

**Magistère de Mathématiques d'Orsay (Paris-Sud XI)**

Parcours Mathématiques Fondamentales et Appliquées

Options : Probabilités et Statistiques, Informatique Théorique et  
Mathématiques Appliquées

# **MEMOIRE DE MAGISTERE :**

**promotion 2009-2012**

---

**Nicolas Drougard**

Toulouse, le 19 octobre 2012

## TABLE DES MATIÈRES

|   |     |
|---|-----|
| 1. Introduction   | 2   |
| 2. Travaux de deuxième année  | 2   |
| 2.1. TER : les Estimateurs à Noyau (statistiques)                   | 3   |
| 2.2. Informatique Théorique : les Graphes Expander                  | 34  |
| 3. Stage de Magistère   | 53  |
| 4. Travaux de dernière année  | 78  |
| 4.1. Traitement d'Image : Synthèse de Textures                      | 78  |
| 4.2. Processus de Markov à Décision : suite du stage                | 94  |
| 4.3. PhyloHMM   | 108 |
| 5. Présentation de mon domaine de recherche et stage de fin d'étude | 118 |

## 1. INTRODUCTION

Ce mémoire de Magistère de Mathématiques est organisé autour des différents travaux que j'ai effectués durant ma formation. C'est un bref aperçu de l'expérience qu'un étudiant peut vivre durant ces trois années dédiées aux Mathématiques.

Mes deux premières années post bac m'ont menées vers le Magistère de Mathématiques d'Orsay. Ayant besoin d'une approche plus détaillée des théories physiques, je quitte la première année d'une classe préparatoire classique Maths-Physique pour la deuxième année de classe préparatoire universitaire Physique-Maths. L'approche entreprise par cette dernière formation répond si bien à mes attentes que j'obtiens alors une vision plus claire me permettant de choisir d'entreprendre une formation de Mathématiques : pourquoi pas un Magistère de Mathématiques ?

Ma première année de Magistère (dernière année de licence) était cruciale puisque organisée dans le but d'assurer une base générale en Mathématiques pour les années suivantes : Topologie, Calcul Différentiel, Equations Différentielles, Fonctions Holomorphes, Théorie de Fourier, Algèbre, Probabilités, Algorithmique, Informatique Théorique, Théorie des graphes... La vision d'ensemble donnée par cette année laborieuse m'a permis de réfléchir à une première orientation : la théorie des Probabilités ainsi que l'Informatique m'avaient déjà séduit.

## 2. TRAVAUX DE DEUXIÈME ANNÉE

Cet engouement m'a mené aux choix de mes spécialisations pour la seconde année de Magistère (première année de Master) : Probabilités puis Statistiques (ainsi que les Mathématiques Appliquées qui en découlent), et

enfin un approfondissement d'Informatique théorique. Cette spécialisation est bien illustrée par les deux projets que j'ai dû effectuer cette année là.

**2.1. TER : les Estimateurs à Noyau (statistiques).** Un cours très théorique de Statistiques nous était dispensé, en parallèle duquel j'ai effectué mon TER (Travail Encadré de Recherche) : le cours restant principalement autour de l'estimation paramétrique, j'ai choisi le travail proposé par Thanh Mai PHAM NGOC dans lequel il était question d'estimation non-paramétrique, qu'il fallait illustrer par des simulations informatiques. Le rapport qui suit résume le travail effectué : j'y présente une famille d'estimateurs non paramétriques appelés « Estimateurs à Noyaux ».

Professeur référente : Thanh Mai PHAM NGOC  
Université Paris-Sud XI  
TER (projet Master 1)

# Les estimateurs à noyau

---

Nicolas Drougard

Orsay, juin 2011

# Table des matières

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Présentation du problème</b>                                  | <b>2</b>  |
| <b>2</b> | <b>Estimateurs à noyau de la densité</b>                         | <b>3</b>  |
| 2.1      | Recherche heuristique d'estimateurs . . . . .                    | 3         |
| 2.2      | Premières simulations . . . . .                                  | 4         |
| <b>3</b> | <b>La méthode de validation croisée</b>                          | <b>6</b>  |
| 3.1      | Construction de $h \mapsto CV(h)$ . . . . .                      | 6         |
| 3.2      | Utilisation de la méthode . . . . .                              | 9         |
| 3.3      | Hypothèses et résultat de Stone . . . . .                        | 11        |
| 3.3.1    | Hypothèses . . . . .   | 11        |
| 3.3.2    | Résultat de Stone . . . . .                                      | 11        |
| 3.3.3    | Existence d'un $h > 0$ minimisant $h \mapsto ISE(h) : h_{ise}$ . | 12        |
| 3.4      | Problème lié à la discrétisation des observations . . . . .      | 15        |
| <b>4</b> | <b>Validation croisée type maximum de vraisemblance</b>          | <b>18</b> |
| 4.1      | Présentation . . . . .   | 18        |
| 4.2      | Une limite de cette méthode . . . . .                            | 19        |
| 4.3      | Première utilisation . . . . .                                   | 19        |
| <b>5</b> | <b>Utilisation de ces outils</b>                                 | <b>21</b> |
| 5.1      | Mise en pratique sur une densité connue . . . . .                | 21        |
| 5.1.1    | Cross validation . . . . .                                       | 22        |
| 5.1.2    | Likelihood cross validation . . . . .                            | 24        |
| 5.2      | Application à des observations dont la loi est inconnue . . . .  | 26        |

## 1 Présentation du problème

Nous avons accès à la réalisation de variables aléatoires réelles  $\{X_i\}_{i \in \{1 \dots n\}}$  indépendantes et identiquement et distribuées dont nous faisons l'hypothèse qu'elles possèdent une densité  $\mathbf{p}$ .

Une approche classique afin d'estimer la densité de ces variables aléatoires est de faire une hypothèse beaucoup plus restrictive : supposer que la densité en question appartient à une famille  $\{p_\theta\}$  avec  $\theta \in R^k$  ( $k \in N$ ). L'estimation de la densité se résume alors à l'estimation de  $\theta$ . Ce type d'étude est nommé *estimation paramétrique*, le paramètre à estimer vivant dans un espace fini-dimensionnel.

Cependant, faire ce type d'hypothèse paraît difficile en pratique si l'on manque d'information sur les variables aléatoires observées  $\{X_i\}_{i \in \{1 \dots n\}}$ . Nous allons donc nous intéresser à *l'estimation non-paramétrique* de cette densité : nous souhaitons trouver un bon estimateur de  $\mathbf{p}$ , en supposant seulement que  $\mathbf{p}$  est une densité absolument continue par rapport à la mesure de Lebesgue. Nous aurons aussi parfois besoin de rajouter quelques hypothèses de régularité qui laisseront  $\mathbf{p}$  dans un espace de fonctions ne réduisant pas le problème à une étude paramétrique.

## 2 Estimateurs à noyau de la densité

### 2.1 Recherche heuristique d'estimateurs

Les seules informations à notre disposition sont les réalisations des variables étudiées  $\{X_i\}_{i \in \{1 \dots n\}}$  ainsi que le fait qu'elles possèdent une densité dominée par la mesure de Lebesgue.

Considérons la fonction de répartition empirique construite à l'aide de ce n-échantillon :  $F_n = \frac{1}{n} \sum_{i=1}^n 1_{[X_i, +\infty[}$ . Elle nous est accessible puisqu'elle dépend uniquement des  $n$  observations.

La loi forte des grands nombres nous permet d'affirmer que  $F_n$  converge presque sûrement vers  $F(t) = \int_{-\infty}^t \mathbf{p}(x)dx$ , la fonction de répartition de la loi à densité recherchée.

Si nous supposons que  $\mathbf{p}$  était continue, on aurait trivialement  $\mathbf{p} = F'$ .

Ces remarques nous amènent à considérer un **taux d'accroissement de la fonction de répartition empirique  $F_n$** .

En effet, en raisonnant naïvement, ce taux d'accroissement devrait être proche de la dérivée de  $F_n$ . Comme  $\forall x \in R$ ,  $F_n(x)$  converge presque sûrement vers  $F(x)$ , on peut espérer qu'un taux d'accroissement de  $F_n(x)$  soit proche de  $\mathbf{p}$ .

Explicitons ce taux d'accroissement :

$$\hat{r}_n = \frac{F_n(x+h) - F_n(x-h)}{2h}$$

pour  $h > 0$ .

On appelle cet estimateur l' **estimateur de Rosenblatt**.  
Il est possible de l'écrire :

$$\begin{aligned}\hat{r}_n(x) &= \frac{1}{2nh} \sum_{i=1}^n 1_{X_i \leq x+h} - 1_{X_i \leq x-h} = \frac{1}{2nh} \sum_{i=1}^n 1_{x-h < X_i \leq x+h} \\ &= \frac{1}{2nh} \sum_{i=1}^n 1_{-1 < \frac{X_i - x}{h} \leq 1} = \frac{1}{nh} \sum_{i=1}^n K_R\left(\frac{X_i - x}{h}\right)\end{aligned}$$

où  $K_R(x) = \frac{1}{2} 1_{-1 < x \leq 1}$ .

On appelle :

- $K_R$  le noyau (rectangulaire) qui est d'intégrale 1
- $h > 0$  la fenêtre

Plus généralement, définissons les **estimateurs à noyau**, les estimateurs de densité de la forme :

$$\hat{p}_n = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{X_i - x}{h}\right) \quad (1)$$

avec le noyau  $K : \mathbb{R} \rightarrow \mathbb{R}$  mesurable tel que  $\int K(t)dt = 1$ , et la fenêtre  $h > 0$ .

Notre travail consiste alors à trouver un noyau  $K$  et une fenêtre  $h$  qui font de  $\hat{p}_n$  un "bon" estimateur.

## 2.2 Premières simulations

Considérons l'estimateur de Rosenblatt  $\hat{r}_n$  (ie l'estimateur à noyau construit avec le noyau  $K_R$ ) et supposons que l'on étudie (sans le savoir !) des variables aléatoires suivant une loi normale centrée réduite.

La figure 1 représente trois réalisations de  $\hat{r}_n$  (en trait plein) construit avec  $n=100$  observations, et respectivement les fenêtres  $h=0.1$ ,  $h=1$  et  $h=10$ . La densité de la loi normale centrée réduite est en pointillé.

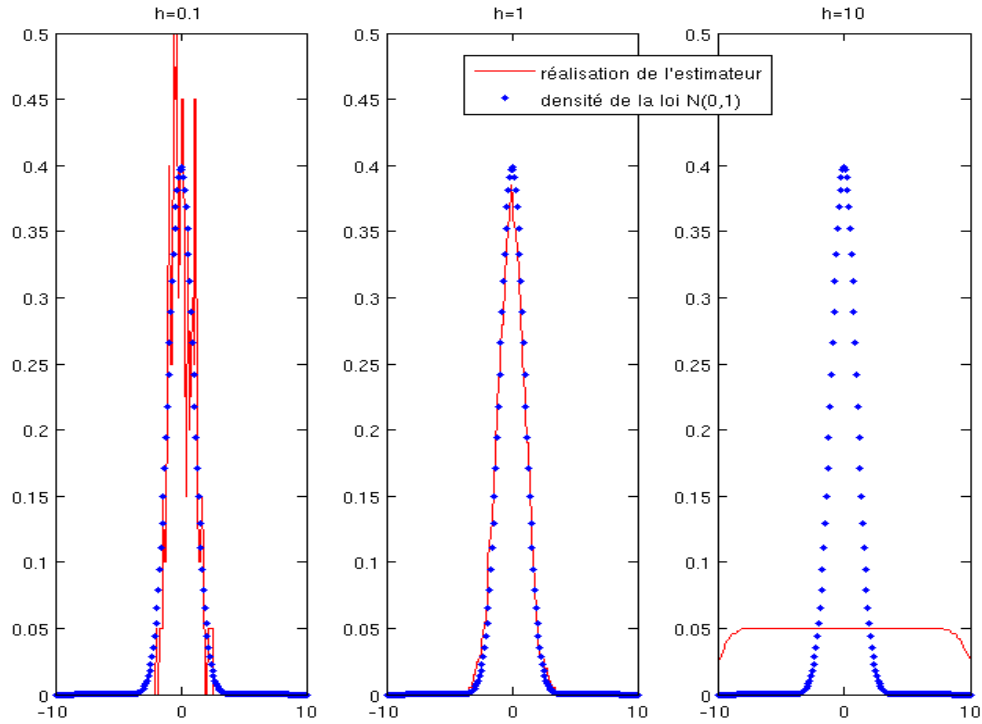


FIGURE 1 – 3 réalisations de  $\hat{r}_n$ ,  $n=100$ ,  $h=0.1$ ,  $h=1$ ,  $h=10$  (MATLAB)

Les phénomènes rencontrés pour  $h=0.1$  et  $h=10$  sont appelés respectivement sous-lissage et sur-lissage (**under/oversmoothing**).

Ces premiers essais nous montrent l'importance du choix de la fenêtre  $h$  :  
**comment choisir une fenêtre adaptée ?**

Introduisons le **risque quadratique intégré** de  $\hat{p}_n$   
(ou Mean Integrated Squared Error) :

$$MISE := E_p \left[ \int (\hat{p}_n(x) - \mathbf{p}(x))^2 dx \right]. \quad (2)$$

Notre but est évidemment de minimiser cette quantité afin que l'estimateur à noyau  $\hat{p}_n$  soit le plus proche possible (en moyenne sous  $\mathbf{p}$  et au sens de la distance  $L^2$ ) de la densité réelle .



### 3 La méthode de validation croisée

Tout comme  $\hat{p}_n$ , le risque quadratique intégré dépend de la fenêtre  $h$ . Nous cherchons un  $h > 0$  pour lequel  $MISE(h)$  est minimal.

Cependant, le calcul du MISE demande la connaissance de la densité que l'on cherche à approcher. Nous allons donc construire une fonction  $CV(h)$  qui dépend seulement des observations  $\{X_i\}_{i \in \{1 \dots n\}}$  et dont la minimisation en  $h$  donne une fenêtre  $h_{cv}$  proche de celle donnée par la minimisation en  $h$  "théorique" du  $MISE(h)$  fournissant  $h_{mise}$ . ("CV" pour "Cross Validation", nom de la méthode en anglais)

**Remarque :** Encore une fois,  $h_{mise}$  dépend de  $\mathbf{p}$ , densité que nous ne connaissons pas, et même que nous cherchons à déterminer.  $h_{mise}$  n'est donc pas accessible dans la pratique, d'où la recherche de  $h_{cv}$ .

#### 3.1 Construction de $h \mapsto CV(h)$

1<sup>ère</sup> étape,  $CV_1$  : si  $\int \mathbf{p}^2(x)dx < \infty$ ,

$$\begin{aligned} MISE(h) &:= E_p[\int (\hat{p}_n(x) - \mathbf{p}(x))^2 dx] \\ &= E_p[\int \hat{p}_n^2(x) dx - 2 \int \hat{p}_n(x) \mathbf{p}(x) dx] + \int \mathbf{p}^2(x) dx \end{aligned}$$

Ainsi, si  $\int \mathbf{p}^2(x)dx < \infty$ , trouver une fenêtre qui minimise le MISE est équivalent à en trouver une qui minimise

$$CV_1(h) := E_p[\int \hat{p}_n^2(x) dx - 2 \int \hat{p}_n(x) \mathbf{p}(x) dx] \quad (3)$$

puisque  $\int \mathbf{p}^2(x)dx$  ne dépend pas de la fenêtre  $h$ .

2<sup>ème</sup> étape,  $CV_2$  : Maintenant, cherchons à enlever  $\mathbf{p}$  de l'expression puisque nous n'y avons pas accès. Pour cela définissons :

$$\hat{p}_{n,-i}(x) := \frac{1}{(n-1)h} \sum_{j \neq i} K\left(\frac{X_j - x}{h}\right) \quad (4)$$

qui n'est rien d'autre que l'estimateur de la densité si l'on avait pas l'observation  $X_i$ .

Ainsi, la nouvelle expression

$$CV_2(h) := \int \widehat{p}_n^2(x) dx - \frac{2}{n} \sum_{i=1}^n \widehat{p}_{n,-i}(X_i) \quad (5)$$

ne dépend que des observations  $\{X_i\}_{i \in \{1 \dots n\}}$ ,  
et nous allons voir que c' est un **estimateur sans biais** de  $CV_1(h)$   
dès lors que  $(x, y) \mapsto K(\frac{x-y}{h})\mathbf{p}(x)\mathbf{p}(y)$  est intégrable.

---

Remarquons que dans ce cas,

$$\forall i \in \{1 \dots n\}, E_p[\widehat{p}_{n,-i}(X_n)]$$

$$= E_p\left[\frac{1}{(n-1)h} \sum_{j \neq i} K\left(\frac{X_j - X_i}{h}\right)\right] = \frac{1}{h} \int \int K\left(\frac{x-y}{h}\right) \mathbf{p}(x) \mathbf{p}(y) dx dy$$

$$\text{Ainsi, } E_p\left[\frac{1}{n} \sum_{i=1}^n \widehat{p}_{n,-i}(X_i)\right]$$

$$= E_p[\widehat{p}_{n,-n}(X_n)] = \frac{1}{h} \int \int K\left(\frac{x-y}{h}\right) \mathbf{p}(x) \mathbf{p}(y) dx dy \quad (6)$$

$$\text{et d'autre part, } E_p\left[\int \widehat{p}_n(x) p(x) dx\right]$$

$$= E_p\left[\int \frac{1}{nh} \sum_{i=1}^n K\left(\frac{X_i - x}{h}\right) p(x) dx\right] = \frac{1}{h} \int \int K\left(\frac{x-y}{h}\right) \mathbf{p}(x) \mathbf{p}(y) dx dy \quad (7)$$

par le théorème de Fubini (version intégrable).

On a donc  $E_p\left[\int \widehat{p}_n(x) p(x) dx\right] = E_p\left[\frac{1}{n} \sum_{i=1}^n \widehat{p}_{n,-i}(X_i)\right]$  par (6) et (7),

ce qui donne  $CV_1(h) = E_p\left[\int \widehat{p}_n^2(x) dx\right] - 2E_p\left[\int \widehat{p}_n(x) p(x) dx\right]$

$$= E_p\left[\int \widehat{p}_n^2(x) dx\right] - 2E_p\left[\frac{1}{n} \sum_{i=1}^n \widehat{p}_{n,-i}(X_i)\right] = E_p[CV_2(h)] \quad (8)$$

par le calcul précédent.

$CV_2(h)$  est donc bien un estimateur sans biais de  $CV_1(h)$

Les dernières modifications du  $MISE(h)$  ont pour but de rendre le calcul et l'expression de la fonction finale  $CV(h)$  plus simple :

3<sup>eme</sup> **étape** : la première partie de  $CV_2(h)$  étant une intégrale, il est préférable de la simplifier pour limiter les calculs lors d' une simulation informatique. Définissons  $K^{(2)}$  la convolution du noyau  $K$  avec lui-même et rajoutons comme condition sur  $K$  qu'il soit symétrique.

$$\begin{aligned}\int \widehat{p}_n^2(x)dx &= \int \frac{1}{nh} \sum_{i=1}^n K\left(\frac{X_i-x}{h}\right) \frac{1}{nh} \sum_{j=1}^n K\left(\frac{X_j-x}{h}\right) dx \\ &= \int \frac{1}{(nh)^2} \sum_{i=1}^n \sum_{j=1}^n K\left(\frac{X_i-x}{h}\right) K\left(\frac{X_j-x}{h}\right) dx\end{aligned}$$

par le changement de variable  $u = \frac{X_j - x}{h}$  ( $dx=hd u$ ),  
 $x = X_j - hu$ , on obtient

$$\begin{aligned}\int \widehat{p}_n^2(x)dx &= \frac{1}{n^2h} \sum_{i=1}^n \sum_{j=1}^n \int K\left(\frac{X_i-X_j+hu}{h}\right) K(u) du \\ &= \frac{1}{n^2h} \sum_{i=1}^n \sum_{j=1}^n \int K\left(\frac{X_i-X_j}{h} + u\right) K(u) du \\ &= \frac{1}{n^2h} \sum_{i=1}^n \sum_{j=1}^n \int K\left(\frac{X_j-X_i}{h} - u\right) K(u) du \text{ car } K \text{ est symétrique.} \\ &= \frac{1}{n^2h} \sum_{i=1}^n \sum_{j=1}^n K^{(2)}\left(\frac{X_j - X_i}{h}\right)\end{aligned}$$

Sous cette nouvelle condition, on obtient

$$CV_2(h) = \frac{1}{n^2h} \sum_{i,j=1}^n K^{(2)}\left(\frac{X_i-X_j}{h}\right) - \frac{2}{n} \sum_{i=1}^n \widehat{p}_{n,-i}(X_i)$$

**Remarque** : L'introduction de  $K^{(2)}$  permet d'éviter le calcul de l'intégrale  $\int \widehat{p}_n^2(x)dx$ , mais il devient nécessaire de connaître son expression.

Par exemple, considérons le noyau triangulaire  $K_T(x) = (1 - |x|)1_{|x| \leq 1}$ .

L'expression de  $K_T$  étant simple, on peut s'attendre à calculer  $K_T^{(2)}$  facilement. Il n'en est rien, et sauf erreur de ma part,

$$K_T^{(2)}(x) = \left(\frac{2}{3} - x^2 + \frac{|x|^3}{2}\right)1_{|x| \leq 1} + \left(\frac{4}{3} - 2|x| + x^2 - \frac{|x|^3}{6}\right)1_{1 < |x| \leq 2}$$

4<sup>eme</sup> **et dernière étape, CV** :

Enfin, nous simplifions l'écriture de la deuxième partie de  $CV_2$  :

$$\begin{aligned}\frac{1}{n} \sum_{i=1}^n \hat{p}_{n,-i}(X_i) &= \frac{1}{n} \sum_{i=1}^n \frac{1}{(n-1)h} \sum_{j \neq i} K\left(\frac{X_j - X_i}{h}\right) \\ &= \frac{1}{n} \frac{1}{(n-1)h} \sum_{i=1}^n \sum_{j=1}^n K\left(\frac{X_j - X_i}{h}\right) - \frac{1}{(n-1)h} K(0)\end{aligned}$$

(on a simplement ôté les cas  $i=j$ ).

Remplaçons maintenant les  $(n-1)$  par des  $n$  ; l'expression devient

$$\frac{1}{n^2 h} \sum_{i,j=1}^n K\left(\frac{X_i - X_j}{h}\right) - \frac{1}{nh} K(0)$$

Toutes ces modifications nous mènent enfin à

$$CV(h) := \frac{1}{n^2 h} \sum_{i,j=1}^n K^*\left(\frac{X_i - X_j}{h}\right) + \frac{2}{nh} K(0) \quad (9)$$

où  $K^* := K^{(2)} - 2K$

$h \mapsto CV(h)$  est facilement calculable à partir des observations  $\{X_i\}_{i \in \{1 \dots n\}}$ , ce qui peut nous permettre de la minimiser en  $h$ , et de trouver  $h_{cv}$ .

## 3.2 Utilisation de la méthode

La densité de la loi normale  $p_{N(0,1)}$  et le noyau rectangulaire  $K_R$  vérifiant :

- $\int p_{N(0,1)}^2(x) dx < \infty$ ,
- $\int \int p_{N(0,1)}(x) K_R\left(\frac{x-y}{h}\right) p_{N(0,1)}(y) dx dy < \infty$ , et
- $K_R$  symétrique,

essayons de retrouver par la méthode de validation croisée le fait que  $h=1$  est une bonne fenêtre pour  $\hat{p}_n$  : les figures 2 et 3 sont la représentation de CV en fonction de  $h$  avec  $n=100$  réalisations.

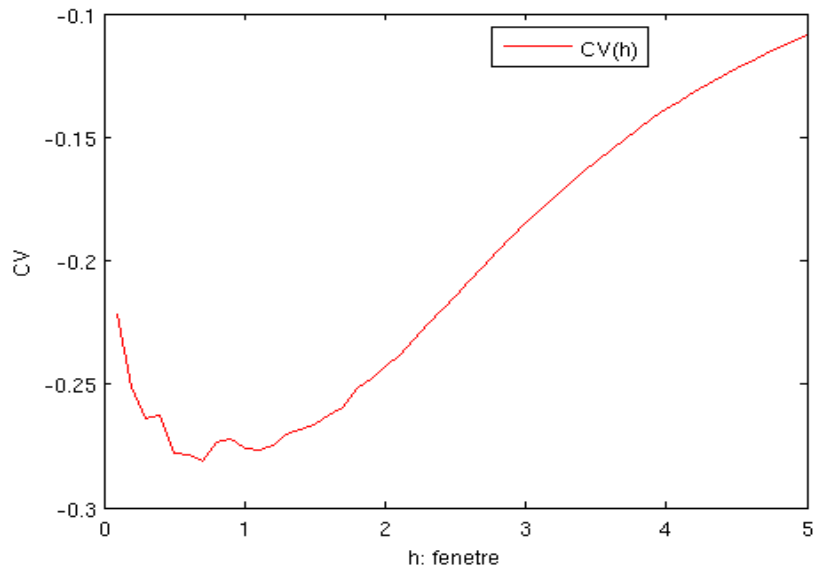


FIGURE 2 –  $CV(h)$  avec  $n=100$  réalisations. On utilise le même échantillon pour tous les calculs :  $h \mapsto CV(h)$  continue pour  $\{X_i\}_{i \in \{1 \dots n\}}$  fixées (MATLAB)

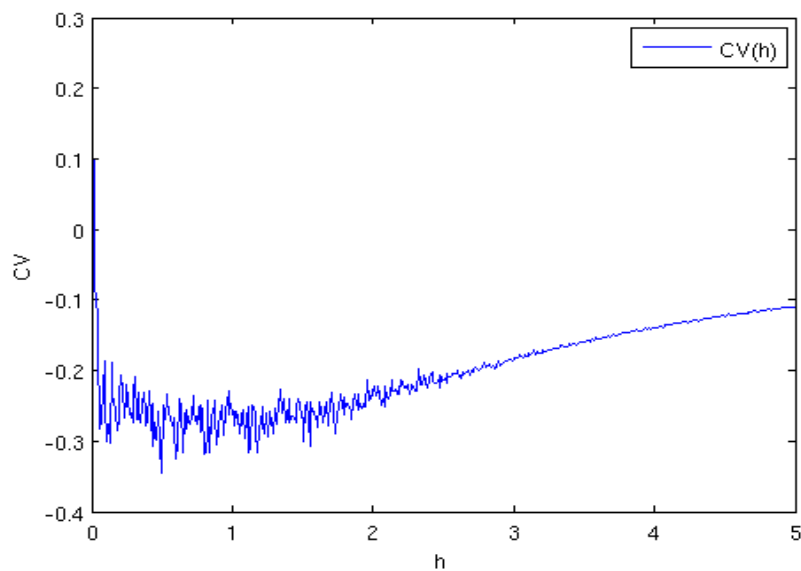


FIGURE 3 –  $CV(h)$  avec  $n=100$  réalisations, pour chaque  $h$  un nouvel  $n$ -échantillon est utilisé pour calculer  $CV(h)$  (MATLAB)

Pour  $h=1$ , la valeur de CV pour  $n=100$  réalisations fait bien partie des valeurs minimales.

Maintenant, qu'est-ce qui nous permet d'affirmer qu'un  $h$  minimisant  $CV(h)$  nous donnerait un  $h$  minimisant l'écart quadratique entre l'estimateur

$$\hat{p}_{(n,h)} = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{X_i - x}{h}\right) \text{ et la densité recherchée } \mathbf{p} ?$$

Mr Stone a répondu à cette question en 1984.

### 3.3 Hypothèses et résultat de Stone

#### 3.3.1 Hypothèses

Mr Stone a eu besoin de quelques hypothèses supplémentaires pour prouver l'efficacité de la méthode de validation croisée.

Il a rajouté aux précédentes conditions qui étaient :

- $\int p^2(x)dx < \infty$ ,
- $(x, y) \mapsto K\left(\frac{x-y}{h}\right)\mathbf{p}(x)\mathbf{p}(y)$  intégrable,
- $K$  symétrique,

les hypothèses suivantes :

- $K$  à support compact,
- $K$  Höldérienne,
- $0 < K^{(2)}(0) < 2K(0)$ ,
- $\mathbf{p}$  bornée,
- $\mathbf{p}$  possède une dérivée seconde uniformément continue, et de carré intégrable.

#### 3.3.2 Résultat de Stone

Définissons tout d'abord l'**optimalité asymptotique** d'une suite de fenêtres  $(h_n)_{n \in \mathbb{N}}$  (indexée par la taille de l'échantillon) :

on dit qu'une suite de fenêtres  $(h_n)_{n \in \mathbb{N}}$  est asymptotiquement optimale si, avec probabilité 1,

$$\frac{\int (\hat{p}_{(n,h_n)}(x) - p(x))^2 dx}{\min_{h>0} \left( \int (\hat{p}_{(n,h)}(x) - p(x))^2 dx \right)} = \frac{ISE(h_n)}{ISE(h_{ise,n})} \longrightarrow 1 \quad (10)$$

lorsque  $n \longrightarrow \infty$ .

**Remarque 1 :** Nous avons ici introduit l'ISE, qui n'est rien d'autre que l'erreur quadratique intégrée (Integrated Squared Error), c'est-à-dire le carré

de la distance  $L^2$  entre la densité recherchée  $\mathbf{p}$  et l'estimateur (les observations étant fixées).

**Remarque 2 :**  $h_{ise}$  désigne une valeur de  $h$  qui minimise  $ISE(h) = \int (\hat{p}_{(n,h)}(x) - p(x))^2 dx$ , les observations  $\{X_i\}_{i \in \{1 \dots n\}}$  étant fixées : nous montrerons que cette valeur existe sous les hypothèses de Stone.

Si  $(h_n)_{n \in N}$  est asymptotiquement optimale, on sait que presque sûrement  $\forall \varepsilon > 0, \exists n_0$  tel que  $\forall n \geq n_0, (1 - \varepsilon)ISE(h_{ise,n}) < ISE(h_n) < (1 + \varepsilon)ISE(h_{ise,n})$ , ie pour  $n$  assez grand, la réduction de l'erreur est aussi proche que l'on souhaite de la réduction optimale.

**Stone nous assure que, sous les hypothèses données ci-dessus et si l'on définit  $h_{cv,n}$  comme étant une fenêtre  $h$  minimisant  $h \mapsto CV_n(h)$  (pour  $n$  observations), la suite  $(h_{cv,n})_{n \in N}$  est une suite asymptotiquement optimale.**

### 3.3.3 Existence d'un $h > 0$ minimisant $h \mapsto ISE(h) : h_{ise}$

Montrons que sous les hypothèses de la section 3.3.1, il existe bien une fenêtre qui réduit au maximum la différence  $L^2$  entre l'estimateur à noyau et la densité  $\mathbf{p}$  recherchée.

a) On peut remarquer que  $\lim_{h \rightarrow 0} (ISE(h) - \int \mathbf{p}^2(x) dx) \geq 0$ .

En effet,

$$\begin{aligned} ISE(h) - \int \mathbf{p}^2(x) dx &= \int (\hat{p}_{(n,h)}(x) - \mathbf{p}(x))^2 dx - \int \mathbf{p}^2(x) dx \\ &= \int \hat{p}_{(n,h)}^2(x) dx - 2 \int \hat{p}_{(n,h)}(x) \mathbf{p}(x) dx \\ &= \int \left( \frac{1}{nh} \sum_{i=1}^n K\left(\frac{X_i - x}{h}\right) \right)^2 dx - \frac{2}{nh} \int \mathbf{p}(x) \sum_{i=1}^n K\left(\frac{X_i - x}{h}\right) dx \end{aligned} \quad (11)$$

Comme  $K$  est à support compact, pour  $h$  assez proche de 0,

$$\left( \frac{1}{nh} \sum_{i=1}^n K\left(\frac{X_i - x}{h}\right) \right)^2 - \frac{2\mathbf{p}(x)}{nh} \sum_{i=1}^n K\left(\frac{X_i - x}{h}\right) = \left( \frac{1}{nh} \sum_{\{i|X_i=x\}} K(0) \right)^2 - \frac{2}{nh} \mathbf{p}(x) \sum_{\{i|X_i=x\}} K(0)$$

cette somme est donc égale à zéro partout sauf sur chaque observation donc nulle presque partout. L'intégrale est donc nulle.

b) On observe aussi que lorsque  $h \longrightarrow +\infty$ ,  $\int \widehat{p}_{(n,h)}^2(x)dx \sim \frac{1}{h}K^{(2)}(0)$ .

$$\begin{aligned} \text{En effet, } \int \widehat{p}_{(n,h)}^2(x)dx &= \int \left(\frac{1}{nh} \sum_{i=1}^n K\left(\frac{X_i-x}{h}\right)\right)^2 dx \\ &= \frac{1}{n^2 h^2} \sum_{i,j=1}^n \int K\left(\frac{X_i-x}{h}\right) K\left(\frac{X_j-x}{h}\right) dx = \frac{1}{n^2 h} \sum_{i,j=1}^n K^{(2)}\left(\frac{X_j-X_i}{h}\right) \end{aligned} \quad (12)$$

par le même changement de variable que lors de la construction de CV (troisième étape, section 3.1), et en utilisant de la même manière que  $K$  est symétrique.

Quels que soient  $i$  et  $j \in \{1 \dots n\}$ ,  $K^{(2)}\left(\frac{X_j-X_i}{h}\right) \longrightarrow K^{(2)}(0)$  lorsque  $h \longrightarrow +\infty$  puisque  $K^{(2)}$  est continue.

**Remarque :** Le fait que  $K^{(2)}$  soit continue n'apparaît pas dans les hypothèses de Stone, mais cela en découle ;

$K$  Höldérienne  $\Rightarrow K$  continue

$K$  continue et à support compact  $\Rightarrow K$  bornée

$\Rightarrow K^{(2)}$  continue en utilisant le fait que  $K$  soit bornée ainsi que la continuité de la translation dans  $L^1$ .

Ainsi on obtient  $h \int \widehat{p}_{(n,h)}^2(x)dx \longrightarrow \frac{1}{n^2} n^2 K^{(2)}(0)$  lorsque  $h \longrightarrow +\infty$ , d'où le résultat.

c) Enfin, lorsque  $h \longrightarrow +\infty$ ,  $\int \widehat{p}_{(n,h)}(x)\mathbf{p}(x)dx \sim \frac{1}{h}K(0)$ .

$$\text{En effet, } \int \widehat{p}_{(n,h)}(x)\mathbf{p}(x)dx = \frac{1}{nh} \sum_{i=1}^n \int \mathbf{p}(x) K\left(\frac{X_i-x}{h}\right) dx$$

et pour tout  $x$ ,  $\mathbf{p}(x) K\left(\frac{X_i-x}{h}\right) \longrightarrow \mathbf{p}(x) K(0)$  lorsque  $h \longrightarrow +\infty$ .

Cette convergence simple implique, en appliquant le théorème de convergence dominée de Lebesgue ( $K$  est borné donc majoration par  $\mathbf{p}(x)\|K\|_\infty$  qui est intégrable) :

$$\frac{1}{n} \sum_{i=1}^n \int \mathbf{p}(x) K\left(\frac{X_i-x}{h}\right) dx \longrightarrow \frac{1}{n} \sum_{i=1}^n \int \mathbf{p}(x) K(0) dx = \frac{1}{n} K(0) \int \mathbf{p}(x) dx = K(0)$$



lorsque  $h \longrightarrow +\infty$ , d'où le résultat.

Ainsi avec b) et c),  $ISE(h) - \int \mathbf{p}^2(x)dx \sim_{h\infty} \frac{1}{h}(K^{(2)}(0) - 2K(0)) < 0$   
par la troisième hypothèse rajoutée par Stone ( $0 < K^{(2)}(0) < 2K(0)$ ).

Pour les observations  $\{X_i\}_{i \in \{1 \dots n\}}$  fixées,  $h \longmapsto ISE(h) - \int \mathbf{p}^2(x)dx$  est continue sur  $R_+^*$ .

En effet,  $ISE(h) - \int \mathbf{p}^2(x)dx =$

$$\frac{1}{n^2 h} \sum_{i,j=1}^n K^{(2)}\left(\frac{X_j - X_i}{h}\right) \quad (13)$$

$$- \frac{1}{h} \int \frac{2\mathbf{p}(x)}{n} \sum_{i=1}^n K\left(\frac{X_i - x}{h}\right) dx \quad (14)$$

par (11) et (12).

Pour (13), la continuité de  $K^{(2)}$  a déjà été remarquée (remarque précédente), donc l'expression est continue en  $h$  sur  $R_+^*$ .

Pour (14),  $K$  étant continue, l'intégrant est continu en  $h$  sur  $R_+^*$ . Il suffit alors d'utiliser le théorème de continuité des intégrales à paramètre de Lebesgue avec pour fonction intégrable majorante  $2\mathbf{p}(x)\|K\|_\infty$ .

La fonction  $h \longmapsto ISE(h) - \int \mathbf{p}^2(x)dx :$

- est continue
- tend vers 0 lorsque  $h$  tend vers 0
- tend vers 0 par des valeurs strictement négatives lorsque  $h$  tend vers l'infini

Par continuité, elle admet au moins un  $h > 0$  qui la minimise, et ce  $h$  minimise l'ISE.

Le résultat de Stone nous a donné une indication asymptotique à propos de la fenêtre idéale. Nous allons voir qu'en pratique, l'utilisation de la validation croisée peut avoir des limites.

### 3.4 Problème lié à la discrétisation des observations

En pratique, les observations sont souvent discrétisées. Les histogrammes sont en effet d'utilisation pratique, mais n'oublions pas non plus qu'un ordinateur ne manipule jamais des réels mais des approximations rationnelles plus ou moins fines.

Soient

- $n$  le nombre d'observations,
- $k$  le nombre de valeurs dont nous nous servons pour décrire les observations (l'abscisse de l'histogramme),
- $\{l_i\}_{i \in \{1 \dots k\}}$  les valeurs de l'histogramme  
ie  $l_i = \text{card}(\{j \in \{1 \dots n\} \mid X_j = i^{\text{eme}} \text{ valeur}\})$ ,
- $m$  le nombre de couples  $(i, j)$  tels que  $i < j$  et  $X_i = X_j$ .

On obtient facilement que  $m = \frac{1}{2} \sum_{i=1}^k l_i(l_i - 1)$

d'où  $m = \frac{1}{2} \sum_{i=1}^k l_i^2 - \frac{n}{2}$  puisque  $\sum_{i=1}^k l_i = n$ .

Enfin,  $\frac{1}{k} \sum_{i=1}^k l_i^2 \geq (\frac{1}{k} \sum_{i=1}^k l_i)^2 = \frac{n^2}{k^2}$  par l'inégalité de Jensen.

En effet, considérons  $L$  la variable aléatoire qui vaut  $l_i$  avec probabilité  $\frac{n_i}{k}$ , où  $n_i = \text{card}(\{j \in \{1, \dots, k\} \mid l_j = l_i\})$ .

Alors en regroupant les  $l_i$  égaux et en notant  $k' (\leq k)$  le nombre de valeurs distinctes,

$$\frac{1}{k} \sum_{i=1}^k l_i^2 = \sum_{j=1}^{k'} l_j^2 \frac{n_j}{k} = E[L^2] \geq (E[L])^2 = (\sum_{j=1}^{k'} l_j \frac{n_j}{k})^2 = (\frac{1}{k} \sum_{i=1}^k l_i)^2$$

On voit alors mieux où l'inégalité de Jensen agit.

On a alors  $m = \frac{k}{2} \frac{1}{k} \sum_{i=1}^k l_i^2 - \frac{n}{2} \geq \frac{k}{2} \frac{n^2}{k^2} - \frac{n}{2}$  et donc  $\frac{m}{n} \geq \frac{1}{2} \frac{n}{k} - \frac{1}{2}$ .

Nous venons de trouver une minoration naturelle de  $\frac{m}{n}$ .

Cependant, lorsque la minoration de  $\frac{m}{n}$  est trop forte, la méthode de validation croisée trouve ses limites. Plus précisément, sous les hypothèses de la section 3.3.1, quand  $\frac{m}{n} > \alpha := \frac{K^{(2)}(0)}{4K(0) - 2K^{(2)}(0)}$ ,  $CV(h) \rightarrow -\infty$

lorsque  $h \longrightarrow 0$  et alors, la fenêtre proposée par cette méthode est la valeur dégénérée  $h=0$ .

En effet, on a :

$$hCV(h) = \frac{1}{n^2} \sum_{i,j=1}^n K^*\left(\frac{X_i - X_j}{h}\right) + \frac{2}{n}K(0)$$

et il y a  $2m+n$  termes de la somme qui donnent  $K^*(0)$  (par définition de  $m$ ), donc

$$hCV(h) = \frac{(2m+n)}{n^2}K^*(0) + \sum_{i,j|X_i \neq X_j} K^*\left(\frac{X_i - X_j}{h}\right) + \frac{2}{n}K(0)$$

et comme  $K$  est à support compact,  $K^* = K^{(2)}(0) - 2K(0)$  l'est aussi, et donc  $\lim_{h \rightarrow 0}(hCV(h)) =$

$$\begin{aligned} \frac{(2m+n)}{n^2}K^*(0) + \frac{2}{n}K(0) &= \frac{(2m+n)}{n^2}(K^{(2)}(0) - 2K(0)) + \frac{2}{n}K(0) \\ &= \frac{2m}{n^2}(K^{(2)}(0) - 2K(0)) + \frac{1}{n}K^{(2)}(0) = -\frac{m}{n^2}\alpha^{-1}K^{(2)}(0) + \frac{1}{n}K^{(2)}(0) \\ &= \frac{(-mn^{-1}\alpha^{-1} + 1)}{n}K^{(2)}(0) < 0 \end{aligned}$$

car  $K^2(0) > 0$  et comme  $\frac{m}{n} > \alpha$ ,  $mn^{-1}\alpha^{-1} > 1$

$CV$  tend donc bien vers  $-\infty$  lorsque  $h$  tend vers 0, soit le résultant embêtant.

### Illustration du problème :

Considérons le noyau d'Epanechnikov

$$K_E(x) = \frac{3}{4}(1 - x^2)1_{|x| \leq 1} \quad (15)$$

On peut calculer (et c'est déjà moins compliqué que pour le noyau triangulaire) la convolution de  $K_E$  avec lui-même :

$$K_E^{(2)}(x) = \frac{9}{16}\left(\frac{16}{15} - \frac{4}{3}x^2 + \frac{2}{3}|x|^3 - \frac{1}{30}|x|^5\right)1_{0 \leq |x| \leq 2}$$

La minoration toujours vraie de  $\frac{m}{n}$  étant  $\frac{m}{n} \geq \frac{1}{2} \frac{n}{k} - \frac{1}{2}$ , trouvons  $n$  et  $k$  tels que  $\frac{1}{2} \frac{n}{k} - \frac{1}{2} > \alpha := \frac{K_E^{(2)}(0)}{4K_E(0) - 2K_E^{(2)}(0)} = \frac{1}{3}$ .

On peut donc choisir  $n=100$  et  $k=25$  :  $\frac{1}{2} \frac{100}{25} - \frac{1}{2} = \frac{1}{2} 4 - \frac{1}{2} = 1,5 > \frac{1}{3}$   
 La figure 4 montre que  $h \mapsto CV_E(h)$  tend vers  $-\infty$  lorsque  $h \rightarrow 0$  pour  $n=100$  observations (issues d'une loi de Cauchy) discrétisées en 25 valeurs.  
 On peut la comparer à la figure 5 qui représente  $h \mapsto CV_E(h)$  lorsque les  $n=100$  observations ne sont pas discrétisées.

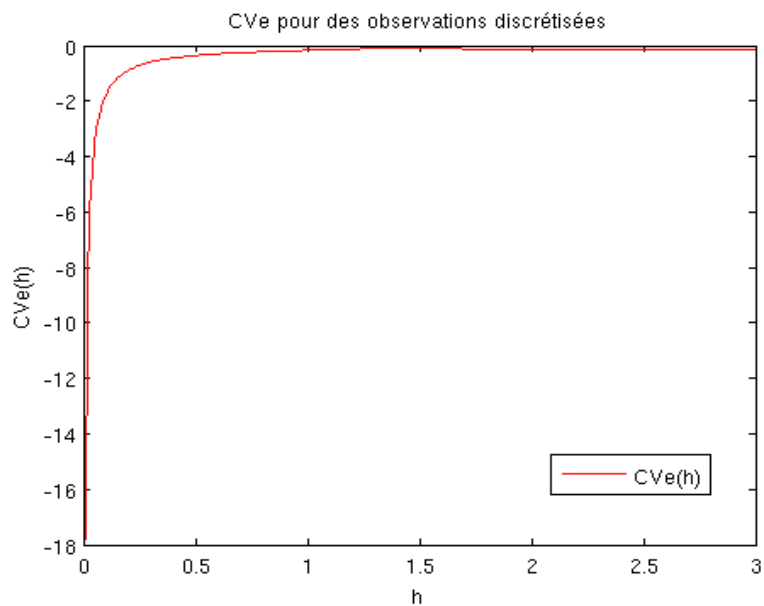


FIGURE 4 –  $CV_E(h)$  avec  $n=100$  observations (issues d'une loi de Cauchy) discrétisées en 25 valeurs (MATLAB)

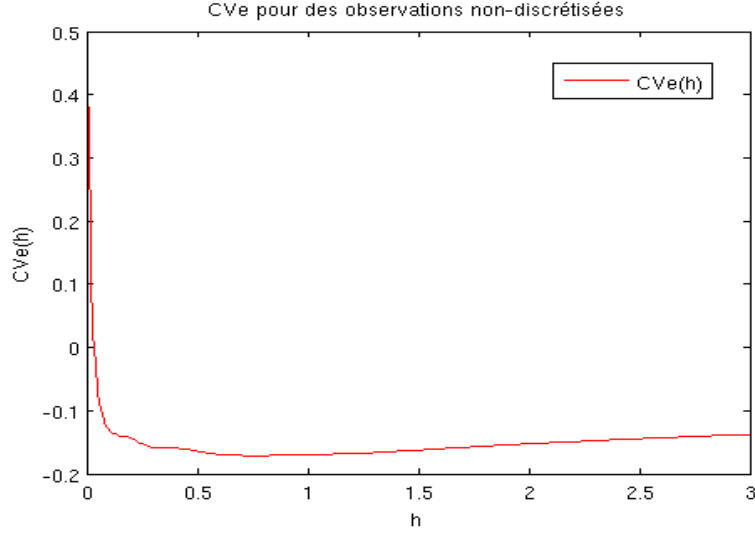


FIGURE 5 –  $CV_E(h)$  avec  $n=100$  observations (issues d'une loi de Cauchy) non discrétisées (MATLAB)

## 4 Validation croisée type maximum de vraisemblance

En étude paramétrique, lorsque l'on cherche l'estimateur du maximum de vraisemblance d'un paramètre  $\theta$  de la loi étudiée, on cherche le paramètre qui maximise la vraisemblance, c'est-à-dire une fonction  $\hat{\theta}(Y)$  des observations  $Y$  qui maximise la vraisemblance  $\theta \rightarrow p_\theta(Y)$ .

### 4.1 Présentation

Imaginons que l'on possède une  $(n+1)^{ieme}$  observation  $Y$  de densité  $\mathbf{p}$ . Regardons  $\{\hat{p}_{(n,h)}\}_h$  comme une famille de densités de paramètre  $h$  parmi lesquelles se trouverait  $\mathbf{p}$ .

$\log(\hat{p}_{(n,h)}(Y))$  est alors la log-vraisemblance. Comme nous ne disposons que de  $n$  observations, considérons  $\hat{p}_{n,-i}(x) := \frac{1}{(n-1)h} \sum_{j \neq i} K(\frac{X_j - x}{h})$  l'estimateur de la densité sans  $X_i$ , et prenons  $Y = X_i$ .

On obtient la log-vraisemblance  $\log(\hat{p}_{n,-i}(X_i))$ . Le choix de  $i$  étant arbitraire, nous considérons plutôt

$$LCV(h) := \frac{1}{n} \sum_{i=1}^n \log(\hat{p}_{n,-i}(X_i)) \quad (16)$$

(LCV pour Likelihood Cross Validation).

Le choix de  $h$  par validation croisée type maximum de vraisemblance sera donc une valeur de  $h$  qui maximise  $LCV(h)$ .

## 4.2 Une limite de cette méthode

La fonction  $h \rightarrow LCV(h)$  est très sensible aux valeurs abérantes lorsque l'on utilise un noyau à support compact.

En effet, supposons que  $f$  ait un support non borné et que  $K$  ait un support  $=[-1,1]$ . Si une des observations, disons  $X_1$ , est distante d'au moins  $R > 0$  de toutes les autres, alors si  $h < R$ ,  $|\frac{X_i - X_1}{h}| > 1$ . De ce fait,

$$\hat{p}_{n,-1}(X_1) = \frac{1}{(n-1)h} \sum_{i \neq 1} K\left(\frac{X_i - X_1}{h}\right) = 0$$

Ainsi  $LCV(h)$  serait  $-\infty$  pour tout  $h < R$ . Cette méthode conduirait donc à choisir systématiquement  $h \geq R$ , et pourrait nous mener à du sous-lissage (dans le cas, on le rappelle, d'un noyau à support compact).

## 4.3 Première utilisation

Testons cette méthode avec le noyau gaussien

$$K_G(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \quad (17)$$

dont le support est  $\mathbb{R}$  tout entier, et un  $n$ -échantillon de loi exponentielle de paramètre 1.

On observe que le maximum de  $LCV$  est atteint pour  $h$  entre 0.2 et 0.7 (figures 6 et 7).

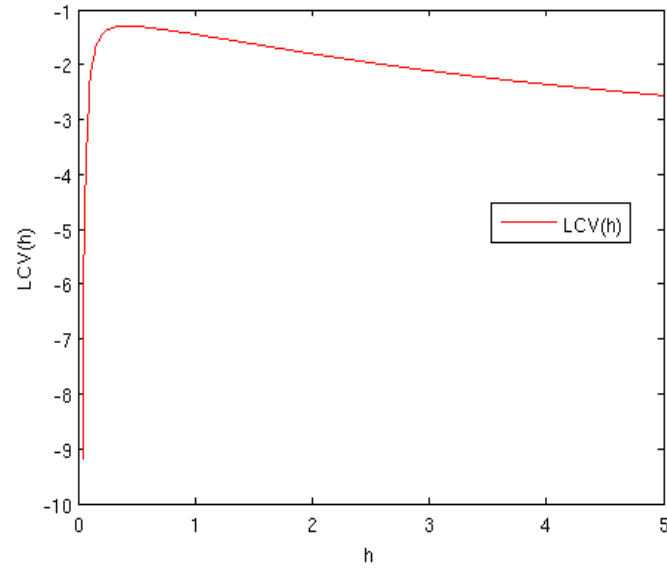


FIGURE 6 –  $LCV(h)$  avec  $n=100$  réalisations de  $va$  de loi exponentielle. On utilise le même échantillon pour tous les calculs :  $h \mapsto LCV(h)$  continue pour  $\{X_i\}_{i \in \{1 \dots n\}}$  fixées (MATLAB)

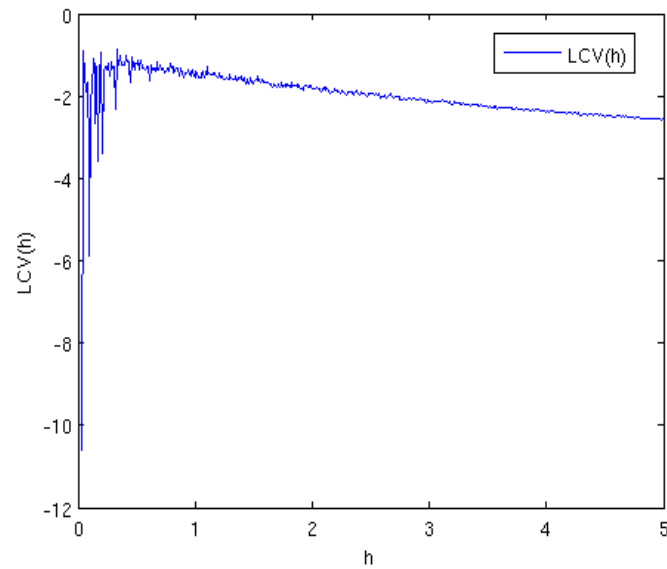


FIGURE 7 –  $CV(h)$  avec  $n=100$  réalisations, pour chaque  $h$  un nouvel  $n$ -échantillon est utilisé pour calculer  $LCV(h)$  (MATLAB)

$h=0.4$  semble être la meilleure fenêtre si l'on regarde les réalisations de l'estimateur (figure 8).

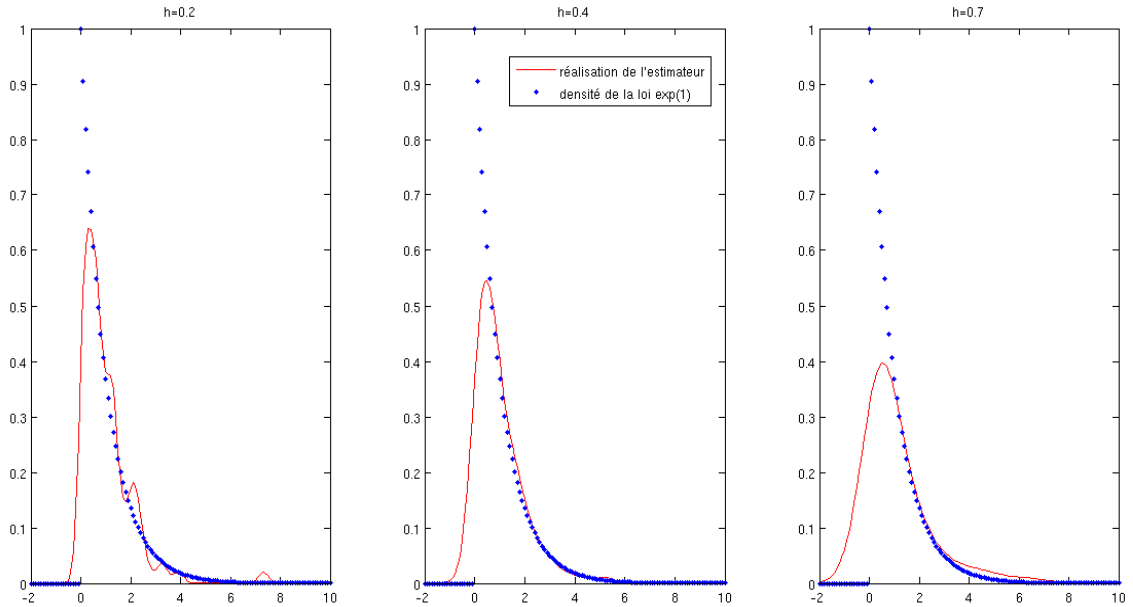


FIGURE 8 – trois réalisations de l'estimateur à noyau construit avec le noyau gaussien, avec  $n=100$  observations et  $h=0.2, h=0.4, h=0.7$  comme fenêtres (MATLAB)

## 5 Utilisation de ces outils

Cette dernière partie est consacrée à la mise en pratique des outils présentés dans les précédentes sections. Nous allons tout d'abord travailler sur des observations dont nous connaissons la loi de probabilité.

### 5.1 Mise en pratique sur une densité connue

Considérons la densité bimodale

$$p(x) = \frac{1}{2\sqrt{8\pi}}e^{-\frac{x^2}{8}} + \frac{1}{2\sqrt{2\pi}}e^{-\frac{(x-4)^2}{2}}$$

représentée figure 9.



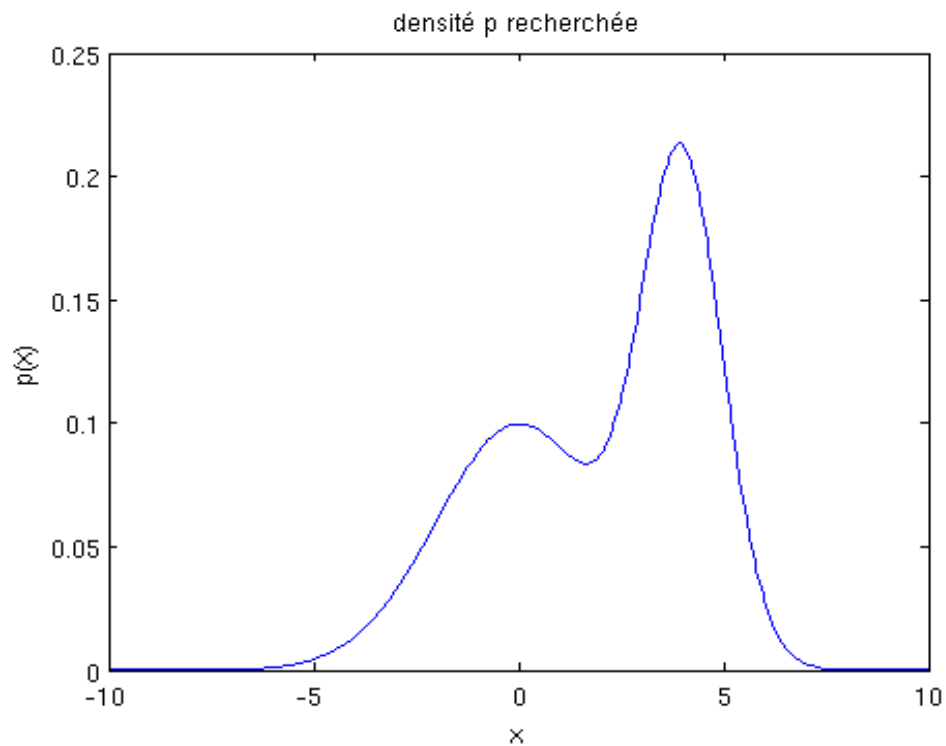


FIGURE 9 – densité bimodale que nous cherchons à approcher (MATLAB)

Nous pouvons simuler, par la méthode d’acceptation-rejet, 100 réalisations de variables aléatoires possédant cette densité.

### 5.1.1 Cross validation

Utilisons la méthode de validation croisée avec le noyau d’Epanechnikov (15). Regardons la courbe  $h \longrightarrow CV(h)$  (figure 10) :

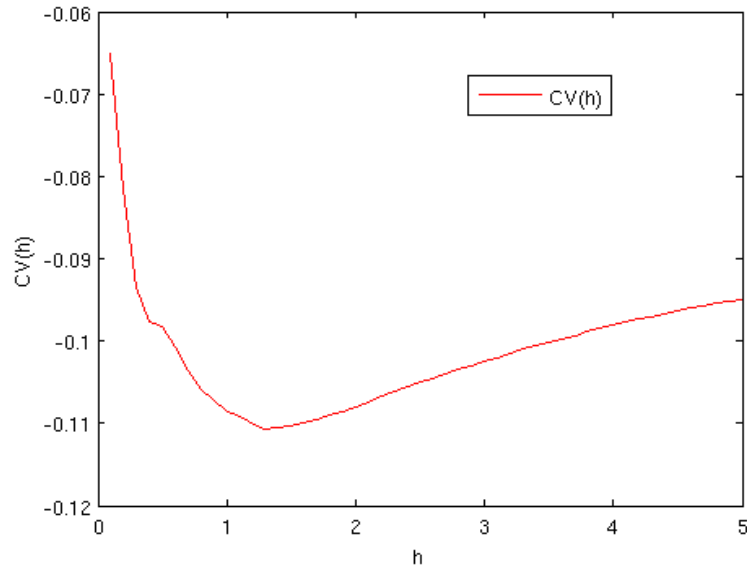


FIGURE 10 –  $h \longrightarrow CV(h)$  pour les 100 observations étudiées (MATLAB)

La validation croisée nous encourage à choisir une fenêtre  $h \in [1, 2]$ . Vérifions cela en calculant  $h \longrightarrow ISE(h)$  sur  $[0.5, 3]$  puisque nous connaissons  $\mathbf{p}$ . C'est ce que représente la figure 11.

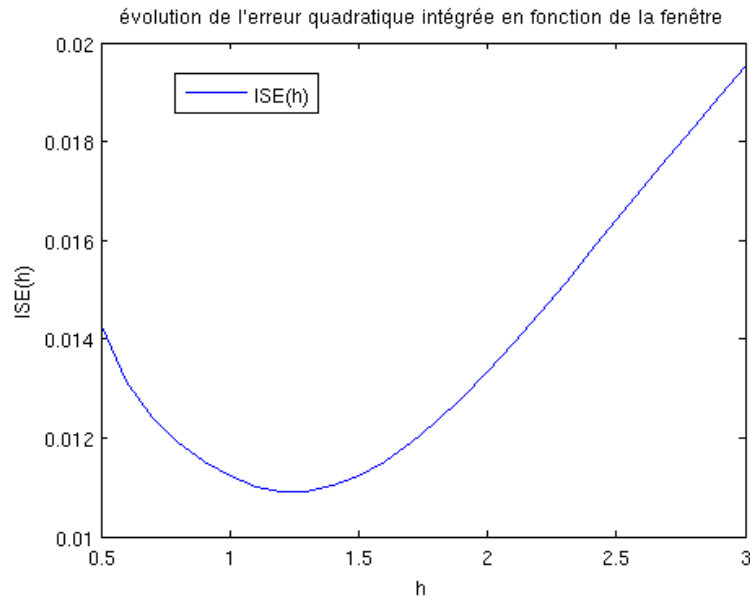


FIGURE 11 –  $h \longrightarrow ISE(h)$  calculée par la méthode des trapèzes (MATLAB)

$h=1.25$  est donc la fenêtre qui minimise la distance  $L^2$  entre l'estimateur à noyau et la densité  $\mathbf{p}$ . La figure 12 représente  $\mathbf{p}$  (en pointillé) et l'estimateur à noyau construit avec le noyau d'Epanechnikov (en trait plein).

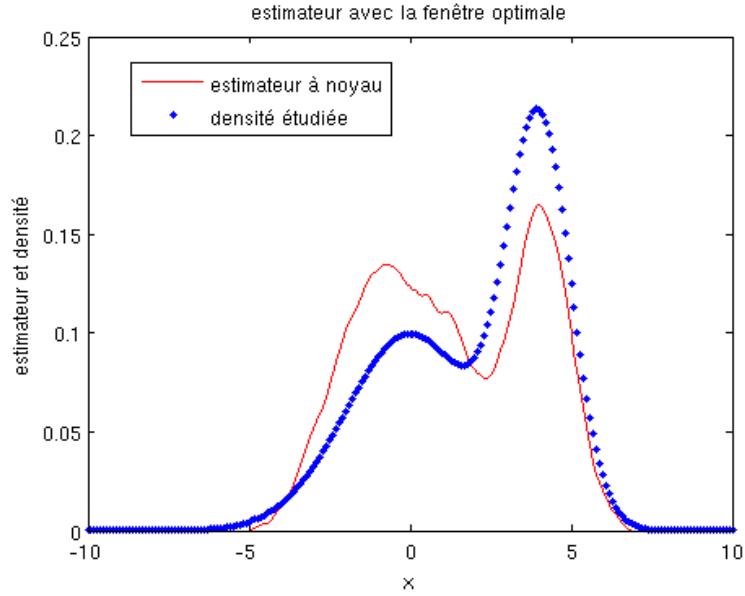


FIGURE 12 – estimateur à noyau de la densité avec la fenêtre optimale et le noyau d'Epanechnikov (MATLAB)

### 5.1.2 Likelihood cross validation

Utilisons maintenant la méthode de validation croisée type maximum de vraisemblance avec le noyau gaussien (17). Regardons la courbe  $h \rightarrow LCV(h)$  (figure 13) :

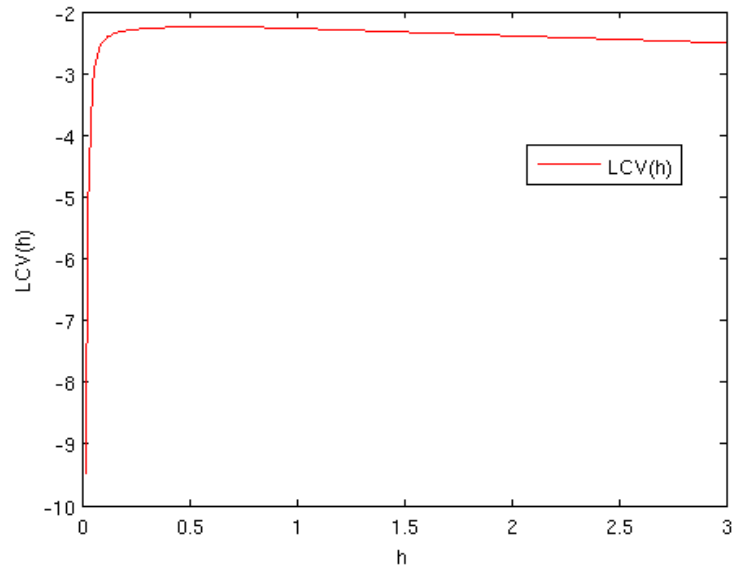


FIGURE 13 –  $h \longrightarrow LCV(h)$  pour les 100 observations étudiées (MATLAB)

Ici, la méthode nous propose  $h \in [0.1, 1.5]$ . Encore une fois, regardons ce qu'il en est pour la distance  $L^2$ , et traçons l'ISE en fonction de la fenêtre (figure 14).

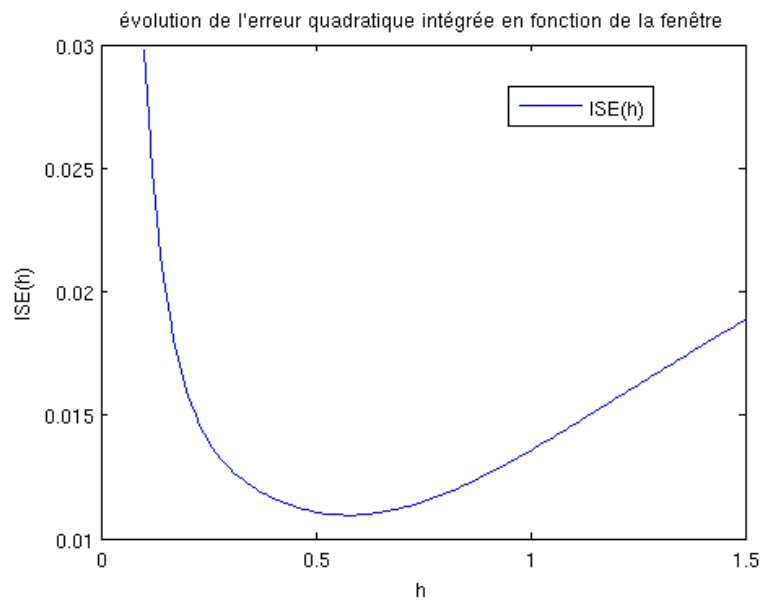


FIGURE 14 –  $h \longrightarrow ISE(h)$  calculée par la méthode des trapèzes (MATLAB)

$h=0.58$  est la fenêtre qui réduit l'ISE au maximum. Comme précédemment, regardons ce que cela donne avec ce choix de fenêtre : la figure 15 représente  $\mathbf{p}$  (en pointillé) et l'estimateur à noyau construit avec le noyau gaussien (en trait plein).

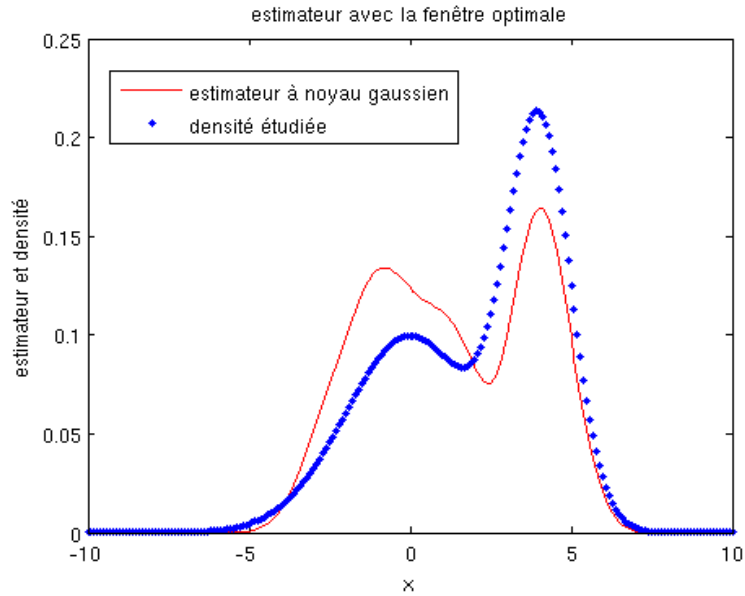


FIGURE 15 – estimateur à noyau de la densité avec la fenêtre optimale et un noyau gaussien (MATLAB)

Les calculs de l'ISE nous ont permis de voir que la recherche de fenêtre par ces deux méthodes pouvait porter ses fruits puisqu'elles nous ont donné des choix qui ont été confirmés par l'étude "théorique".

Ici, les défauts des courbes sont donc à mettre sur le dos du choix du noyau ou de la forme initiale de l'estimateur (ie le fait d'utiliser des estimateurs à noyau).

Lançons nous donc confiants dans l'étude de données inconnues.

## 5.2 Application à des observations dont la loi est inconnue

Nous allons ici considérer les vitesses en  $\frac{km}{s}$  de 82 galaxies. Nous faisons alors l'hypothèse que la variable aléatoire donnant la vitesse d'une galaxie prise au hasard suit une loi à densité dominée par la mesure de Lebesgue.

Comme dans la section 5.1, l'étude par validation croisée se fera avec le noyau d'Epanechnikov (15) car ce noyau possède un support compact, et l'étude par LCV avec le noyau Gaussien (17) car son support est  $\mathbb{R}$  (cela éliminera les problèmes liés valeurs abérantes). Les figures 16 et 17 représentent respectivement  $h \rightarrow CV(h)$  et  $h \rightarrow LCV(h)$  pour ces deux noyaux.

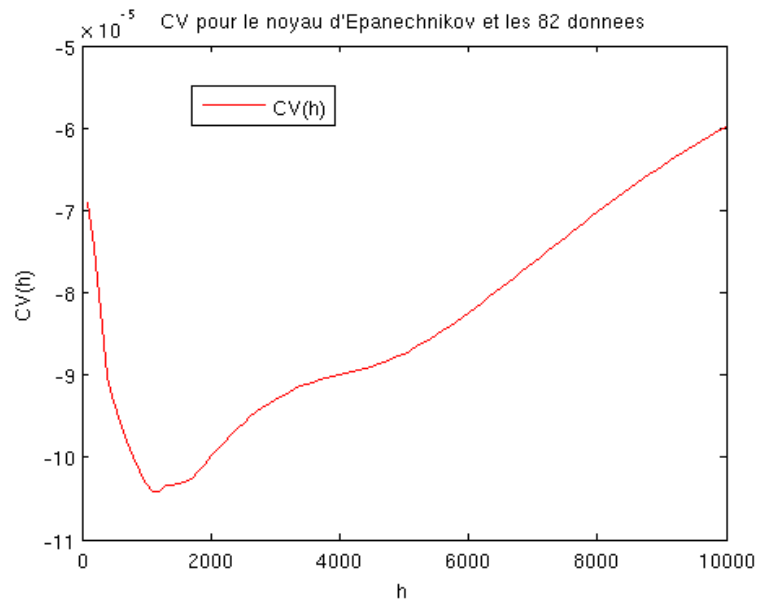


FIGURE 16 –  $h \rightarrow CV(h)$  pour les 82 données étudiées : le minimum est atteint en  $h=1154$  (MATLAB)

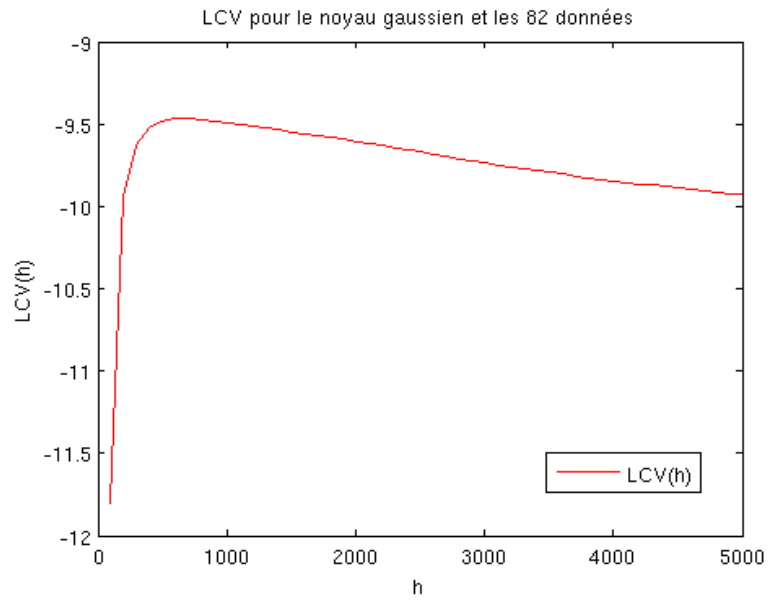


FIGURE 17 –  $h \longrightarrow LCV(h)$  pour les 82 données étudiées : le maximum est atteint en  $h=645$  (MATLAB)

A ce stade, les deux méthodes nous proposent respectivement  $h = 1154$  et  $h = 645$ . Nous terminons ce mémoire avec la représentation des deux estimateurs à noyau en utilisant les fenêtres proposées (figures 18 et 19).

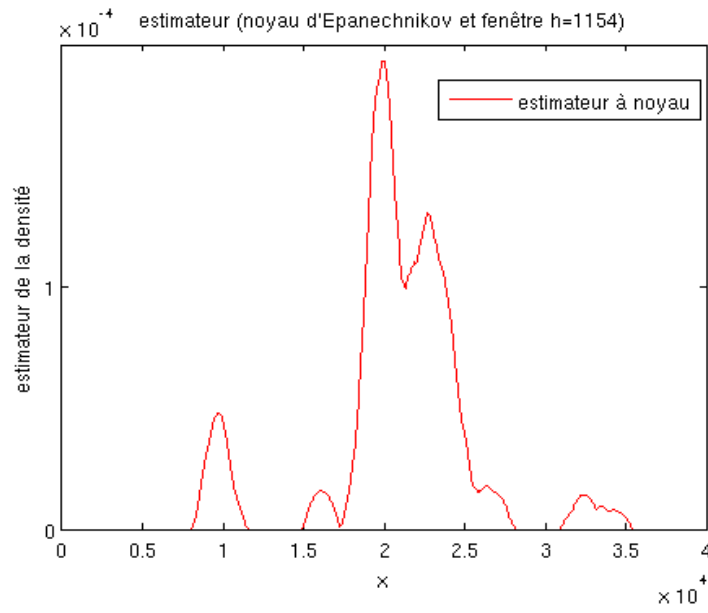


FIGURE 18 –  $x \longrightarrow \hat{p}_{n,h,K_E}(x)$   $n=82$ ,  $h=1154$  (MATLAB)

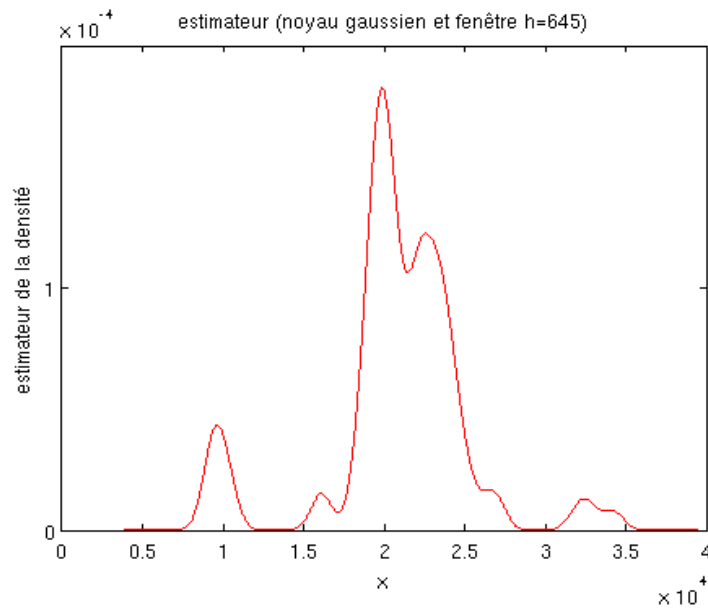


FIGURE 19 –  $x \longrightarrow \hat{p}_{n,h,K_G}(x)$   $n=82$ ,  $h=645$  (MATLAB)



**BIBLIOGRAPHIE :**

- Introduction à l'estimation non paramétrique, Alexandre B. Tsybakov
- Density Estimation for Statistics and Data Analysis, B.W. Silverman
- An asymptotically optimal window selection rule for kernel density estimates, Charles J. Stone

**2.2. Informatique Théorique : les Graphes Expander.** Dans le cadre du cours d'Informatique Théorique de première année de master, plusieurs travaux de recherche étaient proposés. J'ai choisi de travailler sur des graphes appelés **graphes expanders**.

Le rapport qui suit contient la présentation de ces graphes, ainsi qu'une méthode de construction. Comme une grande partie du cours était réservée à la présentation d'algorithmes probabilistes, ce travail se termine par l'utilisation des graphes expanders pour « dérandomiser » de tels algorithmes sans trop augmenter leur complexité.

Professeur référente : Sophie Laplante  
Université Paris-Sud XI  
Informatique théorique (projet Master 1)

# Les graphes EXPANDERS

---

Nicolas Drougard

Orsay, juin 2011

# Table des matières

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Contexte</b>   | <b>2</b>  |
| <b>2</b> | <b>Définitions d'un graphe expander</b>   | <b>2</b>  |
| 2.1      | Définition combinatoire . . . . .   | 3         |
| 2.2      | Présentation de $\lambda(G)=SLEV$ : . . . . .   | 3         |
| 2.3      | Définition algébrique du graphe expander . . . . .  | 5         |
| <b>3</b> | <b>Relation entre les deux définitions</b>  | <b>5</b>  |
| 3.1      | Définition algébrique $\Rightarrow$ définition combinatoire . . . . .   | 5         |
| 3.2      | Définition combinatoire + $\forall k \in \{1 \dots n\}, (k, k) \in E \Rightarrow$ définition algébrique . . . . . | 6         |
| <b>4</b> | <b>Construction d'une famille de graphes expanders</b>  | <b>9</b>  |
| 4.1      | Quelques définitions utiles . . . . .   | 9         |
| 4.2      | Construction de la famille . . . . .  | 12        |
| <b>5</b> | <b>Réduction de l'erreur de RP par les graphes expanders</b>  | <b>14</b> |

## 1 Contexte

Nous considérons un graphe non-dirigé  $G=(V,E)$  :

- $V=\{1 \dots n\}$  l'ensemble de ses sommets,
- l'ensemble des arêtes  $E$  est un multi-ensemble formé des éléments de  $\{(i, j) \in V^2 | i \leq j\}$  (ie les arêtes liant un seul sommet sont autorisées, ainsi que les arêtes parallèles).

Nous prendrons ce graphe d-régulier,

ie  $\forall v \in V, card(\{e \in E | e \text{ contient } v\}) = d$ .

## 2 Définitions d'un graphe expander

Remarque préliminaire :

Considérons un sous-graphe d'un graphe de taille  $n$ , d-régulier (sous graphe signifie  $(V', E')$  tel que  $V' \subset V$  et  $E' \subset E$  avec  $\forall e \in E', e(1)$  et  $e(2) \in V'$ ).

$|E(V', \overline{V'})|$  désigne le nombre d'arêtes liant un sommet de  $V'$  à un sommet du complémentaire de  $V'$  dans  $V$ .

Comme le nombre d'arêtes liés à un sommet  $v \in V'$  est égal à  $d$ ,

$|E(V', \overline{V'})| \leq d \cdot |V'|$ .

La définition combinatoire du graphe expander minore ce  $|E(V', \overline{V'})|$  par  $\rho d \cdot |V'|$  avec  $\rho > 0$  (et par la remarque,  $\rho \leq 1$ ).

## 2.1 Définition combinatoire

On dit que  $G$  est un  $(n, d, \rho)$ -**graphe expander**, avec  $\rho > 0$ , un graphe de  $n$  sommets,  $d$ -régulier tel que  $\forall$  sous-graphe  $S \subset V$  avec  $|S| \leq \frac{n}{2}$ ,

$$|E(S, \overline{S})| \geq \rho d \cdot |S|.$$

Ainsi  $|E(S, \overline{S})| = \Omega(|S|)$  (par la remarque).

La notion d'expansion se comprend bien avec cette définition : lorsque  $\rho \geq \frac{1}{d}$ , il y a plus d'arêtes liant  $S$  et  $\overline{S}$  que d'éléments de  $S$ .

Le cycle de taille  $n$  est 2-régulier, et pour tout sous-graphe  $S$  de taille  $\leq \frac{n}{2}$ ,  $|E(S, \overline{S})| \geq 2$ .

Donc le cycle est un  $(n, 2, \rho)$ -graphe expander avec  $\rho = \frac{2}{n}$ . En effet,  $\frac{2}{n} d |S| \leq \frac{2}{n} d \frac{n}{2} = d = 2 \leq |E(S, \overline{S})|$ .

Cependant  $\rho$  décroît avec  $n$  vers 0, et nous allons avoir besoin de familles de graphes expanders tels que  $\rho$  soit minoré par une constante (indépendante de la taille du graphe). Ces familles sont définies dans ce qui suit. La construction de telles familles se révélera plus compliqué...

Une **famille de graphes expanders**  $\{G_n\}_{n \in \mathbb{N}}$  est une suite de  $(n, d, \rho)$ -graphes expanders,  $\rho > 0$  fixé, indexée par  $n$ , le nombre de sommets.

## 2.2 Présentation de $\lambda(G)$ =SLEV :

Définissons  $\lambda(G)$  pour un graphe  $G$  de  $n$  sommets et  $d$ -régulier.

On note  $B$  la matrice d'adjacence de  $G$ , et  $A = \frac{1}{d} B$ . On appelle  $A$  la matrice de marche aléatoire sur  $G$ .

Ce nom vient du fait que  $A$  est une matrice stochastique :

considérons la marche aléatoire sur  $G$  qui consiste à passer d'un sommet  $v$  à un voisin  $w$  avec une probabilité proportionnelle aux nombre d'arêtes entre  $v$  et  $w$ .

Choisissons une loi de probabilité  $p_0$  (vecteur colonne de taille  $n$ ) sur l'ensemble des sommets du graphe ( $p_0(i)$  pour  $i \in V$  est la probabilité de choisir  $i$  comme sommet initial),  $A p_0$  est la loi de probabilité sur les sommets de  $G$

après un pas de cette marche aléatoire, et  $A^k p$  est la probabilité sur  $V$  après  $k$  pas.

Considérons le vecteur colonne de taille  $n$ ,  $\mathbf{1} = (\frac{1}{n}, \dots, \frac{1}{n})$  (qui correspond à la distribution uniforme sur  $V$ ).

Définition de  $\lambda(G)$  :

" $\lambda(G)$  est la seconde plus grande valeur propre de  $A$ " (the Second Largest EigenValue)

$\Updownarrow$

" $\lambda(G) = \max_{u \in (\mathbf{1}^\perp) \cap (\|u\|_2=1)} (\|Au\|_2)$ , soit la valeur maximale atteinte par  $(\|Au\|_2)$  lorsque le vecteur  $u$  est orthogonal à  $\mathbf{1}$  de norme euclidienne égale à un"

Preuve de l'équivalence :

Tout comme la matrice d'adjacence d'un graphe non-orienté,  $A$  est symétrique. On peut donc trouver une base orthonormée de vecteurs propres  $(u_1, \dots, u_n)$ , de valeurs propres réelles correspondantes  $(\lambda_1, \dots, \lambda_n)$  que l'on peut classer pour assurer que  $|\lambda_n| \leq \dots \leq |\lambda_1|$ .

Notons que  $A\mathbf{1} = \mathbf{1}$  : en effet nous avons remarqué précédemment que  $A$  était stochastique ie la somme des coefficients de chaque ligne vaut un (et par symétrie, la somme des coefficients de chaque colonne vaut un).

Donc  $\mathbf{1}$  est un vecteur propre de valeur propre 1.

Nous allons montrer que toutes les valeurs propres de  $A$  sont de valeur absolue plus petites que 1.

Ainsi, comme  $\mathbf{1}^\perp = \text{vect}(u_2, \dots, u_n)$ , c'est  $u_2 \in (\mathbf{1}^\perp) \cap (u \in R^n \mid \|u\|_2=1)$  qui maximise  $\|Au\|_2$  ie  $\lambda(G) = \lambda_2$ , "la seconde plus grande valeur propre".

Pour montrer que les valeurs propres de  $A$  sont de valeur absolue plus petite que 1, montrons que la norme d'une matrice stochastique symétrique est égale à un. Si c'est vrai,  $\|A\|_2=1 \Rightarrow \forall u$  tel que  $\|u\|_2=1$ ,  $\|Au\|_2 \leq 1$ . Donc pour un vecteur  $u_i$  de la base orthonormée de vecteurs propres,  $\|Au_i\|_2 = \|\lambda_i u_i\|_2 = |\lambda_i| \leq 1$ .

Montrons alors que la norme d'une matrice stochastique symétrique est plus petite que un :

soit  $u$  de norme égale à un et  $k \in \mathbb{N}$ . Le produit de deux matrices stochastiques (resp. symétriques) est stochastique (resp. symétrique), donc  $A^k$  pour

$k \in \mathbb{N}$  est stochastique (resp. symétrique).

Notons  $(a_{ij})$  les coefficients de  $A^k$ ,  $u(i)$  ceux de  $u$ .

$$\begin{aligned} \|A^k u\|_2^2 &= \sum_{i=1}^n (\sum_{j=1}^n a_{ij} u(j))^2 \leq (\sum_{i=1}^n \sum_{j=1}^n a_{ij} u(j))^2 \\ &= (\sum_{j=1}^n (\sum_{i=1}^n a_{ij}) u(j))^2 \text{ et comme } A^k \text{ est stochastique symétrique,} \\ &= (\sum_{j=1}^n u(j))^2 \leq n^2 \\ \text{car } \|u\|_2^2 &= \sum_{i=1}^n (u(i))^2 = 1 \Rightarrow \forall i, (u(i))^2 \leq 1 \Leftrightarrow \forall i, |u(i)| \leq 1. \\ \text{On a } \|A^k u\|_2^2 &\leq n^2. \end{aligned}$$

En outre,  $\|A^k u\|_2^2 = \langle A^k u, A^k u \rangle$  et comme  $A^k$  est symétrique,  
 $= \langle A^{2k} u, u \rangle$ . On applique Cauchy-Schwarz :  
 $\leq \|A^{2k} u\|_2 \cdot \|u\|_2 = \|A^{2k} u\|_2$ .

Par une récurrence, on obtient  $\forall i \in \mathbb{N}$ ,  
 $\|A u\|_2^{2^i} \leq \|A^{2^i} u\|_2$  et par le calcul précédent  
 $\leq n$ .  
D'où  $\|A u\|_2 \leq n^{\frac{1}{2^i}} \rightarrow 1$  lorsque  $i \rightarrow \infty$ . ■

Nous pouvons donc à présent définir algébriquement le graphe expander :

## 2.3 Définition algébrique du graphe expander

On dit qu'un graphe  $G$  de  $n$  sommets et  $d$ -régulier est un  **$(n, d, \lambda)$ -graphe expander** si  $\lambda(G) \leq \lambda$  pour un  $\lambda < 1$ .

Comme dans l'approche combinatoire, on définit une **famille de graphes expanders**  $\{G_n\}_{n \in \mathbb{N}}$  comme étant une suite de  $(n, d, \lambda)$ -graphes expanders,  $\lambda \leq 1$  fixé, indexée par  $n$ , le nombre de sommets.

## 3 Relation entre les deux définitions

### 3.1 Définition algébrique $\Rightarrow$ définition combinatoire

Résultat : "Si  $G$  est un  $(n, d, \lambda)$ -graphe expander (donc au sens algébrique),  
alors  $G$  est un  $(n, d, \rho)$ -expander avec  $\rho = \frac{(1 - \lambda)}{2}$  (au sens combinatoire)"

preuve :

Soit  $G$  un  $(n, d, \lambda)$ -graphe expander.

Soit  $S$  un sous-graphe de taille  $\leq \frac{n}{2}$ .

Définissons  $x \in R^n$  comme le vecteur tel que :

si  $i \in S$ ,  $x(i) = |\bar{S}|$ ,

si  $i \in \bar{S}$ ,  $x(i) = -|S|$ .

Notons  $(a_{ij})$  les coefficients de  $A$  (la matrice d'adjacence divisée par  $d$ ) et posons  $Z = \sum_{i,j=1}^n a_{ij}(x(i) - x(j))^2$ .

D'un coté, les termes non nuls de cette somme sont ceux tels que  $i \in S$  et  $j \in \bar{S}$  et vice versa. Soit deux fois la somme sur  $i \in S$  et  $j \in \bar{S}$  ce qui donne

$$2 \frac{1}{d} |E(S, \bar{S})| (|S| + |\bar{S}|)^2.$$

D'un autre coté, on peut développer :

$$\begin{aligned} Z &= \sum_{i,j=1}^n a_{ij} x(i)^2 - 2 \sum_{i,j=1}^n a_{ij} x(i) x(j) + \sum_{i,j=1}^n a_{ij} x(j)^2 \\ &= 2 \|x\|_2^2 - 2 \langle x, Ax \rangle \text{ ce qui donne par Cauchy-Schwarz,} \\ &\geq 2 \|x\|_2 (\|x\|_2 - \|Ax\|_2). \end{aligned}$$

$$\text{On obtient } \frac{1}{d} |E(S, \bar{S})| (|S| + |\bar{S}|)^2 \geq \|x\|_2 (\|x\|_2 - \|Ax\|_2).$$

Ensuite comme  $x \perp \mathbf{1}$  (car  $\sum_{i=1}^n \frac{x(i)}{n} = |S| \cdot |\bar{S}| - |\bar{S}| \cdot |S| = 0$ ),

$x \in \text{vect}(u_2, \dots, u_n)$ , donc

$\|Ax\|_2 \leq \lambda_2 \|x\|_2 \leq \lambda \|x\|_2$  par hypothèse.

Ainsi, on a  $\frac{1}{d} |E(S, \bar{S})| (|S| + |\bar{S}|)^2 \geq (1 - \lambda) \|x\|_2^2$ .

Enfin, il suffit de remarquer que  $\|x\|_2^2 = |S| \cdot |\bar{S}|^2 + |\bar{S}| \cdot |S|^2 = |S| \cdot |\bar{S}| \cdot (|S| + |\bar{S}|)$

En remplaçant  $\|x\|_2^2$  dans l'inégalité, on a finalement :

$$|E(S, \bar{S})| \geq (1 - \lambda) \frac{d \cdot |S| \cdot |\bar{S}|}{|S| + |\bar{S}|} \text{ et comme } |S| \leq \frac{n}{2}, |\bar{S}| \geq \frac{n}{2}, \text{ d'où}$$

$$|E(S, \bar{S})| \geq (1 - \lambda) \frac{d \cdot |S| \cdot \frac{n}{2}}{n} = (1 - \lambda) \frac{d \cdot |S|}{2}.$$

$$\Rightarrow \rho = \frac{(1 - \lambda)}{2}. \blacksquare$$

### 3.2 Définition combinatoire + $\forall k \in \{1 \dots n\}, (k, k) \in E \Rightarrow$ définition algébrique

Résultat : "Si  $G = (V, E)$  est un  $(n, d, \rho)$ -graphe expander (donc au sens combinatoire) et que  $\forall k \in \{1 \dots n\}, (k, k) \in E$ ,

alors  $G$  est un  $(n, d, \lambda)$ -graphe expander (au sens algébrique)

avec  $\lambda = \max(1 - \frac{\rho^2}{2}, 1 - \frac{2}{d})$ ."

preuve :

Soit  $A$  la matrice de marche aléatoire sur  $G$ ,  $(a_{i,j})$  ses coefficients.



♣ Tout d'abord, montrons que  $\lambda_2$  (la seconde plus grande valeur propre de A) est inférieure à  $1 - \frac{\rho^2}{2}$ .

Soit  $u$  un vecteur propre associé à la valeur propre  $\lambda_2$ .  $u \perp 1 = (\frac{1}{n}, \dots, \frac{1}{n})$  (car A est diagonalisable en base orthogonale puisque symétrique, et 1 est un vecteur propre de valeur propre associée  $\lambda_1=1$ ).

On écrit  $u=v+w$  :

les coordonnées de  $v$  sont égales aux coordonnées de  $u$  lorsqu'elles sont positives, et égales à zéro sinon ;

les coordonnées de  $w$  sont égales aux coordonnées de  $u$  lorsqu'elles sont négatives, et égales à zéro sinon.

Comme  $u \perp 1$  et  $u$  n'est pas le vecteur nul,  $v$  et  $w$  sont tous les deux non nuls.

On peut supposer que moins de  $\frac{n}{2}$  coordonnées de  $v$  sont non nuls (sinon, on considère  $-u$  à la place de  $u$ ).

On pose  $Z = \sum_{i,j=1}^n a_{i,j}(v(i)^2 - v(j)^2)$ .

Quitte à permuter les numéros des sommets, on a  $v(1) \geq v(2) \geq \dots \geq v(n)$  avec  $v(i)=0$  pour tout  $i > \frac{n}{2}$ .

Alors,  $Z = \sum_{i,j=1}^n a_{i,j}(v(i)^2 - v(j)^2) = 2 \sum_{i < j}^n a_{i,j} \sum_{k=i}^{j-1} (v(k)^2 - v(k+1)^2)$ .

Pour chaque  $k$ , on compte le nombre de fois que  $(v(k)^2 - v(k+1)^2)$  apparaît dans cette somme : c'est à chaque fois que  $(i,j)$  est tel que  $i \leq k < j$ , et il est multiplié par  $2a_{i,j}$  donc en tout il y en a  $\frac{2}{d} |E(\{1 \dots k\}, \{k+1 \dots n\})|$ .

Pour obtenir  $Z$ , comme  $v(k)=0 \forall k > \frac{n}{2}$ , on somme  $(v(k)^2 - v(k+1)^2)$  pour  $k$  entre 1 et  $\frac{n}{2}$  pondéré par le nombre d'apparitions.

On obtient  $Z = \frac{2}{d} \sum_{k=1}^{\frac{n}{2}} |E(\{1 \dots k\}, \{k+1 \dots n\})| (v(k)^2 - v(k+1)^2)$ ,

et comme, par hypothèse,  $G=(V,E)$  est un  $(n,d,\rho)$ -graphe expander,

on a  $|E(\{1 \dots k\}, \{k+1 \dots n\})| \geq \rho dk$ ,

$Z \geq \frac{2}{d} \sum_{k=1}^{\frac{n}{2}} \rho dk (v(k)^2 - v(k+1)^2) = 2\rho \sum_{k=1}^{\frac{n}{2}} k(v(k)^2 - v(k+1)^2)$

$= 2\rho (\sum_{k=1}^{\frac{n}{2}} kv(k)^2 - \sum_{k=1}^{\frac{n}{2}} kv(k+1)^2) = 2\rho (\sum_{k=1}^{\frac{n}{2}} kv(k)^2 - (k-1)v(k)^2)$

$= 2\rho \|v\|_2^2$

On a donc, d'un coté  $2\rho \|v\|_2^2 \leq Z$ .

D'un autre coté, on va montrer que  $Z^2 \leq 8(1 - \lambda_2) \|v\|_2^4$ .

Ainsi,  $4\rho^2 \|v\|_2^4$ .

Ce qui donnera  $4\rho^2 \|v\|_2^4 \leq 8(1 - \lambda_2) \|v\|_2^4$  ie

$\lambda_2 \leq 1 - \frac{\rho^2}{2}$ , soit le résultat attendu.

$$\begin{aligned}
& \langle Av, v \rangle + \langle Aw, v \rangle = \langle Au, v \rangle = \lambda_2 \langle u, v \rangle = \lambda_2 \langle v + w, v \rangle \\
& = \lambda_2 \|v\|_2^2 + \lambda_2 \langle w, v \rangle = \lambda_2 \|v\|_2^2, \text{ et comme } \langle Aw, v \rangle \text{ est négatif,} \\
& \langle Av, v \rangle \geq \lambda_2 \|v\|_2^2 \iff \\
& 1 - \lambda_2 \geq 1 - \frac{\langle Av, v \rangle}{\|v\|_2^2} \text{ puisque } v \text{ est non nul,} \\
& = \frac{\|v\|_2^2 - \langle Av, v \rangle}{\|v\|_2^2} = \frac{\sum_{i=1}^n v(i)^2 - \langle Av, v \rangle}{\|v\|_2^2} \\
& = \frac{\sum_{i=1}^n v(i)^2 - 2 \langle Av, v \rangle + \sum_{j=1}^n v(j)^2}{\|v\|_2^2}, \text{ et comme } A \text{ est stochastique symétrique,} \\
& = \frac{\sum_{i,j=1}^n a_{i,j} v(i)^2 - 2 \langle Av, v \rangle + \sum_{i,j=1}^n a_{i,j} v(j)^2}{\|v\|_2^2} \\
& = \frac{\sum_{i,j=1}^n a_{i,j} (v(i) - v(j))^2}{2 \|v\|_2^2} = \frac{(\sum_{i,j=1}^n a_{i,j} (v(i) - v(j))^2) \cdot (\sum_{i,j=1}^n a_{i,j} (v(i) + v(j))^2)}{2 \|v\|_2^2 \cdot (\sum_{i,j=1}^n a_{i,j} (v(i) + v(j))^2)}
\end{aligned}$$

On applique l'inégalité de Cauchy-Schwartz au produit scalaire  $\langle x, y \rangle = \sum_{i,j} a_{i,j} x_{i,j} y_{i,j}$  où  $x$  et  $y$  sont indexé par  $(i,j)$  :

$$1 - \lambda_2 \geq \frac{\{\sum_{i,j=1}^n a_{i,j} (v(i) - v(j))(v(i) + v(j))\}^2}{2 \|v\|_2^2 \cdot (\sum_{i,j=1}^n a_{i,j} (v(i) + v(j))^2)} = \frac{Z^2}{2 \|v\|_2^2 \cdot (\sum_{i,j=1}^n a_{i,j} (v(i) + v(j))^2)}$$

et en développant le dénominateur,  $= \frac{Z^2}{2 \|v\|_2^2 \cdot (2 \|v\|_2^2 + 2 \langle Av, v \rangle)}$ ,

puis comme  $A$  est symétrique stochastique,  $\|A\|_2 \leq 1 \iff \forall v, \|Av\|_2 \leq \|v\|_2$   
 $\Rightarrow$  par Cauchy-Schwartz,  $\langle Av, v \rangle \leq \|v\|_2^2$ .

On obtient  $1 - \lambda_2 \geq \frac{Z^2}{8 \|v\|_2^4}$  soit  $8(1 - \lambda_2) \|v\|_2^4 \geq Z^2$ .

♣ Maintenant, montrons que  $\lambda_2$  est supérieure à  $\frac{2}{d} - 1$ .

On considère  $A'$ , la matrice de marche aléatoire sur le graphe  $G'$ , qui n'est rien d'autre que le graphe  $G=(V,E)$  auquel on a enlevé une arête  $(k,k)$   $\forall k \in \{1...n\}$ ; comme  $G=(V,E)$  avait  $\forall k \in \{1...n\}, (k,k) \in E$ ,  $G'$  est  $d-1$  régulier.  $A'$  est de norme 1 (puisque  $A'$  est stochastique symétrique), donc sa plus petite valeur propre est supérieure à  $-1$ .

On a  $A = \frac{d-1}{d} A' + \frac{1}{d} Id$ , donc si  $u$  est un vecteur propre de  $A$ , c'est un vecteur propre de  $A'$  (tout vecteur est vecteur propre de  $Id$ ), de valeur propre correspondante supérieure à  $-\frac{d-1}{d} + \frac{1}{d} = -1 + \frac{2}{d}$ .

Avec les deux ♣, on a montré que la SLEV  $\lambda_2$  est telle que  $-1 + \frac{2}{d} \leq$

$$\lambda_2 \leq 1 - \frac{\rho^2}{2}.$$

$$\text{Ainsi, } |\lambda_2| \leq \max(1 - \frac{d}{2}, 1 - \frac{\rho^2}{2}).$$

G est donc bien un  $(n,d,\lambda)$ -graphe expander, avec  $\lambda = \max(1 - \frac{d}{2}, 1 - \frac{\rho^2}{2})$ . ■

## 4 Construction d'une famille de graphes expanders

### 4.1 Quelques définitions utiles

Afin d'utiliser les graphes expanders en informatique théorique, il est nécessaire de construire une famille de graphes qui aura des contraintes de complexité. Définissons ce qu'est une famille de graphes fortement explicite :

Définition : Considérons une famille de graphes  $d$ -réguliers  $\{G_n\}_{n \in \mathbb{N}}$ . Cette famille est dite **fortement explicite** si il existe un algorithme qui fonctionne en temps polynomial et qui, sur l'entrée  $(n,v,i)$  où  $v \in V$  et  $i \in \{1 \dots d\}$ , retourne le  $i^{eme}$  sommet de  $v$ .

La famille que nous allons construire sera fortement explicite afin de pouvoir l'utiliser dans la dernière partie.

Définissons maintenant la fonction "voisin" d'un graphe G, que l'on notera  $N_G$ .

Définition :

$$N_G : V \times \{1 \dots d\} \longrightarrow V \times \{1 \dots d\}$$

$$(v,i) \longmapsto (v',j)$$

c'est-à-dire qu'à chaque couple (sommet  $v$ , numéro de voisin) **la fonction "voisin"**  $N_G$  associe le couple (voisin en question  $v'$ , numéro de  $v$  en tant que voisin de  $v'$ ). La fonction "voisin" caractérise le graphe.

Nous construirons la famille de graphes expanders fortement explicite par récurrence à l'aide de trois "produits de graphes" que nous allons définir dès maintenant :

### Définition : le produit matriciel de graphes

Soient  $G_1$  et  $G_2$  deux graphes de même taille,  $B_1$  et  $B_2$  leurs matrices d'adjacence respectives. Le produit matriciel de ces deux graphes, noté  $G_1.G_2$ , est simplement égal au graphe représenté par la matrice d'adjacence  $B_1.B_2$ . Notons que si  $G$  est un  $(n,d,\lambda)$ -graphe expander (au sens algébrique),  $G^k$  est un  $(n,d^k,\lambda^k)$ -graphe expander.

En effet,

- $G^k$  garde bien le même nombre de sommets puisque  $B^k$  est toujours de taille  $n \times n$ .

- si on note  $b_{i,j}$  les coefficients de  $B$  et  $b_{i,j}^{(k)}$  ceux de  $B^k$ ,

$$b_{i,j}^{(k)} = \sum_{(i_1, \dots, i_{k-1}) \in \{1 \dots n\}^{k-1}} b_{i,i_1} b_{i_1,i_2} \dots b_{i_{k-1},j},$$

$$\text{et le degré de } G^k \text{ est } \sum_{j=1}^n b_{i,j}^{(k)} = \sum_{(i_1, \dots, i_{k-1}) \in \{1 \dots n\}^{k-1}} b_{i,i_1} b_{i_1,i_2} \dots b_{i_{k-2},i_{k-1}} \sum_{j=1}^n b_{i_{k-1},j}$$

$$= \sum_{(i_1, \dots, i_{k-1}) \in \{1 \dots n\}^{k-1}} b_{i,i_1} b_{i_1,i_2} \dots b_{i_{k-2},i_{k-1}} d \text{ puisque } G \text{ est de degré } d$$

(la somme des coefficients de chaque ligne =  $d$ )

$$= d \sum_{(i_1, \dots, i_{k-2}) \in \{1 \dots n\}^{k-2}} b_{i,i_1} b_{i_1,i_2} \dots b_{i_{k-3},i_{k-2}} \sum_{j=1}^n b_{i_{k-2},j} = \dots = d^k.$$

On rappelle que  $A = \frac{1}{d}B$  est la matrice de marche aléatoire sur  $G$ . On

a donc montré que la matrice de marche aléatoire sur  $G^k$  est  $A^k$ .

- Soit  $\lambda$  une valeur propre de  $A = \frac{1}{d}B$ ,  $x$  un vecteur propre associé.

Alors on a trivialement  $A^k x = \lambda^k x$ . Donc les valeurs propres de  $A^k$  sont les puissances  $k^{ieme}$  des valeurs propres de  $A$ . On a donc  $\lambda(G^k) = \lambda(G)^k$  (voir section 2.2 pour la définition de  $\lambda(G)$ , la SLEV).

Nous aurons besoin de cette remarque lors de la construction de la famille de graphes.

### Définition : le produit tensoriel de graphes.

Soient

- $G_1 = (V_1, E_1)$  un graphe de taille  $n_1$ ,  $d_1$ -régulier

- $G_2 = (V_2, E_2)$  un graphe de taille  $n_2$ ,  $d_2$ -régulier.

Soient  $(v_1, v_2) \in V_1 \times V_2$  et  $(i, j) \in \{1 \dots d_1\} \times \{1 \dots d_2\}$  tels que :

- $N_{G_1}(v_1, i) = (v'_1, i')$

- $N_{G_2}(v_2, j) = (v'_2, j')$

Alors,

$$G = G_1 \otimes G_2$$

$$\Updownarrow$$

$G = (V, E)$  avec  $V = (V_1 \times V_2)$  et  $N_G((v_1, v_2), (i, j)) = ((v'_1, v'_2), (i', j'))$

En d'autres termes,

- chaque couple (sommet de  $G_1$ , sommet de  $G_2$ ) est un sommet du graphe  $G$
- un sommet  $v$  de  $G$  est un voisin d'un sommet  $w$  ssi les coordonnées du couple représenté par  $v$  sont respectivement voisins des coordonnées du couple représenté par  $w$ .

Ainsi,  $G = G_1 \otimes G_2$  est de taille  $n_1 n_2$  et  $d_1 d_2$ -régulier.

Montrons que si  $G_1$  est un  $(n_1, d_1, \lambda_1)$ -graphe expander et  $G_2$  un  $(n_2, d_2, \lambda_2)$ -graphe expander (au sens algébrique),  $G = G_1 \otimes G_2$  est un  $(n_1 n_2, d_1 d_2, \max(\lambda_1, \lambda_2))$ -graphe expander.

Il reste à montrer que  $\lambda(G) \leq \max(\lambda_1, \lambda_2)$ . Soient  $B$ ,  $B_1$  et  $B_2$  les matrices d'adjacences respectives de  $G$ ,  $G_1$  et  $G_2$ . On note  $b_{(i,i'),(j,j')}$ ,  $b_{i,j}^1$  et  $b_{i',j'}^2$  leurs coefficients. D'après la définition de  $G = G_1 \otimes G_2$ , le nombre d'arêtes liant deux sommets  $v = (v_1, v_2)$  et  $w = (w_1, w_2)$  de  $G$  est égal au nombre d'arêtes liant  $v_1$  à  $w_1$  multiplié par le nombre d'arêtes liant  $v_2$  à  $w_2$ . On a donc  $\forall ((i, i'), (j, j')) \in \{1 \dots n_1\}^2 \times \{1 \dots n_2\}^2$ ,  $b_{(i,i'),(j,j')} = b_{i,j}^1 b_{i',j'}^2$ , et la même égalité pour les coefficients des matrices de marche aléatoire. Soient  $\lambda_1$ ,  $\lambda_2$  des valeurs propres respectivement de  $A_1 = \frac{1}{d_1} B_1$  et  $A_2 = \frac{1}{d_2} B_2$ , et  $x$ ,  $y$  des vecteurs propres associés. Définissons  $z = x \otimes y$  comme étant le vecteur dont les coordonnées sont  $z_{i,j} = x_i \cdot y_j$  ( $(i, j) \in \{1 \dots n_1\} \times \{1 \dots n_2\}$ ). Montrons que  $\lambda_1 \lambda_2$  est valeur propre de  $A = \frac{1}{d_1 d_2} B$  de vecteur propre associé  $z$ . Ainsi  $\lambda(G) = \lambda(G_1) \cdot 1$  ou  $= 1 \cdot \lambda(G_2)$ , d'où le résultat.

$$\begin{aligned} (Az)_{(i,j)} &= \sum_{(i',j') \in \{1 \dots n_1\} \times \{1 \dots n_2\}} A_{((i,j),(i',j'))} z_{(i',j')} \\ &= \sum_{(i',j') \in \{1 \dots n_1\} \times \{1 \dots n_2\}} a_{(i,i')}^1 a_{(j,j')}^2 x_{i'} y_{j'} \\ &= (A_1 x)_i \cdot (A_2 y)_j = \lambda_1 x_i \lambda_2 y_j = \lambda_1 \lambda_2 z_{(i,j)} \end{aligned}$$

On a donc bien ce que l'on voulait.

**Définition : le produit de remplacement.**

Nous considérons ici :

- $G_1 = (V_1, E_1)$  un graphe de taille  $n$ ,  $d$ -régulier
- $G_2 = (V_2, E_2)$  un graphe de taille  $d$ ,  $d'$ -régulier.

Alors,

$$G = G_1 \star G_2$$

$$\Updownarrow$$

- $G = (V, E)$  avec  $V = (V_1 \times V_2)$ .
- si  $N_{G_2}(v_2, j) = (w_2, j')$ , alors  $\forall v_1 \in V_1$ ,  $N_G((v_1, v_2), (i, j)) = ((v_1, w_2), (i, j'))$
- $\forall v_1 \in V_1$ , si  $N_{G_1}(v_1, i) = (w_1, j)$ , alors il y a  $d'$  arêtes entre  $(v_1, i = v_2) \in G$  et  $(w_1, j = w_2) \in G$ .

En fait, puisque le degré de  $G_1$  est égal au nombre de sommets de  $G_2$ , on a simplement remplacé chaque sommet de  $G_1$  par une copie du graphe  $G_2$  (avec ses arêtes), et lorsque le  $i^{eme}$  voisin d'un sommet  $v_1$  de  $G_1$  est  $w_1$  pour lequel  $v_1$  est le  $j^{eme}$  voisin, on relie avec d' sommets le  $i^{eme}$  sommet de la copie de  $G_2$  située en  $v_1$  au  $j^{eme}$  sommet de la copie de  $G_2$  située en  $w_1$ .

Ainsi,  $G$  est de taille  $nd$ , et de chaque sommet partent

- $d'$  arêtes en tant que sommet de la copie de  $G_2$
- $d'$  autres arêtes en tant que représentation du  $i^{eme}$  degré de  $G_1$

$G$  est donc de degré  $2d'$ . On peut aussi montrer, mais c'est très long et peu intéressant, que si  $\lambda(G_1) \leq 1 - \epsilon_1$  et  $\lambda(G_2) \leq 1 - \epsilon_2$ , alors  $\lambda(G_1 \star G_2) \leq 1 - \frac{1 - \epsilon_1 \epsilon_2^2}{24}$ . Donc si  $G_1$  est un  $(n, d, 1 - \epsilon_1)$ -graphe expander et  $G_2$  un  $(d, d', 1 - \epsilon_2)$ -graphe expander (au sens algébrique), alors  $G = G_1 \star G_2$  est un  $(nd, 2d', 1 - \frac{1 - \epsilon_1 \epsilon_2^2}{24})$ -graphe expander.

Nous avons maintenant les produits de graphe nous permettant de construire une famille de graphes expanders, ainsi que leurs différentes propriétés.

## 4.2 Construction de la famille

Construisons une la famille  $\{G_k\}_{k \in \mathbb{N}}$  où  $G_k$  est de taille  $(2d)^{100k}$ . Soient  $d \in \mathbb{N}$ , et  $G$  un  $((2d)^{100}, d, 0.01)$ -graphe expander, définissons  $\{G_k\}_{k \in \mathbb{N}}$  par récurrence comme suit :

- $G_1$  un  $((2d)^{100}, 2d, \frac{1}{2})$ -graphe expander,
- $G_2$  un  $((2d)^{200}, 2d, \frac{1}{2})$ -graphe expander,
- $\forall k > 2$ ,  $G_k = (G_{\lfloor \frac{k-1}{2} \rfloor} \otimes G_{\lfloor \frac{k}{2} \rfloor})^{50} \star G$

Montrons par récurrence que  $G_k = (V_k, E_k)$  est un  $((2d)^{k100}, 2d, 1 - \frac{1}{50})$ -graphe expander :

les initialisations sont triviales.

Si l'on note  $n_k = |V_k|$ , par les propriétés des trois produits définis précédemment,

$$\begin{aligned} n_k &= n_{\lfloor \frac{k-1}{2} \rfloor} n_{\lfloor \frac{k}{2} \rfloor} (2d)^{100} = (2d)^{100 \lfloor \frac{k-1}{2} \rfloor} (2d)^{100 \lfloor \frac{k}{2} \rfloor} (2d)^{100} \\ &= (2d)^{100(\lfloor \frac{k-1}{2} \rfloor + \lfloor \frac{k}{2} \rfloor + 1)} \end{aligned}$$

par l'hypothèse de récurrence. Or,  $\lfloor \frac{k-1}{2} \rfloor + \lfloor \frac{k}{2} \rfloor + 1 = k$ .  
Donc  $n_k = (2d)^{k \cdot 100}$ .

Si le degré de  $G_{\lfloor \frac{k-1}{2} \rfloor}$  et celui de  $G_{\lfloor \frac{k}{2} \rfloor}$  sont égaux à  $2d$  (hypothèse de récurrence), le degré de  $G_{\lfloor \frac{k-1}{2} \rfloor} \otimes G_{\lfloor \frac{k}{2} \rfloor}$  est  $(2d)^2$ , celui de  $(G_{\lfloor \frac{k-1}{2} \rfloor} \otimes G_{\lfloor \frac{k}{2} \rfloor})^{50}$  est  $(2d)^{2 \times 50} = (2d)^{100}$  et enfin celui de  $(G_{\lfloor \frac{k-1}{2} \rfloor} \otimes G_{\lfloor \frac{k}{2} \rfloor})^{50} \star G$  est  $2d$ .

Enfin, si  $\lambda(G_{\lfloor \frac{k-1}{2} \rfloor})$  et  $\lambda(G_{\lfloor \frac{k}{2} \rfloor})$  sont inférieurs à  $1 - \frac{1}{50}$  (hypothèse de récurrence),  
 $\lambda(G_{\lfloor \frac{k-1}{2} \rfloor} \otimes G_{\lfloor \frac{k}{2} \rfloor}) \leq \max(\lambda(G_{\lfloor \frac{k-1}{2} \rfloor}), \lambda(G_{\lfloor \frac{k}{2} \rfloor})) \leq 1 - \frac{1}{50}$ ,  
 $\lambda((G_{\lfloor \frac{k-1}{2} \rfloor} \otimes G_{\lfloor \frac{k}{2} \rfloor})^{50}) \leq \lambda((G_{\lfloor \frac{k-1}{2} \rfloor} \otimes G_{\lfloor \frac{k}{2} \rfloor})^{50}) \leq (1 - \frac{1}{50})^{50}$ ,  
donc,  $\lambda((G_{\lfloor \frac{k-1}{2} \rfloor} \otimes G_{\lfloor \frac{k}{2} \rfloor})^{50} \star G) \leq 1 - \frac{(1 - (1 - \frac{1}{50})^{50})(1 - 0.01)}{24} \leq 1 - \frac{1}{50}$ .

Pour calculer la fonction "voisin" de  $G_k$ , on appelle les fonctions "voisin" de  $G_{\lfloor \frac{k-1}{2} \rfloor}$  et  $G_{\lfloor \frac{k}{2} \rfloor}$ . Il y a alors 100 appels récursifs et la profondeur de la récursivité vaut au plus  $\log_2(k)$ . Donc on a environ  $k^7$  appels pour un graphe de taille  $(2d)^{100k}$ . La famille est donc bien fortement explicite.

**Remarque :** Pour avoir un tel graphe pour tout  $n \in N$ , il suffit de regrouper les sommets du graphe  $G$  de taille  $(2d)^{100k}$  tel que  $(2d)^{100(k-1)} < n < (2d)^{100k}$ . Plus précisément, on veut construire  $G'$  de taille  $n$  à partir de  $G$ . On peut toujours écrire  $0 < (2d)^{100k} - n = n_1 n_2$  avec  $n_1 \leq n$ . On pose  $a = n_2$  et  $b = n_1 + 1$ , et le graphe  $G'$  de taille  $n$  sera construit de la manière suivante : on a bien  $ab = n_1 n_2 + n_1 \leq (2d)^{100k}$  donc on considère  $a$  sous-graphes disjoints de taille  $b$  de  $G$ . Ces sous-graphes ainsi que les sommets qui restent formeront les sommets de  $G'$ . Un sommet de  $G'$  issu d'un sous-graphe de  $G$  conserve toutes les arêtes qui partaient d'un sommet présents dans le sous-graphe. Ainsi, pour rendre  $G'$  régulier, on rajoute à tout sommet de  $G'$  issu d'un sommet de  $G$  des arêtes le liant tout seul. De cette manière, la

taille de  $G'$  vaut  $(2d)^{100k} - ab + a = (2d)^{100k} - a(b-1) = (2d)^{100k} - n_1 n_2 = n$ . En raisonnant de manière combinatoire,  $G'$  reste un graphe expander si  $G$  en était un (et son degré augmente).

## 5 Réduction de l'erreur de RP par les graphes expanders

Rappelons qu'un langage  $L$  est dans RP lorsqu'il existe une machine de Turing  $M$  fonctionnant en temps polynomial telle que :

- si  $x$  appartient à  $L$ ,  $\Pr[M(x) \text{ accepte}] \geq \frac{2}{3}$
- si  $x$  n'appartient pas à  $L$ ,  $\Pr[M(x) \text{ rejète}] = 1$

La réduction de l'erreur de RP "classique" consiste à exécuter  $k$  fois l'algorithme, et à prendre la réponse majoritaire. On réduit alors l'erreur en passant de  $\frac{1}{3}$  à  $2^{-\Omega(k)}$ . La méthode que nous allons voir permet la même réduction, mais en utilisant moins de bits aléatoires (si l'algorithme nécessite  $n$  bits aléatoires, la méthode classique en utilisera  $k.n$ ).

Soit  $\{0, 1\}^n$  l'ensemble des bits possible pour l'algorithme. Prenons un  $(2^n, d, \frac{1}{10})$ -graphe expander  $G=(V, E)$  fortement explicite de matrice de marche aléatoire  $A$ . Notre méthode consiste à choisir un sommet  $v_1$  uniformément sur  $V$ , et à procéder à une marche aléatoire de longueur  $k-1$  sur  $G$  pour obtenir  $v_2, v_3, \dots, v_k$ . Le graphe est fortement explicite : on rappelle que cela signifie qu'il fonctionne en temps polynomial en prenant en entrée  $(|V|, v \in V, i \in \{1, \dots, d\})$  et donnant le  $i^{eme}$  voisin de  $v$ . L'entrée et la sortie sont donc de taille  $\log(|V|)$ . Ainsi, il fonctionne en temps  $\text{polylog}(|V|)$ , et donc ici,  $\text{polylog}(2^n)$  ce qui reste polynomial. Cette marche aléatoire est donc bien créée en temps polynomial. On identifie  $V$  à  $\{0, 1\}^n$ , et on lance  $k$  fois l'algorithme : la première fois avec  $v_1$  comme série de bits pour l'algorithme, puis  $v_2, \dots$ , puis  $v_k$ . A la fin, on accepte si l'algorithme a accepté au moins une fois, sinon on rejète. On a alors utilisé  $n$  bits aléatoires pour choisir  $v_1$ , et  $O(\log(d))=O(1)$  pour chaque  $v_k$ , soit  $n+O(k)$  bits. On suppose que le mot en entrée est dans le langage, et on pose  $S$  le sous-ensemble de  $V$  qui représente l'ensemble des  $n$ -uplets de "mauvais" bits, c'est-à-dire l'ensemble des  $n$ -uplets de bits qui font que l'algorithme rejète alors que le mot est dans le langage. Puisque le langage est dans RP, on peut affirmer que  $|S| \leq \frac{2^n}{3}$ .

Afin de montrer que ce nouvel algorithme réduit bien l'erreur, nous allons



montrer que  $\Pr[\forall 1 \leq i \leq k, v_i \in S] \leq 2^{-\Omega(k)}$ , car c'est dans ce cas que l'algorithme que nous avons décrit rejette :

Notons  $S_i = \{X_i \in S\}$ ,  $m = 2^n$ , et  $1$  la distribution uniforme sur  $V = \{1, \dots, m\}$ . Soit  $Z$  l'application qui à tout vecteur de  $R^m$ , annule les coefficients dont l'indice n'est pas dans  $S$ .

Soit  $p_i$  la loi de  $v_i$  conditionnée par  $S_1, \dots, S_i$ , et  $1 = (\frac{1}{m}, \dots, \frac{1}{m})$ .

On a

$$\begin{aligned} - p_1 &= \frac{1}{\Pr[S_1]} Z1 \\ - p_2 &= \frac{1}{\Pr[S_1, S_2]} ZAZ1 = \frac{1}{\Pr[S_1]} \frac{1}{\Pr[S_2|S_1]} ZAZ1 \\ - \dots \\ - p_i &= \frac{1}{\Pr[S_1]\Pr[S_2|S_1]\dots\Pr[S_i|S_1, \dots, S_{i-1}]} (ZA)^{i-1} Z1 \end{aligned}$$

En effet, procédons par récurrence :

$Z1$  est la distribution uniforme dont on a annulé tous les coefficients dont l'indice n'est pas dans  $S$ . En divisant par  $\Pr[S_1]$ , on obtient la probabilité uniforme sur  $S$  ie la loi de probabilité du premier sommet sachant qu'on le choisit dans  $S$ .

Ensuite, supposons que  $p_i = \frac{1}{\Pr[S_1]\Pr[S_2|S_1]\dots\Pr[S_i|S_1, \dots, S_{i-1}]} (ZA)^{i-1} Z1$  soit la loi de probabilité du  $i^{eme}$  sommet sachant que tous les sommets jusqu'au  $i^{eme}$  sont dans  $S$ . Mettons nous dans la situation où tous les sommets jusqu'au  $i^{eme}$  sont dans  $S$ , ie dans la situation  $\bigcap_{j=1}^i S_j$ . On effectue un pas de plus de la marche aléatoire : la loi du sommet trouvé est  $Ap_i$  puisque nous sommes dans  $\bigcap_{j=1}^i S_j$ . Afin d'avoir  $p_i$ , on raisonne comme dans le cas 1 : on annule toutes les coefficients de  $Ap_i$  dont l'indice n'est pas dans  $S$  ie on multiplie par  $Z$ , et on divise par la probabilité que  $v_{i+1}$  soit dans  $S$ . Or ici on a raisonné dans  $\bigcap_{j=1}^i S_j$ , donc on divisera en fait par  $\Pr[S_{i+1}|S_1 \dots S_i]$ . On obtient bien  $p_{i+1} = \frac{1}{\Pr[S_{i+1}|S_1 \dots S_i]} ZAp_i = \frac{1}{\Pr[S_1]\Pr[S_2|S_1]\dots\Pr[S_{i+1}|S_1, \dots, S_i]} (ZA)^i Z1$ . L'hérédité est donc montrée.

$$\text{On a } p_k = \frac{1}{\Pr[S_1]\Pr[S_2|S_1]\dots\Pr[S_k|S_1, \dots, S_{k-1}]} (ZA)^{k-1} Z1 = \frac{1}{\Pr[\bigcap_{j=1}^k S_j]} (ZA)^{k-1} Z1.$$

$p_k$  étant une loi de probabilité, sa "norme 1"  $\|p_k\|_1$ , ie la somme (de la valeur absolue) de ses coordonnées, vaut 1.

Ainsi,  $\| \frac{1}{Pr[\bigcap_{j=1}^k S_k]} (ZA)^{k-1} Z1 \|_1 = 1$ ,

donc  $Pr[\bigcap_{j=1}^k S_k] = \| (ZA)^{k-1} Z1 \|_1$ . Or,  $\forall x \in R^m$ ,  $\| x \|_1 \leq \sqrt{m} \| x \|_2$ . Donc,  $Pr[\bigcap_{j=1}^k S_k] \leq \sqrt{m} \| (ZA)^{k-1} Z1 \|_2$

Définissons  $I$  comme étant la matrice dont tous les coefficients valent  $\frac{1}{m}$ . Soit  $x \in R^m$ .  $x$  s'écrit  $x = y + z$  avec  $y = \omega 1$  et  $z \perp 1$ .

$(10A - 9I)y = (10A - 9I)\omega 1 = 10\omega 1 - 9\omega 1 = y$  puisque  $A$  et  $I$  sont des matrices stochastiques ( $1$  est vecteur propre de valeur propre associée  $1$ ).

$(10A - 9I)z = 10Az$  car  $z \perp 1$  donc la somme de ses coordonnées est nulle (et  $Iz$  est le vecteur dont les coordonnées sont toutes égales à la somme des coordonnées de  $z$ ).

Ainsi,  $(10A - 9I)x = y + 10Az$ .

$\| ((10A - 9I)x) \|_2^2 = \| y + 10Az \|_2^2 = \| y \|_2^2 + 100 \| Az \|_2^2$   
par Pythagore, (car  $A$  étant symétrique,  $\langle Az, 1 \rangle = \langle z, A1 \rangle = \langle z, 1 \rangle = 0$  ie  $z \in 1^\perp$ ),  
 $\leq \| y \|_2^2 + 100\lambda^2 \| z \|_2^2$  car  $z \in 1^\perp$ ,  
 $= \| y \|_2^2 + \| z \|_2^2 = \| x \|_2^2$  encore par Pythagore, et on rappelle que  $\lambda = \frac{1}{10}$ .

On a donc

$$\| (10A - 9I)x \|_2 \leq \| x \|_2 \quad (1)$$

$A = \frac{1}{10}(10A - 9I) + \frac{9}{10}I$ , donc  $ZA = \frac{1}{10}Z(10A - 9I) + \frac{9}{10}ZI$  et  $\forall x \in R^m$ ,  
 $\| ZAx \|_2 \leq \frac{1}{10} \| Z(10A - 9I)x \|_2 + \frac{9}{10} \| ZIx \|_2$ .

Or,  $\| Z1 \|_2 = \sqrt{\sum_{i=1}^m (Z1)_i^2} = \sqrt{|S| \frac{1}{m^2}} \leq \sqrt{\frac{m}{3} \frac{1}{m^2}} = \frac{1}{\sqrt{3m}}$  puisque la taille de  $S$  est plus petit qu'un tiers de la taille du graphe (on rappelle que  $S$  représente l'ensemble des "mauvais" n-uplets de bits).

Ainsi, comme  $Ix = (\sum_{i=1}^n x_i)1$ ,  
 $\| ZIx \|_2 = | \sum_{i=1}^n x_i | \| Z1 \|_2 \leq \sum_{i=1}^n |x_i| \| Z1 \|_2 = \| x \|_1 \cdot \| Z1 \|_2$   
 $\leq \sqrt{m} \frac{\| x \|_2}{\sqrt{3m}} = \frac{\| x \|_2}{\sqrt{3}}$  par l'inégalité entre norme 1 et norme 2 donnée plus haut.

On a donc

$$\| Z1 \|_2 \leq \frac{1}{\sqrt{3m}} \quad (2)$$

et

$$\forall x \in R^m, \| ZIx \|_2 \leq \frac{\| x \|_2}{\sqrt{3}} \quad (3)$$

De plus,

$$\forall x \in R^m, \| Zx \|_2 \leq \| x \|_2 \quad (4)$$

puisque Z ne fait qu'annuler certaines coordonnées.

On revient à l'inégalité précédente :

$$\begin{aligned} \| ZAx \|_2 &\leq \frac{1}{10} \| Z(10A - 9I)x \|_2 + \frac{9}{10} \| ZIx \|_2 \\ &\leq \frac{1}{10} \| (10A - 9I)x \|_2 + \frac{9}{10} \frac{\| x \|_2}{\sqrt{3}} \text{ par (4) et (3),} \\ &\leq \frac{\| x \|_2}{10} + \frac{9}{10} \frac{\| x \|_2}{\sqrt{3}} \text{ par (1).} \end{aligned}$$

Donc,

$$\forall x \in R^m, \| ZAx \|_2 \leq \| x \|_2 \left( \frac{1}{10} + \frac{9}{10\sqrt{3}} \right) \quad (5)$$

Enfin,  $Pr[\bigcap_{j=1}^k S_k] \leq \sqrt{m} \| (ZA)^{k-1} Z1 \|_2$

$\leq \sqrt{m} \left( \frac{1}{10} + \frac{9}{10\sqrt{3}} \right) \| (ZA)^{k-2} Z1 \|_2$  par (5), qu'on réapplique k-2 fois,

$$\begin{aligned} &\vdots \\ &\leq \sqrt{m} \left( \frac{1}{10} + \frac{9}{10\sqrt{3}} \right)^{k-1} \| Z1 \|_2 \\ &\leq \sqrt{m} \left( \frac{1}{10} + \frac{9}{10\sqrt{3}} \right)^{k-1} \frac{1}{\sqrt{3m}} = \frac{1}{\sqrt{3}} \left( \frac{1}{10} + \frac{9}{10\sqrt{3}} \right)^{k-1} \\ &\leq \frac{10}{\sqrt{3}+9} 2^{\ln_2(\frac{\sqrt{3}+9}{10\sqrt{3}})k} = 2^{\ln_2(\frac{10}{\sqrt{3}+9}) + \ln_2(\frac{\sqrt{3}+9}{10\sqrt{3}})k} = 2^{\ln_2(\frac{1}{\sqrt{3}})k} = 2^{-ck} \text{ avec } c > 0. \end{aligned}$$

On a bien majoré la probabilité d'erreur par  $2^{-\Omega(k)}$  en utilisant seulement  $m+O(k)$  bits aléatoires.

La vision intuitive du fait que les graphes expanders ont pu permettre la réduction de l'erreur, peut être résumé par ce qui suit : la définition combinatoire du graphe expander nous apprend qu'un sous-graphe plus petit que la moitié du graphe possède beaucoup d'arêtes sortantes, ie d'arêtes qui sortent du sous-graphe en question : une marche aléatoire sur le graphe a beaucoup de chances de sortir d'un tel sous-graphe. Comme l'erreur est identifiée à ce sous-graphe (l'erreur pour RP est même plus petite que  $\frac{1}{3}$ ), on a beaucoup de chances de "sortir de l'erreur" si le mot est dans le langage. Si la marche

n'en sort pas, c'est-à-dire si l'algorithme n'a jamais accepté pour tous les n-uplets-sommets proposés par la marche, c'est vraisemblablement que le mot n'est pas dans le langage.

#### **BIBLIOGRAPHIE :**

- Computational Complexity : A Modern Approach, Sanjeev Arora and Boaz Barak.
- Eigenvalues, expanders and superconcentrators (1984), N. Alon and V. D. Milman.
- Eigenvalues and expanders (1986), N. Alon.
- How to recycle random bits (1989) R. Impagliazzo and D. Zuckerman.
- Pseudorandom walks on regular digraphs and the RL vs. L problem. (2006) O. Reingold, L. Trevisan, and S. Vadhan.
- Derandomized squaring of graphs (2005), E. Rozenman and S. Vadhan.

### 3. STAGE DE MAGISTÈRE

J'ai effectué le stage de Magistère en fin de deuxième année. Je souhaitais profiter de ce stage pour apprendre un langage informatique classique tel que le langage c++. Sylvain Peyronnet, alors chercheur à l'Institut National de Recherche en Informatique et Automatique, me propose un projet à réaliser dans le langage que je souhaite. Deux chercheurs en économie, M.Chevé et R.Congar, ont résolu analytiquement un problème d'optimisation d'une politique économique pour certaines données initiales définissant le modèle économique : le but du stage sera de développer un algorithme permettant une résolution numérique approchée pour n'importe quelles données initiales du modèle.

Mon responsable de stage a fait en sorte de me laisser relativement autonome dans la recherche de l'outil mathématique permettant la résolution du problème, ainsi que dans le développement du programme, me donnant cependant des conseils techniques, et me prévenant des pièges dans lesquels il ne fallait pas tomber : cette pédagogie et la confiance qu'il m'a accordée m'ont fait beaucoup évoluer, et ont sans doute contribué à l'orientation que j'ai choisie par la suite. Le rapport de ce stage, ainsi qu'un petit résumé détaillant l'organisation du code c++ suivent.

Maitre de stage: Sylvain Peyronnet  
Institut d'accueil: INRIA  
Université Paris-Sud XI  
Stage de Magistère de Mathématiques (fin de M1)

# Approximation des solutions du modèle Chevé Congar

---

Nicolas Drougard

Saclay, le 17 octobre 2011

# Table des matières

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Notations</b>  | <b>3</b>  |
| <b>2</b> | <b>Présentation du problème</b>   | <b>3</b>  |
| <b>3</b> | <b>Démarche et but</b>  | <b>4</b>  |
| <b>4</b> | <b>Mise en place de l'équation de Programmation Dynamique</b>                   | <b>4</b>  |
| 4.1      | Introduction de la formule d'Itô . . . . .                                      | 5         |
| 4.2      | L'équation . . . . .  | 5         |
| <b>5</b> | <b>Discrétisation du problème</b>   | <b>8</b>  |
| 5.1      | Problème de contrôles feedback de chaînes de Markov . . . . .                   | 8         |
| 5.2      | Discrétisation en temps . . . . .   | 12        |
| 5.3      | Différences finies et discrétisation en espace . . . . .                        | 14        |
| <b>6</b> | <b>Les algorithmes de résolution</b>  | <b>15</b> |
| 6.1      | L' itération sur les valeurs . . . . .  | 16        |
| 6.2      | L'algorithme d'Howard . . . . .   | 16        |
| <b>7</b> | <b>Description des étapes de calculs dans la résolution du problème initial</b> | <b>17</b> |
| 7.1      | Calcul de la politique optimale de consommation . . . . .                       | 18        |
| 7.2      | Simulation d'une trajectoire . . . . .  | 19        |

# 1 Notations

Voici les notations que nous utiliserons dans ce mémoire.

- $:=$  désignera une définition,
- $E_\alpha[.]$  sera l'espérance de . sachant que le processus vaut  $\alpha$  en 0,
- $x_+ := x$  si  $x$  est positif, 0 sinon,
- $x_- := -x$  si  $x$  est négatif, 0 sinon,
- 1, suivant le contexte, pourra être le vecteur colonne avec des 1 partout

# 2 Présentation du problème

Nous cherchons une méthode informatique afin d'illustrer les solutions du problème de contrôle suivant <sup>1</sup> :

- $E_t$  est l'état de l'environnement,  $E_0$  l'état initial,
- $C_t$  est la consommation.

Le problème consiste à trouver une consommation (un contrôle) en boucle fermée (feedback)

$C : E_t \mapsto C_t = C(E_t)$ , c'est-à-dire une fonction dépendante de l'état courant de l'environnement, qui vérifie :

$$dE_t = [f(E_t) - \varepsilon C_t]dt + \sigma E_t dW_t \quad (1)$$

et  $C = (C_t)_{\{t > 0\}}$  maximise

$$J(\alpha, C) = E_{E_0=\alpha} \left[ \int_0^\infty e^{-\rho t} U(E_t, C_t) dt \right]$$

avec

- $\varepsilon > 0$ ,
- $\sigma > 0$ ,
- $f$  telle que  $f'' \leq 0$ ,
- $dW$  un incrément du processus de Wiener,
- $U(., .)$  la fonction d'utilité ou de "bien être social",
- $\rho > 0$ .

Notons  $J(\alpha) := \max_{(C)} E_{E_0=\alpha} \left[ \int_0^\infty e^{-\rho t} U(E_t, C_t) dt \right]$  la fonction valeur.

---

1. Morgane Chevé et Ronan Congar, Uncertainty in Natural Regeneration and the Robust Management of Environmental Resources <http://www.univ-rouen.fr/droit-sceco-gestion/Recherche/CARE/docs/papiers/S20071213.pdf>



### 3 Démarche et but

Le travail de Morgane Chevé et de Ronan Congar nous donne une solution analytique pour  $U(E_t, C_t) = \ln E_t + \ln C_t$  et  $f(E) = E \ln(\frac{E_{max}}{E})$ .

Nous allons donc nous concentrer sur la recherche d'une méthode numérique permettant de calculer le contrôle optimal en boucle fermée ainsi que la fonction valeur. Une fois l'algorithme implémenté en C++, nous pourrons nous appliquer à comparer les résultats informatiques aux résultats théoriques, à simuler une trajectoire de  $E_t$  sous ce contrôle, et à la visualiser à l'aide de Gnuplot.

### 4 Mise en place de l'équation de Programmation Dynamique

**Remarque préliminaire :** le contrôle stochastique étant largement utilisé en économie, la fonction valeur est souvent un coût que l'on cherche à minimiser.

La littérature traite en effet très souvent de la fonction valeur :

$$\min_{(C)} E_{E_0} \left[ \int_0^\infty U(E_t, C_t) dt \right]$$

Ce n'est pas un problème puisque l'on sait que  $\max(A) = -\min(-A)$ . Nous cherchons

$$\begin{aligned} J(E_0) &= \max_{(C)} E_{E_0} \left[ \int_0^\infty e^{-\rho t} U(E_t, C_t) dt \right] \\ &= -\min_{(C)} \left\{ -E_{E_0} \left[ \int_0^\infty e^{-\rho t} U(E_t, C_t) dt \right] \right\} \\ &= -\min_{(C)} \left\{ E_{E_0} \left[ \int_0^\infty e^{-\rho t} (-U(E_t, C_t)) dt \right] \right\} \end{aligned}$$

par linéarité de l'espérance et de l'intégrale.

Il suffit alors de considérer  $V := -U$ , et de chercher

$$K(\alpha) = \min_{(C)} E_\alpha \left[ \int_0^\infty V(E_t, C_t) dt \right]$$

Ainsi,  $J = -K$  et la commande trouvée est optimale.

## 4.1 Introduction de la formule d'Itô

Une démonstration de cette formule se trouve dans [2].

Soit  $f \in C^1(R)$  et soit  $(E_t)_{(t \geq 0)}$  vérifiant  $E_t = E_0 + \int_0^t g(s)ds$ . Si  $g$  est continue,  $E_t$  est  $C^1$  et  $\frac{d}{dt}(E_t) = g(t)$ , donc  $\frac{d}{dt}(f(E_t)) = f'(E_t)g(t)$  et ainsi  $f(E_t) - f(E_0) = \int_0^t f'(E_s)g(s)ds$ , ce qui se note

$$df(E_t) = f'(E_t)dE_t = f'(E_t)g(t)dt$$

Considérons un processus défini par  $E_t = E_0 + \int_0^t g(s)ds + \int_0^t \sigma(s)dW_s$  où  $g$  et  $\sigma$  sont assez régulières ([1] propose : mesurables et bornées sur  $[0, T]$   $\forall T > 0$ ) : ici, la formule d'Itô affirme que, pour  $f \in C^2(R)$

$$f(E_t) - f(E_0) = \int_0^t f'(E_s)g(s)ds + \int_0^t f'(E_s)\sigma(s)dW_s + \frac{1}{2} \int_0^t f''(E_s)\sigma^2(s)ds$$

ce qui se note

$$df(E_t) = [f'(E_t)g(t) + \frac{1}{2}f''(E_t)\sigma^2(t)]dt + f'(E_t)\sigma(t)dW_t$$

## 4.2 L'équation

Elle concernera la fonction valeur :

$$K(E_0) = \min_{(C)} E_{E_0} \left[ \int_0^\infty e^{-\rho t} V(E_t, C_t) dt \right]$$

Nous souhaitons montrer que  $K$  vérifie :

$$\rho K(E_0) = \min_\alpha [V(E_0, \alpha) + K'(E_0)[f(E_0) - \varepsilon \alpha] + \frac{1}{2} K''(E_0) \sigma^2 E_0^2]$$

Soit  $\Delta > 0$ , et supposons qu'il existe une consommation optimale en boucle fermée  $C^1(E)$ . Nous avons

$$\begin{aligned} K(E_0) &= \min_{(C)} E_{E_0} \left[ \int_0^\infty e^{-\rho t} V(E_t, C_t) dt \right] = E_{E_0} \left[ \int_0^\infty e^{-\rho t} V(E_t, C^1(E_t)) dt \right] \\ &= E_{E_0} \left[ \int_0^\Delta e^{-\rho t} V(E_t, C^1(E_t)) dt \right] + E_{E_0} \left[ \int_\Delta^\infty e^{-\rho t} V(E_t, C^1(E_t)) dt \right] \quad (2) \end{aligned}$$

Or,

$$E_{E_0} \left[ \int_\Delta^\infty e^{-\rho t} V(E_t, C^1(E_t)) dt \right] = e^{-\rho \Delta} E_{E_0} \left[ \int_\Delta^\infty e^{-\rho(t-\Delta)} V(E_t, C^1(E_t)) dt \right]$$

par linéarité de l'intégrale et de l'espérance,  
et par le changement de variable  $t - \Delta = s$ ,

$$= e^{-\rho\Delta} E_{E_0} \left[ \int_0^\infty e^{-\rho s} V(E_{s+\Delta}, C^1(E_{s+\Delta})) ds \right]$$

et comme  $E_t$  est alors un processus Markovien (diffusion), par la propriété de Markov il vient que,

$$\begin{aligned} &= e^{-\rho\Delta} E_{E(0)=E_0} [E_{E'(0)=E_\Delta} \left[ \int_0^\infty e^{-\rho s} V(E'_s, C^1(E'_s)) ds \right]] \\ &= e^{-\rho\Delta} E_{E(0)=E_0} [K(E_\Delta)] \end{aligned}$$

En effet, l'espérance d'une fonction de la trajectoire  $E_t$  sur  $[\Delta, +\infty]$  générée par (1) avec la condition initiale  $E_0$ , est égale à l'espérance de la fonction de la trajectoire  $E'_t$  sur  $[0, +\infty]$  définie suivant (1) et  $E'_0 = E_\Delta$  ( $E_\Delta$  est aléatoire tout comme  $E_t$ ).

Ainsi (2) devient,

$$\begin{aligned} K(E_0) &= E_{E_0} \left[ \int_0^\Delta e^{-\rho t} V(E_t, C^1(E_t)) dt \right] + e^{-\rho\Delta} E_{E_0} [K(E_\Delta)] \\ &= E_{E_0} \left[ \int_0^\Delta e^{-\rho t} V(E_t, C^1(E_t)) dt \right] + e^{-\rho\Delta} K(E_\Delta) \end{aligned}$$

et nous obtenons,

$$\begin{aligned} 0 &= \frac{1}{\Delta} E_{E_0} \left[ \int_0^\Delta e^{-\rho t} V(E_t, C^1(E_t)) dt \right] + e^{-\rho\Delta} K(E_\Delta) - K(E_0) \\ &= E_{E_0} \left[ \frac{1}{\Delta} (K(E_\Delta) - K(E_0)) \right] + E_{E_0} \left[ \frac{1}{\Delta} \int_0^\Delta e^{-\rho t} V(E_t, C^1(E_t)) dt \right] + E_{E_0} \left[ \frac{e^{-\rho\Delta} - 1}{\Delta} K(E_\Delta) \right] \end{aligned}$$

déjà, si  $K$  est continue, nous avons

$$\frac{e^{-\rho\Delta} - 1}{\Delta} K(E_\Delta) \longrightarrow_{\Delta \rightarrow 0} -\rho K(E_0)$$

et si  $V(E_t, C^1(E_t))$  est continue,

$$\frac{1}{\Delta} \int_0^\Delta e^{-\rho t} V(E_t, C^1(E_t)) dt \longrightarrow_{\Delta \rightarrow 0} V(E_0, C^1(E_0))$$

Ensuite, la formule d'Itô s'écrit ici, en supposant que  $K$  est assez régulière ( $\in C^2(R)$  et bornée par exemple),  $K(E_\Delta) - K(E_0)$

$$= \int_0^\Delta K'(E_s)[f(E_s) - \varepsilon C^1(E_s)]ds + \int_0^\Delta K'(E_s)\sigma E_s dW_s + \frac{1}{2} \int_0^\Delta K''(E_s)\sigma^2 E_s^2 ds$$

avec

$$\frac{1}{\Delta} \int_0^\Delta K'(E_s)[f(E_s) - \varepsilon C^1(E_s)]ds \longrightarrow_{\Delta \rightarrow 0} K'(E_0)[f(E_0) - \varepsilon C^1(E_0)]$$

$$\frac{1}{2\Delta} \int_0^\Delta K''(E_s)\sigma^2 E_s^2 ds \longrightarrow_{\Delta \rightarrow 0} \frac{1}{2} K''(E_0)\sigma^2 E_0^2$$

si les intégrands sont continus, et l'intégrale stochastique est d'espérance nulle :

$$E_{E_0}[\int_0^\Delta K'(E_s)\sigma E_s dW_s] = 0$$

Nous pouvons donc écrire

$$\begin{aligned} \rho K(E_0) &= V(E_0, C^1(E_0)) + K'(E_0)[f(E_0) - \varepsilon C^1(E_0)] \\ &\quad + \frac{1}{2} K''(E_0)\sigma^2 E_0^2 \end{aligned} \quad (3)$$

Maintenant si on remplace la consommation précédente par la consommation  $C_t^2$  égale à  $\alpha$  sur  $[0, \Delta[$  et à la consommation optimale feedback  $C^1$  sur  $[\Delta, +\infty[$ , on obtient par optimalité :

$$\begin{aligned} K(E_0) &= \min_{(C)} E_{E_0}[\int_0^\infty e^{-\rho t} V(E_t, C_t) dt] \\ &\leq E_{E_0}[\int_0^\infty e^{-\rho t} V(E_t, C^2(E_t)) dt] = E_{E_0}[\int_0^\Delta e^{-\rho t} V(E_t, \alpha) dt] + E_{E_0}[\int_\Delta^{+\infty} e^{-\rho t} V(E_t, C^1(E_t)) dt] \end{aligned}$$

par les mêmes raisons que précédemment, le terme de droite devient  $e^{-\rho\Delta} E_{E_0}[K(E_\Delta)]$ .

Nous obtenons

$$0 \leq E_{E_0}[\frac{1}{\Delta} \int_0^\Delta e^{-\rho t} V(E_t, \alpha) dt] + E_{E_0}[\frac{e^{-\rho\Delta} - 1}{\Delta} K(E_\Delta)] + E_{E_0}[\frac{1}{\Delta} [K(E_\Delta) - K(E_0)]]$$

avec

$$\begin{aligned} \frac{1}{\Delta} \int_0^\Delta e^{-\rho t} V(E_t, \alpha) dt &\longrightarrow_{\Delta \rightarrow 0} V(E_0, \alpha) \\ \frac{e^{-\rho\Delta} - 1}{\Delta} K(E_\Delta) &\longrightarrow_{\Delta \rightarrow 0} -\rho K(E_0) \end{aligned}$$

(si  $K$  et l'intégrand sont continus). Ici, comme le processus état  $E_t$  était sous la consommation  $\alpha$  de 0 à  $\Delta$ , la formule d'Ito nous donne  $K(E_\Delta) - K(E_0)$

$$= \int_0^\Delta K'(E_s)[f(E_s) - \varepsilon\alpha]ds + \int_0^\Delta K'(E_s)\sigma E_s dW_s + \frac{1}{2} \int_0^\Delta K''(E_s)\sigma^2 E_s^2 ds$$

et pour les mêmes raisons que lorsque la consommation était optimale, nous obtenons  $\forall \alpha \in$  l'espace d'arrivée des contrôles,

$$\begin{aligned} \rho K(E_0) &\leq V(E_0, \alpha) + K'(E_0)[f(E_0) - \varepsilon\alpha] \\ &\quad + \frac{1}{2} K''(E_0)\sigma^2 E_0^2 \end{aligned} \quad (4)$$

En utilisant 3 et 4 on trouve,

$$\rho K(E_0) = \min_\alpha [V(E_0, \alpha) + K'(E_0)[f(E_0) - \varepsilon\alpha] + \frac{1}{2} K''(E_0)\sigma^2 E_0^2] \quad (5)$$

que l'on appelle équation de programmation dynamique du système.

## 5 Discrétisation du problème

Nous nous aidons du travail de Frédéric Bonnans<sup>2</sup> dans cette section ainsi que dans la suivante.

### 5.1 Problème de contrôles feedback de chaînes de Markov

**Théorème 5.1** *Nous supposons que  $\{e_n\}_{(0 \leq n)}$  est une chaîne de Markov d'espace d'états  $\{\omega_1, \dots, \omega_m\}$  commandée par  $c$  (à choisir dans  $C_i$  lorsque la chaîne est dans l'état  $\omega_i$ ) et de matrice de transition  $M(c)$ .  $v(.,.)$  la fonction d'utilité est bornée.*

1) la fonction valeur définie par  $\forall i \in \{1, \dots, m\}$

$$k(\omega_i) := k_i := \inf_c \{E_{e_0=\omega_i} [\sum_{t=0}^{\infty} \beta^{t+1} v(e_t, c(e_t))]\}$$

où  $\beta \in ]0, 1[$ ,

---

2. "Quelques Aspects Numeriques de la Commande Optimale Stochastique" [www.cmap.polytechnique.fr/~bonnans/notes/cos.pdf](http://www.cmap.polytechnique.fr/~bonnans/notes/cos.pdf)

satisfait

$$\|k\|_\infty \leq \frac{\beta}{1-\beta} \|v\|_\infty \quad (6)$$

Tout comme  $k_i = k(\omega_i)$ , nous avons noté ici  $v_i(c) := v(\omega_i, c(\omega_i))$ ,  $\|v\|_\infty = \sup_{c,i} \{v_i(c)\}$  et  $c_i = c(\omega_i)$ .

De plus, c'est la solution unique de l'équation de programmation dynamique suivante : trouver  $k \in R^m$  tel que  $\forall i \in \{1, \dots, m\}$

$$k_i = \beta \inf_{c_i \in C_i} \{v(\omega_i, c_i) + \sum_{j=1}^m M_{ij}(c_i) k_j\} \quad (7)$$

2) Soit  $\varepsilon \geq 0$  et  $c$  une consommation (contrôle, commande ou stratégie) telle que  $\forall i \in \{1, \dots, m\}$  :

$$\beta(v(\omega_i, c_i) + \sum_j M_{ij}(c_i) k_j) \leq k_i + \varepsilon \quad (8)$$

Posons  $\varepsilon' := (1-\beta)^{-1}\varepsilon$ . Alors la consommation  $c$  est  $\varepsilon'$  sous optimale, dans le sens où la valeur associée  $k(c)$  satisfait  $\forall i \in \{1, \dots, m\}$

$$k_i(c) \leq k_i + \varepsilon' \quad (9)$$

3) L'existence d'une stratégie optimale est assurée si  $C_i$  est métrique compact pour tout  $i$ , et les fonction  $v_i(c)$  et  $M_{ij}(c)$  sont continues pour tout  $i$  et  $j$ .

**Démonstration** Seuls les points 1) et 2) seront démontrés.

Soient  $c$  une consommation en feedback, et  $i \in \{1, \dots, m\}$ . Montrons tout d'abord que la fonction valeur associée à cette consommation vérifie l'équation  $k(c) = \beta(v(c) + M(c)k(c))$  :

$$\begin{aligned} k_i(c) &= \beta v(\omega_i, c(\omega_i)) + E_{e_0=\omega_i} \left[ \sum_{t=1}^{\infty} \beta^{t+1} v(e_t, c(e_t)) \right] \\ &= \beta v(\omega_i, c(\omega_i)) + \beta E_{e_0=\omega_i} \left[ \sum_{t=1}^{\infty} \beta^t v(e_t, c(e_t)) \right] \end{aligned}$$

changement de variable  $s = t - 1$

$$= \beta v(\omega_i, c(\omega_i)) + \beta E_{e_0=\omega_i} \left[ \sum_{s=0}^{\infty} \beta^{s+1} v(e_{s+1}, c(e_{s+1})) \right]$$

par le théorème de Markov et par définition de  $k(c)$ ,

$$\begin{aligned}
&= \beta v(\omega_i, c(\omega_i)) + \beta E_{e_0=\omega_i} [E_{e'_0=e_1} [\sum_{s=0}^{\infty} \beta^{s+1} v(e'_s, c(e'_s))]] \\
&= \beta v(\omega_i, c(\omega_i)) + \beta E_{e_0=\omega_i} [k_{\{j|e_1=\omega_j\}}(c)] \\
&= \beta v_i(c) + \beta \sum_{j=1}^m M_{ij}(c) k_j(c)
\end{aligned}$$

soit le résultat voulu.

Montrons maintenant que (7) possède une solution unique. Cette équation est de la forme  $k = Tk$ , avec

$$(Tk)_i := \beta \inf_{c_i \in C_i} \{v(\omega_i, c_i) + \sum_{j=1}^m M_{ij}(c_i) k_j\}$$

Montrons que  $T$  est un opérateur contractant dans  $(R^m, \|\cdot\|_{\infty})$  : notons  $v(\omega_i, c_i) := v_i(c_i)$  (c'est alors un vecteur de  $R^m$  fonction de la consommation).  $\forall c$  vecteur consommation de  $R^m$ ,  $\|Tk\|_{\infty} \leq \beta(\|v(c)\|_{\infty} + \|Mk\|_{\infty})$ , et comme  $M(c)$  est stochastique  $\forall c$  (ie  $\forall i \in \{1, \dots, m\}$ ,  $\sum_j M_{ij} = 1$ , et  $\forall (i, j) \in \{1, \dots, m\}^2$ ,  $M_{ij} \geq 0$ ), la norme subordonnée  $\|M\|_{\infty} \leq 1$ . En effet, soit  $k$  tel que  $\|k\|_{\infty} \leq 1 : \forall i \in \{1, \dots, m\}$ ,  $|(Mk)_i| \leq \sum_j M_{ij} |k_j| \leq \sum_j M_{ij} = 1$ . Ainsi, nous avons  $\forall c$  vecteur consommation de  $R^m$ ,  $\|Tk\|_{\infty} \leq \beta(\|v(c)\|_{\infty} + \|k\|_{\infty})$  et comme  $\sup_c \{\|v(c)\|_{\infty}\} < \infty$  ( $v$  est bornée),

$$\|Tk\|_{\infty} \leq \beta(\|v\|_{\infty} + \|k\|_{\infty}) \quad (10)$$

En utilisant le fait que  $|\inf_c \{a(c)\} - \inf_c \{b(c)\}| \leq \sup_c \{|a(c) - b(c)|\}$  ( $a(c) - b(c) \leq \sup_c \{|a(c) - b(c)|\}$ , puis  $a(c) \leq \sup_c \{|a(c) - b(c)|\} + b(c)$  et on prend l'inf de chaque coté et on a alors  $\inf_c \{a(c)\} - \inf_c \{b(c)\} \leq \sup_c \{|a(c) - b(c)|\}$  et si on inverse le rôle de  $a(\cdot)$  et  $b(\cdot)$ , nous obtenons cette règle), nous pouvons écrire

$$\begin{aligned}
|(Tk')_i - (Tk)_i| &:= \beta |\inf_{c_i \in C_i} \{ \sum_{j \in \{1, \dots, m\}} M_{ij}(c_i) k'_j \} - \inf_{c_i \in C_i} \{ \sum_{j \in \{1, \dots, m\}} M_{ij}(c_i) k_j \} | \\
&\leq \beta \sup_{c_i \in C_i} \{ | \sum_{j \in \{1, \dots, m\}} M_{ij}(c_i) (k'_j - k_j) | \}
\end{aligned}$$

$$\begin{aligned} &\leq \beta \sup_{c_i \in C_i} \left\{ \sum_{j=1}^m M_{ij}(c_i) |k'_j - k_j| \right\} \\ &\leq \beta \|k' - k\|_\infty \end{aligned}$$

De ce fait,  $T$  est une contraction de rapport  $\beta$  dans  $(R^m, \|\cdot\|_\infty)$  ( $\|T(k' - k)\|_\infty \leq \beta \|k' - k\|_\infty$ ). Il découle alors du théorème du point fixe de Banach-Picard (nous sommes dans le Banach  $(R^m, \|\cdot\|_\infty)$ ) que (7) admet une unique solution  $k^*$  vérifiant  $Tk^* = k^*$ .

Nous obtenons alors aussi par (10),

$$\|k^*\|_\infty \leq \beta(\|v\|_\infty + \|k^*\|_\infty)$$

ce qui donne (6) si la fonction valeur  $k$  est bien la solution de l'équation de programmation dynamique (ie si on a prouvé le "de plus" de 1)).

Maintenant, soient  $k^*$  la solution unique de (6),  $c$  une consommation et  $k(c)$  la valeur associée (on a vu précédemment que  $k(c)$  vérifiait  $k(c) = \beta(v(c) + M(c)k(c))$ ).

Montrons que  $k^* \leq k(c)$ . Les inégalités signifient maintenant  $\forall i \in \{1, \dots, m\}$ ,  $(\cdot)_i \leq (\cdot)_i$ . Utilisant

$$k^* \leq \beta(v(c) + M(c)k^*)$$

(car  $k^*$  est l'inf), il vient

$$k^* - k(c) \leq \beta M(c)(k^* - k(c))$$

et comme  $M$  est stochastique,

$$M(c)(k^* - k(c)) \leq \sup_i \{(k^* - k(c))_i\} 1$$

et donc

$$\begin{aligned} k^* - k(c) &\leq \beta \sup_i \{(k^* - k(c))_i\} 1 \\ \sup_i \{(k^* - k(c))_i\} 1 &\leq \beta \sup_i \{(k^* - k(c))_i\} 1 \end{aligned}$$

et comme  $0 < \beta < 1$ , on a  $\sup_i \{(k^* - k(c))_i\} \leq 0$ , soit le résultat cherché. Ainsi, toute fonction valeur associée à un contrôle feedback est supérieure à  $k^*$ .

Etant donné  $\varepsilon$ , soit  $c$  une stratégie satisfaisant (8) avec  $k^*$  ( $\beta(v(c) + M(c)k^*) \leq k^* + \varepsilon 1$ ) (8'); une telle stratégie existe si  $\varepsilon > 0$  par définition de l'inf, et l'existence de la solution de (7) et  $k(c)$  une consommation feedback (elle vérifie donc  $k(c) = \beta(v(c) + M(c)k(c))$ ).



Par (8'), il vient

$$k(c) - k^* \leq \varepsilon 1 + \beta M(c)(k(c) - k^*)$$

On en déduit (9') car  $M(c)$  est stochastique et alors

$$\begin{aligned} k(c) - k^* &\leq \varepsilon 1 + \beta \sup_i \{(k(c) - k^*)_i\} 1 \\ \sup_i \{(k(c) - k^*)_i\} 1 &\leq \varepsilon 1 + \beta \sup_i \{(k(c) - k^*)_i\} 1 \\ \sup_i \{(k(c) - k^*)_i\} 1 &\leq \varepsilon (1 - \beta)^{-1} 1 := \varepsilon' 1 \end{aligned}$$

et donc,

$$k(c) \leq k^* + \varepsilon' 1$$

soit (9') ie (9) avec  $k^*$ ). Ainsi, nous pouvons approcher autant que nous voulons  $k^*$  avec des  $k(c)$ , et  $\forall$  la consommation  $c$ ,  $k^* \leq k(c)$ . Il en résulte que  $k^* = k$  ie  $k^*$  est bien la fonction valeur par définition, d'où 1) et 2).

## 5.2 Discrétisation en temps

Il est proposé dans le travail de Frédéric Bonnans de considérer le problème de commande optimale stochastique suivant avec  $h$  la discrétisation en temps :

- $K(E_0) = \min_C E_{E_0} [h \sum_{k=0}^{\infty} (1 + \rho h)^{-k-1} V(E_k, C_k)]$
- $E_{k+1} = E_k + h(f(E_k) - \varepsilon C_k) + \sqrt{h} \sigma E_k \delta w_k$
- $E_0$  donné.

Ici  $\delta w_k$  est une variable aléatoire de loi uniforme sur  $\{-1, 1\}$ , donc de moyenne nulle et de variance unité. Par ailleurs, si  $0 \leq s < t < \infty$ ,  $s = k_0 h$  et  $t = k_1 h$ , alors  $\sqrt{h} \sum_{k_0}^{k_1-1} \delta w_k$  est une variable asymptotiquement gaussienne (quand le pas de la discrétisation  $h$  tend vers 0), de moyenne nulle et de variance  $t - s$  (par le théorème de la limite centrale), ce qui est cohérent avec le problème continu. En effet, en posant  $h = \frac{t-s}{n}$  et  $k_1 - k_0 = n$ , c'est la somme de  $n$  variables aléatoires iid d'espérance nulle et de variance 1, multipliée par  $\sqrt{\frac{t-s}{n}}$ .

L'équation de programmation dynamique de ce problème est :

$$\begin{aligned} &(1 + \rho h)K(E) \\ &= \min_{\alpha} \{hV(E, \alpha) + E_E [K(E + h(f(E) - \varepsilon \alpha) + \sqrt{h} \sigma E \delta w)]\} \end{aligned} \quad (11)$$

En effet, comme dans le cas continu, soit  $C$  une consommation optimale,  $E_0$  l'état initial,

$$K(E_0) = h(1 + \rho h)^{-1}V(E_0, C_0) + E_{E_0}[h \sum_{k=1}^{\infty} (1 + \rho h)^{-k-1}V(E_k, C(E_k))]$$

on pose  $t = k - 1$ ,

$$\begin{aligned} &= h(1 + \rho h)^{-1}V(E_0, C_0) + E_{E_0}[h(1 + \rho h)^{-1} \sum_{t=0}^{\infty} (1 + \rho h)^{-t-1}V(E_{t+1}, C(E_{t+1}))] \\ &= h(1 + \rho h)^{-1}\{V(E_0, C_0) + E_{E_0}[E_{(E'_0=E_1)}[\sum_{t=0}^{\infty} (1 + \rho h)^{-t-1}V(E'_t, C(E'_t))]]\} \end{aligned}$$

puisque c'est une chaine de Markov, et par définition de  $K$  puis de la chaine,

$$\begin{aligned} &= h(1 + \rho h)^{-1}\{V(E_0, C_0) + E_{E_0}[K(E_1)]\} \\ &= h(1 + \rho h)^{-1}\{V(E_0, C_0) + E_{E_0}[K(E_0 + h(f(E_0) - \varepsilon C_0) + \sqrt{h}\sigma E_0 \delta w_0)]\} \end{aligned}$$

Maintenant, en prenant la consommation  $C'$  valant n'importe qu'elle valeur au temps 0, et valant  $C$  ensuite, nous obtenons de la même manière l'inégalité

$$K(E_0) \leq h(1 + \rho h)^{-1}\{V(E_0, C_0) + E_{E_0}[K(E_0 + h(f(E_0) - \varepsilon C_0) + \sqrt{h}\sigma E_0 \delta w_0)]\}$$

et cela prouve la validité de l'équation de programmation dynamique.

Si  $K$  est  $C^2$  de dérivée seconde uniformément bornée sur  $R$ , Ito nous donne :

$$\begin{aligned} &K(E + h(f(E) - \varepsilon \alpha) + \sqrt{h}\sigma E \delta w) \\ &= K(E) + hK'(E)(f(E) - \varepsilon \alpha) + \sqrt{h}K'(E)\sigma E \delta w + \frac{1}{2}hK(E)\sigma^2 E^2 \delta w^2 + o(h) \end{aligned}$$

ce qui donne en espérance :

$$E[K(E + h(f(E) - \varepsilon \alpha) + \sqrt{h}\sigma E \delta w)] = K(E) + hK'(E)(f(E) - \varepsilon C) + \frac{1}{2}hK(E)\sigma^2 E^2 + o(h)$$

En remplaçant dans l'équation (6), on obtient en divisant par  $h$  :

$$\rho K(E) = \min_{\alpha} \{V(E, C) + K'(E)(f(E) - \varepsilon \alpha) + \frac{1}{2}K''(E)\sigma^2 E^2 + o(1)\}$$

ce qui est encore cohérent avec le problème initial.

L'état du système ne vit pas dans un espace fini, ce qui rend la résolution de l'équation de HJB difficile. La sous-section suivante utilise directement l'équation en temps continu afin de trouver une équation sur un espace fini.

Cependant, la définition récurrente de la chaîne permettra de simuler une approximation du processus une fois la consommation optimale trouvée (voir section 7).

### 5.3 Différences finies et discrétisation en espace

Notons  $h$  le pas de la discrétisation en temps, et maintenant  $h_1$  celui de la discrétisation en espace. Un schéma de résolution numérique de l'équation numérique de l'équation de programmation dynamique (ou de Hamilton-Jacobi-Bellman (5)) peut être :

$$\begin{aligned} \rho k_i = \min_C \{ & V(\omega_i, C) \\ & + (f(\omega_i) - \varepsilon C)_+ \frac{k_{i+1} - k_i}{h_1} + (f(\omega_i) - \varepsilon C)_- \frac{k_{i-1} - k_i}{h_1} \\ & + \frac{\sigma^2 \omega_i^2}{2} \frac{k_{i+1} - 2k_i + k_{i-1}}{h_1^2} \} \end{aligned}$$

(approximation des dérivées dans (5) par des taux d'accroissements)

avec :

- $\omega_i$  la condition initiale  $\in E^{h_1} := [-E_{max}, E_{max}] \cap h_1.Z$ , l'espace discrétisé,
- $k_i = K(\omega_i)$  la fonction valeur vecteur de  $R^{|E^{h_1}|}$ ,
- et  $C$  la consommation dans  $[-C_{max}, C_{max}]$

On regroupe les  $k_i$  :

$$\begin{aligned} \rho k_i = \min_C \{ & V(\omega_i, C) - k_i \left( \frac{|f(\omega_i) - \varepsilon C|}{h_1} + \frac{\sigma^2 \omega_i^2}{h_1^2} \right) \\ & + k_{i+1} \left( \frac{(f(\omega_i) - \varepsilon C)_+}{h_1} + \frac{\sigma^2 \omega_i^2}{2h_1^2} \right) + k_{i-1} \left( \frac{(f(\omega_i) - \varepsilon C)_-}{h_1} + \frac{\sigma^2 \omega_i^2}{2h_1^2} \right) \} \end{aligned}$$

On multiplie par  $h > 0$ , le pas de discrétisation en temps, et on ajoute  $k_i$  à chaque membre :

$$k_i = (h\rho + 1)^{-1} \min_C \{ hV(\omega_i, C) + k_i \left( 1 - \frac{h|f(\omega_i) - \varepsilon C|}{h_1} - \frac{\sigma^2 \omega_i^2}{h_1^2} \right) \}$$

$$+k_{i+1}(\frac{h(f(\omega_i) - \varepsilon C)_+}{h_1} + \frac{h\sigma^2\omega_i^2}{2h_1^2}) + k_{i-1}(\frac{h(f(\omega_i) - \varepsilon C)_-}{h_1} + \frac{h\sigma^2\omega_i^2}{2h_1^2})\} \quad (12)$$

Supposons que

$$\frac{h}{h_1} \sup_{(i \in \{1, \dots, |E^{h_1}|\}, C \in [-C_{max}, C_{max}])} \{|f(\omega_i) - \varepsilon C|\} + \frac{h}{h_1^2} \sigma^2 E_{max}^2 \leq 1 \quad (13)$$

La condition (13) assure que, dans la formule (12), le poids de  $k_i$  est positif. De plus, la somme de ces poids vaut 1.

Nous pouvons donc interpréter l'équation (12) comme l'équation de programmation dynamique d'un problème de contrôle de chaîne de Markov (voir section 5.1) avec ici

- $\beta = (1 + \rho h)^{-1}$ ,
  - l'espace d'état  $E^{h_1}$ ,
  - l'espace des consommations  $C_i$  fixé  $\forall i = [-C_{max}, C_{max}]$ ,
  - la fonction d'utilité  $v = hV$ ,
- et  $\forall i \in \{1, \dots, |E^{h_1}|\}$ ,

$$\begin{aligned} - M_{ii}(c) &= 1 - \frac{h|f(\omega_i) - \varepsilon c|}{h_1} - \frac{h\sigma^2\omega_i^2}{h_1^2} \\ - M_{i,i+1}(c) &= \frac{h(f(\omega_i) - \varepsilon c)_+}{h_1} + \frac{h\sigma^2\omega_i^2}{2h_1^2} \\ - M_{i,i-1}(c) &= \frac{h(f(\omega_i) - \varepsilon c)_-}{h_1} + \frac{h\sigma^2\omega_i^2}{2h_1^2} \end{aligned}$$

les autres coefficients de  $M(c)$  étant nuls.

Ainsi, le théorème 5.1 nous affirme que si  $V(.,.)$  est bornée, le schéma (12) possède une solution unique, telle que

$$\|k\|_\infty \leq \frac{(1 + \rho h)^{-1}}{1 - (1 + \rho h)^{-1}} \|hV\|_\infty$$

i.e.,

$$\|k\|_\infty \leq \rho^{-1} \|V\|_\infty$$

et que c'est une équation de point fixe contractant pour la norme uniforme, de rapport de contraction  $(1 + \rho h)^{-1}$ .

## 6 Les algorithmes de résolution

Nous allons ici introduire deux algorithmes permettant de résoudre le schéma numérique (12) afin d'obtenir la fonction valeur, ainsi qu'une consommation optimale. Nous utiliserons les résultats donnés par le théorème

section 5.1. Par ailleurs, nous reprenons les notations du théorème, notre problème ayant été traduit en terme de chaîne de Markov contrôlée.

## 6.1 L'itération sur les valeurs

C'est la méthode la plus simple :  $\forall i \in \{1, \dots, m\}$ , définissons la suite  $(k_i^q)_{(0 \leq q)}$  de la manière suivante

$$k_i^{q+1} = \beta \inf_{c \in C} \{v_i(c) + \sum_j M_{ij}(c) k_j^q\} \quad (14)$$

Nous avons montré dans le théorème que l'opérateur  $T$  défini sur  $(R^m, \|\cdot\|_\infty)$  qui permet de réécrire l'algorithme  $k_i^{q+1} = T k_i^q$ , était contractant de rapport  $\beta$ . Le théorème du point fixe de Banach-Picard nous donne alors

$$\forall k^0 \in l^\infty, \|k^q - k^*\|_\infty \leq \beta^q \|k^0 - k^*\|_\infty$$

où  $k^*$  est le vecteur fonction valeur recherché. La suite tend donc vers le résultat recherché.

## 6.2 L'algorithme d'Howard

Cet algorithme développé dans [3] est aussi appelé algorithme d'itération sur les stratégies. Nous supposons les conditions du troisième point du théorème satisfaites. Chaque itération de l'algorithme comporte deux étapes :

- étant donné une consommation  $c^q \in C$  compact, calculer la valeur  $k^q$  associée, solution de l'équation linéaire  $k^q = \beta(v(c^q) + M(c^q)k^q)$
- calculer  $\forall i \in \{1, \dots, m\}$ ,  $c_i^{q+1}$  solution de
$$c_i^{q+1} \in \operatorname{argmin}_{c \in C} \{v_i(c) + \sum_j M_{ij}(c) k_j^q\}$$

Le système linéaire de la première étape a une solution unique car c'est l'équation de point fixe d'un opérateur contractant ; il suffit d'invoquer le théorème du point fixe. Le minimum dans la deuxième étape est atteint par les conditions du troisième point du théorème.

Les deux étapes impliquent

$$\begin{aligned} \beta^{-1}(k^{q+1} - k^q) &= v(c^{q+1}) + M(c^{q+1})k^{q+1} - v(c^q) - M(c^q)k^q \\ &\leq v(c^{q+1}) + M(c^{q+1})k^{q+1} - v(c^{q+1}) - M(c^{q+1})k^q \end{aligned}$$

$$\begin{aligned}
& (\text{car par définition } v(c^{q+1}) + M(c^{q+1})k^q \leq v(c^q) - M(c^q)k^q \\
& = M(c^{q+1})(k^{q+1} - k^q)
\end{aligned}$$

et comme dans le théorème, cela implique que  $k^{q+1} \leq k^q$ . Donc la suite  $k^q$  est décroissante. Notons  $k'^{q+1}$  la valeur calculée à partir de  $k^q$  par l'itération sur les valeurs (algorithme précédent).

Nous savons que  $\|k'^{q+1} - k^*\|_\infty \leq \beta \|k^q - k^*\|_\infty$  par la section précédente.

Montrons que  $k^{q+1} \leq k'^{q+1}$ , ce qui assurera que

$$k^{q+1} - k^* \leq k'^{q+1} - k^*$$

et comme par définition  $k^* \leq k^{q+1}$ ,

$$\|k^{q+1} - k^*\|_\infty \leq \|k'^{q+1} - k^*\|_\infty$$

et donc

$$\|k^{q+1} - k^*\|_\infty \leq \beta \|k^q - k^*\|_\infty$$

ce qui prouve la validité de l'algorithme.

$$\begin{aligned}
\beta^{-1}(k^{q+1} - k'^q) &= v(c^{q+1}) + M(c^{q+1})k^{q+1} - v(c^{q+1}) - M(c^{q+1})k^q \\
&= M(c^{q+1})(k^{q+1} - k^q) \leq 0
\end{aligned}$$

car  $k^{q+1} \leq k^q$  (la suite est décroissante).

Nous avons bien montré que  $k^{q+1} \leq k'^{q+1}$ .

## 7 Description des étapes de calculs dans la résolution du problème initial

Nous décrivons maintenant les étapes de calculs permettant de calculer la fonction valeur, la consommation optimale, et une trajectoire sous cette consommation.

Données INPUT :

- $h > 0$ ,  $h_1 > 0$ ,  $h_2 > 0$  les pas de discrétisation (en temps, état, et consommation)
- $f$
- $\varepsilon > 0$
- $\sigma > 0$

- $U(.,.)$  la fonction d'utilité ou de "bien être social"
- $\rho > 0$
- $T_{max}, E_{max}, C_{max},$
- $E_0 \in E^{h_1} := [-E_{max}, E_{max}] \cap h_1.Z$

OUTPUT : en sortie,

- la fonction valeur  $J(E_0)$
- la consommation  $C(E)$  (en feedback ie vecteur de taille  $|E^{h_1}|$ )
- une trajectoire  $(E_n)_{(0 \leq n)}$  commandée par  $C$

Nous commençons par créer la grille d'état  $E^{h_1}$  et de consommation  $C^{h_2} := [-C_{max}, C_{max}] \cap h_2.Z$ . Ensuite déclarons  $V = -U$  comme expliqué section 4.

Les calculs matriciels sont assurés par la création d'un objet matrice avec surcharge d'opérateurs. La résolution des équations linéaire se fait à l'aide de l'algorithme de Thomas<sup>3</sup>, algorithme rapide réservé aux matrices tridiagonales ( $M(c)$  est bien tridiagonale, voir plus haut)

## 7.1 Calcul de la politique optimale de consommation

Le nombre d'itérations des deux algorithmes sera à choisir en pratique pour concilier rapidité du calcul et précision (donnée section 6)

### Avec l'algorithme d'itération sur les valeurs :

$C :=$  vecteur nul de  $(C^{h_2})^{|E^{h_1}|}$

$k :=$  vecteur nul de  $R^{|E^{h_1}|}$

$j := 0$ ;

(boucle1) tant que  $j < n$  :

$l :=$  vecteur  $k$ ;

$k := (1 + \rho h)^{-1}(v(c) + M(c)l)$ , avec  $c$  consommation quelconque,

(boucle2)  $\forall c \in C^{h_2}$  :

calcul de  $M :=$ matrice  $M(c)$ ;

calcul de  $z := (1 + \rho h)^{-1}(v(c) + M(c)l)$ ;

(boucle3)  $\forall 1 \leq i \leq |E^{h_1}|$  :

si  $z_i < k_i$ ,

---

3. Paola GOATIN. Analyse Numérique. <http://www-sop.inria.fr/members/Paola.Goatin/anum.pdf>

$k_i := z_i$  et  $C_i := c$ ;  
 fin de boucle3  
 fin de boucle2  
 $j++$ ;  
 fin de boucle1  
 on retourne  $-k$  ( $= J$ ) et  $C$

#### Avec l'algorithme d'Howard :

$k$  := vecteur nul de  $R^{|E^{h_1}|}$ ;  
 $C$  := vecteur nul de  $(C^{h_2})^{|E^{h_1}|}$   
 $j := 0$ ;  
 (boucle1) tant que  $j < n$  :  
 résolution de  $(Id - \beta M(C))x = \beta v(C)$  par l'algorithme de Thomas  
 ( $(Id - \beta M(C))$  est tridiagonale);  
 $k := x$  (la solution);  
 $w := (v(C) + M(C)k)$ ;  
 (boucle2)  $\forall c \in C^{h_2}$  :  
 calcul de  $z := (v(c) + M(c)k)$ ;  
 (boucle3)  $\forall 1 \leq i \leq |E^{h_1}|$  :  
 si  $z_i < w_i$ ,  
 $w_i := z_i$  et  $C_i := c$ ;  
 fin de boucle3  
 fin de boucle2  
 $j++$ ;  
 fin de boucle1  
 on retourne  $-k$  ( $= J$ ) et  $C$

## 7.2 Simulation d'une trajectoire

A ce stade, nous avons donc une consommation optimale  $C$  et nous utilisons la discrétisation en temps de la section 5.2.

$t := h$ ;  
 $E := E_0$ ;  
 (boucle1) tant que  $t \leq T_{max}$  :  
 choisir de manière équiprobable  $w := -1$  ou  $1$ ;  
 trouver  $i$  tel que  $\omega_i \leq E < \omega_{i+1}$ ;  
 (recherche de l'élément de la grille le plus proche de  $E$ )  
 calcul de  $C := C_i + \frac{C_{i+1} - C_i}{\omega_{i+1} - \omega_i}(E - \omega_i)$ ; (approximation linéaire des points  
 non traités)



calcul et mémorisation de  $E_t := E + h(f(E) - \varepsilon C) + \sqrt{h}\sigma Ew$ ;  
 $E := E_t$ ;  
 $t := t + h$ ;  
fin de boucle1  
retourner la suite  $E_t$

## Références

-[1] Harold J. Kushner et Paul G. Dupuis, Numérical Methods for Stochastic Control Problems in Continuous Time.

-Harold J. Kushner, Probability Methods for Approximations in Stochastic Control and for Elliptic Equations.

-[2] N.V.Krylov, Controlled Diffusion Processes.

-Wendell H. Fleming et H. Mete Soner. Controlled Markov Processes and Viscosity Solutions.

-Wendell H. Fleming et Raymond W. Rishel. Deterministic and Stochastic Optimal Control.

-[3] Howard. Dynamic Probabilistic Systems Volume II : Semimarkov and Decision Processes.

-R. E. Bellman et S. E. Dreyfus. La Programmation Dynamique et ses Applications.

# Approche résumée de la résolution numérique pour comprendre et utiliser les fichiers sources

Modèle Chevé-Congar, Nicolas Drougard

17 octobre 2011

Ce résumé est une première lecture du travail sur le calcul numérique des solutions du modèle Chevé-Congar, décrit dans "Approximation des solutions du modèle Chevé-Congar" [ACC].

## 0.1 Problème étudié

Nous cherchons une méthode informatique afin d'illustrer les solutions du problème de contrôle suivant<sup>1</sup> :

- $E_t$  est l'état de l'environnement sur une échelle  $[0, E_{max}]$ ,  $E_0$  l'état initial,
- $C_t$  est la consommation.

Le problème consiste à trouver une consommation (un contrôle) en boucle fermée (feedback)

$C : E_t \mapsto C_t = C(E_t)$ , c'est-à-dire une fonction dépendante de l'état courant de l'environnement, qui vérifie :

$$dE_t = [f(E_t) - \varepsilon C_t]dt + \sigma E_t dW_t \quad (1)$$

et  $C = (C_t)_{\{t > 0\}}$  maximise

$$J(\alpha, C) = \mathbb{E}_{E_0=\alpha} \left[ \int_0^\infty e^{-\rho t} U(E_t, C_t) dt \right]$$

avec

- $\varepsilon > 0$ ,
- $\sigma > 0$ ,
- $f$  telle que  $f'' \leq 0$ ,
- $dW$  un incrément du processus de Wiener,
- $U(.,.)$  la fonction d'utilité ou de "bien être social",
- $\rho > 0$ .

## 0.2 Etapes principales de l'approximation

### 0.2.1 Calcul de la consommation et de la fonction valeur

Pour rester proche des références utilisées, et grâce à l'égalité  $\max(A) = -\min(-A)$ , nous allons plutôt chercher

$$K(\alpha) = \min_{(C)} \mathbb{E}_\alpha \left[ \int_0^\infty V(E_t, C_t) dt \right]$$

avec  $V := -U$ . La consommation trouvée sera optimale et  $J = -K$  (voir section 4, remarque préliminaire).

L'équation de programmation dynamique de ce problème est (voir section 4 [ACC]),

$$\rho K(E_0) = \min_\alpha [V(E_0, \alpha) + K'(E_0)[f(E_0) - \varepsilon \alpha] + \frac{1}{2} K''(E_0) \sigma^2 E_0^2] \quad (2)$$

Nous discrétisons l'espace d'état  $[0, E_{max}]$  avec un pas de discrétisation en espace  $h_1$  : un état discret  $\omega_i$  appartient à  $E^{h_1} := [0, E_{max}] \cap h_1 \mathbb{N}$ , l'espace discrétisé. Soit  $h$  un paramètre à adapter pour avoir

$$\frac{h}{h_1} \sup_{(i \in \{1, \dots, |E^{h_1}|\}, C \in [-C_{max}, C_{max}])} \{|f(\omega_i) - \varepsilon C|\} + \frac{h}{h_1^2} \sigma^2 E_{max}^2 \leq 1 \quad (3)$$

Nous obtenons alors par un schéma de différences finies de l'équation de programmation dynamique l'équation suivante : (voir section 5.3 [ACC]) :

$$\begin{aligned} k_i &= (h\rho + 1)^{-1} \min_C \left\{ hV(\omega_i, C) + k_i \left( 1 - \frac{h|f(\omega_i) - \varepsilon C|}{h_1} - \frac{\sigma^2 \omega_i^2}{h_1^2} \right) \right. \\ &\quad \left. + k_{i+1} \left( \frac{h(f(\omega_i) - \varepsilon C)_+}{h_1} + \frac{h\sigma^2 \omega_i^2}{2h_1^2} \right) + k_{i-1} \left( \frac{h(f(\omega_i) - \varepsilon C)_-}{h_1} + \frac{h\sigma^2 \omega_i^2}{2h_1^2} \right) \right\} \end{aligned} \quad (4)$$

Nous retrouvons alors l'équation de programmation dynamique d'un problème de contrôle optimal de chaîne de Markov, problème résolu par le théorème 5.1 (voir section 5.1 [ACC]) :

---

1. Morgane Chevé et Ronan Congar, Uncertainty in Natural Regeneration and the Robust Management of Environmental Resources <http://www.univ-rouen.fr/droit-scenco-gestion/Recherche/CARE/docs/papiers/S20071213.pdf>

$$\forall i \in \{1, \dots, m\}$$

$$k_i = \beta \inf_{c \in C} \{v(\omega_i, c) + \sum_{j=1}^m M_{ij}(c) k_j\} \quad (5)$$

avec

- $\beta = (1 + \rho h)^{-1}$ ,
- l'espace d'état  $E^{h_1}$ ,
- l'espace des consommations  $[0, C_{max}]$ ,
- la fonction d'utilité  $v = hV$ ,

et

$$\begin{aligned} - M_{ii}(c) &= 1 - \frac{h|f(\omega_i) - \varepsilon c|}{h_1} - \frac{h\sigma^2\omega_i^2}{h_1^2}, \forall i \in \{1, \dots, |E^{h_1}|\}, \\ - M_{i,i+1}(c) &= \frac{h(f(\omega_i) - \varepsilon c)_+}{h_1} + \frac{h\sigma^2\omega_i^2}{2h_1^2}, \forall i \in \{1, \dots, |E^{h_1}| - 1\}, \\ - M_{i,i-1}(c) &= \frac{h(f(\omega_i) - \varepsilon c)_-}{h_1} + \frac{h\sigma^2\omega_i^2}{2h_1^2}, \forall i \in \{2, \dots, |E^{h_1}|\}, \end{aligned}$$

les autres coefficients de  $M(c)$  étant nuls.

Les problèmes de ce type sont résolus par deux algorithmes classiques prouvés et décrits section 6 [ACC] : l'algorithme d'itération sur les valeurs et l'algorithme d'itération sur les stratégies (ou d'Howard).

Les minima seront cherchés sur un espace de consommation discrétisé  $C^{h_2} := [0, C_{max}] \cap h_2 \cdot \mathbb{N}$  où  $h_2$  est le pas de discrétisation de l'espace des consommations (voir section 7.2 [ACC]).

### 0.2.2 Simulation d'une trajectoire

La discrétisation en temps repose sur une approximation du mouvement brownien (voir section 5.2 [ACC]). Une fois la consommation feedback  $C(\omega_i)$ ,  $\forall i \in \{1, \dots, |E^{h_1}|\}$  calculée, cette discrétisation définit une suite récurrente stochastique :

$$E_{k+1} = E_k + h(f(E_k) - \varepsilon C(E_k)) + \sqrt{h}\sigma E_k \delta w_k \text{ où } \delta w_k \text{ est une variable aléatoire de loi uniforme sur } \{-1, 1\}.$$

Nous remarquons ici que  $(E_k)_k$  ne reste pas forcément dans  $E^{h_1}$  l'espace discrétisé ;  $C(E_k)$  n'est donc pas définie car nous avons calculé seulement les  $(C_i)_{i \in \{1, \dots, |E^{h_1}|\}} = (C(\omega_i))_{i \in \{1, \dots, |E^{h_1}|\}}$  i.e. la consommation sur la grille d'état. Pour remédier à cela, pour  $E \in [\omega_i, \omega_{i+1}[$ , nous définissons  $C$  de la manière suivante :  $C(E) := C_i + \frac{C_{i+1} - C_i}{\omega_{i+1} - \omega_i} (E - \omega_i)$  (voir section 7.2 [ACC]).

## 0.3 Organisation du code

Les fichiers sources sont organisés de la manière suivante :

**matrice.cpp** contient l'objet "matrice" avec des méthodes et des surcharges d'opérateur permettant une manipulation facile des matrices dans le code.

**fonctions.cpp** contient des fonctions utiles pour les calculs : partie positive, partie négative, la simulation d'une variable aléatoire de bernoulli sur  $\{-1, 1\}$  de paramètre  $\frac{1}{2}$ ...

**algos.cpp** contient trois algorithmes :

l'algorithme d'itération sur les valeurs, l'algorithme d'itération sur les stratégies (Howard), l'algorithme de Thomas qui permet d'inverser les matrices tridiagonales (utilisé dans l'algo d'Howard) et la fonction qui simule une trajectoire.

**constantes.h** contient toutes les constantes du problème :

$e := \varepsilon$ ,  $s := \sigma$ ,  $r := \rho$ ,  $E_{max} := E_{max}$  et  $E_0 := E_0$ .

Il contient aussi les constantes liées aux algorithmes :

- Tmax le temps jusqu'où la simulation est calculée,
- n le nombre d'itérations dans l'algorithme d'Howard,
- h le paramètre à diminuer si l'algo renvoie zero et un message d'erreur,
- Cmax la valeur maximale jusqu'où l'on cherche la consommation,
- n2 le nombre de points dans l'espace des consommations,
- nT le nombre de points de calcul dans le temps  $[0, Tmax]$ ,

- n1 le nombre de points dans l'espace d'état,
- h1 le pas dans l'espace d'état
- et h2 le pas dans l'espace des consommations.

Certaines constantes sont déterminées par les autres (nombre de points  $\sim$  pas).

**main.cpp** construit tout d'abord les fonctions  $f$  et  $V = -U$  sur l'espace discrétisé ( $f$  est donc un vecteur de dimension  $\text{card}(E^{h_1})$  et  $V$  est une matrice de dimension  $\text{card}(E^{h_1}) \times \text{card}(C_{h_2})$ ).

Ensuite il utilise la fonction "howard" qui n'est rien d'autre que l'algorithme d'Howard, qui renvoie une matrice  $\text{card}(E^{h_1}) \times 3$  :

- la première colonne est l'espace discrétisé (si un coefficient vaut  $\omega_i$ , celui de la lignes d'en dessous vaut  $\omega_i + h_1$ ),
- la deuxième colonne est la fonction valeur J=-K (si un coefficient est sur la même ligne que  $\omega_i$ , c'est la fonction valeur en partant de  $\omega_i$ ),
- et enfin la troisième colonne est la consommation optimale (si un coefficient est sur la même ligne que  $\omega_i$ , c'est la fonction consommation en  $\omega_i$  :  $C_i = C(\omega_i)$ ).

Nous rajoutons deux colonnes à cette matrice afin que les solutions théoriques attendues y soient, et que nous puissions comparer le calcul à la théorie : ainsi la 4ième colonne est la fonction valeur théorique, la 5ième la consommation théorique (elles dépendent aussi des constantes du problème).

Ensuite, la consommation est extraite de cette matrice, et est une entrée de "simul", la fonction qui simule une trajectoire solution. Cette fonction renvoie une matrice  $n_T \times 2$ . La première colonne est la grille du temps (le  $i$ ème coefficient vaut  $t_i = i \frac{T_{max}}{n_T}$ ). La deuxième colonne est l'état de l'environnement (le  $i$ ème coefficient est l'état de l'environnement au temps  $t_i$ ).

Enfin, deux fichiers sont ouverts pour recopier les vecteurs solutions : "fonction-valeur-conso.txt" et "simulation.txt".

Après avoir compilé les fichiers sources (écrire "g++ -Wall main.cpp matrice.cpp fonctions.cpp al-gos.cpp -o prog" en ligne de commande dans le dossier où sont les fichiers, ou "bash go") et lancé le programme (écrire "./prog" en ligne de commande), nous pouvons alors utiliser gnuplot ("gnuplot" en ligne de commande) pour visualiser les courbes :

- "plot "fonction\_valeur\_conso.txt" using 1:2 with lines, "fonction\_valeur\_conso.txt" using 1:4 with lines; " *pour voir les courbes (théorique et calculée) de la fonction valeur*,
- "plot "fonction\_valeur\_conso.txt" using 1:3 with lines, "fonction\_valeur\_conso.txt" using 1:5 with lines; " *pour voir les courbes (théorique et calculée) de la consommation optimale*,
- "plot "simulation.txt" using 1:2 with lines;" *pour voir une simulation de trajectoire*.

#### 4. TRAVAUX DE DERNIÈRE ANNÉE

La théorie des Probabilités et Statistiques, ainsi que l'Informatique étant présents dans de nombreux domaines des Mathématiques Appliquées, j'ai décidé, après quelques mois d'hésitation intense, de suivre la deuxième année du Master « Mathématiques Vision Apprentissage » (MVA) de l'ENS Cachan (Master Recherche en Mathématiques Appliquées) plutôt que le M2R « Probabilités-Statistiques » de l'Université Paris-Sud XI : j'ai cependant suivi et validé le cours d'Apprentissage Statistique de ce dernier Master, dans le cadre du Magistère de Mathématiques.

Le M2R MVA étant très prisé par les étudiants de dernière année d'école d'ingénieur, la formation était un peu moins théorique que les années précédentes, mais beaucoup plus appliquée : j'ai ainsi pu m'habituer au cheminement consistant, à partir d'un problème réel (traitement d'image, apprentissage statistique, planification...), à fournir les éléments de démonstration des outils théoriques, à garantir leur validité lors de leur utilisation informatique, et à les utiliser vraiment pour résoudre le problème (développement). La base théorique acquise lors de mes deux premières années de Magistère est indispensable à cette procédure afin d'éviter d'admettre les outils mathématiques utilisés, mais aussi afin de modifier, si nécessaire, ces outils pour coller au problème. Ma dernière année d'étude m'a forcé, à travers les nombreux travaux demandés, à « mettre les mains dans le cambouis » (à réellement résoudre ces problèmes réels à l'ordinateur), et à rédiger le cheminement utilisé. J'intègre dans ce mémoire les deux projets personnels de cette dernière année, ainsi qu'un projet réalisé en groupe de trois étudiants.

**4.1. Traitement d'Image : Synthèse de Textures.** L'image était un des grands thèmes de cette deuxième année de master : l'un des cours que j'ai suivi était dédié à l'état de l'art concernant le traitement d'image. Dans le cadre de ce cours, j'ai travaillé sur la synthèse de texture par méthode statistique : la notion de transformée de Fourier discrète d'une image est à la base de ce travail.

Professeur référent : Gabriel Peyré  
Master MVA 2ème année - ENS Cachan  
Projet du cours: "Parcimonie et Compressed Sensing"

# Synthèse de textures par méthodes statistiques

---

Nicolas Drougard

Cachan, le 3 octobre 2012

# Table des matières

|   |   |    |
|---|---|----|
| 1 | Random Phase Noise                            | 3  |
| 2 | Le texton                                     | 8  |
| 3 | Asymptotic Discrete Spot Noise                | 10 |
| 4 | Synthèse de textures plus grandes que l'input | 12 |
| 5 | Extension aux textures couleur                | 14 |

## Résumé

Ce rapport se concentre autour d'un algorithme de synthèse de texture appelé Random Phase Noise (RPN). Par synthèse, nous entendons création d'une nouvelle texture à partir d'un modèle donné en entrée. Cet algorithme ne s'applique qu'à une classe de textures appelées micro-textures. Il sera introduit par une discussion sur la définition encore imprécise de texture, et la propriété d'invariance par randomisation des phases des coefficients de Fourier des micro-textures. Nous définirons la notion de texton, représentant utile des micro-textures similaires, contenant leurs principales propriétés visuelles et statistiques. L'algorithme Asymptotique Discrete Spot Noise sera abordé. Enfin, ces algorithmes seront développés afin de synthétiser des textures de taille plus grande que celle du modèle en entrée, et étendus aux textures couleur.

## Notations

Ces quelques notations seront utilisées dans ce rapport :

- une image de taille  $n \times p$  (en pixel), sera une fonction de  $\Omega = I_n \times I_p$  dans  $\mathbb{R}$   
où  $I_m = [-\frac{m}{2}, \dots, 0, \dots, \frac{m}{2} - 1]$  si  $m$  est pair,  
 $I_m = [-\frac{m-1}{2}, \dots, 0, \dots, \frac{m-1}{2}]$  sinon.
- $\hat{=}$  sera une égalité type définition

## Introduction

La définition de texture reste encore imprécise. Nous attendons cependant d'une image qualifiée de texture, quelques caractéristiques qui peuvent paraître évidentes.

Parmi ces caractéristiques, nous pouvons citer l'homogénéité dans l'espace : il est possible d'imaginer que l'image n'est qu'une restriction rectangulaire d'une surface plus grande. La texture possède typiquement des structures qui se répètent, avec des positions, des couleurs, des tailles aléatoires... Plusieurs exemples de textures sont présentées figure 1.





FIGURE 1 – six différentes textures : du sable, un damier, de l’herbe, des nuages, un amas de cailloux et un mur de brique (de gauche à droite et de bas en haut).

Ces exemples nous amènent à distinguer deux types de textures. Certaines contiennent des objets visuels qui construisent l’image par répétition spatiale plus ou moins régulière (les cailloux ou les briques par exemple). Nous les appellerons macro-texture. Les micro-textures seront quant à elles les textures dépourvues de ces types d’objets (les nuages ou le sable).

## 1 Random Phase Noise

Les "définitions" précédentes sont très vagues voire subjectives : où est la limite entre une micro et une macro-texture ? Qu’en est-il d’une image de très petits cailloux ? Le nuage n’est-il pas un objet construisant la texture de la quatrième image de la figure 1 ?

Le travail de Julesz [1] nous permet de définir plus rigoureusement les micro-textures. La théorie de la perception des textures qu’il développe, laisse penser que deux textures dont les coefficients de Fourier ont respectivement le même module sont indiscernables. Modifier les phases des coefficients de Fourier d’une texture ne changerait donc pas la perception que nous en avons.

Nous souhaitons donc tester cette affirmation sur les exemples de la figure 1 : nous implémentons un algorithme qui prend en entrée une texture, calcule ses coefficients de Fourier avec l’algorithme *Fast Fourier Transform*, change les phases des coefficients de Fourier, et enfin calcule l’image associée aux coefficients de Fourier modifiés.

Faisons cependant attention à la randomisation des phases : les coefficients de Fourier modifiés doivent rester les coefficients de Fourier d’une image, c’est à dire d’un signal réel. Pour cela, plusieurs conditions sur les phases sont nécessaires. La transformée de Fourier d’une image  $h$  de taille  $n \times p$  (voir section Notation) est une image complexe de même taille définie par  $\hat{h}(\xi) = \sum_{x \in \Omega} h(x) e^{-2i\pi(\frac{1}{n}x_1\xi_1 + \frac{1}{p}x_2\xi_2)}$ .

Ainsi,

$$\hat{h}(-\xi) = \sum_{x \in \Omega} h(x) e^{-2i\pi(-\frac{1}{n}x_1\xi_1 - \frac{1}{p}x_2\xi_2)} = \sum_{x \in \Omega} h(x) e^{2i\pi(\frac{1}{n}x_1\xi_1 + \frac{1}{p}x_2\xi_2)} = \overline{\hat{h}(\xi)}$$

puisque  $h$  est réelle. La phase du coefficient d’indice  $-\xi$  est donc l’opposé de la phase du coefficient d’indice  $\xi$ , et la phase du coefficient 0 vaut 0 ou  $\pi$ .

$$\hat{h}\left(-\frac{n}{2}, 0\right) = \sum_{x \in \Omega} h(x) e^{-\frac{2i\pi}{n}x_1 \times -\frac{n}{2}} = \sum_{x \in \Omega} h(x) e^{i\pi x_1} \in \mathbb{R}$$

car  $e^{i\pi x_1} \in \{-1, 1\}$ . La phase du coefficient d'indice  $\xi = (-\frac{n}{2}, 0)$  si il existe ( $n$  pair), vaut 0 ou  $\pi$ . On montre de la même manière qu'il en est de même pour les coefficients d'indice  $(0, -\frac{p}{2})$  et  $(-\frac{n}{2}, -\frac{p}{2})$ .

Ces conditions sont suffisantes.

En effet, supposons qu'elles soient respectées :  $h(x)$  est égal à  $\sum_{\xi \in \Omega} \hat{h}(\xi) e^{2i\pi(\frac{1}{n}x_1\xi_1 + \frac{1}{p}x_2\xi_2)}$  à un facteur multiplicatif réel près, et si  $n$  et  $p$  sont impairs

$$\begin{aligned}
\sum_{\xi \in \Omega} \hat{h}(\xi) e^{2i\pi(\frac{1}{n}x_1\xi_1 + \frac{1}{p}x_2\xi_2)} &= \hat{h}(0) + \sum_{\xi \in \Omega | \xi_1, \xi_2 > 0} \hat{h}(\xi) e^{2i\pi(\frac{1}{n}x_1\xi_1 + \frac{1}{p}x_2\xi_2)} + \sum_{\xi \in \Omega | \xi_1, \xi_2 < 0} \hat{h}(\xi) e^{2i\pi(\frac{1}{n}x_1\xi_1 + \frac{1}{p}x_2\xi_2)} \\
&+ \sum_{\xi \in \Omega | \xi_1 < 0, \xi_2 > 0} \hat{h}(\xi) e^{2i\pi(\frac{1}{n}x_1\xi_1 + \frac{1}{p}x_2\xi_2)} + \sum_{\xi \in \Omega | \xi_1 > 0, \xi_2 < 0} \hat{h}(\xi) e^{2i\pi(\frac{1}{n}x_1\xi_1 + \frac{1}{p}x_2\xi_2)} \\
&= \hat{h}(0) + \sum_{\xi \in \Omega | \xi_1, \xi_2 > 0} \hat{h}(\xi) e^{2i\pi(\frac{1}{n}x_1\xi_1 + \frac{1}{p}x_2\xi_2)} + \overline{\sum_{\xi \in \Omega | \xi_1, \xi_2 > 0} \hat{h}(\xi) e^{2i\pi(\frac{1}{n}x_1\xi_1 + \frac{1}{p}x_2\xi_2)}} \\
&+ \sum_{\xi \in \Omega | \xi_1 < 0, \xi_2 > 0} \hat{h}(\xi) e^{2i\pi(\frac{1}{n}x_1\xi_1 + \frac{1}{p}x_2\xi_2)} + \overline{\sum_{\xi \in \Omega | \xi_1 < 0, \xi_2 > 0} \hat{h}(\xi) e^{2i\pi(\frac{1}{n}x_1\xi_1 + \frac{1}{p}x_2\xi_2)}} \\
&= \hat{h}(0) + 2\operatorname{Re}\left\{ \sum_{\xi \in \Omega | \xi_1, \xi_2 > 0} \hat{h}(\xi) e^{2i\pi(\frac{1}{n}x_1\xi_1 + \frac{1}{p}x_2\xi_2)} \right\} + 2\operatorname{Re}\left\{ \sum_{\xi \in \Omega | \xi_1 < 0, \xi_2 > 0} \hat{h}(\xi) e^{2i\pi(\frac{1}{n}x_1\xi_1 + \frac{1}{p}x_2\xi_2)} \right\} \in \mathbb{R}
\end{aligned}$$

Si  $n$  et/ou  $p$  est pair, il suffit de sortir des sommes les termes contenant les coefficients de Fourier d'indices  $\xi = (-\frac{n}{2}, 0)$ ,  $\xi = (0, -\frac{p}{2})$ ,  $\xi = (-\frac{n}{2}, -\frac{p}{2})$  : ces termes sont réels car les phases des coefficients de Fourier valent 0 ou  $\pi$ , et on peut composer les parties réelles avec les sommes restantes).

Nous pouvons alors définir ce qu'est une *phase aléatoire uniforme*.

**Définition 1.1** (*Uniform random phase*)

Une image  $\theta$  est une phase aléatoire uniforme si :

- $\theta$  est impaire :  $\forall x \in \Omega, \theta(-x) = -\theta(x)$  ;
- chaque élément  $\theta(x)$  est une variable aléatoire uniforme sur  $] -\pi, \pi]$  si  $x \notin \{(0, 0), (-\frac{n}{2}, 0), (0, -\frac{p}{2}), (-\frac{n}{2}, -\frac{p}{2})\}$ , uniforme sur  $\{0, \pi\}$  sinon ;
- pour toute partie  $S \subset \Omega$  qui ne contient pas deux points symétriques distincts, la famille de variables aléatoires  $\{\theta(x) | x \in S\}$  est indépendante.

Si l'algorithme décrit en début de section laisse invariante la perception que l'on a de la texture, c'est une algorithme de synthèse de texture. Cet algorithme, bien formalisé dans la définition qui suit, est connu sous le nom de *Random Phase Noise* :

**Définition 1.2** (*algorithme RPN*)

- *Input* : image texture  $h$  ;
- création d'une phase aléatoire uniforme  $\theta$  ;
- calcul des coefficients de Fourier de  $h$ ,  $\hat{h}(\xi)$  ;
- calcul de  $\hat{z}$  tel que  $\forall \xi, \hat{z}(\xi) = \hat{h}(\xi) e^{i\theta(\xi)}$  ;
- calcul de l'image  $z$  associée aux coefficients de Fourier  $\hat{z}(\xi)$  ;
- *Output* :  $z$ .

**Remarque** : Cet algorithme est équivalent à un algorithme qui remplacerait la phase de  $\hat{h}(\xi)$  par  $\theta(\xi)$  car si  $\phi$  est la phase de  $\hat{h}$ ,  $(\theta + \phi) \bmod 2\pi$  reste une phase aléatoire uniforme.

Nous pouvons maintenant tester cet algorithme sur les textures "cailloux" et "sable" : les résultats de l'algorithme RPN sont exposés figure 2 et 3.

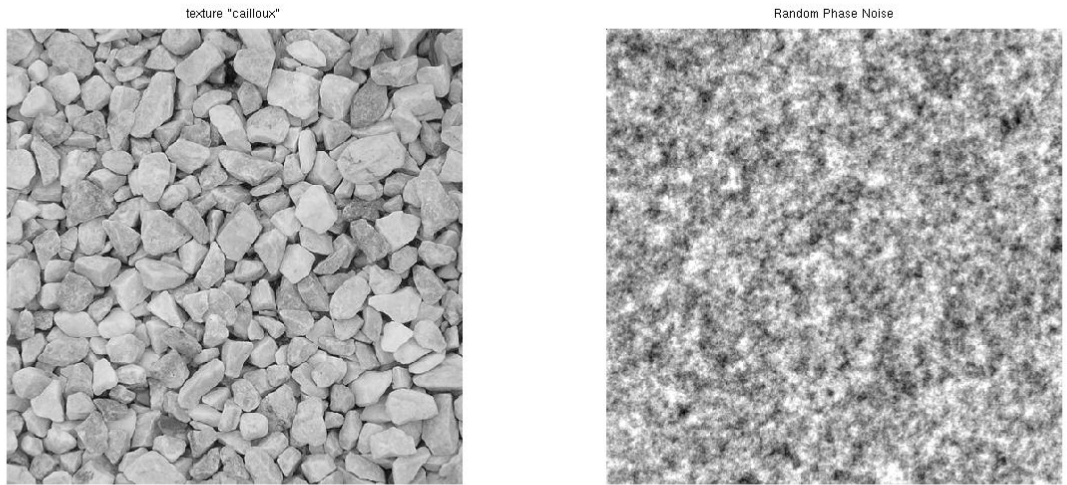


FIGURE 2 – algorithme Random Phase Noise appliqué à la macro-texture "cailloux" en noir et blanc (MATLAB)

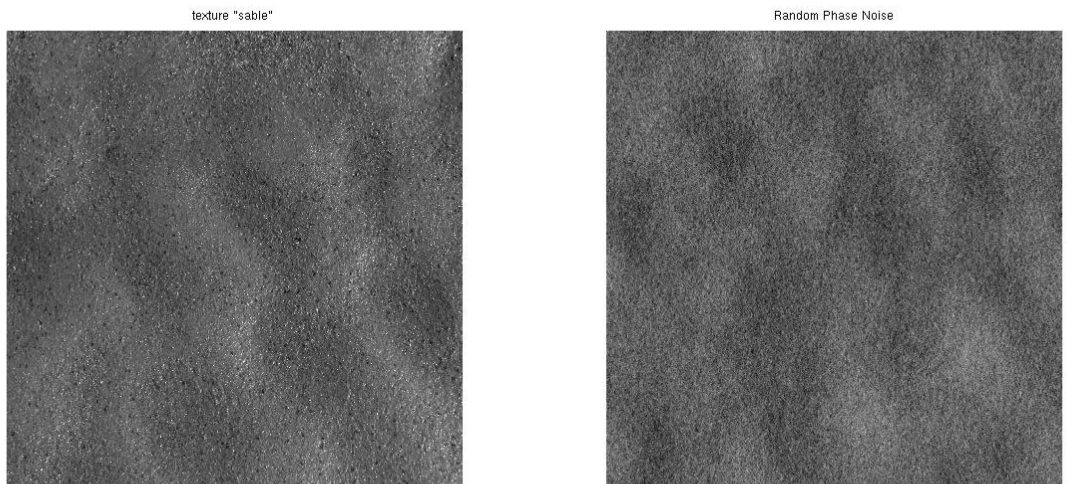


FIGURE 3 – algorithme Random Phase Noise appliqué à la micro-texture "sable" en noir et blanc (MATLAB)

Nous remarquons que la perception de la texture "cailloux" n'est pas invariante par randomisation des phases de ses coefficients de Fourier : cette randomisation détruit complètement l'objet visuel élémentaire qu'est le caillou. La perception de la texture "sable" quant à elle n'est pas altérée par ce procédé. Cette observation nous permet de donner une définition un peu plus scientifique de la micro-texture : une **micro-texture** sera désormais une texture dont la **perception est invariante par randomisation des phases de ses coefficients de Fourier**.

Testons maintenant notre algorithme sur la micro-texture "nuage", les résultats étant exposés figure 4.

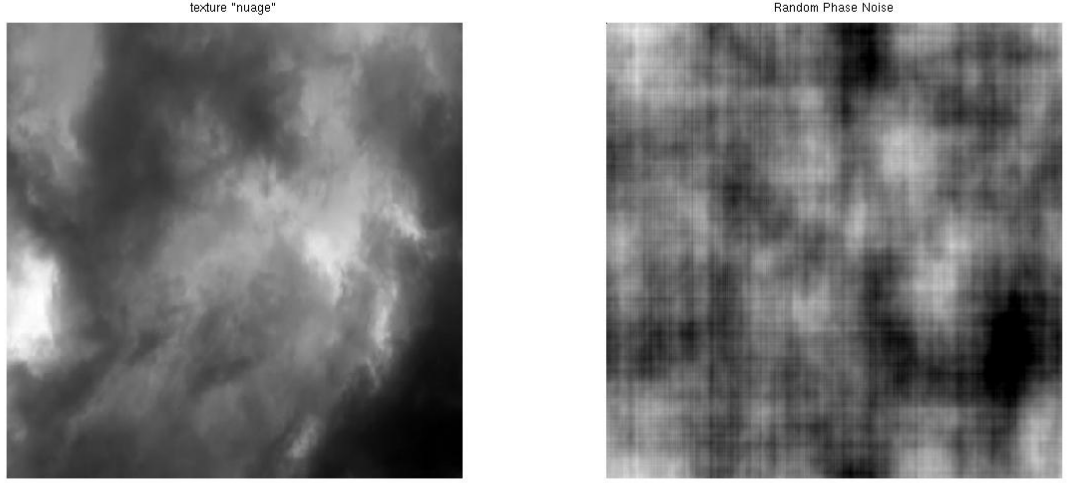


FIGURE 4 – algorithme Random Phase Noise appliqué à la micro-texture "nuage" en noir et blanc (MATLAB)

Malgré sa qualification de micro-texture, la perception de la texture "nuage" ne semble pas invariante par l'application de l'algorithme RPN. Ce résultat est en effet dû au fait que l'utilisation de l'algorithme FFT (sur lequel est basé l'algorithme RPN) suppose que l'image en entrée est périodique. Pour remédier aux artefacts produits par la non-périodicité de l'image en entrée, il convient de remplacer l'image en entrée  $h$  par sa composante périodique  $p_h$  dont les coefficients de Fourier sont définis par :

$$\forall \xi \neq 0, \quad \hat{p}_h(\xi) = \hat{h}(\xi) + \frac{\hat{A}(\xi)}{4(\sin^2(\frac{\pi \xi_1}{n}) + \sin^2(\frac{\pi \xi_2}{p}))}$$

$$\hat{p}_h(0) = \hat{h}(0) = \text{mean}(h)$$

où  $A = A_1 + A_2$  avec

$$\begin{aligned} \forall y \in I_p, \quad & A_1(\min(I_n), y) = h(\max(I_n), y) - h(\min(I_n), y) \\ & A_1(\max(I_n), y) = h(\min(I_n), y) - h(\max(I_n), y) \\ & \forall x \neq \min(I_n), \max(I_n), \quad A_1(x, y) = 0 \\ \forall x \in I_n, \quad & A_2(x, \min(I_p)) = h(x, \max(I_p)) - h(x, \min(I_p)) \\ & A_2(x, \max(I_p)) = h(x, \min(I_p)) - h(x, \max(I_p)) \\ & \forall y \neq \min(I_p), \max(I_p), \quad A_2(x, y) = 0 \end{aligned}$$

La construction détaillée de cette composante périodique est exposée dans le travail de Lionel Moisan, *Periodic plus smooth image decomposition* [2]. Le but de cette décomposition est d'écrire l'image  $h$  comme la somme d'une image  $p_h$  périodique proche de  $h$ , et d'une image  $s_h$  lisse de moyenne nulle. Il paraît alors raisonnable de remplacer  $h$  par  $p_h$  afin d'éviter les artefacts dus au fait que  $h$  n'est pas périodique.

Ici, on comprend bien que la zone blanche à droite de la texture "nuage" et la zone noire à gauche représentent un saut qui va malheureusement être pris en compte par la *DFT*. La figure 4 représente la composante périodique de la texture "nuage", ainsi que l'image résultant de l'algorithme RPN.

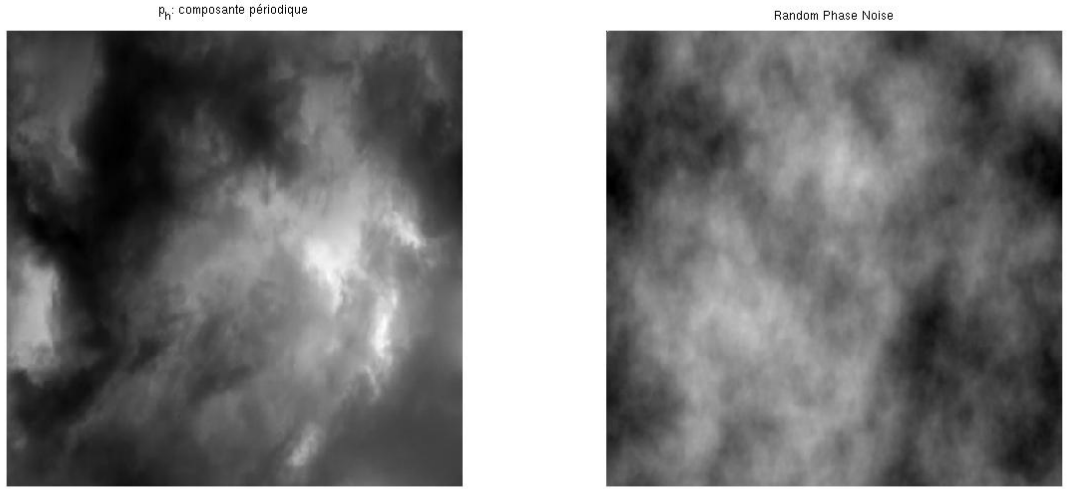


FIGURE 5 – algorithme Random Phase Noise appliqué à la composante périodique (décomposition PPS) de la micro-texture "nuage" en noir et blanc (MATLAB)

Nous pouvons utiliser un bruit blanc gaussien pour implémenter le RPN : en effet, soit  $G$  une image formée de variables aléatoires indépendantes suivant une loi gaussienne centrée et réduite. Les coefficients de Fourier d'indices  $\xi \in \{\xi \mid \xi_1 > 0, \xi_2 > 0\}$  sont des variables aléatoires gaussiennes complexes indépendantes, c'est-à-dire des variables aléatoires complexes dont les modules sont indépendants et suivent des lois de Rayleigh, et les phases sont *iid* de loi uniforme sur  $]-\pi, \pi]$ . Nous pouvons donc réécrire l'algorithme Random Phase Noise comme suit :

**Définition 1.3** (*algorithme RPN utilisant un bruit blanc gaussien*)

- *Input : image texture  $h$  ;*
- *création d'une image formée de variables aléatoires gaussiennes indépendantes centrées réduites  $G$  ;*
- *calcul des coefficients de Fourier de  $h$ ,  $\hat{h}(\xi)$ , et de  $G$ ,  $\hat{G}(\xi)$ ,  $\hat{G}(0) = 1$  ;*
- *calcul de  $\hat{z}$  tel que  $\forall \xi, \hat{z}(\xi) = \hat{h}(\xi) \frac{\hat{G}(\xi)}{|\hat{G}(\xi)|}$  ;*
- *calcul de l'image  $z$  associée aux coefficients de Fourier  $\hat{z}(\xi)$  ;*
- *Output :  $z$ .*

## 2 Le texton

Nous venons de voir que pour la grande classe des micro-textures, remplacer les phases des coefficients de Fourier par des phases aléatoires ne change pas la perception de la texture. Ce fait nous amène à définir un représentant particulier des micro-textures : celui dont toutes les phases sont nulles, et dont la moyenne est la même que celle de la micro-texture. Ce représentant est appelé **texton**.

**Définition 2.1** (*Texton d'une image*)

Soit  $h$  une image de taille  $n \times p$ . Le texton  $T(h)$  de  $h$  est une image définie par

- dans le domaine de Fourier,  $\forall \xi \neq 0$ ,  $T(\hat{h})(\xi) = |\hat{h}(\xi)|$  et  $T(\hat{h})(0) = \hat{h}(0)$  ;
- dans le domaine spatial,  $\forall x \in \Omega$ ,  $T(h)(x) = \frac{1}{np} \hat{h}(0) + \frac{1}{np} \sum_{\xi \in \Omega, \xi \neq 0} |\hat{h}(\xi)| e^{2i\pi(\frac{1}{n}x_1\xi_1 + \frac{1}{p}x_2\xi_2)}$ .

Les figures 6 et 7 exhibent les textons associés respectivement à la micro-texture "sable" et à la composante périodique de la micro-texture "nuages".

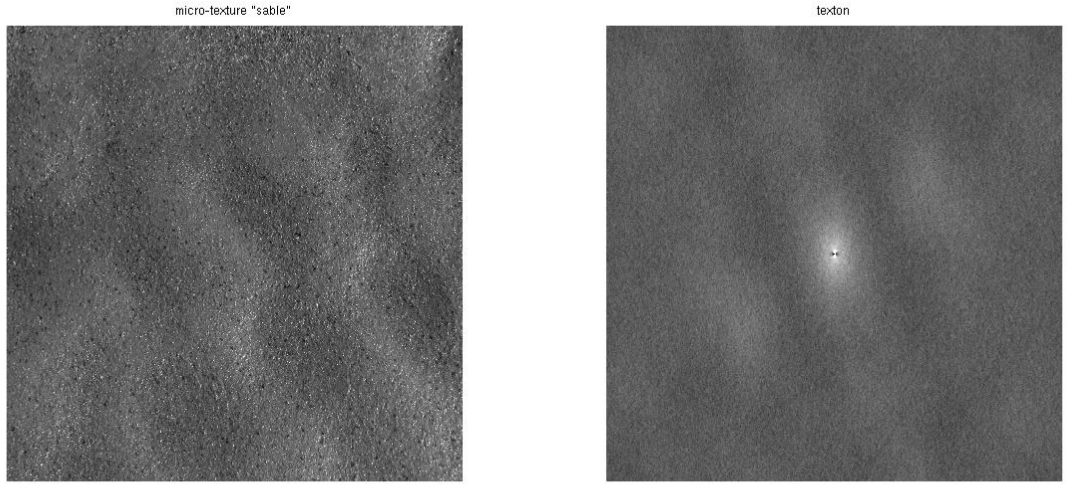


FIGURE 6 – à droite le texton associé à la micro-texture "sable" (MATLAB)

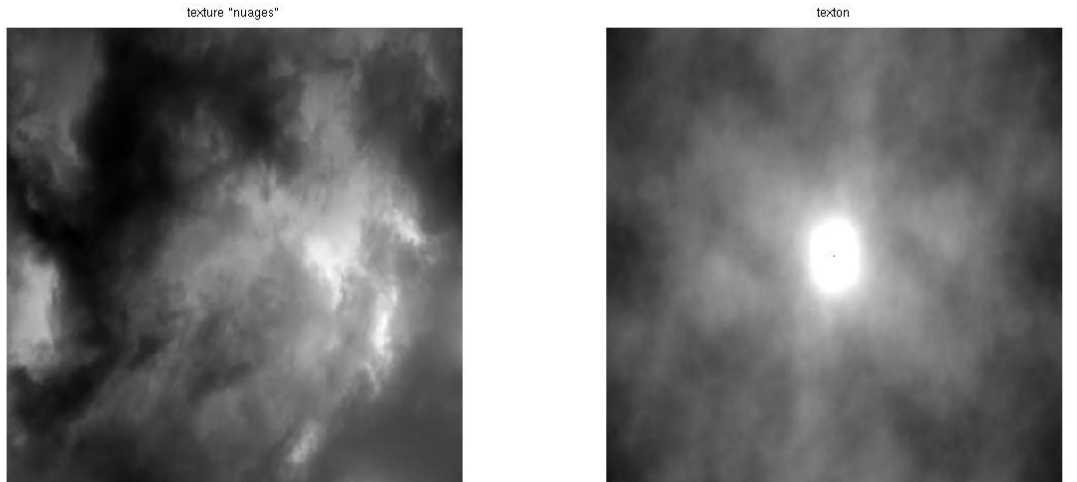


FIGURE 7 – à droite le texton associé à la composante périodique de la micro-texture "nuages" (MATLAB)

Nous observons sur ces exemples que le texton vérifie une certaine concentration spatiale en zéros. Nous pouvons voir facilement que le texton atteint sa valeur maximale en zéro : supposons, quitte à soustraire au texton sa valeur minimale, que  $\forall x \in \Omega, T(x) > 0$ ,

$$\begin{aligned} T(h)(x) = |T(h)(x)| &= \left| \frac{1}{np} \hat{h}(0) + \frac{1}{np} \sum_{\xi \in \Omega, \xi \neq 0} |\hat{h}(\xi)| e^{2i\pi(\frac{1}{n}x_1\xi_1 + \frac{1}{p}x_2\xi_2)} \right| \\ &\leq \frac{1}{np} \hat{h}(0) + \frac{1}{np} \sum_{\xi \in \Omega, \xi \neq 0} |\hat{h}(\xi)| = T(h)(0) \end{aligned}$$

par l'inégalité triangulaire.

Le texton a une importance capitale dans l'analyse des micro-textures. Il contient un "résumé" des propriétés de la texture, telles que la taille, l'anisotropie (la dépendance aux directions) et les statistiques du second ordre.

Le calcul suivant nous éclaire sur cette dernière propriété : supposons, quitte à soustraire la moyenne, que l'image  $h$  est de moyenne nulle. La covariance empirique de cette image est définie par

$$C_h(x) \triangleq \frac{1}{np} \sum_{z \in \Omega} h(z)h(z-x) = \frac{1}{np} h * \tilde{h}(x)$$

où  $\tilde{h}(x) = h(-x)$ .

Les coefficients de Fourier de la covariance empirique sont

$$\hat{C}_h(\xi) = \frac{1}{np} \widehat{(h * \tilde{h})}(\xi) = \frac{1}{np} \hat{h}(\xi) \hat{\tilde{h}}(\xi) = \frac{1}{np} \hat{h}(\xi) \hat{h}(-\xi)$$

D'autre part,

$$\frac{1}{np} T(\widehat{h * \tilde{h}})(\xi) = \frac{1}{np} (T(\hat{h})(\xi))^2 = \frac{1}{np} |\hat{h}(\xi)|^2 = \frac{1}{np} \hat{h}(\xi) \bar{\hat{h}}(\xi) = \frac{1}{np} \hat{h}(\xi) \hat{h}(-\xi)$$

$C_h$  et  $\frac{1}{np} T(u) * T(u)$  ayant les mêmes coefficients de Fourier, ils sont égaux.

C'est cette égalité

$$C_h = \frac{1}{np} T(u) * T(u)$$

qui nous amène à affirmer que le texton nous informe sur les statistiques du second ordre de la texture. Il est moins flou que la covariance (qui résulte de l'autoconvolution du texton), et reste une image alors que les valeurs de la covariance sont des carrés d'intensité.

### 3 Asymptotic Discrete Spot Noise

L'algorithme Asymptotic Discrete Spot Noise est aussi un algorithme de synthèse de texture, qui donne un résultat très similaire au résultat du RPN. Il est d'ailleurs souvent confondu avec ce dernier dans la littérature, bien que ces deux algorithmes soient théoriquement différents.

Il est cependant intéressant d'en dire quelques mots puisque sa construction peut-être assez intuitive. Un algorithme appelé Discrete Spot Noise (DSN) fut introduit par van Wijk pour synthétiser de nouvelles textures : il consistait simplement à générer des translations aléatoires d'une image initiale (un point, d'où "spot"), et de les sommer.

Soient  $X_i$  une suite de variables aléatoires *iid* de loi uniforme sur  $\Omega$ , et  $h$  l'image d'un point. Le résultat de l'algorithme est alors une réalisation de l'image aléatoire définie par

$$\forall x \in \Omega, f_k(x) = \sum_{i=1}^k h(x - X_i)$$

La figure 8 représente les résultats de cet algorithme pour différents nombres d'images sommées.

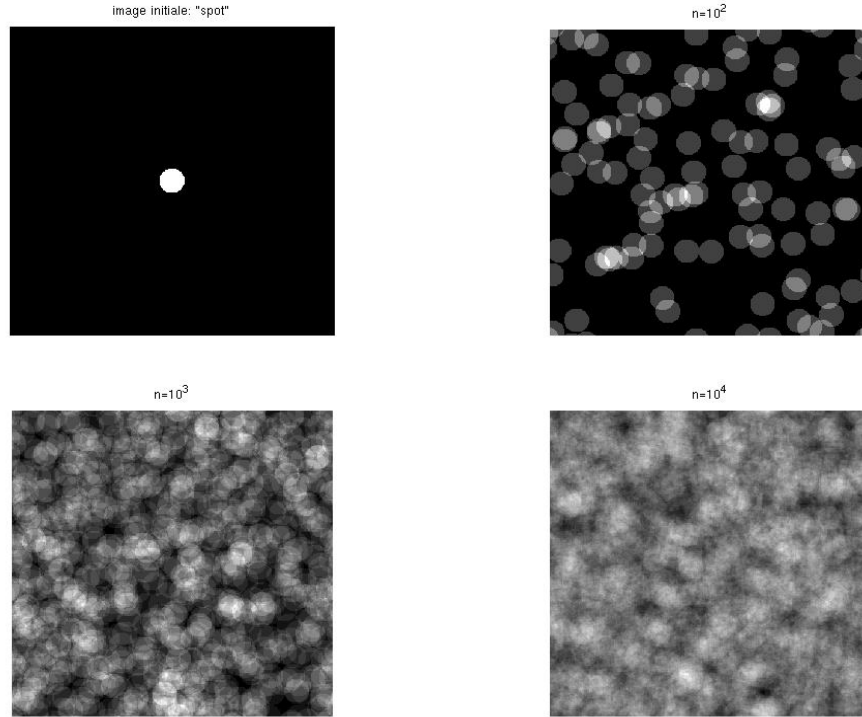


FIGURE 8 – de gauche à droite et de haut en bas, l'image initiale (spot), puis le résultat de l'algorithme DSN avec  $10^2$ ,  $10^3$  et  $10^4$  images sommées (MATLAB)

Le résultat de l'algorithme ADSN est alors simplement une limite de cette séquence  $f_k$  bien normalisée : calculons l'espérance de l'image définie par  $H(x) = h(x - X_i)$ . Soit  $x \in \Omega$  :

$$\mathbb{E}(H(x)) = \frac{1}{np} \sum_{z \in \Omega} h(x - z) = \frac{1}{np} \sum_{y \in \Omega} h(y) = \text{mean}(h)$$

donc l'espérance de  $H$  est l'image constante égale à  $m = \text{mean}(h)$ .

Calculons maintenant sa covariance :

$$\text{Cov}(H(x), H(y)) = \mathbb{E}[(H(x) - m)(H(y) - m)] = \frac{1}{np} \sum_{z \in \Omega} [(h(x - z) - m)(h(y - z) - m)] \triangleq AC_h$$

où  $AC_h$  est la matrice d'autocorrélation de  $h$ .



Le théorème central limite donne alors une limite en loi à la suite

$$\frac{1}{\sqrt{k}}(f_k - kM)$$

où  $M$  est l'image constante égale à  $m$ . Cette suite converge en loi vers la loi normale de dimension  $n \times p$  centrée et de covariance  $AC_h$ . Cette limite est le résultat de l'algorithme ADSN.

Montrons maintenant que cette limite peut-être simulée facilement : soit  $Y$  une image dont les pixels sont des variables aléatoires iid et suivent une loi gaussienne centrée réduite. Alors

$$Z = \frac{1}{\sqrt{np}}(h - M) * Y$$

est le résultat de l'algorithme ASDN appliqué à  $h$ .

En effet, soit  $x, y \in \Omega$  :

$Z$  suit une loi normale, comme combinaison linéaire de pixels de  $Y$ . De plus,

$$\begin{aligned} \mathbb{E}\left[\frac{1}{\sqrt{np}}(h - M) * Y(x)\right] &= \frac{1}{\sqrt{np}} \mathbb{E}\left[\sum_{z \in \Omega} (h(x - z) - m)Y(z)\right] \\ &= \frac{1}{\sqrt{np}} \sum_{z \in \Omega} (h(x - z) - m) \mathbb{E}[Y(z)] = 0 \end{aligned}$$

$$\begin{aligned} \text{Cov}(Z(x), Z(y)) &= \mathbb{E}[Z(x)Z(y)] = \frac{1}{np} \mathbb{E}\left[\left(\sum_{z \in \Omega} (h(x - z) - m)Y(z)\right)\left(\sum_{w \in \Omega} (h(y - w) - m)Y(w)\right)\right] \\ &= \frac{1}{np} \sum_{z \in \Omega} \sum_{w \in \Omega} (h(x - z) - m)(h(y - w) - m) \mathbb{E}[Y(z)Y(w)] \end{aligned}$$

or  $\mathbb{E}[Y(z)Y(w)] = 1$  si  $w = z$ , 0 sinon,

donc

$$\text{Cov}(Z(x), Z(y)) = \frac{1}{np} \sum_{z \in \Omega} (h(x - z) - m)(h(y - z) - m) \hat{=} AC_h$$

$Z$  suit donc une loi normale de dimension  $n \times p$  centrée et de covariance  $AC_h$ , c'est donc bien le résultat attendu.

La convolution étant une multiplication "pixel par pixel" dans le domaine de Fourier, ce résultat nous apporte une implémentation simple de l'algorithme ADSN basé sur l'algorithme FFT :

- Input : image  $h$
- création d'une image de bruit blanc gaussien  $Y$
- calcul des coefficients de Fourier de  $h - M$  et de  $Y$
- calcul de  $\hat{Z} = \frac{1}{\sqrt{np}}(\widehat{h - M})\hat{Y}$
- calcul de  $Z$ , l'image associée aux coefficients de Fourier  $\hat{Z}$ .
- Output :  $Z$

Nous remarquons la proximité entre cet algorithme, et la deuxième implémentation de RPN. Nous multiplions cependant ici les coefficients de Fourier de l'input par  $\hat{Y}$  dont les modules ne sont pas égaux à 1, mais suivent une loi de Rayleigh. Cet algorithme modifie donc les modules des coefficients de Fourier de l'input, contrairement à l'algorithme RPN. Le résultat reste cependant similaire comme le montre la figure 9 :

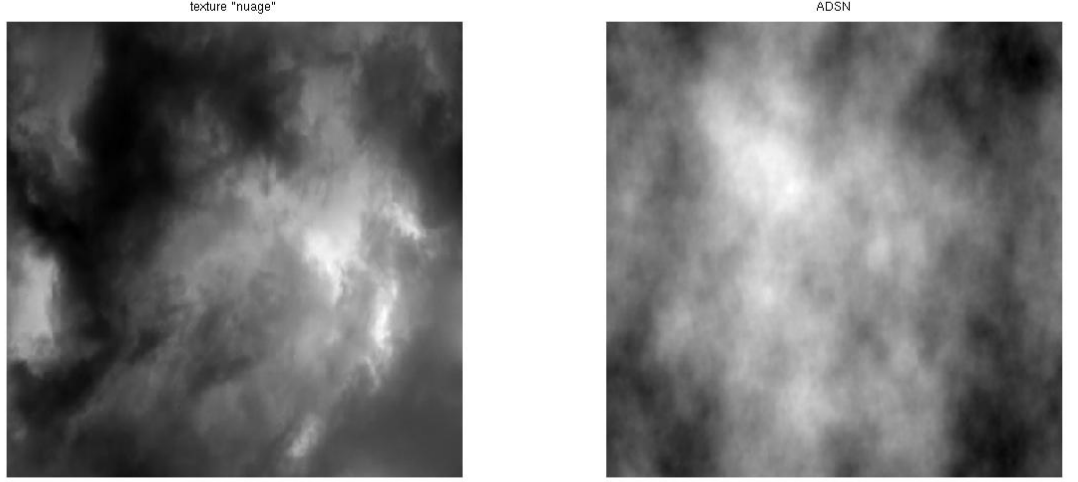


FIGURE 9 – algorithme ADSN appliqué à la texture "nuages" (MATLAB)

## 4 Synthèse de textures plus grandes que l'input

L'intérêt de la synthèse de texture réside en particulier dans le fait de pouvoir synthétiser des images qui sont plus grandes que l'image sur lequel se base l'algorithme (RPN ou ASDN). [3] suggère de créer une image  $h'$  de taille  $n_2 \times p_2$  à partir de l'image en entrée  $h$  de taille  $n_1 \times p_1$  ( $n_2 > n_1$ ,  $p_2 > p_1$ ), puis d'appliquer l'algorithme de synthèse à cette nouvelle image.  $h'$  est définie comme suit :

$$h'(x) = m_h + \sqrt{\frac{n_2 p_2}{n_1 p_1}} (p_h(x) - m_h) \mathbb{1}_{R_1}(x)$$

où  $m_h$  est la moyenne de  $h$ ,  $\mathbb{1}_{R_1}(x)$  l'indicatrice du rectangle défini par le bord de  $h$ , et  $p_h$  la composante périodique de la décomposition PPS de  $h$ . L'image  $h'$  est alors la composante  $p_h - m_h$  dont intensité est multipliée par un réel  $> 1$ , "baignant" dans sa moyenne.

[4] élabore une technique du même type à partir du texton associé à la micro-texture.

Nous allons utiliser une méthode inspirée de travaux évoqués dans [2] : mettre bout à bout des copies renversées de l'image  $h$  de manière à éviter les artefacts de non-périodicité, puis appliquer le RPN à cette nouvelle image. Cette méthode ne permet que de générer des images dont la taille est un multiple de celle de départ, mais les résultats sont plus satisfaisant. Les figures 10 et 11 montrent les résultats obtenus par cette méthode avec nos deux micro-textures.

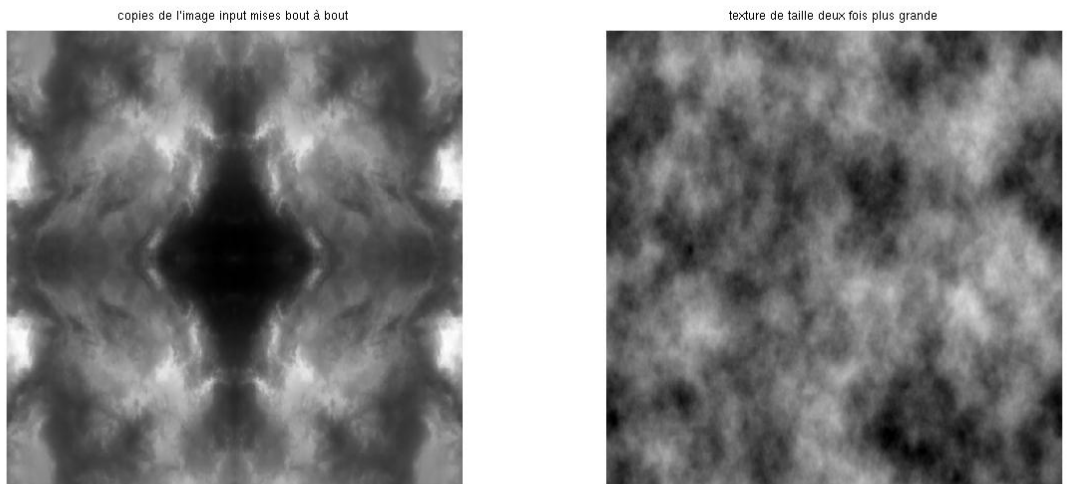


FIGURE 10 – à gauche les copies renversées de la texture "nuages" mises bout à bout (nouvelle image input), à droite la texture deux fois plus grande synthétisée (MATLAB)

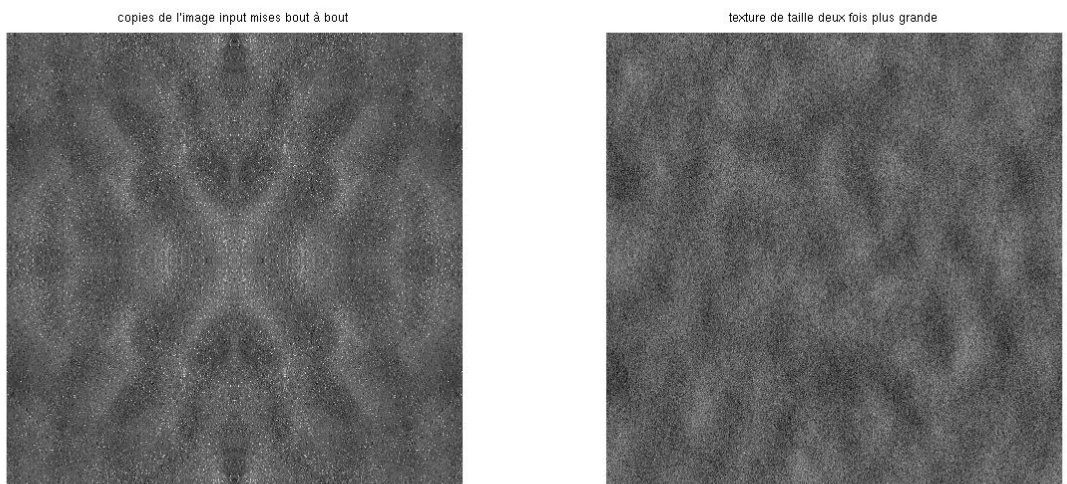


FIGURE 11 – à gauche les copies renversées de la texture "sable" mises bout à bout (nouvelle image input), à droite la texture deux fois plus grande synthétisée (MATLAB)

## 5 Extension aux textures couleur

L'erreur à ne pas commettre pour étendre les algorithmes RPN et ADSN aux textures couleur, est de les appliquer indépendamment aux trois chaînes. Pour ne pas créer de nouvelles couleurs, il faut que la même phase aléatoire uniforme soit utilisée pour les trois chaînes de l'input pour le RPN, le même bruit blanc pour l'ADSN.

Les figures 12 et 13 montrent respectivement l'algorithme bien et mal implémenté.



FIGURE 12 – RPN appliqué à la micro-texture couleur "sable" (MATLAB)

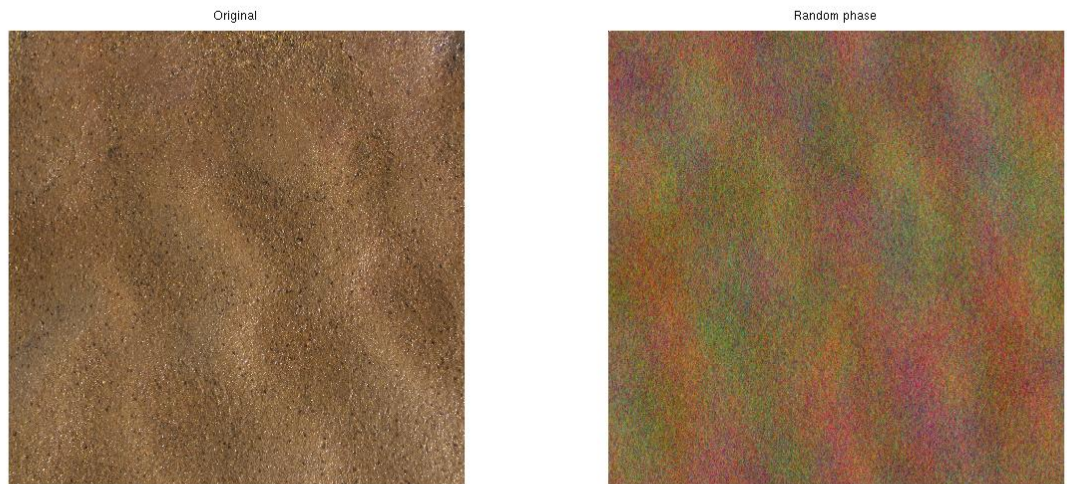


FIGURE 13 – RPN implémenté indépendamment à chaque chaîne de l'image (MATLAB)

## Conclusion

Nous avons pu ressentir à travers ce rapport l'aide que nous apporte la transformée de Fourier discrète pour la synthèse et l'analyse des micro-textures : la définition même du texton en dépend, et les algorithmes présentés utilisent tous l'algorithme FFT. Elle requiert cependant des hypothèses de périodicité, que deux méthodes (Periodic plus Smooth decomposition et la méthode section 4) nous ont permis de satisfaire. D'autres algorithmes basés sur les ondelettes existent.

La proximité théorique et pratique entre les algorithmes RPN et ADSN a été mise en avant, et il reste intéressant d'avoir à l'esprit qu'ils sont issus de deux raisonnements totalement différents.

Le cas des macro-textures a l'air d'un tout autre ordre, et n'a pas été abordé dans ce rapport.

## Bibliographie

- [1] B. Julesz. Visual pattern discrimination. *IRE transactions on information theory*, Février 1962
- [2] L. Moisan Periodic plus smooth image decomposition. preprint, MAP5, Université Paris Descartes, 2009.
- [3] B. Galerne, Y. Gousseau et J.-M. Morel, Random Phase Textures : Theory and Synthesis, *IEEE Transactions on Image Processing*, Vol. 20, No. 1, pp. 257-267, 2011
- [4] A.Desolneux, L. Moisan et S.Ronsin, *Vers un texton pour les micro-textures*, 2011

**4.2. Processus de Markov à Décision : suite du stage.** J'ai profité du cours d'Apprentissage par Renforcement de R.Munos pour m'attaquer au modèle robuste de M.Chevé et R.Congar, que je n'avais pas traité lors de mon stage. Les nouveaux outils étudiés lors de ce cours m'ont permis d'envisager de nouvelles techniques de résolution, et d'expliquer plus précisément la théorie sous-jacente.

# Approximation des solutions d'une optimisation de politique économique

---

Nicolas Drougard

# Table des matières

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Introduction</b>  | <b>2</b>  |
| 1.1      | Un exemple discret . . . . .   | 2         |
| 1.2      | L'exemple continu et le premier modèle . . . . .                                   | 3         |
| 1.3      | Le second modèle . . . . .   | 3         |
| <b>2</b> | <b>Calcul de la consommation optimale du premier modèle par différences finies</b> | <b>4</b>  |
| 2.1      | Equation de Programmation Dynamique (HJB) . . . . .                                | 4         |
| 2.2      | Discrétisation en espace de l'équation . . . . .                                   | 6         |
| 2.3      | Algorithme d'Howard et résultats . . . . .   | 7         |
| <b>3</b> | <b>Une autre méthode de résolution</b>   | <b>9</b>  |
| 3.1      | Modèle génératif . . . . .   | 9         |
| 3.2      | Algorithme d'itérations sur les Q-valeurs avec approximation . . . . .             | 9         |
| <b>4</b> | <b>Calcul de la consommation optimale pour le second modèle</b>                    | <b>10</b> |
| 4.1      | Equation de Programmation Dynamique et discrétisation . . . . .                    | 10        |
| 4.2      | Formulation du problème de contrôle de chaîne de Markov associé . . . . .          | 11        |

## 1 Introduction

Morgane Chevé et Ronan Congar se sont penchés sur la mise en place de modèles économiques basés sur des ressources renouvelables<sup>1</sup>. Ces modèles ont pour but de définir une politique de consommation optimale à long terme et robuste à certaines fluctuations.

Pour introduire ces modèles, nous nous appuyerons sur deux exemples simples.

### 1.1 Un exemple discret

L'exploitant d'une réserve de poissons cherche à maximiser la rentabilité de son entreprise. Il tire des bénéfices dans la vente des poissons pêchés, et dans les visites de ses aquariums.

Le nombre de poissons dans sa réserve ne peut pas dépasser  $x_{max}$  : le processus défini par le nombre de poissons en fonction du temps est  $j \in \mathbb{N} \mapsto x_j \in [0, x_{max}]$ . Notons  $\pi_j$  le nombre de poissons pêchés à le jour  $j$ . Supposons que le nombre moyen de poissons naissant le jour  $j$  ne dépende que du nombre de poissons dans la réserve à ce jour-ci, et soit égal à  $f(x_j)$ . La dynamique du système s'exprimerait donc ainsi :

$$x_{j+1} = x_j + [f(x_j) - \pi_j] + \sigma x_j w_j$$

où  $(w_j)_{j \in \mathbb{N}}$  est une suite de variables indépendantes aléatoires de moyennes nulles, représentant les fluctuations présentes dans la reproduction des poissons et  $\sigma > 0$ . Nous remarquons que les fluctuations sont plus importantes lorsque le nombre de poissons présents dans la réserve est grand.

Son bénéfice chaque jour vaut  $U(x_j, \pi_j)$  où  $U$  est une fonction croissante en  $x$  et en  $\pi$  appelée fonction d'utilité (c'est la récompense) : il tire profit à vendre les poissons, ainsi qu'à en avoir une bonne quantité dans les aquariums afin que ses locaux soient visités. Il cherche donc à maximiser la fonction valeur

$$V^\pi(x) = \mathbb{E} \left[ \sum_{j=0}^{+\infty} \gamma^j U(x_j, \pi_j) \mid x_0 = x, \pi \right], \quad 0 < \gamma < 1$$

*i.e.* il cherche la politique de consommation  $\pi$  qui maximise  $V$ .

Comme cet exploitant est capable de compter rapidement le nombre de poissons présents dans sa réserve, il cherche une politique de consommation en "feedback", c'est-à-dire une politique  $\pi$  telle que  $\pi_j$  ne dépende que de la quantité de poissons au jour  $j$  :  $\pi_j = \pi(x_j)$ .

1. Morgane Chevé et Ronan Congar, Uncertainty in Natural Regeneration and the Robust Management of Environmental Resources <http://www.univ-rouen.fr/droit-scenco-gestion/Recherche/CARE/docs/papiers/S20071213.pdf>



## 1.2 L'exemple continu et le premier modèle

Considérons maintenant le responsable d'une entreprise produisant de l'énergie. Ce responsable a monté cette entreprise pour améliorer le confort de ses concitoyens.

Cependant, pour produire cette énergie, il doit rejeter beaucoup de gazs à effet de serre dans l'atmosphère. En outre, cette pollution commence vraiment à être ressentie par la population. Ce responsable s'aperçoit ainsi qu'il a dévié de son but initial : il désire donc prendre en compte ses émissions de gazs dans ses calculs d'optimisation.

Il symbolise l'état de l'environnement (ici la qualité de l'air) par une échelle de 0 à  $x_{max}$  : 0 est le point de non-retour (atmosphère trop polluée pour se régénérer),  $x_{max}$  un air sans pollution. Tout comme dans l'exemple des poissons, nous considérons que la régénération moyenne de l'air ne dépend que de l'état de l'environnement, soit  $f(x_t)$ . La production d'énergie  $\pi$  ( $t \in \mathbb{R}_+ \mapsto \pi_t \in [0, \pi_{max}]$ ) affecte l'environnement de manière linéaire :  $-\varepsilon\pi_t$  avec  $\varepsilon > 0$ .

De plus les fluctuations seront d'autant plus importantes que l'état du système est bon. Nous obtenons alors premier modèle présenté par Morgane Chevé et Ronan Congar :

$$dx_t = [f(x_t) - \varepsilon\pi_t]dt + \sigma x_t dw_t \quad (1)$$

où  $w_t$  est un processus de Wiener,  $\sigma > 0$ , et  $\pi_t$  maximise la fonction valeur

$$V^\pi(x) = E\left[\int_0^\infty e^{-\rho t} U(x_t, \pi_t) dt \mid x_0 = x, \pi\right] \quad (2)$$

avec  $\rho > 0$ . La politique de consommation sera aussi cherchée en feedback,  $\pi_t = \pi(x_t)$ , la mesure de la pollution étant possible.

Montrons que ce modèle définit un processus de décision Markovien lorsque la politique est en feedback :

**Proposition 1.1**  $(x_t)_{t \geq 0}$  est un processus Markovien

Explicitement,

$$x_t = x_0 + \int_0^t [f(x_s) - \varepsilon\pi(x_s)]ds + \int_0^t \sigma x_s dw_s$$

La solution de cette équation différentielle stochastique existe et est unique (au sens des loi de probabilité) si  $x \mapsto [f(x) - \varepsilon\pi(x)]$  est lipschitzienne, du fait que  $x \mapsto \sigma x$  l'est.

Supposons que nous connaissons  $(x_s)_{0 \leq s \leq T}$ , et soit  $t > T$  :

$$\begin{aligned} x_t &= x_T + \int_T^t [f(x_s) - \varepsilon\pi(x_s)]ds + \int_T^t \sigma x_s dw_s \\ &= x_T + \int_0^{t-T} [f(x_{s+T}) - \varepsilon\pi(x_{s+T})]ds + \int_0^{t-T} \sigma x_{s+T} dw_{s+T} \\ &= x_T + \int_0^{t-T} [f(y_s) - \varepsilon\pi(y_s)]ds + \int_0^{t-T} \sigma y_s dw_s \\ &= y_{t-T} \end{aligned}$$

où  $(y_t)_{0 < t}$  est l'unique solution de l'équation différentielle stochastique de condition initiale  $y_0 = x_T$ .

Ainsi, conditionnellement à la tribu du passé (engendrée par  $(x_t)_{0 \leq t \leq T}$ ),  $(x_{t+T})_{0 < t}$  a la même loi que  $(y_t)_{0 < t}$

## 1.3 Le second modèle

Les prévisions des spécialistes des changements climatiques deviennent cependant très pessimiste quant à l'évolution de l'état de l'environnement. Le responsable désire donc rendre son modèle encore plus robuste : il considère qu'il joue contre la Nature dont le but est de minimiser la fonction valeur en

agissant à travers ces fluctuations. Le processus de Wiener  $dw_t$  est alors remplacé par un processus  $dz_t$  défini par

$$dz_t = \nu_t dt + dw_t$$

où  $\nu$  ( $t \in \mathbb{R}_+ \mapsto \nu_t$  est le processus d'action de la Nature, contraint par l'inégalité suivante :

$$\int_0^{+\infty} e^{-\rho t} \frac{1}{2} \nu_t^2 dt \leq \eta, \quad \eta > 0 \quad (3)$$

La dynamique du système devient alors :

$$dx_t = [f(x_t) + \sigma x_t \nu_t - \varepsilon \pi_t] dt + \sigma x_t dw_t \quad (4)$$

et le but du responsable de l'entreprise est de trouver la politique de consommation  $\pi_t$  qui maximise

$$V^\pi(x) = \min_{\nu \in \Omega_\eta} \mathbb{E} \left[ \int_0^{+\infty} e^{-\rho t} U(x_t, \pi_t) dt \mid x_0 = x, \pi, h \right] \quad (5)$$

où  $\Omega_\eta$  est l'ensemble des processus d'action de la Nature qui respectent la contrainte (3).

(3), (4) et (5) définissent alors le second modèle de Morgane Chevé et Ronan Congar.

## 2 Calcul de la consommation optimale du premier modèle par différences finies

### 2.1 Equation de Programmation Dynamique (HJB)

Nous cherchons ici l'équation de programmation dynamique, vérifiée par la fonction valeur optimale  $V^*$  :

$$V^*(x) = \max_{\pi} \mathbb{E} \left[ \int_0^\infty e^{-\rho t} U(x_t, \pi_t) dt \mid x_0 = x, \pi \right]$$

Nous souhaitons montrer que  $V^*$  vérifie :

$$\rho V^*(x) = \max_{\alpha \in \Pi} [U(x, \alpha) + \frac{d}{dx} V^*(x) [f(x) - \varepsilon \alpha] + \frac{1}{2} \frac{d^2}{dx^2} V^*(x) \sigma^2 x^2]$$

où  $\Pi = [0, \pi_{max}]$ .

Soit  $\Delta > 0$ , et supposons qu'il existe une consommation optimale en boucle fermée  $x \in X (= [0, x_{max}]) \mapsto \pi^*(x) \in \Pi$ . Nous avons

$$\begin{aligned} V^*(x) &= \max_{\pi} \mathbb{E} \left[ \int_0^\infty e^{-\rho t} U(x_t, \pi_t) dt \mid x_0 = x, \pi \right] = \mathbb{E} \left[ \int_0^\infty e^{-\rho t} U(x_t, \pi^*(x_t)) dt \mid x_0 = x, \pi^* \right] \\ &= \mathbb{E} \left[ \int_0^\Delta e^{-\rho t} U(x_t, \pi^*(x_t)) dt \mid x_0 = x, \pi^* \right] + \mathbb{E} \left[ \int_\Delta^\infty e^{-\rho t} U(x_t, \pi^*(x_t)) dt \mid x_0 = x, \pi^* \right] \end{aligned} \quad (6)$$

Or,

$$\mathbb{E} \left[ \int_\Delta^\infty e^{-\rho t} U(x_t, \pi^*(x_t)) dt \mid x_0 = x, \pi^* \right] = e^{-\rho \Delta} \mathbb{E} \left[ \int_\Delta^\infty e^{-\rho(t-\Delta)} U(x_t, \pi^*(x_t)) dt \mid x_0 = x, \pi^* \right]$$

par linéarité de l'intégrale et de l'espérance.

Par le changement de variable  $t - \Delta = s$ ,

$$= e^{-\rho \Delta} \mathbb{E} \left[ \int_0^\infty e^{-\rho s} U(x_{s+\Delta}, \pi^*(x_{s+\Delta})) ds \mid x_0 = x, \pi^* \right]$$

et comme  $x_t$  est alors un processus Markovien (appelé diffusion), par la propriété de Markov faible il vient que,

$$\begin{aligned} &= e^{-\rho \Delta} \mathbb{E} \left[ \mathbb{E} \left[ \int_0^\infty e^{-\rho s} U(y_s, \pi^*(y_s)) ds \mid y_0 = x_\Delta, \pi^* \right] \mid x_0 = x, \pi^* \right] \\ &= e^{-\rho \Delta} \mathbb{E} [V^*(x_\Delta) \mid x_0 = x, \pi^*] \end{aligned}$$

En effet, la trajectoire  $(x_{\Delta+t})_{0 \leq t \leq \Delta}$  a la même loi conditionnellement à  $(x_t)_{0 \leq t \leq \Delta}$  que la trajectoire  $(y_t)_{0 \leq t \leq \Delta}$  générée par (1) et de condition initiale  $y_0 = x_\Delta$  ( $x_\Delta$  est aléatoire tout comme  $x_t$ ).

Ainsi (6) devient,

$$\begin{aligned} V^*(x) &= \mathbb{E}\left[\int_0^\Delta e^{-\rho t} U(x_t, \pi^*(x_t)) dt \mid x_0 = x, \pi^*\right] + e^{-\rho\Delta} \mathbb{E}[V^*(x_\Delta) \mid x_0 = x, \pi^*] \\ &= \mathbb{E}\left[\int_0^\Delta e^{-\rho t} U(x_t, \pi^*(x_t)) dt + e^{-\rho\Delta} V^*(x_\Delta) \mid x_0 = x, \pi^*\right] \end{aligned}$$

et nous obtenons,

$$\begin{aligned} 0 &= \frac{1}{\Delta} \mathbb{E}\left[\int_0^\Delta e^{-\rho t} U(x_t, \pi^*(x_t)) dt + e^{-\rho\Delta} V^*(x_\Delta) - V^*(x) \mid x_0 = x, \pi^*\right] \\ &= \mathbb{E}\left[\frac{1}{\Delta} (V^*(x_\Delta) - V^*(x)) \mid x_0 = x, \pi^*\right] + \mathbb{E}\left[\frac{1}{\Delta} \int_0^\Delta e^{-\rho t} U(x_t, \pi^*(x_t)) dt \mid x_0 = x, \pi^*\right] \\ &\quad + \mathbb{E}\left[\frac{e^{-\rho\Delta} - 1}{\Delta} V^*(x_\Delta) \mid x_0 = x, \pi^*\right] \end{aligned}$$

déjà, si  $t \mapsto V^*(x_t)$  est continue, nous avons

$$\lim_{\Delta \rightarrow 0} \left( \frac{e^{-\rho\Delta} - 1}{\Delta} V^*(x_\Delta) \right) = -\rho V^*(x)$$

et si  $t \mapsto U(x_t, \pi^*(x_t))$  est continue,

$$\lim_{\Delta \rightarrow 0} \frac{1}{\Delta} \int_0^\Delta e^{-\rho t} U(x_t, \pi^*(x_t)) dt = U(x, \pi^*(x))$$

Ensuite, la formule d'Itô s'écrit ici, en supposant que  $V^*$  est assez régulière ( $\in C^2(\mathbb{R})$  et bornée par exemple),

$$\begin{aligned} V^*(x_\Delta) - V^*(x) &= \int_0^\Delta \frac{dV^*}{dx}(x_s) [f(x_s) - \varepsilon \pi^*(x_s)] ds + \int_0^\Delta \frac{dV^*}{dx}(x_s) \sigma x_s dw_s \\ &\quad + \frac{1}{2} \int_0^\Delta \frac{d^2 V^*}{dx^2}(x_s) \sigma^2 x_s^2 ds \end{aligned}$$

avec

$$\begin{aligned} \lim_{\Delta \rightarrow 0} \frac{1}{\Delta} \int_0^\Delta \frac{dV^*}{dx}(x_s) [f(x_s) - \varepsilon \pi^*(x_s)] ds &= \frac{dV^*}{dx}(x) [f(x) - \varepsilon \pi^*(x)] \\ \lim_{\Delta \rightarrow 0} \frac{1}{2\Delta} \int_0^\Delta \frac{d^2 V^*}{dx^2}(x_s) \sigma^2 x_s^2 ds &= \frac{1}{2} \frac{d^2 V^*}{dx^2}(x) \sigma^2 x^2 \end{aligned}$$

si les intégrands sont continus, et l'intégrale stochastique est d'espérance nulle :

$$\mathbb{E}\left[\int_0^\Delta \frac{dV^*}{dx}(x_s) \sigma x_s dw_s \mid x_0 = x, \pi^*\right] = 0$$

Par définition l'intégrale stochastique  $\int_0^\Delta \frac{dV^*}{dx}(x_s) \sigma x_s dw_s$  est la limite ( $i \rightarrow +\infty$ ) en moyenne quadratique de  $\sum_k \frac{dV^*}{dx}(x_{t_k^i}) \sigma x_{t_k^i} (w_{t_{k+1}} - w_{t_k})$  qui est d'espérance nulle, où  $((t_k^i)_{0 \leq k \leq i})_{i \in \mathbb{N}}$  est une suite croissante de subdivisions de  $[0, \Delta]$ .

Nous pouvons donc écrire

$$\rho V^*(x) = U(x, \pi^*(x)) + \frac{dV^*}{dx}(x) [f(x) - \varepsilon \pi^*(x)] + \frac{1}{2} \frac{d^2 V^*}{dx^2}(x) \sigma^2 x^2 \quad (7)$$

Maintenant si on remplace la consommation précédente par la consommation  $\pi^\Delta$  égale à  $\alpha$  sur  $[0, \Delta[$  et à la consommation optimale feedback  $\pi^*$  sur  $[\Delta, +\infty[$ , on obtient par optimalité :

$$\begin{aligned} V^*(x) &= \max_{\pi} \mathbb{E} \left[ \int_0^\infty e^{-\rho t} U(x_t, \pi_t) dt \mid x_0 = x, \pi \right] \\ &\geq \mathbb{E} \left[ \int_0^\infty e^{-\rho t} U(x_t, \pi_t^\Delta) dt \mid x_0 = x, \pi^\Delta \right] \\ &= \mathbb{E} \left[ \int_0^\Delta e^{-\rho t} U(x_t, \alpha) dt \mid x_0 = x, \alpha \right] + \mathbb{E} \left[ \int_\Delta^{+\infty} e^{-\rho t} U(x_t, \pi^*(x_t)) dt \mid x_0 = x, \pi^\Delta \right] \end{aligned}$$

par les mêmes raisons que précédemment, le terme de droite devient  $e^{-\rho\Delta} \mathbb{E}[V^*(x_\Delta) \mid x_0 = x, \alpha]$ .

Nous obtenons

$$0 \geq \mathbb{E} \left[ \frac{1}{\Delta} \int_0^\Delta e^{-\rho t} U(x_t, \alpha) dt \mid x_0 = x, \alpha \right] + \mathbb{E} \left[ \frac{e^{-\rho\Delta} - 1}{\Delta} V^*(x_\Delta) \mid x_0 = x, \alpha \right] + \mathbb{E} \left[ \frac{1}{\Delta} (V^*(x_\Delta) - V^*(x)) \mid x_0 = x, \alpha \right]$$

avec

$$\begin{aligned} \lim_{\Delta \rightarrow 0} \frac{1}{\Delta} \int_0^\Delta e^{-\rho t} U(x_t, \alpha) dt &= U(x, \alpha) \\ \lim_{\Delta \rightarrow 0} \frac{e^{-\rho\Delta} - 1}{\Delta} V^*(x_\Delta) &= -\rho V^*(x) \end{aligned}$$

(si  $t \mapsto V^*(x_t)$  et l'intégrand sont continus). Ici, comme le processus état  $t \mapsto x_t$  était sous la consommation  $\alpha$  de 0 à  $\Delta$ , la formule d'Ito nous donne

$$V^*(x_\Delta) - V^*(x) = \int_0^\Delta \frac{dV^*}{dx}(x_s) [f(x_s) - \varepsilon\alpha] ds + \int_0^\Delta \frac{dV^*}{dx}(x_s) \sigma x_s dw_s + \frac{1}{2} \int_0^\Delta \frac{d^2 V^*}{dx^2}(x_s) \sigma^2 x_s^2 ds$$

et pour les mêmes raisons que lorsque la consommation était optimale, nous obtenons  $\forall \alpha \in$  l'espace d'arrivée des contrôles  $\Pi$ ,

$$\rho V^*(x) \geq U(x, \alpha) + \frac{dV^*}{dx}(x) [f(x) - \varepsilon\alpha] + \frac{1}{2} \frac{d^2 V^*}{dx^2}(x) \sigma^2 x^2 \quad (8)$$

En utilisant (7) et (8) on trouve,

$$\rho V^*(x) = \max_{\alpha \in \Pi} [U(x, \alpha) + \frac{dV^*}{dx}(x) [f(x) - \varepsilon\alpha] + \frac{1}{2} \frac{d^2 V^*}{dx^2}(x) \sigma^2 x^2] \quad (9)$$

que l'on appelle équation de programmation dynamique du système.

## 2.2 Discrétisation en espace de l'équation

Notons  $h$  le pas de la discrétisation en temps, et maintenant  $h_1$  celui de la discrétisation en espace. Un schéma de résolution numérique de l'équation de programmation dynamique (ou de Hamilton-Jacobi-Bellman (9)) peut être :

$$\begin{aligned} \rho V_i &= \max_{\alpha} \{ U(x_i, \alpha) \\ &\quad + (f(x_i) - \varepsilon\alpha)_+ \frac{V_{i+1} - V_i}{h_1} + (f(x_i) - \varepsilon\alpha)_- \frac{V_{i-1} - V_i}{h_1} \\ &\quad + \frac{\sigma^2 x_i^2}{2} \frac{V_{i+1} - 2V_i + V_{i-1}}{h_1^2} \} \end{aligned}$$

(approximation des dérivées dans (9) par des taux d'accroissements)

avec :

- $x_i$   $i^{eme}$  élément de  $X^{h_1} := [0, x_{max}] \cap h_1 \cdot \mathbb{N}$ , l'espace discrétisé,
- $V_i = V(x_i)$  la fonction valeur, vecteur de  $R^{|X^{h_1}|}$ ,
- $\alpha$  la consommation dans  $\Pi$
- $(x)_+$  signifie  $\begin{cases} x & \text{si } x > 0 \\ 0 & \text{sinon} \end{cases}$
- $(x)_-$  signifie  $\begin{cases} x & \text{si } x < 0 \\ 0 & \text{sinon} \end{cases}$

On regroupe les  $V_i$  :

$$\rho V_i = \max_{\alpha} \{U(x_i, \alpha) - V_i(\frac{|f(x_i) - \varepsilon\alpha|}{h_1} + \frac{\sigma^2 x_i^2}{h_1^2}) + V_{i+1}(\frac{(f(x_i) - \varepsilon\alpha)_+}{h_1} + \frac{\sigma^2 x_i^2}{2h_1^2}) + V_{i-1}(\frac{(f(x_i) - \varepsilon\alpha)_-}{h_1} + \frac{\sigma^2 x_i^2}{2h_1^2})\}$$

On multiplie par  $h > 0$ , le pas de discrétisation en temps, et on ajoute  $V_i$  à chaque membre :

$$\begin{aligned} V_i &= (h\rho + 1)^{-1} \max_{\alpha} \{hU(x_i, \alpha) + V_i(1 - \frac{h|f(x_i) - \varepsilon\alpha|}{h_1} - \frac{h\sigma^2 x_i^2}{h_1^2}) \\ &\quad + V_{i+1}(\frac{h(f(x_i) - \varepsilon\alpha)_+}{h_1} + \frac{h\sigma^2 x_i^2}{2h_1^2}) + V_{i-1}(\frac{h(f(x_i) - \varepsilon\alpha)_-}{h_1} + \frac{h\sigma^2 x_i^2}{2h_1^2})\} \end{aligned} \quad (10)$$

Supposons que

$$\frac{h}{h_1} \sup_{i \in \{1, \dots, |X^{h_1}|\}, \alpha \in \Pi} \{|f(x_i) - \varepsilon\alpha| + \frac{\sigma^2 x_i^2}{h_1}\} \leq 1 \quad (11)$$

La condition (11) assure que, dans la formule (10), le poids de  $V_i$  est positif. De plus, la somme de ces poids vaut 1.

Nous pouvons donc interpréter l'équation (10) comme l'équation de programmation dynamique d'un problème de contrôle de chaîne de Markov avec ici

- $\gamma = (1 + \rho h)^{-1}$  le coefficient d'actualisation,
- l'espace d'état  $X^{h_1}$ ,
- l'espace des politiques de consommation  $\Pi$ ,
- la fonction d'utilité  $u = hU$ , avec la notation  $u(x_i, \alpha) = u_i(\alpha)$ .

et

- $\forall i \in \{1, \dots, |X^{h_1}|\}, P_{ii}(\alpha) = 1 - \frac{h|f(x_i) - \varepsilon\alpha|}{h_1} - \frac{h\sigma^2 x_i^2}{h_1^2}$
- $\forall i \in \{1, \dots, |X^{h_1}| - 1\}, P_{i,i+1}(\alpha) = \frac{h(f(x_i) - \varepsilon\alpha)_+}{h_1} + \frac{h\sigma^2 x_i^2}{2h_1^2}$
- $\forall i \in \{2, \dots, |X^{h_1}|\}, P_{i,i-1}(\alpha) = \frac{h(f(x_i) - \varepsilon\alpha)_-}{h_1} + \frac{h\sigma^2 x_i^2}{2h_1^2}$

les autres coefficients de  $P(\alpha)$  étant nuls.

Ainsi, l'équation se réécrit

$$V = \gamma \max_{\alpha} \{u(\alpha) + P(\alpha)V\} = TV$$

avec  $T$  l'opérateur de Bellman. Nous retrouvons donc l'équation de Bellman d'un Processus de Markov Discret avec les notations du cours<sup>2</sup>. Les algorithmes d'itération sur les valeurs, et d'itération sur les politiques permettent alors le calcul de  $V^*$  et  $\pi^*$ .

Cependant, nous remarquons que la matrice  $P$  définie précédemment n'est pas stochastique : les coefficients de la première ainsi que ceux de la dernière n'ont pas pour somme 1. Nous pouvons choisir d'ajuster les coefficients de  $P$  pour qu'ils respectent les hypothèses des algorithmes, ou bien rester proche du modèle : les résultats sont similaires.

## 2.3 Algorithme d'Howard et résultats

Nous implémentons l'algorithme d'itération sur les politiques (Howard) décrit dans le cours :  $\pi_0$  politique feedback constante, et

1) Evaluation de la politique  $\pi_k$  : on calcule  $V^{\pi_k} = u(\pi_k) + P(\pi_k)V^{\pi_k}$

2) Amélioration de la politique : on calcule  $\pi_{k+1}$  déduite de  $V^{\pi_k} : \forall i \in \{1, \dots, |X^{h_1}|\},$

$$\pi_{k+1}(x_i) \in \operatorname{argmax}_{\alpha \in \Pi} [u_i(\alpha) + \gamma((P_{i,i-1}(\alpha)V_{i-1}^{\pi_k})\mathbb{1}_{(i \neq 1)} + P_{i,i}(\alpha)V_i^{\pi_k} + (P_{i,i+1}(\alpha)V_{i+1}^{\pi_k})\mathbb{1}_{(i \neq |X^{h_1}|)})]$$

On s'arrête quand  $V^{\pi_k} = V^{\pi_{k+1}}$ .

L'étape 1) requiert une inversion de matrice. En effet,  $V^{\pi_k} = (Id + P(\pi_k))^{-1}u(\pi_k)$ . Puisque  $Id + P(\pi_k)$  est tridiagonale (voir définition de  $P$ ), l'algorithme de Thomas<sup>3</sup> permet un calcul de complexité linéaire

2. Rémi Munos, Introduction à l'apprentissage par renforcement <http://researcher.lille.inria.fr/~munos/master-mva/>

3. Paola GOATIN. Analyse Numérique. <http://www-sop.inria.fr/members/Paola.Goatin/anum.pdf>

en  $|X^{h_1}|$ .

Pour implémenter l'étape 2), l'espace des politiques de consommation doit être discrétisé : nous introduisons donc le pas de discrétisation  $h_2$ , et  $\Pi^{h_2} = \Pi \cap h_2.\mathbb{N}$ . Ainsi, l'argmax est cherché sur  $\alpha \in \Pi^{h_2}$ .

Dans le papier de M.Chevé et R.Congar, une solution analytique est calculée : elle nous permet de visualiser graphiquement la performance de cette méthode.

Cette solution est calculée pour :

- $U(x, \alpha) = \ln(x) + \ln(\alpha)$
- $f(x) = x \ln(\frac{x_{max}}{x})$

Pour ces hypothèses, la fonction valeur optimale et la politique de consommation optimale sont respectivement :

$$V^*(x) = a \ln(x) + b \quad (12)$$

$$\pi^*(x) = \frac{(\rho + 1)}{2\varepsilon} x \quad (13)$$

avec

- $a = \frac{2}{\rho+1}$
- $b = \frac{1}{\rho}(\log(x_{max})\frac{2}{\rho+1} - \log(\varepsilon) - \log(a) - 1 - \frac{\sigma^2}{\rho+1})$

La fonction valeur optimale calculée analytiquement (12) et celle résultant de la méthode numérique sont représentées figure 1 en fonction de l'état initial. La politique de consommation optimale calculée analytiquement (13) sont représentées figure 2 en fonction de l'état courant.

Les constantes utilisées sont :

- $\varepsilon = 0.5$
- $\sigma = 0.2$
- $\rho = 1$
- $x_{max} = 10$
- $|X^{h_1}| = 200$
- $h = 10^{-5}$
- $\pi_{max} = 50$
- $|\Pi^{h_2}| = 400$

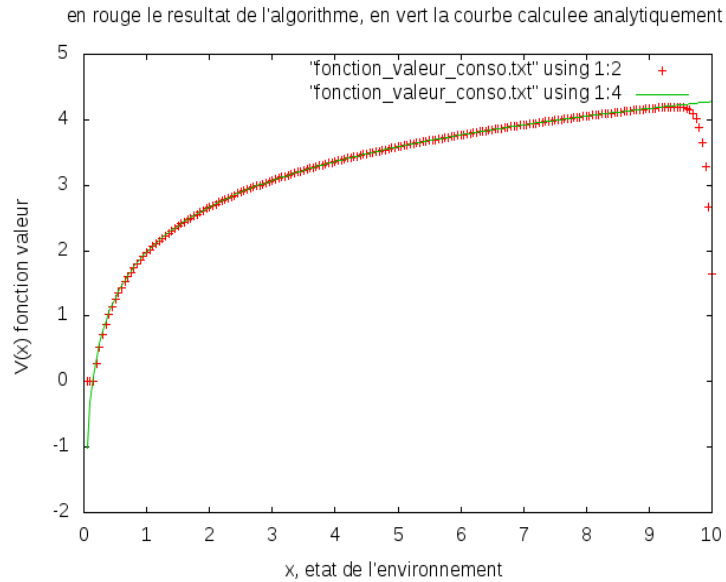


FIGURE 1 – Fonction valeur : les points rouges sont calculés par l'algorithme d'Howard, la courbe verte est le résultat calculé analytiquement (gnuplot)

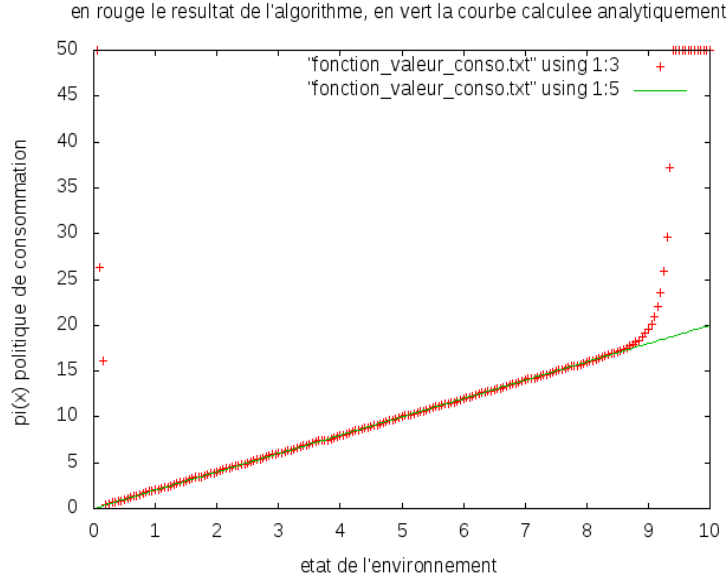


FIGURE 2 – Politique de consommation : les points rouges sont calculés par l'algorithme d'Howard, la courbe verte est le résultat calculé analytiquement (gnuplot)

Nous observons des "effets de bords" qui semblent dûs à la discrétisation avec les différences finies.

### 3 Une autre méthode de résolution

#### 3.1 Modèle génératif

Cette méthode nécessite de pouvoir générer des trajectoires. Pour celà, notre modèle doit être discrétisé en temps : pour  $h$  assez petit, le processus discret défini par

$$x_{t+1} = x_t + (f(x_t) - \varepsilon\pi_t) + \sqrt{h}\sigma x_t w_t \quad (14)$$

avec  $w_k$  valant 1 ou  $-1$  avec probabilité  $\frac{1}{2}$ , a un comportement similaire au processus continu

$$dx_t = [f(x_t) - \varepsilon\pi_t]dt + \sigma x_t dw_t$$

#### 3.2 Algorithme d'itérations sur les Q-valeurs avec approximation

Comme l'espace d'état est infini, les algorithmes usuels (IP,IV) ne peuvent s'appliquer pour calculer la fonction valeur. Cependant il est possible de calculer une approximation de la fonction valeur optimale en utilisant l'algorithme présenté dans le cours : la fonction  $Q$ -valeur optimale, définie par

$$\sum_y p(y|x, a)[r(x, a, y) + \gamma V^*(y)]$$

est le point fixe de l'opérateur de Bellman  $\mathcal{T}$ , pour les fonctions  $Q$  définies sur  $[0, x_{max}] \times [0, \pi_{max}]$  :

$$\mathcal{T}Q(x, a) = \sum_y p(y|x, a)[r(x, a, y) + \gamma \max_b Q(y, b)]$$

Notons  $\mathcal{F}$  l'espace vectoriel des fonctions définies sur  $[0, x_{max}] \times [0, \pi_{max}]$  engendrées par les fonctions de base  $\phi_1, \dots, \phi_d$ .

$$\mathcal{F} = \left\{ Q_\alpha(x, a) = \sum_{j=1}^d \alpha_j \phi_j(x, a) \mid \alpha \in \mathbb{R}^d \right\} \quad (15)$$

Nous cherchons à calculer une approximation empirique de

$$Q_{k+1} = \operatorname{argmin}_{Q \in \mathcal{F}} \mathbb{E}[(Q(X, A) - \mathcal{T}Q_k(X, A))^2] \quad (16)$$

où  $X$  suit une loi  $\mu$ , et  $A$  une distribution uniforme sur  $[0, \pi_{max}]$ . L'algorithme suivant permet cette approximation :

1. Simuler un  $n$ -échantillon  $(X_i, A_i)_{1 \leq i \leq n}$  selon la loi  $\mu \times u$  (où  $u$  est la distribution uniforme).
2. Calculer pour chaque  $i$  une approximation empirique de  $TQ_k(X_i, A_i)$  :  
simulons pour cela  $M$  variables aléatoires  $(Y_i^m, R_i^m)_{1 \leq m \leq M}$  où  $Y_i \sim p(\cdot | X_i, A_i)$  (donné par 14 ).  
On définit

$$Z_i = \frac{1}{M} \sum_{m=1}^M R_i^m + \gamma \max_b Q_k(Y_i^m, b) \quad (17)$$

$Z_i$  converge par la loi faible des grands nombres vers  $TQ_k(X_i, A_i)$

3. Résolution du problème d'apprentissage :  
Nous minimisons alors le risque empirique en  $Q_\alpha$ , i.e.  $\frac{1}{n} \sum_{i=1}^n (Q_\alpha(X_i, A_i) - Z_i)^2$ . Il s'agit d'un problème aux moindres carrés et la valeur de  $\alpha$  optimale est donnée par

$$\hat{\alpha} = (M' M)^{-1} M' Z \quad (18)$$

où  $M(i, j) = \phi_j(X_i, A_i)$

## 4 Calcul de la consommation optimale pour le second modèle

### 4.1 Equation de Programmation Dynamique et discrétisation

Nous rappelons le second modèle :

$$dx_t = [f(x_t) + \sigma x_t \nu_t - \varepsilon \pi_t] dt + \sigma x_t dw_t$$

où  $t \in \mathbb{R}_+ \mapsto \nu_t$  est l'action malveillante de la Nature.

Nous cherchons une politique de consommation feedback  $\pi^*$  telle que

$$V^{\pi^*}(x) = \max_{\pi} \min_{\nu \in \Omega_\eta} \mathbb{E} \left[ \int_0^{+\infty} e^{-\rho t} U(x_t, \pi_t) dt | x_0 = x, \pi, \nu \right]$$

où  $\Omega_\eta$  est l'ensemble des processus d'action de la Nature  $t \in \mathbb{R}_+ \mapsto \nu_t$  qui respectent l'inégalité :

$$\int_0^{+\infty} e^{-\rho t} \frac{1}{2} \nu_t^2 dt \leq \eta, \quad \eta > 0$$

Il est possible de réécrire ce problème comme suit :

$$\begin{aligned} V^{\pi^*}(x) &= \max_{\pi} \min_{\nu} \mathbb{E} \left[ \int_0^{+\infty} e^{-\rho t} U(x_t, \pi_t) dt | x_0 = x, \pi, \nu \right] + \theta \mathbb{E} \left[ \int_0^{+\infty} e^{-\rho t} \frac{1}{2} \nu_t^2 dt \right] \\ &= \max_{\pi} \min_{\nu} \mathbb{E} \left[ \int_0^{+\infty} e^{-\rho t} \left( U(x_t, \pi_t) + \frac{\theta}{2} \nu_t^2 \right) dt | x_0 = x, \pi, \nu \right] \end{aligned}$$

où  $\theta > 0$  est le multiplicateur de Lagrange associé à la contrainte  $\nu \in \Omega_\eta$ . Ainsi, plus  $\theta$  est petit, plus le modèle est robuste.

Une équation de Programmation Dynamique est présentée dans le papier de M.Chevé et R.Congar :

$$\rho V(x) = \max_{\alpha \in \Pi} \min_{\nu} \left\{ U(x, \alpha) + \frac{\theta}{2} \nu^2 + (f(x) + \sigma x \nu - \varepsilon \alpha) \frac{dV}{dx}(x) + \frac{1}{2} \sigma^2 x^2 \frac{d^2 V}{dx^2}(x) \right\} \quad (19)$$

Comme en section 2.2, notons  $h$  le pas de la discrétisation en temps, et  $h_1$  celui de la discrétisation en espace. Le même schéma de résolution numérique, mais issu de l'équation (19) est :

$$\begin{aligned} \rho V_i &= \max_{\alpha} \min_{\nu} \left\{ U(x_i, \alpha) + \frac{\theta}{2} \nu^2 \right. \\ &\quad \left. + (f(x_i) + \sigma x_i \nu - \varepsilon \alpha) \frac{V_{i+1} - V_i}{h_1} + (f(x_i) + \sigma x_i \nu - \varepsilon \alpha) \frac{V_{i-1} - V_i}{h_1} \right. \\ &\quad \left. + \frac{\sigma^2 x_i^2}{2} \frac{V_{i+1} - 2V_i + V_{i-1}}{h_1^2} \right\} \end{aligned}$$



avec les mêmes notations qu'en section 2.2.

Regroupement des  $V_i$  :

$$\rho V_i = \max_{\alpha} \min_{\nu} \left\{ U(x_i, \alpha) + \frac{\theta}{2} \nu^2 - V_i \left( \frac{|f(x_i) + \sigma x_i \nu - \varepsilon \alpha|}{h_1} + \frac{\sigma^2 x_i^2}{h_1^2} \right) \right. \\ \left. + V_{i+1} \left( \frac{(f(x_i) + \sigma x_i \nu - \varepsilon \alpha)_+}{h_1} + \frac{\sigma^2 x_i^2}{2h_1^2} \right) + V_{i-1} \left( \frac{(f(x_i) + \sigma x_i \nu - \varepsilon \alpha)_-}{h_1} + \frac{\sigma^2 x_i^2}{2h_1^2} \right) \right\}$$

Multiplication par  $h > 0$ , ajout de  $V_i$  à chaque membre :

$$V_i = (h\rho + 1)^{-1} \max_{\alpha} \min_{\nu} \left\{ hU(x_i, \alpha) + h\frac{\theta}{2} \nu^2 \right. \\ \left. + V_i \left( 1 - \frac{h|f(x_i) + \sigma x_i \nu - \varepsilon \alpha|}{h_1} - \frac{h\sigma^2 x_i^2}{h_1^2} \right) \right. \\ \left. + V_{i+1} \left( \frac{h(f(x_i) + \sigma x_i \nu - \varepsilon \alpha)_+}{h_1} + \frac{h\sigma^2 x_i^2}{2h_1^2} \right) \right. \\ \left. + V_{i-1} \left( \frac{h(f(x_i) + \sigma x_i \nu - \varepsilon \alpha)_-}{h_1} + \frac{h\sigma^2 x_i^2}{2h_1^2} \right) \right\} \quad (20)$$

Supposons que

$$\frac{h}{h_1} \sup_{(i \in \{1, \dots, |X^{h_1}|\}, \alpha \in \Pi, \nu)} \left\{ |f(x_i) + \sigma x_i \nu - \varepsilon \alpha| + \frac{\sigma^2 x_{|X^{h_1}|}^2}{h_1} \right\} \leq 1 \quad (21)$$

La condition (21) assure que, dans la formule (20), le poids de  $V_i$  est positif. De plus, la somme de ces poids vaut 1.

Ainsi, en notant :

- $r(\alpha, \nu) \in \mathbb{R}^{|X^{h_1}|}$ , tel que  $r_i(\alpha, \nu) = r(x_i, \alpha, \nu) = hU(x_i, \alpha) + h\frac{\theta}{2} \nu^2$
- $P \in \mathbb{R}^{|X^{h_1}| \times |X^{h_1}|}$ , telle que :
- $\forall i \in \{2, \dots, |X^{h_1}| - 1\}$ ,  $\begin{cases} P_{i,i-1} = \frac{h(f(x_i) + \sigma x_i \nu - \varepsilon \alpha)_-}{h_1} + \frac{h\sigma^2 x_i^2}{2h_1^2} \\ P_{i,i} = \left( 1 - \frac{h|f(x_i) + \sigma x_i \nu - \varepsilon \alpha|}{h_1} - \frac{h\sigma^2 x_i^2}{h_1^2} \right) \\ P_{i,i+1} = \frac{h(f(x_i) + \sigma x_i \nu - \varepsilon \alpha)_+}{h_1} + \frac{h\sigma^2 x_i^2}{2h_1^2} \end{cases}$
- $\begin{cases} P_{1,2} = \frac{h(f(x_1) + \sigma x_1 \nu - \varepsilon \alpha)_+}{h_1} + \frac{h\sigma^2 x_1^2}{2h_1^2} \\ P_{1,1} = 1 - P_{1,2} \end{cases}$
- $\begin{cases} P_{|X^{h_1}|, |X^{h_1}| - 1} = \frac{h(f(x_{|X^{h_1}|}) + \sigma x_{|X^{h_1}|} \nu - \varepsilon \alpha)_+}{h_1} + \frac{h\sigma^2 x_{|X^{h_1}|}^2}{2h_1^2} \\ P_{|X^{h_1}|, |X^{h_1}|} = 1 - P_{|X^{h_1}|, |X^{h_1}| - 1} \end{cases}$
- $0 < \gamma = (1 + \rho h)^{-1} < 1$
- $V \in \mathbb{R}^{|X^{h_1}|}$  le vecteur "fonction valeur", tel que  $V_i = V(x_i)$

l'équation (19) devient :

$$V = \gamma \max_{\alpha} \min_{\nu} \{r(\alpha, \nu) + P(\alpha, \nu)V\} \quad (22)$$

## 4.2 Formulation du problème de contrôle de chaîne de Markov associé

Nous pouvons maintenant définir l'opérateur de Bellman  $T$  :

$$TV = \gamma \max_{\alpha} \min_{\nu} \{r(\alpha, \nu) + P(\alpha, \nu)V\} \quad (23)$$

**Proposition 4.1** *L'opérateur de Bellman ainsi défini est contractant pour la norme sup*

Soit  $x_i \in X^{h_1}$ ,  $W_1$  et  $W_2$  deux fonctions valeurs  $\in \mathbb{R}^{|X^{h_1}|}$

Pour montrer cette proposition, nous avons besoin de rappeler deux propriétés des fonctions min et max : (24) et (26).

Soient  $f_1$  et  $f_2$  deux fonctions de  $\omega$ .

$$|\max_{\omega} f_1(\omega) - \max_{\omega} f_2(\omega)| \leq \max_{\omega} |f_1(\omega) - f_2(\omega)| \quad (24)$$

car

$$\begin{aligned} \max_{\omega} f_1(\omega) &= \max_{\omega} \{f_1(\omega) - f_2(\omega) + f_2(\omega)\} \\ &\leq \max_{\omega} \{f_1(\omega) - f_2(\omega)\} + \max_{\omega} f_2(\omega) \\ &\leq \max_{\omega} |f_1(\omega) - f_2(\omega)| + \max_{\omega} f_2(\omega) \end{aligned}$$

et comme cela reste vrai en partant de  $f_2$ , nous obtenons (24).

Avec  $(f_k(\alpha))_{k=1,2} = \min_{\nu} \{r_i(\alpha, \nu) + \sum_j P_{i,j}(\alpha, \nu) W_k(x_j)\}$ , nous obtenons

$$\begin{aligned} |TW_1(x_i) - TW_2(x_i)| &= \gamma |\max_{\alpha \in \Pi} \min_{\nu} \{r_i(\alpha, \nu) + \sum_j P_{i,j}(\alpha, \nu) W_1(x_j)\} \\ &\quad - \max_{\alpha \in \Pi} \min_{\nu} \{r_i(\alpha, \nu) + \sum_j P_{i,j}(\alpha, \nu) W_2(x_j)\}| \\ &\leq \gamma \max_{\alpha \in \Pi} |\min_{\nu} \{r_i(\alpha, \nu) + \sum_j P_{i,j}(\alpha, \nu) W_1(x_j)\} \\ &\quad - \min_{\nu} \{r_i(\alpha, \nu) + \sum_j P_{i,j}(\alpha, \nu) W_2(x_j)\}| \end{aligned} \quad (25)$$

Maintenant,

$$|\min_{\omega} f_1(\omega) - \min_{\omega} f_2(\omega)| \leq \max_{\omega} |f_1(\omega) - f_2(\omega)| \quad (26)$$

car

$$\begin{aligned} \forall \omega, f_1(\omega) - f_2(\omega) &\leq \max_{\omega} |f_1(\omega) - f_2(\omega)| \\ \Rightarrow \forall \omega, f_1(\omega) &\leq \max_{\omega} |f_1(\omega) - f_2(\omega)| + f_2(\omega) \\ \Rightarrow \min_{\omega} f_1(\omega) &\leq \max_{\omega} |f_1(\omega) - f_2(\omega)| + \min_{\omega} f_2(\omega) \end{aligned}$$

et en inversant les rôles de  $f_1$  et  $f_2$ , nous obtenons (26).

Fixons  $\alpha \in \Pi$  : avec  $(f_k(\nu))_{k=1,2} = r_i(\alpha, \nu) + \sum_j P_{i,j}(\alpha, \nu) W_k(x_j)$ , nous obtenons

$$\begin{aligned} |\min_{\nu} \{r_i(\alpha, \nu) + \sum_j P_{i,j}(\alpha, \nu) W_1(x_j)\} - \min_{\nu} \{r_i(\alpha, \nu) + \sum_j P_{i,j}(\alpha, \nu) W_2(x_j)\}| \\ \leq \max_{\nu} |\sum_j P_{i,j}(\alpha, \nu) W_1(x_j) - \sum_j P_{i,j}(\alpha, \nu) W_2(x_j)| \\ = \max_{\nu} |\sum_j P_{i,j}(\alpha, \nu) (W_1(x_j) - W_2(x_j))| \\ \leq \|W_1 - W_2\|_{\infty} \end{aligned}$$

puisque  $P$  est une matrice stochastique.

Nous obtenons enfin, par croissance de la fonction maximum (si  $\forall \omega, f_1(\omega) \leq f_2(\omega)$ ,  $\max_{\omega}(f_1(\omega)) \leq \max_{\omega}(f_2(\omega))$ ),

$$(25) \leq \gamma \max_{\alpha} (\|W_1 - W_2\|_{\infty}) = \gamma \|W_1 - W_2\|_{\infty}$$

Nous avons donc bien  $\forall x_i \in X^{h_1}$ ,

$$|TW_1(x_i) - TW_2(x_i)| \leq \gamma \|W_1 - W_2\|_\infty$$

*i.e.*  $T$  est un opérateur contractant pour la norme uniforme.

Nous pouvons donc implémenter l'algorithme d'itération sur les valeurs ( $V^*$  est un point fixe de  $T$ , donc  $T^k V^0 \rightarrow V^*$  lorsque  $k \rightarrow +\infty$  par le théorème du point fixe), ce qui termine ce rapport.

## Bibliographie

- Morgane Chevé et Ronan Congar, Uncertainty in Natural Regeneration and the Robust Management of Environmental Resources, Février 2007.

- Harold J. Kushner et Paul G. Dupuis, Numérical Methods for Stochastic Control Problems in Continuous Time.

- Harold J. Kushner, Probability Methods for Approximations in Stochastic Control and for Elliptic Equations.

- N.V.Krylov, Controlled Diffusion Processes.

- Wendell H. Fleming et H. Mete Soner. Controlled Markov Processes and Viscosity Solutions.

- Wendell H. Fleming et Raymond W. Rishel. Deterministic and Stochastic Optimal Control.

- Howard. Dynamic Probabilistic Systems Volume II : Semimarkov and Decision Processes.

- R. E. Bellman et S. E. Dreyfus. La Programmation Dynamique et ses Applications.

4.3. **PhyloHMM.** Plusieurs projets à réaliser en groupe étaient proposés dans le cadre du cours « Modèles Graphiques Probabilistes » : nous sommes trois élèves à avoir choisi de travailler sur une modélisation de l'évolution génétique grâce à un modèle hybride mêlant outils phylogénétiques et chaînes de Markov cachées.

# Phylogénétique Hidden Markov Model

Edwin Carlinet, Nicolas Drougard, Matthieu Lê

5 janvier 2012

## Table des matières

|          |   |          |
|----------|---|----------|
| <b>1</b> | <b>Introduction</b>   | <b>1</b> |
| <b>2</b> | <b>Modèles Phylo-HMM</b>  | <b>1</b> |
| <b>3</b> | <b>Définition Formelle</b>  | <b>2</b> |
| <b>4</b> | <b>Application 1 : A Toy Gene Finder</b>                          | <b>4</b> |
| 4.1      | Génération des données . . . . .                                  | 4        |
| 4.2      | Identification des gènes . . . . .                                | 5        |
| 4.2.1    | L'algorithme de Viterbi . . . . .                                 | 5        |
| 4.3      | L'algorithme de Felsenstein . . . . .                             | 6        |
| 4.4      | Evaluation . . . . .  | 6        |
| <b>5</b> | <b>Application 2 : Identification of Highly Conserved Regions</b> | <b>7</b> |
| 5.1      | Définition du Modèle . . . . .                                    | 7        |
| 5.2      | Génération des Données . . . . .                                  | 8        |
| 5.3      | Calcul des Probabilités Posterieures . . . . .                    | 8        |
| <b>6</b> | <b>Conclusion</b>   | <b>9</b> |

## 1 Introduction

Le phylo-HMM (Phylogenetic Hidden Markov Models) sont des modèles probabilistes qui considèrent les évolutions moléculaires comme une combinaison de deux processus de Markov. Le premier processus modélise les différences génomiques spatiales, le deuxième modélise les transformations dans le temps afin de comprendre la structure et l'évolution d'une séquence. Ces modèles peuvent être appliqués à des problèmes tels que la prédiction de gènes ou l'identification d'éléments conservés.

## 2 Modèles Phylo-HMM

Un phylo-HMM génère des alignements, colonne par colonne, où chaque colonne est définie par un modèle phylogénétique et le passage d'une colonne à l'autre est réalisé selon un HMM.

Les modèles phylogénétiques considérés ici sont des processus stochastiques qui agissent indépendamment sur chaque site du génome. On procède comme suit :

- Un caractère (par exemple A, B, C ou D) est généré aléatoirement et placé à la racine de l'arbre.
- Sur chaque branche de l'arbre, les substitutions se réalisent suivant le modèle défini par l'arbre de la racine jusqu'aux feuilles.
- Les caractères qui subsistent sur les feuilles définissent ainsi une colonne.

Ainsi, un modèle phylogénétique génère des alignements de colonnes avec une structure de corrélation qui reflète le processus de substitution. Les différents modèles phylogénétiques utilisés permettent de modéliser différents taux de substitution (par exemple ceux des régions conservées ou non conservées, codantes ou non codantes).

Pour les HMM, on a typiquement à disposition une séquence d'observation. Cependant, la séquence des états par lesquels le processus est passé reste cachée. On utilise les algorithmes classiques de HMM pour calculer le chemin d'états qui maximise la vraisemblance, la probabilité a posteriori que tout état a généré une observation, la probabilité totale des observations considérant tous les chemins possibles.

### 3 Définition Formelle

Un phylo-HMM est défini par  $\theta = (S, \psi, A, b)$  un quadruplet composé :

- $S = \{s_1, \dots, s_M\}$  un ensemble d'état.
- $\psi = \{\psi_1, \dots, \psi_M\}$  un ensemble de modèles phylogénétiques associés à ces états.
- $A = \{a_{j,k}\}, 1 \leq j, k \leq M$  une matrice de probabilités de transition d'états,  $a_{j,k}$  est la probabilité de visiter l'état  $k$  à un site  $i$  sachant qu'on a visité l'état  $j$  à l'état  $i - 1$ .
- $b = (b_1, \dots, b_M)$  le vecteur de probabilités des états initiaux,  $b_j$  est la probabilité que l'état  $j$  soit visité en premier.

On note  $X$  la matrice des alignements composée de  $L$  colonnes (sites) et  $n$  lignes (une par espèce). On note  $X_i$  ( $1 \leq i \leq L$ ) la  $i$ -ème colonne. Ainsi, à chaque étape, on visite un état  $s_j$  défini par un modèle phylogénétique  $\psi_j = (Q_j, \pi_j, \tau_j, \beta_j)$  où  $Q_j$  est la matrice des taux de substitution,  $\pi_j$  est un vecteur de fréquence,  $\tau_j$  un arbre binaire et  $\beta_j$  un ensemble de branches. Le modèle est défini suivant un alphabet  $\Sigma$  (par exemple  $\{A, C, G, T\}$ ) de taille  $d$ . Dans ce projet,  $Q_j$  est de dimension  $d \times d$  et  $\pi$  est de dimension  $d$ . L'arbre  $\tau_j$  a  $n$  feuilles. Les éléments de  $\beta_j$  sont associés aux branches de l'arbre. On note :

- $P(X_i|\psi_j)$  la probabilité qu'une colonne  $X_i$  soit émise par l'état  $s_j$ .
- La probabilité de substitution d'une base  $a$  par une base  $b$  appartenant à  $\Sigma$  le long d'une branche de longueur  $t$  ainsi notée  $P(b|a, t, \psi_j)$ . Cette probabilité est basée sur un modèle de Markov à temps continu où le taux de substitution d'une nucléotide par une autre est défini par une matrice de taux  $Q$ . Ainsi à chaque noeud  $j$  de l'arbre phylogénétique, on associe la matrice  $d \times d : P_j(t) = \exp(Q_j t)$ .

Différents modèles de substitutions pour  $Q$  ont été proposé (Siepel and Haussler, 2004a). Le premier - le plus général - est le modèle *UNR* à 11 paramètres libres. En imposant la contrainte de réversibilité,  $\pi_i q_{i,j} = \pi_j q_{j,i}$  le nombre de paramètre tombe à 5. Enfin, le modèle le plus utilisé et que nous utilisons dans nos expérimentations est la *HKY* (Hasegawa et al., 1985). Celui-ci comporte un seul paramètre  $K$  qui est le rapport transition/transversion. En effet, à cause des propriétés physico-chimiques des bases azotées, la probabilité de substitution entre pyrimidine et purine (transversion) est moins forte que celle des substitutions *pyrimidine*  $\leftrightarrow$  *pyrimidine* ou *purine*  $\leftrightarrow$  *purine*.  $Q_j$  prend ainsi la forme :

$$Q_j = \begin{pmatrix} - & \pi_{C_j} & K_j \pi_{G_j} & \pi_{T_j} \\ \pi_{A_j} & - & \pi_{G_j} & K_j \pi_{T_j} \\ K_j \pi_{A_j} & \pi_{C_j} & - & K_j \pi_{T_j} \\ \pi_{A_j} & K_j \pi_{C_j} & \pi_{G_j} & - \end{pmatrix}$$

où  $\pi_j = (\pi_{A_j}, \pi_{C_j}, \pi_{G_j}, \pi_{T_j})$  et  $K_j$  représente le taux de transition/transversion du modèle  $\psi_j$ . En tant que matrice de Markov à temps continu, la somme par ligne est contrainte à 0, le symbole  $-$  doit donc être remplacé pour remplir cette contrainte. La Figure 1 montre un exemple d'évolution de probabilité de substitution au cours du temps. Pour calculer  $Q_j$  de manière efficace sans passer par le développement de Taylor  $\exp(Q_j t) = \sum_{k=0}^{+\infty} \frac{(Q_j \cdot t)^k}{k!}$ , on utilise la décomposition  $Q_j = S_j \cdot D_j \cdot S_j^{-1}$  où  $S_j$  est la matrice composée des vecteurs propres de  $Q_j$  et  $D_j$  la matrice

diagonale composée des valeurs propres  $\{\lambda_j^i\}$ . L'exponentiation se simplifie alors en  $\exp(Q_j t) = S \cdot \exp(D_j t) \cdot S^{-1}$  avec  $\exp(D_j t)_{i,i} = e^{\lambda_j^i \cdot t}$ .

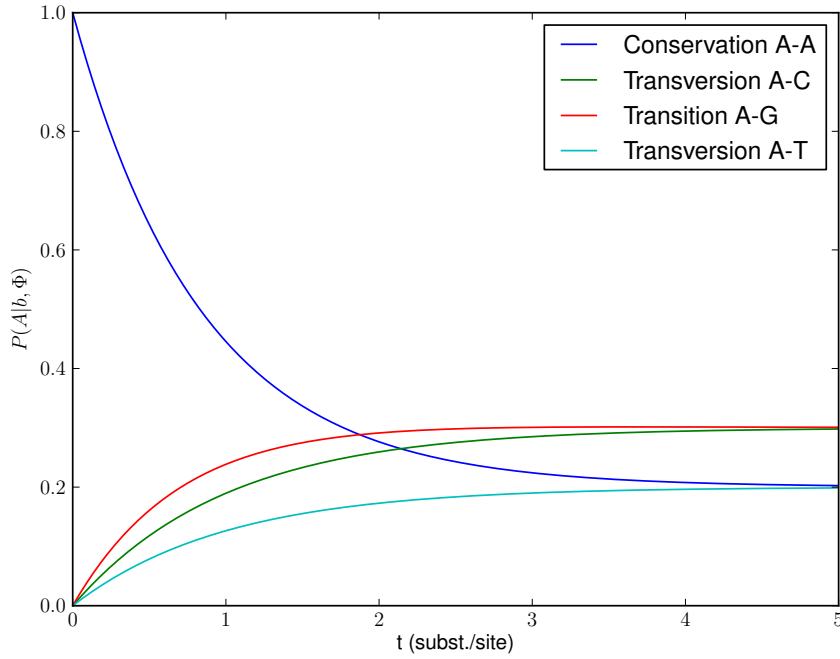


FIGURE 1 – Evolution de la substitution de l'adénine au cours du temps. Le vecteur de fréquences initiales est  $\pi = (\pi_A = 0.2, \pi_C = 0.3, \pi_G = 0.3, \pi_T = 0.2)$ . Vers  $+\infty$ , les courbes de substitutions convergent vers le vecteur de configuration initial. A noter que pour une même configuration initiale ( $C$  et  $G$ ), choisir  $K > 1$  (ici  $K = 1.7$ ) favorise la transition  $A \leftrightarrow G$  au dépit de la transversion  $A \leftrightarrow C$

La vraisemblance est donnée par

$$P(X|\theta) = \sum_{\phi} P(\phi, X|\theta)$$

Et le chemin maximisant la vraisemblance est

$$\hat{\phi} = \arg \max_{\phi} P(\phi, X|\theta)$$

La représentation graphique du modèle phylo-HMM ainsi que la transition des états sont représentés sur Figure 2.

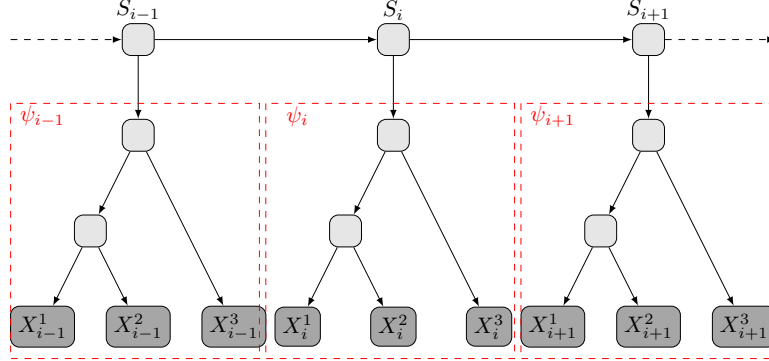


FIGURE 2 – Un modèle Phylo-HMM où figurent 3 états. L'état est associé à une phylogénie de 3 espèces. Les  $X_i^j$  représentent le nucléotide de la  $i$ ème colonne de l'espèce  $j$  dans l'alignement.

## 4 Application 1 : A Toy Gene Finder

On utilise les phylo-HMM pour la découverte de gène. On considère un modèle composé de quatre états. Un état modélise 3 positions codantes et une position non-codante. On veut identifier les gènes dans un ensemble de données synthétiques en utilisant les alignements de séquences et le modèle.

### 4.1 Génération des données

Les paramètres du modèle utilisé dans le papier ne sont pas tous fournis et bien qu'en dise l'auteur, ils sont indisponibles sur son site <http://www.cse.ucsc.edu/~acs/context/> (Siepel and Haussler, 2004b). Bien qu'il soit possible d'apprendre ces paramètres, le pré-traitement des données est en soit une tâche longue et qui n'entre pas réellement dans le cadre des modèles graphiques :

- Récupération des séquences Fasta des chromosome 7 de l'homme et recherche des homologues chez les 7 autres espèces (le gène est alors composé de 1.8 millions de nucléotides) pour la région non-codante et des gènes codants de l' $ARN_m$  présents à la fois chez l'homme et chez les autres espèces
- Alignement multiple des séquences
- Post processing des séquences pour gérer les trous d'alignement, les cadres de lecture menant aux codons stops...

Les paramètres de l'arbre phylogénétique pour l'état non-codant étant donnés, nous avons extrapolé ces paramètres (division par 10 + bruit) pour les états codants. Une région codante est moins sujette aux mutations car les mutations peuvent entraîner une modification de la fonction protéique. Cependant, les mutations restent possibles car d'une part, elles peuvent être silencieuses vis-à-vis de la séquence polypeptidique ensuite générée (dû à la redondance du code génétique), d'autre part, de faibles modifications de la chaîne de peptides n'entraînent pas pour autant une modification de la structure 3D qui définit en grande partie la fonction de la protéine (voir Figure 3).

Nous avons ensuite généré un alignement de 100000 nucléotides selon ce modèle en conservant la séquence des états cachés pour la recherche des gènes. Un extrait d'une séquence générée est présentée ci-après : une partie codante correspond aux états 1,2,3 et la partie non-codante est générée à l'état 4.



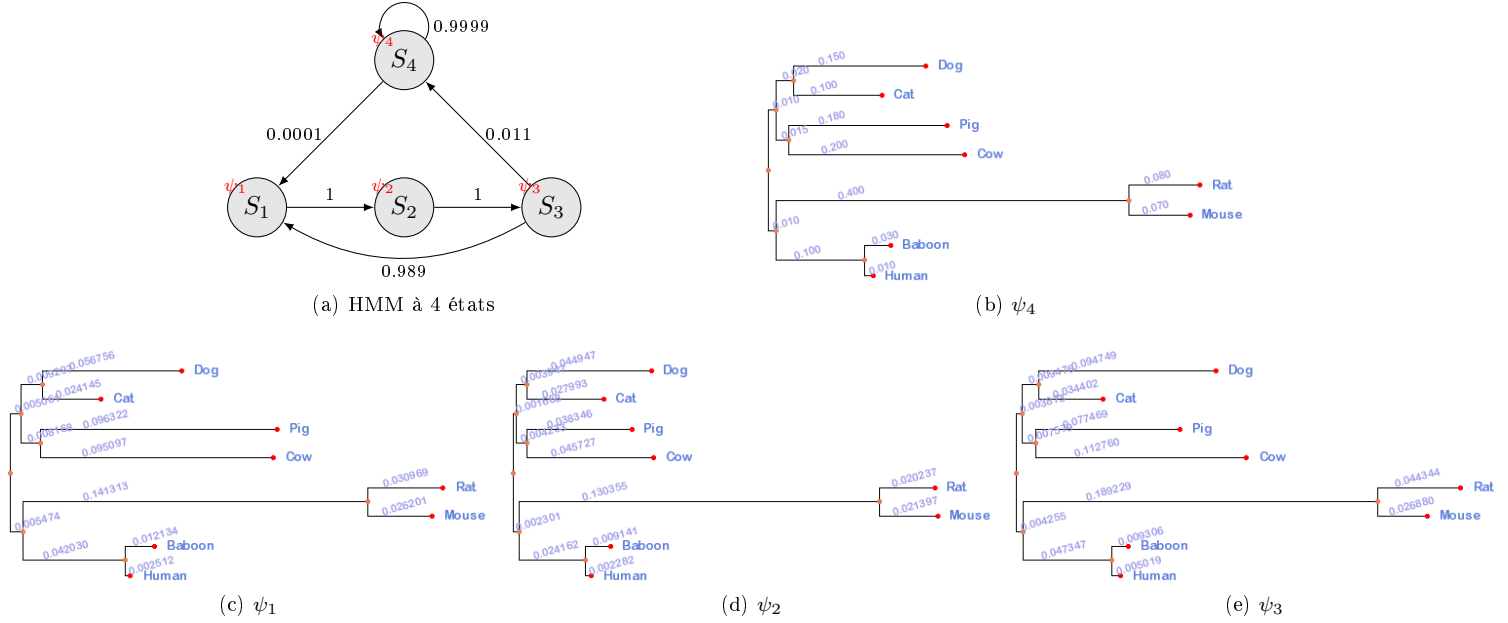


FIGURE 3 – Modèle utilisé pour la génération des données. (a) Les transitions entre états. (b) Arbre phylogénétique pour l'état non-codant (paramètres du modèle proposé par Siepel and Haussler (2005)). (c-d-e) Arbres phylogénétiques pour les états codants extrapolés à partir de (b) avec bruit.

|           | Génération codante      | Génération non-codante |    |
|-----------|-------------------------|------------------------|----|
| Human     | CACCCACGGCTGGTAGGACA    | ATATATGTATCGTTGCTTTA   | 41 |
| Baboon    | CACCCACGGCTGGTTGGACAA   | CACTTGTACCAATAATTATG   | 41 |
| Mouse     | CACCCACGGATGGTAGGACATA  | ATTTTAATCTGTTTTTAA     | 41 |
| Rat       | CACCCACGGATGATAGGACATT  | CCTAAATGATCTGTTTCT     | 41 |
| Cow       | CCCGACATGGATGGTAGGGCATT | TCACCTTTTAATCTACAGT    | 41 |
| Pig       | CCCGACATGGATGGTAGGGCAAT | AACTAACAAAAATCCTCG     | 41 |
| Cat       | CACGACAAGGCTGGTAGGACATT | ATCCGGGTTATCTAATTA     | 41 |
| Dog       | CACGACAAGGCTGGTAGGACATT | ATCTTTGATATCCAACCTC    | 41 |
| consensus | CaCC.CA.GGaTGGTAGGaCA   | ttta...t...at...t...   |    |

☐ non conserved  
 X  $\geq 50\%$  conserved  
 X  $\geq 80\%$  conserved

## 4.2 Identification des gènes

A partir de l'alignement, l'objectif est de retrouver la séquences d'états cachés  $\phi$  ayant mené à ces séquences pour identifier les exons et les introns (l'exons étant une partie codante responsable de l'expression du gène). Pour retrouver, cette séquence cachée on utilise l'algorithme Viterbi.

### 4.2.1 L'algorithme de Viterbi

L'algorithme calcule la chemin le plus probable  $\hat{\phi} = \arg \max_{\phi} P(\phi, X|\theta)$  de manière dynamique similaire à l'algorithme *forward*. On pose :

$$V_{i,k} = \max_{\phi_1 \dots \phi_i} P(\phi_i = k | \phi_0 \dots \phi_{i-1}, X_0 \dots X_i, \Theta)$$

la probabilité de la séquence  $\phi_1 \dots \phi_i$  se terminant par  $k$  la plus vraisemblable ayant généré  $X_0 \dots X_i$ . On a donc  $\max_k V_{L,k}$  la probabilité du chemin le plus vraisemblable ayant mené à toute les

observations. De façon récursive,  $V_{i_k}$  s'exprime ainsi :

$$\begin{aligned} V_{0,k} &= b_k P(X_0|\psi_k) \\ V_{i,k} &= \max_j \{V_{i-1,j} \cdot a_{j,k}\} \cdot P(X_i|\psi_k) \end{aligned}$$

En stockant les chemins lors de la passe *forward* lorsque l'on choisit le maximum, il suffit ensuite de backtracker en partant de  $\max_k V_{L,k}$  pour récupérer le chemin de  $\phi$ . Reste maintenant à calculer  $P(X_i|\psi_k)$  à partir de l'arbre phylogénétique en utilisant l'algorithme de Felsenstein (1981).

### 4.3 L'algorithme de Felsenstein

L'objectif est de calculer  $P(X_i|\psi_k)$ , c'est-à-dire la vraisemblance que le modèle phylogénétique  $k$  aie produit la colonne  $X_i$ . Cependant les données sont incomplètes, on connaît les séquences homologues aux feuilles de l'arbre mais pas les gènes ancestraux. La vraisemblance consiste donc en la somme de celles de tous les chemins possibles menant à la séquence  $X_i$  aux feuilles (ce qui correspond à marginaliser selon les noeuds ancestraux). Soit  $u$  un noeud de l'arbre,  $X_u$  la v.a. de la valeur de ce noeud et  $X_i^u$  les nucléotides observés en dessous dans la clade de  $u$  (les feuilles du sous-arbre enraciné en  $u$ ). L'algorithme de Felsenstein (1981) applique le principe d'élimination pour calculer la vraisemblance du sous-arbre enraciné en  $u$ ,  $P(X_i^u|X_u, \psi)$  et se comporte ainsi comme l'algorithme *backward* des HMM pour calculer de manière récursive la vraisemblance du sous-arbre enraciné au noeud parent de  $u$  :

$$P(X_i^u|X_u = a) = \begin{cases} \mathbb{1}_{X_i=a} & \text{si } u \text{ est une feuille} \\ \prod_{v \text{ fils de } u} \left\{ \sum_{a \in \Sigma} P(b|a, t_v, \psi) \cdot P(X_i^v|x_v = b) \right\} & \text{sinon} \end{cases}$$

où  $t_u$  est la longueur de la branche reliant  $u$  à son père. On obtient ensuite la vraisemblance de  $X_i$  selon  $\psi$  en marginalisant par rapport à la racine de l'arbre :

$$P(X_i|\psi) = \sum_{a \in \Sigma} \pi_a \cdot P(X_i^{root}|X_{root} = a) \quad (1)$$

où  $\pi$  est le vecteur de configuration initiale associé à  $\psi$ .

### 4.4 Evaluation

Pour évaluer la pertinence de la méthode, nous avons généré 80 séquences de  $L = 100000$  nucléotides selon le modèle en conservant le chemin  $\phi$  des états comme référence. Avec ce modèle, chaque séquence contient en moyenne 5 exons. Le chemin de référence  $\phi$  est ensuite comparé au chemin d'états  $\hat{\phi}$  estimé avec viterbi. Deux critères de performance sont utilisés :

- Le taux de bonne classification par état :

$$\tau = \frac{1}{L} \sum_{i=0}^L \mathbb{1}_{\phi_i = \hat{\phi}_i} \quad (2)$$

- Le taux de bonne classification par état codant (qui est celui qui nous interesse réellement dans le cas de la prédiction) :

$$\tau = \frac{1}{L} \sum_{i=0}^L \mathbb{1}_{\phi_i=4 \text{ et } \hat{\phi}_i=4} + \mathbb{1}_{\phi_i \neq 4 \text{ et } \hat{\phi}_i \neq 4} \quad (3)$$

Le modèle obtient 92.1% de bonne prédiction selon le premier critère et 99.9% selon le second. Cette différence s'explique facilement. La prédiction commet essentiellement des erreurs aux niveaux des transitions entre introns-exons. Une mauvaise transition intron-exon entraîne un décalage de la séquence d'états codants de l'exon qui fait chuter le taux de bonne classification. Si maintenant, on ne considère plus que la prédiction en terme d'états codants/non codants, le décalage n'est plus pris en compte dans le calcul de l'erreur.

Nous avons également étudié l'influence du nombre d'espèces dans la phylogénie et comparé le modèle Phylo-HMM, avec un modèle non phylogénétique qui suppose l'indépendance entre nucléotides d'une même colonne  $X_i$ . Le modèle non-phylogénétique se rapporte ainsi à un HMM dont la probabilité d'émission suit une loi multinomiale de paramètre  $\pi$ , le vecteur de fréquences initiales de  $\psi$  :

$$P(X_i|\psi) = \prod_{j=1}^n \pi_{X_i^j}$$

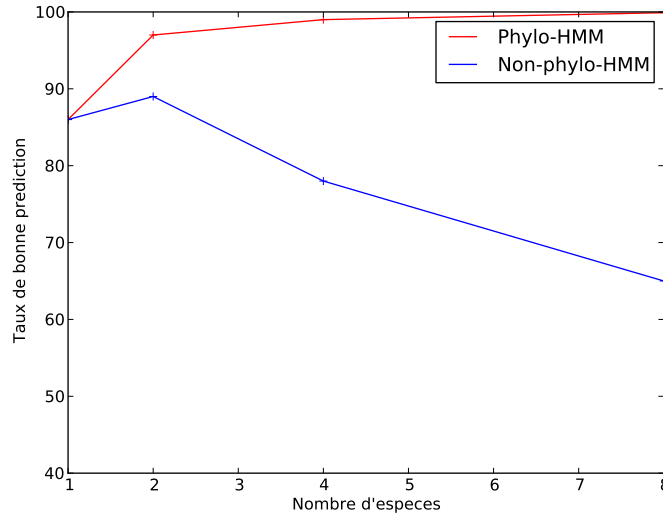


FIGURE 4 – Comparaison des performances de prédiction des modèles Phylo-HMM et non phylogénétiques en fonction du nombre d'espèces dans la phylogénie.

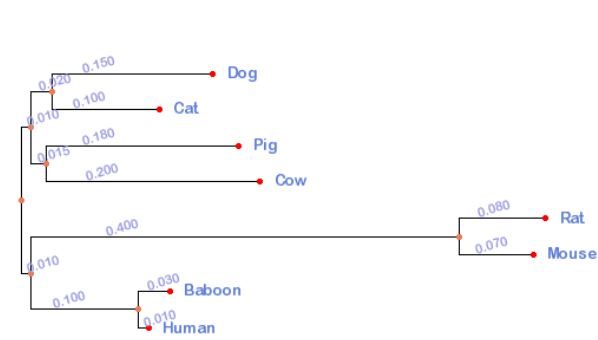
La Figure 4 montre l'importance du modèle phylogénétique pour prendre en compte les corrélations entre les nucléotides d'une colonne. La supposition d'indépendance faite par le modèle non-phylogénétique entraîne des erreurs critiques de prédictions d'états lorsque des mutations aléatoires apparaissent dans l'alignement.

## 5 Application 2 : Identification of Highly Conserved Regions

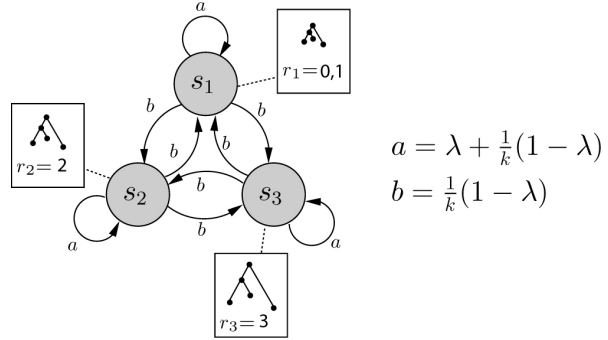
On considère ici un modèle où les différents états ( $k = 3$ ) correspondent à différents taux. Chaque état est défini d'après un arbre de base dont la longueur des branches est multipliée par un taux  $r$ . Le but est de détecter les régions conservées, qui correspondent à une probabilité a posteriori d'être sur l'état de plus petit taux élevée. Autrement dit, sur ces régions, les nucléotides de l'ancêtre ont peu de chance d'être modifiées au cours du temps.

### 5.1 Définition du Modèle

On considère l'arbre représenté sur la 5a et une transition d'état définie sur la 5b.



(a) Arbre phylogénétique de base dont la longueur des branches est multipliée par le taux du modèle. On considère donc 8 espèces pour chaque site de nucléotide.



(b) Transition des états selon un paramètre  $\lambda$  fixé à 0,94.

La transition des états est modélisée selon un unique paramètre  $\lambda$ . Les taux  $r_1, r_2, r_3$  sont fixés afin de bien séparer les trois états.

## 5.2 Génération des Données

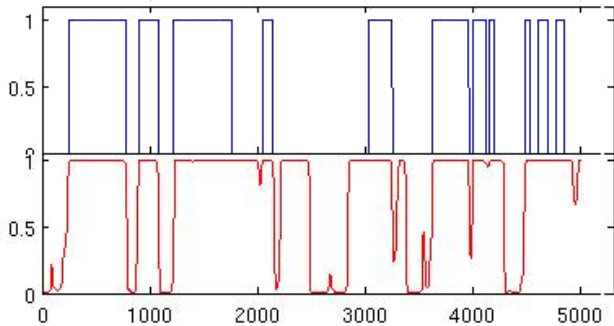
On procède de la même manière que pour l'exemple précédent, en modifiant la matrice de transition d'état, la probabilité initiale ainsi que les paramètres des modèles phylogénétiques.

## 5.3 Calcul des Probabilités Posteriores

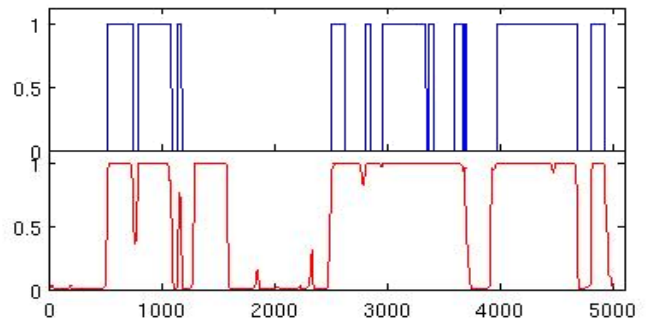
Pour calculer les probabilités a posteriori  $P(\Phi_i|X)$ , on procède comme suit :

- On suppose connu les paramètres de la HMM :  $A, \lambda, \pi$ .
- Avec l'algorithme de Felsenstein, on calcule les probabilités d'émission  $P(X_i|\Psi_j)$ .
- Avec l'algorithme forward, backward, on calcule les probabilités a posteriori.

On s'intéresse en particulier à  $P(\Phi_i = 1|X)$  qui correspond à la probabilité d'être sur l'état 1 sachant les données, le premier état définissant les zones fonctionnelles. En appliquant cet algorithme à une séquence de 100000 sites pour 8 espèces différentes. Les probabilités a posteriori pour des taux  $r_1 = 0,01, r_2 = 1, r_3 = 1,1$  sont représentées sur 5c, celles pour les taux  $r_1 = 0,01, r_2 = 0,1, r_3 = 0,2$  sont représentées 5d.



(c) Exemple 1 : Représentation des états cachés  $\Phi_i = 1$  (en bleu) et des probabilités a posteriori  $P(\Phi_i = 1|X_j)$ .



(d) Exemple 2 : Représentation des états cachés  $\Phi_i = 1$  (en bleu) et des probabilités a posteriori  $P(\Phi_i = 1|X_j)$ .

FIGURE 5 – Exemples de prédiction de regions conservées.

On remarque que les probabilités a posteriori permettent de détecter les sites définis par l'état le plus conservateur. Cette méthode permet donc de détecter les zones fonctionnelles des chaînes

nucléotidiques.

## 6 Conclusion

L'utilisation des modèles graphiques dans le cadre de la phylogénie permet d'utiliser des algorithmes (Felsenstein, Viterbi, forward backward, ...) qui permettent de calculer les probabilités a posteriori et les chemins cachés exactement et efficacement. Nous avons vu des exemples d'applications des modèles graphiques à la génétique. Même si les données correspondent ici exactement aux modèles, nous avons vu que ces méthodes sont pertinentes pour résoudre des problèmes courants de la génétique tels que la prédiction des gènes et la recherche des domaines conservés dans une phylogénie. Nous avons également mis en évidence la pertinence du modèle phylogénétique couplé aux HMM pour ces tâches. Avec un modèle adéquat aux données réelles biologiques, ces méthodes figurent parmi celles à l'état de l'art.

## Bibliographie

### Références

- J. Felsenstein. Evolutionary trees from dna sequences : a maximum likelihood approach. *Journal of molecular evolution*, 17(6) :368–376, 1981.
- M. Hasegawa, H. Kishino, and T. Yano. Dating of the human-ape splitting by a molecular clock of mitochondrial dna. *Journal of molecular evolution*, 22(2) :160–174, 1985.
- A. Siepel and D. Haussler. Combining phylogenetic and hidden markov models in biosequence analysis. *Journal of Computational Biology*, 11(2-3) :413–428, 2004a.
- A. Siepel and D. Haussler. Phylogenetic estimation of context-dependent substitution rates by maximum likelihood. *Molecular Biology and Evolution*, 21(3) :468–488, 2004b.
- A. Siepel and D. Haussler. Phylogenetic hidden markov models. *Statistical methods in molecular evolution*, pages 325–351, 2005.

---

## 5. PRÉSENTATION DE MON DOMAINE DE RECHERCHE ET STAGE DE FIN D'ÉTUDE

Le rapport suivant est la nouvelle rédaction de ce mémoire : elle permet d'introduire chronologiquement et rigoureusement les modèles mathématiques sur lesquels je travaille aujourd'hui en thèse, et de critiquer ces modèles une fois le cadre théorique présenté. Il est suivi de mon rapport de stage de fin d'étude, et constitue une bonne introduction pour ce dernier.

## Magistère de Mathématiques d'Orsay (Paris-Sud XI)

Parcours Mathématiques Fondamentales et Appliquées, Options Probabilités et Statistiques, Informatique Théorique et Mathématiques Appliquées

### PRÉSENTATION DU DOMAINE DE RECHERCHE

NICOLAS DROUGARD

#### 1. INTRODUCTION

Mon domaine de recherche mêle probabilités, statistiques, informatique et traitement d'image, ainsi qu'applications robotiques. Une fois les outils mathématiques clarifiés (nécessitant seulement des notions assez basiques de théorie des probabilités) ce rapport présentera les modèles autour desquels mon travail de recherche s'organise : ces derniers permettent l'optimisation de problèmes de décision séquentielle dans l'incertain probabiliste. Nous expliquerons alors leurs limites, et quelques théories alternatives potentiellement utiles à l'amélioration de la modélisation seront présentées. Certaines d'entre elles seront très certainement au coeur de ma thèse.

#### 2. DE MARKOV AUX POMDPs

Cette partie a pour but d'introduire mon domaine de recherche en construisant le modèle auquel je m'intéresse. Nous remontons pour cela un siècle dans le passé.

**2.1. Les chaînes de Markov.** C'est dans les premières années du vingtième siècle qu'Andreï Markov (dont le directeur de thèse fut Pafnouti Tchebychev) met en place les chaînes de Markov : pour l'anecdote, c'est en étudiant la succession des lettres d'un roman (dont chaque lettre dépendait essentiellement de la précédente) qu'il aurait eu l'idée de définir ce type de suite de variables aléatoires.

**Définition 1.** Soient  $\mathcal{S}$  un ensemble dénombrable, et  $(S_n)_{n \in \mathbb{N}}$  une suite de variables aléatoires à valeur dans  $\mathcal{S}$ . La suite  $(S_n)_{n \in \mathbb{N}}$  est une **chaîne de Markov** si

$$\begin{aligned} \forall n \geq 0, \forall (s_0, s_1, \dots, s_{n+1}) \in \mathcal{S}^{n+2}, \\ \mathbb{P}(S_{n+1} = s_{n+1} \mid S_0 = s_0, S_1 = s_1, \dots, S_n = s_n) \\ = \mathbb{P}(S_{n+1} = s_{n+1} \mid S_n = s_n) \end{aligned}$$

i.e.  $\forall n \geq 0$ , la loi de  $S_{n+1}$  ne dépend que de la variable aléatoire précédente  $S_n$ .

L'ensemble  $\mathcal{S}$  est appelé l'*ensemble des états*. Numérotions les éléments (dits *états*) de  $\mathcal{S} = \{s^{(1)}, s^{(2)}, \dots, s^{(\text{Card } \mathcal{S})}\}$ . La dynamique probabiliste de toute chaîne de Markov peut donc être représentée par une suite  $(M_n)_{n \in \mathbb{N}}$  de matrices stochastiques<sup>1</sup> de taille  $\text{Card } \mathcal{S} \times \text{Card } \mathcal{S}$  définies par

$$(M_n)_{i,j} = \mathbb{P} \left( S_{n+1} = s^{(j)} \mid S_n = s^{(i)} \right).$$

Le cours de Raphael Cerf (Arnaud Durand et Claire Lacour) en première année de Master (deuxième année de Magistère) nous a présenté de manière approfondie les chaînes de Markov homogènes, *i.e.* les chaînes de Markov telles que la suite  $(M_n)_{n \in \mathbb{N}}$  est constante.

Montrons quelques résultats utiles concernant les chaînes de Markov non homogènes. Rappelons tout d'abord la définition la plus générale de l'espérance conditionnelle par rapport à une variable aléatoire :

**Définition 2.** Soient  $X$  et  $Y$  deux variables aléatoires intégrables définies sur  $(\Omega, \mathcal{F}, \mathbb{P})$ ,  $X$  à valeur dans  $\mathbb{R}$  et  $Y$  à valeur dans un ensemble  $\mathcal{Y}$  muni d'une tribu. Notons  $\sigma(Y)$  la tribu engendrée par  $Y$ . L'espérance conditionnelle de  $X$  sachant  $Y$ , notée  $\mathbb{E}[X \mid Y]$ , est l'unique variable aléatoire de  $L^1(\Omega, \mathcal{F}, \mathbb{P})$  qui est  $\sigma(Y)$ -mesurable et qui vérifie

$$\forall B \in \sigma(Y),$$

$$\int_B \mathbb{E}[X \mid Y] d\mathbb{P} = \int_B X d\mathbb{P}.$$

Un petit travail de théorie de la mesure nous permet de remplacer la condition de  $\sigma(Y)$ -mesurabilité par

$$\exists \varphi : \mathcal{Y} \rightarrow \mathbb{R} \text{ mesurable telle que } \mathbb{E}[X \mid Y] = \varphi(Y).$$

En effet,  $\varphi(Y)$  est  $\sigma(Y)$ -mesurable puisque  $Y$  l'est, et que  $\varphi$  est mesurable. Il faut alors montrer que l'ensemble des fonctions s'écrivant  $\varphi(Y)$  avec  $\varphi$  mesurable (notons le  $\Phi$ ) contient l'ensemble des fonctions  $\sigma(Y)$ -mesurables : comme  $\forall A \in \sigma(Y)$ ,  $\exists B$  borélien tel que  $A = \{Y \in B\}$ , nous avons  $\mathbb{1}_A = \mathbb{1}_{Y \in B} = \mathbb{1}_B(Y) \in \Phi$ , et la stabilité par combinaison linéaire et limite croissante de  $\Phi$  permet de conclure qu'il contient toutes les fonctions  $\sigma(Y)$ -mesurables.

Cette formulation a l'avantage d'être plus intuitive que celle de la définition, et notamment en pratique. Nous retrouvons par ailleurs la définition classique de l'espérance conditionnelle pour les variables aléatoires à valeur dans les espaces dénombrables :  $X$  et  $Y$  sont donc considérées maintenant à valeur dans  $\mathcal{S}$ , et nous introduisons  $f : \mathcal{S} \rightarrow \mathbb{R}$  mesurable. Comme  $\mathbb{E}[f(X) \mid Y]$  est  $\sigma(Y)$ -mesurable, elle peut s'écrire  $\varphi(Y)$  avec  $\varphi$  mesurable, et la seconde condition de la définition entraîne, en considérant uniquement les singletons

---

1. Une matrice stochastique est une matrice à coefficients positifs ou nuls dont les lignes somment à 1



de  $\mathcal{S}$ ,

$$\begin{aligned}
& \forall y \in \mathcal{S}, \\
& \int_{Y \in \{y\}} \varphi(Y) d\mathbb{P} = \int_{Y \in \{y\}} f(X) d\mathbb{P} \\
\Leftrightarrow & \varphi(y) \mathbb{P}(Y = y) = \int_{Y=y} f(X) d\mathbb{P} \\
\Leftrightarrow & \varphi(y) = \int_{\Omega} f(X) \frac{\mathbf{1}_{Y=y} d\mathbb{P}}{\mathbb{P}(Y=y)} \\
& = \int_{\Omega} \sum_{x \in \mathcal{S}} f(x) \mathbf{1}_{X=x} \frac{\mathbf{1}_{Y=y} d\mathbb{P}}{\mathbb{P}(Y=y)} \\
& = \sum_{x \in \mathcal{S}} f(x) \frac{\mathbb{P}(X=x, Y=y)}{\mathbb{P}(Y=y)} \\
& = \sum_{x \in \mathcal{S}} f(x) \mathbb{P}(X=x \mid Y=y)
\end{aligned}$$

par le théorème de Fubini.

Le second point de la définition 2 peut aussi être remplacé par :  $\forall Z$   $\sigma(Y)$ -mesurable,

$$\mathbb{E}[Z \mathbb{E}[X \mid Y]] = \mathbb{E}[ZX].$$

En effet, si  $Z = \mathbf{1}_B$  avec  $B \in \sigma(Y)$ , nous retrouvons le second point de la définition. Par linéarité de l'espérance, cela reste vrai pour  $Z$  étagée, et le théorème de convergence monotone nous fournit le résultat  $\forall Z$  mesurable. Cette formulation nous permet de montrer facilement que

**Résultat 1.**  $\forall X, Y_1, Y_2$  variable aléatoires réelles intégrables,

$$\mathbb{E}[\mathbb{E}(X \mid Y_1, Y_2) \mid Y_2] = \mathbb{E}[X \mid Y_2].$$

*Démonstration.* Les deux quantités étant  $\sigma(Y_2)$ -mesurables, il suffit de montrer que  $\forall Z$   $\sigma(Y_2)$ -mesurable,  $\mathbb{E}\{Z \mathbb{E}[\mathbb{E}(X \mid Y_1, Y_2) \mid Y_2]\} = \mathbb{E}[ZX]$  : soit  $Z$   $\sigma(Y_2)$ -mesurable,

$$\begin{aligned}
\mathbb{E}\{Z \mathbb{E}[\mathbb{E}(X \mid Y_1, Y_2) \mid Y_2]\} &= \mathbb{E}[Z \mathbb{E}(X \mid Y_1, Y_2)] \\
&= \mathbb{E}[ZX]
\end{aligned}$$

car si  $Z$  est  $\sigma(Y_2)$ -mesurable, elle est  $\sigma(Y_1, Y_2)$ -mesurable.  $\square$

Nous montrons maintenant un résultat classique sur les chaînes de Markov :

**Résultat 2.** Soit  $(S_n)_{n \in \mathbb{N}}$  une chaîne de Markov à valeurs dans  $\mathcal{S}$ .

$\forall n \in \mathbb{N}, \forall f : \mathcal{S} \rightarrow \mathbb{R}$  bornée,

$$\mathbb{E}[f(S_{n+1}) \mid S_0, \dots, S_n] = \sum_{s \in \mathcal{S}} f(s) \mathbb{P}(S_{n+1} = s \mid S_n) \quad \mathbb{P}\text{-presque surement}.$$

*Démonstration.* Soient  $n \in \mathbb{N}, (s_0, s_1, \dots, s_n) \in \mathcal{S}^n$ , et  $s' \in \mathcal{S}$  : d'une part,

$$\int_{\{S_0=s_0, \dots, S_n=s_n\}} \mathbb{1}_{S_{n+1}=s'} d\mathbb{P} = \mathbb{P}(S_0 = s_0, \dots, S_n = s_n, S_{n+1} = s'),$$

d'autre part

$$\begin{aligned} \int_{\{S_0=s_0, \dots, S_n=s_n\}} \mathbb{P}(S_{n+1} = s' \mid S_n) d\mathbb{P} \\ = \mathbb{P}(S_{n+1} = s' \mid S_n = s_n) \mathbb{P}(S_0 = s_0, S_1 = s_1, \dots, S_n = s_n), \end{aligned}$$

et les deux intégrales sont donc égales puisque  $(S_n)$  est une chaîne de Markov. Comme les événements de  $\sigma(S_0, \dots, S_n)$  sont des unions d'événements de la forme  $\{S_0 = s_0, S_1 = s_1, \dots, S_n = s_n\}$ , pour tout  $B \in \sigma(S_0, \dots, S_n)$

$$\int_B \mathbb{1}_{S_{n+1}=s'} d\mathbb{P} = \int_B \mathbb{P}(S_{n+1} = s' \mid S_n) d\mathbb{P}$$

et ainsi, par la définition 2,  $\mathbb{P}(S_{n+1} = s' \mid S_n)$  étant  $\sigma(S_0, \dots, S_n)$ -mesurable,

$$\mathbb{P}(S_{n+1} = s' \mid S_n) = \mathbb{E}[\mathbb{1}_{S_{n+1}=s'} \mid S_0, \dots, S_n]$$

$\mathbb{P}$ -presque surement.

Maintenant,

$$\begin{aligned} \mathbb{E}[f(S_{n+1}) \mid S_0, \dots, S_n] &= \mathbb{E}\left[\sum_{s' \in \mathcal{S}} f(s') \mathbb{1}_{S_{n+1}=s'} \mid S_0, \dots, S_n\right] \\ &= \sum_{s' \in \mathcal{S}} f(s') \mathbb{E}[\mathbb{1}_{S_{n+1}=s'} \mid S_0, \dots, S_n] \\ &= \sum_{s' \in \mathcal{S}} f(s') \mathbb{P}(S_{n+1} = s' \mid S_0, \dots, S_n) \\ &= \sum_{s' \in \mathcal{S}} f(s') \mathbb{P}(S_{n+1} = s' \mid S_n) \\ &= \mathbb{E}[f(S_{n+1}) \mid S_n] \end{aligned}$$

$\mathbb{P}$ -presque surement. □

Ces deux résultats vont nous aider à mettre en place rigoureusement dans la partie qui suit les modèles autour desquels se situe mon domaine de recherche.

**2.2. Les Processus Décisionnels de Markov (MDP).** ([1]) Une institution de Recherche et Développement créée en 1945 par l'armée de l'air américaine, engage en 1949 un jeune professeur de mathématiques spécialiste de la théorie des nombres : Richard Bellman. Cette période est marquée par la modernisation des armées de nombreux pays, et cette évolution nécessite en effet le savoir faire scientifique. On demande alors à R.Bellman de travailler sur les processus de décision à étapes multiples.

Les processus décisionnels de Markov font partie de ces outils : ces modèles probabilistes sont conçus pour représenter une situation où une entité (appelée *agent*) évolue avec son environnement de manière markovienne au cours du temps (le temps modélisé par l'ensemble des entiers naturels  $\mathbb{N}$ ). L'ensemble composé de l'agent et de son environnement est appelé le *système* : il est dans un état  $s \in \mathcal{S}$  à chaque instant  $t \in \mathbb{N}$ . L'agent modifie la dynamique probabiliste du système en utilisant à chaque étape  $t \in \mathbb{N}$ , une action  $a \in \mathcal{A}$  où  $\mathcal{A}$  est un ensemble fini. À chaque *fonction de décision*  $\left\{ \begin{array}{l} \pi : \mathcal{S} \rightarrow \mathcal{A} \\ s \mapsto \pi(s) \end{array} \right.$  est associée une matrice stochastique  $M^\pi$  décrivant les probabilités de transition de la chaîne de Markov  $(S_n)_{n \in \mathbb{N}}$  qui représente l'état du système :  $(M^\pi)_{i,j}$  est la probabilité de quitter l'état  $s^{(i)}$  pour arriver dans l'état  $s^{(j)}$  lorsque l'action  $\pi(s^{(i)})$  a été utilisée par l'agent. Considérons une suite de fonctions de décision :  $(\pi_n)_{n \in \mathbb{N}}$  : une telle suite est appelée une *politique* et définit alors une suite de matrices stochastiques  $(M_n)_{n \in \mathbb{N}}$  avec  $M_n = M^{\pi_n}$ . Toute politique définit donc entièrement les paramètres d'une chaîne de Markov. À chaque couple  $(s, a) \in \mathcal{S} \times \mathcal{A}$ , est associée une récompense  $r(s, a) \in \mathbb{R}$  afin de modéliser l'importance de passer par l'état  $s$  et d'y effectuer l'action  $a$ . Une récompense terminale associée à chaque état  $s$ ,  $R(s)$  modélise le but de l'agent. En effet, résoudre un problème de commande optimale d'un processus de Markov (d'horizon  $n$ ) consiste alors à trouver une politique qui maximise l'espérance de la somme des récompenses :

$$V_n(s, (\pi_t)_{0 \leq t \leq n-1}) := \mathbb{E} \left[ \sum_{t=0}^{n-1} r(S_t, \pi_t(S_t)) + R(s_n) \mid S_0 = s \right]$$

Cette fonction de la taille de l'horizon, de l'état initial et de la politique est appelée *fonction valeur*. L'ensemble des politiques d'horizon  $n$ , *i.e.* séquences de  $n$  fonctions de décision  $\pi$  numérotées de 0 à  $n-1$ , se note  $\Pi_n$ .

**2.3. La programmation dynamique.** Richard Bellman a trouvé une méthode très efficace pour résoudre ce problème de contrôle optimal des chaînes de Markov : un algorithme utilisant le *principe de la programmation dynamique*. Laurent Rosaz nous a présenté ce principe dans son cours d'Algorithmique de première année de Magistère : il permet dans certaines situations de donner une solution à un problème global en le décomposant en sous-problèmes plus simples à résoudre, et ainsi de réduire la complexité de résolution (l'exemple typique est celui du problème de plus court chemin, ou de chemin de moindre coût dans un graphe).

Nous mettons maintenant en place l'équation permettant le calcul rapide de  $V_n^*(s) = \max_{\Pi_n} \mathbb{E} \left[ \sum_{t=0}^{n-1} r(S_t, \pi_t(S_t)) + R(s_n) \mid S_0 = s \right]$  et d'une politique  $(\pi_t^*)_{0 \leq t \leq n-1}$  telle que  $\forall s \in \mathcal{S}, V_n^*(s) = V_n(s, (\pi_t^*)_{0 \leq t \leq n-1})$ . Notons  $\mathbf{p}(s' \mid s, a)$

la probabilité que le système passe de l'état  $s$  à l'état  $s'$  lorsque l'action  $a$  a été utilisée par l'agent.

**Résultat 3.** *Equation de Bellman*

Si  $n \in \mathbb{N}$  est l'horizon du problème,

$$\forall 0 \leq i < n, \forall s \in \mathcal{S},$$

$$V_{i+1}^*(s) = \max_{a \in \mathcal{A}} \left\{ r(s, a) + \sum_{s' \in \mathcal{S}} \mathbf{p}(s' | s, a) V_i(s) \right\},$$

$$\text{et } V_0(s) = R(s).$$

De même,

$$\forall 0 \leq t < n, \forall s \in \mathcal{S},$$

$$\pi_t^*(s) \in \operatorname{argmax}_{a \in \mathcal{A}} \left\{ r(s, a) + \sum_{s' \in \mathcal{S}} \mathbf{p}(s' | s, a) V_{n-t-1}(s) \right\}.$$

*Démonstration.* La taille courante de l'horizon,  $i$ , est l'indice utilisé pour la fonction valeur  $V^*$ . C'est aussi l'opposé modulo  $n$  de  $t$ , le temps, indice utilisé pour la politique. L'initialisation  $V_0^*(s) = R(s)$  est évidente : lorsque l'horizon est de taille nulle, aucune action n'est à exécuter par l'agent, et il reçoit seulement la récompense terminale. Ensuite, soit  $0 \leq i < n$ , et posons  $t_0 = n - i - 1$  :

$$V_{i+1}^*(s) = \max_{(\pi_t)_{t_0 \leq t < n} \in \Pi_n} \mathbb{E} \left[ \sum_{t=t_0}^{n-1} r(S_t, \pi_t(S_t)) + R(s_n) \mid S_{t_0} = s \right].$$

$$\begin{aligned} \text{Or } & \mathbb{E} \left[ \sum_{t=t_0}^{n-1} r(S_t, \pi_t(S_t)) + R(s_n) \mid S_{t_0} \right] \\ &= \mathbb{E} \left[ r(S_{t_0}, \pi_{t_0}(S_{t_0})) + \mathbb{E} \left[ \sum_{t=t_0+1}^{n-1} r(S_t, \pi_t(S_t)) + R(s_n) \mid S_{t_0}, S_{t_0+1} \right] \mid S_{t_0} \right] \\ &= \mathbb{E} \left[ r(S_{t_0}, \pi_{t_0}(S_{t_0})) + \sum_{s' \in \mathcal{S}} \mathbf{p}(s' | S_{t_0}, \pi_{t_0}(S_{t_0})) V_i(s', (\pi_t)_{t_0+1 \leq t < n}) \mid S_{t_0} \right] \end{aligned}$$

grâce aux résultats 1 et 2. En intégrant l'égalité sur  $\{S_{t_0} = s\}$ , nous obtenons

$$\begin{aligned}
V_{i+1}^*(s) &= \max_{(\pi_t)_{t_0 \leq t < n} \in \Pi_n} \left\{ r(s, \pi_{t_0}(s)) + \sum_{s' \in \mathcal{S}} \mathbf{p}(s' \mid s, \pi_{t_0}(s)) V_i(s', (\pi_t)_{t_0+1 \leq t < n}) \right\} \\
&= \max_{(a, (\pi'_t)_{t_0+1 \leq t < n})} \left\{ r(s, a) + \sum_{s' \in \mathcal{S}} \mathbf{p}(s' \mid s, a) V_i(s', (\pi_t)_{t_0+1 \leq t < n}) \right\} \\
&= \max_{a \in \mathcal{A}} \left\{ r(s, a) + \sum_{s' \in \mathcal{S}} \mathbf{p}(s' \mid s, a) \max_{(\pi'_t)_{t_0+1 \leq t < n}} V_i(s', (\pi_t)_{t_0+1 \leq t < n}) \right\} \quad (2.1) \\
&= \max_{a \in \mathcal{A}} \left\{ r(s, a) + \sum_{s' \in \mathcal{S}} \mathbf{p}(s' \mid s, a) V_i^*(s') \right\}
\end{aligned}$$

où 2.1 se justifie par :

- l'inégalité triviale  $\max_{(\pi'_t)_{t_0+1 \leq t < n}} \sum_{s' \in \mathcal{S}} \mathbf{p}(s' \mid s, a) V_i(s', (\pi_t)_{t_0+1 \leq t < n}) \leq \sum_{s' \in \mathcal{S}} \mathbf{p}(s' \mid s, a) \max_{(\pi'_t)_{t_0+1 \leq t < n}} V_i(s', (\pi_t)_{t_0+1 \leq t < n})$
- soit une politique  $(\bar{\pi})_{t_0+1 \leq t < n}$  telle que  $\forall s \in \mathcal{S}$ ,  $\bar{\pi}_{t_0+1}(s) \in \operatorname{argmax}_{a \in \mathcal{A}} \max_{(\pi_t)_{t_0+2 \leq t < n} \in \Pi_{i-1}} V(s, (a, (\pi_t)_{t_0+2 \leq t < n}))$  et optimale ensuite (lorsque  $t > t_0 + 1$ ). Alors,

$$\begin{aligned}
&\sum_{s' \in \mathcal{S}} \mathbf{p}(s' \mid s, a) \max_{(\pi'_t)_{t_0+1 \leq t < n}} V_i(s', (\pi_t)_{t_0+1 \leq t < n}) \\
&= \sum_{s' \in \mathcal{S}} \mathbf{p}(s' \mid s, a) V_i(s', (\bar{\pi}_t)_{t_0+1 \leq t < n}) \\
&\leq \max_{(\pi_t) \in \Pi_{i-1}} \sum_{s' \in \mathcal{S}} \mathbf{p}(s' \mid s, a) V_i(s', (\pi_t)_{t_0+1 \leq t < n}). \quad \square
\end{aligned}$$

L'équation de Bellman permet alors la résolution rétrograde du problème de commande optimale de chaîne de Markov. Mon rapport de stage de fin de master démontre seulement le cas de l'horizon infini, qui est moins intuitif, et dont l'équation de Bellman mène à l'algorithme d'itération sur les valeurs. C'est parce que la recherche en mathématiques n'était pas appréciée par le milieu militaire que R. Bellman a nommé son travail programmation dynamique : « programmation » signifiait alors planification et ordonnancement, et le terme « dynamique » évoquait l'idée d'évolution dans le temps. La résolution a ensuite été améliorée par Howard (algorithme d'itération sur les politiques). Le contrôle de processus de diffusion, *i.e.* la version continue en temps et en espace de ce modèle, a été étudiée par Kushner et Dupuis à la fin du vingtième siècle : j'utilise leurs approximations dans le travail de mon stage de Magistère qui avait pour but de déterminer la consommation optimale d'une entreprise. Enfin, ce modèle est très présent en intelligence artificielle comme le montre la suite de ce rapport.

## 3. MOTIVATIONS DE LA THÈSE

Maintenant que le cadre théorique est posé, intéressons-nous à un modèle plus adapté à la robotique, ainsi qu'aux problèmes qu'il soulève.

**3.1. Des MDPs partiellement observable en robotique.** Lorsque ce type de modélisation est utilisé en robotique, l'agent est le robot puisque c'est à lui de prendre les décisions. En pratique, il n'a pas toujours connaissance de l'état dans lequel se trouve le système, mais peut l'estimer à l'aide d'observations  $o \in \mathcal{O}$  qu'il reçoit à chaque instant. Les *processus de décision markoviens partiellement observables* (POMPD) sont alors plus adaptés : l'ensemble des observations possibles  $\mathcal{O}$  est fini, et pour tous les  $o' \in \mathcal{O}$ ,  $s' \in \mathcal{S}$  et  $a \in \mathcal{A}$ , la probabilité  $\mathbf{p}(o' | s', a)$ , que l'agent observe  $o'$  s'il a effectué l'action  $a$  et si le système vient d'arriver dans l'état  $s'$ , est connue. La connaissance de l'agent à l'instant  $t$  est modélisée par une distribution de probabilité sur  $\mathcal{S}$  appelée *fonction de croyance* :  $\forall s \in \mathcal{S}$ ,

$$\mathbf{b}_t(s) := \mathbb{P}(S_t = s | i_t)$$

où  $i_t = (\mathbf{b}_0, a_0, o_1, a_1, \dots, a_{t-1}, o_t)$  est la croyance initiale de l'agent  $\mathbf{b}_0$ , et la suite des actions réalisées et des observations reçues jusqu'au temps  $t$  *i.e.* toute l'information disponible depuis l'instant initial. Notons que

$$\begin{aligned} \mathbf{b}_{t+1}(s') &:= \mathbb{P}(S_{t+1} = s' | i_{t+1}) \\ &= \mathbb{P}(S_{t+1} = s' | i_t, a_t, o_{t+1} = o') \\ &= \frac{\mathbb{P}(S_{t+1} = s', o_{t+1} = o' | i_t, a_t)}{\mathbb{P}(o_{t+1} = o' | i_t, a_t)} \\ &= \frac{\mathbb{P}(S_{t+1} = s' | i_t, a_t) \mathbb{P}(o_{t+1} = o' | S_{t+1} = s', i_t, a_t)}{\sum_{\bar{s} \in \mathcal{S}} \mathbb{P}(S_{t+1} = \bar{s}, o_{t+1} = o' | i_t, a_t)}. \end{aligned}$$

On remarque que le dénominateur est la somme sur  $s' \in \mathcal{S}$  du numérateur. De même,

$$\begin{aligned} \mathbb{P}(S_{t+1} = s' | i_t, a_t) &= \sum_{s \in \mathcal{S}} \mathbb{P}(S_t = s | i_t, a_t) \mathbb{P}(S_{t+1} = s' | S_t = s, i_t, a_t) \\ &= \sum_{s \in \mathcal{S}} \mathbb{P}(S_t = s | i_t) \mathbb{P}(S_{t+1} = s' | S_t = s, a_t) \\ &= \sum_{s \in \mathcal{S}} \mathbf{b}_t(s) \mathbf{p}(s' | s, a_t) \end{aligned}$$

car l'état  $S_t$  ne dépend pas de l'action au temps  $t$  (mais de  $a_{t-1}$  !), et  $S_{t+1}$  ne dépend que de  $S_t$  et  $a_t$ , et pas de l'historique complet  $i_t$  (c'est un processus de Markov).

De plus, nous rappelons que l'observation  $o_{t+1}$  ne dépend pas de l'historique

complet à l'instant  $t$ , mais seulement de  $S_{t+1}$ , et  $a_t$ . Ainsi,

$$\begin{aligned} \mathbf{b}_{t+1}(s') &= \frac{\mathbf{p}(o' | s', a) \sum_{s \in \mathcal{S}} \mathbf{b}_t(s) \mathbf{p}(s' | s, a)}{\sum_{s, \bar{s} \in \mathcal{S}^2} \mathbf{b}_t(s) \mathbf{p}(\bar{s} | s, a) \mathbf{p}(o' | \bar{s}, a)} \\ &= \tau(\mathbf{b}_t, o', a). \end{aligned} \quad (3.1)$$

La suite des fonctions de croyance est donc une chaîne de Markov dont nous pouvons calculer les probabilités de transition :

$$\mathbb{P}(\mathbf{b}_{t+1} = \mathbf{b}' | \mathbf{b}_t = \mathbf{b}, a) = \sum_{o' \in \mathcal{O} \text{ tq } \mathbf{b}' = \tau(\mathbf{b}, a, o')} \mathbf{p}(o' | \mathbf{b}, a) \quad (3.2)$$

où

$$\mathbf{p}(o' | \mathbf{b}, a) = \sum_{s' \in \mathcal{S}} \mathbf{p}(o' | s', a) \sum_{s \in \mathcal{S}} \mathbf{p}(s' | s, a) \mathbf{b}(s).$$

En définissant des récompenses sur ces nouveaux états de croyance de la manière suivante

$$r(\mathbf{b}, a) = \sum_{s \in \mathcal{S}} r(s, a) \mathbf{b}(s),$$

nous obtenons un nouvel MDP dont l'équation de Bellman s'écrit

$$V_{i+1}^*(\mathbf{b}) = \max_{a \in \mathcal{A}} r(\mathbf{b}, a) + \sum_{o \in \mathcal{O}} \mathbf{p}(o | \mathbf{b}, a) V_i^*(\tau(\mathbf{b}, o, a)).$$

Cependant, l'espace d'état de ce MDP est l'ensemble des probabilités sur  $\mathcal{S}$ , soit un espace continu. Smallwood et Sondik ont apporté une manière de résoudre tout de même ce problème dans les années 1970 [6]. Ils ont montré pour cela que l'ensemble des fonctions linéaires par morceaux et convexes (LMC) était stable par l'itération de Bellman.

Ainsi comme  $V_0(\mathbf{b}) = \sum_s r(s, a) \mathbf{b}(s)$  est linéaire, nous ne manipulons que des fonctions LMC à chaque itération, ce qui est facilement faisable en informatique.

Ce modèle a cependant des défauts en pratique. En effet, les observations sont souvent issues d'algorithme de traitement du signal dont il est difficile d'extraire une loi de probabilité sur leurs sorties : les valeurs de  $\mathbf{p}(o' | s', a)$  peuvent être très peu fiables. La connaissance de  $\mathbf{b}_0$  pose aussi un problème : il existe des missions où le robot ne connaît initialement rien de l'état du système, ce que ne modélise pas tout à fait une probabilité uniforme. Lorsqu'aucune expérience passée n'est disponible, il n'est pas évident d'attribuer une fréquence à un évènement initial...

**3.2. Alternatives envisagées.** De nombreuses modélisations de l'incertitude alternatives existent dans la littérature. Il est possible, par exemple, de considérer qu'uniquement les bornes inférieures et supérieures des probabilités sont connues.

Durant mon stage de fin d'étude, je me suis intéressé à la théorie des possibilités (introduite par Dubois et Prade, [3]), et à un homologue possibiliste du modèle MDP [4]. Nous raisonnons sur un ensemble fini  $\Omega$ . Une mesure

de possibilité  $\Pi$  (qui n'est pas une mesure classique, mais une mesure floue, puisque non additive), est telle que  $\Pi : 2^\Omega \rightarrow [0, 1]$ , et  $\forall A, B \subset \Omega$ ,

- $\Pi(\emptyset) = 0$ ,  $\Pi(\Omega) = 1$  (normalisation),
- $\Pi(A \cup B) = \max(\Pi(A), \Pi(B))$  lorsque  $A \cap B = \emptyset$

La mesure de nécessité associée est :  $\mathcal{N}(A) = 1 - \Pi(A^c)$

- $\pi(\omega) = 1 \quad \forall \omega \in \Omega \implies \eta(\omega) = 1 - \Pi(\{\omega\}^c) = 0$  : les événements  $\omega$  sont possibles, mais pas nécessaires : c'est une vraie modélisation de l'**ignorance**.
- Une interprétation probabiliste est qu'à une possibilité est associée l'ensemble des mesures de probabilité  $\mathbb{P}$  telles que  $\Pi(A) \geq \mathbb{P}(A)$ .
- $\pi_1$  est plus **spécifique** (donne plus d'information) que  $\pi_2$  si  $\forall \omega \in \Omega$ ,  $\pi_1(\omega) \leq \pi_2(\omega)$ .

Ainsi, cet outil permet la modélisation du manque de connaissance, ce qui est difficile avec des probabilités.

Il est aussi possible d'affecter de la masse  $m$  à tous les sous-ensembles de  $\Omega$ , et pas seulement aux singletons ( $\sum_{A \in \Omega} m(A) = 1$ ) : c'est la théorie de Dempster-Shafer [5], qui permet de borner une distribution de probabilité entre la plausibilité  $Pl(A) = \sum_{\{B \mid B \cap A \neq \emptyset\}} m(B)$  et la croyance  $Bel(A) = \sum_{\{B \subset A\}} m(B)$ .

#### 4. CONCLUSION

Bien d'autres modèles existent, et pourraient permettre l'amélioration du modèle POMDP. L'intérêt que je porte à ce problème, est qu'il mêle théorie et réflexion sur la modélisation en pratique. Les modèles alternatifs qui seront proposés seront utilisés pour la détection de cible (voiture) par un petit hélicoptère automatique [2].

#### RÉFÉRENCES

- [1] Richard Bellman. A Markovian Decision Process. *Indiana Univ. Math. J.*, 6 :679–684, 1957.
- [2] Caroline Ponzoni Carvalho Chanele, Jean-Loup Farges, Florent Teichteil-Königsbuch, and Guillaume Infantes. Pomdp solving : what rewards do you really expect at execution? In Thomas Ågotnes, editor, *STAIRS*, volume 222 of *Frontiers in Artificial Intelligence and Applications*, pages 50–62. IOS Press, 2010.
- [3] Didier Dubois and Henri Prade. *Possibility Theory : An Approach to Computerized Processing of Uncertainty (traduction revue et augmentée de "Théorie des Possibilités")*. Plenum Press, New York, 1988.
- [4] Régis Sabbadin. A possibilistic model for qualitative sequential decision problems under uncertainty in partially observable environments. In *15th Conference on Uncertainty in Artificial Intelligence (UAI99)*, Stockholm, 30/07/99-01/08/99, pages 567–564, San Francisco, juillet 1999. Morgan Kaufmann.
- [5] Glenn Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, Princeton, 1976.
- [6] Richard D. Smallwood and Edward J. Sondik. *The Optimal Control of Partially Observable Markov Processes Over a Finite Horizon*, volume 21. INFORMS, 1973.



## **Rapport de stage de fin d'étude**

M2-R Mathématiques Vision Apprentissage (ENS Cachan)  
Magistère de Mathématiques d'Orsay (Paris-Sud XI)

Organise d'accueil : ONERA - The french Aerospace Lab  
Maîtres de stage : Florent Teichtel et Jean-Loup Farges

# **Percevoir pour planifier et planifier pour percevoir :**

intégration fine et duale de sources d'information imprécises et de  
problèmes de décision séquentielle dans l'incertain

---

**Nicolas Drougard**

Toulouse, le 1<sup>er</sup> octobre 2012

## TABLE DES MATIÈRES

|  |    |
|--|----|
| Notations  | 2  |
| 1. Introduction  | 3  |
| 2. MDP-POMDP   | 3  |
| 2.1. Processus Decisionnel Markovien                                 | 3  |
| 2.1.1. Définition  | 4  |
| 2.1.2. Résolution d'un MDP   | 5  |
| 2.2. Partially Observable Markov Decision Process                    | 10 |
| 2.2.1. Définition  | 11 |
| 2.2.2. Fonction de croyance  | 12 |
| 2.2.3. Fonctions valeurs PWLC et résolution d'un POMDP               | 15 |
| 3. Alternatives à la modélisation de l'incertitude                   | 17 |
| 3.1. Théorie des Possibilités  | 17 |
| 3.1.1. Consistance, Préservation de la préférence et Transformations | 21 |
| 3.1.2. Critères qualitatifs de décision                              | 25 |
| 3.2. Le modèle $\pi$ -MDP  | 29 |
| 3.2.1. Mise en place du modèle                                       | 29 |
| 3.2.2. Résolution d'un $\pi$ -MDP à horizon quelconque               | 33 |
| 3.2.3. Application du $\pi$ -MDP en robotique                        | 39 |
| 3.3. Le modèle $\pi$ -POMDP  | 39 |
| 3.3.1. Distinction entre variables visibles et invisibles            | 44 |
| 3.3.2. Application à l'identification de cibles                      | 48 |
| 4. Conclusion  | 50 |
| Références   | 51 |

## NOTATIONS

Dans ce rapport, nous adopterons ces quelques notations :

- $\mathbb{P}$  est une mesure de probabilité sur  $\mathcal{X}$ ,
  - $\mathbf{p}(y) = \mathbb{P}(\{Y = y\})$  si  $Y$  est la variable aléatoire dont les valeurs possibles sont les  $y \in \mathcal{X}$ . Cette notation tiendra pour les probabilités conditionnelles  $\mathbf{p}(y | z) = \mathbb{P}(Y = y | Z = z)$ ,
  - $\Delta\mathcal{X}$  est l'ensemble des mesures de probabilité sur  $\mathcal{X}$ .  
De même,
  - $\Pi$  est une mesure de possibilité,
  - $\pi(y) = \Pi(\{Y = y\})$ ,
  - $\mathcal{N}$  est une mesure de nécessité,
  - $\eta(y) = \mathcal{N}(\{Y = y\})$ ,
  - $\Sigma\mathcal{X}$  est l'ensemble des mesures de possibilité sur  $\mathcal{X}$ .
- 
- Lorsque des processus évoluent dans le temps, par exemple  $((y_t) \in \mathcal{X}^{\mathbb{N}})$ , nous noterons au temps  $t$ ,  $y$  les valeurs possibles de  $y_t$ , et  $y'$  les valeurs possibles de  $y_{t+1}$ ,

- $A^c$  sera l'ensemble complémentaire de  $A$ .

## 1. INTRODUCTION

Ce stage a été effectué au département DCSD (Département Commande des Systèmes et Dynamique du vol) de l'ONERA. Les recherches de ce département permettent l'amélioration des fonctions de contrôle et de conduite des systèmes, notamment autour de automatique et des systèmes aérospatiaux.

Le modèle POMDP, autour duquel ce travail est développé, est utilisé lorsque le système peut se modéliser à l'aide de probabilités. Il permet aussi bien de modéliser les incertitudes sur l'observation du système, que sur sa dynamique. Nous commencerons donc par présenter un modèle plus simple (MDP, où l'observation est parfaite) afin de construire plus facilement le modèle POMDP.

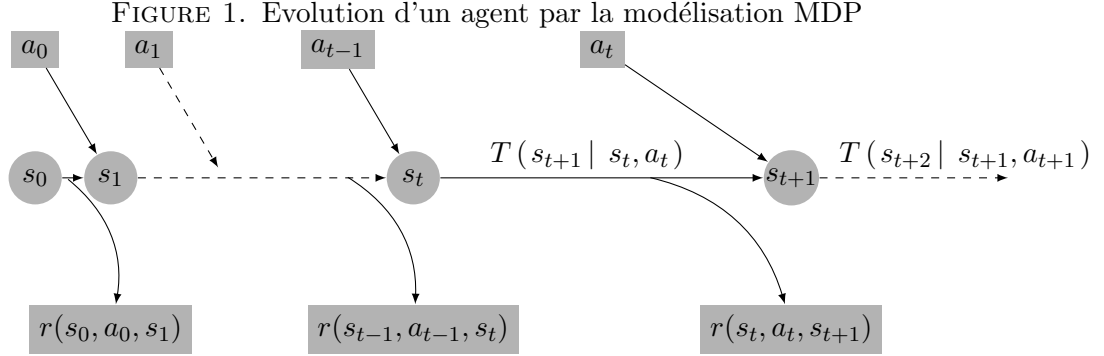
Dans le modèle POMDP, les distributions de probabilité sont supposées parfaitement connues a priori. En pratique, ce n'est pas toujours le cas, notamment lorsque les observations de l'environnement sont filtrées par des algorithmes de traitement du signal. De plus, l'agent est supposé avoir une croyance initiale de l'état du système : sa croyance initiale n'est pas précise en pratique (extraire des fréquences d'événements initiaux n'est pas forcément possible), et une probabilité uniforme ne représente pas réellement cette méconnaissance. Il existe aussi des situations dans lesquelles, donner de l'importance à l'information de la croyance (au sens entropique) améliore la politique en pratique [4].

Ces problèmes pratiques peuvent trouver des solutions grâce à d'autres modélisations de l'incertitude que la théorie des probabilités. C'est pourquoi ce rapport s'intéresse ensuite à des modèles utilisant la théorie des possibilités ( $\pi$ -MDP et  $\pi$ -POMDP). Ces deux modèles introduits par R.Sabbadin sont présentés, puis une preuve est donnée à l'homologue de l'algorithme d'itération sur les valeurs. Enfin, l'astuce de O.Buffet *et al.* est adaptée pour le modèle  $\pi$ -POMDP.

## 2. MDP-POMDP

Ces modèles probabilistes sont conçus pour représenter de manière simple une situation où une entité (appelée *agent*) et le monde qui l'entoure peuvent être dans différents états  $s \in \mathcal{S}$  au cours du temps (ce dernier, modélisé par l'ensemble des entiers naturels  $\mathbb{N}$ ). L'agent choisit une action parmi celles dont il dispose à chaque étape de temps  $t \in \mathbb{N}$ , et le système comprenant l'agent et son environnement évolue de manière Markovienne dans  $\mathcal{S}$  ([2],[3]).

### 2.1. Processus Decisionnel Markovien.



2.1.1. *Définition.* Un MDP (pour Markov Decision Process) est défini par le 5-uplet  $\langle \mathcal{S}, \mathcal{A}, T, r, s_0 \rangle$  :

- $\mathcal{S}$  l'ensemble des états où peuvent se trouver l'agent et son environnement,
- $\mathcal{A}$  l'ensemble fini des actions que l'agent peut effectuer (cet ensemble dépend parfois de l'état courant :  $\mathcal{A}_s$ ),
- $T$  l'ensemble des probabilités de transition du système d'un état  $s$  à un état  $s'$  sous l'action  $a$  :  $\forall s \in \mathcal{S}, \forall a \in \mathcal{A}$ , la mesure de probabilité  $\mathbf{p}(\cdot | s, a) \in T$  est définie sur  $\mathcal{S}$ . Afin d'avoir des notations similaires au modèle POMDP, nous noterons  $\forall s' \in \mathcal{S}, T(s' | s, a) := \mathbf{p}(s' | s, a)$ .
- $r(s, a, s')$  la récompense obtenue lorsque l'agent utilise l'action  $a$  et que le système passe de l'état  $s$  à l'état  $s'$ . Cette fonction est bornée sur  $\mathcal{S} \times \mathcal{A} \times \mathcal{S}$ .

Supposons que nous ayons un plan, *i.e.* une suite d'actions indexées par le temps  $t \in \mathbb{N}$  :  $(a_t)_{t \in \mathbb{N}}$ .

L'agent est initialement dans l'état  $s_0$ . Ensuite, à chaque étape de temps ( $t = 0, 1, \dots$ ), l'agent est dans l'état  $s_t \in \mathcal{S}$ , exécute l'action  $a_t \in \mathcal{A}_{(s)}$  et change d'état selon les lois de probabilité de transition  $T(\cdot | s_t, a_t)$  pour arriver dans l'état  $s_{t+1} \in \mathcal{S}$ .

Enfin, il obtient la récompense  $r(s_t, a_t, s_{t+1})$  qu'il prend en compte avec les récompenses obtenues durant les précédentes étapes.

La figure 1 illustre ce processus.

Une politique est une fonction dépendant du temps (de l'étape du processus), et de l'état courant :

$$d : (\mathbb{N}, \mathcal{S}) \rightarrow \mathcal{A}$$

$$(t, s) \mapsto a_t$$

Nous présentons ici uniquement le cas de l'horizon infini, c'est-à-dire lorsqu'il n'y a pas de limite dans le temps. La politique recherchée, notée ici  $d$ , est dite *stationnaire i.e.* elle ne dépend pas du temps. Comme elle ne dépend

de l'état courant  $s_t$ , elle est dite *en boucle fermée* :  $\left\{ \begin{array}{lcl} d : \mathcal{S} & \longrightarrow & \mathcal{A} \\ s_t & \longmapsto & a_t \end{array} \right.$

Nous appelons *récompense actualisée* au temps  $t$ , la quantité  $\gamma^t r(s_t, a_t, s_{t+1})$ , avec  $0 < \gamma < 1$ . Ce facteur  $\gamma$  permet la convergence de la somme actualisée des récompenses  $\sum_{t=0}^{+\infty} \gamma^t r(s_t, d(s_t), s_{t+1})$  quelle que soit la trajectoire  $(s_t)_{t \in \mathbb{N}}$  et la politique  $d$ . En effet, la fonction récompense  $r$  étant bornée,  $\exists B > 0$  tel que  $\forall s, s' \in \mathcal{S}, \forall a \in \mathcal{A}, r(s, a, s') < B$ . Ainsi,

$$\begin{aligned} \sum_{t=0}^{+\infty} \gamma^t r(s_t, d(s_t), s_{t+1}) &\leq B \sum_{t=0}^{+\infty} \gamma^t \\ &\leq \frac{B}{1-\gamma} < +\infty \end{aligned}$$

Il est possible d'interpréter ce facteur  $\gamma$  comme la probabilité que le processus continue après chaque étape de temps  $t$ , ou bien comme une manière de modéliser le fait que les récompenses à long terme importent moins qu'à court terme.

Ainsi, une fois qu'un problème a été modélisé en MDP, le résoudre consiste à trouver la politique maximisant l'espérance de la somme actualisée des récompenses. Cette quantité, appelée *fonction valeur*, est notée  $V$  et dépend de l'état initial :

$$V^d(s) := \mathbb{E} \left[ \sum_{t=0}^{+\infty} \gamma^t r(s_t, d(s_t), s_{t+1}) \mid s_0 = s \right] \quad (2.1)$$

**2.1.2. Résolution d'un MDP.** L'espace des états  $\mathcal{S}$  n'est pas forcément fini. Cependant, nous faisons l'hypothèse que  $\forall s \in \mathcal{S}, \forall a \in \mathcal{A}$ , le support des probabilités de transitions  $T(\cdot \mid s, a)$  est fini, *i.e.*  $\exists \mathcal{S}_{s,a} \subset \mathcal{S}$ , tel que  $\text{Card}(\mathcal{S}_{s,a}) < +\infty$ , et  $\forall s' \in \mathcal{S} - \mathcal{S}_{s,a}, T(s' \mid s, a) = 0$ .

Les quelques lignes de calcul qui suivent nous permettent de dégager une équation caractérisant cette fonction valeur 2.1 : l'équation de Bellman.

$$\begin{aligned}
 V^d(s) &:= \mathbb{E} \left[ \sum_{t=0}^{+\infty} \gamma^t r(s_t, d(s_t), s_{t+1}) \mid s_0 = s \right] \\
 &= \mathbb{E} \left[ r(s_0, d(s_0), s_1) + \sum_{t=1}^{+\infty} \gamma^t r(s_t, d(s_t), s_{t+1}) \mid s_0 = s \right] \\
 &= \sum_{s' \in \mathcal{S}_{s, d(s)}} T(s' \mid s, d(s)) r(s, d(s), s') + \mathbb{E} \left[ \mathbb{E} \left[ \sum_{t=1}^{+\infty} \gamma^t r(s_t, d(s_t), s_{t+1}) \mid s_1 = s' \right] \mid s_0 = s \right] \\
 &= \sum_{s' \in \mathcal{S}_{s, d(s)}} T(s' \mid s, d(s)) \left( r(s, d(s), s') + \gamma \mathbb{E} \left[ \sum_{t=1}^{+\infty} \gamma^{t-1} r(s_t, d(s_t), s_{t+1}) \mid s_1 = s' \right] \right) \\
 &= \sum_{s' \in \mathcal{S}_{s, d(s)}} T(s' \mid s, d(s)) \left( r(s, d(s), s') + \gamma \mathbb{E} \left[ \sum_{t'=0}^{+\infty} \gamma^{t'} r(s_{t'}, d(s_{t'}), s_{t'+1}) \mid s_0 = s' \right] \right) \\
 &= \sum_{s' \in \mathcal{S}_{s, d(s)}} T(s' \mid s, d(s)) \left( r(s, d(s), s') + \gamma V^d(s') \right) \tag{2.2}
 \end{aligned}$$

Nous obtenons la quatrième ligne grâce à la propriété de Markov. Notons  $\mathcal{F}_B(\mathcal{S}, \mathbb{R})$  l'ensemble des fonctions bornées de  $\mathcal{S}$  dans  $\mathbb{R}$ .

L'opérateur de Bellman

$$\begin{cases} \mathcal{B}^d : \mathcal{F}_B(\mathcal{S}, \mathbb{R}) \longrightarrow \mathcal{F}_B(\mathcal{S}, \mathbb{R}) \\ s \mapsto V(s) \longmapsto s \mapsto (\mathcal{B}^d V)(s) := \sum_{s' \in \mathcal{S}_{s, d(s)}} T(s' \mid s, d(s)) (r(s, d(s), s') + \gamma V(s')) \end{cases}$$

permet de réécrire l'équation de Bellman 2.2 :

$$V^d = \mathcal{B}^d V^d$$

Considérons la norme sup sur  $\mathcal{F}_B(\mathcal{S}, \mathbb{R})$  :  $\|V\|_\infty = \sup_{s \in \mathcal{S}} |V(s)|$ .

Puisque  $(\mathbb{R}, |\cdot|)$  est complet,  $(\mathcal{F}_B(\mathcal{S}, \mathbb{R}), \|\cdot\|_\infty)$  est un espace de Banach.

En effet, soit une suite de Cauchy  $(V_n)_{n \in \mathbb{N}}$  dans  $(\mathcal{F}_B(\mathcal{S}, \mathbb{R}), \|\cdot\|_\infty)$ .

$\forall s \in \mathcal{S}, (V_n(s))_{n \in \mathbb{N}}$  est une suite de Cauchy de  $(\mathbb{R}, |\cdot|)$  car  $|V_n(s)| \leq \|V_n\|_\infty$ .

Ainsi,  $\forall s \in \mathcal{S}, (V_n(s))_{n \in \mathbb{N}}$  a une limite dans  $\mathbb{R}$  que l'on note  $V(s)$ . Une suite de Cauchy est bornée, donc  $\exists M > 0$  tel que  $\forall n \in \mathbb{N}, \|V_n\|_\infty \leq M$ . Ainsi,  $\forall s \in \mathcal{S}, |V_n(s)| \rightarrow |V(s)| \leq M$ , et donc  $V \in \mathcal{F}_B(\mathcal{S}, \mathbb{R})$ .

Enfin,  $(V_n)$  étant de Cauchy,

$$\begin{aligned}
 \forall \varepsilon > 0, \exists N \geq 0, \text{ tel que } \forall n \geq N, p \geq 0, \forall s \in \mathcal{S}, & |V_n(s) - V_{n+p}(s)| < \varepsilon \\
 \Rightarrow \forall n \geq N, \forall s \in \mathcal{S}, & |V_n(s) - V(s)| < \varepsilon \\
 \Rightarrow \forall n \geq N, & \|V_n - V\|_\infty < \varepsilon
 \end{aligned}$$

Ainsi  $V_n \rightarrow V$  dans  $\mathcal{F}_B(\mathcal{S}, \mathbb{R})$  lorsque  $n \rightarrow +\infty$ , et donc  $(\mathcal{F}_B(\mathcal{S}, \mathbb{R}), \|\cdot\|_\infty)$  est bien un espace de Banach.

Nous montrons maintenant que  $\mathcal{B}^d$  est un opérateur contractant sur  $\mathcal{F}_B(\mathcal{S}, \mathbb{R})$  : soient  $(V, V') \in \mathcal{F}_B(\mathcal{S}, \mathbb{R})^2$ ,

$$\begin{aligned} \left\| \mathcal{B}^d V - \mathcal{B}^d V' \right\|_{\infty} &\leq \gamma \sup_{s \in \mathcal{S}} \sum_{s' \in \mathcal{S}_{s, d(s)}} T(s' \mid s, d(s)) |V(s') - V'(s')| \\ &\leq \gamma \sup_{s \in \mathcal{S}} \sum_{s' \in \mathcal{S}_{s, d(s)}} T(s' \mid s, d(s)) \|V - V'\|_{\infty} \\ &\leq \gamma \|V - V'\|_{\infty} \end{aligned}$$

puisque  $\forall s \in \mathcal{S}$ ,  $T(\cdot \mid s, d(s))$  est une distribution de probabilité.

Le théorème du point fixe de Picard nous affirme alors que l'équation de Bellman  $V = \mathcal{B}^d V$  admet une unique solution, et c'est donc la fonction valeur  $V^d \in \mathcal{F}_B(\mathcal{S}, \mathbb{R})$ . Cette équation est donc bien une caractérisation de la fonction valeur pour une politique  $d$  donnée.

Introduisons maintenant l'opérateur de Programmation Dynamique  $\mathcal{B}^*$  :

$$\left\{ \begin{array}{ll} \mathcal{B}^* : \mathcal{F}_B(\mathcal{S}, \mathbb{R}) &\longrightarrow \mathcal{F}_B(\mathcal{S}, \mathbb{R}) \\ s \mapsto V(s) &\longmapsto s \mapsto (\mathcal{B}^* V)(s) := \max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}_{s, a}} T(s' \mid s, a) (r(s, a, s') + \gamma V(s')) \end{array} \right.$$

Afin d'alléger les notations, notons pour  $a \in \mathcal{A}$ ,

$$\left\{ \begin{array}{ll} \mathcal{B}^a : \mathcal{F}_B(\mathcal{S}, \mathbb{R}) &\longrightarrow \mathcal{F}_B(\mathcal{S}, \mathbb{R}) \\ s \mapsto V(s) &\longmapsto (\mathcal{B}^a V)(s) := \sum_{s' \in \mathcal{S}_{s, a}} T(s' \mid s, a) (r(s, a, s') + \gamma V(s')) \end{array} \right.$$

La remarque, ainsi que les quelques lignes qui la suivent montrent que  $\mathcal{B}^*$  est aussi contractant.

**Remarque :** Soient  $f$  et  $g$  deux fonctions de  $\mathcal{A}$  dans  $\mathbb{R}$ .

$$\begin{aligned} \forall a \in \mathcal{A}, \quad f(a) - g(a) &\leq \max_{a \in \mathcal{A}} |f(a) - g(a)| \\ \Rightarrow \forall a \in \mathcal{A}, \quad f(a) &\leq \max_{a \in \mathcal{A}} g(a) + \max_{a \in \mathcal{A}} |f(a) - g(a)| \\ \Rightarrow \max_{a \in \mathcal{A}} f(a) - \max_{a \in \mathcal{A}} g(a) &\leq \max_{a \in \mathcal{A}} |f(a) - g(a)| \end{aligned}$$

Enfin, les mêmes inégalités tiennent en partant de  $g(a) - f(a)$ , ainsi

$$\left| \max_{a \in \mathcal{A}} f(a) - \max_{a \in \mathcal{A}} g(a) \right| \leq \max_{a \in \mathcal{A}} |f(a) - g(a)|$$

L'inégalité de contraction de l'opérateur  $\mathcal{B}^d$  pour  $(V, V') \in \mathcal{F}_B(\mathcal{S}, \mathbb{R})^2$  est vraie pour toute politique stationnaire  $d$  :

$$\begin{aligned} \forall d : \mathcal{S} &\mapsto \mathcal{A} \quad \left\| \mathcal{B}^d V - \mathcal{B}^d V' \right\|_{\infty} \leq \gamma \|V - V'\|_{\infty} \\ \Rightarrow \forall d \text{ politique}, \forall s \in \mathcal{S}, \quad &\left| (\mathcal{B}^d V)(s) - (\mathcal{B}^d V')(s) \right| \leq \gamma \|V - V'\|_{\infty} \\ \Rightarrow \forall a \in \mathcal{A}, \forall s \in \mathcal{S}, \quad &\left| (\mathcal{B}^a V)(s) - (\mathcal{B}^a V')(s) \right| \leq \gamma \|V - V'\|_{\infty} \\ \Rightarrow \forall s \in \mathcal{S}, \quad &\max_a \left| (\mathcal{B}^a V)(s) - (\mathcal{B}^a V')(s) \right| \leq \gamma \|V - V'\|_{\infty} \end{aligned}$$

et la remarque nous donne alors  $\forall s \in \mathcal{S}$ ,  
 $|\max_a(\mathcal{B}^a V)(s) - \max_a(\mathcal{B}^a V')(s)| \leq \gamma \|V - V'\|_\infty$

$$\Rightarrow \|\mathcal{B}^* V - \mathcal{B}^* V'\|_\infty \leq \gamma \|V - V'\|_\infty$$

Le théorème du point fixe nous permet d'affirmer que l'équation de programmation dynamique

$$V = \mathcal{B}^* V \quad (2.3)$$

admet une unique solution dans  $\mathcal{F}_B(\mathcal{S}, \mathbb{R})$  que l'on note  $V^*$ .

Nous montrons maintenant que  $V^*$  est la fonction valeur optimale<sup>1</sup>, *i.e.*

$$V^*(s) = V^{opt}(s) := \max_{d \text{ politique}} \mathbb{E} \left[ \sum_{t=0}^{+\infty} \gamma^t r(s_t, d(s_t), s_{t+1}) \mid s_0 = s \right]$$

Soit  $s \in \mathcal{S}$ , et  $d = (a, d')$  la politique consistant à faire l'action  $a$  au temps  $t = 0$ , puis les actions  $(a_1 := d'_1(s_1), a_2 := d'_2(s_2), \dots)$  ensuite. Moyennant un décalage d'origine en  $t$ ,  $d'$  peut également être considéré comme une politique. Cette politique n'est pas forcément stationnaire ( $d'_{t_1} : \mathcal{S} \rightarrow \mathcal{A}$  peut être différent de  $d'_{t_2}$  pour  $t_1 \neq t_2 \in \mathbb{N}^*$ ).

$$\begin{aligned} V^{opt}(s) &:= \max_{d \text{ politique}} \mathbb{E} \left[ \sum_{t=0}^{+\infty} \gamma^t r(s_t, d(s_t), s_{t+1}) \mid s_0 = s \right] \\ &= \max_{(a, d') \text{ politique}} \sum_{s' \in \mathcal{S}_{s,a}} T(s' \mid s, a) \left( r(s, a, s') + \gamma V^{d'}(s') \right) \\ (2.4) \quad &= \max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}_{s,a}} T(s' \mid s, a) \left( r(s, a, s') + \gamma \max_{d' \text{ politique}} V^{d'}(s') \right) \\ &= \max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}_{s,a}} T(s' \mid s, a) \left( r(s, a, s') + \gamma V^{opt}(s') \right) \\ &= \mathcal{B}^* V^{opt} \end{aligned}$$

(toujours par la propriété de Markov).

L'égalité (2.4) se justifie par

- l'inégalité  $\max_{d' \text{ politique}} \sum_{s \in \mathcal{S}_{s,a}} T(s' \mid s, a) V^{d'} \leq \sum_{s \in \mathcal{S}_{s,a}} T(s' \mid s, a) \max_{d' \text{ politique}} V^{d'}$
- et le fait que pour une politique  $\bar{d} = (\bar{d}_0, \bar{d}_1, \dots)$  définie par  $\bar{d}(s') := (\bar{d}_0(s'), \bar{d}_1(s'), \dots) = \operatorname{argmax}_{d'} V^{d'}(s')$ ,  

$$\begin{aligned} \sum_{s \in \mathcal{S}_{s,a}} T(s' \mid s, a) \max_{d' \text{ politique}} V^{d'} &= \sum_{s \in \mathcal{S}_{s,a}} T(s' \mid s, a) V^{\bar{d}} \\ &\leq \max_{d' \text{ politique}} \sum_{s \in \mathcal{S}_{s,a}} T(s' \mid s, a) V^{d'} \end{aligned}$$

Nous obtenons bien  $V^* = V^{opt}$  par unicité de la solution de l'équation de Programmation Dynamique 2.3. Nous avons donc aussi montré qu'il suffisait

---

1. Démonstration issue du cours de R.Munos <http://researchers.lille.inria.fr/~munos/master-mva/lecture01.pdf>



de chercher parmi les politiques stationnaires.

**Parenthèse :** Ecrire  $V^{opt}(s) := \max_d \mathbb{E} [\sum_{t=0}^{+\infty} \gamma^t r(s_t, d(s_t), s_{t+1}) \mid s_0 = s]$  est abusif avant de savoir que le maximum est atteint, puisque l'ensemble des politiques est éventuellement infini. Dans le cas où  $\text{Card } \mathcal{S} < \infty$ , comme  $\text{Card } \mathcal{A} < \infty$ , le nombre de politiques est égal à  $(\text{Card } \mathcal{A})^{\text{Card } \mathcal{S}}$  et la démonstration est vraie.

Cela ne change en rien le résultat lorsque  $\mathcal{S}$  n'est pas un ensemble fini : il suffit de remplacer les  $\max_d$  par des  $\sup_d$  dans les égalités, et dans le premier point de la justification. Pour le deuxième point, nous pouvons écrire que  $\forall \varepsilon > 0$ , il existe une politique  $\bar{d}_\varepsilon$  telle que

$$\begin{aligned} \sum_{s \in S_{s,a}} T(s' \mid s, a) \sup_{d' \text{ politique}} V^{d'} - \varepsilon &\leq \sum_{s \in S_{s,a}} T(s' \mid s, a) V^{\bar{d}_\varepsilon} \\ &\leq \sup_{d' \text{ politique}} \sum_{s \in S_{s,a}} T(s' \mid s, a) V^{d'} \end{aligned}$$

L'inégalité pour tout  $\varepsilon > 0$  implique l'inégalité sans le  $\varepsilon$ , et la parenthèse se ferme.

Définissons maintenant  $d^* : \mathcal{S} \ni s \mapsto a \in \arg\max_{a \in A} (\mathcal{B}^a V^*)(s)$  (souvent appelée politique *gloutonne*).

$$\mathcal{B}^{d^*} V^* = \mathcal{B}^* V^* = V^* = V^{opt}$$

$V^*$  est alors un point fixe de  $\mathcal{B}^{d^*}$  :  $V^* = V^{opt}$  est donc égal à  $V^{d^*}$ . Cela signifie que  $d^*$  est une politique optimale.

Lorsque  $\mathcal{S}$  est fini, l'algorithme dit d'*itération sur les valeurs* noté  $IV_{MDP}$ , directement issu du théorème du point fixe, permet de calculer la fonction valeur optimale.

Ce dernier nous assure en effet que  $\forall V_0 \in \mathcal{F}_B(\mathcal{S}, \mathbb{R})$ ,  $(\mathcal{B}^*)^n V_0 \rightarrow V^*$  lorsque  $n \rightarrow +\infty$  (en norme  $\|\cdot\|_\infty$ ). Si l'on connaît un majorant de  $\|V^{opt} - V^0\|_\infty$  (disons  $M > 0$ ), comme  $\mathcal{B}^*$  est contractant de contraction  $\gamma$ , l'algorithme retourne une fonction valeur  $V^N$  (notée  $V$  dans Algo 1) qui vérifie

$$\begin{aligned} \|V^N - V^*\|_\infty &= \|(\mathcal{B}^*)^N V^0 - V^*\|_\infty = \|(\mathcal{B}^*)^N V^0 - (\mathcal{B}^*)^N V^*\|_\infty \\ (2.5) \quad &\leq \gamma^N \|V^0 - V^*\|_\infty \leq M \gamma^N \rightarrow 0 \\ &\quad (\text{lorsque } N \rightarrow +\infty) \end{aligned}$$

Cela nous permet alors de choisir le bon nombre d'itérations ( $N \in \mathbb{N}^*$ ) selon la borne sur le regret souhaité.

Enfin, l'équation de Bellman pour la politique  $d$  retournée par l'algorithme est  $V^d = \mathcal{B}^d V^d$ , et  $V^N = \mathcal{B}^* V^{N-1} = \mathcal{B}^d V^{N-1}$  vient de la dernière itération de l'algorithme (en notant  $V^{N-1}$  l'avant dernière fonction valeur calculée).

**Algorithm 1** Value Iteration Algorithm for MDP

---

```

 $V \leftarrow V^0$ 
 $i \leftarrow 1$ 
while  $i \leq N$  do
  for  $s \in \mathcal{S}$  do
     $V'(s) \leftarrow \max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} T(s' | s, a) (r(s, a, s') + \gamma V(s')) \quad (= (\mathcal{B}^* V)(s))$ 
  end for
   $V \leftarrow V'$ 
   $i++$ 
end while

for  $s \in \mathcal{S}$  do
   $d(s) \in \operatorname{argmax}_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} T(s' | s, a) (r(s, a, s') + \gamma V(s'))$ 
end for
return  $V, d$ 

```

---

Ces deux égalités nous permettent d'écrire

$$\begin{aligned}
\|V^d - V^N\|_\infty &= \|\mathcal{B}^d V^d - \mathcal{B}^d V^{N-1}\|_\infty \leq \gamma \|V^d - V^{N-1}\|_\infty \\
&\leq \gamma (\|V^d - V^N\|_\infty + \|V^N - V^{N-1}\|_\infty) \\
&\Rightarrow \|V^d - V^N\|_\infty \leq \frac{\gamma}{1-\gamma} \|V^N - V^{N-1}\|_\infty \quad (2.6)
\end{aligned}$$

Ainsi, (2.5) et 2.6 nous donnent

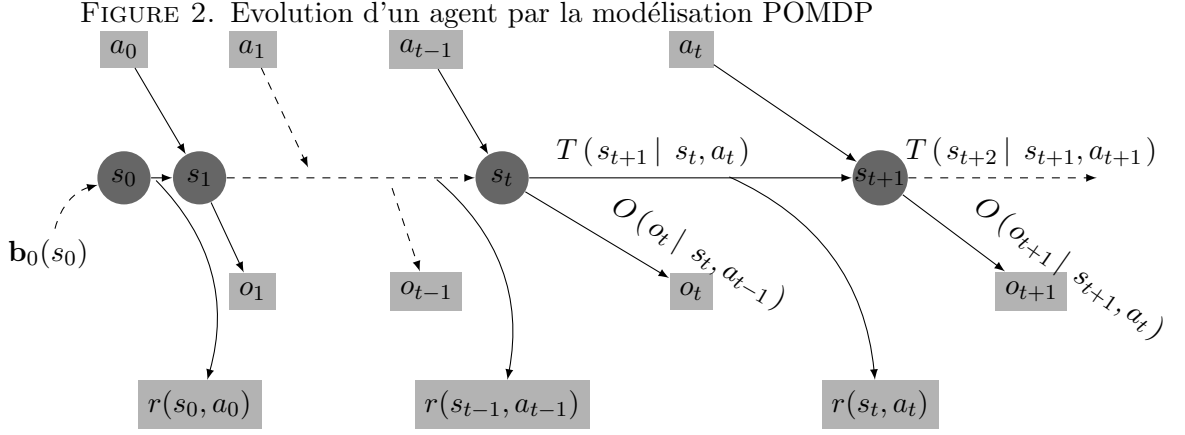
$$\begin{aligned}
\|V^d - V^*\|_\infty &\leq \frac{\gamma}{1-\gamma} (\|V^N - V^*\|_\infty + \|V^* - V^{N-1}\|_\infty) \\
&\leq \frac{\gamma}{1-\gamma} (M\gamma^N + M\gamma^{N-1}) \\
&\leq \frac{M(1+\gamma)}{1-\gamma} \gamma^N \longrightarrow 0 \text{ lorsque } N \rightarrow +\infty
\end{aligned}$$

Nous connaissons donc une borne sur le regret de la politique calculée, en fonction du nombre d'itérations.

Si l'on ne peut pas calculer  $M$  facilement, il existe d'autres bornes intéressantes (voir Puterman) comme

$$\begin{aligned}
\|V^N - V^*\| &\leq \frac{\gamma}{1-\gamma} \|V^N - V^{N-1}\| \\
&= \frac{\gamma^N}{1-\gamma} \|V^1 - V^0\|
\end{aligned}$$

**2.2. Partially Observable Markov Decision Process.** Le Processus Décisionnel Markovien Partiellement Observable est une généralisation du modèle MDP qui permet de modéliser les cas où il est impossible pour l'agent de connaître l'état du système avec certitude. Il doit se servir seulement des observations qu'il reçoit à chaque étape de temps pour prendre une décision d'action.



2.2.1. *Définition.* Un POMDP est défini par un 7-uplet  $\langle \mathcal{S}, \mathcal{A}, \mathcal{O}, T, O, r, \mathbf{b}_0 \rangle$  :

- $\mathcal{S}$  l'ensemble fini des états où peut se trouver le système,
- $\mathcal{A}$  l'ensemble fini des actions que l'agent peut effectuer,
- $\mathcal{O}$  l'ensemble fini des observations que peut recevoir l'agent,
- $T$  l'ensemble des probabilités de transition du système d'un état  $s$  à un état  $s'$  sous l'action  $a$  :  $\forall s \in \mathcal{S}, a \in \mathcal{A}$ , les distributions  $\mathbf{p}(\cdot | s, a) \in T$  sont définies sur  $\mathcal{S}$ . Pour mieux distinguer les différentes lois de probabilité (de transition, d'observation, de croyance ...), nous écrirons  $T(\cdot | s, a)$  à la place de  $\mathbf{p}(\cdot | s, a)$ ,
- $O$  l'ensemble des probabilités que l'agent observe  $o'$  si il a atteint l'état  $s'$  sous l'action  $a$  :  $\forall s' \in \mathcal{S}, a \in \mathcal{A}$ , les distributions  $\mathbf{p}(\cdot | s', a) \in O$  sont définies sur  $\mathcal{O}$ . Nous les notons  $O(\cdot | s', a)$ ,
- $r(s, a, s')$  la récompense obtenue par l'agent dans la situation décrite pour  $O$ , lorsqu'il part de  $s$ <sup>2</sup>,
- $bel_0 \in \Delta \mathcal{S}$  telle que  $\forall s \in \mathcal{S}, bel_0(s) = \mathbb{P}(s_0 = s)$  la croyance initiale sur les états. C'est une probabilité épistémique dans le sens où elle demande d'être interprétée de manière subjective (dans le monde de l'agent, l'état courant est une variable aléatoire puisqu'il ne la perçoit pas).

Considérons une politique  $(a_t)_{t \in \mathbb{N}}$ . L'agent est dans l'état  $s_0$  avec probabilité  $\mathbf{b}_0$ . Ensuite, tout comme le modèle MDP, à chaque étape de temps ( $t = 0, 1, \dots$ ) l'agent est dans l'état  $s_t \in \mathcal{S}$ , exécute l'action  $a_t \in \mathcal{A}_{(s)}$  et change d'état selon les lois de probabilité de transition  $T(\cdot | s_t, a_t)$  pour arriver dans l'état  $s_{t+1} \in \mathcal{S}$ . Il reçoit alors la récompense  $r(s_t, a_t, s_{t+1})$  et enfin l'observation  $o_{t+1}$  selon les lois de probabilité d'observation  $O(\cdot | s_{t+1}, a_t)$ . La figure 2 résume les étapes de ce processus.

2. Mais nous pouvons toujours écrire  $r'(s, a) = \sum_{s' \in \mathcal{S}} T(s' | s, a) r(s, a, s')$ , et  $r(s, a, s')$  apparaît systématiquement sous cette forme  $r'$  dans les équations : dans la suite nous écrirons directement  $r(s, a)$  pour rendre les calculs plus clairs.

2.2.2. *Fonction de croyance.* Puisque, à l'instant  $t \in \mathbb{N}$ , l'agent n'a pas accès à son état  $s_t$ , l'espace de définition des politiques et de la fonction valeur ne pourra pas être  $\mathcal{S}$ , mais un espace d'états d'information noté  $I$ .

**Définition 1.** *Séquence d'états d'information d'un POMDP :*

*Suite de variables aléatoires  $(i_t)_{t \in \mathbb{N}} \in I^{\mathbb{N}}$ , telle que*

$$\forall t \in \mathbb{N}, \quad \mathbf{p}(i_{t+1} \mid i_0, i_1, \dots, i_t, o_{t+1}, a_t) = \mathbf{p}(i_{t+1} \mid i_t, o_{t+1}, a_t)$$

*i.e.  $(i_t)$  est une chaîne de Markov contrôlée.*

**Remarque :** L'état d'information complet au temps  $t$ ,  $i_t^C$ , est constitué de  $\mathbf{b}_0, (o_1, \dots, o_t)$  et  $(a_1, \dots, a_{t-1})$ . La séquence d'états d'information complets est trivialement une chaîne de Markov contrôlée.

**Définition 2.** *Une séquence d'états d'information d'un POMDP est un processus d'information suffisant si :*

- $i_t = \tau(i_{t-1}, o_t, a_{t-1})$ , i.e.  $i_t$  se calcule directement à partir de l'état d'information précédent, de l'action précédente, ainsi que de l'observation courante, grâce à une fonction  $\tau : I \times \mathcal{O} \times \mathcal{A} \rightarrow I$ ,
- $\mathbf{p}(s_t \mid i_t) = \mathbf{p}(s_t \mid i_t^C)$ , i.e. la connaissance de l'état d'information complet  $i_t^C$  n'apporte pas plus d'information sur les états courants que la connaissance de  $i_t$ .

Considérons un POMDP  $\langle \mathcal{S}, \mathcal{A}, \mathcal{O}, T, O, r, \mathbf{b}_0 \rangle$ , et une politique  $(a_t)_{t \in \mathbb{N}}$ . Supposons que nous disposons d'un processus d'information suffisant i.e. nous disposons de  $\tau$ , de  $i_0 \in I$ , et le processus  $(i_t)_{t \in \mathbb{N}}$  généré par les observations du POMDP et les actions  $a_t$ , vérifie la définition 2 (nous pouvons calculer  $\mathbf{p}(s \mid i)$ ).

Supposons que le POMDP étudié est à l'étape  $t \in \mathbb{N}$ , et que l'agent a effectué l'action  $a_t$ .  $\tau$  a permis de mettre à jour l'information  $i_n \forall n \in \{1, \dots, t\}$ .

Notons

$$T(s' \mid i_t, a_t) = \sum_{s \in \mathcal{S}} T(s' \mid s, a_t) \mathbf{p}(s \mid i_t) \quad (2.7)$$

la probabilité que l'action  $a_t$  mène l'agent à l'état  $s_{t+1} = s'$ .

Ce calcul nous permet de noter

$$O(o' \mid i_t, a_t) = \sum_{s' \in \mathcal{S}} O(o' \mid s', a_t) T(s' \mid i_t, a_t) \quad (2.8)$$

la probabilité que l'action  $a_t$  mène à l'observation de  $o'$ .

L'agent atteindra  $i'$  ssi l'observation  $o' = o_{t+1}$  reçue est telle que  $i' = \tau(i_t, a_t, o')$ . 2.8 nous amène au calcul de la probabilité qu'il atteigne  $i'$  :

$$\mathbb{P}(i_{t+1} = i' \mid i_t, a_t) = \sum_{o' \in \mathcal{O} \text{ tq } i' = \tau(i_t, a_t, o')} O(o' \mid i_t, a_t) \quad (2.9)$$

Nous pouvons transformer ce POMDP en un MDP sur  $I$  (les paramètres de ce MDP sont indexés par 2) :

- $\mathcal{S}_2 = I$ ,
- $\mathcal{A}_2 = \mathcal{A}$ ,
- $\forall i, i' \in I^2 = \mathcal{S}_2^2, \forall a \in \mathcal{A}_2, T_2(i' | i, a) = \mathbf{p}(i' | i, a)$  définie en 2.9,
- $\forall i \in I = \mathcal{S}_2, \forall a \in \mathcal{A}_2, r_2(i, a) = \sum_{s \in \mathcal{S}} r(s, a) \mathbf{p}(s | i)$ ,
- $s_{0,2} = i_0 = \mathbf{b}_0$ .

Soit  $\begin{cases} d: I \rightarrow \mathcal{A} \\ i_t \mapsto d(i_t) = a_t \end{cases}$  une politique stationnaire. La fonction valeur associée à cette politique est

$$V^d(i) = \mathbb{E} \left[ \sum_{t=0}^{+\infty} r_2(i_t, d(i_t)) \middle| i_0 = i \right]$$

Cette quantité s'interprète comme la somme actualisée des récompenses que l'agent croit obtenir en moyenne.

Notons que l'ensemble des observations  $\mathcal{O}$  est fini. D'après le premier point de la définition, il y a au maximum  $\text{Card } \mathcal{O}$  successeurs de  $i_t$  possibles, soit un nombre fini. Pour chaque  $(i, a) \in I \times \mathcal{A}$ , notons  $I_{i,a}$  l'ensemble fini des états de  $\mathcal{S}_2 = I$  atteignables par l'agent s'il est dans l'état  $i$  et choisit l'action  $a$ .  $I_{i,a} = \{i' \in I \mid \exists o' \in \mathcal{O} \text{ tel que } i' = \tau(i, a, o')\}$ .

Ce MDP vérifie les hypothèses de la définition donnée section 2.1.1, et l'hypothèse de support fini ( $\text{Card}(I_{i,a}) < +\infty$ ) de la probabilité de transition évoquée section 2.1.2.

Nous pouvons donc écrire l'équation de Bellman 2.2 établie page 6 qui caractérise la fonction valeur  $V^d : \forall i \in I$ ,

$$\begin{aligned} V^d(i) &= (\mathcal{B}^d V^d)(i) \\ &= \sum_{i' \in I_{i,d(i)}} T_2(i' | i, d(i)) (r_2(i, d(i)) + \gamma V^d(i')) \\ &= r_2(i, d(i)) + \gamma \sum_{i' \in I_{i,d(i)}} T_2(i' | i, d(i)) V^d(i') \\ &= \sum_{s \in \mathcal{S}} r_2(s, d(i)) \mathbf{p}(s | i) + \gamma \sum_{o' \in \mathcal{O}} O(o' | i, d(i)) V^d(\tau(i, o', d(i))) \end{aligned}$$

par 2.9. L'équation de Programmation Dynamique ( $V^* = \mathcal{B}^* V^*$ ) s'obtient en rajoutant  $\max_{a \in \mathcal{A}}$  au début, et en remplaçant tous les  $d(i)$  par  $a$  (voir 2.3). Elle caractérise

$$V^{opt}(i) = \max_{d \text{ politique}} \mathbb{E} \left[ \sum_{t=0}^{+\infty} \gamma^t r_2(i_t, d(i_t)) \middle| i_0 = i \right]$$

**Définition 3.** *Etat de croyance :*

L'état de croyance à l'instant  $t$  est  $\mathbf{b}_t(s) = \mathbb{P}(s_t = s \mid i_t^C)$ .

Nous souhaitons montrer que c'est un processus d'information suffisant.  $\forall t \in \mathbb{N}$ , le deuxième point de la définition 2 est évident :

$$\mathbb{P}(s_t | b_t) = b_t(s_t) := \mathbb{P}(s_t | i_t^C)$$

Le premier point de cette définition implique le caractère markovien du processus de croyance. En effet, si  $\mathbf{b}_{t+1} = \tau(\mathbf{b}_t, o_{t+1}, a_t)$ , alors  $\mathbf{b}_{t+1}$  ne dépend pas des états de croyance antérieurs à  $t$ . La formule 2.9 appliquée à  $\mathbf{b}$ , qui n'utilise que l'hypothèse d'avoir une transformation  $\tau$ , permet de bien visualiser que  $(\mathbf{b}_t)_{t \in \mathbb{N}}$  est un processus décisionnel de Markov.

Montrons maintenant le premier point de 2 :

$$\begin{aligned} \mathbf{b}_{t+1}(s') &:= \mathbb{P}(s_{t+1} = s' \mid i_{t+1}^C) \\ &= \mathbb{P}(s_{t+1} = s' \mid i_t^C, a_t = a, o_{t+1} = o') \\ &= \frac{\mathbb{P}(s_{t+1} = s', o_{t+1} = o' \mid i_t^C, a_t = a)}{\mathbb{P}(o_{t+1} = o' \mid i_t^C, a_t = a)} \\ &= \frac{\mathbb{P}(s_{t+1} = s' \mid i_t^C, a_t = a) \mathbb{P}(o_{t+1} = o' \mid s_{t+1} = s', i_t^C, a_t = a)}{\sum_{\bar{s} \in \mathcal{S}} \mathbb{P}(s_{t+1} = \bar{s}, o_{t+1} = o' \mid i_t^C, a_t = a)} \end{aligned}$$

On remarque que le dénominateur est la somme sur  $s' \in \mathcal{S}$  du numérateur. De même,

$$\begin{aligned} \mathbb{P}(s_{t+1} = s' \mid i_t^C, a_t = a) &= \sum_{s \in \mathcal{S}} \mathbb{P}(s_t = s \mid a_t, i_t^C) \mathbb{P}(s_{t+1} = s' \mid s_t = s, i_t^C, a_t = a) \\ &= \sum_{s \in \mathcal{S}} \mathbb{P}(s_t = s \mid i_t^C) \mathbb{P}(s_{t+1} = s' \mid s_t = s, a_t = a) \\ &= \sum_{s \in \mathcal{S}} \mathbf{b}_t(s) T(s' \mid s, a) \end{aligned}$$

car l'état  $s_t$  ne dépend pas de l'action au temps  $t$  (mais de  $a_{t-1}$  !), et  $s_{t+1}$  ne dépend que de  $s_t$  et  $a_t$ , et pas de l'historique complet  $i_t^C$  (c'est un processus de Markov).

De plus, comme l'observation  $o_{t+1}$  ne dépend pas de l'historique complet à l'instant  $t$ , mais seulement de  $s_{t+1}$ , et  $a_t$  :

$$\mathbb{P}(o_{t+1} = o' \mid s_{t+1} = s', i_t^C, a_t = a) = O(o' \mid s', a)$$

Ainsi,

$$\begin{aligned} \mathbf{b}_{t+1}(s') &= \frac{O(o' \mid s', a) \sum_{s \in \mathcal{S}} \mathbf{b}_t(s) T(s' \mid s, a)}{\sum_{s, \bar{s} \in \mathcal{S}^2} \mathbf{b}_t(s) T(\bar{s} \mid s, a) O(o' \mid \bar{s}, a)} \\ &= \tau(\mathbf{b}_t, o', a) \end{aligned} \tag{2.10}$$

L'avantage de  $(\mathbf{b}_t)_{t \in \mathbb{N}}$  sur  $(i_t^C)_{t \in \mathbb{N}}$  est que sa taille n'augmente pas au cours du temps : cela reste un élément de  $\Delta\mathcal{S}$ .

2.2.3. *Fonctions valeurs PWLC et résolution d'un POMDP.* Nous raisonnons donc sur un espace continu  $\Delta\mathcal{S}$ , c'est pourquoi résoudre un POMDP a longtemps été très compliqué.

En posant, tout comme 2.9 et 2.8,

$$T(s' | \mathbf{b}, a) = \sum_s T(s' | s, a) \mathbf{b}(s)$$

et

$$O(o | \mathbf{b}, a) = \sum_{s'} O(o | s', a) T(s' | \mathbf{b}, a)$$

l'équation de Bellman du modèle POMDP est

$$V(\mathbf{b}) = \sum_{s \in \mathcal{S}} r(s, a) \mathbf{b}(s) + \gamma \sum_{o \in \mathcal{O}} O(o | \mathbf{b}, a) V(\tau(\mathbf{b}, o, a))$$

Définissons  $\mathcal{B}^*$  pour l'état d'information  $\mathbf{b}$  :

$$\mathcal{B}^*V(\mathbf{b}) = \max_{a \in \mathcal{A}} \sum_{s \in \mathcal{S}} r(s, a) \mathbf{b}(s) + \gamma \sum_{o \in \mathcal{O}} O(o | \mathbf{b}, a) V(\tau(\mathbf{b}, o, a))$$

Le résultat suivant, dû à R D. Smallwood et E J. Sondik [11], est à l'origine de nombreux algorithmes de résolution des POMDPs :

**Théorème 1.** *les fonctions linéaires par morceaux et convexes sont stables par  $\mathcal{B}^*$ .*

*Démonstration.* En effet, soit  $V : \Delta\mathcal{S} \rightarrow \mathbb{R}$  linéaire par morceau et convexe.  $\exists \Gamma \subset \mathbb{R}^{\mathcal{S}}$ ,  $\text{Card } \Gamma < +\infty$ , tel que

$$V(\mathbf{b}) = \max_{\alpha \in \Gamma} \sum_{s \in \mathcal{S}} \mathbf{b}(s) \alpha(s) = \max_{\alpha \in \Gamma} \mathbf{b} \cdot \alpha$$

.

Pour  $\mathbf{b} \in \Delta\mathcal{S}$ , l'équation de Programmation Dynamique s'écrit

$$\begin{aligned} \mathcal{B}^*V(\mathbf{b}) &= \max_{a \in \mathcal{A}} \sum_{s \in \mathcal{S}} r(s, a) \mathbf{b}(s) + \gamma \sum_{o \in \mathcal{O}} O(o | \mathbf{b}, a) V(\tau(\mathbf{b}, o, a)) \\ &= \max_{a \in \mathcal{A}} \sum_{s \in \mathcal{S}} r(s, a) \mathbf{b}(s) + \gamma \sum_{o \in \mathcal{O}} O(o | \mathbf{b}, a) \max_{\alpha \in \Gamma} \sum_{s' \in \mathcal{S}} \tau(\mathbf{b}, a, o)(s') \alpha(s') \\ &= \max_{a \in \mathcal{A}} \sum_{s \in \mathcal{S}} r(s, a) \mathbf{b}(s) + \max_{(\alpha_o)_{o \in \mathcal{O}} \in \Gamma^{\mathcal{O}}} \gamma \sum_{o \in \mathcal{O}} O(o | \mathbf{b}, a) \sum_{s' \in \mathcal{S}} \tau(\mathbf{b}, a, o)(s') \alpha_o(s') \end{aligned}$$

où  $(\alpha_o)_{o \in \mathcal{O}} \in \Gamma^{\mathcal{O}}$  est une séquence de vecteurs  $\alpha \in \Gamma$  indexé par  $o \in \mathcal{O}$ .

Rappelons la formule de mise à jour de la croyance avec les notations précédentes :  $\forall s' \in \mathcal{S}$ ,

$$\tau(\mathbf{b}, a, o)(s') = \frac{O(o | s', a) T(s' | \mathbf{b}, a)}{O(o | \mathbf{b}, a)}$$

Ainsi,  $\mathcal{B}^*V(\mathbf{b})$

$$\begin{aligned}
 &= \max_{a \in \mathcal{A}} \max_{(\alpha_o) \in \Gamma^{\mathcal{O}}} \sum_{s \in \mathcal{S}} r(s, a) \mathbf{b}(s) + \gamma \sum_{o \in \mathcal{O}} O(o | \mathbf{b}, a) \sum_{s' \in \mathcal{S}} \frac{O(o | s', a) T(s' | \mathbf{b}, a)}{O(o | \mathbf{b}, a)} \alpha_o(s') \\
 &= \max_{a \in \mathcal{A}, (\alpha_o) \in \Gamma^{\mathcal{O}}} \sum_{s \in \mathcal{S}} r(s, a) \mathbf{b}(s) + \gamma \sum_{o \in \mathcal{O}} \sum_{s' \in \mathcal{S}} O(o | s', a) \sum_{s \in \mathcal{S}} T(s' | s, a) \mathbf{b}(s) \alpha_o(s') \\
 &= \max_{a \in \mathcal{A}, (\alpha_o) \in \Gamma^{\mathcal{O}}} \sum_{s \in \mathcal{S}} \left( r(s, a) + \gamma \sum_{o \in \mathcal{O}} \sum_{s' \in \mathcal{S}} O(o | s', a) T(s' | s, a) \alpha_o(s') \right) \mathbf{b}(s) \\
 &= \max_{\alpha' \in \Gamma'} \sum_{s \in \mathcal{S}} \alpha'(s) \mathbf{b}(s)
 \end{aligned}$$

où  $\Gamma' = \left\{ \begin{array}{l} \alpha' \in \mathbb{R}^{\mathcal{S}} | \forall s \in \mathcal{S}, \alpha'(s) = r(s, a) + \gamma \sum_{o \in \mathcal{O}} \sum_{s' \in \mathcal{S}} O(o | s', a) T(s' | s, a) \alpha_o(s') \\ \text{et } a \in \mathcal{A}, (\alpha_o)_{o \in \mathcal{O}} \in \Gamma^{\mathcal{O}} \end{array} \right\}$   
 qui est fini de cardinal  $\text{Card } \mathcal{A} (\text{Card } \Gamma)^{\text{Card } \mathcal{O}}$ .

$\mathbf{b} \mapsto \mathcal{B}^*V(\mathbf{b})$  est donc bien linéaire par morceaux et convexe (PWLC-Piecewise Linear and Convex), et l'on appelle les éléments de  $\Gamma'$ , les  $\alpha$ -vecteurs.  $\square$

Il est donc enfin possible de trouver une politique optimale pour ce problème avec la même stratégie que  $IV_{MDP}$  1 : partons d'une fonction linéaire par morceaux et convexe  $V^0(\mathbf{b}) = \max_{\alpha \in \Gamma^0} \sum_{s \in \mathcal{S}} \alpha(s) \mathbf{b}(s)$ .

Pour coller avec le cas de l'horizon fini, nous pouvons prendre les vecteurs récompense :  $\Gamma = \{ \alpha \in \mathbb{R}^{\mathcal{S}} | \exists a \in \mathcal{A} \text{ telle que } \forall s \in \mathcal{S}, \alpha(s) = r(s, a) \}$ .

Nous avons vu dans la section 2.1.1 que l'itération de l'opérateur de Programmation Dynamique  $\mathcal{B}^*$  faisait tendre n'importe quelle fonction valeur  $V^0 \in \mathcal{F}_B(\mathbb{R}, \mathcal{S})$  vers la fonction valeur optimale  $V^{opt}$ . Cela fonctionne ici aussi pour  $\mathcal{S}_2 = \Delta\mathcal{S}$ , et nous pouvons maintenant en tirer un algorithme 2 puisque nous savons que les fonctions valeurs calculées resteront PWLC si  $V^0$  l'est : nous allons manipuler un nombre fini d'  $\alpha$ -vecteurs  $\Gamma^n$  à chaque itération  $n \in \{1, \dots, N\}$ .

Nous rappelons qu'une politique gloutonne  $d^n$  pour  $n$  itérations, se calcule en identifiant pour chaque  $\mathbf{b} \in \Delta\mathcal{S}$  l'action maximisante dans l'opérateur  $\mathcal{B}^*$ . Supposons que  $V^0$  ait subi  $n$  itérations de  $\mathcal{B}^*$ , et que nous ayons calculé  $\Gamma^n$ . Soit  $\alpha_a$  un  $\alpha$ -vecteur de  $\Gamma^n$  construit à partir de l'action  $a$  et d'une séquence de vecteurs  $\alpha_o \in (\Gamma^{n-1})^{\mathcal{O}}$ . Il suffit de déterminer la région de  $\Delta\mathcal{S}$  où ce vecteur  $\alpha_a$  maximise la somme  $\sum_{s \in \mathcal{S}} \alpha(s) \mathbf{b}(s)$ . Quelle que soit la croyance  $\mathbf{b}$  de cette région,  $d^n(\mathbf{b}) = a$ .

Plus simplement, si l'agent est dans l'état de croyance  $\mathbf{b}$ , il suffit d'évaluer  $\sum_{s \in \mathcal{S}} \alpha(s) \mathbf{b}(s)$  pour chaque  $\alpha$ -vecteur de  $\Gamma^n$ , et de retourner l'action associée à l'  $\alpha$ -vecteur qui maximise cette somme. L'algorithme 2 présente cette manière de résoudre un POMDP. Elle est très naïve car le nombre d'  $\alpha$ -vecteurs augmente très rapidement avec les itérations : si  $g^n = \text{Card } \Gamma^n$ ,



$g^n = \text{Card } \mathcal{A} \times (g^{n-1})^{\text{Card}(\mathcal{O})}$ . De nombreuses techniques peuvent être utilisées pour enlever à chaque itération  $n$ , les vecteurs  $\alpha_{bad} \in \Gamma^n$  tels que  $\max_{\alpha} \sum_{s \in \mathcal{S}} \alpha(s) \mathbf{b}(s) > \sum_{s \in \mathcal{S}} \alpha_{bad}(s) \mathbf{b}(s) \forall \mathbf{b} \in \Delta$ .<sup>3</sup> L'algorithme Witness de Cassandra utilise la programmation linéaire pour élaguer les mauvais  $\alpha$ -vecteurs.

---

**Algorithm 2** POMDP resolution

---

```

 $\Gamma \leftarrow \Gamma^0$ 
 $i \leftarrow 1$ 
while  $i \leq N$  do
  for  $a \in \mathcal{A}, (\alpha_o) \in \Gamma^{\mathcal{O}}$  do
    for  $i \in \mathcal{S}$  do
       $\alpha(s) \leftarrow r(s, a) + \gamma \sum_{o \in \mathcal{O}} \sum_{s' \in \mathcal{S}} O(o | s', a) T(s' | s, a) \alpha_o(s')$ 
    end for
     $\Gamma' \leftarrow \{\Gamma', \alpha\}$ 
  end for
   $\Gamma \leftarrow \Gamma'$ 
   $i++$ 
end while
return  $\Gamma$ 

```

---

### 3. ALTERNATIVES À LA MODÉLISATION DE L'INCERTITUDE

belief subjectiviste (de finetti ?) + caroline : l'agent cherche-t-il à mieux observer ? Maximiser l'espérance des gains vraiment perçus + Buffet récompense sur les beliefs + incertitudes liées à l'observation. Problématique

**3.1. Théorie des Possibilités.** Considérons un ensemble fini d'évènements élémentaires  $\omega \in \Omega$ . Une mesure de possibilité  $\Pi$  sur  $\Omega$ , est une fonction de l'ensemble des parties de  $\Omega$ ,  $\mathcal{P}(\Omega)$ , vers le segment réel  $[0, 1]$ .

**Définition 4.** *Mesure de Possibilité*

$\Pi : 2^\Omega \rightarrow [0, 1]$  telle que

- la possibilité de l'ensemble vide est nulle :  $\Pi(\emptyset) = 0$ ,
- la possibilité de  $\Omega$  est maximale :  $\Pi(\Omega) = 1$ ,
- la possibilité de l'union de deux parties disjointes de  $\Omega$  est le maximum des possibilités de chacune des deux parties :  
 $\forall A, B \subset \Omega$  telles que  $A \cap B = \emptyset$ ,

$$\Pi(A \cup B) = \max \{ \Pi(A), \Pi(B) \}$$

Comme  $\Omega$  est fini, nous déduisons du troisième point de la définition que  $\forall A \subset \Omega$ ,  $\Pi(A) = \max_{\omega \in A} \pi(\omega)$ . Ainsi, on peut montrer facilement que  $\Pi(A \cup B) = \max \{ \Pi(A), \Pi(B) \}$  même lorsque  $A \cap B \neq \emptyset$ .  $\Pi$  n'est donc pas une mesure dans le sens classique (sinon elle se devrait d'être additive), mais

---

3. voir <http://researchers.lille.inria.fr/~munos/papers/files/bouquinPDMIA.pdf>

une *mesure floue*<sup>4</sup>.

Comme nous raisonnons sur un ensemble fini, attribuer une possibilité à chaque singleton  $\{\omega\}$  permet de définir entièrement la mesure de possibilité. Ainsi, comme  $\Pi(\Omega) = 1 = \max_{\omega \in \Omega} \pi(\omega)$ , il existe forcément un élément  $\omega_0 \in \Omega$  tel que  $\pi(\omega_0) = 1$ . Le deuxième point de la définition est donc une normalisation comparable à  $\sum_{\omega \in \Omega} \mathbf{p}(\omega) = 1$  en théorie des probabilités. Cette normalisation signifie qu'il existe toujours un évènement complètement possible.

Une mesure duale nommée mesure de *nécessité* peut-être définie à partir de la mesure de possibilité : la nécessité d'un ensemble est d'autant plus grande que la possibilité de l'ensemble complémentaire est faible.

**Définition 5.** *Mesure de Nécessité associée à  $\Pi$*

La mesure de nécessité  $\mathcal{N} : 2^\Omega \rightarrow [0, 1]$  associée à  $\Pi$  est telle que  $\forall A \subset \Omega$ ,

$$\mathcal{N}(A) = 1 - \Pi(A^c)$$

C'est aussi une mesure floue :  $\mathcal{N}(\emptyset) = 1 - \Pi(\Omega) = 0$ ,  $\mathcal{N}(\Omega) = 1 - \Pi(\emptyset) = 1$ , et si  $A \subset B$ ,  $\mathcal{N}(A) = 1 - \Pi(A^c) \leq 1 - \Pi(B^c) = \mathcal{N}(B)$ .

Considérons maintenant  $A, B \subset \Omega$ . Nous avons le résultat suivant

$$\begin{aligned} \mathcal{N}(A \cap B) &= 1 - \Pi((A \cap B)^c) \\ &= 1 - \Pi(A^c \cup B^c) \\ &= 1 - \max\{\Pi(A^c), \Pi(B^c)\} \\ &= \min\{1 - \Pi(A^c), 1 - \Pi(B^c)\} \\ &= \min\{\mathcal{N}(A), \mathcal{N}(B)\} \end{aligned}$$

Les mesures de possibilité permettent la modélisation de l'incertitude liée à la méconnaissance d'un évènement  $\subset \Omega$  de manière très intuitive, tandis que les mesures de probabilité modélisent une incertitude liée à la variabilité d'un évènement.

Par exemple, la situation d'ignorance totale peut-être modélisée par l'attribution à chaque évènement élémentaire  $\omega \in \Omega$  d'une possibilité valant 1 :  $\forall \omega, \pi(\omega) = 1$ . Cette distribution de possibilité s'interprète comme le fait que tout évènement est complètement possible. De même,  $\forall A \subset \Omega$ ,  $A \neq \Omega$ ,  $\mathcal{N}(A) = 1 - \Pi(A^c) = 1 - \max_{\omega \in A^c} \pi(\omega) = 0$  donc, outre l'ensemble total  $\Omega$ , aucun des évènements n'est nécessaire. C'est donc bien une situation d'ignorance totale. Au contraire, attribuer une possibilité de 1 à  $\omega_0 \in \Omega$ , et une possibilité nulle à tous les autres éléments de l'ensemble, modélise une connaissance parfaite d'une situation :  $\eta(\omega_0) = 1 - \Pi(\{\omega_0\}^c) = 1$  : l'évènement  $\omega_0$  est nécessaire, tous les autres sont impossibles.

---

4. Une mesure floue est définie comme une mesure  $\mu$  vérifiant les deux premiers points de la définition 4 ainsi que la condition  $\forall A \subset B \subset \Omega$ ,  $\mu(A) \leq \mu(B)$

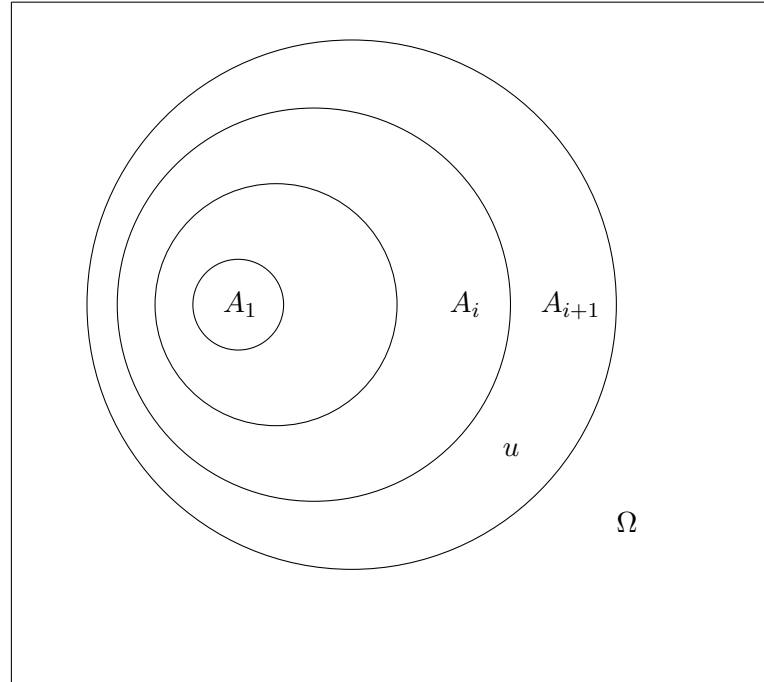
Ces exemples nous amènent à évaluer l'information contenue dans une distribution de possibilité par la notion de *spécificité* [référence au bibtex de Possibility theory and statistical reasoning d.dubois et yager [115]] : une mesure de possibilité  $\pi_1$  est au moins aussi spécifique (*i.e.* informative) que  $\pi_2$  ssi  $\forall \omega \in \Omega, \pi_1(\omega) \leq \pi_2(\omega)$ .

Nous pouvons aussi remarquer que si un évènement  $A \subset \Omega$  n'est pas complètement possible ( $\Pi(A) < 1$ ), alors il n'est pas du tout nécessaire ( $\mathcal{N}(A) = 0$ ). De même, si cet évènement est de nécessité non nulle, alors il est complètement possible. En effet, dans le premier cas, l'évènement élémentaire de possibilité 1 n'appartient pas à  $A$ , donc appartient à  $A^c$  :  $\mathcal{N}(A) = 1 - \Pi(A^c) = 0$ . Dans le deuxième cas,  $\Pi(A^c) = 1 - \mathcal{N}(A) < 1$ . Ainsi, l'évènement élémentaire de possibilité 1 appartient à  $A$  :  $\Pi(A) = 1$ .

Afin d'avoir une image plus claire de l'outil possibiliste, une distribution de possibilité peut être représentée par une famille de sous-ensembles imbriqués  $\{A_1, A_2, \dots, A_n\}$ ,  $A_1 \subset \dots \subset A_i \subset A_{i+1} \subset \dots \subset A_n$ . À chacun de ces sous-ensembles  $A_i$ , nous associons un niveau de confiance  $\lambda_i$ , qui sera le degré de nécessité de l'ensemble *i.e.*  $\mathcal{N}(A_i)$ . La séquence  $(\lambda_i)_{i=1}^n$  est donc croissante :  $\forall i \in \{1, \dots, n-1\}, \lambda_i = \mathcal{N}(A_i) \leq \mathcal{N}(A_{i+1}) = \lambda_{i+1}$ .

À ce stade, plusieurs distributions de possibilités peuvent être associées à cette configuration. Nous choisissons la moins spécifique, *i.e.* la distribution de possibilité la plus grande.

Ci-contre, un schéma de la situation.



Par définition,

$$\begin{aligned}\lambda_i &= \mathcal{N}(A_i) \\ &= 1 - \Pi(A_i^c) \\ &= 1 - \max_{\omega \notin A_i} \pi(\omega)\end{aligned}$$

Soit  $\omega \notin A_i$ . Nous avons alors  $\pi(\omega) \leq 1 - \lambda_i$ . Or, si  $\omega \notin A_i$ , alors  $\omega \notin A_{i-1}, \dots, A_1$ , et donc

$$\pi(\omega) \leq 1 - \lambda_j \quad \forall j \in \{1, \dots, i\}$$

$$\pi(\omega) = \begin{cases} 1 & \text{si } \omega \in A_1 \\ \min_{i|\omega \notin A_i} 1 - \lambda_i & \text{sinon} \end{cases}$$

est alors un degré de possibilité qui convient, et c'est le plus grand possible.

Inversement, la donnée d'une distribution de possibilité permet de retrouver des sous-ensembles imbriqués ainsi que leurs niveaux de confiance (nécessité) : soient  $\{\alpha_1 = 1 \geq \alpha_2 \geq \dots \geq \alpha_n\}$  l'ensemble des valeurs de la distribution de possibilité  $\{x \in [0, 1] \mid \exists \omega \in \Omega \text{ tel que } \pi(\omega) = x\}$  classées par ordre décroissant. Posons, avec  $\alpha_{n+1} = 0$ ,

$$A_i = \{\omega \mid \pi(\omega) \geq \alpha_i\}$$

qui sont des sous-ensembles imbriqués, et

$$\begin{aligned}\lambda_i &= 1 - \max_{\omega \notin A_i} \pi(\omega) \\ &= 1 - \alpha_{i+1}\end{aligned}$$

Cette vision donne une idée [trouver références] de simulation (probabiliste) d'évènements dont l'incertitude est modélisée en terme de possibilités :

- Simuler  $\alpha$  de loi uniforme sur  $[0, 1]$ ,
- Tirer uniformément  $\omega$  dans  $\{\omega \mid \pi(\omega) \geq \alpha\}$

Ainsi, la probabilité que l'évènement  $\omega$  se produise est

$$\begin{aligned}\mathbf{p}(\omega) &= \int_0^1 \mathbf{p}(\omega, \alpha) d\alpha \\ &= \int_0^1 \mathbf{p}(\omega \mid \alpha) d\alpha \\ &= \int_0^1 \frac{\mathbb{1}_{(\pi(\omega) \geq \alpha)}}{\text{Card}\{\omega' \in \Omega \mid \pi(\omega') \geq \alpha\}} d\alpha \\ &= \sum_{i=1}^n \int_{\alpha_{i+1}}^{\alpha_i} \frac{\mathbb{1}_{(\pi(\omega) \geq \alpha_i)}}{\text{Card}\{\omega' \in \Omega \mid \pi(\omega') \geq \alpha_i\}} d\alpha \\ &= \sum_{i=i_0 \mid \pi(\omega) = \alpha_{i_0}}^n \frac{\alpha_i - \alpha_{i+1}}{\text{Card } A_i}\end{aligned} \tag{3.1}$$

Nous retrouverons cette idée avec la transformation réversible  $\Pi \leftrightarrow \mathbb{P}$  dans la section qui suit.

3.1.1. *Consistance, Préservation de la préférence et Transformations.* Il est essentiel de bien saisir que probabilités et possibilités ne modélisent pas la même facette de l'incertitude : une distribution de probabilité pourra bien exprimer l'aléa, voire l'indécision (probabilité uniforme), et la théorie des possibilités est idéale pour représenter l'ignorance. Cela nous amène à affirmer qu'une distribution de possibilité contient moins d'information qu'une distribution de probabilité.

Cependant, ces mesures restent des évaluations de l'incertitude, qui est un mélange des différentes facettes dont nous avons parlées. Nous remarquons aussi qu'elles coïncident lors de la connaissance totale (1 pour l'évènement certain, 0 pour les autres). Ces quelques éléments nous laissent penser que les deux mesures sont corrélées lorsqu'elles évaluent le même phénomène. Sous cet angle de vue, nous souhaitons définir des *transformations* entre possibilités et probabilités qui prennent en compte ces dites corrélations. Des principes à respecter permettent aux transformations  $\mathbb{P} \leftrightarrow \Pi$  de garder un sens.

Formellement, une mesure de possibilité  $\Pi$  sur  $\Omega$  est équivalente à la famille de mesures de probabilité  $\mathcal{P}(\Pi) = \{\mathbb{P} \in \Delta\Omega \mid \forall A \subset \Omega, \mathbb{P}(A) \leq \Pi(A)\}$  comme expliqué dans [6]. Il paraît alors naturel de chercher des transformations respectant le principe de consistance.

**Définition 6** (Principe de consistance).

$$\forall A \subset \Omega, \quad \Pi(A) \geq \mathbb{P}(A)$$

Notons que

$$\begin{aligned} &\Rightarrow \forall A \subset \Omega, \quad 1 - \mathcal{N}(A^c) \geq 1 - \mathbb{P}(A^c) \\ &\Rightarrow \forall A \subset \Omega, \quad \mathcal{N}(A^c) \leq \mathbb{P}(A^c) \end{aligned}$$

et donc  $\forall A \subset \Omega, \mathcal{N}(A) \leq \mathbb{P}(A)$ .

Le principe suivant paraît encore plus évident : un évènement est plus probable qu'un autre si et seulement si il est plus possible qu'un autre. L'ordre de préférence induit par une distribution de possibilité est en fait la signification principale de cette distribution.

**Définition 7** (Principe de la conservation de préférence).

$$\forall \omega_1, \omega_2 \in \Omega^2, \quad \mathbf{p}(\omega_1) > \mathbf{p}(\omega_2) \Leftrightarrow \pi(\omega_1) > \pi(\omega_2)$$

Ces deux principes nous guideront dans la construction de la transformation que nous recherchons.

**Une transformation naïve** peut facilement nous venir à l'esprit : nous rappelons que la normalisation utilisée en théorie des possibilités est

$\max_{\omega \in \Omega} \pi(\omega) = 1$ . Compte tenu des normalisations dans les deux théories, une transformation pourrait être :

$$\begin{aligned} \forall \omega \in \Omega, \quad \pi(\omega) &= \frac{\mathbf{p}(\omega)}{\max_{\omega' \in \Omega} \mathbf{p}(\omega')} \\ \mathbf{p}(\omega) &= \frac{\pi(\omega)}{\sum_{\omega' \in \Omega} \pi(\omega')} \end{aligned}$$

Il est évident que cette transformation respecte le principe de la conservation de préférence 7. Cependant, le contre-exemple suivant nous assure que le principe de consistance 6 n'est pas respecté en général : considérons 3 évènements, ( $\text{Card } \Omega = 3$ ) et la distribution de probabilité  $\mathbf{p}(\omega_1) = 0.9$ ,  $\mathbf{p}(\omega_2) = \mathbf{p}(\omega_3) = 0.05$ .

Cette probabilité devient, par la transformation naïve, la distribution de possibilité  $\pi(\omega_1) = 1, \pi(\omega_2) = \pi(\omega_3) = \frac{1}{18}$ .

Ainsi,

$$\Pi(\{\omega_2, \omega_3\}) = \max(\pi(\omega_2), \pi(\omega_3)) = \frac{1}{18} < 0.1 = \mathbb{P}(\{\omega_1, \omega_2\})$$

donc cette transformation ne respecte pas le principe de consistance.

Reprenons le calcul de 3.1 qui est une transformation  $\pi \rightarrow \mathbf{p}$ . Cette **transformation réversible** nous satisfera : elle est proposée dans [5] et respecte les principes de consistance 6 et de conservation de la préférence 7.

Supposons que  $\Omega = \{\omega_1, \dots, \omega_m\}$  avec  $\pi(\omega_1) \geq \pi(\omega_2) \geq \dots \geq \pi(\omega_m)$ . Nous rappelons que  $\alpha_1, \dots, \alpha_n$  sont les différentes valeurs présentes dans la distribution de possibilité, triées par ordre décroissant. Nous rappelons aussi que  $A_i = \{\omega \in \Omega \mid \pi(\omega) \geq \alpha_i\}$ . Ainsi, pour  $k \in \{1, \dots, m\}$

$$\begin{aligned} \mathbf{p}(\omega_k) &= \sum_{i=i_0 \mid \pi(\omega_k) = \alpha_{i_0}}^n \frac{\alpha_i - \alpha_{i+1}}{\text{Card } A_i} \\ &= \sum_{i=k}^m \frac{\pi(\omega_i) - \pi(\omega_{i+1})}{i} \end{aligned}$$

En effet,  $\frac{\pi(\omega_i) - \pi(\omega_{i+1})}{i}$  est nul lorsque  $\omega_i$  et  $\omega_{i+1}$  ont la même possibilité. Les seuls termes de la dernière somme qui sont non nuls sont ceux tels que  $\pi(\omega_i) - \pi(\omega_{i+1}) > 0$ , soit ceux du type  $\frac{\pi(\omega_{\text{Card}(A_i)}) - \pi(\omega_{\text{Card}(A_i)+1})}{\text{Card}(A_i)}$  pour  $i \in \{1, \dots, n\}$ , car  $\omega_{\text{Card}(A_i)} \in A_i$  et  $\omega_{\text{Card}(A_i)+1} \notin A_i$ .

Explicitons la transformation proposée : Nous notons  $\Omega = \{\omega_1, \dots, \omega_m\}$ , avec  $\mathbf{p}(\omega_1) \geq \mathbf{p}(\omega_2) \geq \dots \geq \mathbf{p}(\omega_m)$ .

Nous pouvons alors écrire la transformation  $\mathbf{p} \rightarrow \pi$  comme suit :

$$\forall i \in \{1, \dots, m-1\} \quad \pi(\omega_i) = i \cdot \mathbf{p}(\omega_i) + \sum_{j=i+1}^m \mathbf{p}(\omega_j) \quad (3.2)$$

avec  $\pi(\omega_m) = m \cdot \mathbf{p}(\omega_m)$

De même si les  $\omega \in \Omega$  sont tels que  $\pi(\omega_1) \geq \pi(\omega_2) \geq \dots \geq \pi(\omega_m)$ , la transformation réciproque  $\pi \rightarrow \mathbf{p}$ , équivalente à 3.1, est alors :

$$\forall i \in \{1, \dots, m\} \quad \mathbf{p}(\omega_i) = \sum_{j=i}^m \frac{\pi(\omega_j) - \pi(\omega_{j+1})}{j} \quad (3.3)$$

avec la notation  $\pi(\omega_{m+1}) = 0$

Vérifions que les deux formules sont équivalentes, puis montrons que les principes exposés précédemment sont bien vérifiés.

*Démonstration.* Montrons que la première formule implique la seconde. Tout d'abord, nous avons  $\mathbf{p}(\omega_m) = \frac{\pi(\omega_m)}{m}$  par la seconde ligne de 3.2, et ainsi

$$\begin{aligned} \pi(\omega_{m-1}) &= (m-1)\mathbf{p}(\omega_{m-1}) + \mathbf{p}(\omega_m) \\ &\Downarrow \\ \mathbf{p}(\omega_{m-1}) &= \frac{\pi(\omega_{m-1})}{m-1} - \frac{\pi(\omega_m)}{m(m-1)} \\ &= \frac{\pi(\omega_{m-1}) - \pi(\omega_m)}{m-1} + \frac{\pi(\omega_m)}{m} \end{aligned}$$

Maintenant, supposons que pour  $i_0 \in \{2, \dots, m-1\}$  3.3 est vrai et montrons l'égalité au rang  $i_0 - 1$  :

$$\begin{aligned} \pi(\omega_{i_0-1}) - \pi(\omega_{i_0}) &= (i_0-1) \cdot \mathbf{p}(\omega_{i_0-1}) + \sum_{j=i_0-1}^m \mathbf{p}(\omega_j) - i_0 \cdot \mathbf{p}(\omega_{i_0}) - \sum_{j=i_0}^m \mathbf{p}(\omega_j) \\ &= (i_0-1) \cdot \mathbf{p}(\omega_{i_0-1}) - i_0 \cdot \mathbf{p}(\omega_{i_0}) + \mathbf{p}(\omega_{i_0-1}) \\ &= (i_0-1)(\mathbf{p}(\omega_{i_0-1}) - \mathbf{p}(\omega_{i_0})) \\ \Rightarrow \mathbf{p}(\omega_{i_0-1}) &= \mathbf{p}(\omega_{i_0}) + \frac{\pi(\omega_{i_0-1}) - \pi(\omega_{i_0})}{i_0-1} \\ &= \sum_{j=i_0-1}^m \frac{\pi(\omega_j) - \pi(\omega_{j+1})}{j} \end{aligned} \quad (3.4)$$

Montrons maintenant l'implication réciproque : la seconde formule pour  $i = m$  nous donne directement la seconde ligne de 3.2. Puis,

$$\begin{aligned}
 \mathbf{p}(\omega_{m-1}) &= \sum_{j=m-1}^m \frac{\pi(\omega_j) - \pi(\omega_{j-1})}{j} \\
 &= \frac{\pi(\omega_{m-1}) - \pi(\omega_m)}{m-1} + \mathbf{p}(\omega_m) \\
 &\Downarrow \\
 \pi(\omega_{m-1}) &= \pi(\omega_m) + (m-1)(\mathbf{p}(\omega_{m-1}) - \mathbf{p}(\omega_m)) \\
 &= m\mathbf{p}(\omega_m) + (m-1)(\mathbf{p}(\omega_{m-1}) - \mathbf{p}(\omega_m)) \\
 &= (m-1)\mathbf{p}(\omega_{m-1}) - \mathbf{p}(\omega_m)
 \end{aligned}$$

Maintenant, supposons que la première formule (3.2) est vraie pour  $i_0 \in \{2, \dots, m-1\}$ , et montrons qu'elle est vraie au rang  $i_0 - 1$ .

$$\begin{aligned}
 \mathbf{p}(\omega_{i_0-1}) - \mathbf{p}(\omega_{i_0}) &= \frac{\pi(\omega_{i_0-1}) - \pi(\omega_{i_0})}{i_0 - 1} \\
 \Rightarrow \pi(\omega_{i_0-1}) &= \pi(\omega_{i_0}) + (i_0 - 1)(\mathbf{p}(\omega_{i_0-1}) - \mathbf{p}(\omega_{i_0})) \\
 &= i_0 \cdot \mathbf{p}(\omega_{i_0}) + \sum_{j=i_0+1}^m \mathbf{p}(\omega_j) + (i_0 - 1)(\mathbf{p}(\omega_{i_0-1}) - \mathbf{p}(\omega_{i_0})) \\
 &= \sum_{j=i_0}^m \mathbf{p}(\omega_j) + (i_0 - 1)\mathbf{p}(\omega_{i_0-1})
 \end{aligned}$$

Les deux formules sont donc équivalentes.

Montrons que le principe de consistance 6 est respecté :

Soit  $A \in \Omega$ , et  $j_0 = \min \{j \mid \omega_j \in A\}$ .

$$\begin{aligned}
 \Pi(A) &= \max_{\omega \in A} (\pi(\omega)) \\
 &= j_0 \cdot \mathbf{p}(\omega_{j_0}) + \sum_{j=j_0+1}^m \mathbf{p}(\omega_j) \geq \sum_{j=j_0}^m \mathbf{p}(\omega_j) \\
 &\geq \sum_{\omega \in A} \mathbf{p}(\omega) = \mathbb{P}(A)
 \end{aligned}$$

par définition de  $j_0$ . La consistance est donc assurée.

Le principe de conservation de la préférence 7 se voit bien avec la deuxième formule :

$$\begin{aligned}
 \mathbf{p}(x_i) - \mathbf{p}(x_{i+1}) &= \sum_{k=i}^i \frac{\pi(\omega_j) - \pi(\omega_{j+1})}{j} \\
 &= \frac{\pi(\omega_j) - \pi(\omega_{j+1})}{j}
 \end{aligned}$$



$$\mathbf{p}(x_i) - \mathbf{p}(x_{i+1}) > 0 \Leftrightarrow \pi(\omega_j) - \pi(\omega_{j+1}) > 0. \quad \square$$

Bien d'autres transformations existent [7], cependant, pour les travaux que nous allons présenter, cette transformation réversible est suffisante.

**3.1.2. Critères qualitatifs de décision.** Puisque ajouter deux possibilités n'a pas de sens, nous en déduisons que l'information principale contenue dans une mesure de possibilité est qualitative : la valeur de la possibilité (son aspect quantitatif) avait un sens, par exemple, dans le principe de consistance, ou bien dans l'exemple de simulation présenté en fin de section 3.1 : changer les valeurs de la possibilité (en conservant l'ordre) avait une conséquence directe sur son interprétation probabiliste.

Le travail que nous présenterons dans la prochaine section n'utilise que le sens qualitatif de l'outil possibiliste (qui est sa première caractéristique). Nous raisonnerons alors sur une échelle  $L$  (un ensemble totalement ordonné) : le plus grand élément est noté  $1_L$ , le plus petit  $0_L$ . Considérons les conséquences  $x \in \mathcal{X}$  d'une action  $a \in \mathcal{A}$  en l'état  $s \in \mathcal{S}$ .

Une distribution de possibilité  $\mathcal{X} \ni x \mapsto \pi(x | a, s) \in L$  classe ces conséquences :

$$\begin{aligned} \pi(x_1 | a, s) &\leq \pi(x_2 | a, s) \\ \pi(x_3 | a, s) &= 0_L \\ \pi(x_4 | a, s) &= 1_L \end{aligned}$$

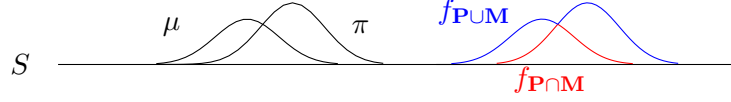
signifie que la conséquence  $x_1$  est moins plausible que  $x_2$ , que  $x_4$  est tout à fait possible, et que  $x_3$  est impossible.

De même, une fonction d'utilité  $\mathcal{X} \ni x \mapsto \mu(x) \in L$  classe ces conséquences en terme de satisfaction :

$$\begin{aligned} \mu(x_1) &\leq \mu(x_2) \\ \mu(x_3) &= 0_L \\ \mu(x_4) &= 1_L \end{aligned}$$

signifie que la conséquence  $x_1$  est moins satisfaisante que  $x_2$ , que  $x_4$  nous satisfait totalement, et que  $x_3$  est une conséquence redoutée.

FIGURE 3. Représentation de l'union et de l'intersection de deux ensembles flous



R.Sabbadin et al. ont mis en place dans [9] et [10] des critères de décision possibilistes dits :

$$\text{optimiste : } u^*(a, s) = \max_{x \in \mathcal{X}} \min\{\pi(x | a, s), \mu(x)\} \quad (3.5)$$

$$\text{et pessimiste : } u_*(a, s) = \min_{x \in \mathcal{X}} \max\{n(\pi(x | a, s)), \mu(x)\} \quad (3.6)$$

avec  $n : L \rightarrow L$  la fonction qui inverse l'ordre de  $L$  (pour  $l \in L$ ,  $n(l)$  est l'inverse de  $l$  modulo  $\text{Card}(L)$  si l'on prend  $L = \{0 := 0_L, \dots, \text{Card}(L) - 1 := 1_L\}$ , ou bien  $n(l) = 1 - l$  si  $L = \left\{ \frac{k}{\text{Card}(L)-1} \mid k \in \{0, \dots, \text{Card}(L) - 1\} \right\}$ ). Ils sont à maximiser, pour chaque état  $s \in \mathcal{S}$ , en l'action  $a \in \mathcal{A}$ . La théorie des possibilités est issue de la théorie des ensembles flous<sup>5</sup> : dans ce formalisme, il convient de considérer les fonctions  $\pi$  et  $\mu$  comme les fonctions d'appartenance de deux ensembles flous.

- **P** l'ensemble flou des conséquences  $x \in \mathcal{X}$  qui sont possibles : sa fonction caractéristique  $f_P = \pi$ ,
- **M** l'ensemble flou des conséquences  $x \in \mathcal{X}$  qui sont satisfaisantes :  $f_M = \mu$ .

Ainsi, comme la fonction d'appartenance de l'intersection de deux ensembles flous est le minimum des deux fonctions d'appartenance, et la fonction d'appartenance de leur union, le maximum (voir figure 3),

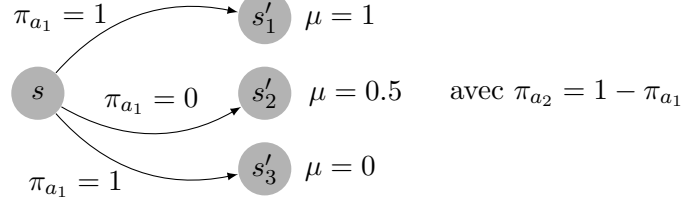
- $I(x) = \min\{\pi(x | a, s), \mu(x)\}$
- $U(x) = \max\{n(\pi(x | a, s)), \mu(x)\}$

sont respectivement la fonction d'appartenance de l'intersection de ces deux ensembles flous  $P \cap M$ , et la fonction d'appartenance de l'union du complémentaire du premier ensemble flou avec le deuxième  $P^c \cup M$ .

**Remarque :** Le simple fait de comparer ces deux grandeurs, qui ne signifient pas la même chose, demande un effort dans la modélisation : possibilités et préférences doivent être *commensurables*, c'est-à-dire que la comparaison doit avoir une signification. Lorsque  $\mu$  est à valeurs dans  $\{0_L, 1_L\}$ , cet effort n'est pas nécessaire : les états de préférence 1 sont des buts à atteindre, et les autres n'en sont pas.

5. Un ensemble flou (fuzzy set) est une généralisation de l'ensemble pour laquelle la fonction caractéristique (ou d'appartenance) de l'ensemble peut être n'importe quelle fonction de l'ensemble vers  $[0, 1]$  et non uniquement des indicatrices (vers  $\{0, 1\}$ ). Un élément d'un ensemble flou est plus ou moins dans cet ensemble selon la valeur que prend la fonction d'appartenance en cet élément.

FIGURE 4. Illustration de l'exemple de la section 3.1.2



3.5 est un critère optimiste car, connaissant l'état courant  $s$ , nous cherchons l'action qui maximise la "meilleure appartenance" à l'ensemble flou  $P \cap S$  des conséquences possibles et satisfaisantes : si, par malheur, une conséquence différente de celle qui maximise la fonction d'appartenance  $I(.)$  se réalise, rien n'assure alors qu'elle sera satisfaisante.

**Exemple :** Soient  $\mathcal{X} = \{x_1, x_2, x_3\}$ ,  $A = \{a_1, a_2\}$ . Pour alléger les notations, l'état courant  $s$  est sous-entendu ; il n'intervient pas dans cette illustration de l'optimisme du premier critère. Supposons que

- $1_L = \mu(x_1) > \mu(x_2) > \mu(x_3) = 0_L$
- $\pi(x_1|a_1) = \pi(x_3|a_1) = 1_L$ ,  $\pi(x_2|a_1) = 0_L$
- $\pi(x_1|a_2) = \pi(x_3|a_2) = 0_L$ ,  $\pi(x_2|a_2) = 1_L$

La figure 3.1.2 est un schéma de la situation. Nous obtenons alors

$$I^{a_1}(x) = \min\{\pi(x|a_1), \mu(x)\} = \begin{cases} 1_L & \text{si } x = x_1 \\ 0_L & \text{si } x = x_2 \text{ ou } x_3 \end{cases}$$

$$I^{a_2}(x) = \min\{\pi(x|a_2), \mu(x)\} = \begin{cases} \mu(x_2) & \text{si } x = x_2 \\ 0_L & \text{si } x = x_1 \text{ ou } x_3 \end{cases}$$

et donc  $u^*(a, s) = \begin{cases} 1_L & \text{si } a = a_1 \\ \mu(x_2) & \text{si } a = a_2 \end{cases}$

Ce critère est maximisé par l'action  $a_1$ , alors qu'une fois cette action choisie la conséquence  $x_1$  est tout aussi possible que la conséquence  $x_3$  dont la satisfaction est nulle. Cette "équipossibilité" se traduit par une nécessité nulle :  $\eta(\{x_1\}) = 1_L - \pi(\{x_2, x_3\}) = 1_L - \max(\pi(x_2), \pi(x_3)) = 0_L$ . Une action plus prudente serait de choisir  $a_2$ , dont la conséquence est certaine, mais la satisfaction moins bonne.

Cette prudence est prise en compte dans le critère (3.6), qui est pessimiste. Nous cherchons, en effet, l'action qui maximise la "plus mauvaise appartenance" à l'ensemble flou  $\bar{P} \cup S$ , *i.e.* qui minimise la "meilleure appartenance" à l'ensemble flou complémentaire  $P \cap \bar{S}$  (car si  $x$  a un degré d'appartenance à  $A$  de  $A(x)$ , son degré d'appartenance à  $A^c$  décroît lorsque  $A(x)$  croît).

Le critère 3.5 choisissait l'action qui maximise la possibilité et la satisfaction de la conséquence qui a ces deux qualités avec le plus de certitude. Quant à lui, le critère pessimiste (3.6) choisit l'action qui minimise la possibilité et l'insatisfaction de la conséquence qui est le plus susceptible de l'être. Formellement, comme  $n$  échange l'ordre de  $L$ ,

$$\begin{aligned}
 & \operatorname{argmax}_{a \in A} \quad \min_{x \in \mathcal{X}} \max\{n(\pi(x|a, s)), \mu(x)\} \\
 &= \operatorname{argmin}_{a \in A} \quad n(\min_{x \in \mathcal{X}} \max\{n(\pi(x|a, s)), \mu(x)\}) \\
 &= \operatorname{argmin}_{a \in A} \quad \max_{x \in \mathcal{X}} n(\max\{n(\pi(x|a, s)), \mu(x)\}) \\
 &= \operatorname{argmin}_{a \in A} \quad \max_{x \in \mathcal{X}} \min\{n(n(\pi(x|a, s))), n(\mu(x))\} \\
 &= \operatorname{argmin}_{a \in A} \quad \max_{x \in \mathcal{X}} \min\{\pi(x|a, s), n(\mu(x))\} \\
 & \quad (\text{car } n \text{ est une involution } n \circ n = Id)
 \end{aligned}$$

**Retour à l'exemple :** Avec le critère (3.6), nous obtenons

$$\begin{aligned}
 U^{a_1}(x) &= \max\{n(\pi(x|a_1)), \mu(x)\} = \begin{cases} 1_L & \text{si } x = x_1 \text{ ou } x_2 \\ 0_L & \text{sinon} \end{cases} \\
 U^{a_2}(x) &= \max\{n(\pi(x|a_2)), \mu(x)\} = \begin{cases} 1_L & \text{si } x = x_1 \text{ ou } x_3 \\ \mu(x_2) & \text{sinon} \end{cases} \\
 \text{et donc } u_*(a, s) &= \begin{cases} 0_L & \text{si } a = a_1 \\ \mu(x_2) & \text{si } a = a_2 \end{cases}
 \end{aligned}$$

Ici, c'est l'action  $a_2$  qui maximise le critère. Ce choix est plus prudent car, sous cette action,  $\eta(\{x_2\}) = 1_L - \pi(\{x_1, x_3\}) = 1_L$ , ce qui signifie que  $x_2$  se produit de manière certaine. Le gain est moindre, mais sûr.

Nous distinguons bien ici différentes similitudes entre probabilités et possibilités : nous comprenons que, dans la théorie possibiliste, le min et le max sont comparables respectivement au produit et à la somme de la théorie stochastique.

En effet, pour avoir la probabilité d'un événement  $A$  fini, il suffit de sommer les probabilités élémentaires de  $A$  :  $\mathbb{P}(A) = \sum_{\omega \in A} \mathbf{p}(\omega)$  tandis que la possibilité de  $A$  serait  $\max_{\omega \in \Omega} \pi(\omega)$ . Ces opérateurs (+ et max) sont dits des « opérateurs d'élimination », ou « de marginalisation ».

De même, le premier critère

$$\max_{x \in \mathcal{X}} \min\{\pi(x|s, a), \mu(x)\}$$

peut-être vu comme une espérance de la préférence (voir les intégrales de Sugeno), en comparaison avec le critère homologue probabiliste

$$\mathbb{E}[\mu | s] = \sum_{x \in \mathcal{X}} \mathbf{p}\{x | s, a\} \mu(x)$$

D'autre part, définissons la possibilité conditionnelle :

**Définition 8** (Possibilité conditionnelle).  $\forall A, B \subset \Omega$ ,

$$\Pi(A|B) = \begin{cases} 1_L & \text{si } \Pi(B) = \Pi(A \cap B) > 0_L \\ \Pi(A \cap B) & \text{sinon} \end{cases} \quad (3.7)$$

Supposons que la réalisation de  $A$  et  $B$  est possible ( $\Pi(A \cap B) > 0$ ), et de même degré de possibilité que l'évènement  $B$  (premier cas). Alors, si l'on sait que  $B$  se réalise,  $A$  est complètement possible. Cette affirmation est plus facile à comprendre en considérant un évènement élémentaire de  $A \cap B$  de possibilité maximale. Notons le  $\omega_0$  :  $\pi(\omega_0) = \Pi(A \cap B) > 0$ . Puisque  $\Pi(B) = \Pi(A \cap B)$ , cet évènement élémentaire est aussi de possibilité maximale dans  $B$ . Si l'univers des possibles devient  $B$ , cet évènement élémentaire  $\omega_0$  devient l'évènement élémentaire de possibilité maximale dans l'univers. Il paraît naturel, pour normaliser la nouvelle possibilité sachant  $B$ , d'attribuer à  $\omega_0$  une possibilité de  $1_L$ . Ainsi, comme  $\omega_0 \in A$ ,  $\Pi(A|B) = 1_L$ . Les possibilités qualitatives définissant uniquement un ordre, les évènements qui n'étaient pas de possibilité maximale dans  $B$  conservent une possibilité de degré  $\Pi(A \cap B)$  (second cas).

Nous pouvons remarquer que cette définition de  $\Pi(.|B)$  est la solution la moins spécifique de l'équation suivante, qui est l'homologue de l'équation de Bayes en théorie des possibilités qualitatives :

$$\min \{ \Pi(A|B), \Pi(B) \} = \Pi(A \cap B)$$

Ainsi, si  $A$  est indépendant de  $B$ , alors  $\Pi(A|B) = \Pi(A)$ , et l'équation devient  $\min \{ \Pi(A), \Pi(B) \} = \Pi(A \cap B)$ . C'est un point commun du min possibiliste avec le  $\times$  stochastique, que l'on appelle « opérateurs de combinaison ».

Remarquons cependant qu'en général  $\Pi(A \cap B) \leq \Pi(A)$  et  $\leq \Pi(B)$ , donc  $\Pi(A \cap B) \leq \min \{ \Pi(A), \Pi(B) \}$ .

Un formalisme plus général incluant ces quelques remarques est développé dans [8].

**3.2. Le modèle  $\pi$ -MDP.** Ce modèle, présenté dans [9] par R.Sabbadin, est une version possibiliste et qualitative du modèle MDP (détaillé section 2.1.1) basé sur les critères (optimiste et pessimiste) de la section 3.1.2.

**3.2.1. Mise en place du modèle.** Nous notons toujours  $\mathcal{S}$  l'ensemble fini des états dans lequel l'agent et son environnement peuvent se trouver,  $s_0$  l'état de départ,  $\mathcal{A}$  l'ensemble fini des actions disponibles, et  $L$  l'échelle (ensemble fini et totalement ordonné dont le plus grand élément est noté  $1_L$ , et le plus

petit  $0_L$ ).

$T$  sera maintenant l'ensemble des possibilités de transition, c'est-à-dire l'ensemble des distributions de possibilité sur l'état suivant lorsque l'état courant  $s \in \mathcal{S}$  et l'action  $a \in \mathcal{A}$  sont donnés :  $\pi(\cdot \mid s, a)$ , à valeurs dans  $L$ . Nous supposons qu'il existe une action « ne rien faire », que nous notons  $\bar{a}$ , qui permet de laisser le système dans l'état courant avec nécessité  $1_L$  :  $\exists a \in \mathcal{A}$  telle que  $\forall s \in \mathcal{S}, \pi(s \mid s, a) = 1_L$  et  $\forall s \neq s', \pi(s' \mid s, \bar{a}) = 0_L$ .  $\mu : \mathcal{S} \rightarrow L$  sera la distribution de préférence sur les états.

L'ensemble des politiques d'horizon  $n \in \mathbb{N}$ ,  $\delta = (\delta_0, \delta_1, \dots, \delta_{n-1})$  avec  $\forall i \in \{0, \dots, n-1\}, \delta_i : \begin{cases} \mathcal{S} & \rightarrow \mathcal{A} \\ s & \mapsto \delta_i(s) \end{cases}$ , est noté  $\Delta_n$ . L'ensemble des politiques finies est noté  $\Delta = \cup_{i \in \mathbb{N}} \Delta_i$ .

De même, nous notons  $\mathcal{T}_n$  l'ensemble des trajectoires de taille  $n$ , i.e.  $\mathcal{S}^{\{1, \dots, n\}}$ , et  $\mathcal{T} = \cup_{i \in \mathbb{N}} \mathcal{T}_i$ . Nous notons  $\#\tau$  la taille de la trajectoire  $\tau$ .

La fonction valeur qui suit est l'homologue optimiste de la fonction valeur des MDPs classiques. Une politique  $\delta \in \Delta$  qui réalise le maximum est une politique optimiste optimale :

$$u^\delta(s_0) = \max_{\tau \in \mathcal{T}} \min \left\{ \min_{i=0, \dots, \#\tau-1} \pi(s_{i+1} \mid s_i, \delta_i(s_i)), \mu(s_{\#\tau}) \right\} \quad (3.8)$$

Nous retrouvons l'opérateur de combinaison min qui permet de calculer la possibilité de la trajectoire  $\min_{i=0, \dots, \#\tau-1} \pi(s_{i+1} \mid s_i, \delta_i(s_i))$ , ainsi que l'intégrale de type max min tous deux évoqués en fin de section 3.1.2.

Comme d'habitude, la terminologie de la théorie des possibilités permet d'interpréter facilement la situation : une politique optimale pour ce critère est la meilleure politique pour rendre possible l'atteinte d'un état satisfaisant, et le plus satisfaisant possible.

La fonction valeur pour un horizon fini  $n$  et une politique  $\delta \in \Delta_n$  est définie par

$$u_n^\delta(s_0) = \max_{(s_1, \dots, s_n) \in \mathcal{T}_n} \min \left\{ \min_{i=0, \dots, n-1} \pi(s_{i+1} \mid s_i, \delta_i(s_i)), \mu(s_n) \right\} \quad (3.9)$$

Montrons que pour  $n > 0$ ,  $u_n^*(s) = \max_{\delta \in \Delta_n} u_n^\delta(s)$  est solution de l'équation de Bellman

$$u_n^*(s_0) = \max_{a \in \mathcal{A}} \max_{s' \in \mathcal{S}} \min \left\{ \pi(s' \mid s_0, a), u_{n-1}^*(s') \right\}, \quad (3.10)$$

avec la notation  $u_0^*(s_0) = \mu(s_0)$ .

*Démonstration.* Le cas  $n = 1$  est exactement la définition de la fonction valeur optimale (en effet dans le cas de l'horizon 1, une politique et une

trajectoire se résument respectivement à une action et un état).

Si  $n > 1$ ,

$$\begin{aligned}
 u_n^*(s_0) &= \max_{\delta \in \Delta_n} \max_{(s_1, \dots, s_n) \in \mathcal{T}_n} \min \left\{ \min_{i=0, \dots, n-1} \pi(s_{i+1} \mid s_i, \delta_i(s_i)), \mu(s_n) \right\} \\
 &= \max_{(a, \delta') \in \mathcal{A} \times \Delta_{n-1}} \max_{(s_1, \dots, s_n) \in \mathcal{T}_n} \min \left\{ \pi(s_1 \mid s_0, a), \min \left\{ \min_{i=1, \dots, n-1} \pi(s_{i+1} \mid s_i, \delta_i(s_i)), \mu(s_n) \right\} \right\} \\
 &= \max_{a \in \mathcal{A}} \max_{\delta' \in \Delta_{n-1}} \max_{s_1 \in \mathcal{S}} \max_{\tau' \in \mathcal{T}_{n-1}} \min \left\{ \pi(s_1 \mid s_0, a), \min \left\{ \min_{i=1, \dots, n-1} \pi(s_{i+1} \mid s_i, \delta_i(s_i)), \mu(s_n) \right\} \right\} \\
 &= \max_{a \in \mathcal{A}} \max_{s_1 \in \mathcal{S}} \min \left\{ \pi(s_1 \mid s_0, a), \max_{\delta' \in \Delta_{n-1}} \max_{\tau' \in \mathcal{T}_{n-1}} \min \left\{ \min_{i=1, \dots, n-1} \pi(s_{i+1} \mid s_i, \delta_i(s_i)), \mu(s_n) \right\} \right\} \\
 &= \max_{a \in \mathcal{A}} \max_{s_1 \in \mathcal{S}} \min \left\{ \pi(s_1 \mid s_0, a), u_{n-1}^*(s_1) \right\}
 \end{aligned} \tag{3.11}$$

où  $\delta' = (\delta_1, \dots, \delta_{n-1})$ ,  $\tau' = (s_2, \dots, s_n)$  et  $a = \delta(s_0)$ .

La ligne 3.11 s'explique par le resultat suivant :

$$\max_{x \in \mathcal{X}} \min \{A, B(x)\} = \min \left\{ A, \max_{x \in \mathcal{X}} B(x) \right\} \tag{3.12}$$

avec

- $x = (\delta', \tau') \in \Delta_{n-1} \times \mathcal{T}_n$ ,
- $A = \pi(s_1 \mid s_0, a)$  qui ne dépend ni de  $\delta'$ , ni de  $\tau'$ ,
- et  $B(x) = \min \{ \min_{i=1, \dots, n-1} \pi(s_{i+1} \mid s_i, \delta_i(s_i)), \mu(s_n) \}$ .

En effet, cette égalité est vraie

- lorsque  $A \geq \max_{x \in \mathcal{X}} B(x)$   
 $(\Rightarrow \forall x \in \mathcal{X}, A \geq B(x) \Rightarrow \text{les deux membres valent } \max_{x \in \mathcal{X}} B(x))$ ,
- et lorsque  $A \leq \max_{x \in \mathcal{X}} B(x)$   
 $(\Rightarrow \text{le membre de droite vaut } A. \forall x \in \mathcal{X}, \min \{A, B(x)\} \leq A \text{ et } \min \{A, B(x)\} = A \text{ pour } x \text{ tel que } B(x) \geq A, \text{ donc le membre de gauche aussi.})$

□

Ce résultat permet la mise en place de l'algorithme 3. La politique  $\delta$  retournée par cet algorithme est optimale car

$$\begin{aligned}
 u_n^*(s_0) &= \max_{a \in \mathcal{A}} \max_{s_1 \in \mathcal{S}} \min \{ \pi(s_1 | s_0, a), u_{n-1}^*(s_1) \} \\
 &= \max_{s_1 \in \mathcal{S}} \min \{ \pi(s_1 | s_0, \delta_0(s)), u_{n-1}^*(s_1) \} \\
 &= \max_{s_1 \in \mathcal{S}} \min \left\{ \pi(s_1 | s_0, \delta_0(s)), \max_{a \in \mathcal{A}} \max_{s_2 \in \mathcal{S}} \min \{ \pi(s_2 | s_1, a), u_{n-2}^*(s_2) \} \right\} \\
 &= \max_{s_1 \in \mathcal{S}} \min \left\{ \pi(s_1 | s_0, \delta_0(s)), \max_{s_2 \in \mathcal{S}} \min \{ \pi(s_2 | s_1, \delta_1(s_1)), u_{n-2}^*(s_2) \} \right\} \\
 &= \max_{s_1 \in \mathcal{S}} \max_{s_2 \in \mathcal{S}} \min \{ \pi(s_1 | s_0, \delta_0(s)), \min \{ \pi(s_2 | s_1, \delta_1(s_1)), u_{n-2}^*(s_2) \} \} \\
 &= \max_{s_1 \in \mathcal{S}} \max_{s_2 \in \mathcal{S}} \min \left\{ \min_{i=0,1} \pi(s_{i+1} | s_i, \delta_i(s)), u_{n-2}^*(s_2) \right\} \\
 &= \vdots \\
 &= \max_{(s_1, \dots, s_n) \in T_n} \min \left\{ \min_{i=0, \dots, n-1} \pi(s_{i+1} | s_i, \delta_i(s)), \mu(s_n) \right\}
 \end{aligned}$$

toujours par le résultat 3.12.

L'indice  $k$  de  $\delta_k$  désigne l'étape du processus à laquelle il faut utiliser

---

**Algorithm 3** possibilistic backward induction
 

---

```

for  $s \in \mathcal{S}$  do
     $u_0^*(s) \leftarrow \mu(s)$ 
end for
for  $i \in \{1, \dots, n\}$  do
    for  $s \in \mathcal{S}$  do
         $u_i^*(s) \leftarrow \max_{a \in \mathcal{A}} \max_{s' \in \mathcal{S}} \min \{ \pi(s' | s, a), u_{i-1}^*(s') \}$ 
         $\delta_{n-i}(s) \leftarrow \operatorname{argmax}_{a \in \mathcal{A}} \max_{s' \in \mathcal{S}} \min \{ \pi(s' | s, a), u_n^*(s') \}$ 
    end for
end for
return  $u_n^*, \delta$ 
    
```

---

$\delta_k : \mathcal{S} \rightarrow \mathcal{A}$  lors de l'exécution, et celui de  $u_k$  est la taille de l'horizon. C'est pourquoi ces indices sont opposés modulo  $n$ .

Nous remarquons maintenant que  $\forall s_0 \in \mathcal{S}, \forall n \in \mathbb{N}, u_{n+1}^*(s_0) \geq u_n^*(s_0)$ . En effet, soit  $\delta \in \Delta_n$  une politique optimale pour un horizon de taille  $n$ , et  $\tau = (s_1, \dots, s_n)$  la trajectoire qui maximise

$$\min \left\{ \min_{i=0, \dots, n-1} \pi(s_{i+1} | s_i, \delta_i(s_i)), \mu(s_n) \right\}$$

Posons  $s_{n+1} = s_n$ . La politique  $\delta' = (\delta, \bar{a}) \in \Delta_{n+1}$  est telle que

$$\begin{aligned}
 u_n^*(s_0) &= \min \left\{ \min_{i=0, \dots, n} \pi(s_{i+1} | s_i, \delta'_i(s_i)), \mu(s_{n+1}) \right\} \\
 &= u_{n+1}^{\delta'}(s_0) \leq u_{n+1}^*(s_0)
 \end{aligned} \tag{3.13}$$



car  $s_{n+1} = s_n$  et donc  $\pi(s_{n+1} \mid s_n, \bar{a}) = 1_L$ .

**3.2.2. Résolution d'un  $\pi$ -MDP à horizon quelconque.** L'algorithme 4 permet de calculer le maximum en  $\delta \in \Delta$  de 3.8, *i.e.* la fonction valeur optimale pour un horizon quelconque

$$u^*(s_0) = \max_{\delta \in \Delta} \max_{\tau \in \mathcal{T}} \min \left\{ \min_{i=0, \dots, \#\tau-1} \pi(s_{i+1} \mid s_i, \delta_i(s_i)), \mu(s_{\#\tau}) \right\}$$

ainsi qu'une politique optimale stationnaire, *i.e.*  $\delta^*$  telle que

$$\max_{\delta \in \Delta} u^\delta = u^{\delta^*}$$

---

**Algorithm 4** optimistic  $\pi$ -MDP iteration value algorithm

---

```

for  $s \in \mathcal{S}$  do
     $u^*(s) \leftarrow \mu(s)$ 
     $u^c(s) \leftarrow 0$ 
     $\delta(s) \leftarrow \bar{a}$ 
end for
while  $u^* \neq u^c$  do
    for  $s \in \mathcal{S}$  do
         $u^c(s) \leftarrow \max_{a \in \mathcal{A}} \max_{s' \in \mathcal{S}} \min \{ \pi(s' \mid s, a), u^*(s') \}$ 
        if  $u^c(s) > u^*(s)$  then
             $\delta(s) \in \operatorname{argmax}_{a \in \mathcal{A}} \max_{s' \in \mathcal{S}} \min \{ \pi(s' \mid s, a), u^*(s') \}$ 
        end if
    end for
     $u^* = u^c$ 
end while
return  $u^*, \delta$ 
    
```

---

Cet algorithme est l'homologue de l'algorithme d'itération sur les valeurs de la résolution des MDPs (section 2.1.2). Nous notons que l'initialisation n'est pas arbitraire contrairement à l'algorithme 1.

Nous complétons maintenant la preuve de [9] afin de s'assurer que l'algorithme 4 nous retourne bien la fonction valeur optimale et une politique optimale. Tout d'abord, notons que la variable  $u^*(s)$  dans l'algorithme prend la valeur  $u_n^*$  à la  $n^{ieme}$  itération. Puisque l'action  $\bar{a}$  (rester) existe, comme expliqué en 3.13, la suite  $(u_n^*(s))_{n \in \mathbb{N}}$  est croissante au cours de l'algorithme  $\forall s \in \mathcal{S}$ . Par ailleurs, la politique retournée consistera à exécuter dans l'état  $s$  l'action qui a fait augmenter la variable  $u^*(s)$  de l'algorithme pour la dernière fois, ou bien  $\bar{a}$  si  $(u_n^*(s))_{n \in \mathbb{N}}$  a été constante depuis le début. Rappelons que les distributions de possibilités et de préférences prennent

leurs valeurs dans l'échelle  $L$  qui est un ensemble fini totalement ordonné. A chaque itération  $u_n^*$  dépend de  $u_{n-1}^*$  :

$$u_n^*(s) \leftarrow \max_{a \in \mathcal{A}} \max_{s' \in \mathcal{S}} \min \{ \pi(s' | s, a), u_{n-1}^*(s') \}$$

De ce fait, si, lors de l'itération  $n$ ,  $\forall s \in \mathcal{S}$ ,  $u_n^*(s) = u_{n+1}^*(s)$ , alors, à l'itération suivante  $(n+1)$ , la situation sera la même ( $u_{n+1}^*(s) = u_{n+2}^*(s) \forall s \in \mathcal{S}$ ).

Cette situation se produit au bout d'un nombre fini d'itérations. En effet, dans le pire des cas, la variable de l'algorithme  $u^*(s)$  passe d'une valeur de  $L$  à la valeur de  $L$  juste supérieure à chaque itération, pour un état  $s \in \mathcal{S}$  seulement. Ainsi, le nombre d'itération est limité au nombre de  $\text{Card}(L) \times \text{Card}(\mathcal{S})$ , et la convergence de  $u^*$  a bien lieu en temps fini. Cette fonction converge vers la fonction valeur optimale car elle correspond à la fonction valeur pour un horizon  $\text{Card}(L) \times \text{Card}(\mathcal{S})$ , et nous savons maintenant qu'elle ne peut être plus grande.

Avant de montrer que la politique stationnaire retournée par l'algorithme est optimale, démontrons deux petits résultats : notons  $\delta^*$  cette politique stationnaire.

*Démonstration.* Soient  $s_0 \in \mathcal{S}$ , et  $n$  le plus petit entier tel que  $u_n^*(s_0) = u^*(s_0)$ , où  $u^*$  est ici la fonction valeur optimale pour un horizon quelconque (la variable  $u^*(s)$  de algorithme 4 n'augmente plus après  $n$  itérations). L'algorithme 3 nous fournit une politique optimale que nous notons  $\delta^{(s_0)} \in \Delta_n$ . Avec cette notation,  $\delta^*(s) = \delta_0^{(s)}(s)$ .

Considérons une trajectoire

$$\tau = (s_1, s_2, \dots, s_n) \in \text{argmax}_{T_n} \min \left\{ \min_{i=0, \dots, n-1} \pi(s_{i+1} | s_i, \delta_i^{(s_0)}(s_i)), \mu(s_n) \right\}$$

(1) Soit  $k \in \{1, \dots, n\}$ .

$$\begin{aligned} u^*(s_0) &= \min \left\{ \min_{i=0, \dots, n-1} \pi(s_{i+1} | s_i, \delta_i^{(s_0)}(s_i)), \mu(s_n) \right\} \\ &= \min \left\{ \min_{i=0, \dots, k-1} \pi(s_{i+1} | s_i, \delta_i^{(s_0)}(s_i)), \min \left\{ \min_{i=k, \dots, n-1} \pi(s_{i+1} | s_i, \delta_i^{(s_0)}(s_i)), \mu(s_n) \right\} \right\} \\ &= \min \left\{ \min_{i=0, \dots, k-1} \pi(s_{i+1} | s_i, \delta_i^{(s_0)}(s_i)), u_{n-k}^*(s_k) \right\} \\ &\leq u_{n-k}^*(s_k) \leq u^*(s_k) \end{aligned} \tag{3.14}$$

(2) Maintenant, supposons que  $u^*(s_0) = u^*(s_k)$ . Alors comme

$$u^*(s_0) \leq u_{n-k}^*(s_k) \leq u^*(s_k), \quad u_{n-k}^*(s_k) = u^*(s_k).$$

La fonction valeur en  $s_k$  est donc optimisée par un horizon  $n - k$ .

Mais nous avons  $u_{n-k-m}^*(s_k) < u^*(s_k)$  *i.e.* un horizon plus petit  $n - k - m$  ( $n - k \geq m > 0$ ) ne l'optimise pas, sinon la fonction valeur en  $s_0$  serait optimisée par un horizon  $n - m$  : la politique  $\delta' = (\delta_0^{(s_0)}, \delta_1^{(s_0)}, \dots, \delta_{k-1}^{(s_0)}, \delta_0^{(s_k)}, \dots, \delta_{n-k-m-1}^{(s_k)}) \in \Delta_{n-m}$  serait optimale car

$$\begin{aligned} u^*(s_0) &= \min \left\{ \min_{i=0, \dots, k-1} \pi \left( s_{i+1} \mid s_i, \delta_i^{(s_0)}(s_i) \right), u_{n-k}^*(s_k) \right\} \\ &= \min \left\{ \min_{i=0, \dots, k-1} \pi \left( s_{i+1} \mid s_i, \delta_i^{(s_0)}(s_i) \right), u^*(s_k) \right\} \end{aligned}$$

et comme  $u^*(s_k)$  serait optimisée par un horizon  $n - k - m$ , en notant  $\tau^{(1)} = (s_1^{(1)}, \dots, s_{n-k-m}^{(1)}) \in T_{n-k-m}$  une trajectoire maximisante,

$$u^*(s_k) = \min \left\{ \min_{i=0, \dots, n-k-m-1} \pi \left( s_{i+1}^{(1)} \mid s_i^{(1)}, \delta_i^{(s_k)}(s_i^{(1)}) \right), \mu(s_{n-k-m}^{(1)}) \right\}$$

et donc, en notant  $\tau' = (s'_1, \dots, s'_{n-m}) = (s_1, \dots, s_{k-1}, s_0^{(1)}, \dots, s_{n-k-m}^{(1)})$ ,

$$u^*(s_0) = \min \left\{ \min_{i=0, \dots, n-m-1} \pi \left( s'_{i+1} \mid s'_i, \delta'_i(s'_i) \right), \mu(s'_{n-m}) \right\}$$

*i.e.*  $\exists n' < n$  tel que  $u^*(s_0) = u_{n'}^*(s_0)$  ce qui contredit le choix initial de  $n$ .

Ainsi,

$$\begin{aligned} \text{si} \quad & u^*(s_0) = u^*(s_k), \\ \text{alors} \quad & \delta^*(s_k) := \delta_0^{(s_k)}(s_k) = \delta_k^{(s_0)}(s_k). \end{aligned} \quad (3.15)$$

Montrons maintenant que  $\delta^*$  optimise le critère, *i.e.* la fonction valeur pour un horizon quelconque.

Si  $\forall k \in \{1, \dots, n-1\}$   $\delta^*(s_k) := \delta_0^{(s_k)}(s_k) = \delta_k^{(s_0)}(s_k)$  et si  $\delta^*(s_n) = \bar{a}$ ,  $\delta^{(s_0)}(s) = \delta^*(s) \forall s \in \tau$ , et donc le critère est maximisé par  $\delta^*$  puisque maximisé par  $\delta^{(s_0)}$ . Sinon, soit  $k$  le plus petit entier  $\in \{1, \dots, n-1\}$  tel que  $\delta_0^{(s_k)}(s_k) \neq \delta_k^{(s_0)}(s_k)$  (si il n'existe pas,  $\delta^*(s_n) \neq \bar{a}$  et  $k = n$ ). Par les deux résultats 3.14 et 3.15, nous avons  $u^*(s_k) > u^*(s_0)$ , et même  $u^*(s_k) > u^*(s_i) \forall i < k$  (sinon, sélectionner un  $k$  inférieur, jusqu'à ce que cela soit vrai).

Nous réitérons avec  $s_0^{(1)} = s_k$  : soit  $n^{(1)}$  le nombre d'itérations avant que la variable de l'algorithme  $u^*(s^{(1)})$  converge.

soit  $\tau^{(1)} \in \operatorname{argmax}_{T_n^{(1)}} \min \left\{ \min_{i=0, \dots, n^{(1)}-1} \pi \left( s_{i+1} \mid s_i, \delta_i^{(1)}(s_i) \right), \mu(s_n^{(1)}) \right\}$ .

Nous sélectionnons, s'il existe,  $k^{(1)}$  de la même manière que précédemment, et réitérons avec  $s_0^{(2)} = s_{k^{(1)}}^{(1)}$  qui est tel que  $u^*(s_{k^{(1)}}^{(1)}) > u^*(s_0^{(1)})$ ,

et  $u^*(s_{k^{(1)}}^{(1)}) > u^*(s_i^{(1)}) \forall i < k^{(1)}$ .

Tous les états des trajectoires sélectionnées sont différents : en effet, au sein d'une même trajectoire  $\tau^{(m)}$ , deux états égaux remettrait en cause l'hypothèse d'horizon minimum. Si une trajectoire  $\tau^{(m)}$  a un état  $\bar{s}$  en commun

avec une autre trajectoire  $\tau^{(m')}$  ( $m' < m$ ),  $u^*(s_0^{(m)}) \leq u^*(\bar{s}) < u^*(s_0^{(m'+1)})$  (d'après 3.14 et l'hypothèse que  $u^*(s_0^{m'+1})$  est strictement supérieur à la fonction valeur optimale en chacun des états  $s_0^{(m')}, \dots, s_{k^{(m')}-1}^{(m')}$ ), alors que, par hypothèse,  $u^*(s^{(m'+1)}) \leq u^*(s^{(m)})$  puisque  $m' < m$ . C'est une contradiction, et donc tous les états sélectionnés sont différents.

Ainsi, le procédé s'arrête *i.e.*  $\exists m \in \mathbb{N}$  tel que  $k^{(m)}$  n'existe pas, car  $\text{Card}(\mathcal{S})$  est fini. Nous notons  $n^*$  le nombre total d'état sélectionnés ( $k + \sum_{i=0}^{m-1} k^{(i)} + n^{(m)}$ ). La politique  $\delta' = (\delta_0, \dots, \delta_{k-1}, \delta_0^{(1)}, \dots, \delta_{k^{(1)}-1}^{(1)}, \dots, \delta_{n^{(m)}-1}^{(m)})$  correspond alors à  $\delta^*$  sur  $\tau' = (s'_1, \dots, s'_{n^*}) = \left\{ s_0, s_1, \dots, s_{k-1}, s_0^{(1)}, \dots, s_{k^{(1)}-1}^{(1)}, \dots, s_{n^{(m)}-1}^{(m)} \right\}$ , et nous avons montré que la politique était optimale car  $u^*(s_0) = u^{\delta^*}(s_0)$  :

$$\begin{aligned} u^*(s_0) &= \min \left\{ \min_{i=0, \dots, k-1} \pi(s'_{i+1} \mid s'_i, \delta'(s'_i)), u_{n-k}(s_k) \right\} \\ &\leq \min \left\{ \min_{i=0, \dots, k-1} \pi(s'_{i+1} \mid s'_i, \delta'(s'_i)), u^*(s_k) \right\} \\ &= \min \left\{ \min_{i=0, \dots, k^{(1)}-1} \pi(s'_{i+1} \mid s'_i, \delta'(s'_i)), u_{n^{(1)}-k^{(1)}}(s_{k^{(1)}}) \right\} \\ &\vdots \\ &\leq \min \left\{ \min_{i=0, \dots, n^*-1} \pi(s'_{i+1} \mid s'_i, \delta'(s'_i)), u^*(s'_{n^*}) \right\} \end{aligned}$$

les  $\ll \leq \gg$  sont en fait des  $\ll = \gg$  sinon nous aurions trouvé une politique telle que  $u^{\delta'}(s_0) > u^*(s_0)$ . Comme  $\delta^*(s'_{n^*}) = \bar{a}$ , la variable  $u^*(s'_{n^*})$  n'a jamais évolué au cours de l'algorithme (rester en  $s'_{n^*}$  est nécessaire sous l'action  $\bar{a}$ ), et donc  $u^*(s'_{n^*}) = \mu(s'_{n^*})$ , ce qui nous donne

$$u^*(s_0) = \min \left\{ \min_{i=0, \dots, n^*-1} \pi(s'_{i+1} \mid s'_i, \delta^*(s'_i)), \mu(s'_{n^*}) \right\}$$

□

Le critère pessimiste 3.6 mène aussi à une fonction valeur (3.16) qu'il est possible d'optimiser pour obtenir  $u_*(s) = \max_{\delta} u_{\delta}(s)$  et pour laquelle nous pouvons disposer d'une politique optimale grâce à l'algorithme 5.

$$u_{\delta}(s_0) = \min_{\tau \in T} \max \left\{ n \left( \min_{i=0, \dots, \# \tau - 1} \pi(s_{i+1} \mid s_i, \delta(s_i)) \right), \mu(s_{\# \tau}) \right\} \quad (3.16)$$

Ce critère favorise les politiques qui rendent peu plausible les états peu satisfaisants. La validité de cet algorithme se prouve d'une manière similaire à la preuve de l'algorithme 4. Montrons tout de même l'équation de Bellman de cette version pessimiste. Nous partons de la fonction valeur optimale pour un horizon  $n$ , pour aboutir à l'équation de Bellman liant l'horizon  $n$  à l'horizon  $n-1$

$$u_{n*}(s_0) = \max_{\delta \in \Delta_n} \min_{\tau \in T_n} \max \left\{ n \left( \min_{i=0, \dots, n-1} \pi(s_{i+1} \mid s_i, \delta_i(s_i)) \right), \mu(s_n) \right\}$$

---

**Algorithm 5** pessimistic  $\pi$ -MDP iteration value algorithm
 

---

```

for  $s \in \mathcal{S}$  do
     $u_*(s) \leftarrow \mu(s)$ 
     $u_c(s) \leftarrow 0$ 
     $\delta(s) \leftarrow \bar{a}$ 
end for
while  $u_* \neq u^c$  do
    for  $s \in \mathcal{S}$  do
         $u_c(s) \leftarrow \max_{a \in \mathcal{A}} \min_{s' \in \mathcal{S}} \max \{ n(\pi(s' | s, a)), u^*(s') \}$ 
        if  $u_c(s) > u_*(s)$  then
             $\delta(s) \in \operatorname{argmax}_{a \in \mathcal{A}} \min_{s' \in \mathcal{S}} \max \{ n(\pi(s' | s, a)), u^*(s') \}$ 
        end if
    end for
     $u_* = u_c$ 
end while
return  $u_*, \delta$ 
    
```

---

comme  $\forall l_1, l_2 \in L, n(\min \{l_1, l_2\}) = \max \{n(l_1), n(l_2)\}$ ,

$$\begin{aligned}
 & n(\min_{i=0, \dots, n-1} \pi(s_{i+1} | s_i, \delta_i(s_i))) \\
 &= n\left(\min \left\{ \pi(s_1 | s_0, \delta_0(s_0)), \min_{i=1, \dots, n-1} \pi(s_{i+1} | s_i, \delta_i(s_i)) \right\}\right) \\
 &= \max \left\{ n(\pi(s_1 | s_0, \delta_0(s_0))), n\left(\min_{i=1, \dots, n-1} \pi(s_{i+1} | s_i, \delta_i(s_i))\right) \right\}
 \end{aligned}$$

ainsi,

$u_{n*}(s_0)$

$$= \max_{\delta \in \Delta_n} \min_{\tau \in T_n} \max \left\{ n(\pi(s_1 | s_0, \delta_0(s_0))), \max \left\{ n\left(\min_{i=1, \dots, n-1} \pi(s_{i+1} | s_i, \delta_i(s_i))\right), \mu(s_n) \right\} \right\}$$

Nous montrons maintenant par récurrence sur  $2 \leq n$  que pour toute fonction

$$F : \begin{cases} S_1 \times \dots \times S_n \times \mathcal{A}_1 \times \dots \times \mathcal{A}_n \rightarrow L \\ (s_1, \dots, s_n, a_1, \dots, a_n) \mapsto F(s_1, \dots, s_n, a_1, \dots, a_n) \end{cases},$$

$$\begin{aligned}
 & \max_{(\delta_1, \dots, \delta_n)} \min_{(s_1, \dots, s_n) \in \mathcal{S}^n} F(s_1, \dots, s_n, \delta_1(s_1), \dots, \delta_n(s_n)) \\
 &= \max_{\delta_1 \in \mathcal{A}_1^{S_1}} \min_{s_1 \in S_1} \dots \max_{\delta_n \in \mathcal{A}_n^{S_n}} \min_{s_n \in S_n} F(s_1, \dots, s_n, \delta_1(s_1), \dots, \delta_n(s_n))
 \end{aligned}$$

*Démonstration.* Tout d'abord, soit  $L$  fini et totalement ordonné,  $\mathcal{A}$  et  $\mathcal{S}$  deux ensembles finis,

$$\begin{aligned}
 \mathcal{A}^{\mathcal{S}} &= \{ \delta : \mathcal{S} \rightarrow \mathcal{A} \} \quad (fini) \\
 f : \mathcal{S} \times \mathcal{A} &\longrightarrow L
 \end{aligned}$$

Alors,  $\max_{\delta \in \mathcal{A}^S} \min_{s \in \mathcal{S}} f(s, \delta(s)) = \min_{s \in \mathcal{S}} \max_{\delta \in \mathcal{A}^S} f(s, \delta(s))$ .

En effet,

$$\begin{aligned} & \forall \delta \in \mathcal{A}^S, \forall s \in \mathcal{S} \quad \min_{\xi \in \mathcal{S}} f(\xi, \delta(\xi)) \leq f(s, \delta(s)) \\ \Rightarrow & \forall \delta \in \mathcal{A}^S, \forall s \in \mathcal{S} \quad \min_{\xi \in \mathcal{S}} f(\xi, \delta(\xi)) \leq \max_{\gamma \in \mathcal{A}^S} f(s, \gamma(s)) \\ \Rightarrow & \forall s \in \mathcal{S} \quad \max_{\gamma \in \mathcal{A}^S} \min_{\xi \in \mathcal{S}} f(\xi, \gamma(\xi)) \leq \max_{\gamma \in \mathcal{A}^S} f(s, \gamma(s)) \\ \Rightarrow & \max_{\gamma \in \mathcal{A}^S} \min_{\xi \in \mathcal{S}} f(\xi, \gamma(\xi)) \leq \min_{\xi \in \mathcal{S}} \max_{\gamma \in \mathcal{A}^S} f(\xi, \gamma(\xi)) \end{aligned}$$

Ensuite,

$$\forall \delta \in \mathcal{A}^S, \max_{\gamma \in \mathcal{A}^S} \min_{\xi \in \mathcal{S}} f(\xi, \gamma(\xi)) \geq \min_{\xi \in \mathcal{S}} f(\xi, \delta(\xi))$$

Prenons alors  $\delta$  tel que  $\forall s \in \mathcal{S}, \delta(s) = \operatorname{argmax}_{a \in \mathcal{A}} f(s, a)$ . Nous obtenons ainsi

$$\max_{\gamma \in \mathcal{A}^S} \min_{\xi \in \mathcal{S}} f(\xi, \gamma(\xi)) \geq \min_{\xi \in \mathcal{S}} \max_{\gamma \in \mathcal{A}^S} f(\xi, \gamma(\xi))$$

et nous avons donc bien l'égalité.

Nous initialisons maintenant la récurrence grace au résultat précédent :

$$\begin{aligned} & \max_{(\delta_1, \delta_2) \in \mathcal{A}_1^{S_1} \times \mathcal{A}_2^{S_2}} \min_{(s_1, s_2) \in \mathcal{S}_1 \times \mathcal{S}_2} F(s_1, s_2, \delta_1(s_1), \delta_2(s_2)) \\ &= \min_{s_1, s_2} \max_{\delta_1, \delta_2} F(s_1, s_2, \delta_1(s_1), \delta_2(s_2)) \\ &\geq \min_{s_1} \max_{\delta_1} \min_{s_2 \in \mathcal{S}} \max_{\delta_2} F(s_1, s_2, \delta_1(s_1), \delta_2(s_2)) \\ &= \min_{s_1} \max_{\delta_1} \max_{\delta_2} \min_{s_2} F(s_1, s_2, \delta_1(s_1), \delta_2(s_2)) \\ &\geq \max_{\delta_1} \max_{\delta_2} \min_{s_1} \min_{s_2} F(s_1, s_2, \delta_1(s_1), \delta_2(s_2)) \end{aligned}$$

Comme la première et la dernière ligne sont égales, toutes les inégalités sont des égalités, et nous avons donc le résultat au rang  $n = 2$ .

Maintenant, faisons l'hypothèse que l'égalité est vraie au rang  $n$  et montrons le rang  $n+1$  : en notant  $s'_n = (s_n, s_{n+1})$ ,  $\mathcal{S}' = \mathcal{S}_n \times \mathcal{S}_{n+1}$ ,  $\delta' = (\delta_n, \delta_{n+1})$ , et  $\mathcal{A}'_n = \mathcal{A}_n \times \mathcal{A}_{n+1}$ ,

$$\begin{aligned} & \max_{\delta_1, \dots, \delta_{n+1}} \min_{s_1, \dots, s_{n+1}} F(s_1, \dots, s_{n+1}, \delta_1, \dots, \delta_{n+1}) \\ &= \max_{\delta_1, \dots, \delta'_n} \min_{s_1, \dots, s'_n} F(s_1, \dots, s'_n, \delta_1, \dots, \delta'_n) \\ &= \max_{\delta_1 \in \mathcal{A}_1^{S_1}} \min_{s_1 \in \mathcal{S}_1} \dots \max_{\delta'_n \in (\mathcal{A}'_n)^{S'_n}} \min_{s'_n \in \mathcal{S}'_n} F(s_1, \dots, s'_n, \delta_1, \dots, \delta'_n) \\ &= \max_{\delta_1 \in \mathcal{A}^S} \min_{s_1 \in \mathcal{S}} \dots \max_{(\delta_n, \delta_{n+1}) \in (\mathcal{A}'_n)^{S'_n}} \min_{(s_n, s_{n+1}) \in \mathcal{S}'_n} F(s_1, \dots, s_n, s_{n+1}, \delta_1, \dots, \delta_n, \delta_{n+1}) \\ &= \max_{\delta_1 \in \mathcal{A}_1^{S_1}} \min_{s_1 \in \mathcal{S}_1} \dots \max_{\delta_n \in \mathcal{A}_n^{S_n}} \min_{s_n \in \mathcal{S}_n} \max_{\delta_{n+1} \in \mathcal{A}_{n+1}^{S_{n+1}}} \min_{s_{n+1} \in \mathcal{S}_{n+1}} F(s_1, \dots, s_n, s_{n+1}, \delta_1, \dots, \delta_n, \delta_{n+1}) \end{aligned}$$

Continuons la recherche de l'équation de Bellman :

$$\begin{aligned}
& u_{n*}(s_0) \\
&= \max_{\delta \in \Delta_n} \min_{\tau \in T_n} \max \left\{ n(\pi(s_1 | s_0, \delta_0(s_0))), \max \left\{ n \left( \min_{i=1, \dots, n-1} \pi(s_{i+1} | s_i, \delta_i(s_i)) \right), \mu(s_n) \right\} \right\} \\
&= \max_{\delta_0} \max_{\delta_1} \min_{s_1} \dots \max_{\delta_{n-1}} \min_{s_{n-1}} \max_{s_n} \left\{ n(\pi(s_1 | s_0, \delta_0(s_0))), U_{s_1, \dots, s_n, \delta_1(s_1), \dots, \delta_{n-1}(s_{n-1})} \right\} \\
&= \max_{\delta_0} \min_{s_1} \max_{\delta_1} \dots \max_{\delta_{n-1}} \min_{s_{n-1}} \max_{s_n} \left\{ n(\pi(s_1 | s_0, \delta_0(s_0))), U_{s_1, \dots, s_n, \delta_1(s_1), \dots, \delta_{n-1}(s_{n-1})} \right\} \\
&= \max_{\delta_0} \min_{s_1} \max \left\{ n(\pi(s_1 | s_0, \delta_0(s_0))), \max_{\delta_1} \dots \max_{\delta_{n-1}} \min_{s_{n-1}} \min_{s_n} U_{s_1, \dots, s_n, \delta_1(s_1), \dots, \delta_{n-1}(s_{n-1})} \right\} \\
&= \max_{\alpha} \min_{s_1} \max \left\{ n(\pi(s_1 | s_0, \alpha)), u_{n-1*}(s_1) \right\}
\end{aligned}$$

□

en utilisant des résultats similaires à 3.12

$$\begin{aligned}
& (\max_{x \in \mathcal{X}} \max \{A, B(x)\} = \max \{A, \max_{x \in \mathcal{X}} B(x)\}, \text{ et} \\
& \min_{x \in \mathcal{X}} \max \{A, B(x)\} = \max \{A, \min_{x \in \mathcal{X}} B(x)\}.
\end{aligned}$$

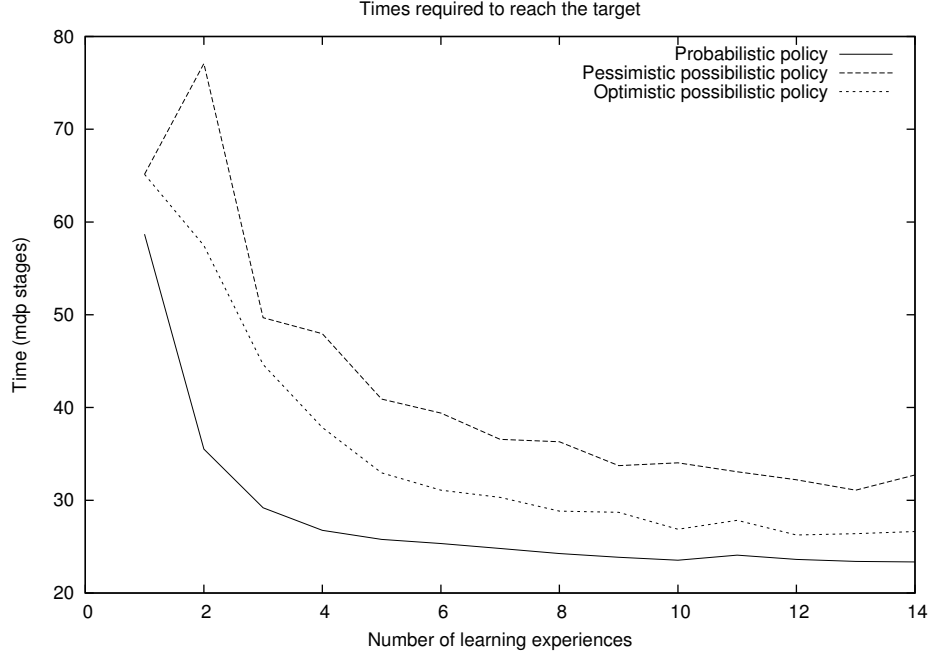
**3.2.3. Application du  $\pi$ -MDP en robotique.** Considérons un robot sur une grille finie à 2 dimensions. L'ensemble des états  $\mathcal{S}$  sera l'ensemble des positions possibles du robot (les cases de la grille). Il doit atteindre un but qui est une des cases de la grille notée  $s_G$ . Le robot peut utiliser 5 actions :  $S$  (ne pas bouger),  $T$  (avancer d'une case vers le nord),  $D$  (vers le sud),  $L$  (vers l'ouest), et  $R$  (vers l'est).  $\forall s \neq s_G, r(s) = -1$  et  $r(s_G) = 100$ . Nous notons  $T$  les probabilités de transitions utilisées pour simuler le robot. Nous suivons la procédure suivante, et obtenons la figure 3.2.3

- calcul de fréquence transition avec  $N$  expériences ( $\hat{\mathbf{p}}(s'|s, a) = \frac{1}{N} \sum_{i=1}^N \mathbb{1}_{s_i=s'}$ )
- les fréquences, qui sont des estimations des probabilités  $T$ , sont transformées en possibilités par la transformation réversible présentée section 3.1.1
- $\forall s \neq s_G, \mu(s) = 0$  et  $\mu(s_G) = 1$
- calcul de la politique par l'algorithme d'itération sur les valeurs pour les MDP, avec les fréquences  $\hat{\mathbf{p}}(s'|s, a)$
- calcul de la politique possibiliste
- simulations du robot avec les différentes stratégies
- calcul de la moyenne du temps d'atteinte de la cible

**3.3. Le modèle  $\pi$ -POMDP.** R.Sabbadin propose aussi dans [9] l'homologue possibiliste du modèle POMDP nommé  $\pi$ -POMDP.

Comme dans la section 2.2, l'agent n'a pas la connaissance de l'état du système : il ne peut le deviner qu'à travers les observations  $o \in \mathcal{O}$  qu'il reçoit à chaque étape avec possibilité  $\pi(o | s, a)$ . Nous raisonnerons alors de la même manière que dans le cas stochastique : les nouveaux états seront les croyances  $\beta \in \mathcal{S}$ , *i.e.* l'ensemble des distributions de possibilité sur  $\mathcal{S}$ . Notons déjà que  $\Sigma\mathcal{S}$  est fini (il y a  $\text{Card}(\mathcal{S})$  choix pour l'état de possibilité  $1_L$ , et  $\text{Card}(L)$  valeurs possibles pour les autres états :  $\text{Card}(\Sigma\mathcal{S}) =$

FIGURE 5. temps moyen d'atteinte de la cible en fonction de la taille de l'échantillon permettant de calculer les fréquences de transition



$$\text{Card}(\mathcal{S}) \times \text{Card}(L)^{\text{Card}(\mathcal{S})-1}.$$

Etudions la mise à jour de la croyance possibiliste : pour garder les notations de [9], nous noterons la croyance sur l'état suivant, après avoir utilisé  $a$ ,  $\mathbf{b}_a(s') = T(s' | \mathbf{b}, a) := \sum_{s \in \mathcal{S}} T(s' | s, a) \mathbf{b}(s)$ . L'équivalent possibiliste sera donc

$$\beta_a(s') = \max_{s \in \mathcal{S}} \min \{ \pi(s' | s, a), \beta(s) \} \quad (3.17)$$

La croyance sur l'observation  $o' \in \mathcal{O}$  après avoir utilisé l'action  $a \in \mathcal{A}$  est notée

$$\begin{aligned} \mathbf{b}_a(o') = O(o' | \mathbf{b}, a) &:= \sum_{s' \in \mathcal{S}} O(o' | s', a) T(s' | \mathbf{b}, a) \\ &= \sum_{(s, s') \in \mathcal{S}^2} O(o' | s', a) T(s' | s, a) \mathbf{b}(s) \end{aligned}$$

et son équivalent possibiliste

$$\beta_a(o') = \max_{(s, s') \in \mathcal{S}} \min \{ \pi(o' | s', a), \beta_a(s') \} \quad (3.18)$$



De même, la nouvelle croyance pour l'étape suivante, *i.e.* la croyance sur l'état suivant après avoir observé  $o' \in \mathcal{O}$  sera noté  $\mathbf{b}_a^{o'} := \tau(\mathbf{b}, a, o')$ . Rappelons la formule 2.10 :

$$\begin{aligned} \mathbf{b}_a^{o'}(s') &:= \mathbf{b}_{t+1}(s') = \frac{O(o' | s', a) \sum_{s \in \mathcal{S}} \mathbf{b}_t(s) T(s' | s, a)}{\sum_{s, \bar{s} \in \mathcal{S}^2} \mathbf{b}_t(s) T(\bar{s} | s, a) O(o' | \bar{s}, a)} \\ &= \frac{O(o' | s', a) T(s' | \mathbf{b}, a)}{\mathbf{b}_a(o')} \\ &= \frac{\mathbb{P}(\{o_{t+1} = o'\} \cap \{s_{t+1} = s'\} | \mathbf{b}, a)}{\mathbb{P}(o_{t+1} = o' | \mathbf{b}, a)} \end{aligned}$$

Si nous avons noté  $\Theta$  l'évènement  $\{o_{t+1} = o'\}$  et  $\Sigma = \{s_{t+1} = s'\}$ , nous cherchions finalement  $\mathbb{P}(\Sigma | \Theta)$  à partir de  $\mathbb{P}(\Sigma \cap \Theta)$  et  $\mathbb{P}(\Theta)$  (en sous-entendant le conditionnement à  $\mathbf{b}$  et  $a$  dans les notations).

Pour calculer le nouvel état de croyance possibiliste  $\beta_a^{o'}(s') = \Pi(\Sigma | \Theta)$  qui est la possibilité d'être dans l'état  $s'$  conditionnellement à l'observation  $o'$ , nous avons donc besoin de la loi  $\Pi(\Theta)$  qui est  $\beta_a(o')$ , et de la loi jointe  $\Pi(\Sigma \cap \Theta)$  que nous calculons maintenant :

$$\begin{aligned} \Pi(\Sigma \cap \Theta) &= \min \{ \pi(o' | s', a), \beta_a(s') \} \\ &= \min \left\{ \pi(o' | s', a), \max_{s \in \mathcal{S}} \min \{ \pi(s' | s, a), \beta(s) \} \right\} \\ &= \max_{s \in \mathcal{S}} \min \{ \pi(o' | s', a), \pi(s' | s, a), \beta(s) \} \end{aligned}$$

En passant, nous comprenons pourquoi le max est, tout comme la somme, qualifié d'« opérateur d'élimination ou de marginalisation » :

$$\max_{s' \in \mathcal{S}} \Pi(\Theta \cap \Sigma) = \max_{s' \in \mathcal{S}} \min \{ \pi(o' | s', a), \beta_a(s') \} = \beta_a(o') = \Pi(\Theta) \text{ par 3.18.}$$

Ainsi, d'après la définition de la possibilité conditionnelle qualitative 8,

$$\beta_a^{o'}(s') = \Pi(\Sigma | \Theta) = \begin{cases} 1_L, & \text{si } \min\{\pi(o' | s', a), \beta_a(s')\} = \beta_a(o') > 0_L \\ & \text{i.e. } \Pi(\Theta \cap \Sigma) = \Pi(\Theta) \\ \min\{\pi(o' | s', a), \beta_a(s')\} & \text{sinon.} \end{cases}$$

R.Sabbadin écrira, en partant de la formule de Bayse possibiliste :

$$\min\{\beta_a^{o'}(s'), \beta_a(o)\} = \min\{\pi(o' | s', a), \beta_a(s')\}. \quad (3.19)$$

Cela vient du fait que :

$$\begin{aligned} \beta_a^{o'}(s') &= \pi(s' | a, o', \beta), \\ \beta_a(o') &= \pi(o' | a, \beta), \\ \beta_a(s') &= \pi(s' | a, \beta), \\ \pi(o' | s', a) &= \pi(o' | s', a, \beta). \end{aligned}$$

Et après, on en déduit la valeur de  $\beta_a^{o'}(s')$  :

- Soit  $\beta_a(o') = \min \{ \pi(o' | s', a), \beta_a(s') \}$  et, par minimum de spécificité,  $\beta_a^{o'}(s') = 1_L$ .
- Soit  $\beta_a(o') > \min \{ \pi(o' | s', a), \beta_a(s') \}$  et

$$\beta_a^{o'}(s') = \min \{ \pi(o' | s', a), \beta_a(s') \}.$$

Nous souhaitons maintenant exprimer un  $\pi$ -POMDP en terme de  $\pi$ -MDP afin de retrouver facilement un algorithme de résolution : la possibilité que le système arrive dans l'état  $\beta'$  sachant qu'il était dans l'état courant  $\beta$  et que l'agent a utilisé l'action  $a$ , est

$$\pi(\beta' | \beta, a) = \max_{o' \in \mathcal{O} \text{ tq } \beta' = \beta_a^{o'}} \beta_a(o')$$

où l'on prend la convention  $\max_{x \in \emptyset} f(x) = 0_L$  i.e.  $\pi(\beta' | \beta, a) = 0_L$  si  $\forall o' \in \mathcal{O}, \beta' \neq \beta_a^{o'}$ .

La préférence d'une croyance possibiliste sera du type pessimiste :

$$\mu(\beta) = \min_{s \in \mathcal{S}} \max \{ n(\beta(s)), \mu(s) \} \quad (3.20)$$

Une préférence du type optimiste ( $\mu_{opt}(\beta) = \max_{s \in \mathcal{S}} \min \{ \beta(s), \mu(s) \}$ ) préférerait les croyances pour lesquelles il est plausible que l'agent soit dans un état préféré. 3.20 quant à lui préfère les croyances pour lesquelles il est peu plausible qu'il soit dans un état non satisfaisant.

Nous notons que la version optimiste est vraiment déraisonnable puisque une croyance ignorante ( $\forall s \in \mathcal{S}, \beta(s) = 1_L$ ) aurait la préférence la plus grande. La version pessimiste est une manière de modéliser le souhait d'améliorer la croyance afin d'arriver au but de la mission. Il pourrait être intéressant de définir et d'étudier d'autre préférences de croyance : l'ensemble de ces croyances étant fini ( $\text{Card}(\Sigma\mathcal{S}) < +\infty$ ), il est plus facile d'attribuer les préférences que l'on souhaite à des croyances possibilistes qu'à des croyances probabilistes qui vivent dans un espace infini ( $\Delta\mathcal{S}$ ).

La fonction valeur optimiste d'horizon  $n$  pour une politique  $\delta \in \Delta_n$  est

$$u_n^\delta(\beta_0) = \max_{\tau \in T_n} \min \left\{ \min_{i=0, \dots, n-1} \max_{o' \in \mathcal{O} | \beta_{i+1} = (\beta_i)_{\delta_i^{o'}}} (\beta_i)_a(o'), \mu(\beta_n) \right\}$$

où  $T_n$  désigne maintenant l'ensemble des trajectoires de croyances  $(\beta_1, \beta_2, \dots, \beta_n) \in (\Sigma\mathcal{S})^n$ .

La fonction valeur optimale pour un horizon quelconque est

$$u^*(\beta_0) = \max_{\delta \in \Delta_n} \max_{\tau \in T} \min \left\{ \min_{i=0, \dots, \#\tau-1} \max_{o' \in \mathcal{O} | \beta_{i+1} = (\beta_i)_{\delta_i^{o'}}} (\beta_i)_a(o'), \mu(\beta_{\#\tau}) \right\}$$

L'équation de Bellman (à la base de l'algorithme de résolution) qui tire la fonction valeur optimiste optimale d'horizon  $n$  ( $u_n^*(\beta_0) = \max_{\delta \in \Delta_n} u_n^\delta(\beta_0)$ ) de celle d'horizon  $n-1$ , s'en déduit :

$$\begin{aligned}
 & u_n^*(\beta_0) \\
 & := \max_{\delta \in \Delta_n} u_n^\delta(\beta_0) \\
 & = \max_{\delta \in \Delta_n} \max_{\tau \in T_n} \min \left\{ \min_{i=0, \dots, n-1} \pi(\beta_{i+1} \mid \beta_i, \delta_i(\beta_i)), \mu(\beta_n) \right\} \\
 & = \max_{(a, \delta') \in \mathcal{A}} \max_{(\beta_1, \tau') \in T_n} \min \left\{ \pi(\beta_1 \mid \beta_0, \delta_0(\beta_0)), \min \left\{ \min_{i=1, \dots, n-1} \pi(\beta_{i+1} \mid \beta_i, \delta_i(\beta_i)), \mu(\beta_n) \right\} \right\} \\
 & = \max_{a \in \mathcal{A}} \max_{\beta_1 \in \Sigma \mathcal{S}} \min \left\{ \pi(\beta_1 \mid \beta_0, \delta_0(\beta_0)), u_{n-1}^*(\beta_1) \right\} \text{ par le resultat 3.12} \\
 & = \max_{a \in \mathcal{A}} \max_{\beta_1 \in \Sigma \mathcal{S}} \min \left\{ \max_{o' \in \mathcal{O} \mid \beta_1 = (\beta_0)_a^{o'}} (\beta_0)_a(o'), u_{n-1}^*(\beta_1) \right\} \\
 & = \max_{a \in \mathcal{A}} \max_{\beta_1 \in \Sigma \mathcal{S}} \max_{o' \in \mathcal{O} \mid \beta_1 = (\beta_0)_a^{o'}} \min \left\{ (\beta_0)_a(o'), u_{n-1}^*(\beta_1) \right\} \\
 & = \max_{a \in \mathcal{A}} \max_{o' \in \mathcal{O}} \min \left\{ (\beta_0)_a(o'), u_{n-1}^* \left( (\beta_0)_a^{o'} \right) \right\}
 \end{aligned}$$

De même, la fonction valeur pessimiste optimale pour un horizon quelconque est

$$u_*(\beta_0) = \max_{\delta \in \Delta} \min_{\tau \in T} \max \left\{ n \left( \min_{i=0, \dots, \#\tau-1} \pi(\beta_{i+1} \mid \beta_i, \delta_i(\beta_i)) \right), \mu(\beta_{\#\tau}) \right\}$$

et l'équation de Bellman associée

$$u_{n*}(\beta_0) = \max_{a \in \mathcal{A}} \min_{o' \in \mathcal{O}} \max \left\{ n \left( (\beta_0)_a(o') \right), u_{n-1*} \left( (\beta_0)_a^{o'} \right) \right\}$$

où

$$u_{n*}(\beta_0) = \max_{\delta \in \Delta_n} \min_{\tau \in T_n} \max \left\{ n \left( \min_{i=0, \dots, n-1} \pi(\beta_{i+1} \mid \beta_i, \delta_i(\beta_i)) \right), \mu(\beta_n) \right\}$$

Nous détaillons les algorithmes optimiste et pessimistes en 6 et 7.

---

**Algorithm 6** optimistic  $\pi$ -POMDP iteration value algorithm
 

---

```

for  $\beta \in \Sigma \mathcal{S}$  do
     $u^*(\beta) \leftarrow \mu(\beta)$ 
     $u^c(s) \leftarrow 0$ 
     $\delta(\beta) \leftarrow \bar{a}$ 
end for
while  $u^* \neq u^c$  do
    for  $\beta \in \mathcal{S}$  do
         $u^c(\beta) \leftarrow \max_{a \in \mathcal{A}} \max_{o' \in \mathcal{O}} \min \left\{ (\beta)_a(o'), u^* \left( (\beta)_{a'}^{o'} \right) \right\}$ 
        if  $u^c(\beta) > u^*(\beta)$  then
             $\delta(\beta) \in \operatorname{argmax}_{a \in \mathcal{A}} \max_{o' \in \mathcal{O}} \min \left\{ (\beta)_a(o'), u^* \left( (\beta)_{a'}^{o'} \right) \right\}$ 
        end if
    end for
     $u^* = u^c$ 
end while
return  $u^*, \delta$ 
    
```

---



---

**Algorithm 7** pessimistic  $\pi$ -POMDP iteration value algorithm
 

---

```

for  $\beta \in \Sigma \mathcal{S}$  do
     $u_*(\beta) \leftarrow \mu(\beta)$ 
     $u_c(s) \leftarrow 0$ 
     $\delta(\beta) \leftarrow \bar{a}$ 
end for
while  $u_* \neq u_c$  do
    for  $\beta \in \mathcal{S}$  do
         $u_c(\beta) \leftarrow \max_{a \in \mathcal{A}} \min_{o' \in \mathcal{O}} \max \left\{ n \left( (\beta)_a(o') \right), u_* \left( (\beta)_{a'}^{o'} \right) \right\}$ 
        if  $u_c(\beta) > u_*(\beta)$  then
             $\delta(\beta) \in \operatorname{argmax}_{a \in \mathcal{A}} \min_{o' \in \mathcal{O}} \max \left\{ n \left( (\beta)_a(o') \right), u_* \left( (\beta)_{a'}^{o'} \right) \right\}$ 
        end if
    end for
     $u_* = u_c$ 
end while
return  $u_*, \delta$ 
    
```

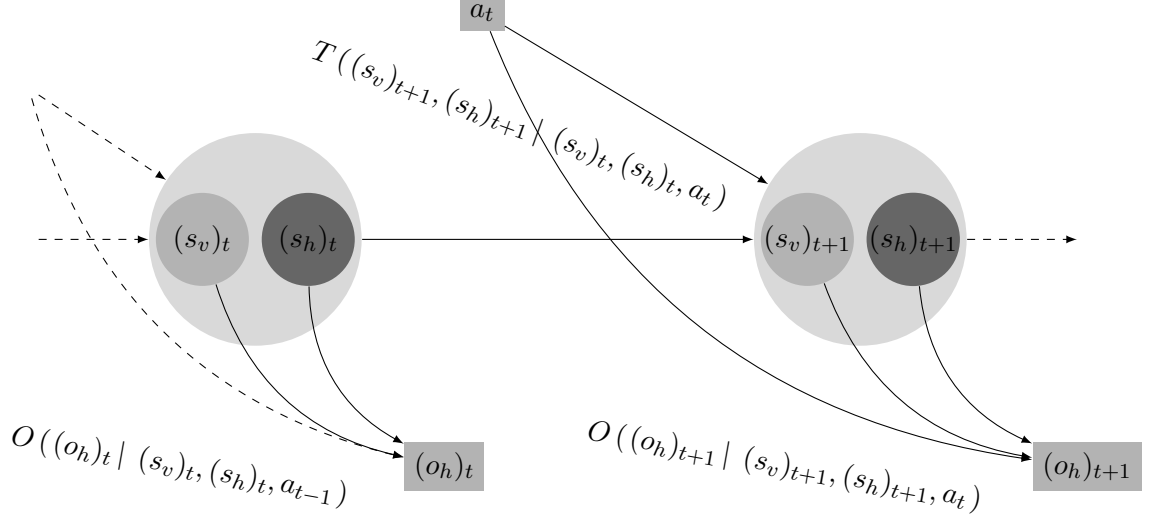
---

3.3.1. *Distinction entre variables visibles et invisibles.* Inspirés par [1], nous considérons un modèle où l'ensemble des états est le produit cartésien d'un ensemble d'*états visibles* par l'agent  $\mathcal{S}_v$ , et d'un ensemble d'*états cachés*  $\mathcal{S}_h$  :

$$\mathcal{S} = \mathcal{S}_v \times \mathcal{S}_h$$

Ce modèle s'appelle vsv-POMDP pour « visible state variables POMDP ». L'agent connaît à chaque instant  $t \in \mathbb{N}$  la composante visible  $s_v \in \mathcal{S}_v$  de l'état  $s = (s_v, s_h)$ , et observe  $o'_h \in \mathcal{O}_h$  afin de mettre à jour sa croyance sur l'état caché suivant  $s'_h \in \mathcal{S}_h$ . Pour cela, il dispose toujours des probabilités de transition  $T(s' | s, a)$  et d'observation sur  $\mathcal{O}_h$ ,  $O_h(o'_h | s', a)$ ,  $\forall s', s, a$  et  $o'_h$ . La figure 6 résume la situation. En pratique, beaucoup de situations

FIGURE 6. Modèle POMDP contenant des variables visibles



peuvent se modéliser par ce modèle hybride, mais elles sont classiquement modélisées par un simple POMDP dans lequel  $\mathcal{O} = \mathcal{S}_v \times \mathcal{O}_h$ , *i.e.* au lieu de considérer que  $s_v \in \mathcal{S}_v$  est visible par l'agent, la variable visible est intégrée à l'observation via  $o_v$  égale à  $s_v$  avec probabilité 1 : la probabilité d'observation sur  $\mathcal{O}$  est donc

$$\begin{aligned} O(o' \mid s', a) &= O((o'_v, o'_h) \mid (s'_v, s'_h), a) \\ &= \mathbf{p}(o'_h \mid (s'_v, s'_h), a) \mathbf{p}(o'_v \mid s'_v, s'_h, a, o'_h) \\ &= O_h(o'_h \mid s', a) \mathbb{1}_{\{s'_v = o'_v\}} \end{aligned}$$

Avec une modélisation POMDP classique, l'état visible initial  $(s_v)_0$  étant connu, seuls les états  $s = (s_v, s_h)$  pour lesquels  $s_v = (s_v)_0$  sont tels que  $\mathbf{b}_0(s) > 0$ . Notons  $(\mathbf{b}_h)_0(s_h) = \sum_{s_v \in \mathcal{S}} \mathbf{b}_0(s_v, s_h) = \mathbf{b}_0((s_v)_0, s_h)$  la croyance initiale sur les états cachés. Si, à l'étape  $t$ ,  $\forall s = (s_v, s_h) \in \mathcal{S}$  tels que  $s_v \neq (s_v)_t$ ,  $\mathbf{b}_t(s) = 0$ , à l'étape suivante, après avoir utilisé l'action  $a_t$ , être arrivé dans un état  $s_{t+1}$  dont la composante visible vaut  $(s_v)_{t+1} = (o_v)_{t+1}$  et avoir reçu l'observation  $o_{t+1} = ((o_v)_{t+1}, (o_h)_{t+1})$ , la croyance en  $\bar{s} = (\bar{s}_v, \bar{s}_h) \in \mathcal{S}$  devient

$$\begin{aligned}
 & \mathbf{b}_{t+1}(\bar{s}) \\
 &= \frac{O(o_{t+1} \mid \bar{s}, a_t) \sum_{s \in \mathcal{S}} T(\bar{s} \mid s, a_t) \mathbf{b}_t(s)}{\sum_{s' \in \mathcal{S}} O(o_{t+1} \mid s', a_t) \sum_{s \in \mathcal{S}} T(s' \mid s, a_t) \mathbf{b}_t(s)} \\
 &= \frac{O_h((o_h)_{t+1} \mid \bar{s}, a_t) \mathbb{1}_{\{(o_v)_{t+1} = \bar{s}_v\}} \sum_{(s_v, s_h) \in \mathcal{S}} T(\bar{s} \mid s_v, s_h, a_t) \mathbf{b}_t(s_v, s_h)}{\sum_{(s'_v, s'_h) \in \mathcal{S}} O_h((o_h)_{t+1} \mid s'_v, s'_h, a_t) \mathbb{1}_{\{(o_v)_{t+1} = s'_v\}} \sum_{(s_v, s_h) \in \mathcal{S}} T(s' \mid s_v, s_h, a_t) \mathbf{b}_t(s_v, s_h)} \\
 &= \frac{O_h((o_h)_{t+1} \mid \bar{s}, a_t) \mathbb{1}_{\{(o_v)_{t+1} = \bar{s}_v\}} \sum_{s_h \in \mathcal{S}_h} T(\bar{s} \mid (s_v)_t, s_h, a_t) (\mathbf{b}_h)_t(s_h)}{\sum_{s'_h \in \mathcal{S}_h} O_h((o_h)_{t+1} \mid (s_v)_{t+1}, s'_h, a_t) \sum_{s_h \in \mathcal{S}_h} T((s_v)_{t+1}, s'_h \mid (s_v)_t, s_h, a_t) (\mathbf{b}_h)_t(s_h)}
 \end{aligned}$$

Ainsi,  $\mathbf{b}_{t+1}(\bar{s}_v, \bar{s}_h) = 0$  si  $\bar{s}_v \neq (s_v)_{t+1}$ , et

$$\begin{aligned}
 & (\mathbf{b}_h)_{t+1}(\bar{s}_h) \\
 &= \sum_{\bar{s}_v} \mathbf{b}_{t+1}(\bar{s}_v, \bar{s}_h) \\
 &= \frac{O_h((o_h)_{t+1} \mid (s_v)_{t+1}, \bar{s}_h, a_t) \sum_{s_h \in \mathcal{S}_h} T((s_v)_{t+1}, \bar{s}_h \mid (s_v)_t, s_h, a_t) (\mathbf{b}_h)_t(s_h)}{\sum_{s'_h \in \mathcal{S}_h} O_h((o_h)_{t+1} \mid (s_v)_{t+1}, s'_h, a_t) \sum_{s_h \in \mathcal{S}_h} T((s_v)_{t+1}, s'_h \mid (s_v)_t, s_h, a_t) (\mathbf{b}_h)_t(s_h)}
 \end{aligned}$$

Il est bien expliqué dans [1] pourquoi raisonner sur  $(s_v, \mathbf{b}_h) \in \mathcal{S}_v \times \Delta \mathcal{S}_h$  et non sur  $\mathbf{b} \in \Delta \mathcal{S}$  permet la réduction de la complexité lors de la résolution. Nous explicitons maintenant la modification de l'équation de Bellman faisant apparaître  $\mathbf{b}_h$  afin d'en tirer son homologue possibiliste par la suite :

$$\begin{aligned}
 V(\mathbf{b}) &= V(s_v, \mathbf{b}_h) \\
 &= \max_{a \in \mathcal{A}} \sum_{s \in \mathcal{S}} r(s, a) \mathbf{b}(s) + \sum_{o' \in \mathcal{O}} O(o' \mid \mathbf{b}, a) V(\tau(\mathbf{b}, a, o)) \\
 &= \max_{a \in \mathcal{A}} \sum_{(\overline{s}_v, s_h) \in \mathcal{S}} r(\overline{s}_v, s_h, a) \mathbf{b}(\overline{s}_v, s_h) \\
 &\quad + \sum_{o'_v, o'_h \in \mathcal{O}} \sum_{s'_v, s'_h} O(o'_v, o'_h \mid s'_v, s'_h, a) T(s'_v, s'_h \mid \mathbf{b}, a) V(\tau(\mathbf{b}, a, o'_v, o'_h)) \\
 &= \max_{a \in \mathcal{A}} \sum_{(\overline{s}_v, s_h) \in \mathcal{S}} r(\overline{s}_v, s_h, a) \mathbf{b}(\overline{s}_v, s_h) \\
 &\quad + \sum_{o'_v, o'_h \in \mathcal{O}} \sum_{s'_v, s'_h} O(o'_v, o'_h \mid s'_v, s'_h, a) T(s'_v, s'_h \mid s_v, \mathbf{b}_h, a) V(\tau(\mathbf{b}, a, o'_v, o'_h)) \\
 &= \max_{a \in \mathcal{A}} \sum_{s_h \in \mathcal{S}} r(s_v, s_h, a) \mathbf{b}_h(s_h) \\
 &\quad + \sum_{o'_h \in \mathcal{O}_h} \sum_{s'_v, s'_h} O_h(o'_h \mid s'_v, s'_h, a) T(s'_v, s'_h \mid s_v, \mathbf{b}_h, a) V(\tau(\mathbf{b}_h, a, s'_v, o'_h, s_v))
 \end{aligned} \tag{3.21}$$

Nous avons tous les outils pour raisonner de la même manière en théorie possibiliste : nous disposons des distributions de possibilité de transition

$$\pi(s' \mid s, a) = \pi((s'_v, s'_h) \mid (s_v, s_h), a)$$

ainsi que des distributions de possibilité d'observation sur  $\mathcal{O}_h$ ,

$$\pi_h(o'_h \mid s', a) = \pi_h(o'_h \mid (s'_v, s'_h), a)$$

Tout comme en théorie probabiliste, une manière naïve de modéliser une situation où une partie de l'état courant est visible, serait de considérer que lorsque le système est dans un état dont la composante visible est  $s'_v$ ,  $o'_v = s'_v$  est observée avec nécessité 1 : nous poserions  $\mathcal{O} = \mathcal{S}_v \times \mathcal{O}_h$  ainsi que la possibilité d'observation sur  $\mathcal{O}$ ,

$$\begin{aligned}
 \pi(o' \mid s', a) &= \min \{ \pi(o'_h \mid (s'_v, s'_h), a), \pi(o'_v \mid s'_v, s'_h, a, o'_h) \} \\
 &= \min \{ \pi_h(o'_h \mid s'_v, s'_h, a), \mathbb{1}_{\{s'_v, o'_v\}} \}
 \end{aligned}$$

Au contraire, suivons le même raisonnement que pour les vsv-POMDP. Pour cela, nous posons les distributions de possibilité suivantes

- Nous notons  $\beta_h(s_h)$  la croyance possibiliste sur l'état caché courant  $s_h \in \mathcal{S}_h$ ,
- $\beta_a^{s_v}(s'_v, s'_h) = \max_{s \in \mathcal{S}_h} \min \{ \pi(s'_v, s'_h \mid s_v, s, a), \beta_h(s) \}$ ,
- $\beta_{h,a}^{s_v, s'_v}(o'_h) = \max_{s' \in \mathcal{S}_h} \min \{ \pi_h(o'_h \mid s'_v, s', a), \beta_a^{s_v}(s'_v, s') \}$ .
- De même, la mise à jour de la croyance sur les états cachés est  $(\beta_h)_{t+1}(s'_h) = \tau_h((\beta_h)_t, (s_v)_t, (s_v)_{t+1}, (o_h)_{t+1}, a_t)$

$$= \begin{cases} 1_L \text{ si } \beta_{h,a}^{(s_v)^t, (s_v)^{t+1}}((o_h)_{t+1}) = \min \left\{ \pi((o_h)_{t+1} \mid (s_v)_{t+1}, s'_h, a), \beta_a^{(s_v)^t}((s_v)_{t+1}, s'_h) \right\} > 0 \\ \min \left\{ \pi((o_h)_{t+1} \mid (s_v)_{t+1}, s'_h, a), \beta_a^{(s_v)^t}((s_v)_{t+1}, s'_h) \right\} \text{ sinon} \end{cases}$$

Nous explicitons maintenant l'homologue possibiliste de 3.21, *i.e.* l'équation de Bellman possibiliste optimiste :

$$u_{n+1}^*(s_v, \beta_h) = \max_{a \in \mathcal{A}} \max_{o'_h \in \mathcal{O}_h} \max_{s'_v \in \mathcal{S}_v} \min \left\{ \beta_{h,a}^{s_v, s'_v}(o'_h), u_n^*(s'_v, \tau_h(\beta_h, s_v, s'_v, o'_h, a)) \right\} \quad (3.22)$$

et l'équation de Bellman pessimiste :

$$u_{*n+1}(s_v, \beta_h) = \max_{a \in \mathcal{A}} \min_{s'_v \in \mathcal{S}_v} \min_{o'_h \in \mathcal{O}_h} \max \left\{ n \left( \beta_{h,a}^{s_v, s'_v}(o'_h) \right), u_n^*(s'_v, \tau_h(\beta_h, s_v, s'_v, o'_h, a)) \right\} \quad (3.23)$$

**3.3.2. Application à l'identification de cibles.** Considérons un robot sur une grille finie à 2 dimensions. Sur cette grille se trouvent deux cibles fixes,  $A$  et  $B$ . Le robot sait où il est à chaque instant, ainsi que la position des deux cibles, cependant, il ne connaît pas la nature des cibles ( $A$  ou  $B$ ). Sa mission est d'identifier la cible  $A$  à l'aide des observations qu'il reçoit à chaque étapes (plus ou moins bien selon la distance qui le sépare de chaque cible), et de s'y rendre.

L'ensemble des états visibles  $\mathcal{S}_v$  sera l'ensemble des positions possibles du robot (les cases de la grille). L'ensemble des états cachés  $\mathcal{S}_h$  sera la nature des cibles :  $s_{h,1}$  est l'état du système lorsque la cible en position 1 (notée  $s_{v,1}$ ) est  $A$  et celle en position 2 ( $s_{v,2}$ ) est  $B$ , et  $s_{h,2}$  est la situation inverse. Le robot peut utiliser 5 actions :  $S$  (ne pas bouger),  $T$  (avancer d'une case vers le nord),  $D$  (vers le sud),  $L$  (vers l'ouest), et  $R$  (vers l'est). Nous pouvons imaginer qu'en pratique les cibles peuvent se reconnaître visuellement, et que les deux cibles sont prises en photo à chaque étape. Un algorithme de traitement d'image renvoie alors au robot, pour chaque photo, la nature estimée de la cible avec une erreur liée à la qualité des photos, et donc à la distance des cibles par rapport au robot : c'est l'observation. Elle est notée  $o_{h,AA}$  si les deux photos sont estimées être  $A$ ,  $o_{h,AB}$  si l'algorithme estime que la cible en position 1 est  $A$  et l'autre  $B$ , et nous définissons dans cette logique  $o_{hBA}$  et enfin  $o_{hBB}$ . Si le robot est dans l'état  $s_v$  (nous notons  $x_{s_v}$  son abscisse sur la grille, et  $y_{s_v}$  son ordonnée),  $\forall i \in \{1, 2\}$  la probabilité que l'algorithme de traitement d'image estime bien la nature de la cible qui se trouve en position  $i$  est égale à  $\kappa_i(s_v) = \frac{1}{2} + \frac{1}{2} \exp \left( -\frac{\sqrt{(x_{s_v,i} - x_{s_v})^2 + (y_{s_v,i} - y_{s_v})^2}}{\lambda} \right)$  où  $\lambda > 0$  permet d'ajuster la perception (plus  $\lambda$  est grand, meilleur est l'algorithme).



La probabilité que l'algorithme se trompe vaut donc  $1 - \kappa_i(s_v)$ .

Les distributions de probabilité suivantes modéliseront notre problème :

$$\begin{aligned}
 \bullet T_v(s'_v | s_v, a) &= \begin{cases} 1 - 4\varepsilon & \text{si } s'_v \text{ est la case collée} \\ & \text{à } s_v \text{ dans la direction } a, \\ \varepsilon & \text{si } s'_v \text{ est l'une des 4 cases légèrement} \\ & \text{à droite ou à gauche de la direction } a \end{cases} \\
 \bullet O_h(o_h | s_{h,1}, s_v) &= \begin{cases} \kappa_1(s_v)(1 - \kappa_2(s_v)) & \text{si } o_h = o_{hAA} \\ \kappa_1(s_v)\kappa_2(s_v) & \text{si } o_h = o_{hAB} \\ (1 - \kappa_1(s_v))(1 - \kappa_2(s_v)) & \text{si } o_h = o_{hBA} \\ (1 - \kappa_1(s_v))\kappa_2(s_v) & \text{si } o_h = o_{hBB} \end{cases} \\
 \bullet O_h(o_h | s_{h,2}, s_v) &= \begin{cases} (1 - \kappa_1(s_v))\kappa_2(s_v) & \text{si } o_h = o_{hAA} \\ (1 - \kappa_1(s_v))(1 - \kappa_2(s_v)) & \text{si } o_h = o_{hAB} \\ \kappa_1(s_v)\kappa_2(s_v) & \text{si } o_h = o_{hBA} \\ \kappa_1(s_v)(1 - \kappa_2(s_v)) & \text{si } o_h = o_{hBB} \end{cases}
 \end{aligned}$$

Dans notre exemple, les états cachés (la nature des cibles) ne change jamais au cours du processus :  $T_h(s'_h | s_h) = \mathbb{1}_{\{s'_h = s_h\}}$ .

Le remarque précédente, ainsi que le fait que  $T(s'_v, s'_h | s_v, s_h, a) = T_v(s'_v | s_v, a) T_h(s'_h | s_h, a)$ , permet la simplification la mise à jour de la croyance :

$$\begin{aligned}
 & (\beta_h(\bar{s}_h))_{t+1} \\
 & O_h((o_h)_{t+1} | (s_v)_{t+1}, \bar{s}_h, a_t) \sum_{s_h \in \mathcal{S}_h} T_h(\bar{s}_h | s_h, a_t) (\mathbf{b}_h)_t(s_h) \\
 &= \frac{\sum_{s'_h \in \mathcal{S}_h} O_h((o_h)_{t+1} | (s_v)_{t+1}, s'_h, a_t) \sum_{s_h \in \mathcal{S}_h} T_h(\bar{s}_h | s_h, a_t) (\mathbf{b}_h)_t(s_h)}{\sum_{s'_h \in \mathcal{S}_h} O_h((o_h)_{t+1} | (s_v)_{t+1}, s'_h, a_t) (\mathbf{b}_h)_t(\bar{s}_h)} \\
 &= \frac{O_h((o_h)_{t+1} | (s_v)_{t+1}, \bar{s}_h, a_t) (\mathbf{b}_h)_t(\bar{s}_h)}{\sum_{s'_h \in \mathcal{S}_h} O_h((o_h)_{t+1} | (s_v)_{t+1}, s'_h, a_t) (\mathbf{b}_h)_t(\bar{s}_h)} \quad (3.24)
 \end{aligned}$$

Nous utilisons la transformation présentée section 3.1.1 pour obtenir des distributions de possibilités. Ainsi, nous obtenons les distributions de possibilité  $\pi_h(o'_h | s'_v, s'_h, a)$ ,  $\pi_v(s'_v | s_v, a)$  et  $\pi_h(s'_h | s_h, a)$  qui sont telles que  $\pi_h(s'_h | s_h, a) = 0$  si  $s_h \neq s'_h$  (1 sinon), et

$$\pi(s'_v, s'_h | s_v, s_h, a) = \min(\pi_v(s'_v | s_v, a), \pi_h(s'_h | s_h, a))$$

La simplification 3.24 nous permet d'écrire la mise à jour de la croyance sur les états cachés : posons

$$\begin{aligned}
 \bullet \pi_h(o'_h, s'_h | s'_v, a, \beta_h) &= \min\{\pi_h(o'_h | s'_v, s'_h, a), \beta_h(s'_h)\}, \text{ et} \\
 \bullet \beta_{h,a}^{s'_v}(o'_h) &= \max_{s'_h \in \mathcal{S}_h} \pi_h(o'_h, s'_h | s'_v, a, \beta_h). \\
 (\beta_h)_{t+1}(\bar{s}_h) &= \tau_h((\beta_h)_t, (o_h)_{t+1}, (s_v)_{t+1}, a_t) (\bar{s}_h)
 \end{aligned}$$

$$= \begin{cases} 1_L & \text{si } (\beta_h)_{t+1}(\bar{s}_h) = \pi_h((o_h)_{t+1}, \bar{s}_h | (s_v)_{t+1}, a_t, (\beta_h)_t) > 0 \\ \pi_h((o_h)_{t+1}, \bar{s}_h | (s_v)_{t+1}, a_t, (\beta_h)_t) & \text{sinon} \end{cases}$$

Nous calculons les distributions utilisées dans les formules de la section précédente,

$$\begin{aligned}
 \beta_a^{s_v}(s'_v, s'_h) &= \max_{s \in \mathcal{S}_h} \min \left\{ \min \left( \pi_v(s'_v \mid s_v, a), \pi_h(s'_h \mid s, a) \right), \beta_h(s) \right\} \\
 &= \min \left\{ \pi_v(s'_v \mid s_v, a), \max_{s \in \mathcal{S}_h} \min \left( \pi_h(s'_h \mid s, a), \beta_h(s) \right) \right\} \\
 &= \min \left\{ \pi_v(s'_v \mid s_v, a), \beta_h(s'_h) \right\} \\
 \beta_{h,a}^{s_v, s'_v}(o'_h) &= \max_{s' \in \mathcal{S}_h} \min \left\{ \pi_h(o'_h \mid s'_v, s', a), \min \left\{ \pi_v(s'_v \mid s_v, a), \beta_h(s') \right\} \right\} \\
 &= \min \left\{ \max_{s' \in \mathcal{S}_h} \min \left\{ \pi_h(o'_h \mid s'_v, s', a), \beta_h(s') \right\}, \pi_v(s'_v \mid s_v, a) \right\} \\
 &= \min \left\{ \beta_{h,a}^{s'_v}(o'_h), \pi_v(s'_v \mid s_v, a) \right\}
 \end{aligned}$$

Enfin, l'équation de Bellman possibiliste optimiste devient :

$$\begin{aligned}
 u_{n+1}^*(s_v, \beta_h) &= \max_{a \in \mathcal{A}} \max_{o'_h \in \mathcal{O}_h} \max_{s'_v \in \mathcal{S}_v} \min \left\{ \beta_{h,a}^{s'_v}(o'_h), \pi_v(s'_v \mid s_v, a), u_n^*(s'_v, \tau_h(\beta_h, s'_v, o'_h, a)) \right\} \\
 &= \max_{a \in \mathcal{A}} \max_{s'_v \in \mathcal{S}_v} \min \left\{ \pi_v(s'_v \mid s_v, a), \max_{o'_h \in \mathcal{O}_h} \min \left\{ \beta_{h,a}^{s'_v}(o'_h), u_n^*(s'_v, \tau_h(\beta_h, s'_v, o'_h, a)) \right\} \right\}
 \end{aligned}$$

et la version pessimiste est

$$\max_{a \in \mathcal{A}} \min_{s'_v \in \mathcal{S}_v} \max \left\{ n \left( \pi_v(s'_v \mid s_v, a) \right), \min_{o'_h \in \mathcal{O}_h} \max \left\{ n \left( \beta_{h,a}^{s'_v}(o'_h) \right), u_n^*(s'_v, \tau_h(\beta_h, s'_v, o'_h, a)) \right\} \right\}$$

**Remarque :** Une étude plus poussée des intégrales de Sugeno nous permettrait d'envisager de choisir séparément les variantes optimistes et pessimistes pour l'incertitude liée à la transition, et celle liée à l'observation.

En modifiant légèrement les algorithmes d'itération sur les valeurs pour les  $\pi$ -POMDP 6 et 7 (la fonction valeur est maintenant définie sur  $\mathcal{S}_v \times \Sigma \mathcal{S}_h$ , et l'équation de Bellman simplifiée), chaque itération ne calcule plus que  $\text{Card} \{ \mathcal{S}_v \times \Sigma \mathcal{S}_h \} = \text{Card} \{ \text{cases de la grille} \} \times \text{Card} \{ L \}^2$  points de la fonction valeur au lieu de  $\text{Card} \{ \Sigma(\mathcal{S}_v \times \mathcal{S}_h) \} = \text{Card} \{ L \}^{2 \text{Card} \{ \text{cases de la grille} \} }$ .

#### 4. CONCLUSION

Dans ce rapport, les modèles possibilistes assez proches des modèles MDP et POMDP ont été étudiés : la complexité de résolution est croissante avec  $L$ , le nombre de degrés de possibilité utilisés. Nous avons apporté une preuve à l'homologue possibiliste de l'algorithme d'itération sur les valeurs, et utilisé l'astuce de séparation de l'état en état visible et état caché avec le modèle  $\pi$ -POMDP. Nous avons aussi pu constater que, l'espace des états de croyances étant fini pour le modèle  $\pi$ -POMDP, des récompenses peuvent

être directement attribuées aux croyances. Mettre en évidence les qualités de la politique pessimiste pourrait se faire à travers un problème où le système est à la fois mal connu et dangereux.

Enfin, le modèle MDPST qui utilise la théorie de Dempster-Shafer, les intégrales de Sugeno ainsi que la théorie des probabilités imprécises seront des domaines à explorer afin de trouver des solutions aux problèmes liés au modèle POMDP.

## RÉFÉRENCES

- [1] M. Araya-López, V. Thomas, O. Buffet, and F. Charpillet. Des pomdps avec des variables d'état visibles. In *Actes des cinquièmes journées francophones planification, décision, apprentissage pour la conduite de systèmes (JFPDA'10)*, 2010.
- [2] Richard Bellman. A Markovian Decision Process. *Indiana Univ. Math. J.*, 6 :679–684, 1957.
- [3] D. P. Bertsekas. *Dynamic Programming and Optimal Control, Vol. I, 2nd Ed.* Athena Scientific, Belmont, MA, 2001.
- [4] Caroline Ponzoni Carvalho Chanele, Jean-Loup Farges, Florent Teichteil-Königsbuch, and Guillaume Infantes. Pomdp solving : what rewards do you really expect at execution ? In Thomas Ågotnes, editor, *STAIRS*, volume 222 of *Frontiers in Artificial Intelligence and Applications*, pages 50–62. IOS Press, 2010.
- [5] Didier Dubois and Henri Prade. Unfair coins and necessity measures : a possibilistic interpretation of histograms. *Fuzzy Sets and Systems*, 10(1) :15–20, 1983. DP034.
- [6] Didier Dubois and Henri Prade. *Possibility Theory : An Approach to Computerized Processing of Uncertainty (traduction revue et augmentée de "Théorie des Possibilités")*. Plenum Press, New York, 1988.
- [7] Didier Dubois, Henri Prade, and Sandra Sandri. On possibility/probability transformations. In R Lowen and M Roubens, editors, *Fuzzy Logic - State of the Art*, pages 103–112. Kluwer Academic Publishers, [http ://www.wkap.nl/](http://www.wkap.nl/), 1993.
- [8] Cédric Pralet, Cédric Verfaillie, and Thomas Schiex. Un cadre graphique et algébrique pour les problèmes de décision incluant incertitudes, faisabilités et utilités. *Revue d'Intelligence Artificielle*, 21(3) :419–448, 2007.
- [9] Régis Sabbadin. A possibilistic model for qualitative sequential decision problems under uncertainty in partially observable environments . In *15th Conference on Uncertainty in Artificial Intelligence (UAI99)* , Stockholm, 30/07/99-01/08/99, pages 567–564, San Francisco, juillet 1999. Morgan Kaufmann.
- [10] Régis Sabbadin, Hélène Fargier, and Jérôme Lang. Towards qualitative approaches to multi-stage decision making, 1998.
- [11] Richard D. Smallwood and Edward J. Sondik. *The Optimal Control of Partially Observable Markov Processes Over a Finite Horizon*, volume 21. INFORMS, 1973.