta_{\leq}) = \{1\}. \end{cases}$$

The constraints associated to this LS is given by:

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 2 & 2 & 2 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 \end{pmatrix} \leftarrow \begin{matrix} 1 \\ 1 \\ 2 \end{matrix} \begin{pmatrix} \phi_a \\ \phi_b \\ \phi_c \\ \phi_d \\ \phi_\alpha \\ \phi_\beta \end{pmatrix} = \begin{pmatrix} 50 \\ 10 \\ 50 \end{pmatrix},$$

which is equivalent to:

$$\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 2 & 2 & 2 \\ 1 & 1 & 0 & 0 \end{pmatrix} \leftarrow \begin{matrix} 1 \\ 1 \\ 2 \end{matrix} \begin{pmatrix} \phi_a \\ \phi_b \\ \phi_c \\ \phi_d \end{pmatrix} \leq \begin{pmatrix} 50 \\ 10 \\ 50 \end{pmatrix},$$

Basic LS: Let $\Lambda = (U, \tau, p, \mathcal{A}, m)$ be an LS. In LS formalism, each index $\Upsilon \in U$ represents a resource *variable* since index 1 represents the resource *constant*. These resources are weighted for each constraint $a \in \mathcal{A}$. Weighting signifies that resources may be less or more costly. A resource weighted 2 is equivalent to two resources weighted 1. A negative weight signifies a reversed cost. It may be used to handle resource generation, and also conditionality on resources. For example, to explain that a total of λ resources is made available by the system, we should use a resource constant weighted by $-\lambda$. Here comes a distinction between resources availability, which depends on the system and is associated to constant index, and resource properties characterized by constraints on variable indices. For this reason, some operations on resources explained in the sequel, will be applied rather on LSs with zero weighted constants. Such LS, verifying also $\forall a \in \mathcal{A}, a(1) = 0$, are called *linear* LS. Otherwise, pure constant LS may be defined and used together with linear LS. For $\lambda \in \mathbb{R}$, a LS also named λ is defined by the quintuplet:

$$\lambda \triangleq (\emptyset, \emptyset, \emptyset, \{a\}, m), \text{ with } a(1) = \lambda, m(a) = \mathcal{T}. \quad (16)$$

Coming back to the matrix formalism, constant LS λ is described for $T = 4$ by the *empty matrix* $A_\lambda = (\emptyset, \emptyset, \emptyset, \emptyset)^t$ and the vector $\psi_\lambda = (-\lambda, -\lambda, -\lambda, -\lambda)^t$.

LS for renewable resources: Assume be given a resource type capable of renew after $\Delta T \in \mathbb{N}^* \cup \{\infty\}$ periods (time for replenishment, for moving, ...). Let

us consider $\Lambda^{R\Delta T} = (U^{R\Delta T}, \tau^{R\Delta T}, p^{R\Delta T}, \mathcal{A}^{R\Delta T}, m^{R\Delta T})$, the linear LS associated. Search is assumed to be done over T periods, so that $\mathcal{T} = \{1, \dots, T\}$. $p^{R\Delta T}$ is supposed known. Since the resource is simple (one mode and one type), $\tau^{R\Delta T}$ is one to one and there is one variable for each period. Thus, $U^{R\Delta T} = \{\Upsilon_k/k \in \mathcal{T}\}$ with $\tau^{R\Delta T}(\Upsilon_k) = k$. For same reason, there is a one to one correspondence between constraints and periods. Thus, $\mathcal{A}^{R\Delta T} = \{a_k/k \in \mathcal{T}\}$ and $m^{R\Delta T}(a_k) = \{k\}$. It remains to define the constraint a_k associated with the period k . It results from the resource balance at period k and is recursively deduced. For period 1, only the cost associated to ϕ_1 is involved. For period k , the cost of ϕ_k and the possible negative cost (since renew) of $\phi_{k-\Delta T}$ are added to the statement of period $k-1$:

$$\begin{cases} a_1(\Upsilon_1) = 1 \text{ and } a_1(\Upsilon) = 0 \text{ else,} \\ \forall k > 1, \begin{cases} a_k(\Upsilon_k) = a_{k-1}(\Upsilon_k) + 1, \\ \forall k > \Delta T, a_k(\Upsilon_{k-\Delta T}) = a_{k-1}(\Upsilon_{k-\Delta T}) - 1, \\ a_k(\Upsilon) = a_{k-1}(\Upsilon) \text{ else.} \end{cases} \end{cases} \quad (17)$$

Consider the example \mathcal{A}^{R3} for $T = 4$. Using the recursive definition, the balance of period 1 is given by the currently used resources, Υ_1 :

$$a_1(\Upsilon_1) = 1, a_1(\Upsilon_2) = 0, a_1(\Upsilon_3) = 0, a_1(\Upsilon_4) = 0, a_1(1) = 0. \quad (18)$$

This balance is duplicated at period 2, with addition of currently used resources, Υ_2 . Likewise, an updated balance is obtained from period 2 to period 3:

$$\begin{cases} a_2(\Upsilon_1) = 1, a_2(\Upsilon_2) = 1, a_2(\Upsilon_3) = 0, a_2(\Upsilon_4) = 0, a_2(1) = 0, \\ a_3(\Upsilon_1) = 1, a_3(\Upsilon_2) = 1, a_3(\Upsilon_3) = 1, a_3(\Upsilon_4) = 0, a_3(1) = 0. \end{cases} \quad (18b)$$

At period 4, the balance involves not only the addition of used resources, Υ_4 , but also the release of resources used at period 1, Υ_1 :

$$a_4(\Upsilon_1) = 0, a_4(\Upsilon_2) = 1, a_4(\Upsilon_3) = 1, a_4(\Upsilon_4) = 1, a_4(1) = 0. \quad (18c)$$

This LS has the following matrix form:

$$A^{R3} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix} \text{ and } \psi^{R3} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Weighted renewable resources: Resource type is considered, which renews gradually according with a parameter sequence ρ . More precisely, for one resource used at period k , $\rho_{\Delta k}$ resource is then regenerated at period $k + \Delta k$. Property $\sum_k \rho_k \leq 1$ also holds. Denote $\Lambda^R(\rho) = (U_\rho^R, \tau_\rho^R, p_\rho^R, \mathcal{A}_\rho^R, m_\rho^R)$ the linear LS associated. As previously, there is a one to one correspondence between variables, periods and constraints. We can write $U_\rho^R = \{\Upsilon_k/k \in \mathcal{T}\}$, $\mathcal{A}_\rho^R = \{a_k/k \in \mathcal{T}\}$, with $\tau_\rho^R(\Upsilon_k) = k$ and $m_\rho^R(a_k) = \{k\}$. Then, a_k is defined by:

$$\begin{cases} a_1(\Upsilon_1) = 1 \text{ and } a_1(\Upsilon) = 0 \text{ else,} \\ \forall k > 1, \begin{cases} a_k(\Upsilon_k) = a_{k-1}(\Upsilon_k) + 1, \\ \forall 1 \leq l < k, a_k(\Upsilon_l) = a_{k-1}(\Upsilon_l) - \rho_{k-l}, \\ a_k(\Upsilon) = a_{k-1}(\Upsilon) \text{ else.} \end{cases} \end{cases} \quad (19)$$

Consider the example $\mathcal{A}^R(\frac{1}{3}, \frac{1}{2})$ for $T = 4$. Using the recursive definition, the balance of period 1 is given by the currently used resources, Υ_1 :

$$a_1(\Upsilon_1) = 1, a_1(\Upsilon_2) = 0, a_1(\Upsilon_3) = 0, a_1(\Upsilon_4) = 0, a_1(1) = 0. \quad (20)$$

This balance is duplicated at period 2, with addition of currently used resources, Υ_2 , and subtraction of released resources. At this period, $\frac{1}{3}$ of the resources consumed at period 1, Υ_1 , are released:

$$a_2(\Upsilon_1) = \frac{2}{3}, a_2(\Upsilon_2) = 1, a_2(\Upsilon_3) = 0, a_2(\Upsilon_4) = 0, a_2(1) = 0. \quad (20b)$$

At period 3, the balance involves likewise the addition of used resources, Υ_3 , and subtraction of released resources. Naturally, $\frac{1}{3}$ of the resources consumed at period 2, Υ_2 , are released. But in addition, $\frac{1}{2}$ of the resources consumed at period 1, Υ_1 , are released again:

$$a_3(\Upsilon_1) = \frac{1}{6}, a_3(\Upsilon_2) = \frac{2}{3}, a_3(\Upsilon_3) = 1, a_3(\Upsilon_4) = 0, a_3(1) = 0. \quad (20c)$$

Period 4 runs like period 3, with release of resources used at period 2 and 3, but no more resources consumed at period 1 are released:

$$a_4(\Upsilon_1) = \frac{1}{6}, a_4(\Upsilon_2) = \frac{1}{6}, a_4(\Upsilon_3) = \frac{2}{3}, a_4(\Upsilon_4) = 1, a_4(1) = 0. \quad (20)$$

This LS has the following matrix form:

$$A^R\left(\frac{1}{3}, \frac{1}{2}\right) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{2}{3} & 1 & 0 & 0 \\ \frac{1}{6} & \frac{2}{3} & 1 & 0 \\ \frac{1}{6} & \frac{1}{6} & \frac{2}{3} & 1 \end{pmatrix} \text{ and } \psi^R\left(\frac{1}{3}, \frac{1}{2}\right) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Obviously, there is some semantics, behind our model. There is a temporal semantic, which underlies the recursive construction of the LS seen above. It is needless to develop anymore argumentation about it. But it is important to respect the temporal causality when constructing LS. This prevents from irrelevant use of the model, such as *informational incompleteness*. There is also a global semantic, which establish relation between already constructed LS. This aspect will be more detailed, with the construction of operators on LS.

Global operators: Let $\lambda, \mu \in \mathbb{R}$. Notation $\Lambda^\alpha = (U^\alpha, \tau^\alpha, p^\alpha, \mathcal{A}^\alpha, m^\alpha)$ is used for $\alpha = a, b, c, d$. Indices between two LSs Λ^a and Λ^b are assumed compatible, that is $\tau^a(\Upsilon) = \tau^b(\Upsilon)$ and $p^a(\Upsilon) = p^b(\Upsilon)$ whenever $\Upsilon \in U^a \cap U^b$.

Scalar product operator: The scalar product $\lambda\Lambda^a$ is given by the quintuplet $(U^a, \tau^a, p^a, \lambda\mathcal{A}^a, \lambda m^a)$, where $\lambda\mathcal{A}^a = \{\lambda a/a \in \mathcal{A}^a\}$ and λm^a is the minimal object defined by:

$$\forall a \in \mathcal{A}, m^a(a) \subset \lambda m^a(\lambda a). \quad (21)$$

So, the weight of each variable and of the constant is multiplied by λ .

Operator \otimes : We set $\Lambda^a \otimes \Lambda^b = (U^a \cup U^b, \tau^a \cup \tau^b, p^a \cup p^b, \mathcal{A}^a \otimes \mathcal{A}^b, m^a \otimes m^b)$. Function $\tau^a \cup \tau^b$ is defined by $\forall \Upsilon \in U^\alpha, \tau^\alpha \cup \tau^b(\Upsilon) = \tau^\alpha(\Upsilon)$ for $\alpha = a, b$. Similar definition holds for $p^a \cup p^b$. Now, for $a^a \in \mathcal{A}^a$ and $a^b \in \mathcal{A}^b$, completions a_\otimes^a and ${}_\otimes a^b$ are weighting for $U^a \cup U^b \cup \{1\}$ defined such way:

$$\begin{cases} \forall \Upsilon \in U^a \cup \{1\}, a_\otimes^a(\Upsilon) = a^a(\Upsilon) \text{ and } \forall \Upsilon \in U^b \setminus U^a, a_\otimes^a(\Upsilon) = 0, \\ \forall \Upsilon \in U^a \setminus U^b, {}_\otimes a^b(\Upsilon) = 0 \text{ and } \forall \Upsilon \in U^b \cup \{1\}, {}_\otimes a^b(\Upsilon) = a^b(\Upsilon). \end{cases}$$

Set $\mathcal{A}^a \otimes \mathcal{A}^b$ and mapping $m^a \otimes m^b$ are minimal objects defined by:

$$\begin{cases} \forall a^a \in \mathcal{A}^a, [a_\otimes^a \in \mathcal{A}^a \otimes \mathcal{A}^b \text{ and } m^a(a^a) \subset m^a \otimes m^b(a_\otimes^a)] , \\ \forall a^b \in \mathcal{A}^b, [{}_\otimes a^b \in \mathcal{A}^a \otimes \mathcal{A}^b \text{ and } m^b(a^b) \subset m^a \otimes m^b({}_\otimes a^b)] . \end{cases} \quad (22)$$

Assume as example that Λ^a is given by (18) and Λ^b is given by (20). In fact, we suppose that a and b are independent, that is $U^a \cap U^b = \emptyset$. The variable indices will be also distinguished by exponent a or b . Then,

$\Lambda^a \otimes \Lambda^b$ is given by the quintuplet $(U, \tau, p, \mathcal{A}, m)$, where U , τ , \mathcal{A} and m are defined as follows:

$$\left\{ \begin{array}{l} U = \{\Upsilon_1^a, \Upsilon_2^a, \Upsilon_3^a, \Upsilon_4^a, \Upsilon_1^b, \Upsilon_2^b, \Upsilon_3^b, \Upsilon_4^b\}, \tau(\Upsilon_k^a) = \tau(\Upsilon_k^b) = k, \\ \mathcal{A} = \{a_1^a, a_2^a, a_3^a, a_4^a, a_1^b, a_2^b, a_3^b, a_4^b\}, m(a_k^a) = m(a_k^b) = \{k\}, \\ a_1^a(\Upsilon_1^a) = 1, a_1^a(\Upsilon_2^a) = 0, a_1^a(\Upsilon_3^a) = 0, a_1^a(\Upsilon_4^a) = 0, a_1^a(\Upsilon_k^b) = 0, a_1^a(1) = 0, \\ a_2^a(\Upsilon_1^a) = 1, a_2^a(\Upsilon_2^a) = 1, a_2^a(\Upsilon_3^a) = 0, a_2^a(\Upsilon_4^a) = 0, a_2^a(\Upsilon_k^b) = 0, a_2^a(1) = 0, \\ a_3^a(\Upsilon_1^a) = 1, a_3^a(\Upsilon_2^a) = 1, a_3^a(\Upsilon_3^a) = 1, a_3^a(\Upsilon_4^a) = 0, a_3^a(\Upsilon_k^b) = 0, a_3^a(1) = 0, \\ a_4^a(\Upsilon_1^a) = 0, a_4^a(\Upsilon_2^a) = 1, a_4^a(\Upsilon_3^a) = 1, a_4^a(\Upsilon_4^a) = 1, a_4^a(\Upsilon_k^b) = 0, a_4^a(1) = 0, \\ a_1^b(\Upsilon_k^a) = 0, a_1^b(\Upsilon_1^b) = 1, a_1^b(\Upsilon_2^b) = 0, a_1^b(\Upsilon_3^b) = 0, a_1^b(\Upsilon_4^b) = 0, a_1^b(1) = 0, \\ a_2^b(\Upsilon_k^a) = 0, a_2^b(\Upsilon_1^b) = \frac{2}{3}, a_2^b(\Upsilon_2^b) = 1, a_2^b(\Upsilon_3^b) = 0, a_2^b(\Upsilon_4^b) = 0, a_2^b(1) = 0, \\ a_3^b(\Upsilon_k^a) = 0, a_3^b(\Upsilon_1^b) = \frac{1}{6}, a_3^b(\Upsilon_2^b) = \frac{2}{3}, a_3^b(\Upsilon_3^b) = 1, a_3^b(\Upsilon_4^b) = 0, a_3^b(1) = 0, \\ a_4^b(\Upsilon_k^a) = 0, a_4^b(\Upsilon_1^b) = \frac{1}{6}, a_4^b(\Upsilon_2^b) = \frac{1}{6}, a_4^b(\Upsilon_3^b) = \frac{2}{3}, a_4^b(\Upsilon_4^b) = 1, a_4^b(1) = 0, \end{array} \right.$$

and this LS has the following matrix form:

$$A = \begin{pmatrix} 1000 & 0000 \\ 1100 & 0000 \\ 1110 & 0000 \\ 0111 & 0000 \\ 0000 & 1000 \\ 0000 & \frac{2}{3}100 \\ 0000 & \frac{1}{6}\frac{2}{3}10 \\ 0000 & \frac{1}{6}\frac{1}{6}\frac{2}{3}1 \end{pmatrix} \text{ and } \psi = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Operator \oplus : We set $\Lambda^a \oplus \Lambda^b = (U^a \cup U^b, \tau^a \cup \tau^b, p^a \cup p^b, \mathcal{A}^a \oplus \mathcal{A}^b, m^a \oplus m^b)$. For all $(a^a, a^b) \in \mathcal{A}^a \times \mathcal{A}^b$, let $a^a \oplus a^b$ be stated by:

$$\left\{ \begin{array}{l} \forall \Upsilon \in U^a \cup U^b, a^a \oplus a^b(\Upsilon) = \sum_{U^\alpha \ni \Upsilon} a^\alpha(\Upsilon), \\ a^a \oplus a^b(1) = a^a(1) + a^b(1). \end{array} \right.$$

Set $\mathcal{A}^a \oplus \mathcal{A}^b$ and mapping $m^a \oplus m^b$ are minimal objects defined by:

$$\forall (a^a, a^b) \in \mathcal{A}^a \times \mathcal{A}^b, \left[k \in m(a^a) \cap m(a^b) \Rightarrow \left\{ \begin{array}{l} a^a \oplus a^b \in \mathcal{A}^a \oplus \mathcal{A}^b \\ k \in m^a \oplus m^b(a^a \oplus a^b) \end{array} \right. \right]. \quad (23)$$

Assume again that Λ^a is given by (18) and Λ^b is given by (20). We still suppose that a and b are independent. The variable indices will be

also distinguished by exponent a or b . Then, $\Lambda^a \oplus \Lambda^b$ is given by the quintuplet $(U, \tau, p, \mathcal{A}, m)$, where U , τ , \mathcal{A} and m are defined as follows:

$$\left\{ \begin{array}{l} U = \{\Upsilon_1^a, \Upsilon_2^a, \Upsilon_3^a, \Upsilon_4^a, \Upsilon_1^b, \Upsilon_2^b, \Upsilon_3^b, \Upsilon_4^b\}, \tau(\Upsilon_k^a) = \tau(\Upsilon_k^b) = k, \\ \mathcal{A} = \{a_1, a_2, a_3, a_4\}, m(a_k) = \{k\}, \\ a_1(\Upsilon_1^a) = 1, a_1(\Upsilon_2^a) = 0, a_1(\Upsilon_3^a) = 0, a_1(\Upsilon_4^a) = 0, \\ a_1(\Upsilon_1^b) = 1, a_1(\Upsilon_2^b) = 0, a_1(\Upsilon_3^b) = 0, a_1(\Upsilon_4^b) = 0, a_1(1) = 0, \\ a_2(\Upsilon_1^a) = 1, a_2(\Upsilon_2^a) = 1, a_2(\Upsilon_3^a) = 0, a_2(\Upsilon_4^a) = 0, \\ a_2(\Upsilon_1^b) = \frac{2}{3}, a_2(\Upsilon_2^b) = 1, a_2(\Upsilon_3^b) = 0, a_2(\Upsilon_4^b) = 0, a_2(1) = 0, \\ a_3(\Upsilon_1^a) = 1, a_3(\Upsilon_2^a) = 1, a_3(\Upsilon_3^a) = 1, a_3(\Upsilon_4^a) = 0, \\ a_3(\Upsilon_1^b) = \frac{1}{6}, a_3(\Upsilon_2^b) = \frac{2}{3}, a_3(\Upsilon_3^b) = 1, a_3(\Upsilon_4^b) = 0, a_3(1) = 0, \\ a_4(\Upsilon_1^a) = 0, a_4(\Upsilon_2^a) = 1, a_4(\Upsilon_3^a) = 1, a_4(\Upsilon_4^a) = 1, \\ a_4(\Upsilon_1^b) = \frac{1}{6}, a_4(\Upsilon_2^b) = \frac{1}{6}, a_4(\Upsilon_3^b) = \frac{2}{3}, a_4(\Upsilon_4^b) = 1, a_4(1) = 0, \end{array} \right.$$

and this LS has the following matrix form:

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & \frac{2}{3} & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & \frac{1}{6} & \frac{2}{3} & 1 & 0 \\ 0 & 1 & 1 & 1 & \frac{1}{6} & \frac{1}{6} & \frac{2}{3} & 1 \end{pmatrix} \text{ and } \psi = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Operator \succ : We set $\Lambda^a \succ \Lambda^b = (U^a \cup U^b, \tau^a \cup \tau^b, p^a \cup p^b, \mathcal{A}^a \succ \mathcal{A}^b, m^a \succ m^b)$. $\mathcal{A}^a \succ \mathcal{A}^b$ will glue variable blocks which have the same temporal label as the line containing them. Define δ by $\delta_{kl}^\alpha = 0$ if $l \neq k$, $\delta_{kk}^a = -1$ and $\delta_{kk}^b = 1$. Function $a^a \succ_k a^b$ is stated as follows:

$$\left\{ \begin{array}{l} \forall \Upsilon \in U^a \cup U^b, a^a \succ_k a^b(\Upsilon) = \sum_{U^\alpha \ni \Upsilon} \delta_{k, \tau^\alpha(\Upsilon)}^\alpha a^\alpha(\Upsilon), \\ a^a \succ_k a^b(1) = 0. \end{array} \right.$$

Set $\mathcal{A}^a \succ \mathcal{A}^b$ and mapping $m^a \succ m^b$ are minimal objects defined by:

$$\forall (a^a, a^b) \in \mathcal{A}^a \times \mathcal{A}^b, \left[k \in m(a^a) \cap m(a^b) \Rightarrow \left\{ \begin{array}{l} a^a \succ_k a^b \in \mathcal{A}^a \succ \mathcal{A}^b \\ k \in m^a \succ m^b(a^a \succ_k a^b) \end{array} \right\} \right]. \quad (24)$$

Assume again that Λ^a is given by (18) and Λ^b is given by (20), and are independent. Then, $\Lambda^a \succ \Lambda^b$ is given by the quintuplet $(U, \tau, p, \mathcal{A}, m)$,

where U , τ , \mathcal{A} and m are defined as follows:

$$\begin{cases} U = \{\Upsilon_1^a, \Upsilon_2^a, \Upsilon_3^a, \Upsilon_4^a, \Upsilon_1^b, \Upsilon_2^b, \Upsilon_3^b, \Upsilon_4^b\}, \tau(\Upsilon_k^a) = \tau(\Upsilon_k^b) = k, \\ \mathcal{A} = \{a_1, a_2, a_3, a_4\}, m(a_k) = \{k\}, \\ a_1(\Upsilon_1^a) = -1, a_1(\Upsilon_2^a) = 0, a_1(\Upsilon_3^a) = 0, a_1(\Upsilon_4^a) = 0, \\ a_1(\Upsilon_1^b) = 1, a_1(\Upsilon_2^b) = 0, a_1(\Upsilon_3^b) = 0, a_1(\Upsilon_4^b) = 0, a_1(1) = 0, \\ a_2(\Upsilon_1^a) = 0, a_2(\Upsilon_2^a) = -1, a_2(\Upsilon_3^a) = 0, a_2(\Upsilon_4^a) = 0, \\ a_2(\Upsilon_1^b) = 0, a_2(\Upsilon_2^b) = 1, a_2(\Upsilon_3^b) = 0, a_2(\Upsilon_4^b) = 0, a_2(1) = 0, \\ a_3(\Upsilon_1^a) = 0, a_3(\Upsilon_2^a) = 0, a_3(\Upsilon_3^a) = -1, a_3(\Upsilon_4^a) = 0, \\ a_3(\Upsilon_1^b) = 0, a_3(\Upsilon_2^b) = 0, a_3(\Upsilon_3^b) = 1, a_3(\Upsilon_4^b) = 0, a_3(1) = 0, \\ a_4(\Upsilon_1^a) = 0, a_4(\Upsilon_2^a) = 0, a_4(\Upsilon_3^a) = 0, a_4(\Upsilon_4^a) = -1, \\ a_4(\Upsilon_1^b) = 0, a_4(\Upsilon_2^b) = 0, a_4(\Upsilon_3^b) = 0, a_4(\Upsilon_4^b) = 1, a_4(1) = 0, \end{cases}$$

and this LS has the following matrix form:

$$A = \begin{pmatrix} -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 \end{pmatrix} \text{ and } \psi = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Operator \succ establish some conditional dependency between two LSs. Operator \otimes parallelizes two LSs or make a logical *AND* between two LSs. Operator \oplus merges two LSs into a unique, but multi-modal, resource type. Scalar product adjusts the *power* of each resource or mode. Operators \otimes and \oplus are associative and commutative. The scalar product is left-distributive w.r.t. \oplus and \otimes . It is also right-distributive w.r.t. \oplus . Operator \oplus is distributive w.r.t. \otimes . These properties are summarized below with, in addition, some properties on \succ :

Property 1 *The following holds for compatible LSs $\Lambda^a, \Lambda^b, \Lambda^c, \Lambda^d$ and scalar λ, μ :*

$$\left| \begin{array}{l} \Lambda^a \oplus \Lambda^b = \Lambda^b \oplus \Lambda^a \\ \Lambda^a \otimes \Lambda^b = \Lambda^b \otimes \Lambda^a \\ \lambda(\Lambda^a \oplus \Lambda^b) = (\lambda\Lambda^a) \oplus (\lambda\Lambda^b) \\ \lambda(\Lambda^a \otimes \Lambda^b) = (\lambda\Lambda^a) \otimes (\lambda\Lambda^b) \\ (\lambda + \mu)\Lambda^a = (\lambda\Lambda^a) \oplus (\mu\Lambda^a) \end{array} \right| \left| \begin{array}{l} (\Lambda^a \otimes \Lambda^b) \oplus \Lambda^c = (\Lambda^a \oplus \Lambda^c) \otimes (\Lambda^b \oplus \Lambda^c) \\ (\Lambda^a \succ \Lambda^c) \otimes (\Lambda^b \succ \Lambda^c) = (\Lambda^a \otimes \Lambda^b) \succ \Lambda^c \\ (\Lambda^a \succ \Lambda^b) \otimes (\Lambda^a \succ \Lambda^c) = \Lambda^a \succ (\Lambda^b \otimes \Lambda^c) \\ (\Lambda^a \succ \Lambda^b) \oplus (\Lambda^c \succ \Lambda^d) = (\Lambda^a \oplus \Lambda^c) \succ (\Lambda^b \oplus \Lambda^d) \\ (\Lambda^a \succ \Lambda^b) \succ \Lambda^c = (\Lambda^a \oplus \Lambda^b) \succ \Lambda^c \end{array} \right| \quad (25)$$

2.2 Some typical applications

Multi-mode scheduling: Resources are used, which operate in one of the modes a and b , with proper LS Λ^a and Λ^b . The objective is to optimize the choice between the

operating modes so as to minimize P_{nd} . This is done by $\Lambda^a \oplus \Lambda^b$, which concatenates the constraints of Λ^a and of Λ^b . Refer to 2.3 for example.

Multi-resource scheduling: Resources a and b , with proper LS Λ^a and Λ^b , are given. They are independent, that is $U^a \cap U^b = \emptyset$. We want a and b to be parallelized, i.e. resources a can run independently and in same period as resources b . The objective is then to allocate a and b together so as to minimize P_{nd} . It is simply done by $\Lambda^a \otimes \Lambda^b$, since constraint $\mathcal{A}^a \otimes \mathcal{A}^b$ combines simultaneously constraints \mathcal{A}^a and \mathcal{A}^b .

Resource dependency: Typically, suppose that resource a can work only when resource b is used. Then, the conditional LS $\Lambda^b \succ \Lambda^a$ has to be glued by \otimes -means to the main LS. Refer to 2.3 for example.

$(\Lambda \otimes \Lambda^a) \oplus (\Lambda \otimes \Lambda^b)$ and $\Lambda \otimes (\Lambda^a \oplus \Lambda^b)$: These two examples are really different. In first case, resources could be used in mode 1, with parallelized resources Λ and Λ^a , and in mode 2, with parallelized resources Λ and Λ^b . Nevertheless, resources Λ released in mode 1 cannot be used directly in mode 2 without releasing resources Λ^a . Parallelized resources have to be *combined* before jumping from one mode to another. In the second case, Λ is parallelized with resources working in mode Λ^a and Λ^b . Λ is used independently of Λ^a and Λ^b .

2.3 Practical examples

Three examples are given here in order to clarify, if needed, the abstract formalism of ICLS. For didactic reason, the constraints of an ICLS Λ^α will be represented by a matrix form $A^\alpha \phi^\alpha \leq \psi^\alpha$. First, we consider independent linear LSs $\Lambda^a = \Lambda^{R\infty}$ and $\Lambda^b = \Lambda^{R(\frac{1}{4}, \frac{1}{2})}$ defined on 4 periods. According to the definition, constraints for Λ^a and Λ^b are respectively:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix} \phi^a \leq \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{3}{4} & 1 & 0 & 0 \\ \frac{1}{4} & \frac{3}{4} & 1 & 0 \\ \frac{1}{4} & \frac{1}{4} & \frac{3}{4} & 1 \end{pmatrix} \phi^b \leq \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} .$$

Define a multi-resource scheduling of a and b with respective amount of resources 7 and 5. This problem reverts to $(\Lambda^a \oplus -7) \otimes (\Lambda^b \oplus -5)$, yielding constraint:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{3}{4} & 1 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{4} & \frac{3}{4} & 1 \\ 0 & 0 & 0 & 0 & \frac{1}{4} & \frac{1}{4} & \frac{3}{4} & 1 \end{pmatrix} \begin{pmatrix} \phi^a \\ \phi^b \end{pmatrix} \leq \begin{pmatrix} 7 \\ 7 \\ 7 \\ 7 \\ 5 \\ 5 \\ 5 \\ 5 \end{pmatrix} .$$

Assume now that resources can run in mode a or in mode b , and that b needs 2 times more resources than a . Assume moreover that the total amount of resource is 20 when running in mode a . This problem reverts to ICLS $\Lambda^a \oplus (2\Lambda^b) \oplus -20$, thus yielding constraint:

$$\begin{pmatrix} 1 & 0 & 0 & 2 & 0 & 0 & 0 \\ 1 & 1 & 0 & \frac{3}{2} & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & \frac{1}{2} & \frac{3}{2} & 2 \\ 1 & 1 & 1 & 1 & \frac{1}{2} & \frac{1}{2} & \frac{3}{2} & 2 \end{pmatrix} \begin{pmatrix} \phi^a \\ \phi^b \end{pmatrix} \leq \begin{pmatrix} 20 \\ 20 \\ 20 \\ 20 \end{pmatrix} .$$

As in the previous example, let us consider a conditional problem, where the use of a resource b is conditioned to the use of 2 resources a . Let the total amount of resources be 10 for a and 5 for b . The suitable ICLS is $(\Lambda^a \oplus -10) \otimes (\Lambda^b \oplus -5) \otimes (\Lambda^a \succ (2\Lambda^b))$, yielding:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{3}{4} & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{4} & \frac{3}{4} & 1 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{4} & \frac{1}{4} & \frac{3}{4} & 1 \\ -1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 2 \end{pmatrix} \begin{pmatrix} \phi^a \\ \phi^b \end{pmatrix} \leq \begin{pmatrix} 10 \\ 10 \\ 10 \\ 10 \\ 5 \\ 5 \\ 5 \\ 5 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} .$$

3 Numerical resolution

An algorithm is now given for solving ECLS. The ECLS is approximated by series of Brown's problems, and the optimum of the ECLS is given by the optimum of the nearest approximation. To command the choice of these approximations, a differential method (relative to the intermediate optima) is used.

3.1 Algorithm

Let $\Lambda = (U, \tau, p, \mathcal{A}, m)$ be an ECLS, and let A and ψ be its constraint matrix and vector. For each ϕ verifying:

$$A\phi = \psi, \quad (26)$$

we define a Brown-like problem $B(\Lambda, \phi)$, which consists to minimize $\mathbf{P}_{nd}(\varphi)$ given in (13) under constraint (10). Resolution of $B(\Lambda, \phi)$ is done by an algorithm almost identical to Brown's algorithm. It will not be explained here. ECLS Λ will be approximated by problems $B(\Lambda, \phi)$, where ϕ is pushed by infinitesimal variations $d\phi$ to an optimal value ϕ_Λ . Criteria have to be found so as to define valid and optimal choice of $d\phi$, satisfying (26). Consider also \tilde{A} a matrix such that $\ker A = \text{Im}(\tilde{A})$ and a vector ϕ_0 such that $A\phi_0 = \psi$ (ϕ_0 will be taken as starting value for ϕ). Then, for each vector ϕ fulfilling (26), there is a vector ν such that $\phi = \phi_0 + \tilde{A}\nu$. Likewise (26) holds true for all ϕ of preceding form. Thus, all valid $d\phi$ verifies:

$$d\phi = \tilde{A}d\nu \quad (27)$$

Valid choice of $d\phi$ reduces to a choice of $d\nu$. To find an optimal value ϕ_Λ for ϕ , i.e. which minimizes $\min_\varphi \mathbf{P}_{nd}(\varphi)$, a choice criterion on $d\nu$, based on the calculation of $d \min_\varphi \mathbf{P}_{nd}(\varphi)$, is defined in next section. This calculation depends on the current optimal solution φ associated to ϕ . Hence the following algorithm:

1. Compute \tilde{A} , ϕ_0 and initialize $\nu = 0$, $\phi = \phi_0$
2. Run Brown's algorithm for $B(\Lambda, \phi)$
3. Compute $d\nu$ with solution obtained in 2.
4. Update ν and ϕ by $\nu := \nu + d\nu$ and $\phi := \phi_0 + \tilde{A}\nu$
5. Return to 2 until convergence.

3.2 Optimal choice of $d\nu$

Differential behavior of $\min_{\varphi} \mathbf{P}_{nd}(\varphi)$: For each $\mathcal{U} \in \tau^{-1}(\mathcal{T})$, let $\mathcal{T}_{\mathcal{U}} = \mathcal{T} \setminus \tau(\mathcal{U})$, $t = \text{card}(\mathcal{T}_{\mathcal{U}})$ and:

$$\beta_{\mathcal{U}}^{\varphi}(x_{\tau(\mathcal{U})}) = \int_{E^t} \alpha(\vec{x}) \prod_{k \in \mathcal{T}} \prod_{\Upsilon \in \tau^{-1}(k) \setminus \mathcal{U}} p_{\Upsilon, x_k}(\varphi_{\Upsilon}(x_k)) \prod_{k \in \mathcal{T}_{\mathcal{U}}} dx_k .$$

Assume φ and $\varphi + d\varphi$ be optimal solutions for ϕ and $\phi + d\phi$ respectively. By definition, $d \min_{\varphi} \mathbf{P}_{nd}(\varphi)$ is given by $\mathbf{P}_{nd}(\varphi + d\varphi) - \mathbf{P}_{nd}(\varphi)$. A first order expansion relatively to $d\varphi$ results in linearization (28):

$$d \min_{\varphi} \mathbf{P}_{nd}(\varphi) = \sum_{\Upsilon \in \tau^{-1}(\mathcal{T})} \int_E \beta_{\Upsilon}^{\varphi}(x) \times (p_{\Upsilon, x}((\varphi_{\Upsilon} + d\varphi_{\Upsilon})(x)) - p_{\Upsilon, x}(\varphi_{\Upsilon}(x))) dx . \quad (28)$$

This value has to be as negative as possible. Since minimizing a sum reverts to minimizing its members, each $d\varphi_{\Upsilon}$ must minimize $\int_E \beta_{\Upsilon}^{\varphi}(x) p_{\Upsilon, x}((\varphi_{\Upsilon} + d\varphi_{\Upsilon})(x)) dx$. Such minimization is a kind of 1-period search problem on $\varphi_{\Upsilon} + d\varphi_{\Upsilon}$. Optimality conditions of de Guenin (see (7)) are also found. Since φ is optimal too, two de Guenin's conditions are obtained:

$$\begin{cases} \varphi_{\Upsilon}(x) > 0 \Rightarrow p'_{\Upsilon, x}(\varphi_{\Upsilon}(x)) \beta_{\Upsilon}^{\varphi}(x) = \eta_{\Upsilon} \\ (\varphi_{\Upsilon} + d\varphi_{\Upsilon})(x) > 0 \Rightarrow p'_{\Upsilon, x}(\varphi_{\Upsilon}(x) + d\varphi_{\Upsilon}(x)) \beta_{\Upsilon}^{\varphi}(x) = \eta_{\Upsilon} + d\eta_{\Upsilon} \end{cases} \quad (29)$$

Let $X_{\Upsilon}^{\varphi} \triangleq \{x \in E / \varphi_{\Upsilon}(x) > 0\}$, $W_{\Upsilon}^{\varphi} \triangleq X_{\Upsilon}^{\varphi+d\varphi} \cap X_{\Upsilon}^{\varphi}$ and $Y_{\Upsilon}^{\varphi} \triangleq X_{\Upsilon}^{\varphi+d\varphi} \cup X_{\Upsilon}^{\varphi}$. Two cases are considered:

Case $\varphi_{\Upsilon} > 0$: Subtraction of equations (29) yields:

$$\begin{cases} \forall x \in W_{\Upsilon}, \beta_{\Upsilon}^{\varphi}(x) p''_{\Upsilon, x}(\varphi_{\Upsilon}(x)) d\varphi_{\Upsilon}(x) = d\eta_{\Upsilon} , \\ \forall x \in E \setminus Y_{\Upsilon}, d\varphi_{\Upsilon}(x) = 0 . \end{cases}$$

Now, the border integrals $\int_{Y_{\Upsilon}^{\varphi} \setminus W_{\Upsilon}^{\varphi}} \varphi_{\Upsilon}(x) dx$ and $\int_{Y_{\Upsilon}^{\varphi} \setminus W_{\Upsilon}^{\varphi}} (\varphi_{\Upsilon}(x) + d\varphi_{\Upsilon}(x)) dx$ are second order infinitesimals, thus yielding :

$$\int_{X_{\Upsilon}^{\varphi}} \frac{dx}{\beta_{\Upsilon}^{\varphi}(x) p''_{\Upsilon, x}(\varphi_{\Upsilon}(x))} d\eta_{\Upsilon} = \int_E d\varphi_{\Upsilon}(x) dx .$$

Differentiating equation (10) we obtain $\int_E d\varphi_{\Upsilon}(x) dx = d\phi_{\Upsilon}$, so that:

$$d\eta_{\Upsilon} = d\phi_{\Upsilon} \left/ \int_{X_{\Upsilon}^{\varphi}} (\beta_{\Upsilon}^{\varphi}(x) p''_{\Upsilon,x}(\varphi_{\Upsilon}(x)))^{-1} dx \right. .$$

An expression of $d\varphi_{\Upsilon}(x)$ is then given by:

$$\forall x \in W_{\Upsilon}^{\varphi}, d\varphi_{\Upsilon}(x) = d\phi_{\Upsilon} \frac{\frac{1}{\beta_{\Upsilon}^{\varphi}(x) p''_{\Upsilon,x}(\varphi_{\Upsilon}(x))}}{\int_{X_{\Upsilon}^{\varphi}} \frac{dx}{\beta_{\Upsilon}^{\varphi}(x) p''_{\Upsilon,x}(\varphi_{\Upsilon}(x))}} .$$

Component Υ in sum (28) is thus rewritten:

$$\begin{aligned} \int_E \beta_{\Upsilon}^{\varphi}(x) (p_{\Upsilon,x}((\varphi_{\Upsilon} + d\varphi_{\Upsilon})(x)) - p_{\Upsilon,x}(\varphi_{\Upsilon}(x))) dx = \\ d\phi_{\Upsilon} \int_{X_{\Upsilon}^{\varphi}} \frac{p'_{\Upsilon,x}(\varphi_{\Upsilon}(x))}{p''_{\Upsilon,x}(\varphi_{\Upsilon}(x))} dx \left/ \int_{X_{\Upsilon}^{\varphi}} \frac{dx}{\beta_{\Upsilon}^{\varphi}(x) p''_{\Upsilon,x}(\varphi_{\Upsilon}(x))} \right. . \end{aligned} \quad (30)$$

Case $\phi_{\Upsilon} = 0$: Let $\eta_{\Upsilon}^0 = \max_{x \in E} (\beta_{\Upsilon}^{\varphi}(x) p'_{\Upsilon,x}(0))$. De Guenin's conditions yields:

$$\beta_{\Upsilon}^{\varphi}(x) \leq (\eta_{\Upsilon}^0 + d\eta_{\Upsilon}) / p'_{\Upsilon,x}(0) \Rightarrow d\varphi_{\Upsilon}(x) = 0 .$$

Then, as soon as $d\varphi_{\Upsilon} > 0$, property $\beta_{\Upsilon}^{\varphi}(x) p'_{\Upsilon,x}(0) = \eta_{\Upsilon}^0$ holds. Since $\varphi_{\Upsilon} = 0$ (deduced from $\phi_{\Upsilon} = 0$), equation (31) follows:

$$\begin{aligned} \int_E \beta_{\Upsilon}^{\varphi}(x) (p_{\Upsilon,x}((\varphi_{\Upsilon} + d\varphi_{\Upsilon})(x)) - p_{\Upsilon,x}(\varphi_{\Upsilon}(x))) dx = \\ \int_{\beta_{\Upsilon}^{\varphi}(x) \geq \frac{\eta_{\Upsilon}^0 + d\eta_{\Upsilon}}{p'_{\Upsilon,x}(0)}} \beta_{\Upsilon}^{\varphi}(x) p'_{\Upsilon,x}(0) d\varphi_{\Upsilon}(x) dx = \eta_{\Upsilon}^0 d\phi_{\Upsilon} . \end{aligned} \quad (31)$$

From properties (30) and (31), equality (28) becomes $d \min_{\varphi} \mathbf{P}_{nd}(\varphi) = \mathbf{V}^t d\phi$, with \mathbf{V} defined by (33). Next proposition is deduced from (27).

Property 2 *The variation of the probability of non detection around φ is given by:*

$$\boxed{d \min_{\varphi} \mathbf{P}_{nd}(\varphi) = \mathbf{V}^t A \tilde{d}\nu} \quad (32)$$

where \mathbf{V} is the vector defined by:

$$\left\{ \begin{array}{l} \mathbf{V}_\Upsilon = \frac{\int_{X_\Upsilon^\varphi} \frac{p'_{\Upsilon,x}(\varphi_\Upsilon(x))}{p''_{\Upsilon,x}(\varphi_\Upsilon(x))} dx}{\int_{X_\Upsilon^\varphi} \frac{dx}{\beta_\Upsilon^\varphi(x) p''_{\Upsilon,x}(\varphi_\Upsilon(x))}}, \text{ when } \phi_\Upsilon > 0 \text{ and } \Upsilon \in \tau^{-1}(\mathcal{T}) \\ \mathbf{V}_\Upsilon = \eta_\Upsilon^0, \text{ when } \phi_\Upsilon = 0 \text{ and } \Upsilon \in \tau^{-1}(\mathcal{T}) \\ \mathbf{V}_\Upsilon = 0, \text{ when } \tau(\Upsilon) = \mathbf{s} \end{array} \right. \quad (33)$$

Choice of $d\nu$: $d\nu$ is chosen, as a minimizer of $d \min_\varphi \mathbf{P}_{nd}(\varphi)$. Practically, discretization of $d\nu$, $\Delta\nu$, results in positivity gaps. Thus, $\Delta\nu$ is given by (34):

$$\begin{aligned} & \text{Minimize } \mathbf{V}^t A \tilde{\Delta\nu}, \\ & \text{subject to } A \tilde{\Delta\nu} + \phi \geq 0 \text{ and } \|A \tilde{\Delta\nu}\| \leq \Delta t. \end{aligned} \quad (34)$$

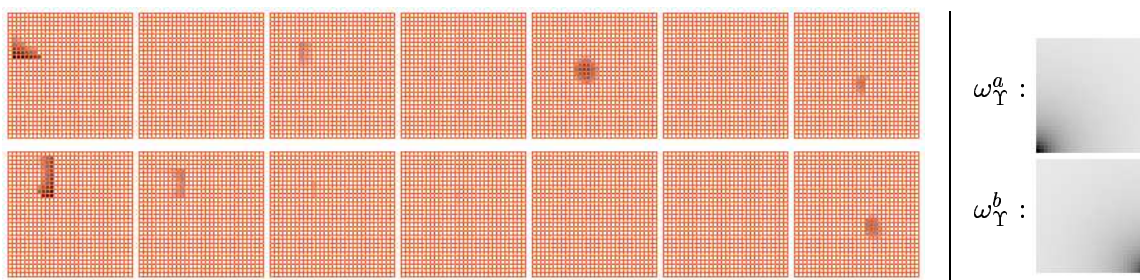
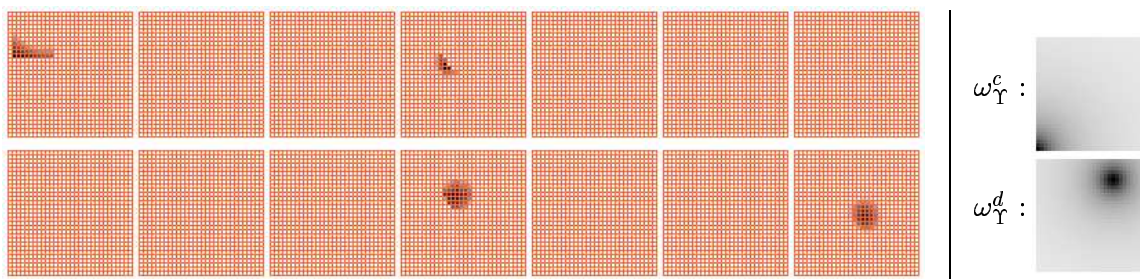
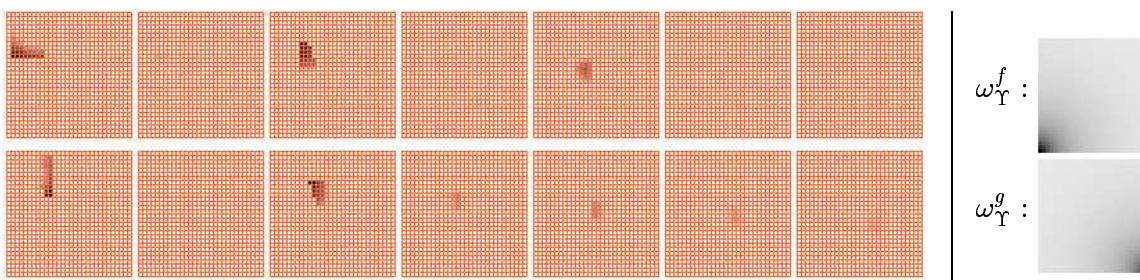
When $\| \cdot \| = \| \cdot \|_\infty$, (34) is linear and can be straightforwardly solved by means of the Simplex algorithm.

4 Results

Our algorithm is rather fast. Computation of $\Delta\nu$ is negligible, compared to Brown's process. Thus, the speed of our algorithm is of the same order than Brown's one. In this section, two examples are presented. They are produced with operators \otimes and \oplus from basic models of resources, and correspond to multi-resource and multi-mode scheduling. The space search E is here a square of 30×30 cells. The detection occurs during $T = 7$ periods. The target has a diffusion movement toward down and right. Its starting position is uniform in the top-left 10×10 square, centered around the point $(1, 1)$. So, the probabilistic prior about the target is $\alpha(\vec{x}) = s(x_1) \prod_{k=1}^6 m(x_{k+1} - x_k)$, where:

$$\begin{cases} s(x) = \frac{1}{100} \text{ if } (1, 1) \leq x \leq (10, 10), \quad s(x) = 0 \text{ else,} \\ m(0, 0) = m(3, 3) = \frac{3}{14}, \quad m(2, 3) = m(3, 2) = \frac{2}{14}, \\ m(0, 3) = m(3, 0) = m(1, 3) = m(3, 1) = \frac{1}{14}, \\ m(\mathbf{x}) = 0 \text{ else.} \end{cases}$$

All non detection function will be exponential, i.e. $p_{\Upsilon,x}(\varphi) = \exp(-\omega_{\Upsilon,x}\varphi)$. The visibility parameter $\omega_{\Upsilon,x}$ will not depend on the detection period.

Figure 1: First example: $\Lambda^a \otimes \Lambda^b$ Figure 2: Second example: $(2\Lambda^c) \oplus \Lambda^d \oplus -200$ Figure 3: Third example: $(\Lambda^e \succ (\Lambda^f \oplus \Lambda^g)) \otimes \Lambda^e \otimes \Lambda^f \otimes \Lambda^g$

Iter.	1	2	3	5	6	7	8	9	10
Proba.	16.67	16.80	16.82	16.83	16.84	16.85	16.86	16.87	16.89
Iter.	11	12	13	14	16	17	19	23	28
Proba.	16.92	16.94	16.95	16.97	16.98	16.99	17.00	17.01	17.02

Table 1: First example: evolution of detection probability versus the iteration.

First example: Two basic resources are used, corresponding to LS $\Lambda^a = \Lambda^{R4} \oplus -100$ and to LS $\Lambda^b = \Lambda^{R\infty} \oplus -200$. LS Λ^b represents non renewable resources with an amount equal to 200. Likewise, Λ^a represents resources renewable after 4 periods and with amount equal to 100. Non detection functions p^a and p^b are defined by their visibility parameters $\omega_{\Upsilon,x}^a$ and $\omega_{\Upsilon,x}^b$. Representations of $\omega_{\Upsilon,x}^a$ and $\omega_{\Upsilon,x}^b$ are given in figure 1. Moreover, p^a is stronger down-left, decreasing with the radius, while p^b is stronger down-right. Our algorithm has been applied to the resource scheduling of Λ^a and Λ^b . As already seen, such problem is described by ICLS $\Lambda^a \otimes \Lambda^b$. Results are viewed in figure 1. For resources a , the global amounts of resources, ϕ_{Υ}^a , are 78, 0, 22, 0, 78, 0, 22, when Υ varies from periods 1 to 7. The spatial distributions φ_{Υ}^a are represented in the same order in the upper part of figure 1. For resources b , the global amounts of resources are 110, 37, 6, 2, 0, 0, 44. The spatial distributions are represented in the lower part of figure 1. Note that the sum of four consecutive global amounts of resources a equals 100. In the same way, the sum of all global amounts of resources b equals 200, which agree with the renew constraints on a and b . Since the target is dispersive, it is natural that most resources are used in first periods. Again, since a renews after 4 periods, resources a appear at period 5. Now, visibility for b is better down-right and thus some resources are used at period 7. Spatially, resources a are rather placed down-left, since resources b are rather down-right, according with the visibility parameters. Some surrounding occurs at the first periods. The optimal spatial distribution is obtained roughly after 60 algorithm iterations. The evolution of detection probability is presented in table 1. In practice, the optimal probability is obtained after 28 iterations. It is also noteworthy that probability convergence is really quicker than the spatial distribution convergence. In other terms, an approximative spatial sharing is sufficient to handle a good probability detection. In this example, the initialization is really good and the detection probability does not really change. This is not representative of the general case.

Iter.	1	2	3	4	5	6	7	8	9	10
Proba.	12.49	12.53	12.55	12.57	12.61	12.67	12.76	12.88	13.05	13.30
Iter.	11	12	13	14	15	16	17	19	22	
Proba.	13.66	14.18	14.90	15.90	17.24	19.03	19.76	19.78	19.79	

Table 2: Second example: evolution of detection probability versus the iteration.

Second example: Two basic resource modes are used, corresponding to LS $\Lambda^c = \Lambda^{R3}$ and to LS $\Lambda^d = \Lambda^{R\infty}$. LS Λ^c represents also a 3-periods renewable resource mode. Likewise, Λ^d represents a non renewable resource mode. Functions p^c and p^d are represented in figure 2. p^c is stronger down-left, since p^d is stronger near the up-right side. Our algorithm has been applied to the mode scheduling of Λ^c and Λ^d . The resources can either run like resources c or like resources d , but it is assumed that mode c uses two times more resources than mode d . Total resource amount is fixed such that it equals 200, when running in mode d all the time. Such problem is described by ICLS $(2\Lambda^c) \oplus \Lambda^d \oplus -200$. Results are viewed in figure 2. For mode c , global amounts of resources are 100, 0, 0, 48, 0, 0, 0. The local distributions are represented by the upper part of figure 2. For mode d , global amounts of resources are 0, 0, 0, 104, 0, 0, 96. The local distributions are represented by the lower part of figure 2. It appears that it is better to use resources in renewable mode c at the beginning, and in non renewable mode d (d is more powerful and profitable but destroys resources) at the end. Since 1 resource c is equivalent to 2 resources d , $\frac{200}{2} = 100$ resources are used at period 1 in mode c . The resources used in mode c renew after 4 periods. That is why 48 resources in mode c and $2 \times 52 = 104$ resources in mode d become available at period 4. Then 48 resources renew from mode c at period 7 (no renew from mode d). Thus, at period 7, only $2 \times 48 = 96$ resources are eventually available in mode d . Spatially, resources c are rather placed down-left, whereas resources d are rather up-right. A slight surrounding occurs at period 1. The optimal spatial distribution is obtained roughly after 40 algorithm iterations. The evolution of detection probability is presented in table 2. In practice, the optimal probability is obtained after 22 iterations.

Third example: This last example involves three kinds of resources, say e , f and g . Resources e are 2-periods renewable with amount equal to 150. Their corresponding LS is $\Lambda^e = \Lambda^{R2} \oplus -150$. Resources f are non renewable with amount equal to 200. Their corresponding LS is $\Lambda^f = \Lambda^{R\infty} \oplus -200$. Resources g are gradually re-

newable following sequence $(\frac{1}{8}, \frac{1}{4}, \frac{1}{8})$. When 1 resource g is used at period k , then $\frac{1}{8}$ of this resource is renewed at period $k + 1$, $\frac{1}{4}$ at period $k + 2$ and $\frac{1}{8}$ at period $k + 3$. So, $\frac{1}{2}$ of the resource is definitively lost. The amount of these resources is 100. Their corresponding LS is then $\Lambda^g = \Lambda^R(\frac{1}{8}, \frac{1}{4}, \frac{1}{8}) \oplus -100$. Resource e will be used as a *control* resource for the two other, that means in this example that the use of either 1 resource f or 1 resource g requires the use of 1 resource e . To be more precise, it requires 2 resources e , when using at the same time 1 resource f and 1 resource g . Such condition is stated by LS $\Lambda^e \succ (\Lambda^f \oplus \Lambda^g)$. Otherwise, all resources are assumed to run independently, and the whole problem is represented by mean of ICLS $(\Lambda^e \succ (\Lambda^f \oplus \Lambda^g)) \otimes \Lambda^e \otimes \Lambda^f \otimes \Lambda^g$. Functions p^f and p^g are represented in figure 3. p^f is stronger down-left, while p^g is stronger down-right. Since resource e is only used for control, it accomplish no detection, so that $p^e = 0$. Results are viewed in figure 3. For mode e , global amounts of resources are 147, 3, 135, 15, 54, 16, 8. The local distributions are given by adding up the local distributions of f and g . For mode f , global amounts of resources are 84, 3, 74, 0, 37, 2, 0. The local distributions are represented by the upper part of figure 3. For mode g , global amounts of resources are 63, 0, 61, 15, 17, 14, 8. The local distributions are represented by the lower part of figure 3. The reader could verify that the constraints on resources are actually checked. It is rather uneasy to make comments on such complex example. Nevertheless, it is noteworthy that the resources f and g are not put mainly on the first period, where target dispersion begins. The reason is the lack of control resource e , which is limited by 150. Thus, the use of resources are more spread between the search periods. Surrounding aspect are stronger in this conditioned example.

5 Perspectives

Two main extensions come in mind. First, the current formalism supposes that detections are done independently, whatever the different resources are combined. This view point is somewhat naïve. It may happen that detection quality varies non linearly with the fusion of different sensors. This induces some generalization on the evaluation function of our problem. Nevertheless, in some simple cases, such problems are easily simulated by our current approach. They will be explained in section 5.1. A second important issue is to deal with the false alarm probability. A real treatment of such problem should involve dynamic optimization of the resources, that is to consider the evolution of the prior on target after each period of search (a false alarm is a change of prior, which have to be verified). Such optimization problem is really difficult. Now, our current problem is not dynamic. That is our

algorithm build a search plan and we will not change it whatever will happen. It is however possible to cope with false alarms, by introducing an evaluation function as well as a specific constraint to handle it. This aspect induces important changes in our current Brown's formalism (see section 5.2).

5.1 Complex local interaction between resources

In equation (13), a strong hypothesis about independence of resources was done. To handle complex interaction between resources (for same period and same location), it is necessary to use non detection function p_{k,x_k} which involves all the resource investments φ_{Υ} for the current period and the current cell. The function of non detection is then changed into the new following one:

$$\mathbf{P}_{nd}(\varphi) = \int_{E^T} \alpha(\vec{x}) \prod_{k=1}^{k=T} p_{k,x_k}(\varphi_{\Upsilon}(x_k) | \Upsilon \in \tau^{-1}(k)) dx_k \quad (35)$$

All the other parts of the ICLS are kept unchanged. Solving such problems is a bit more complex, but FAB algorithm principle should still apply, since we keep independence between each period of search. However, the real difficulty lies in the choice of the non detection functions p_k . Realistic functions may be difficult to construct and may also not be convex. Nevertheless, it is possible to simulate simple problem of interacting resources by means of our current formalism (see example below).

Triptych example: Resources combination can be handled by our formalism, when these combinations are planned *separately*. Assume three types of resources be given, say Λ^1 , Λ^2 , Λ^3 (LSs will be specified later). Each of these resource types are split into two subsets of resources. Resources Λ^1 is also split into resource sets $\Lambda^{1\beta}$ and $\Lambda^{1\gamma}$, resources Λ^2 is also split into resource sets $\Lambda^{2\gamma}$ and $\Lambda^{2\alpha}$ and resources Λ^3 is also split into resource sets $\Lambda^{3\alpha}$ and $\Lambda^{3\beta}$. Of course, resources Λ^{ij} remain of the same nature than resources Λ^i . These splittings are obviously conditioned by the semantical LSs $\Lambda^1 \succ (\Lambda^{1\beta} \oplus \Lambda^{1\gamma})$, $\Lambda^2 \succ (\Lambda^{2\gamma} \oplus \Lambda^{2\alpha})$ and $\Lambda^3 \succ (\Lambda^{3\alpha} \oplus \Lambda^{3\beta})$. Now, we dispose of subsets of our basic resources. These subsets are then combined one to one, so as to obtain a *combined mode* of use of these resources. In this example, it is assumed that 1 combination of two resources involves 1 amount of the first resource and 1 amount of the second resource. Denote these combinations by the LSs Λ^α , Λ^β and Λ^γ . Because of the preceding assumptions, we obtain the three conditioning

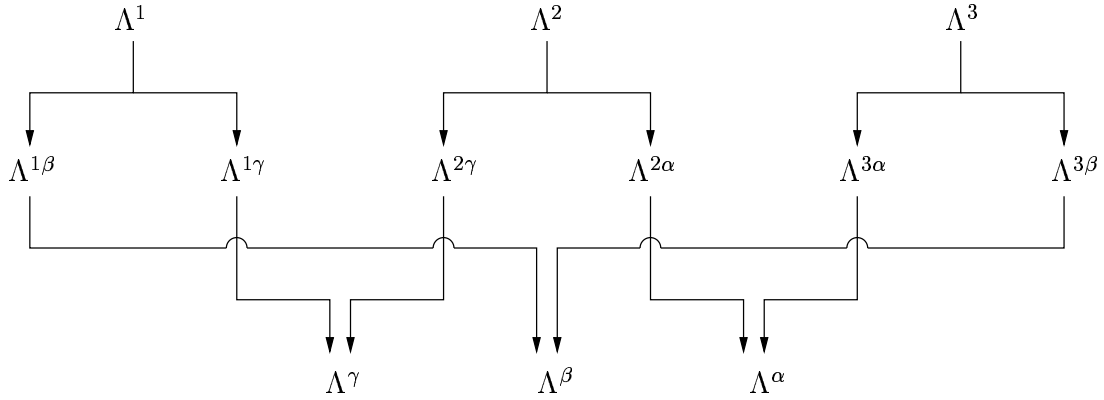


Figure 4: Diagram of the resource combination

LSs $(\Lambda^{3\alpha} \otimes \Lambda^{2\alpha}) \succ \Lambda^\alpha$, $(\Lambda^{1\beta} \otimes \Lambda^{3\beta}) \succ \Lambda^\beta$ and $(\Lambda^{2\gamma} \otimes \Lambda^{1\gamma}) \succ \Lambda^\gamma$. The whole partitioning procedure is summarized in figure 4. Our detection will be accomplished by means of these combined modes only. Thus, non detection functions p^α , p^β and p^γ are considered. These functions are represented in figure 5, and $\mathbf{P}_{nd}(\varphi)$ is given by:

$$\mathbf{P}_{nd}(\varphi) = \int_{E^T} \alpha(\vec{x}) \prod_{k=1}^{k=T} \left(p_{k,x_k}^\alpha(\varphi_k^\alpha(x_k)) p_{k,x_k}^\beta(\varphi_k^\beta(x_k)) p_{k,x_k}^\gamma(\varphi_k^\gamma(x_k)) dx_k \right) \quad (36)$$

The elementary LS Λ^1 , Λ^2 and Λ^3 are: $\Lambda^1 = \Lambda^{R2} \oplus -100$, $\Lambda^2 = \Lambda^R(\frac{1}{8}, \frac{1}{4}, \frac{1}{8}) \oplus -200$ and $\Lambda^3 = \Lambda^{R4} \oplus -300$. The whole constraints are given then by ICLS:

$$\Lambda^1 \otimes \Lambda^2 \otimes \Lambda^3 \otimes (\Lambda^1 \succ (\Lambda^{1\beta} \oplus \Lambda^{1\gamma})) \otimes (\Lambda^2 \succ (\Lambda^{2\gamma} \oplus \Lambda^{2\alpha})) \otimes (\Lambda^3 \succ (\Lambda^{3\alpha} \oplus \Lambda^{3\beta})) \otimes ((\Lambda^{3\alpha} \otimes \Lambda^{2\alpha}) \succ \Lambda^\alpha) \otimes ((\Lambda^{1\beta} \otimes \Lambda^{3\beta}) \succ \Lambda^\beta) \otimes ((\Lambda^{2\gamma} \otimes \Lambda^{1\gamma}) \succ \Lambda^\gamma)$$

Results are viewed in figure 5. For mode α , 2α and 3α global amounts of resources are 141, 25, 4, 38, 0, 19, 12. The local distributions of α are represented by the upper part of figure 5. For mode β , 3β and 1β global amounts of resources are 41, 0, 51, 0, 79, 0, 100. The local distributions of β are represented by the upper part of figure 5. For mode γ , 1γ and 2γ global amounts of resources are 59, 0, 49, 0, 21, 0, 0. The local distributions of γ are represented by the lower part of figure 5.

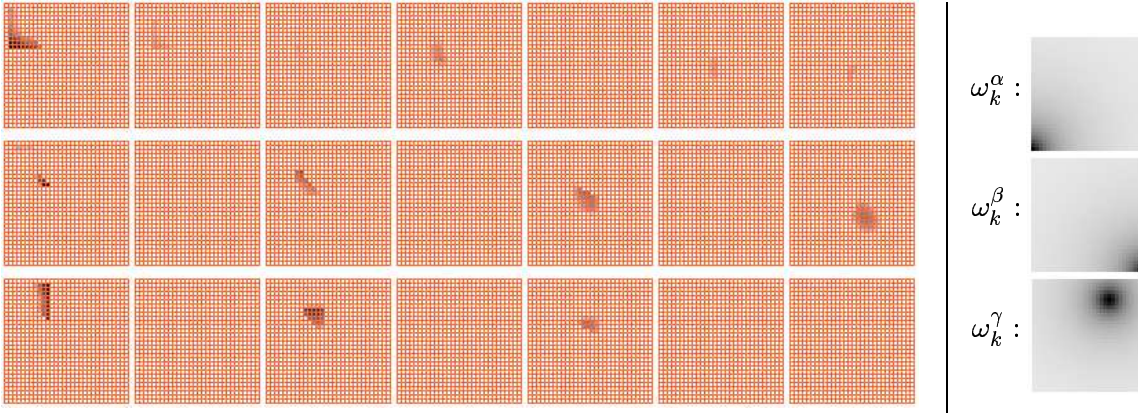


Figure 5: Triptych example.

5.2 Dealing with false alarm probability

A real treatment of false alarm problems requires probably to handle a dynamic approach of the theory. Nevertheless, we propose in this section a simplified formalism for false alarm problems, dealing with a non dynamic point of view. The idea is to use detection resources, indexed by D , together with checking resources, indexed by C . When $\varphi_k^D(x)$ resources of detection and $\varphi_k^C(x)$ checking resources are used at period k and on cell $x \in E$, the false alarm is evaluated by function $g_{k,x}(\varphi_k^D(x), \varphi_k^C(x))$. The availability of checking resources is planned before managing the detection. This plan consists of spatial and temporal splitting $\varphi_k^C(x)$, with respect to a certain probability of false alarm $f_k(x)$ for each period k and cell $x \in E$. A false alarm constraint is imposed so as to bound the false alarm evaluation function, i.e.:

$$\int_E f_k(x) g_{k,x}(\varphi_k^D(x), \varphi_k^C(x)) dx \leq \zeta_k .$$

This probability f is usually independent of the prior α of the target. For mathematical convenience, the preceding inequality will be changed into equality, by deferring the inequality to the LS matrix constraint. More precisely, assume that we are deal-

ing with a simple resource over T periods. The problem may be written as follows:

$$\begin{aligned}
 \text{Minimize: } \quad & \mathbf{P}_{nd}(\varphi^D) = \int_{E^T} \alpha(\vec{x}) \prod_{k=1}^{k=T} (p_{\kappa, x_k}^D(\varphi_k^D(x_k)) dx_k) \\
 \text{Subject to: } \quad & \varphi^D \geq 0, \phi^D \geq 0, \varphi^C \geq 0, \phi^C \geq 0, \\
 & \int_E \varphi_k^D(x) dx = \phi_k^D, \int_E \varphi_k^C(x) dx = \phi_k^C, \\
 & \int_E f_k(x) g_{k,x}(\varphi_k^D(x), \varphi_k^C(x)) dx \leq \zeta_k, \\
 & A(\phi^{Dt}, \phi^{Ct}, \zeta^t)^t \leq \psi.
 \end{aligned} \tag{37}$$

Such problem is very close to de Guenin and Brown's ones. It could be easily proved that de Guenin's equations still hold, when the frontier of the problem is not reached, that is $\beta_{\kappa}^{\varphi^D}(x) p'_{\kappa, x}(\varphi_{\kappa}^D(x)) = \eta_{\kappa}$, for a given period k , when $\varphi_k^D(x) > 0$ and $\varphi_k^C(x) > 0$. Nevertheless, border statements are more intricated, and some changes on the basic Brown's algorithm are necessary.

6 Conclusion

This paper is centered around the management of complex detection systems. For that purpose, an original formalism has been developed, it seems sufficiently general to handle a wide variety of sensor management problems. Its underlying semantic even permits to include conditional aspects for resource allocation problems and to introduce controls in lateral splitting. All these points have been considered in a general setting. Using it, the corresponding optimization problems are quite efficiently solved by means of an extension of the Brown's algorithm. This algorithm inherits the robustness of the original one and its computation load is quite modest. Specific applications should be detection/tracking of moving targets. In these areas (radar, sonar, IR, etc.), more work has to be done, especially in the adequacy of elementary sensor detection models as well as in the modeling of fused detections, but our approach seems sufficiently open.

References

- [1] B.O. KOOPMAN, *Search and Screening: General Principle with Historical Applications*. MORS Heritage Series, Alexandria, VA, 1999.
- [2] A.R. WASHBURN, *Search and Detection, 2-nd edition*. Operations Research Soc. of America, ORSA Books, Arlington, VA, 1989.
- [3] S.J. BENKOVSKI, M.G. MONTICINO and J.R. WEISINGER, *A Survey of the Search Theory Literature*. Naval Research Logistics, vol.-38, pp. 469-491, 1991.
- [4] L.D. STONE, *Theory of Optimal Search, 2-nd ed.* . Operations Research Society of America, Arlington, VA, 1989.
- [5] K. IIDA, *Studies on the Optimal Search Plan*. Lecture Notes in Statistics, vol. 70, Springer-Verlag, 1992.
- [6] J. de GUENIN, *Optimum Distribution of Effort: an Extension of the Koopman Basic Theory*. Operations Research 9, pp 1-7, 1961.
- [7] S.S. BROWN, *Optimal Search for a Moving Target in Discrete Time and Space*. Operations Research 28, pp 1275-1289, 1980.
- [8] A.R. WASHBURN, *Search for a moving Target: The FAB algorithm*. Operations Research 31, pp 739-751, 1983.
- [9] T. IBARAKI and N. KATOH, *Resource Allocation Problems: Algorithmic Approaches*. MIT Press, 1988.
- [10] D.H. WAGNER, W.C. MYLANDER and T.J. SANDERS eds, *Naval Operations Analysis (3-rd edition)*, Chapt. 5. Naval Institute Press, Annapolis, MD, 1999.
- [11] B.O. KOOPMAN, *Search and Screening: General Principle with Historical Applications*. Pergamon Press, New York, 1980.
- [12] W.R. STROMQUIST and L.D. STONE, *Constrained Optimization of Functionals with Search Theory Applications*. Mathematics of Operations Research, Vol. 6, n°4, pp. 518-519, Nov. 1981.
- [13] P.K. VARSHNEY, *Distributed Detection and Data Fusion*. Springer-Verlag 1997.
- [14] H.V. POOR, *An Introduction to Signal Detection and Estimation, 2nd Edition*. Springer-Verlag 1994.

- [15] H.L. VAN TREES, *Detection, Estimation and Modulation Theory, Part I*. Wiley, New-York, 1968.



Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irista, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
<http://www.inria.fr>
ISSN 0249-6399