



**UNIVERSITE SULTAN MOULAY SLIMANE**  
**Faculté des Sciences et Techniques**  
**Béni-Mellal**



**Centre d'Études Doctorales : Sciences et Techniques**

**Formation Doctorale : Mathématiques et Physiques Appliquées**

**THÈSE**

Présentée par

**SEMMOURI ABDELLATIF**

Pour l'obtention du grade de

**DOCTEUR**

**Discipline : Mathématiques**

**Spécialité : Mathématiques**

---



---

**Contribution à l'élimination d'actions non optimales pour les processus Décisionnels de Markov actualisés.**

---



---

**Soutenue le Mardi 08 Mars 2022 à 10h devant la commission d'examen:**

Pr. Mohamed FAKIR	Professeur, Université Sultan Moulay Slimane, F.S.T Béni-Mellal, Maroc	Président
Pr. Mohamed EL FATINI	Professeur, Université Ibn Tofail, Kénitra, Maroc	Rapporteur
Pr. Belaid BOUIKHALENE	Professeur, Université Sultan Moulay Slimane, F.P. Béni-Mellal, Maroc.	Rapporteur
Pr. Cherki DAOUI	Professeur, Université Sultan Moulay Slimane, F.S.T Béni-Mellal, Maroc	Rapporteur
Pr. Rachid EL AYACHI	Professeur, Université Sultan Moulay Slimane, F.S.T. Béni-Mellal, Maroc.	Examineur
Pr. Mostafa JOURHMANE	Professeur, Université Sultan Moulay Slimane, F.S.T. Béni-Mellal, Maroc	Directeur de Thèse

## REMERCIEMENTS

Au terme de ce travail, je tiens à remercier tout d'abord mon directeur de thèse, Professeur Dr. Mostafa Jourhmane, pour la confiance qu'il m'a accordée en me proposant d'effectuer ce travail. Sans qui cette thèse n'aurait certainement pas eu lieu. Je le remercie également pour sa capacité à guider mes recherches tout en me laissant la liberté nécessaire. Il a eu une influence significative sur ma philosophie et mon style de recherche grâce à son expertise en mathématiques. Je tiens à lui exprimer toute ma gratitude pour ses conseils, sa patience et son soutien sans fin tout au long de mes travaux de recherche. Je souhaiterais également exprimer ma gratitude envers ceux qui ont accepté d'évaluer mon travail : les Rapporteurs Prof Dr. Mohamed EL FATINI, Prof Dr. Belaid BOUIKHALENE, Prof Dr. Cherki DAOUI, ainsi que le Président du Jury Prof Dr. Mohamed FAKIR et l'Examineur Prof Dr. Rachid EL AYACHI.

L'aboutissement de cette thèse aurait été plus difficile sans le soutien bienveillant et chaleureux des membres de ma famille. Je garde enfin mes remerciements amoureux pour ma femme Aicha, mon fils Mohammed et ma fille Intissar pour leur soutien et leur encouragement au quotidien durant ces années de thèse.

Enfin, grand merci à tous ceux qui, par leurs travaux, leurs idées, leurs présentations, leurs collaborations ou leurs relectures, ont participé de près ou de loin à la réalisation de ce travail.

## ABSTRACT

*This research work focuses on the problems of planning under uncertainty. These are Markov decision processes (MDPs) with discrete time finite state and action spaces. They represent an elegant framework for modeling the problems of sequential decision making under uncertainty. In this context, the system is characterized by a stochastic process with values in a set of states which evolves over time. The evolution of these state variables depends in part on decisions. The ultimate goal of any decision maker is to find an optimal policy as a function that indicates which action should be selected when the system is in one of its possible states. For this regard, there are two approaches :*

*Firstly, there are problems that are ruled by chance. In this case, we apply the classical laws of probability theory which presents a powerful and richer tool to initiate such a situation. Computing optimal solutions requires either the dynamic programming tool or the linear programming machinery. This allows establishing techniques for reducing the computational complexity efforts. Among these tools, procedures for eliminating suboptimal actions are steadily increasing in the literature of MDPs. Secondly, we encounter problems in the real life when some parameters of which are imprecise. These problems are characterized by the fuzzy aspect. Therefore, fuzzy theory has successfully solved these kinds of problems.*

*Finally, we conclude our contribution by studying a fuzzy electronic circuit. Our primary motivation comes from the fact of building a crisp (ordinary) system from the proposed fuzzy system which can be solved via iterative methods.*

**Keywords :** Markov decision problems, Stochastic optimization, Non-optimality test, Fuzzy decisions, Fuzzy electronic circuit.

*Ce travail de recherche s'intéresse aux problèmes de planification dans l'incertain. Il s'agit des processus décisionnels de Markov (PDMs) à espaces d'état et d'action finis et en temps discret. Ils représentent un cadre qui excite la curiosité des chercheurs intéressés pour modéliser les problèmes de la prise de décision séquentielle dans l'incertitude. Dans ce contexte, le système est caractérisé par un processus stochastique à valeurs dans un ensemble d'états qui évolue avec le temps. L'évolution de ces variables d'état dépend en partie des décisions. Le but ultime de tout décideur est de trouver une stratégie optimale en tant qu'une fonction. Elle indique quelle action doit être sélectionnée lorsque le système est dans l'un de ses états possibles.*

*On distingue deux approches :*

*D'une part, il y a des problèmes qui sont régis par le hasard. Dans ce cas, on applique les lois classiques de la théorie des probabilités qui présente un outil mathématique pour entamer une telle situation. La détermination de la solution nécessite soit l'outil de la programmation dynamique soit la machinerie de la programmation linéaire. Ce qui permet d'établir des techniques pour réduire les efforts de la complexité de calcul. Parmi ces moyens, les procédures d'élimination des actions non optimales ne cessent de s'accroître dans la littérature des PDMs.*

*D'autre part, on rencontre dans la vie réelle des problèmes dont certains paramètres sont imprécis. Ces problèmes sont caractérisés par l'aspect flou. Ainsi, la théorie floue a résolu ces types de problèmes.*

*En fin, on conclut notre contribution par l'étude d'un circuit électronique flou. Notre*

*motivation première provient du fait consistant à élaborer un système ordinaire à partir du système flou proposé qui peut être résolu via des méthodes itératives.*

**Mots-clefs :** Problèmes décisionnels de Markov , Optimisation stochastique , Test de non optimalité , Décisions floues , Circuit électronique flou.

## TABLE DES FIGURES

1.1	Le mathématicien russe Andreï Markov (1856-1922) . . . . .	8
1.2	Marche aléatoire symétrique . . . . .	8
1.3	Arbre de la descendance de $Z_0$ . . . . .	9
1.4	Graphe probabiliste . . . . .	11
2.1	Graphe probabiliste de l'exemple 2.2.4.1 . . . . .	29
2.2	Facteur d'actualisation et actions éliminées . . . . .	34
2.3	Modèle d'un PDM . . . . .	35
2.4	Complexité . . . . .	47
2.5	Le graphe de l'exemple 2.3.4.1 . . . . .	51
2.6	Variations graphique des nombres $\underline{n}_\gamma$ et $\bar{n}_\gamma$ . . . . .	53
2.7	Le graphe de l'exemple 2.3.4.2 . . . . .	56
2.8	Variations graphique des nombres $n_\gamma(1)$ et $n_\gamma(2)$ . . . . .	57
2.9	Variations graphique des nombres $n_\gamma(1)$ et $n_\gamma(2)$ . . . . .	58
2.10	Graphe de l'exemple 2.3.4.3 . . . . .	63
2.11	Comparaison du test <a href="#">2.44</a> avec celui de MacQueen . . . . .	64
2.12	Comparaison du test <a href="#">2.44</a> avec celui Porteus . . . . .	65
2.13	Comparaison du test <a href="#">2.44</a> avec le test De SJE1 . . . . .	65
2.14	Modèle du processus décisionnel de Markov . . . . .	69
2.15	Graphe probabiliste de l'exemple 2.3.4.4 . . . . .	71
2.16	Variation de $n_1(\gamma)$ et $n_2(\gamma)$ . . . . .	72

3.1	Sous-ensemble flou . . . . .	77
3.2	Intersection & réunion floues . . . . .	78
3.3	Complémentaire flou . . . . .	79
3.4	$\alpha$ -coupe d'un sous-ensemble flou . . . . .	80
3.5	Trapezoidal Fuzzy Number . . . . .	84
3.6	Nombre flou triangulaire . . . . .	84
3.7	Parametric form of fuzzy set . . . . .	85
3.8	Fuzzification . . . . .	86
3.9	Le modèle du PDM flou . . . . .	87
3.10	Fonctions d'appartenance des coûts flous . . . . .	91
3.11	Représentation graphique de l'exemple 3.3.2.1 . . . . .	92
3.12	Les fonctions d'appartenance du vecteur valeur optimal . . . . .	93
3.13	Représentation graphique de l'exemple 3.3.2.2 . . . . .	93
3.14	Fonctions d'appartenance des coûts flous . . . . .	94
3.15	Fonctions d'appartenance des composantes floues du vecteur valeur optimal. . . . .	95
4.1	Circuit flou . . . . .	107
4.2	La solution exacte . . . . .	108
4.3	La solution approchée . . . . .	108



## LISTE DES TABLEAUX

2.1	Application de l'algorithme $\underline{2}$ (FIA) . . . . .	29
2.2	Valeurs de $B^t(.,.)$ par l'algorithme IFIA . . . . .	33
2.3	Identification des actions non optimales . . . . .	52
2.4	Variations des indicateurs $\underline{n}_\gamma$ et $\bar{n}_\gamma$ . . . . .	53
2.5	Identification des actions non optimales . . . . .	56
2.6	Variations des indicateurs $\underline{n}_\gamma$ et $\bar{n}_\gamma$ . . . . .	57
2.7	Values of $T^n(.,.)$ by IVIA . . . . .	63
2.8	Actions éliminées par la fonction de test $S^n$ . . . . .	71

# TABLE DES MATIÈRES

<b>Table des figures</b>	<b>vi</b>
<b>Liste des tableaux</b>	<b>viii</b>
<b>Introduction</b> . . . . .	<b>1</b>
0.1 Choix et intérêt du sujet . . . . .	1
0.2 Problématique du travail . . . . .	2
0.3 Hypothèses du travail . . . . .	2
0.4 État de la question . . . . .	3
0.5 Délimitation du sujet et objectif global . . . . .	3
0.6 Méthodes et techniques utilisées . . . . .	5
0.7 Structure du travail . . . . .	5
<b>1 Chaînes de Markov</b>	<b>7</b>
1.1 Introduction . . . . .	7
1.2 Chaînes de Markov discrètes . . . . .	7
1.3 Classement des états : Récurrence et transience . . . . .	15
1.3.1 Préliminaires . . . . .	15
1.3.2 Considérations algébriques et théoriques des graphes probabi- listes . . . . .	18
1.4 Conclusion du chapitre . . . . .	19

<b>2</b>	<b>Processus décisionnels de Markov</b>	<b>21</b>
2.1	Introduction . . . . .	21
2.2	Modèles sur horizon fini . . . . .	22
2.2.1	Notions préliminaires . . . . .	22
2.2.2	Description du modèle et critère d'optimalité . . . . .	24
2.2.3	Approche de la programmation dynamique . . . . .	25
2.2.4	Algorithme d'induction en arrière. . . . .	26
2.2.5	Procédures d'élimination des actions non optimales . . . . .	30
2.3	Modèles sur horizon infini . . . . .	35
2.3.1	Description du modèle et critère d'optimalité à horizon infini .	36
2.3.2	Résolution d'un PDM par l'approche de la programmation dynamique . . . . .	38
2.3.3	Résolution d'un PDM par l'approche de la programmation linéaire . . . . .	44
2.3.4	Procédures d'élimination des actions non optimales . . . . .	47
2.4	Conclusion du chapitre . . . . .	72
<b>3</b>	<b>Le critère du coût flou <math>\gamma</math>-actualisé</b>	<b>74</b>
3.1	Introduction . . . . .	74
3.2	Généralités sur les ensembles flous . . . . .	76
3.2.1	Concepts de base de la théorie floue . . . . .	76
3.3	Le modèle du coût flou actualisé . . . . .	86
3.3.1	Notions préliminaires . . . . .	86
3.3.2	Résultats . . . . .	90
3.4	Conclusion du chapitre . . . . .	95
<b>4</b>	<b>Systèmes d'équations linéaires floues</b>	<b>97</b>
4.1	Introduction . . . . .	97
4.2	Généralités les méthodes de résolution des systèmes linéaires . . . . .	99
4.2.1	Les méthodes directes . . . . .	99
4.2.2	Les méthodes itératives . . . . .	102
4.3	Résolution numérique des systèmes d'équations linéaires floues . . . . .	105

4.4 Conclusion du chapitre . . . . .	109
<b>Conclusion générale . . . . .</b>	<b>110</b>
<b>Bibliographie</b>	<b>114</b>

## 0.1 Choix et intérêt du sujet

*Les processus de Markov, qui font partie de la théorie générale des processus stochastiques, ont énormément évolué au cours de la seconde moitié du dernier siècle. En particulier, les chaînes de Markov ont été introduites par le mathématicien russe A. Markov au début du 20<sup>e</sup> siècle. Plus tard, des processus de décision de Markov (PDMs) ont été développés et sont devenus un cadre mathématique pour la modélisation grâce à la programmation dynamique stochastique.*

*La théorie des chaînes de Markov contrôlées (ou processus de décision de Markov) avait, fait ses premiers pas, et de manière indépendante, par des ingénieurs en contrôle et des chercheurs en Recherche opérationnelle. Les chercheurs spécialisés aux processus de Markov et les chaînes de Markov contrôlées étaient depuis longtemps conscients des synergies entre ces deux domaines.*

*Les processus décisionnels de Markov fournissent un cadre mathématique pour modéliser la prise de décision dans des situations où les résultats sont en partie aléatoires et en partie contrôlés par le décideur. Les PDMs sont utilisés dans divers domaines, notamment la robotique, le contrôle automatisé, la planification, l'économie et la fabrication, la finance et l'intelligence artificielle, etc.*

## 0.2 Problématique du travail

*La programmation dynamique repose entre autre sur deux algorithmes qui ont été appliqués pour la résolution des PDMs ; ce sont l'itération de la valeur et l'itération de la politique. Ces méthodes de programmation dynamique effectuent des balayages dans l'espace d'état et effectuent une opération de sauvegarde complète sur chaque état. Chaque sauvegarde met à jour la valeur d'un seul état en fonction des valeurs de tous les états successeurs possibles et de la probabilité de se retrouver dans cet état. En raison de l'exigence de ces méthodes de programmation dynamique pour effectuer des balayages complets de l'espace d'état, elles demeurent moins pratiques pour les problèmes de grande taille. Mais, en les comparant à d'autres méthodes de résolution des PDMs, on remarque qu'elles sont en fait assez efficaces point de vue rapidité et qu'elles garantissent de trouver une politique optimale en temps polynomial. Ils sont également mieux adaptés à la gestion de grands espaces d'états que les méthodes concurrentes telles que la recherche directe et la programmation linéaire. Une autre qualité intéressante des algorithmes de programmation dynamique est qu'ils ne nécessitent pas que les états soient sauvegardés dans un ordre particulier ou aussi souvent afin de converger. Ils offrent ainsi la possibilité d'utiliser différentes approches pour balayer l'espace des états. Néanmoins, si les tailles des espaces d'états et d'actions sont assez grandes, le problème de complexité en temps et en espace mémoire se pose avec insistance. De plus, si certains paramètres du système sont vagues ou imprécis on sera obligé de faire appel à un autre cadre mathématique qui traite les problèmes d'incertitude provenant de causes non aléatoires.*

*Notre dissertation doctorale, est de contribuer à répondre et rechercher des solutions aux problèmes posés.*

## 0.3 Hypothèses du travail

*On considère d'abord des problèmes décisionnels de Markov à coûts actualisés (ou pondérés) avec des espaces d'états  $S$  et actions  $A$  finis. On suppose, ensuite que les coûts, les probabilités de transitions entre les états et les matrices réelles mises en jeu sont stationnaire en temps. Ces hypothèses principales assurent l'existence et*

*l'unicité des points fixes pour les applications contractantes de l'espace de Banach  $\mathbb{R}^{|S|}$  muni de la norme infinie  $\|\cdot\|_\infty$ .*

## 0.4 État de la question

*Dans la première partie de cette thèse, l'équation d'optimalité de Bellman joue un rôle essentiel pour déterminer la fonction valeur optimale de la performance du système considéré sur l'horizon infini. Cet optimum est l'unique point fixe de l'opérateur de Bellman  $\mathbb{R}^{|S|} \rightarrow \mathbb{R}^{|S|}$  défini par :*

$$(\Phi_\gamma V)_i := \min_{a \in A(i)} \left\{ c(i, a) + \gamma \sum_{j \in S} p(j/i, a) V_j \right\}$$

*Ainsi, la stratégie optimale fournie par ce point fixe dépend de l'action (ou contrôle) sélectionnée dans chaque état.*

*La question qui se poserait, concerne la possibilité de réduction de la taille des espaces d'actions disponibles. Il s'agit d'élaborer des tests efficaces afin d'identifier les décisions non-optimales dans la planification des calculs. Ces actions seront éliminées par la suite.*

## 0.5 Délimitation du sujet et objectif global

*L'effort déployé à travers ce travail est l'amélioration des performances des algorithmes d'approximations successives dans le but essentiel est de résoudre les états et actions finis, à horizon infini pour des processus de décision de Markov stationnaires, discrets et actualisés des problèmes décisionnels de Markov. L'approximation successive est un outil de la programmation dynamique. C'est une méthode simple et couramment utilisée pour résoudre les PDMs. L'approximation successive semble souvent être difficile à appliquer pour résoudre les PDMs à grande échelle en raison de sa complexité de calcul. L'élimination des actions est l'une des techniques utilisées pour accélérer la résolution des PDMs, réduit la taille du problème en éliminant les actions sous-optimales après avoir été identifiées. Dans certains cas, l'approximation successive est terminée lorsque toutes les actions, sauf une action par état,*

sont éliminées.

La détermination des bornes sur les fonctions valeurs est l'élément clé de l'élimination des actions non optimales. De nouveaux termes (gain, gain relatif et gain relatif cumulatif d'action) ont été introduits pour construire des limites plus strictes sur les fonctions de valeur. Ce qui permet d'établir un algorithme amélioré d'élimination des actions. Les performances des algorithmes proposés sont évaluées et comparées à celles d'autres algorithmes. Les algorithmes proposés ont démontré des performances exceptionnelles en termes du nombre d'itérations et du temps de calcul nécessaire pour converger.

Dans la classe des problèmes soumis à l'aléatoire, le choix traditionnel est celui d'utiliser des méthodes basées sur la théorie des probabilités (processus stochastiques, estimations statistiques et théorie des files d'attente). Néanmoins, il y a des situations de l'incertitude qui ne sont pas régies par le hasard. Le rôle de l'incertitude est inhérent à la modélisation mathématique des problèmes de prise de décision et à leurs applications. La présence inévitable d'incertitude dans des environnements complexes facilite le développement de nouvelles techniques pour aider les décideurs dans l'évaluation et l'atténuation des risques. Il existe plusieurs définitions de l'incertitude dans la littérature, il est donc important de clarifier ce que nous entendons par là. L'incertitude est définie comme une situation dans laquelle on n'a pas (ou une connaissance limitée) sur lequel de plusieurs états de la nature s'est produit ou se produira. Cette définition met en évidence le constat que l'incertitude n'est pas seulement présente dans les événements futurs mais qu'elle se rapporte également à l'analyse du passé. Depuis son introduction, la théorie des ensembles flous est devenue un outil très utile dans la modélisation mathématique des problèmes en recherche opérationnelle et dans de nombreux autres domaines. Dans cette thèse, nous étudions un type particulier d'ensemble flou, à savoir les nombres flous.



## 0.6 Méthodes et techniques utilisées

*Cette contributions s'inscrit dans le cadre d'amélioration des performances des algorithmes d'approximations successives et ayant pour but de résoudre les problèmes de décision de Markov à espaces d'états et actions finis sur horizon finie ou infinie. Pour arriver aux résultats escomptés, nous avons utilisé prioritairement l'algorithme de l'itération de la valeur (Value Iteration Algorithm : VIA) avec toutes ses différentes étapes.*

*Nos efforts dans cette thèse ont débouché sur résultats originaux qui ont fait l'objet de dix publications scientifiques.*

— 4 articles dans :

*Annals of Operations Research (Q1).*

*Journal of Advanced Research in Dynamical and Control Systems (Q3).*

*EDP Sciences (Q4).*

*International Journal of Mechanical Engineering (Q4).*

— 4 chapitres dans *Springer*

— 2 papiers de conférences dans *IEEE*.

*Notre méthodologie de travail repose sur deux points principaux :*

- i) On établit des nouveaux tests pour éliminer les actions non optimales à chaque étape de calcul lorsque cela est possible. Grâce à ces résultats développés, on améliore certains algorithmes classiques issus de la programmation dynamique par combinaison ;*
- ii) On propose un nouveau modèle des problèmes décisionnels de Markov flous en s'inspirant de la même littérature que celle des PDMs ordinaires.*

## 0.7 Structure du travail

*Ce mémoire est organisé selon le plan suivant :*

*Dans le premier chapitre, nous présentons la notion de chaînes de Markov et leurs principales propriétés. Dans le deuxième chapitre nous donnons le cadre théorique du modèle étudié des problèmes décisionnel de Markov en détaillant notre méthode*

*utilisée. Ainsi, le troisième chapitre donne un modèle théorique d'un problème décisionnel de Markov flou. On fait appel aux algorithmes classiques de la programmation dynamique par fuzzification des données imprécises. Enfin, dans le quatrième chapitre nous abordons les résultats établis pour résoudre les systèmes d'équations linéaires floues en fournissant un exemple d'illustration pour évaluer la performance de la méthode utilisée.*

*Nous finissons par une conclusion et des perspectives.*

## 1.1 Introduction

Les chaînes de Markov est un cas particulier le plus simple des suites de variables aléatoires qui sont à valeurs dans un ensemble  $E$  appelé espace d'état. Dans tout ce chapitre,  $E$  est un ensemble au plus dénombrable et  $\mathcal{P}(E)$  désigne l'ensemble de ses parties. Elles permettent de modéliser l'évolution dynamique d'un système aléatoire. Les applications des chaînes de Markov sont très nombreuses (réseaux, génétique des populations, finance, gestion des stocks, optimisation stochastique, simulation, problèmes de décision, ... ) (Voir les références [8-14]).

## 1.2 Chaînes de Markov discrètes

**Définition 1.2.0.1** Une suite  $(X_n)_{n \geq 0}$  de variables aléatoires, à valeurs dans  $(E, \mathcal{P}(E))$  et définies sur un espace probabilisé  $(\Omega, \mathcal{A}, \mathbb{P})$  est une chaîne de Markov discrète si pour tout  $n \in \mathbb{N}$ , pour tout  $(n + 1)$ -uplet  $(x_0, x_1, \dots, x_n)$  d'éléments de  $E$  tels que  $\mathbb{P}(X_n = x_n, \dots, X_1 = x_1, X_0 = x_0) > 0$ ,

$$\mathbb{P}(X_{n+1} = x_{n+1} / X_n = x_n, \dots, X_1 = x_1, X_0 = x_0) = \mathbb{P}(X_{n+1} = x_{n+1} / X_n = x_n) \quad (1.1)$$

La chaîne est dite homogène si on a de plus pour tout  $n \in \mathbb{N}$  et tous  $i, j \in E$ ,

$$\mathbb{P}(X_{n+1} = j / X_n = i) = \mathbb{P}(X_1 = j / X_0 = i) \quad (1.2)$$

Autrement dit, la loi conditionnelle de  $X_{n+1}$  sachant  $X_0, X_1, \dots, X_n$  et la loi conditionnelle de  $X_{n+1}$  sachant  $X_n$  sont identiques. Dans un tel processus, dit processus sans mémoire, la prédiction du futur à partir du présent ne nécessite pas la connaissance du passé. La relation (1.1) est dite propriété de Markov.



FIGURE 1.1 – Le mathématicien russe Andreï Markov (1856-1922)

**Exemple 1.2.0.1 (Marche aléatoire symétrique sur  $\mathbb{Z}$ )** On se donne une suite  $(X_n)_{n \geq 1}$  une suite de variables aléatoires indépendantes et identiquement distribuées (*i.i.d*), telle que chaque variable  $X_n$  prend les valeurs  $+1$  ou  $-1$  avec :

$$\mathbb{P}(X_n = 1) = \mathbb{P}(X_n = -1) = \frac{1}{2}$$

On pose :

$$S_0 = 0 \text{ et } S_n = \sum_{i=1}^n X_i \text{ pour } n \geq 1.$$

La suite  $(S_n)_{n \geq 0}$  est appelée la marche aléatoire symétrique sur  $\mathbb{Z}$ . C'est une chaîne de Markov à espace d'états  $E = \mathbb{Z}$ .

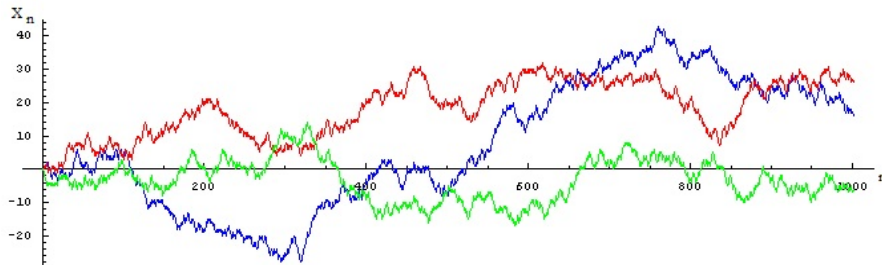


FIGURE 1.2 – Marche aléatoire symétrique

**Exemple 1.2.0.2 (Processus de Galton-Watson)** C'est un processus de branchement  $(Z_n)_{n \in \mathbb{N}}$  où  $Z_n$  désigne le nombre d'individus de sexe mâle de la  $n^{\text{ème}}$  génération portant un nom donné, ces individus étant tous issus d'un même ancêtre, l'unique mâle formant la génération 0 ( $Z_0 = 1$  p.s.). On suppose que chacun des mâles de la  $n^{\text{ème}}$  génération engendre  $X_i^n$  enfants mâles ( $1 \leq i \leq Z_n$ ) de telle sorte que

$$Z_{n+1} = \sum_{i=1}^{Z_n} X_i^n$$

On suppose que les variables aléatoires  $X_i^n$ ,  $i \geq 1$ ,  $n \geq 0$  sont indépendantes et de même loi qu'une variable aléatoire  $X$  de loi discrète sur  $\mathbb{N}$  telle que

$$\forall k \in \mathbb{N} \quad ; \quad \mathbb{P}(X = k) = p_k$$

La suite  $(Z_n)_{n \geq 0}$  est une chaîne de Markov discrète.

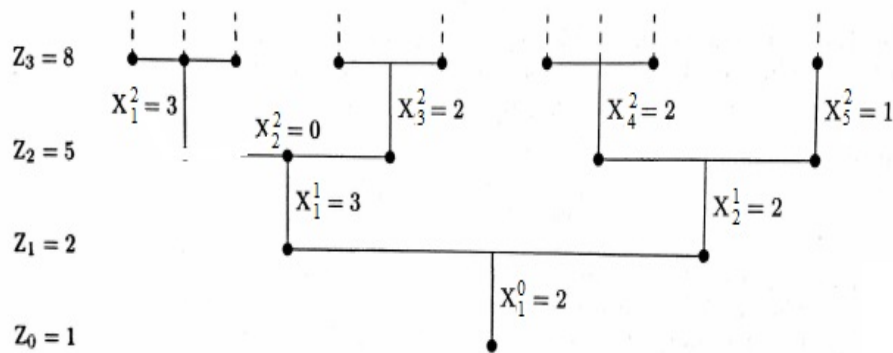


FIGURE 1.3 – Arbre de la descendance de  $Z_0$

**Définition 1.2.0.2 (Matrice stochastique)** Une matrice  $P = (P_{ij})_{i,j \in E}$  est dite matrice stochastique (ou markovienne) si elle vérifie les propriétés suivantes :

- a)  $\forall i, j \in E : P_{ij} \geq 0$  (tous ses coefficients sont positifs) ;
- b)  $\forall i \in E : \sum_{j \in E} P_{ij} = 1$  (la somme sur une ligne des coefficients est égale à 1) ;

Une matrice stochastique est une matrice carrée, avec éventuellement une infinité de lignes et de colonnes lorsque l'espace d'états  $E$  est dénombrable infini.

Pour modéliser une chaîne de Markov discrète à l'aide de représentations synthétiques afin de connaître l'évolution des états du système, on se sert des matrices

ou des graphes. Dans la suite, les chaînes de Markov seront toujours supposées homogènes.

**Définition 1.2.0.3 (Matrice de transition)** *On appelle probabilité de transition (en une seule étape ou à un pas) à l'état  $j$  à l'instant  $n + 1$  sachant que le système a été dans l'état  $i$  à l'instant  $n$ , le réel positif*

$$P_{ij} = \mathbb{P}(X_{n+1} = j / X_n = i) \quad (= \mathbb{P}(X_1 = j / X_0 = i))$$

*La famille de nombres  $P = (P_{ij})_{i,j \in E}$  est dite matrice (ou noyau) de transition de la chaîne de Markov.*

Si l'espace d'états  $E$  est fini (par exemple  $E = \{1, 2, \dots, n\}$ ), alors la matrice de transition de la chaîne de Markov peut s'écrire sous la forme :

$$P = \begin{pmatrix} P_{11} & P_{12} & \cdots & P_{1n} \\ P_{21} & P_{22} & \cdots & P_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ P_{n1} & P_{n2} & \cdots & P_{nn} \end{pmatrix}$$

**Exemple 1.2.0.3** *Soit  $E = \{0, 1\}$ . L'état 0 représente un système à l'arrêt et l'état 1 représente le même système en fonctionnement. Soit  $(X_n)_{n \geq 0}$  une chaîne de Markov sur  $E$ . Pour tout  $n \in \mathbb{N}$ ,  $X_n$  modélise l'état du système après  $n$  jours de fonctionnement. On note :*

$$\alpha = \mathbb{P}(X_{n+1} = 1 / X_n = 0) \quad , \quad \beta = \mathbb{P}(X_{n+1} = 0 / X_n = 1)$$

*La matrice de transition de la chaîne de Markov  $(X_n)_{n \geq 0}$  est donnée par :*

$$P = \begin{pmatrix} 1 - \alpha & \alpha \\ \beta & 1 - \beta \end{pmatrix}$$

**Exemple 1.2.0.4 (Modèle de diffusion d'Ehrenfest)** *On répartit  $N$  molécules d'un gaz dans deux compartiments séparés par une paroi poreuse. Entre les instants  $n$  et  $n+1$ , une molécule au hasard (avec probabilité uniforme) et une seule passe d'un compartiment à l'autre. Soit la variable aléatoire  $X_n$  égale au nombre de molécules*

contenues dans le premier compartiment au temps  $n$ . Alors la suite  $(X_n)_{n \geq 0}$  est une chaîne de Markov à valeurs dans  $E = \{0, 1, \dots, N\}$  de matrice de transition  $P = (P_{ij})_{0 \leq i, j \leq N}$  telle que :

$$P_{ij} = \mathbb{P}(X_{n+1} = j / X_n = i) = \begin{cases} \frac{i}{N} & \text{si } j = i - 1 ; \\ \frac{N-i}{N} & \text{si } j = i + 1 ; \\ 0 & \text{sinon.} \end{cases}$$

En particulier, la matrice de transition de la chaîne d'Ehrenfest définie sur l'espace d'état  $E = \{0, 1, 2, 3, 4\}$  est égale à :

$$P = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ \frac{1}{4} & 0 & \frac{3}{4} & 0 & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{3}{4} & 0 & \frac{1}{4} \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Et son graphe de transition :

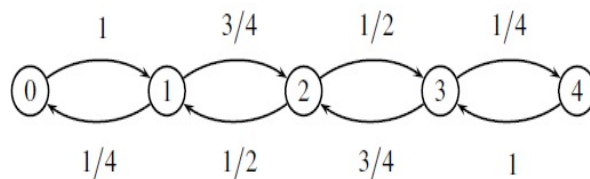


FIGURE 1.4 – Graphe probabiliste

Le théorème suivant donne une caractérisation de la chaîne de Markov.

**Théorème 1.2.0.1** *La loi d'une chaîne de Markov  $(X_n)_{n \geq 0}$  est entièrement caractérisée par le couple constitué de sa matrice de transition  $P$  et de sa loi de probabilité initiale  $\mu$  (la loi de  $X_0$ ). De plus on a, pour tout  $n \in \mathbb{N}^*$ ,  $i_0, i_1, \dots, i_n \in E$ ,*

$$\mathbb{P}(X_0 = i_0, X_1 = i_1, \dots, X_n = i_n) = \mu(i_0) \prod_{k=1}^n P_{i_{k-1}i_k}$$

**Preuve 1.2.0.1** *Sient  $n \in \mathbb{N}$  et  $i_0, i_1, \dots, i_n \in E$ .*

*Si  $\forall m \in \{1, \dots, n-1\}$ ,  $\mathbb{P}(X_0 = i_0, X_1 = i_1, \dots, X_m = i_m) \neq 0$ ; des applications successives de la formule des probabilités composées fournit*

$$\mathbb{P}(X_0 = i_0, X_1 = i_1, \dots, X_n = i_n)$$

$$\begin{aligned}
&= \mathbb{P}(X_0 = i_0)\mathbb{P}(X_1 = i_1/X_0 = i_0)\dots\mathbb{P}(X_n = i_n/X_0 = i_0, X_1 = i_1, \dots, X_{n-1} = i_{n-1}) \\
&= \mu(i_0) \prod_{k=1}^n P_{i_{k-1}i_k}
\end{aligned}$$

S'il existe  $m \in \{1, \dots, n-1\}$  tel que  $\mathbb{P}(X_0 = i_0, X_1 = i_1, \dots, X_m = i_m) = 0$ . Donc  $\mu(i_0) = \mathbb{P}(X_0 = i_0) = 0$  et comme  $m \leq n$  on aura

$$(X_0 = i_0, X_1 = i_1, \dots, X_n = i_n) \subset (X_0 = i_0, X_1 = i_1, \dots, X_m = i_m)$$

Ce qui permet d'affirmer que :  $\mathbb{P}(X_0 = i_0, X_1 = i_1, \dots, X_n = i_n) = 0$ . Par suite, on déduit l'égalité demandée.  $\square$

On peut aussi raisonner par récurrence en utilisant la propriété de Markov.

**Théorème 1.2.0.2 (Critère fondamental)** *Étant donné une suite  $(Y_n)_{n \geq 1}$  de variables aléatoires indépendantes et identiquement distribuées à valeurs dans un espace au plus dénombrable  $F$  et  $f : E \times F \rightarrow F$  une application mesurable. Pour tout  $n \in \mathbb{N}$ , on pose  $X_{n+1} = f(X_n, Y_{n+1})$ .*

*Supposons que la suite  $(X_n)_{n \geq 1}$  est indépendante de  $X_0$ . Alors la suite  $(X_n)_{n \geq 0}$  est une chaîne de Markov homogène.*

**Remarque 1.2.0.1** *Pour  $f(X_n, Y_{n+1}) = X_n + Y_{n+1}, n \in \mathbb{N}$  avec*

$$Y_n = \begin{cases} e_i & i = 1, \dots, d \text{ avec probabilité } \frac{1}{2d}, \\ -e_i & i = 1, \dots, d \text{ avec probabilité } \frac{1}{2d}. \end{cases}$$

*où  $e_i, i = 1, \dots, d$  est la base canonique de  $\mathbb{Z}^d$ , on parle ainsi de la marche aléatoire sur  $\mathbb{Z}^d$ .*

**Théorème 1.2.0.3** *La matrice de transition d'une chaîne de Markov homogène est une matrice stochastique.*

**Théorème 1.2.0.4 (Puissances de la matrice de transition)** *Le nombre :*

$$P_{ij}^{(k)} = \mathbb{P}(X_{n+k} = j/X_n = i)$$

*est la probabilité de transition de l'état  $i$  à l'état  $j$  en  $k$  étapes (pas). La matrice correspondante est égale la puissance  $k^{\text{ème}}$  de la matrice de transition  $P$ .*



$$i-e : \forall k \in \mathbb{N}^*, \forall i, j \in E, P_{ij}^{(k)} = (P^k)_{ij}$$

**Preuve 1.2.0.2** On utilise un raisonnement par récurrence.

Pour  $k = 1$  ; la propriété découle de l'homogénéité de la chaîne de Markov. *i-e* :

$$\mathbb{P}(X_{n+k} = j / X_n = i) = \mathbb{P}(X_{n+1} = j / X_n = i) = \mathbb{P}(X_1 = j / X_0 = i) = P_{ij} = P_{ij}^{(1)}$$

Supposons que la propriété est vraie à l'ordre  $k$  et montrons qu'elle vraie à l'ordre  $k+1$ . Puisque  $\{(X_{n+k} = l)\}_{l \in E}$  est un système complet d'événements, on peut écrire

$$\mathbb{P}(X_{n+k+1} = j / X_n = i) = \sum_{l \in E} \mathbb{P}(X_n = i, X_{n+k} = l, X_{n+k+1} = j) \quad (1.3)$$

En utilisant la formule des probabilités composées on écrit

$$\begin{aligned} \mathbb{P}(X_n = i, X_{n+k} = l, X_{n+k+1} = j) = \\ \mathbb{P}(X_{n+k} = l, X_n = i) \mathbb{P}(X_{n+k+1} = j / X_{n+k} = l, X_n = i) \end{aligned} \quad (1.4)$$

D'après (1.3), (1.4), la propriété de Markov et l'hypothèse de récurrence on déduit

$$\mathbb{P}(X_{n+k+1} = j / X_n = i) = \sum_{l \in E} \mathbb{P}(X_n = i, X_{n+k} = l) P_{lj} = \mathbb{P}(X_n = i) \sum_{l \in E} P_{il}^{(k)} P_{lj}$$

Ainsi, le produit de deux matrices entraîne

$$\mathbb{P}(X_{n+k+1} = j / X_n = i) = \mathbb{P}(X_n = i) P_{ij}^{(k+1)}$$

D'où ce qui fallait démontrer. □

**Théorème 1.2.0.5** Pour tout  $k \in \mathbb{N}$ , la matrice  $P^k$  est stochastique.

**Preuve 1.2.0.3** On a pour tout  $k \in \mathbb{N}$  et tous  $i, j \in E$

$$(P^k)_{ij} = P_{ij}^{(k)} = \mathbb{P}(X_k = j / X_0 = i) \geq 0$$

Il reste à montrer que la somme des coefficients sur une ligne est égale à 1.

On a

$$\forall i \in E, \quad \sum_{j \in E} P_{ij}^{(k)} = \sum_{j \in E} \mathbb{P}(X_k = j / X_0 = i) = 1$$

D'où : Ce qui fallait démontrer. □

**Théorème 1.2.0.6 (Relation de Chapman-Kolmogorov)** *Pour tous  $i, j \in E$  et tous  $n, m \in \mathbb{N}$ , on a*

$$\mathbb{P}(X_{n+m} = j / X_0 = i) = \sum_{k \in E} \mathbb{P}(X_n = k / X_0 = i) \mathbb{P}(X_m = j / X_0 = k)$$

ou encore

$$P_{ij}^{(n+m)} = \sum_{k \in E} P_{ik}^{(n)} P_{kj}^{(m)}$$

**Preuve 1.2.0.4** *On utilise la notation matricielle puis l'associativité du produit des matrices.* □

**Définition 1.2.0.4 (Filtration)** *Soit  $(\Omega, \mathcal{F}, \mathbb{P})$  un espace probabilisé. Une filtration (à temps discret) est une suite croissante  $(\mathcal{F}_n)_{n \in \mathbb{N}}$  (au sens de l'inclusion) de sous-tribus de  $\mathcal{F}$ .*

En particulier, une chaîne de Markov  $(X_n)_{n \in \mathbb{N}}$  définit automatiquement une filtration appelée la filtration naturelle des  $(X_n)$  définie par :

$$\forall n \in \mathbb{N} ; \mathcal{F}_n = \sigma(X_0, X_1, \dots, X_n)$$

**Définition 1.2.0.5 (Temps d'arrêt)** *Soit  $(\Omega, \mathcal{F}, \mathbb{P})$  un espace probabilisé muni d'une filtration  $(\mathcal{F}_n)_{n \in \mathbb{N}}$ . Une variable aléatoire tau à valeurs dans  $\mathbb{N} \cup \{+\infty\}$  est un temps d'arrêt pour la filtration  $(\mathcal{F}_n)_{n \in \mathbb{N}}$  si, et seulement si,*

$$\forall n \in \mathbb{N} ; (\tau = n) \in \mathcal{F}_n$$

ou de manière équivalente :

$$\forall n \in \mathbb{N} ; (\tau \leq n) \in \mathcal{F}_n$$

Si  $(X_n)_{n \in \mathbb{N}}$  est un processus stochastique et  $A$  est un événement, alors le temps d'atteinte

$$\tau_A = \inf\{n \in \mathbb{N} / X_n \in A\}$$

est un temps d'arrêt adapté à la filtration naturelle des  $(X_n)$ .

À tout temps d'arrêt  $\tau$ , on peut associer une sous-tribu de  $\mathcal{F}$  notée  $\mathcal{F}_\tau$  et définie par

$$\mathcal{F}_\tau = \{A \in \mathcal{F} / \forall n \in \mathbb{N}, A \cap (\tau = n) \in \mathcal{F}_n\}$$

C'est la tribu des événements antérieurs à  $\tau$ .

## 1.3 Classement des états : Récurrence et transience

Les chaînes de Markov apparaissent fréquemment dans la modélisation de processus physiques et conceptuels qui évoluent dans le temps. Par exemple, la diffusion de liquides à travers une membrane semi-poreuse, la propagation de maladies au sein d'une population et le flux de personnel dans les rangs d'une organisation peuvent tous être modélisés à l'aide de chaînes de Markov. Dans chacun de ces cas, le système peut être trouvé dans n'importe lequel d'un nombre fini d'états, et les transitions entre les états se produisent à des instants discrets selon des probabilités spécifiées. Dans cette section, certains concepts de base des chaînes de Markov finies sont définis, conduisant à la notion de classification des états de la chaîne en tant que « récurrents » ou « transitoires ». Les approches algébriques et théoriques des graphes s'avèrent utiles pour identifier de tels états. Les techniques algorithmiques pour effectuer la classification des états sont également discutées.

### 1.3.1 Préliminaires

Supposons que  $(X_n)_n$  soit une chaîne de Markov à espace d'états fini  $E$ . A chaque instant (discret)  $n$  la chaîne  $(X_n)_n$  sera dans l'un de ces états. La quantité  $P_{ij}$  désigne la probabilité conditionnelle que  $(X_n)_n$  soit dans l'état  $j$  à l'instant  $n+1$ , étant donné qu'il a été observé dans l'état  $i$  à l'instant  $n$ . Implicitement, nous supposons que la chaîne de Markov est homogène : à savoir que les valeurs  $P_{ij}$  sont indépendantes du temps  $n$ . Ces probabilités de transition d'état (à une étape) définissent la matrice de probabilité de transition  $P = (P_{ij})$ . En général, soit  $P_{ij}^{(k)}$  la probabilité que la chaîne  $(X_n)_n$  passe de l'état  $i$  à l'état  $j$  après  $k$  transitions. Ensuite, la matrice de probabilité de transition en  $k$  étapes  $P(k) = (P_{ij}^{(k)})$  est donnée par  $P(k) = P^k$ , la puissance matricielle  $k^{\text{ième}}$  de la matrice de probabilité de transition  $P$ . Notez que lorsque  $k = 0$  cela produit  $P(0) = P^{(0)} = I_{|E|}$ , la matrice identité d'ordre  $|E|$ , en accord avec l'observation qu'après  $k = 0$  étapes la chaîne de Markov est toujours dans son état initial. La chaîne de Markov est dite irréductible si, pour tout couple d'états  $i$  et  $j$ , il existe  $r, s \geq 0$  avec  $P_{ij}^{(r)} > 0$  et  $P_{ij}^{(s)} > 0$ .

A la chaîne de Markov  $(X_n)_n$  est associé un digraphe (graphe orienté)  $G = G_E$

ayant l'ensemble des états  $E = \{x_1, x_2, \dots, x_n\}$  et l'ensemble des arêtes  $F$ . Chaque état correspond à un état de la chaîne  $X$ , et  $G$  contient l'arête  $(i, j) \in E \times E$  si et seulement si  $p_{ij} > 0$ . Ainsi, le digraphe, ou diagramme de transition d'état,  $G$  capture la structure des transitions d'état possibles en une étape; les valeurs numériques réelles des probabilités de transition d'état sont ignorées. En termes de théorie des graphes,  $P_{ij}^{(k)} > 0$  signifie qu'il existe un chemin dirigé  $Q$  de longueur  $l(Q) = k$  (nombre d'arêtes) de l'état  $i$  à l'état  $j$  dans  $G$ . Si cela est vrai pour un certain  $k \geq 0$ , alors l'état  $j$  est accessible depuis l'état  $i$ , noté  $i \rightarrow j$ . Observons que  $i \rightarrow i$  puisque nous considérons que l'état  $i$  est accessible depuis lui-même par un chemin de longueur 0. S'il n'y a pas de chemin dans  $G$  de  $i$  à  $j$ , alors on écrit  $i \not\rightarrow j$ . Si les deux  $i \rightarrow j$  et  $j \rightarrow i$  sont vérifiés, alors on dit que les états  $i$  et  $j$  communiquent, noté  $i \leftrightarrow j$ . Un chemin joignant un état à lui-même s'appelle un circuit. Si ce circuit ne contient pas d'états répétés, alors c'est un cycle.

Un concept important dans l'analyse des chaînes de Markov est la catégorisation des états comme récurrents ou transitoires. La chaîne de Markov, une fois démarrée dans un état récurrent, reviendra à cet état avec une probabilité égale à 1. Cependant, pour un état transitoire, il existe une probabilité positive que la chaîne, une fois démarrée dans cet état, n'y retournera jamais. Ce même concept peut être éclairé en utilisant des concepts de la théorie des graphes, qui n'impliquent pas les valeurs numériques des probabilités. A savoir, nous définissons l'état  $i$  comme transitoire s'il existe un état  $j$  pour lequel  $i \rightarrow j$  mais  $j \not\rightarrow i$ . Sinon, l'état  $i$  est dit récurrent. (Ces définitions ne s'appliquent qu'aux chaînes de Markov finies.) Soit  $T$  l'ensemble de tous les états transitoires, et soit  $R = E - T$  l'ensemble de tous les états récurrents.

**Définition 1.3.1.1** *Une classe de communication  $\mathcal{C}$  est fermée si, pour tout  $x \in \mathcal{C}$ , il n'y a pas de  $y \notin \mathcal{C}$  tel que  $x \rightarrow y$ .*

Soit  $(X_n)_{n \geq 0}$  une chaîne de Markov à valeurs dans  $E$  et  $P$  sa matrice de transition. On considère le temps d'attente de l'état  $i \in E$  défini par :

$$T_i = \inf\{n \geq 1 : X_n = i\}$$

C'est le temps du premier passage à l'état  $i$  de la chaîne. On montre facilement que  $T_i$  est un temps d'arrêt par rapport à la filtration naturelle  $(\mathcal{F}_n)_{n \in \mathbb{N}}$ .

En effet :

$$\{T_i = n\} = \{X_0 \neq n\} \cap \dots \cap \{X_{n-1} \neq n\} \cap \{X_n = n\} \in \mathcal{F}_n.$$

Les définitions suivantes fournissent les notions formelles des états récurrents et des états transitoires :

**Définition 1.3.1.2 (Récurrence)** *Un état  $i \in E$  est récurrent si :*

$$\mathbb{P}(T_i < +\infty / X_0 = i) = 1$$

*C'est-à-dire : si on part de  $i$ , on y revient sûrement. Un état récurrent est donc visité une infinité de fois.*

**Définition 1.3.1.3 (Transience)** *Un état  $i \in E$  est dit transitoire s'il n'est pas récurrent. Autrement dit :*

$$\mathbb{P}(T_i < +\infty / X_0 = i) < 1$$

*Un état transitoire est visité un nombre fini de fois.*

**Définition 1.3.1.4 (Périodicité)** *Soit  $i \in E$ . La période de l'état  $i$ , notée  $d(i)$ , est le plus grand commun diviseur de tous les entiers  $n \geq 1$  pour lesquels  $P_{ii}^n > 0$  :*

$$d(i) := \text{pgcd}(n \geq 1, P_{ii}^n > 0)$$

*On convient de poser  $\text{pgcd}(\emptyset) = +\infty$ . If  $d(i) = 1$ , on dit que l'état  $i$  est apériodique, sinon il est dit périodique. Si tous les états sont apériodiques, on dit que la chaîne de Markov est une chaîne apériodique.*

**Théorème 1.3.1.1** *Si les états  $i$  et  $j$  communiquent entre eux et l'état  $i$  est périodique, alors l'état  $j$  est aussi périodique tel que  $d(j) = d(i)$ .*

**Preuve 1.3.1.1** *Posons :  $d = d(i)$ . Puisque  $i \leftrightarrow j$  et l'état  $i$  est périodique, il existe des entiers naturels  $n$  et  $m$  tels que :  $P_{ij}^n > 0$ ,  $P_{ji}^m > 0$  et  $P_{ii}^d > 0$ .*

*D'une part, on a :  $P_{jj}^{m+d+n} \geq P_{ji}^m P_{ii}^d P_{ij}^n > 0$ . Donc : la période de l'état  $j$  existe.*

*D'autre part, on a :  $P_{jj}^{m+2d+n} \geq P_{ji}^m P_{ii}^{2d} P_{ij}^n > 0$ . Donc  $d(j)$  divise  $m + d + n$  et  $m + 2d + n$ . Par soustraction, on déduit que  $d(j)$  divise  $d(i)$ . De la même façon, on montre que  $d(i)$  divise  $d(j)$ . Par suite :  $d(j) = d(i)$ .  $\square$*

### 1.3.2 Considérations algébriques et théoriques des graphes probabilistes

Certains états d'une chaîne de Markov se comportent de manière similaire. Le fait algébrique suivant sous-tend ce type de "classification d'état".

**Lemme 1.3.2.1** *La relation de "communication"  $\leftrightarrow$  est une relation d'équivalence définie sur l'ensemble des états  $E$  dont les classes sont appelées les classes de communication de la chaîne de Markov. Identiquement équivalent à :*

- i)  $\forall i \in E, i \leftrightarrow i$ ;
- ii)  $\forall i, j \in E; (i \leftrightarrow j) \Rightarrow (j \leftrightarrow i)$ ;
- iii)  $\forall i, j, k \in E; (i \leftrightarrow j) \text{ et } (j \leftrightarrow k) \Rightarrow (i \leftrightarrow k)$ .

**Preuve 1.3.2.1** *Cette relation est symétrique par construction. Il reste à montrer les propriétés de réflexivité et de transitivité.*

Réflexivité :

Puisque  $P_{ij}^{(0)} = \mathbb{P}(X_0 = i / X_0 = i) = 1 > 0$ , pour tout état  $i \in E$ , la relation de communication est réflexive.

Transitivité :

Si  $i \rightarrow j$  et  $j \rightarrow \ell$ , alors ils existent  $m, n \in \mathbb{N}$  tels que  $P_{ij}^{(m)} > 0$  et  $P_{j\ell}^{(n)} > 0$ . D'après la relation de Chapman-Kolmogorov, on peut écrire

$$P_{i\ell}^{(n+m)} = \sum_{k \in S} P_{ik}^{(m)} P_{k\ell}^{(n)} \geq P_{ij}^{(m)} P_{j\ell}^{(n)} > 0$$

Donc :  $i \rightarrow \ell$ . De la même manière, on montre que  $\ell \rightarrow i$ . D'où ce qui fallait démontrer. □

Par conséquent, la relation d'équivalence  $\leftrightarrow$  partitionne l'ensemble  $E$  en un certain nombre de classes d'équivalence (classes communicantes) : des sous-ensembles disjoints dont l'union est  $E$ . Remarquons que la chaîne  $(X_n)_n$  est irréductible précisément lorsqu'il n'y a qu'une seule classe d'équivalence sous . Un résultat utile est que les nœuds au sein d'une classe d'équivalence donnée se comportent effectivement de manière similaire.

**Lemme 1.3.2.2** *Si l'état  $i$  est récurrent et  $i \leftrightarrow j$  alors l'état  $j$  est récurrent. Si l'état  $i$  est transitoire et  $i \leftrightarrow j$  alors l'état  $j$  est transitoire.*

Dans la théorie des graphes orientés,  $G$  est dit fortement connexe s'il existe un chemin entre n'importe quelle paire de nœuds  $i, j \in G$ . En d'autres termes,  $i \rightarrow j$  est valable pour tout  $i, j$ , ce qui signifie que  $i \leftrightarrow j$  pour tout  $i, j$ . Ainsi, une chaîne de Markov irréductible  $M$  est simplement une chaîne dont le digraphe  $G$  est fortement connexe. En général, les classes communicantes de  $M$  ne sont que les sous-graphes maximaux fortement connectés de  $G$  (les composantes fortes de  $G$ ). On sait que si les nœuds  $i$  et  $j$  se trouvent sur un circuit commun, alors ils appartiennent à la même composante forte, et inversement. En conséquence, lorsque les nœuds de chaque composante forte sont combinés en un nouveau "super-nœud", alors le graphe condensé  $\hat{G}$  régissant ces super-nœuds ne peut contenir aucun cycle. Donc,  $\hat{G}$  est un graphe acyclique.

**Lemme 1.3.2.3** *Les états récurrents du graphe  $G$  sont précisément les états dont les "super-nœuds" correspondants n'ont pas d'arêtes partantes dans  $\hat{G}$ .*

Le concept de communication divise l'espace d'état en différentes classes. Deux états qui communiquent sont dits de la même classe. Tous les membres d'une classe communiquent entre eux. Si une classe n'est accessible depuis aucun état en dehors de la classe, nous définissons la classe comme une classe communicante fermée. Une chaîne de Markov dans laquelle tous les états communiquent, ce qui signifie qu'il n'y a qu'une seule classe, est appelée chaîne de Markov irréductible

## 1.4 Conclusion du chapitre

Si une séquence d'événements présente la propriété de Markov de s'appuyer sur l'état précédent, alors la séquence est dite « markovienne » par nature. Ainsi, une chaîne de Markov est un modèle stochastique (toujours changeant) qui est utilisé pour prédire/estimer/deviner le résultat d'un événement en fonction uniquement de l'état précédent et de son action. Elle permet décrire une séquence d'événements consécutifs où la probabilité ou la chance d'un événement ne dépend que de l'événement qui le précède. Effectivement, l'état fait référence à la « tranche » actuelle de

l'environnement. C'est la condition particulière dans laquelle un agent est contenu à un pas de temps spécifique.

Les chaînes de Markov sont aussi utiles dans l'apprentissage par renforcement. Effectivement, elles nous ont permis d'atteindre et même de dépasser les performances humaines dans de nombreux domaines et problèmes. Ceci a eu lieu depuis l'enseignement aux robots de la manipulation d'objets avec soin jusqu'à la réalisation de jeux Atari difficiles qui n'avaient jamais été battus auparavant par l'apprentissage automatique statistique.



## 2.1 Introduction

Les processus décisionnels de Markov (PDMs) représentent un cadre élégant pour modéliser les problèmes de la prise de décision séquentielle dans l'incertitude. Dans ce cadre, le système est caractérisé par un processus stochastique à valeurs dans un ensemble d'états qui évolue avec le temps. L'évolution de ces variables d'état dépend en partie des décisions. Le but ultime de tout décideur est de trouver une stratégie optimale en tant qu'une fonction qui indique quelle action doit être sélectionnée lorsque le système est dans l'un de ses états possibles.

Au cours des dernières décennies, divers types de problèmes de PDMs ont été proposés et analysés. Dans cette recherche, nous considérerons les états et actions stationnaires et finis, l'horizon infini, les modèles PDMs à temps discret avec le critère du coût  $\gamma$ -pondéré. On sait que pour les modèles PDMs ci-dessus, sous des hypothèses appropriées, il existe une stratégie déterministe stationnaire optimale ; par conséquent, dans cette recherche, nous limitons notre attention à ce type de stratégies (Voir les références [29-30-31],[35],[40],[50] et [53]).

Dans ce chapitre, on va présenter un aperçu sur quelques outils de la programmation dynamique pour résoudre les problèmes décisionnels de Markov sur horizon fini et horizon infini. Ainsi, on établit des nouveaux tests qui permettent d'identifier

les décisions non optimales. Ces procédures d'élimination des actions non optimales visent à réduire la complexité des calculs grâce à l'établissement des algorithmes efficaces dans la même littérature.

## 2.2 Modèles sur horizon fini

### 2.2.1 Notions préliminaires

Dans cette section on va donner un aperçu sur la terminologie utilisée dans les problèmes décisionnels de Markov abordés dans ce manuscrit.

Aux instants finis  $t = 1, 2, \dots, T$ , où  $T$  est un entier donné, un système est observé par un décideur comme étant dans l'un des états d'un espace d'états fini  $S$  sous le processus d'états  $\{X_t, t = 1, \dots, T\}$  qui prend des valeurs en  $S$ . A chaque époque  $t$ , le décideur contrôle le système en choisissant une action  $a$  dans l'espace d'actions fini  $A$ . Si l'action  $a$  est sélectionné dans l'état  $i$ , ce qui suit se produit indépendamment de l'historique du processus à l'instant  $t$  : un coût immédiat  $c_t(i, a)$  est payé et l'état du système au moment de décision suivant est l'état  $j$  avec une probabilité de transition  $p^t(j/i, a)$ , telle que  $p^t(j/i, a) \geq 0$  et  $\sum_{j \in S} p^t(j/i, a) = 1$ .

Ainsi, l'espace échantillon (l'univers) est donné par  $\Omega = \{S \times A\}^T$ , de sorte que la réalisation typique  $\omega$  peut être représentée par  $\omega = (x_1, a_1, x_2, a_2, \dots, x_T, a_T)$ . Les variables aléatoires d'état et d'action  $X_t, A_t$  pour  $t = 1, 2, \dots, T$  sont alors définies comme les applications de coordonnées  $X_t(\omega) := x_t$  et  $A_t(\omega) := a_t$ . Partout, l'espace échantillon  $\Omega$  sera équipé de la  $\sigma$ -algèbre  $\mathcal{B}$  engendrée par les variables aléatoires  $(X_t, A_t, t = 1, 2, \dots, T)$ .

Afin de donner une définition formelle d'une stratégie, laissez d'abord  $\mathcal{Q}$  l'ensemble de toutes les mesures de probabilité sur l'espace d'action  $A$ , i-e :

$$\mathcal{Q} := \{(q_1, q_2, \dots, q_{|A|}) : q_1 + q_2 + \dots + q_{|A|} = 1, q_i \geq 0, 1 \leq i \leq |A|\}$$

où  $|A|$  est le cardinal de l'ensemble  $A$ . Une stratégie  $\pi$  est donc définie comme une séquence  $\pi = (\pi^1, \pi^2, \dots, \pi^T)$  où  $\pi^t : S \times A \rightarrow [0, 1]$  est formellement une fonction telle que  $\pi^t(i, a) = \pi_{i,a}^t, \forall i \in S, a \in A$ .

Pour une stratégie fixe  $\pi$  et un état initial  $i$ , nous pouvons maintenant construire

la mesure de probabilité  $\mathbb{P}_\pi^i$  pour l'espace mesurable  $(\Omega, \mathcal{B})$ . Les distributions en dimension finie de la mesure de probabilité  $\mathbb{P}_\pi^i$  sont définies comme suit :

$$\mathbb{P}_\pi^i(X_1 = x) = 1; \quad (2.1)$$

$$\begin{aligned} \mathbb{P}_\pi^i(A_t = a/X_1 = x_1, A_1 = a_1, \dots, X_{t-1} = x_{t-1}, A_{t-1} = a_{t-1}, X_t = x_t) \\ = \pi^t(x_1, a_1, \dots, x_t, a_t, x_t); \end{aligned} \quad (2.2)$$

$$\begin{aligned} \mathbb{P}_\pi^i(X_{t+1} = y/X_1 = x_1, A_1 = a_1, \dots, X_{t-1} = x_{t-1}, A_{t-1} = a_{t-1}, X_t = x, A_t = a) \\ = p(y/x, a). \end{aligned} \quad (2.3)$$

où  $p(y/x, a)$  est la loi du mouvement, qui est donnée et déterminée à partir de la physique du problème.

Afin d'indiquer la dépendance des probabilités sur la stratégie  $\pi$ , la notation  $\mathbb{P}_{\pi^i}(B)$  désignera la probabilité qu'un événement  $B$  se produise lorsque la politique  $\pi$  est utilisé. A partir d'une application standard du théorème de cohérence de Kolmogorov, nous savons qu'il existe une mesure de probabilité unique  $\mathbb{P}_\pi^i$ , sur  $(\Omega, \mathcal{B})$  telle que (2.1), (2.2) et (2.3) sont valables pour tous les historiques possibles et pour tout  $t \geq 1$  où  $\mathbb{P}_\pi^i(B) = \mathbb{P}_\pi(B/X_1 = i)$ . Ainsi, pour chaque stratégie  $\pi$  et état initial  $i$ , nous avons construit un espace de probabilité  $(\Omega, \mathcal{B}, \mathbb{P}_\pi^i)$ .

Soit  $\Pi$  l'ensemble de toutes les stratégies. En termes de notre modèle, nous serons intéressants dans une classe particulière de stratégies qui ont des propriétés stochastiques très importantes. La règle de décision selon laquelle il dépend des états et des actions précédents du système uniquement à travers l'état actuel du système, on dit qu'il est markovien (sans mémoire). L'ensemble de toutes les stratégies markoviennes est désigné par  $\Pi^M$ . Parmi les stratégies de  $\Pi^M$ , chaque règle de décision qui spécifie une action avec certitude pour chaque état courant est appelée déterministe (ou pure). Leur ensemble est noté  $\Pi^{MD}$ . Avec ces définitions, nous avons :

$$\Pi^{MD} \subset \Pi^M \subset \Pi$$

Pour  $t = 1, 2, \dots, T$ , on voit facilement qu'une règle de décision  $\pi^t$  définit une matrice de transition de probabilité

$$\mathbf{P}(\pi^t) = [p(j/i, \pi^t)]_{i,j \in S}$$

avec des entrées données par

$$p(j/i, \pi^t) = \sum_{a \in A(i)} p^t(j/i, a) \pi_{i,a}^t$$

Puisque  $\sum_{j \in S} p^t(j/i, a) = 1$  pour tout  $i \in S$  et  $a \in A(i)$ ,  $\mathbf{P}(\pi^t)$  est une matrice stochastique (la somme des coefficients de chaque ligne est égale à 1) qui définit de manière unique une chaîne de Markov.

## 2.2.2 Description du modèle et critère d'optimalité

Un PDM est un modèle séquentiel pour la prise de décision dans des environnements totalement observables. Le modèle de contrôle de Markov en temps discret (MCM) est défini par la donnée de la collection :

$$(S, A = \bigcup_{i \in S} A(i), Q = \{p_t(j/i, a)\}, c = \{c_t(i, a)\}, T, \gamma)$$

où :

- $S$  est un ensemble fini de tous les états possibles du système (espace d'états) ;
- $A$  est un ensemble fini de toutes les actions possibles que l'agent peut choisir d'accomplir en vue de modifier l'état de son environnement (espace d'action). En général, l'espace  $A$  peut dépendre de l'état courant. C'est une réunion de sous-ensembles  $A(i)$  contenant toutes les actions disponibles à prendre en compte lorsque le système est dans l'état  $i \in S$ .

- $p_t : S \times A \times S \rightarrow [0, 1]$  est une fonction de transition qui donne la probabilité d'atteindre l'état  $j$  en partant de l'état  $i$  quand l'action  $a$  est exécutée à l'instant  $t$ .

- $c, c_t : S \times A \rightarrow \mathbb{R}$  sont les fonctions de coût. Le nombre réel  $c_t(i, a)$  est le coût à payer par l'agent lorsqu'il exécute l'action  $a$  à partir de l'état  $i$  à l'instant  $t$ .

- $T$  est l'horizon ou le nombre d'étapes après lesquelles le problème se termine.

- $\gamma \in ]0, 1]$  est le facteur d'actualisation.

À toute règle de décision  $\pi^t \in \Pi$ , on fait correspondre un vecteur coût  $c(\pi^t)$  dont les composantes sont définies par :

$$c(\pi^t)_i = \sum_{a \in A(i)} \pi_{i,a}^t c_t(i, a), \quad i \in S$$

Chaque composante s'interprète en tant qu'une espérance des coûts à payer lorsque le système part de l'état  $i$  en choisissant la règle de décision  $\pi^t \in \Pi$  à l'instant  $t$ .

**Définition 2.2.2.1 (Critère du coût actualisé ( $\gamma$ -pondéré))** *Considérons un PDM avec un horizon de planification fini de  $T$  périodes. Pour toute stratégie de contrôle fixe  $\pi$  et état initial  $i \in S$ , le coût actualisé total attendu sur l'horizon  $T$  est défini par :*

$$J_\gamma^T(i, \pi) := \sum_{t=1}^T \gamma^{t-1} \mathbb{E}_\pi^i [c(X_t, A_t)] = \sum_{t=1}^T \sum_{j,a} \gamma^{t-1} c_t(i, a) \pi_{i,a}^t \quad (2.4)$$

La permutation de la sommation et espérance,  $J_\gamma^T(i, \pi)$  peut être défini de la manière suivante :

$$J_\gamma^T(i, \pi) = \mathbb{E}_\pi^i \left[ \sum_{t=1}^T \gamma^{t-1} c(X_t, A_t) \right]$$

On pose :

$$J_\gamma^T(i) := \inf_{\pi \in \Pi} J_\gamma^T(i, \pi), \quad i \in S$$

En notation vectorielle,  $J_\gamma^T = \inf_{\pi \in \Pi} J_\gamma^T(\cdot, \pi)$ . Le vecteur  $J_\gamma^T$  est appelé vecteur valeur.

Ce vecteur optimal peut s'écrire de la forme :

$$J_\gamma^T = \sum_{t=1}^T \gamma^{t-1} P(\pi^1) P(\pi^2) \dots P(\pi^{t-1}) c(\pi^t)$$

Une stratégie  $\pi^* \in \Pi$  est dite optimale si

$$J_\gamma^T = J_\gamma^T(\cdot, \pi^*)$$

## 2.2.3 Approche de la programmation dynamique

Bellman a formulé son fameux principe d'optimalité (le principe de Bellman) comme suit : "Une politique optimale a la propriété que quels que soient l'état initial et la décision initiale, les décisions restantes doivent constituer une politique optimale vis-à-vis de l'état résultant de la première décision".

Puisque les espaces d'états et d'actions sont finis, le principe de Bellman conduit à l'équation d'optimalité suivante

$$J^t(i) = \min_{a \in A(i)} \left\{ c_t(i, a) + \gamma \sum_{j \in S} p_{iaj}^t J^{t+1}(j) \right\} \quad (2.5)$$

pour  $t = 1, 2, \dots, T$  et  $J^{T+1}(i) = 0, \forall i \in S$

## 2.2.4 Algorithme d'induction en arrière.

L'algorithme d'induction vers l'arrière est une procédure itérative. En partant de la fin de l'horizon de planification, on calcule la valeur des périodes précédentes. Ainsi, une stratégie optimale est obtenue après  $T$  itérations.

**Lemme 2.2.4.1** *Pour tout état de départ  $i$  et toute stratégie considérée  $\pi \in \Pi$ , il existe une stratégie markovienne  $\tilde{\pi} \in \Pi^M$  telle que*

$$i) \mathbb{P}_{\tilde{\pi}}^i(X_t = j, A_t = a) = \mathbb{P}_{\pi}^i(X_t = j, A_t = a), \quad t \in \mathbb{N}, \quad j \in S, \quad a \in A(j)$$

$$ii) \mathbb{E}_{\tilde{\pi}}^i[c_t(X_t, A_t)] = \mathbb{E}_{\pi}^i[c_t(X_t, A_t)], \quad t \in \mathbb{N}$$

où  $\mathbb{N}$  désigne l'ensemble des entiers naturels.

**Preuve 2.2.4.1** *Du corollaire 2.1 (pp. 22) dans Feinberg and Shwartz [41], l'assertion i) découle. En utilisant l'assertion i), on aura*

$$\begin{aligned} \mathbb{E}_{\tilde{\pi}}^i[c_t(X_t, A_t)] &= \sum_{j \in S} \sum_{a \in A(j)} \mathbb{P}_{\tilde{\pi}}^i(X_t = j, A_t = a) c_t(j, a) \\ &= \sum_{j \in S} \sum_{a \in A(j)} \mathbb{P}_{\pi}^i(X_t = j, A_t = a) c_t(j, a) = \mathbb{E}_{\pi}^i[c_t(X_t, A_t)] \quad \square \end{aligned}$$

**Théorème 2.2.4.1** *Soit  $W_i^{T+1} = 0$ . Pour  $t = T, T-1, \dots, 1$ , on considère successivement les règles de décision pures  $\pi^{*t}$  et le vecteur  $W^t \in \mathbb{R}^{|S|}$  défini par*

$$\{c(\pi^{*t})\}_i + \gamma \{P(\pi^{*t})W^{t+1}\}_i = \min_{a \in A(i)} \left\{ c_t(i, a) + \gamma \sum_{j \in S} p_{iaj}^t W_j^{t+1} \right\}$$

où  $V_i$  est la  $i^{\text{ème}}$  composante du vecteur  $V \in \mathbb{R}^{|S|}$ .

et

$$W^t = c(\pi^{*t}) + \gamma P(\pi^{*t})W^{t+1}$$

Donc,  $\pi^* = (\pi^{*1}, \pi^{*2}, \dots, \pi^{*T})$  est une stratégie optimale. De plus  $W^1$  est sa fonction valeur correspondante  $J_{\gamma}^T$ .

**Preuve 2.2.4.2** *On raisonne par récurrence sur  $T$ . Soit  $\pi = (\pi^1, \pi^2, \dots, \pi^T)$  une stratégie arbitraire.*

Pour  $T=1$ , on a :

$$J_{\gamma}^1(i, \pi) = \sum_{j \in S} \sum_{a \in A(j)} \mathbb{P}_{\pi}^i(X_1 = j, A_1 = a) c_1(j, a) = \sum_{a \in A(j)} c_1(i, a) \pi_{i,j}^1$$

$$\geq \left[ \min_{a \in A(i)} c_1(i, a) \right] \sum_{a \in A(j)} \pi_{i,j}^1 = \min_{a \in A(i)} c_1(i, a) = W_i^1 = J_\gamma^1(i, \pi^*), \quad i \in S$$

Supposons que le résultat est vérifié pour  $T = 1, 2, \dots, t$ . On considère arbitrairement un état  $i \in S$ . D'après le lemme 2.2.4.1, il existe donc une stratégie markovienne  $\tilde{\pi} = (\tilde{\pi}^1, \tilde{\pi}^2, \dots, \tilde{\pi}^t)$  telle que

$$J_\gamma^{t+1}(i, \tilde{\pi}) = J_\gamma^{t+1}(i, \pi)$$

Définissons la stratégie markovienne  $\rho = (\rho^1, \rho^2, \dots, \rho^t)$  donnée par  $\rho_{j,a}^k = \tilde{\pi}_{j,a}^{k+1}$  pour  $k = 1, 2, \dots, t$ . Par hypothèse de récurrence, il découle que  $J_\gamma^t(j, \rho) \geq W_j^2$ ,  $j \in S$ , du fait que pour l'horizon de planification de  $t+1$  périodes,  $W^2$  est similaire à  $C^1$  pour l'horizon de planification de  $t$  périodes. Ainsi

$$\begin{aligned} J_\gamma^{t+1}(i, \pi) &= J_\gamma^{t+1}(i, \tilde{\pi}) = \sum_{a \in A(i)} \tilde{\pi}_{i,a}^1 \{c_1(i, a) + \gamma \sum_{j \in S} p_{iaj}^1 J_\gamma^t(j, \rho)\} \\ &\geq \sum_{a \in A(i)} \tilde{\pi}_{i,a}^1 \{c_1(i, a) + \gamma \sum_{j \in S} p_{iaj}^1 W_j^2\} \geq \min_{a \in A(i)} \{c_1(i, a) + \gamma \sum_{j \in S} p_{iaj}^1 W_j^2\} \sum_{a \in A(i)} \tilde{\pi}_{i,a}^1 \\ &= \min_{a \in A(i)} \{c_1(i, a) + \gamma \sum_{j \in S} p_{iaj}^1 W_j^2\} = W_i^1 \end{aligned}$$

D'autre part, on a

$$\begin{aligned} W^1 &= c(\pi^{*1}) + \gamma P(\pi^{*1})W^2 = c(\pi^{*1}) + \gamma P(\pi^{*1})[c(\pi^{*2}) + \gamma P(\pi^{*2})W^3] \\ &= c(\pi^{*1}) + \gamma P(\pi^{*1})c(\pi^{*2}) + \gamma^2 P(\pi^{*1})P(\pi^{*2})W^3 \\ &= c(\pi^{*1}) + \gamma P(\pi^{*1})C^2 + \gamma^2 P(\pi^{*1})P(\pi^{*2})W^3 \\ &= \dots \\ &= \sum_{k=1}^{t+1} \gamma^{k-1} [P(\pi^{*1})P(\pi^{*2})\dots P(\pi^{*(k-1)})c(\pi^{*k})] = J_\gamma^{t+1}(\pi^*) \end{aligned}$$

Par suite,  $J_\gamma^{t+1}(i, \pi^*) = W_i^1 \leq J_\gamma^{t+1}(i, \pi)$  pour tout état  $i \in S$ ,  $\pi \in \Pi$ . Par conséquent,  $\pi^*$  est une stratégie optimale et  $C^1$  est la fonction valeur optimal.  $\square$

Le théorème 2.2.4.1 permet d'établir le fameux algorithme suivant :

---

**Algorithme 1 : Backward Induction Algorithm (BIA)**

---

**Input :** MDP =  $(S, A = \bigcup_{i \in S} A(i), Q = \{p_t(j/i, a)\}, c = \{c_t(i, a)\}, T, \gamma)$

**Output :** The optimal value vector and an optimal policy

```
1 Initialization :  $J_i^{T+1} = 0$  for all  $i \in S$ 
2 for  $t=T$  to 1 do
3   for  $i \in S$  do
4      $J_i^t = \min_{a \in A(i)} \{c_t(i, a) + \gamma \sum_{j \in S} p_{iaj}^t J_j^{t+1}\}$ 
5      $\pi^{*t}(i) = \arg \min_{a \in A(i)} \{c_t(i, a) + \gamma \sum_{j \in S} p_{iaj}^t J_j^{t+1}\}$ 
6 return  $J^1$  and  $\pi^*$ 
```

---

En posant le changement de variables :  $V_i^t = C_i^{T-t+1}$ ,  $i \in S$ ,  $t = 0, 1, \dots, T$ , on présente une forme simple et pratique basée sur l'induction directe.

---

**Algorithme 2 : Forward Induction Algorithm (FIA)**

---

**Input :** MDP =  $(S, A = \bigcup_{i \in S} A(i), Q = \{p_t(j/i, a)\}, c = \{c_t(i, a)\}, T, \gamma)$

**Output :** The optimal value vector and a T-optimal policy

```
1 Initialization :  $V_i^0 = 0$  for all  $i \in S$ 
2 for  $t=1$  to  $T$  do
3   for  $i \in S$  do
4      $V_i^t = \min_{a \in A(i)} \{c_t(i, a) + \gamma \sum_{j \in S} p_{iaj}^t V_j^{t-1}\}$ 
5      $\pi^{*t}(i) = \arg \min_{a \in A(i)} \{c_t(i, a) + \gamma \sum_{j \in S} p_{iaj}^t V_j^{t-1}\}$ 
6 return  $V^T$  and  $\pi^*$ 
```

---

**Exemple 2.2.4.1** On considère un PDM donné par :

$$S = \{1, 2\}; A(1) = A(2) = \{a_1, a_2, a_3\}; \gamma = 0.7; T = 10.$$

Probabilités de transition :

$$p_{1a_1 1} = 1; p_{1a_2 1} = 0.7; p_{1a_3 1} = 1; p_{1a_2 2} = 0.3; p_{2a_1 2} = 0.4; p_{2a_2 2} = 1; p_{2a_3 2} = 1; p_{2a_1 1} = 0.6$$



Coûts :

$$c(1, a_1) = -1; c(1, a_2) = 2; c(1, a_3) = 4; c(2, a_1) = 7; c(2, a_2) = -4; c(2, a_3) = 3;$$

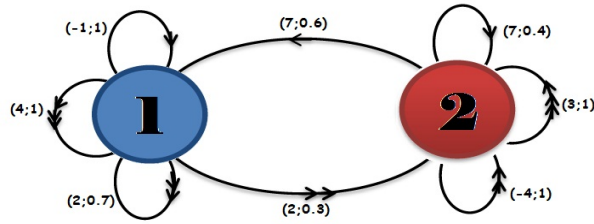


FIGURE 2.1 – Graphe probabiliste de l'exemple 2.2.4.1

En appliquant l'algorithme 2 (FIA), on obtient les résultats numériques suivants :

Iteration t	$J_\gamma^t(1)$	$J_\gamma^t(2)$	$\pi^t(1)$	$\pi^t(2)$
1	-1	-4	$a_1$	$a_2$
2	-1.7	-6.8	$a_1$	$a_2$
3	-2.19	-8.76	$a_1$	$a_2$
4	-2.533	-10.132	$a_1$	$a_2$
5	-2.7731	-11.0924	$a_1$	$a_2$
6	-2.94117	-11.76468	$a_1$	$a_2$
7	-3.058819	-12.235276	$a_1$	$a_2$
8	-3.1411733	-12.5646932	$a_1$	$a_2$
9	-3.19882131	-12.79528524	$a_1$	$a_2$
10	-3.239174917	-12.95669967	$a_1$	$a_2$

TABLE 2.1 – Application de l'algorithme 2 (FIA)

Ainsi, une stratégie optimal et le vecteur valeur correspondant sont donnés par :

$$\pi = (\pi^1, \pi^2, \dots, \pi^{10}) \text{ and } J_\gamma^{10} = (-3.239174917, -12.95669967).$$

où

$$\pi^t(1) = a_1 \text{ et } \pi^t(2) = a_2 \text{ for } t=1, \dots, 10.$$

## 2.2.5 Procédures d'élimination des actions non optimales

L'approximation successive issue de la programmation dynamique est une méthode simple et couramment utilisée pour résoudre les PDMs. Elle semble souvent insoluble pour résoudre des PDMs à grande échelle en raison de sa complexité de calcul. L'élimination des actions (EA) qui est l'une des techniques utilisées pour accélérer la résolution des PDMs, réduit la taille du problème en identifiant et en éliminant les actions non optimales. Dans certains cas, la méthode des approximations successives est terminée lorsque toutes les actions sauf une par état sont éliminées. Pour cet égard, les bornes des fonctions valeurs sont l'élément clé et central de l'élimination des actions. De nouveaux termes (gain, gain relatif et gain relatif cumulatif d'action) ont été introduits pour construire des bornes plus strictes sur les fonctions valeurs et établir un algorithme amélioré d'élimination d'action.

La technique EA est utilisée pour accélérer la résolution des PDMs ; il réduit la taille du problème en identifiant et en éliminant les actions non optimales. Au cours de toute itération de l'algorithme d'approximation successive, s'il s'avère que l'action  $a$  surpasse l'action  $a'$ , où  $a, a' \in A(i)$ , alors  $a$  est une action non optimale, il n'est donc pas nécessaire de considérer  $a$  lors de la mise à jour de la fonction valeur ou politique dans les prochaines itérations. L'idée de EA est très simple et son efficacité de repose sur la façon d'identifier les actions non optimales en moins d'itérations avec un effort de calcul minimum.

Tout au long de cette sous-section, nous supposons que tous les coûts  $\{c_t(i, a), i \in S, a \in A(i), t = 1, \dots, T\}$  sont positifs et stationnaire avec stationnarité des probabilités de transition dans le temps.

On pose :

$$\tilde{c} = \max_{i \in S, a \in A(i)} |c(i, a)|$$

Considérons d'abord l'opérateur de Bellman  $\Phi : \mathbb{R}^{|S|} \rightarrow \mathbb{R}^{|S|}$  donné pour tout vecteur  $V = (V_i)_{i \in S} \in \mathbb{R}^{|S|}$  par :

$$(\Phi V)_i = \min_{a \in A(i)} \{c(i, a) + \gamma \sum_{j \in S} p_{iaj} V_j\}$$

**Lemme 2.2.5.1** Soit  $(V^n)_{n \geq 0}$  la suite de vecteurs générés par l'algorithme 2 (FIA). L'application  $\Phi$  est une contraction croissante sur  $\mathbb{R}^{|S|}$  pour la norme infinie avec  $\gamma$  est le facteur de contraction. De plus :

$$V_i^n \leq J_\gamma^T(i) \leq V_i^n + \frac{\gamma^n - \gamma^T}{1 - \gamma} \tilde{c}, \quad \text{pour tous } i \in S, 0 \leq n \leq T. \quad (2.6)$$

**Preuve 2.2.5.1** La monotonie de l'application  $\Phi$  découle de la proposition 6.3.2 par Puterman et le lemme 1.1.1 par Bertsekas. La propriété de contraction est assurée par la proposition 6.2.4 par Puterman et la proposition 1.4.5 par Bertsekas.

Ainsi, la preuve du théorème 6.2.3 par Puterman fournit le résultat suivant :

$$V_i^n - \frac{\gamma^n - \gamma^T}{1 - \gamma} \tilde{c} \leq J_\gamma^T(i) \leq V_i^n + \frac{\gamma^n - \gamma^T}{1 - \gamma} \tilde{c} \quad (2.7)$$

Puisque  $V_i^0 = 0$ ,  $i \in S$  et le fait que  $\Phi$  est une application croissante, il s'en suit

$$V_i^n \leq J_\gamma^T(i), \quad \text{for } 0 \leq n \leq T \quad (2.8)$$

Donc, des assertions (2.7) et (2.8) on déduit l'assertion demandée (2.6).  $\square$

En s'inspirant du lemme 2.2.5.1 pour établir un nouveau test pour éliminer les actions non optimales dans le théorème suivant :

**Théorème 2.2.5.1 (Nouveau test de non optimalité)** Soit  $(V^n)_{n \geq 0}$  la suite de vecteurs donnés par l'algorithme 2 (FIA). Si pour  $T \geq 2$  et  $1 \leq n \leq T - 1$

$$c(i, a') + \gamma \sum_{j \in S} p_{ia'j} V_j^n > V_i^n + \frac{\gamma^n - \gamma^T}{1 - \gamma} \tilde{c} \quad (2.9)$$

dans un instant  $n$ , alors toute stratégie markovienne qui tient en considération l'action  $a'$  à l'état  $i$  n'est jamais optimale.

**Preuve 2.2.5.2** Supposons que l'action  $a'$  est optimale à l'état  $i$  à la période  $T$ .

Donc, on aura

$$J_\gamma^T(i) = V_i^T = c(i, a') + \gamma \sum_{j \in S} p_{ia'j} V_j^{T-1} \geq c(i, a') + \gamma \sum_{j \in S} p_{ia'j} V_j^n \quad (2.10)$$

De (2.10) et (2.9), on montre que

$$J_\gamma^T(i) > V_i^n + \frac{\gamma^n - \gamma^T}{1 - \gamma} \tilde{c} \quad (2.11)$$

En combinant (2.6) et (2.11), on tire  $J_\gamma^T(i) > J_\gamma^T(i)$ . Ce qui est absurde. D'où ce qui fallait démontrer.  $\square$

Par combinaison de l'algorithme 2 (FIA) et le test précédent (2.9) du théorème 2.2.5.1, on établit l'algorithme itératif suivant :

---

**Algorithme 3 : Improved Forward Induction Algorithm (IFIA)**

---

**Input :** MDP =  $(S, A = \bigcup_{i \in S} A(i), Q = \{p_t(j/i, a)\}, c = \{c_t(i, a)\}, T, \gamma)$

**Output :** the optimal value vector, an optimal policy

```

1 Initialization :  $V_i^0 = 0$  for all  $i \in S$ 
2 for  $n=1$  to  $N$  do
3   for  $i \in S$  do
4     for  $a \in A(i)$  do
5        $Q^n(i, a) = c(i, a) + \gamma \sum_{j \in S} p_{iaj} V_j^{n-1}$ 
6        $V_i^n = \min_{a \in A(i)} Q^n(i, a)$ 
7        $\pi^{*n}(i) = \arg \min_{a \in A(i)} Q^n(i, a)$ 
8       for  $a \in A(i)$  do
9          $B^n(i, a) = Q^n(i, a) - V_i^n - \frac{\gamma^n - \gamma^N}{1 - \gamma} \tilde{c}$ 
10        if  $B^n(i, a) > 0$  then
11           $A(i) \leftarrow A(i) - \{a\}$ 
12 return  $V^N$  and  $\pi^*$ 

```

---

**Exemple 2.2.5.1** *Considérons un PDM avec les données numériques suivantes :*

$$S = \{1, 2\}; A(1) = A(2) = \{a_1, a_2, a_3\}; \gamma = 0.8; T = 20.$$

*Probabilités de transition :*

$$p_{1a_11} = 1; p_{1a_21} = 0.8; p_{1a_31} = 1; p_{1a_22} = 0.2; p_{2a_12} = 0.6; p_{2a_22} = 1; p_{2a_32} = 1; p_{2a_11} = 0.4$$

*Coûts :*

$$c(1, a_1) = 17; c(1, a_2) = 20; c(1, a_3) = 12; c(2, a_1) = 18; c(2, a_2) = 10; c(2, a_3) = 14;$$

En appliquant l'algorithme 3 (IFIA), on obtient les résultats numériques suivants en tant qu'approximations de la fonction test  $B^t(i, a)$  et le vecteur valeur optimal  $C_{0.8}^{20}$  à trois décimales après la virgule :

Iteration t	$B^t(1, a_1)$	$B^t(1, a_2)$	$B^t(1, a_3)$	$B^t(2, a_1)$	$B^t(2, a_2)$	$B^t(2, a_3)$
0	-81.847	-78.847	-86.847	-80.847	-88.847	-84.847
1	-64.247	-61.567	-69.247	-61.887	-70.847	-66.847
2	-50.167	-47.743	-55.167	-46.719	-56.447	-52.447
3	-38.903	-36.684	-43.903	-34.585	-44.927	-40.927
4	-29.892	-27.837	-34.892	-24.877	-35.711	-31.711
5	-22.683	-20.759	-27.683	-17.111	-28.338	-24.338
6	-16.916	-15.096	-21.916	-10.898	-22.440	-18.440
7	-12.302	-10.566	-17.302	-5.928	-17.721	-13.721
8	-8.611	-6.943	-13.611	-1.952	-13.947	-9.947
9	-5.658	-4.043	-10.658	1.229	-10.927	-6.927
10	-3.296	-1.724	-8.296	3.774	-8.511	-4.511
11	-1.406	0.131	-6.406	5.810	-6.578	-2.578
12	0.106	1.616	-4.894	7.438	-5.032	-1.032
13	1.315	2.803	-3.685	8.741	-3.795	0.205
14	2.283	3.753	-2.717	9.784	-2.805	1.195
15	3.057	4.513	-1.943	10.617	-2.014	1.986
16	3.676	5.121	-1.324	11.285	-1.380	2.620
17	4.171	5.607	-0.829	11.818	-0.874	3.126
18	4.568	5.996	-0.432	12.245	-0.468	3.532
19	4.885	6.308	-0.115	12.587	-0.144	3.856

TABLE 2.2 – Valeurs de  $B^t(.,.)$  par l’algorithme IFIA

Le vecteur valeur optimal  $J_\gamma^T$  à l’itération  $T = 20$  pour  $\gamma = 0.8$  est :

$$J_{0.8}^{20} = (59.308, 49.423)$$

**Interprétation de la table 2.2 :**

La couleur rouge indique que l’action sélectionnée à l’itération  $t$  et à l’état  $i$  doit être éliminé des calculs.

– À l’itération **9** on doit éliminer l’action  $a_1$  à l’état 2 du fait que  $B^9(2, a_1) > 0$ .

Donc, le nouveau espace d'actions disponibles pour l'état 2 après la 8<sup>ème</sup> itération sera  $A(2) = \{a_2, a_3\}$ .

– À l'itération **11** on doit éliminer l'action  $a_2$  à l'état 1 car  $B^{11}(1, a_2) > 0$ . Le nouveau espace d'actions disponibles pour l'état 1 après la 10<sup>ème</sup> itération sera  $A(1) = \{a_1, a_3\}$ .

– À l'itération **12** on doit éliminer l'action  $a_1$  à l'état 1 car  $B^{12}(1, a_1) > 0$ . Ainsi, le nouveau espace d'actions disponibles pour l'état 1 après la 11<sup>ème</sup> itération sera  $A(1) = \{a_3\}$ .

– À l'itération **13** on doit éliminer l'action  $a_3$  à l'état 2 car  $B^{13}(2, a_3) > 0$ . Le nouveau espace d'actions disponibles pour l'état 2 après la 12<sup>ème</sup> itération sera  $A(2) = \{a_2\}$ .

En fin, les nouveaux espaces d'actions générés par l'algorithme 3 (IFIA) sont :

$$A(1) = \{a_3\} \text{ après la } 11^{\text{ème}} \text{ itération et } A(2) = \{a_2\} \text{ après } 12^{\text{ème}} \text{ itération.}$$

Le pourcentage des actions éliminées par l'algorithme (IFIA) est donc :

$$\frac{4}{6}\% = 66.67\%$$

La figure ci-dessous donne le pourcentage cumulé des actions éliminées à chaque itération en fonction des valeurs du facteur d'actualisation  $\gamma$  :

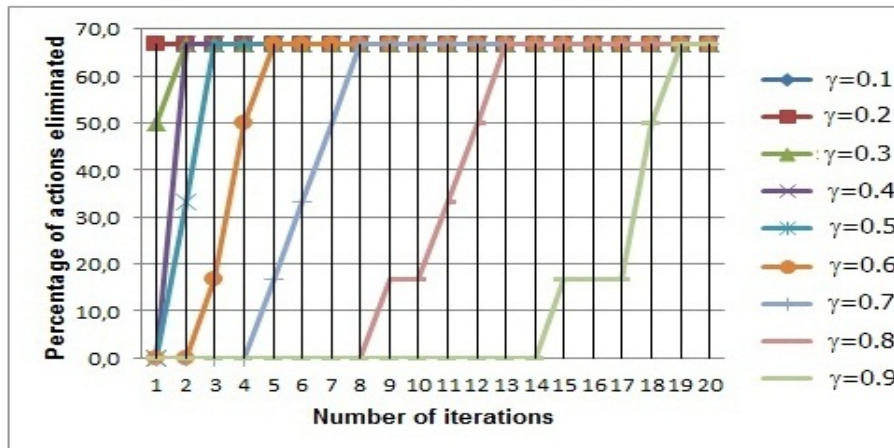


FIGURE 2.2 – Facteur d'actualisation et actions éliminées

**Remarque 2.2.5.1** *Le pourcentage cumulé des actions éliminées décroît au fur et à mesure le facteur d'actualisation  $\gamma$  s'approche de la valeur 1.*

## 2.3 Modèles sur horizon infini

Les processus décisionnels de Markov (PDMs) fournissent un cadre mathématique pour modéliser la prise de décision dans des situations où les résultats sont en partie aléatoires et en partie sous le contrôle du décideur. Ils sont issus de l'étude du contrôle optimal stochastique dans les années 1950 et sont restés d'une importance clé dans ce domaine depuis cette période. Leur théorie ne cesse de se développer au cours des dernières décennies pour s'adapter à un plus large éventail de problèmes et a conduit à une multitude d'idées algorithmiques et d'analyses théoriques communes. Aujourd'hui, les PDMs sont utilisés dans divers domaines, notamment la robotique, le contrôle automatisé, la planification, l'économie et la fabrication.

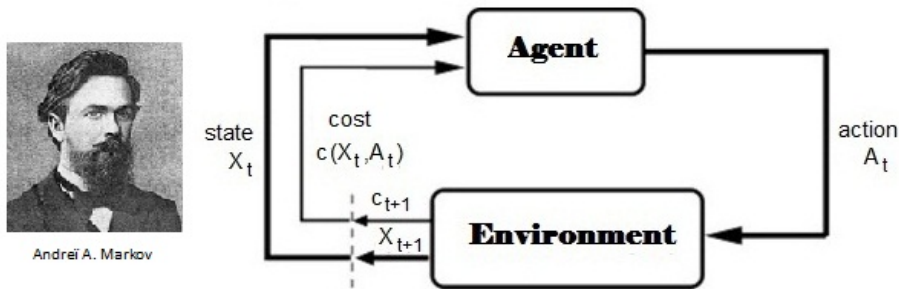


FIGURE 2.3 – Modèle d'un PDM

Un PDM se compose d'un agent et d'un environnement avec lesquels l'agent interagit. Ces interactions se produisent sur une séquence de pas de temps discrets  $t$  ; à chaque pas de temps  $t$ , l'agent perçoit l'état de l'environnement  $X_t$  et sélectionne une action  $A_t$  à effectuer. L'environnement réagit à l'action en effectuant une transition vers un nouvel état  $X_{t+1}$  et paye un coût scalaire  $r(X_t, A_t)$ . L'objectif de l'agent est de minimiser le montant total des coûts qu'il perd de ses interactions avec l'environnement. La dynamique de l'environnement est stationnaire et le signal d'état doit contenir toutes les informations pertinentes, mais sans contrainte autrement.

### 2.3.1 Description du modèle et critère d'optimalité à horizon infini

Un PDM est généralement désigné par un quintuple

$$(S, A = \bigcup_{i \in S} A(i), Q = \{p_t(j/i, a)\}, c = \{c_t(i, a)\}, \gamma)$$

où  $S$  est l'espace d'états,  $A$  est l'espace d'action,  $p(j/i, a)$  est une fonction qui détermine la probabilité que l'action  $a$  dans l'état  $i$  au moment  $t$  conduise à l'état  $j$  au temps  $t$ ,  $c(i, a)$  est une fonction qui renvoie le coût immédiat payé après le passage à l'état  $i$  à partir de des états précédents et  $\gamma$  est le facteur d'actualisation. On dit qu'un PDM est fini si l'espace des états et les ensembles d'actions sont finis. Dans cet exposé, on s'intéressera au PDMs finis avec stationnarité des probabilités de transition et des coûts mis en jeu.

Les principales classes de PDMs étudiées dans la littérature sont répertoriées et le problème considéré est défini. Certains concepts de base essentiels pour poursuivre ce travail sont discutés. Les algorithmes les plus couramment utilisés pour résoudre les PDMs, à savoir : l'approximation successive, l'itération de politique et la programmation linéaire sont discutés. La complexité du problème est mise en évidence, enfin, une revue de la littérature est présentée.

Pour que l'agent puisse minimiser le coût de son interaction avec l'environnement, il doit être capable d'évaluer la valeur d'un état et de mettre en œuvre une cartographie des états aux probabilités de sélection de chaque action possible à chaque pas de temps. Ce mappage est appelé stratégie de l'agent et est noté  $\pi$ , où  $\pi_{i,a}^t$  est la probabilité que l'action  $a$  soit sélectionnée au temps  $t$  si nous sommes dans l'état  $i$  au temps  $t$ . La valeur estimée d'un état est définie en termes de coûts futurs auxquels on peut s'attendre. Bien entendu, les coûts auxquels l'agent peut s'attendre à l'avenir dépendent des actions qu'il entreprendra. En conséquence, sa fonction de valeur est définie par rapport à une stratégie particulière. Par conséquent, la valeur de l'état  $i$  sous la stratégie  $\pi$ , notée  $J_\gamma(i, \pi)$ , est le coût espéré en partant de l'état  $i$  et suivant la stratégie  $\pi$  par la suite et peut être définie formellement par :

$$J_\gamma(i, \pi) := \mathbb{E}_\pi^i \left[ \sum_{t=1}^{\infty} \gamma^{t-1} r(X_t, A_t) \right] \quad (2.12)$$



où  $\mathbb{E}_\pi^i(\cdot)$  désigne la valeur espérée étant donné que l'agent choisit la stratégie  $\pi$  et  $\gamma$  est le taux d'actualisation qui détermine combien nous valorisons le coût futur par rapport au coût immédiat.

Puisque

$$\left| \sum_{t=1}^{\infty} \gamma^{t-1} c(X_t, A_t) \right| \leq c \sum_{t=1}^{\infty} \gamma^{t-1} = \frac{c}{1-\gamma}$$

où

$$c = \max_{i \in S, a \in A(i)} |c(i, a)|$$

Et en vertu du théorème de la convergence dominée de Lebesgue, on peut écrire

$$J_\gamma(i, \pi) := \sum_{t=1}^{\infty} \gamma^{t-1} \mathbb{E}_\pi^i[r(X_t, A_t)] = \sum_{t=1}^{\infty} \gamma^{t-1} \sum_{j \in S} \sum_{a \in A(j)} \mathbb{P}_\pi^i(X_t = j, A_t = a) c(j, a) \quad (2.13)$$

Soit  $\pi = (\pi^1, \pi^1, \dots) \in \Pi$ . En notation vectorielle, on a

$$J_\gamma(\pi) = \sum_{t=1}^{\infty} \gamma^{t-1} P(\pi^1) P(\pi^2) \dots P(\pi^{t-1}) c(\pi^t)$$

En particulier, une stratégie stationnaire  $\pi = (\pi^1, \pi^1, \dots)$  vérifie

$$J_\gamma(\pi) = \sum_{t=1}^{\infty} \gamma^{t-1} P(\pi)^{t-1} c(\pi) \quad (2.14)$$

Or :

$$\sum_{t=1}^{\infty} \gamma^{t-1} P(\pi)^{t-1} = (I - \gamma P(\pi))^{-1}$$

avec  $I$  est la matrice unité d'ordre  $|S|$ .

Par suite,  $J_\gamma(\pi)$  est l'unique solution de l'équation

$$(I - \gamma P(\pi))X = c(\pi), \quad X \in \mathbb{R}^{|S|} \quad (2.15)$$

Lorsque nous résolvons un PDM, nous cherchons à obtenir la stratégie optimale, qui est définie comme celle avec un coût espéré inférieur ou égal à toutes les autres stratégies pour tous les états.

On dit qu'une stratégie  $\pi^* \in \Pi$  est optimal si

$$J_\gamma(\pi^*) = \inf_{\pi \in \Pi} J_\gamma(\pi)$$

Il peut y avoir plus d'une stratégie optimale. Le cadre des PDMs est abstrait, flexible et fournit les outils nécessaires à la solution de nombreux problèmes importants de

la vie réelle. La flexibilité du cadre lui permet non seulement d'être appliqué à de nombreux problèmes différents, mais également de nombreuses manières différentes. Par exemple, les pas de temps peuvent se référer à des étapes successives arbitraires de prise de décision et d'action. Les actions peuvent être toutes les décisions que nous voulons apprendre à prendre et l'état peut contenir tout ce qui pourrait être utile pour les prendre.

### **2.3.2 Résolution d'un PDM par l'approche de la programmation dynamique**

La technique de programmation dynamique (PD) repose sur une idée simple basée sur le principe d'optimalité. Le nom est dû à Bellman, qui a beaucoup contribué à la vulgarisation de PD et à sa transformation en un outil systématique.

Le principe d'optimalité de Bellman suggère qu'une stratégie optimale peut être construite progressivement, en construisant d'abord une stratégie optimale pour le "sous-problème de queue" impliquant la dernière étape, puis en étendant la stratégie optimale au "sous-problème de queue" impliquant les deux dernières étapes, et on continue de cette façon jusqu'à ce qu'une stratégie optimale pour l'ensemble du problème soit élaborée. Ainsi, l'algorithme PD est basé sur cette idée : il procède séquentiellement, en résolvant tous les sous-problèmes de queue d'une durée donnée, en utilisant la solution des sous-problèmes de queue de durée plus courte.

#### **1) Itération de la valeur (VI) :**

L'itération de valeur est un algorithme de la programmation dynamique qui adopte une approche itérative pour obtenir la stratégie optimale. Plutôt que de manipuler directement la stratégie, il obtient la stratégie optimale en calculant la fonction de valeur optimale en utilisant l'équation de Bellman. Pour ce faire, il parcourt l'espace d'états et attribue à chaque état la valeur estimée maximale basée sur la valeur actualisée de ses états voisins. Ce calcul itératif se poursuit jusqu'à ce que le changement maximal de valeur pour tous les états de chaque balayage soit inférieur à un petit nombre positif prédéfini désigné par  $\epsilon$  (condition d'arrêt de l'itération). Plus la valeur de  $\epsilon$  est petite, plus la précision de l'algorithme est élevée. L'itération de

valeur nécessite que chaque état ne soit traité qu'une seule fois dans chaque balayage dans l'espace d'états et élimine ainsi l'un des inconvénients de l'itération de politique, qui est l'évaluation de politique qui peut nécessiter plusieurs balayages dans l'espace d'états.

• **Préliminaire :**

Pour déterminer une stratégie optimale et la fonction de valeur optimale, l'équation d'optimalité de Bellman suivante

$$V_i = \min_{a \in A(i)} \{c(i, a) + \gamma \sum_{j \in S} p(j/i, a) V_j\}, \quad i \in S \quad (2.16)$$

joue un rôle central pour calculer la fonction de valeur optimale d'un PDM.

On considère l'opérateur de Bellman  $\Phi : \mathbb{R}^{|S|} \rightarrow \mathbb{R}^{|S|}$  suivant défini par

$$\Phi(V)_i = \min_{a \in A(i)} \{c(i, a) + \gamma \sum_{j \in S} p(j/i, a) V_j\}, \quad i \in S \quad (2.17)$$

**Théorème 2.3.2.1** *On munit l'ensemble  $\mathbb{R}^{|S|}$  de la norme infinie  $\|\cdot\|_\infty$  définie par  $\|V\|_\infty = \max_{i \in S} |V_i|$  pour tout  $V \in \mathbb{R}^{|S|}$ .*

*L'opérateur de Bellmann  $\Phi$  est une application  $\gamma$ -contractante et monotone au sens de l'ordre partiel  $\tilde{\leq}$  défini sur  $\mathbb{R}^{|S|}$  par :  $\forall V, W \in \mathbb{R}^{|S|}, V \tilde{\leq} W \Leftrightarrow V_i \leq W_i, \forall i \in S$ .*

**Preuve 2.3.2.1** 1) *La preuve de la première assertion découle de l'inégalité*

$$|\min_{i \in I} x_i - \min_{i \in I} y_i| \leq \max_{i \in I} |x_i - y_i|,$$

*pour tout ensemble non vide fini et  $x_i, y_i \in \mathbb{R}, i \in I$ .*

2) *Soient  $V, W \in \mathbb{R}^{|S|}$  tels que  $V \tilde{\leq} W$  et  $i \in S$  fixé. Puisque  $p(j/i, a) \geq 0$  pour tout  $j \in S$  et  $a \in A(i)$ , on aura*

$$c(i, a) + \gamma \sum_{j \in S} p(j/i, a) V_j \leq c(i, a) + \gamma \sum_{j \in S} p(j/i, a) W_j$$

*Par passage au min on déduit que  $\Phi(V) \tilde{\leq} \Phi(W)$ .* □

On sait bien que l'ensemble  $\mathbb{R}^{|S|}$  muni de la norme  $\|\cdot\|_\infty$  est un espace de Banach. Par suite, la théorie des applications contractantes et le théorème [2.3.2.1](#) permettent d'établir le théorème suivant :

**Théorème 2.3.2.2** *La fonction de valeur optimal du PDM est l'unique point fixe de l'opérateur de Bellman  $\Phi$ .*

Grâce au théorème 2.3.2.2, Howard [40] a proposé l'algorithme de l'itération de la valeur pour résoudre les PDMs liés au gain actualisé. Par la suite, C. Derman [35] a étendu cet algorithme pour résoudre les PDMs liés au coût actualisé.

• **Algorithme :**

---

**Algorithme 4 : Value Iteration Algorithm (PJ)**

---

**Input :** MDP =  $(S, A = \bigcup_{i \in S} A(i), Q = \{p_t(j/i, a)\}, c = \{c_t(i, a)\}, \gamma)$

**Output :** An optimal policy, The optimal value vector

```

1 Initialize  $J^0 \in \mathbb{R}^{|S|}$ , specify  $\varepsilon > 0$ 
2  $\mathbf{t} \leftarrow \mathbf{0}$ 
3 repeat
4   for  $i \in S$  do
5     for  $a \in A(i)$  do
6        $Q^n(i, a) = c(i, a) + \gamma \sum_{j \in S} p(j/i, a) J_j^t$ 
7      $J_i^{t+1} = \min_{a \in A(i)} Q^n(i, a);$ 
8    $\mathbf{t} \leftarrow \mathbf{t} + \mathbf{1}$ 
   until  $\|J^{t+1} - J^t\|_\infty < \frac{1-\gamma}{2\gamma} \varepsilon;$ 
9 for  $i \in S$  do
10   $\pi^*(i) \in \arg \min_{a \in A(i)} Q^n(i, a);$ 
11 return  $\pi^*, J^n.$ 

```

---

• **Complexité :**

La complexité en temps de cet algorithme dépend du coût de chaque itération et du nombre d'itérations. Chaque itération exige  $|A||S|^2$  opérations. Dans le pire des cas, le nombre d'itérations nécessaires pour converger est polynomial en  $|S|$ ,  $|A|$ ,  $\gamma$  et le nombre total de bits nécessaires pour représenter les données du problème.

Dans ce contexte, il y avait plusieurs méthodes itératives qui ont été proposées pour accélérer les itérations de l'algorithme de l'itération de la valeur. Dans la suite, on

donne un aperçu sur les principaux schémas. On suppose que  $S = \{1, 2, \dots, |S|\}$ .

*i) Pre-Jacobi (PJ)*

$$V_i^{n+1} = A_{PJ}V^n(i) = \min_{a \in A(i)} \{c(i, a) + \gamma \sum_{j=1}^{|S|} p(j/i, a)V_j^n\} \quad (2.18)$$

Ce schéma est le schéma standard de l'algorithme de l'itération de la valeur. Il est dite Pre-Jacobi (PJ) par analogie avec la notation des systèmes d'équations linéaires.

*ii) Jacobi (J)*

$$V_i^{n+1} = A_JV^n(i) = \min_{a \in A(i)} \{[c(i, a) + \gamma \sum_{j \neq i} p_{iaj}V_j^n]/[1 - \gamma p(i/i, a)]\} \quad (2.19)$$

Pour calculer une composante du vecteur  $V^{n+1}$ , on a besoin de  $|S| - 1$  composantes de  $V^n$ , donc il faut garder entièrement le vecteur  $V^n$  dans la mémoire lors du calcul de  $V^{n+1}$ . On a donc besoin de  $2|S|$  mémoires pour stocker les vecteurs  $V^n$  et  $V^{n+1}$ .

*iii) Pre-Gauss-Seidel (PGS)*

$$V_i^{n+1} = A_{PGS}V^n(i) = \min_{a \in A(i)} \{c(i, a) + \gamma \sum_{j=1}^{i-1} p(j/i, a)V_j^{n+1} + \gamma \sum_{j=i}^{|S|} p(j/i, a)V_j^n\} \quad (2.20)$$

*iv) Gauss-Seidel (GS)*

$$\begin{aligned} V_i^{n+1} &= A_{GS}V^n(i) \\ &= \min_{a \in A(i)} \{[c(i, a) + \gamma \sum_{j=1}^{i-1} p(j/i, a)V_j^{n+1} + \gamma \sum_{j=i+1}^{|S|} p(j/i, a)V_j^n]/[1 - \gamma p(i/i, a)]\} \end{aligned} \quad (2.21)$$

Dès que la  $i^{\text{ème}}$  composante de  $V^{n+1}$  est calculée, la  $i^{\text{ème}}$  composante de  $V^n$  devient inutile. On peut l'écraser et la remplacer par la  $i^{\text{ème}}$  composante de  $V^{n+1}$  dès que celle-ci est calculée. Un avantage de cette méthode est qu'elle ne nécessite plus que  $|S|$  mémoires, les deux vecteurs  $V^n$  et  $V^{n+1}$  pouvant être stockés à la même place.

*v) Successive Over-Relaxation (SOR)*

$$V_i^{n+1} = \omega[A_{GS}V^n(i)] + (1 - \omega)V_i^n \quad (2.22)$$

où  $\omega \in (0, 2)$  est le paramètre de relaxation.

**2) Itération de la politique (PI) :**

L'itération de politique (PI) ou la stratégie améliorée est un algorithme de la programmation dynamique qui manipule la stratégie directement lorsqu'elle est utilisée

pour calculer la stratégie optimale. Il part initialement d'une stratégie arbitraire  $\pi_0$ . Chaque itération comporte deux phases : la première phase d'évaluation de la stratégie en cours consiste à résoudre un système linéaire dérivant de l'équation de Bellman et la deuxième phase d'amélioration met à jour la stratégie courante en utilisant la valeur calculée lors de la phase d'évaluation afin de trouver de meilleures stratégies. Ceci est fait en considérant un écart par rapport à notre stratégie actuelle dans l'état  $i$  où nous voulons savoir si nous devons ou non changer la stratégie pour choisir de manière déterministe une action différente de celle selon  $\pi(i)$ . Nous pouvons déterminer si ce changement de stratégie nous mènera à une meilleure stratégie en sélectionnant  $a$  en  $i$  puis en suivant la stratégie existante  $\pi$ . Si nous découvrons que la valeur de cette nouvelle stratégie est inférieure à celle de notre stratégie existante, alors nous avons réussi à améliorer notre stratégie. Il est naturel que cette ligne de pensée s'étende non seulement pour considérer un changement pour une seule action dans un seul état mais pour toutes les actions dans tous les états.

Nous évaluerions ensuite chaque action dans chaque état et nous sélectionnerions les actions qui produisent le minimum coût. Ce processus d'élaboration d'une nouvelle stratégie qui améliore une stratégie originale, en la rendant gourmande ou presque avide par rapport à la fonction de valeur, s'appelle l'amélioration de la stratégie. Une fois qu'une stratégie  $\pi_{t-1}$  a été améliorée à l'itération  $t - 1$  en utilisant la fonction de valeur correspondante  $J_\gamma(\pi_{t-1})$  pour produire une meilleure stratégie  $\pi_t$ , nous pouvons calculer sa fonction de valeur  $J_\gamma(\pi_t)$  et l'améliorer à nouveau pour produire une politique  $\pi_{t+1}$  encore meilleure où chaque politique est garantie d'être une amélioration stricte par rapport à la précédente.

Comme un PDM fini n'a qu'un nombre fini de politiques, ce processus doit converger vers une politique optimale (soit  $\pi_{t+1}(i) = \pi_t(i), \forall i \in S$ ) et une fonction de valeur optimale en un nombre fini d'itérations. Cette façon d'entrelacer l'évaluation des politiques avec l'amélioration des politiques s'appelle l'itération des politiques et est un algorithme fondamental dans la résolution des PDMs.

• **Algorithme :**

---

**Algorithme 5 : Policy Iteration Algorithm**

---

**Input :** MDP =  $(S, A = \bigcup_{i \in S} A(i), Q = \{p(j/i, a)\}, c = \{c(i, a)\}, \gamma)$

**Output :** An optimal policy, The optimal value vector

```

1 Initialize  $\pi_0 \in \Pi$ 
2  $t \leftarrow 0$ 
3 repeat
    Evaluation phase :
    Solve
4      $J_i^t = c(i, a) + \gamma \sum_{j \in S} p(j/i, \pi_t(i)) J_j^t$ 
    Improvement phase :
5     for  $i \in S$  do
6          $\pi_{t+1}(i) \in \arg \min_{a \in A(i)} \{c(i, a) + \gamma \sum_{j \in S} p(j/i, \pi_t(i)) J_j^t\};$ 
7      $t \leftarrow t + 1;$ 
    until  $\pi_t = \pi_{t+1};$ 
8 return  $\pi_t, J^t$ 

```

---

• **Complexité :**

La complexité de l'algorithme (PI) est en  $O(|A||S|^2) + O(|S|^3)$  par itération avec un nombre d'itération avec un nombre maximum d'itération polynomial en  $|S|$  et  $|A|$  à  $\gamma$  constant.

L'algorithme de l'itération de la politique (PI) demande moins d'itérations pour converger vers la stratégie optimale que celui de l'itération de la valeur (VI). Effectivement, l'algorithme (VI) continue à itérer tant que la valeur de la stratégie peut être améliorée, et ce même si la politique ne change pas. Néanmoins, une itération de (PI) consomme plus de temps qu'une itération de (VI).

### 2.3.3 Résolution d'un PDM par l'approche de la programmation linéaire

Cette section explore la relation entre les problèmes de décision de Markov actualisés et les programmes linéaires (PL). À l'heure actuelle, la programmation linéaire ne s'est pas avérée être une méthode efficace pour résoudre les PDMs actualisés de grande taille. Cependant, les innovations dans les algorithmes LP au cours de la dernière décennie pourraient changer cela. Nous allons faire appel à la formulation de la programmation linéaire en raison de sa facilité d'analyse de sensibilité.

Les méthodes des approximations successives (VI) et d'amélioration des stratégies (PI) peuvent être considérées comme des méthodes de programmation dynamique. Ainsi, il est quelque peu surprenant que la méthode de programmation linéaire puisse également être appliquée. La programmation linéaire est une autre manière de résoudre les problèmes PDMs. Les formulations de programmation linéaire largement utilisées sont dues à Derman, Puterman et Kallenberg. Pour cela, considérons le problème de programmation linéaire :

$$\max \sum_{j \in S} \beta_j v_j \quad (2.23)$$

$$s.c \left\{ v_i \leq c(i, a) + \gamma \sum_{j \in S} p(j/i, a) v_j, \quad i \in S, a \in A(i) \right. \quad (2.24)$$

où  $\beta_j > 0$ ,  $j \in S$  sont des nombres réels donnés avec  $\sum_{j \in S} \beta_j = 1$ . Ce qui permet que la suite  $(\beta_j)_{j \in S}$  peut être interprétée comme une distribution de probabilité sur l'espace d'états  $S$ .

Intuitivement, il est clair que le vecteur fonction de valeur optimal  $J_\gamma^*$  est le plus petit vecteur solution réalisable de ce programme linéaire. C'est la propriété clé de l'approche de programmation linéaire pour calculer le vecteur de valeur. Il s'avère qu'une stratégie optimale peut être obtenue à partir du programme linéaire dual. Il y a une correspondance biunivoque entre les politiques stationnaires et les solutions réalisables du programme dual, de sorte que les points extrêmes correspondent à des stratégies déterministes.

$$\max \sum_{i \in S} \sum_{a \in A(i)} c(i, a) z_{i,a} \quad (2.25)$$



$$s.c \left\{ \begin{array}{l} \sum_{i \in S} \sum_{a \in A(j)} (\delta_{i,j} - \gamma p(j/i, a)) z_{i,a} = \beta_j, \quad j \in S \\ z_{i,a} \geq 0, \quad i \in S, \quad a \in A(i) \end{array} \right. \quad (2.26)$$

où la quantité  $z_{i,a}$  peut être vue comme une mesure d'occupation actualisée de  $(i, a)$ , et  $\delta_{x,y} = 1$  si  $x = y$  et 0 sinon.

Un vecteur  $v \in \mathbb{R}^{|S|}$  est dit  $\gamma$  sous-harmonique si

$$v_i \leq c(i, a) + \gamma \sum_{j \in S} p(j/i, a) v_j, \quad \forall i \in S, a \in A(i) \quad (2.27)$$

**Théorème 2.3.3.1** *Le vecteur fonction de valeur optimal  $J_\gamma^*$  est le plus grand vecteur (composante par composante)  $\gamma$  sous-harmonique.*

**Preuve 2.3.3.1** *On a :*

$$\begin{aligned} J_\gamma^*(i) &= \min_{a' \in A(i)} \{c(i, a') + \gamma \sum_{j \in S} p(j/i, a')\} J_\gamma^*(j) \\ &\leq c(i, a) + \gamma \sum_{j \in S} p(j/i, a) J_\gamma^*(j), \quad \forall i \in S, a \in A(i) \end{aligned}$$

Donc  $J_\gamma^*$  est un vecteur  $\gamma$  sous-harmonique.

Soit  $v \in \mathbb{R}^{|S|}$  un vecteur  $\gamma$  sous-harmonique. Donc en notation vectorielle on aura

$$v \leq c(\pi) + \gamma P(\pi)v, \quad \forall \pi \in \Pi^{MD} \quad (2.28)$$

En particulier, l'inéquation (2.28) reste vérifiée pour la stratégie optimale déterministe  $\pi^*$  fournie par l'algorithme (IV).

Puisque

$$(I - \gamma P(\pi))^{\infty} = \sum_{t=0}^{\infty} \gamma^t P^t(\pi) \geq 0$$

on aura

$$v \leq c(\pi^*) + \gamma P(\pi^*)v \Leftrightarrow v \leq (I - \gamma P(\pi^*))^{-1} c(\pi^*) = J_\gamma^*$$

D'où ce qui fallait démontrer. □

**Corollaire 2.3.3.1** *Le vecteur fonction de valeur optimal  $J_\gamma^*$  est l'unique solution optimal du problème de la programmation linéaire (2.23) et (2.24).*

**Preuve 2.3.3.2** *La preuve découle du théorème 2.3.3.1.* □

Par le corollaire 2.3.3.1, le vecteur de valeur  $J_\gamma^*$  peut être trouvé comme solution optimale du programme linéaire (2.23) et (2.24). Ce programme ne donne pas une stratégie optimale. Cependant, une stratégie optimale peut être obtenue à partir de la solution du programme dual (2.25) et (2.26).

Soit  $\{z_{i,a}\}_{i \in S, a \in A(i)}$  une solution de programme dual. Une stratégie optimale  $\pi^*$  est donnée par :

$$\pi_{i,a}^* = \frac{z_{i,a}}{\sum_{a \in A(i)} z_{i,a}}, \quad i \in S, \quad a \in A(i)$$

Ainsi, le vecteur fonction de valeur optimal du PDM peut être calculé par la relation vectorielle suivante :

$$J_\gamma^* = (I - \gamma P(\pi^*))^{-1} c(\pi^*)$$

### Cas d'un PDM avec des contraintes au niveau des récompenses :

Soit un PDM contraint défini par le modèle suivant :

$$(S, A = \bigcup_{i \in S} A(i), Q = \{p(j/i, a)\}, c = \{c(i, a)\}, \{r_k c(i, a)\}_{k=1, \dots, K}, \{\alpha_k\}_{k=1, \dots, K}, \gamma, i_0)$$

L'objectif est de déterminer une stratégie optimale  $\pi^*$  qui minimise les coûts actualisés cumulatifs espérés tandis que la récompense actualisée cumulative espérée est limitée.

$$\begin{aligned} \min_{\pi \in \Pi} \mathcal{C}_{i_0}(\pi) &= \mathbb{E}_\pi^{i_0} \left[ \sum_{t=1}^{\infty} \gamma^{t-1} c(X_t, A_t) \right] \\ \text{s.c. } \mathcal{R}_{i_0}^k(\pi) &= \mathbb{E}_\pi^{i_0} \left[ \sum_{t=1}^{\infty} \gamma^{t-1} r_k(X_t, A_t) \right] \geq \alpha_k \\ &\quad \forall k = 1, \dots, K \end{aligned}$$

Pour résoudre ce problème suivant :

$$c_i^{i_0} = \min \sum_{i \in S} \sum_{a \in A(i)} c(i, a) z_{i,a}^{i_0}$$

Sous les contraintes

$$\sum_{i \in S} \sum_{a \in A(i)} (\delta_{i,j} - \gamma p(j/i, a)) z_{i,a}^{i_0} = \delta_{i_0,j}, \quad j \in S$$

$$\sum_{i \in S} \sum_{a \in A(i)} r_k(i, a) z_{i,a}^{i_0} \geq \alpha_k, \quad k = 1, \dots, K$$

$$z_{i,a}^{i_0} \geq 0, \quad i \in S, a \in A(i)$$

### 2.3.4 Procédures d'élimination des actions non optimales

Les processus de Markov sont l'outil naturel qui peut modéliser avec succès des problèmes de décision. À cet égard, nous sommes face aux problèmes de décision de Markov (PDMs). Pour réduire les efforts de complexité de calcul. Plusieurs tests de sous-optimalité ont été suggérés en déterminant des bornes appropriées (bornes inférieure et supérieure) sur la performance du système. Lorsque le critère d'actualisation est appliqué dans les PDMs avec des états finis et des espaces d'action sur un horizon de planification infini, les auteurs ont fourni des différents résultats dans le cadre des procédures d'élimination d'actions. L'objectif principal est de réduire la taille des espaces d'action à chaque époque si cela est possible. Ce qui permet ainsi d'accélérer certains algorithmes utilisés afin de calculer la fonction de valeur optimale. Dans ce contexte, lorsque des actions sont identifiées comme des décisions non optimales, elles seront immédiatement éliminées de la planification. Plusieurs tests



FIGURE 2.4 – Complexité

sont proposés pour maximiser le gain total actualisé sur l'horizon infini. MacQueen [15-16] et Porteus [17] ont été les premiers auteurs à apparaître dans la littérature sur les procédures d'élimination d'action. Ils ont donné des tests pour identifier les actions non optimales en utilisant la détermination des bornes inférieure et supérieure sur le gain totale actualisé sur l'horizon de planification infini. Ensuite, il est apparu une multitude de résultats d'autres auteurs tels que Grinold [18], Puterman

et Shin [19] et White [20].

Du côté du critère du coût actualisé sur l'horizon de planification infini, Sladky [21] a proposé un test réservé uniquement aux PDMs à coûts positifs. Ensuite, les travaux de Semmouri et Jourhmane [22-23] et [24] sont apparus dans la même littérature. Puisqu'ils ont transformé tous les coûts en coûts positifs, ils ont fourni un nouveau test dans le cadre de l'extension du test de Sladky. Malgré ces progrès considérables, les résultats concernant l'élimination des actions sont presque rares du côté du critère de coût actualisé sur l'horizon de planification infini.

### Le premier test d'identification des décisions non optimales :

Dans cette sous-section, nous supposons que tous les coûts  $\{c(i, a)\}_{i \in S, a \in A(i)}$ , sont positifs. On pose  $\tilde{c} := \max_{i \in S, a \in A(i)} |c(i, a)|$ .

**Lemme 2.3.4.1** *Soit  $(J^n)_{n \geq 0}$  la suite de vecteurs engendré par l'algorithme 4 (PJ-VI). Alors :*

$$J_i^n - \frac{\gamma^n}{1-\gamma} \tilde{c} \leq J_i^n \leq J_\gamma^* \leq J_i^n + \frac{\gamma^n}{1-\gamma} \tilde{c}, \quad i \in S, n \geq 0 \quad (2.29)$$

**Preuve 2.3.4.1** *La propriété de contraction et la monotonie de l'opérateur de Bellman  $\Phi$  achèvent la preuve.  $\square$*

**Théorème 2.3.4.1 (Le premier test de non optimalité)** *Soit  $(J^n)_{n \geq 0}$  la suite de vecteurs donnés par l'algorithme 4 (PJ-VI). Si*

$$c(i, a') + \gamma \sum_{j \in S} p(j/i, a') J_j^n > J_i^n + \frac{\gamma^n}{1-\gamma} \tilde{c} \quad (2.30)$$

*a lieu dans un moment  $n$ , alors toute stratégie qui utilise l'action  $a'$  n'est pas optimale.*

**Preuve 2.3.4.2** *Supposons par l'absurde que l'action  $a'$  est optimale à l'état  $i$  and in stage  $n$ .*

*On a :*

$$J_\gamma^*(i) = r(i, a') + \gamma \sum_{j \in S} p(j/i, a') J_\gamma^*(j) \quad (2.31)$$

De (2.29) et (2.31), il s'en suit

$$J_\gamma^*(i) \geq r(i, a') + \gamma \sum_{j \in S} p(j/i, a') J_j^n \quad (2.32)$$

En combinant (2.32), (2.30) et (2.29), On aboutit à :  $\mathcal{C}_\gamma^* > \mathcal{C}_\gamma^*$ .

Ce qui est impossible. □

En combinant l'algorithme 4 (VIA) et le théorème 2.3.4.1 on établit le nouveau itératif algorithme.

---

**Algorithme 6 : Improved VIA**

---

**Input :** MDP =  $(S, A = \bigcup_{i \in S} A(i), Q = \{p(j/i, a)\}, c = \{c(i, a)\}, \gamma)$

**Output :** An optimal policy, An optimal value vector

```

1 Initialize  $J_i^0 = 0, i \in S$ , specify  $\varepsilon > 0$ 
2  $n = 0$ 
3 repeat
4   for  $i \in S$  do
5     for  $a \in A(i)$  do
6        $Q^n(i, a) = c(i, a) + \gamma \sum_{j \in S} p(j/i, a) J_j^n$ ;
7        $B^n(i, a) = Q^n(i, a) - J_i^n - \frac{\gamma^n}{1-\gamma} \tilde{c}$ ;
8       if  $B^n(i, a) > 0$  then
9          $A(i) = A(i) - \{a\}$ ;
10       $J_i^{n+1} = \min_{a \in A(i)} Q^n(i, a)$ ;
11     $n = n + 1$ 
12  until  $\|J^{n+1} - J^n\|_\infty < \frac{1-\gamma}{2\gamma} \varepsilon$ ;
13  for  $i \in S$  do
14     $\pi^*(i) \in \arg \min_{a \in A(i)} \{c(i, a) + \gamma \sum_{j \in S} p(j/i, a) J_j^n\}$ ;
15  return  $\pi^*, J^n$ .
```

---

**Généralisation :**

On va transformer les coûts quelconques (positifs ou négatifs) en coûts positifs par

la transformation suivante :

$$\tilde{c}(i, a) := \tilde{c} + c(i, a) \geq 0, \quad i \in S, \quad a \in A(i) \quad (2.33)$$

Pour tous  $i \in S$ ,  $\pi \in \Pi$ , on note

$$\begin{aligned} \tilde{J}_\gamma(i, \pi) &:= \mathbb{E}_\pi^i \left[ \sum_{t=1}^{\infty} \gamma^{t-1} \tilde{c}(X_t, A_t) \right] \\ &= \sum_{t=1}^{\infty} \gamma^{t-1} \sum_{j \in S} \sum_{a \in A(j)} \mathbb{P}_\pi^i(X_t = j, A_t = a) \tilde{c}_t(j, a) \end{aligned}$$

et

$$\tilde{J}_\gamma^*(i) := \min_{\pi \in \Pi^M} \tilde{J}_\gamma(i, \pi), \quad i \in S$$

**Théorème 2.3.4.2** *Toute stratégie optimale du PDM transformé est une stratégie optimale pour le problème original et réciproquement. De plus on a :*

$$\tilde{J}_\gamma^*(i) = \frac{\tilde{c}}{1-\gamma} + J_\gamma^*(i), \quad i \in S \quad (2.34)$$

**Preuve 2.3.4.3** *Dès que  $\tilde{c}(i, a) := \tilde{c} + c(i, a)$ ,  $i \in S$ ,  $a \in A(i)$  et de la linéarité de l'espérance mathématique, il vient*

$$J_\gamma(i, \pi) = \frac{\tilde{c}}{1-\gamma} + \mathcal{C}_\gamma(i, \pi), \quad \text{for all } \pi \in \Pi \text{ and } i \in S$$

*En outre, la quantité  $\frac{\tilde{c}}{1-\gamma}$  est indépendante de la stratégie choisie  $\pi$ . Par conséquent, on déduit que :  $\tilde{J}_\gamma^*(i) = \frac{\tilde{c}}{1-\gamma} + J_\gamma^*(i)$ ,  $i \in S$ .  $\square$*



**Exemple 2.3.4.1** *Considérons un PDM défini par les données suivantes :*

$$S = \{1, 2\}; \quad A(1) = A(2) = \{a_1, a_2, a_3\}; \quad \gamma = 0.7.$$

Probabilités de transition :

$$p(1/1, a_1) = 1 ; p(1/1, a_2) = 0.7 ; p(1/1, a_3) = 1 ; p(2/1, a_2) = 0.3$$

$$p(2/2, a_1) = 0.4 ; p(2/2, a_2) = 1 ; p(2/2, a_3) = 1 ; p(1/2, a_1) = 0.6$$

Coûts :

$$c(1, a_1) = -1 ; c(1, a_2) = 2 ; c(1, a_3) = 4 ;$$

$$c(2, a_1) = 7 ; c(2, a_2) = -4 ; c(2, a_3) = 3 ;$$

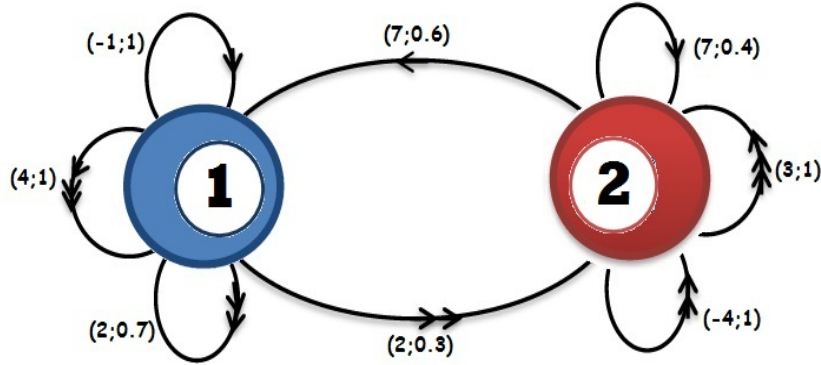


FIGURE 2.5 – Le graphe de l'exemple 2.3.4.1

En appliquant l'algorithme 4 (VIA), on détermine le vecteur fonction de valeur

$$J_\gamma^* = \left( \frac{-10}{3}, \frac{-40}{3} \right)^T$$

et une stratégie optimale  $\pi^*$  définie par

$$\pi^*(1) = a_1 \text{ et } \pi^*(2) = a_2.$$

En appliquant (2.33) et (2.34), le nouveau PDM est comme suit :

$$S = \{1, 2\} ; A(1) = A(2) = \{a_1, a_2, a_3\} ; \gamma = 0.7.$$

Probabilités de transition :

$$p_{1a_1} = 1 ; p_{1a_2} = 0.7 ; p_{1a_3} = 1 ; p_{1a_2} = 0.3$$

$$p_{2a_1} = 0.4 ; p_{2a_2} = 1 ; p_{2a_3} = 1 ; p_{2a_1} = 0.6$$

Coûts :

$$\tilde{c}(1, a_1) = 6 ; \tilde{c}(1, a_2) = 9 ; \tilde{c}(1, a_3) = 11 ;$$

$$\tilde{c}(2, a_1) = 14 ; \tilde{c}(2, a_2) = 3 ; \tilde{c}(2, a_3) = 10 ;$$

En appliquant l'algorithme 6, on obtient les approximations numériques de la fonction test  $B^n(.,.)$  à trois chiffres décimaux :

n	$B^n(1,a_1)$	$B^n(1,a_2)$	$B^n(1,a_3)$	$B^n(2,a_1)$	$B^n(2,a_2)$	$B^n(2,a_3)$
1	-40.667	-37.667	-35.667	-32.667	-43.667	-36.667
2	-28.467	-26.097	-23.467	-18.307	-30.567	-23.567
3	-19.927	-17.998	-14.927	-8.255	-21.397	-14.397
4	-13.949	-12.328	-8.949	-1.218	-14.978	-7.978
5	-9.764	-8.360	-4.764	3.707	-10.484	-3.484
6	-6.835	-5.582	-1.835	7.155	-7.339	-0.339
7	-4.784	-3.637	0.216	9.569	-5.137	1.863
8	-3.349	-2.276	1.651	11.258	-3.596	3.404
9	-2.344	-1.323	2.656	12.441	-2.517	4.483
10	-1.641	-0.656	3.359	13.268	-1.762	5.238
11	-1.149	-0.189	3.851	13.848	-1.233	5.767
12	-0.804	0.137	4.196	14.254	-0.863	6.137
13	-0.563	0.366	4.437	14.537	-0.604	6.396

TABLE 2.3 – Identification des actions non optimales

Ainsi, les nouveaux espaces d'actions valables fournis par le test ci-dessus sont :

$$\tilde{A}(1) = \{a_1\} \text{ après la } 12^{\text{me}} \text{ itération et } \tilde{A}(2) = \{a_2\} \text{ après la } 7^{\text{me}} \text{ itération.}$$

On pose :

$$\underline{n}_\gamma := \min\{n \in \mathbb{N}^* / B^n(i, a) > 0, \text{ pour certain couple } (i, a) \in S \times A(i)\}$$

et

$$\bar{n}_\gamma := \max\{n \in \mathbb{N}^* / B^n(i, a) > 0, \text{ presque pour tout couple } (i, a) \in S \times A(i)\}$$

L'entier naturel  $\underline{n}_\gamma$  donne la première élimination des actions non optimales. Ce qui prouve l'efficacité du test utilisé.

Le tableau suivant donne les variations des paramètres  $\underline{n}_\gamma$  et  $\bar{n}_\gamma$  en variant la valeur du facteur d'actualisation  $\gamma$  dans l'exemple 2.3.4.1.



$\gamma$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
$\underline{n}_\gamma$	2	2	2	3	3	3	5	8	20
$\bar{n}_\gamma$	2	2	3	4	5	7	12	31	45

TABLE 2.4 – Variations des indicateurs  $\underline{n}_\gamma$  et  $\bar{n}_\gamma$ .

En utilisant un des logiciels grapheur, on trace la courbe suivante qui représente les variations des nombres  $\underline{n}_\gamma$  et  $\bar{n}_\gamma$  en fonction du facteur d'actualisation  $\gamma$ .

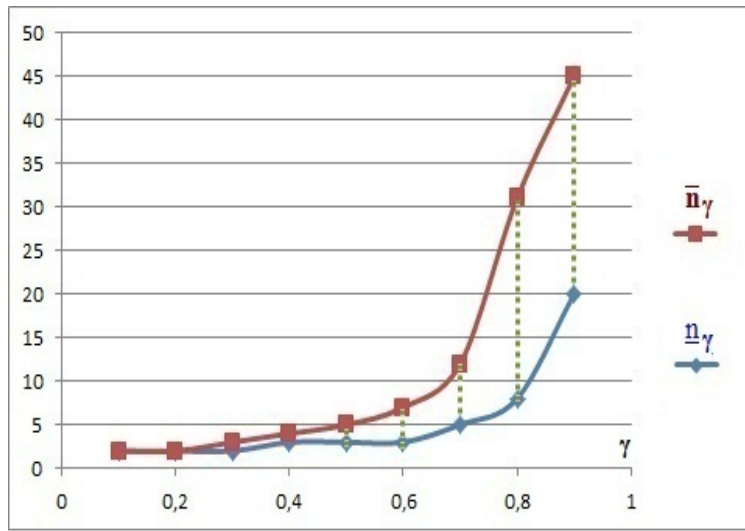


FIGURE 2.6 – Variations graphique des nombres  $\underline{n}_\gamma$  et  $\bar{n}_\gamma$

**Remarque 2.3.4.1** Les valeurs des paramètres  $\underline{n}_\gamma$  et  $\bar{n}_\gamma$  augmentent plus vite dès que le facteur d'actualisation  $\gamma$  s'approche de la valeur 1.

### Le second test d'identification des décisions non optimales :

Le lemme suivant est une clé importante pour construire le test des actions non optimales établi dans le théorème 2.3.4.3.

**Lemme 2.3.4.2** Soit  $(J^n)_{n \geq 0}$  la suite de vecteurs engendré par l'algorithme  $\underline{J}$  (VIA).

Alors :

$$J_i^n - \frac{\gamma^n}{1-\gamma} \tilde{c} \leq J_\gamma^*(i) \leq J_i^n + \frac{\gamma^n}{1-\gamma} \tilde{c}, \quad i \in S, n \geq 0 \quad (2.35)$$

**Preuve 2.3.4.4** *La seule propriété de contraction de l'opérateur de Bellman  $\Phi$  achève la preuve.*  $\square$

Les procédures d'élimination des actions ont attiré l'attention de nombreux chercheurs car elles jouent un rôle très important dans la programmation dynamique. Ils permettent d'accélérer certains algorithmes en réduisant la taille des espaces d'action. Ainsi, nous pouvons réduire la complexité de calcul. De notre côté, nous proposons un nouveau test pour éliminer les actions qui ne font pas partie des décisions optimales dans le théorème suivant.

**Théorème 2.3.4.3 (Le second test de non optimalité)** *Soit  $(J^n)_{n \geq 0}$  la suite de vecteurs donnés par l'algorithme  $\underline{4}$  (VIA). Si*

$$J_i^n + \frac{\gamma^n(1+\gamma)}{1-\gamma} \tilde{c} - Q^n(i, a') < 0 \quad (2.36)$$

*a lieu dans un moment  $n$ , alors toute stratégie qui utilise l'action  $a'$  n'est pas optimale.*

Where

$$Q^n(i, a) = c(i, a) + \gamma \sum_{j \in S} p(j/i, a) J_j^n$$

**Preuve 2.3.4.5** *Supposons que l'actions  $a'$  est optimale dans l'état  $i$  et à l'instant  $n$ . On aura*

$$J_\gamma^* = c(i, a') + \gamma \sum_{j \in S} p(j/i, a') J_j^* \quad (2.37)$$

*De la première inégalité de (2.35) et de (2.36), il découle*

$$J_\gamma^*(i) \geq Q^n(i, a') - \frac{\gamma^{n+1}}{1-\gamma} \tilde{c} \quad (2.38)$$

*En combinant (2.36), (2.38) et la seconde inégalité de (2.35), on obtient  $J_\gamma^* > J_\gamma^*$ . Ce qui finit la preuve.*  $\square$

Tout au long de l'histoire des améliorations subies par l'algorithme d'itération de valeur standard (VIA), un grand défi a été réalisé dans la théorie de la décision de Markov par de nombreuses contributions. En combinant le nouveau test (2.36) et VIA, nous proposons l'algorithme suivant (SJ-VIA) :

---

**Algorithme 7 : Improved VIA (SJ-VIA)**


---

**Input :** MDP =  $(S, A = \bigcup_{i \in S} A(i), Q = \{p(j/i, a)\}, c = \{c(i, a)\}, \gamma)$

**Output :** An optimal policy, An optimal value vector

- 1 **Initialize**  $J_i^0 = 0, \forall i \in S, G = \frac{1+\gamma}{1-\gamma} \tilde{c}$  **and specify**  $\varepsilon > 0$
- 2  $n = 0$
- 3 **repeat**
- 4     **for**  $i \in S$  **do**
- 5         **for**  $a \in A(i)$  **do**
- 6              $Q^n(i, a) = c(i, a) + \gamma \sum_{j \in S} p(j/i, a) J_j^n;$
- 7              $S^n(i, a) = J_i^n - Q^n(i, a) + G\gamma^n;$
- 8             **if**  $S^n(i, a) < 0$  **then**
- 9                  $A(i) = A(i) - \{a\};$
- 10              $J_i^{n+1} = \min_{a \in A(i)} Q^n(i, a);$
- 11      $n = n + 1$
- until**  $\|J^{n+1} - J^n\|_\infty < \frac{1-\gamma}{2\gamma} \varepsilon;$
- 12 **for**  $i \in S$  **do**
- 13      $\pi^*(i) \in \arg \min_{a \in A(i)} Q^n(i, a);$
- 14 **return**  $\pi^*, J^n.$

---

Afin de donner un intérêt scientifique à notre travail, nous devons présenter un exemple numérique. Ensuite, nous montrons le point fort de nos résultats en les comparant avec les précédents.

**Exemple 2.3.4.2** *Considérons un PDM défini par les données suivantes :*

$$S = \{1, 2\}; A(1) = A(2) = \{a_1, a_2, a_3\}; \gamma = 0.7.$$

*Probabilités de transition :*

$$p(1/1, a_1) = 0.5; p(2/1, a_1) = 0.5; p(2/2, a_1) = 1; p(1/1, a_2) = 0.4$$

$$p(2/1, a_2) = 0.6; p(1/2, a_2) = 0.2; p(2/2, a_2) = 0.8;$$

$$p(1/1, a_3) = p(2/2, a_3) = 1$$

Coûts :

$$c(1, a_1) = -7; c(1, a_2) = 6; c(1, a_3) = -10;$$

$$c(2, a_1) = 11; c(2, a_2) = -8; c(2, a_3) = 3.$$

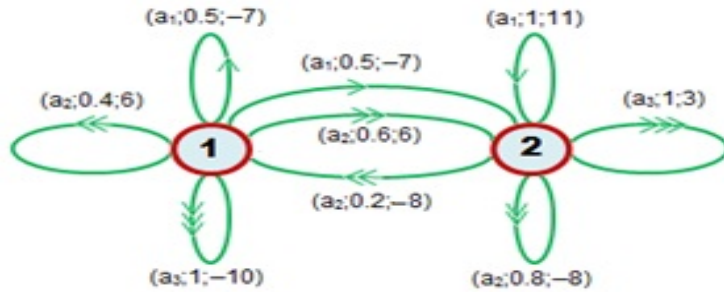


FIGURE 2.7 – Le graphe de l'exemple 2.3.4.2

D'une part, l'algorithme  $\underline{7}$  fournit les résultats suivants :  $J_\gamma^* = (-25, -22.7272)^T$  et une stratégie optimale  $\pi^*$  définie par  $\pi^*(1) = a_3$  et  $\pi^*(2) = a_2$ .

D'autre part, en appliquant l'algorithme  $\underline{7}$  on obtient les approximations numériques de la fonction test  $S^n(.,.)$  à deux chiffres après la virgule.

n	$S^n(1, a_1)$	$S^n(1, a_2)$	$S^n(1, a_3)$	$S^n(2, a_1)$	$S^n(2, a_2)$	$S^n(2, a_3)$
1	28.80	15.68	32.40	12.20	32.16	20.20
2	15.76	2.63	19.44	-0.66	19.41	7.33
3	7.98	-5.15	11.66	-8.42	11.66	-0.42
4	3.31	-9.81	6.99	-13.09	6.99	-5.09
5	0.51	-12.61	4.19	-15.89	4.19	-7.89
6	-1.16	-14.29	2.51	-17.57	2.51	-9.57
7	-2.7	-15.30	1.51	-18.57	1.51	-10.57
8	-2.77	-15.91	0.90	-19.18	0.90	-11.18
9	-3.13	-16.27	0.54	-19.54	0.54	-11.54

TABLE 2.5 – Identification des actions non optimales

Par conséquent, les nouveaux espaces d'actions valables après élimination par le nouveau test deviennent :

$A(1) = \{a_3\}$  après la 6<sup>me</sup> itération.

et

$A(2) = \{a_2\}$  après la 3<sup>me</sup> itération.

On pose :

$$n_\gamma(i) := \min\{n \in \mathbb{N}/S^n(i, a) > 0 \text{ pour certain } a \in A(i)\}, i \in S$$

Le nombre entier naturel  $n_\gamma(i)$  donne le premier temps d'élimination des actions non optimales à l'état  $i$ . Il caractérise l'efficacité du test utilisé. Ensuite, le tableau suivant donne les variations de  $n_\gamma(\cdot)$  en faisant varier le facteur d'actualisation  $\gamma$  inspiré de l'exemple 2.3.4.2.

$\gamma$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
$n_\gamma(1)$	1	1	1	1	2	3	5	9	40
$n_\gamma(2)$	1	1	1	1	2	2	4	8	38

TABLE 2.6 – Variations des indicateurs  $\underline{n}_\gamma$  et  $\bar{n}_\gamma$ .

Le graphe suivant présente les variations du nombre  $n_\gamma$  à l'état  $i$  en fonction du facteur d'actualisation  $\gamma$ .

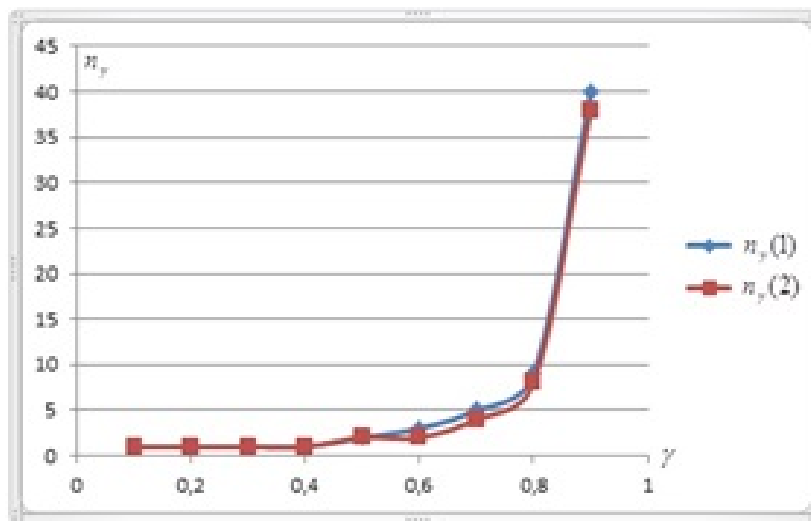


FIGURE 2.8 – Variations graphique des nombres  $n_\gamma(1)$  et  $n_\gamma(2)$

En appliquant l'algorithme de Semmouri et Jourhmane (2019), on obtient la même représentation graphique de  $n_\gamma(\cdot)$  :

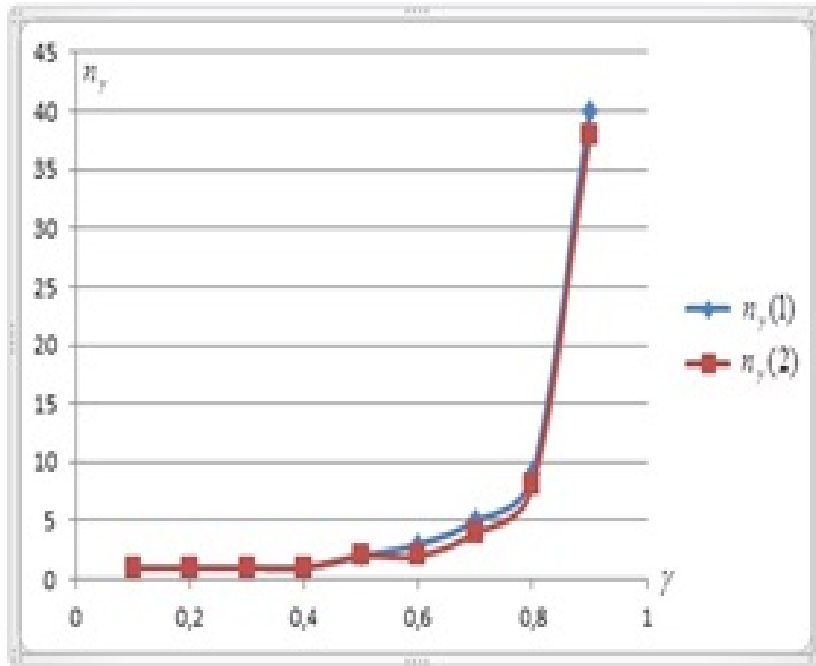


FIGURE 2.9 – Variations graphique des nombres  $n_\gamma(1)$  et  $n_\gamma(2)$

En revanche, notre algorithme n'a pas besoin d'effectuer  $\sum_{i \in S} |A(i)|$  opérations supplémentaires pour transformer les coûts du PDM en question en coûts positifs. La différence entre les deux algorithmes est significative.

### Le troisième test d'identification des décisions non optimales :

Dans un horizon de planification infini, on considère un problème décisionnel de Markov discret décrit formellement par le tuple :

$$(S, A = \bigcup_{i \in S} A(i), Q = \{p(j/i, a)\}_{i,j,a}, r = \{r(i, a)\}_{i,a}, \gamma)$$

où :

- $S$  est un ensemble fini de tous les états possibles du système (espace d'états) ;
- $A$  est un ensemble fini de toutes les actions possibles que l'agent peut choisir d'accomplir en vue de modifier l'état de son environnement (espace d'action). En général, l'espace  $A$  peut dépendre de l'état courant. C'est une réunion de sous-

ensembles  $A(i)$  contenant toutes les actions disponibles à prendre en compte lorsque le système est dans l'état  $i \in S$ .

-  $p(\cdot/\cdot, \cdot) : S \times A \times S \rightarrow [0, 1]$  est une fonction de transition qui donne la probabilité d'atteindre l'état  $j$  en partant de l'état  $i$  quand l'action  $a$  est exécutée.

-  $r : S \times A \rightarrow \mathbb{R}$  sont les fonctions de récompense. Le nombre réel  $r(i, a)$  est la récompense par l'agent lorsqu'il sélectionne l'action  $a$  à partir de l'état  $i$ .

-  $\gamma \in ]0, 1[$  est le facteur d'actualisation.

On transforme le PDM original en un nouveau PDM qui décrit par le tuple :

$$(S, A = \bigcup_{i \in S} A(i), Q = \{p(j/i, a)\}_{i,j,a}, \tilde{c} = \{\tilde{c}(i, a)\}_{i,a}, \gamma)$$

$$\text{où } \tilde{c}(i, a) = R - r(i, a), \forall i \in S, a \in A(i) \text{ et } R := \max_{i \in S, a \in A(i)} |r(i, a)|$$

La fonction objective correspondante est donnée par :

$$\forall \pi \in \Pi, \forall i \in S, \quad J_\gamma(i, \pi) := \sum_{n=1}^{\infty} \gamma^{n-1} \mathbb{E}_\pi^i[\tilde{c}(X_n, A_n)] \quad (2.39)$$

Il est facile de montrer que :

$$\mathcal{V}_\gamma(i, \pi) = \frac{R}{1-\gamma} - J_\gamma(i, \pi), \quad \pi \in \Pi, i \in S$$

Pour  $\pi \in \Pi$ ,  $J_\gamma \in \mathbb{R}^{|S|}$  désigne le vecteur de composantes  $J_\gamma(i) = \min_{\pi \in \Pi} J_\gamma(i, \pi)$ ,  $i \in S$ .

Par des arguments simples, on obtient la relation suivante qui lie entre les fonctions d'utilité :

$$\mathcal{V}_\gamma(i) = \frac{R}{1-\gamma} - J_\gamma(i), \quad i \in S \quad (2.40)$$

Pour  $i \in S$ , on pose :

$$\tilde{c}'(i) := \min_{a \in A(i)} \tilde{c}(i, a), \quad m' := \min_{i \in S} \tilde{c}'(i) \text{ et } M' := \max_{i \in S} \tilde{c}'(i)$$

**Lemme 2.3.4.3** Soit  $(J^n)_{n \geq 0}$  la suite de vecteurs engendré par l'algorithme 4 (VIA).

Alors :

$$W_i^n + \frac{m'\gamma^n}{1-\gamma} \leq J_\gamma(i) \leq W_i^n + \frac{M'\gamma^n}{1-\gamma}, \quad i \in S, a \in A(i) \quad (2.41)$$

**Preuve 2.3.4.6** La démonstration de ce lemme se fait par récurrence sur  $n$ .

Pour  $n = 0$ , le fait que  $W_i^0 = 0, \forall i \in S$  et la section 5 de Sladký [21] garantissent l'assertion suivante :

$$\frac{m'}{1-\gamma} \leq J_\gamma(i) \leq \frac{M'}{1-\gamma}$$

On suppose que la propriété est vraie à l'ordre  $n$  et montrons qu'elle est vraie à l'ordre  $n + 1$ .

Par l'assertion assumption (2.41), on écrit

$$\begin{aligned} \tilde{c}(i, a) + \gamma \sum_{j \in S} p(j/i, a) [W_j^n + \frac{m'\gamma^n}{1-\gamma}] &\leq \tilde{c}(i, a) + \gamma \sum_{j \in S} p(j/i, a) J_\gamma(j) \\ &\leq \tilde{c}(i, a) + \gamma \sum_{j \in S} p(j/i, a) [W_j^n + \frac{M'\gamma^n}{1-\gamma}] \end{aligned}$$

Identiquement equivalent à

$$\tilde{c}(i, a) + \gamma \sum_{j \in S} p(j/i, a) W_j^n + \frac{m'\gamma^{n+1}}{1-\gamma} \leq J_\gamma(i) \leq \tilde{c}(i, a) + \gamma \sum_{j \in S} p(j/i, a) W_j^n + \frac{M'\gamma^{n+1}}{1-\gamma}$$

Dès que

$$\tilde{c}(i, a) + \gamma \sum_{j \in S} p(j/i, a) W_j^n \leq J_\gamma(i) - \frac{m'\gamma^{n+1}}{1-\gamma}$$

et

$$J_\gamma(i) - \frac{M'\gamma^{n+1}}{1-\gamma} \leq \tilde{c}(i, a) + \gamma \sum_{j \in S} p(j/i, a) W_j^n$$

Donc

$$\min_{a \in A(i)} \{ \tilde{c}(i, a) + \gamma \sum_{j \in S} p(j/i, a) W_j^n \} \leq J_\gamma(i) - \frac{m'\gamma^{n+1}}{1-\gamma} \quad (2.42)$$

et

$$J_\gamma(i) - \frac{M'\gamma^{n+1}}{1-\gamma} \leq \min_{a \in A(i)} \{ \tilde{c}(i, a) + \gamma \sum_{j \in S} p(j/i, a) W_j^n \} \quad (2.43)$$

En combinant ( 2.42) and (2.43), On obtient :

$$W_i^{n+1} + \frac{m'\gamma^{n+1}}{1-\gamma} \leq J_\gamma(i) \leq W_i^{n+1} + \frac{M'\gamma^{n+1}}{1-\gamma}, \quad i \in S, a \in A(i)$$

Ce qui achève la prouve. □

**Théorème 2.3.4.4 (Le troisième test de non optimalité)** *Soit  $(W^n)_{n \geq 0}$  la suite de vecteurs donnés par l'algorithme 4 (VIA). Si*

$$Q^n(i, a') + \frac{m'\gamma^{n+1}}{1-\gamma} > W_i^n + \frac{M'\gamma^n}{1-\gamma} \quad (2.44)$$



a lieu dans un moment  $n$ , alors toute stratégie qui utilise l'action  $a'$  n'est pas optimale.

où :

$$Q^n(i, a) = c(i, a) + \gamma \sum_{j \in S} p(j/i, a) W_j^n$$

**Preuve 2.3.4.7** Supposons que l'action  $a'$  est optimale dans l'état  $i$  et à l'instant  $n$ . On obtient

$$J_\gamma(i) = \tilde{c}(i, a') + \gamma \sum_{j \in S} p(j/i, a') J_\gamma(i) \quad (2.45)$$

De (2.43), (2.44) et (2.45) On conclut que

$$\begin{aligned} J_\gamma(i) &\geq \tilde{c}(i, a') + \gamma \sum_{j \in S} p(j/i, a') \left\{ W_i^n + \frac{m' \gamma^n}{1 - \gamma} \right\} \\ &= \tilde{c}(i, a') + \gamma \sum_{j \in S} p(j/i, a') W_i^n + \frac{m' \gamma^{n+1}}{1 - \gamma} \\ &> W_i^n + \frac{M' \gamma^n}{1 - \gamma} \geq J_\gamma(i) \end{aligned}$$

Par conséquent,  $J_\gamma(i) > J_\gamma(i)$ . Ce qui est absurde. Ce qui termine la preuve.  $\square$

L'algorithme d'itération de valeur (VIA) subit plusieurs améliorations dans la littérature pour résoudre les problèmes décisionnels de Markov. Dans ce contexte, nous allons améliorer cet algorithme par l'ajout du test de non-optimalité (2.44) afin d'établir un nouvel algorithme très utile pour la programmation dynamique.

On pose :

$$T^n(i, a) := W_i^n - Q^n(i, a) - \frac{m' \gamma - M'}{1 - \gamma} \gamma^n, \quad i \in S, a \in A(i)$$

Si  $T^n(i, a) > 0$ , alors on doit éliminer l'action  $a \in A(i)$  dans l'état  $i$  à l'instant  $n + 1$  de la machinerie du calcul.

Ainsi, on propose l'algorithme suivant :

---

**Algorithme 8 : Improved Value Iteration Algorithm (IVIA)**


---

**Input :** MDP =  $(S, A = \bigcup_{i \in S} A(i), Q = \{p(j/i, a)\}_{i,j,a}, \tilde{c} = \{\tilde{c}(i, a)\}_{i,a}, \gamma)$

**Output :** The optimal value vector & an optimal policy

1 **Initialize**  $W_i^0 \leftarrow 0, i \in S$ , **specify**  $\varepsilon > 0$

2 **n**  $\leftarrow 0$

**for**  $i \in S$  **do**

3  $\left[ \pi^*(i) \in \arg \min_{a \in A(i)} Q^n(i, a);$

4 **repeat**

**for**  $i \in S$  **do**

**for**  $a \in A(i)$  **do**

5  $Q^n(i, a) \leftarrow \tilde{c}(i, a) + \gamma \sum_{j \in S} p(j/i, a) W_j^n;$

6  $T^n(i, a) \leftarrow W_i^n - Q^n(i, a) - \frac{m' \gamma - M'}{1 - \gamma} \gamma^n;$

7 **if**  $T^n(i, a) < 0$  **then**

8  $\left[ \left[ A(i) \leftarrow A(i) - \{a\}$

$W_i^{n+1} \leftarrow \min_{a \in A(i)} Q^n(i, a);$

**n**  $\leftarrow$  **n** + 1

**until**  $\|W^{n+1} - W^n\|_\infty < \frac{1-\gamma}{2\gamma} \varepsilon;$

**for**  $i \in S$  **do**

$\left[ \pi^*(i) \in \arg \min_{a \in A(i)} Q^n(i, a);$

9 **return**  $W^n, \pi^*.$

---

Pour apprécier et illustrer notre contribution, nous présentons l'exemple suivant.

**Exemple 2.3.4.3** *Considérons un PDM défini par les données suivantes :*

$$S = \{1, 2\}; A(1) = A(2) = \{a_1, a_2, a_3\}; \gamma = 0.6.$$

*Probabilités de transition :*

$$p(1/1, a_1) = 1; p(1/1, a_2) = 0.2; p(1/1, a_3) = 1; p(2/1, a_2) = 0.8;$$

$$p(2/2, a_1) = 0.3; p(2/2, a_2) = 1; p(2/2, a_3) = 1; p(1/2, a_1) = 0.7.$$

*Récompences :*

$$r(1, a_1) = -7; r(1, a_2) = 9; r(1, a_3) = -3;$$

$$r(2, a_1) = 12; r(2, a_2) = -4; r(2, a_3) = 5.$$

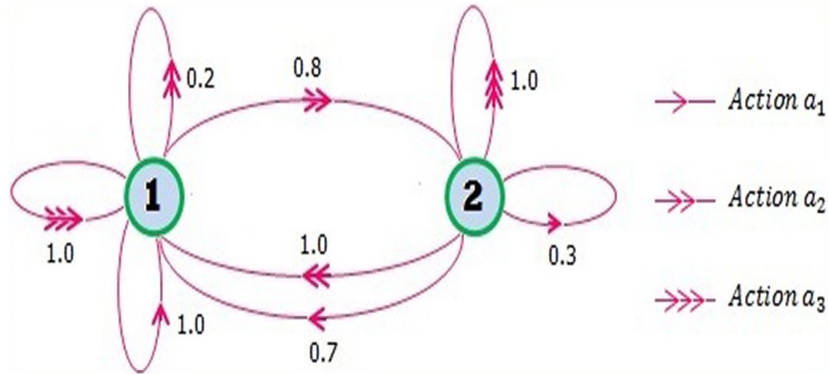


FIGURE 2.10 – Graphe de l'exemple 2.3.4.3

En exploitant l'algorithme 8 (IVIA), on obtient les résultats numériques suivants :

It n	$T^n(1, a_1)$	$T^n(1, a_2)$	$T^n(1, a_3)$	$T^n(2, a_1)$	$T^n(2, a_2)$	$T^n(2, a_3)$
<b>1</b>	-1.5	14.5	2.5	17.5	1.5	10.5
<b>2</b>	-7.3	10.14	-3.3	9.24	-7.3	3.5
<b>3</b>	-11.356	5.652	-7.356	5.922	-10.456	-0.196
<b>4</b>	-13.617	3.521	-9.617	3.440	-12.987	-2.565
...	...	...	...	...	...	...
<b>30</b>	-17.108	5.93E-06	-13.108	5.93E-06	-16.415	-6.031
<b>31</b>	-17.108	3.56E-06	-13.108	3.56E-06	-16.415	-6.031
<b>32</b>	-17.108	2.14E-06	-13.108	2.14E-06	-16.415	-6.031

TABLE 2.7 – Values of  $T^n(.,.)$  by IVIA

Toute valeur de  $T^n(i, a)$  colorée en rouge indique que l'action  $a$  sélectionnée à l'état  $i$  est identifiée en tant que action non optimale qui doit être éliminée au cours des calculs. D'abord, les actions  $a_2$  et  $a_3$  doit être éliminées dans l'état 2 respectivement à la 1<sup>re</sup> itération et à la 2<sup>me</sup> itération. En suite, les actions  $a_1$  et  $a_3$  doit être éliminées dans l'état 1 respectivement à la 2<sup>me</sup> itération et à la 3<sup>me</sup> itération.

Le nouveau problème transformé, le vecteur valeur optimal  $J_\gamma$  est atteint à l'itération  $n = 32$  pour  $\gamma = 0.6$  et est donné par

$$J_{0.6} = (4.730768952, 2.423076645)$$

En outre, les nouveaux espaces d'actions disponibles fournis par cet algorithme sont :  $A(1) = \{a_3\}$  après  $2^{me}$  itération et  $A(2) = \{a_1\}$  après  $3^{me}$  itération.

En utilisant (2.40), on déduit que le vecteur valeur optimal  $\mathcal{V}_\gamma$  de l'original PDM est :

$$\mathcal{V}_{0.6} = (25.26923105, 27.57692336)$$

La comparaison de notre test à ceux de MacQueen et Porteus, on obtient le graphique suivant qui donne le nombre d'itérations nécessaires pour éliminer les décisions non optimales :

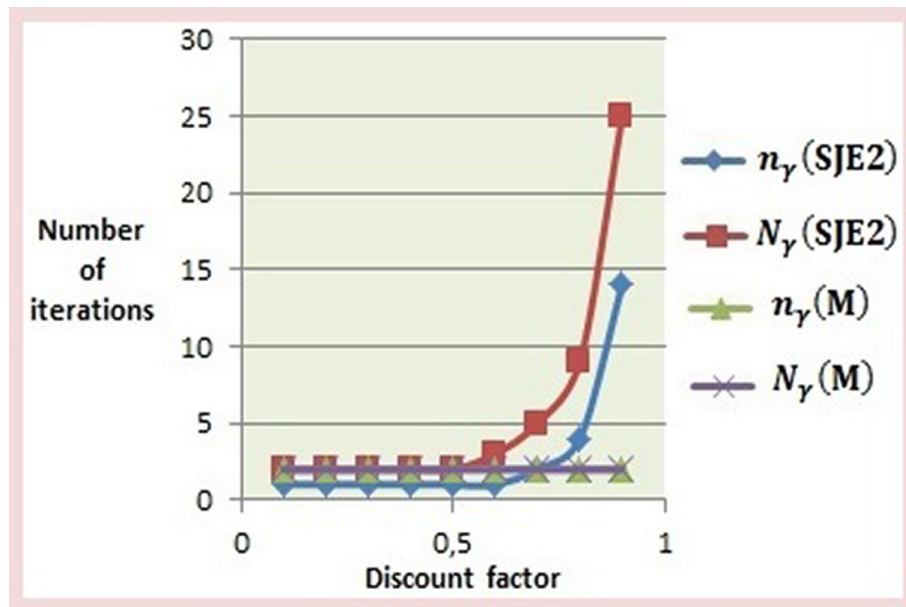


FIGURE 2.11 – Comparaison du test 2.44 avec celui de MacQueen

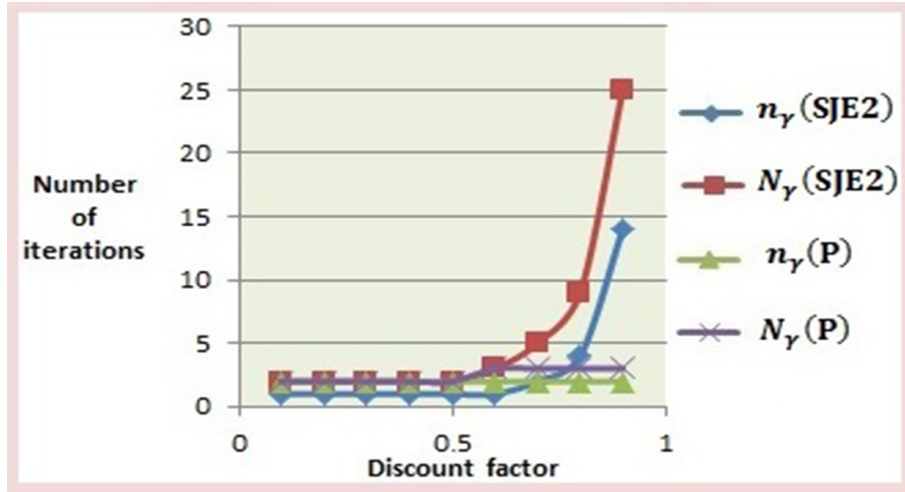


FIGURE 2.12 – Comparaison du test 2.44 avec celui Porteus

où

$$n_\gamma(\text{SJE2}) = \inf\{n \in \mathbb{N} / \exists i \in S, \exists a \in A(i), T^n(i, a) < 0\}$$

et

$$N_\gamma(\text{SJE2}) = \sup\{n \in \mathbb{N} / \exists i \in S, \exists a \in A(i), T^n(i, a) < 0\}$$

Dans les figures 2.11 et 2.12 pour  $\gamma \in ]0, 0.7]$ , notre nouveau test est approximativement similaire aux tests de MacQueen et Porteus. Ce comportement est bien convaincant dans la vie réelle et les applications pratiques.

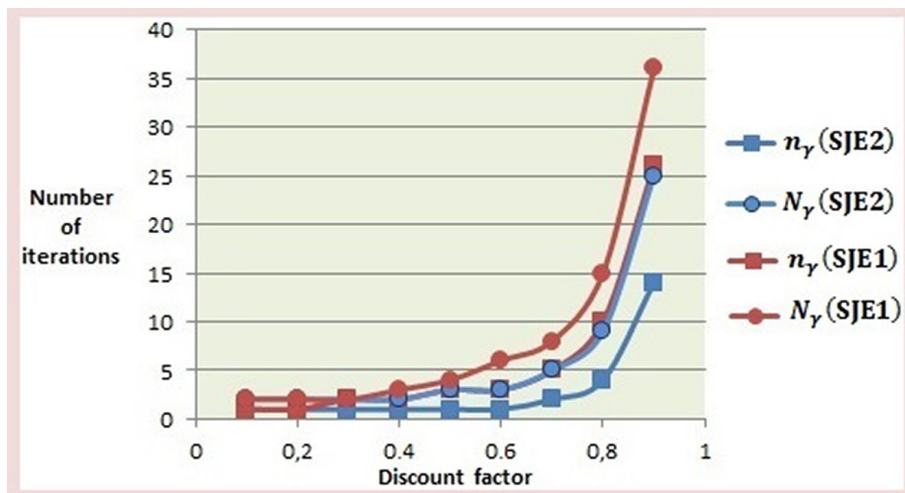


FIGURE 2.13 – Comparaison du test 2.44 avec le test De SJE1

De la figure [2.13](#) ci-dessus, on remarque que les nombres  $n_\gamma(SJE2)$  et  $N_\gamma(SJE2)$  sont respectivement assez petits par rapport aux nombres  $n_\gamma(SJE1)$  and  $N_\gamma(SJE1)$ . Ceci explique pourquoi le nouveau test établi dans cette partie est très efficace par rapport à celui donné dans [24] (SJE1).

### Le quatrième test d'identification des décisions non optimales :

Nous concentrons notre étude sur la complexité d'un problème classique de détermination de stratégie optimale dans les problèmes décisionnels de Markov avec sous le critère gain actualisé. Il est bien connu que ces problèmes sont résolubles en temps polynomial via l'outil de la programmation dynamique. Dans ce contexte, nous améliorons l'algorithme de sur-relaxation successive (SOR) en établissant un nouveau test de décision non optimale. Enfin, nous donnons un exemple pour clarifier et démontrer notre proposition.

Considérons un modèle de contrôle de problème décisionnel de Markov à temps discret représenté par le tuple :

$$\left( S; A = \bigcup_{i \in S} A(i); Q = \{p_n(j/i, a)\}; r = \{r_n(i, a)\}; \gamma \right)$$

où :

- $S$  est un espace d'états fini dans lequel se déroule l'évolution du processus ;
- $A$  l'union de tous les ensembles d'actions finis disponibles qui contrôlent la dynamique de l'état ;
- $p(\cdot/\cdot, \cdot)$  désigne les fonctions de probabilité de transition d'état ;
- $r(\cdot, \cdot)$  donne la fonction de récompense définie sur les transitions d'état ;
- $\gamma \in (0, 1)$  est le facteur d'actualisation.

L'objectif est de déterminer une stratégie qui, à chaque instant de décision, maximise les performances du système sur l'horizon infini. Cette performance est exprimée sous la forme de la fonction d'utilité du gain actualisé sur l'horizon de planification.

**Définition 2.3.4.1** *Si la stratégie  $\pi \in \Pi$  est utilisée et que l'état  $i \in S$  est l'état de départ, le gain actualisé noté  $J_\gamma(i, \pi)$  est défini par :*

$$J_\gamma(i, \pi) := \mathbb{E}_\pi^i \left[ \sum_{n=1}^{\infty} \gamma^{n-1} r(X_n, A_n) \right] = \sum_{n=1}^{\infty} \gamma^{n-1} \mathbb{E}_\pi^i [r(X_n, A_n)] \quad (2.46)$$

$$= \sum_{n=1}^{\infty} \gamma^{n-1} \sum_{j \in S} \sum_{a \in A(j)} \mathbb{P}_{\pi}^i(X_n = j, A_n = a) r(j, a)$$

**Définition 2.3.4.2** La fonction valeur optimale est définie par :

$$J_{\gamma}^*(i) := \sup_{\pi \in \Pi} J_{\gamma}(i, \pi), \quad i \in S \quad (2.47)$$

La stratégie  $\pi^* \in \Pi$  est dite optimale si :  $J_{\gamma}^*(i) = J_{\gamma}(i, \pi^*), \forall i \in S$ .

Puisque l'espace d'états et les espaces d'action sont finis, l'existence et la détermination de stratégies optimales sont étudiés dans la théorie de la décision de Markov. Dans le cadre de l'analyse numérique, la méthode de sur-relaxation successive (en abrégé SOR) est une variante de la méthode de Gauss-Seidel pour résoudre un système d'équations linéaires. La convergence de cet algorithme est généralement plus rapide. Cette méthode a été découverte simultanément par David M. Young, Jr. et Stan Frankel en 1950 dans le but de résoudre automatiquement des systèmes linéaires avec des ordinateurs. Ensuite, les méthodes de sur-relaxation ont été utilisées pour résoudre des problèmes décisionnels de Markov en tant qu'une variante de l'algorithme d'itération de la valeur standard (VIA).

On pose :

$$\omega^* = \min_{i,a} \frac{1}{1 - \gamma p(i/i, a)} \quad \text{et} \quad R = \max_{i,a} r(i, a)$$

Pour  $\omega \in ]0, \omega^*]$ , On définit les quantités  $\gamma_i(\omega), i \in S$  par la formule récursive :

$$\gamma_i(\omega) = \omega \left[ \max_{a \in A(i)} \left\{ r(i, a) + \gamma \sum_{j=1}^{i-1} p(j/i, a) \gamma_j(\omega) + \gamma \sum_{j=i}^{|S|} p(j/i, a) \right\} \right] + 1 - \omega$$

et

$$\rho(\omega) = \max_{i \in S} \gamma_i(\omega)$$

On considère l'opérateur de Bellman  $\Phi : \mathbb{R}^{|S|} \rightarrow \mathbb{R}^{|S|}$  donné pour tout vecteur  $V = (V_i)_{i \in S} \in \mathbb{R}^{|S|}$  par

$$\begin{aligned} (\Phi_{\omega} V)_i = \omega \left[ \max_{a \in A(i)} \left\{ r(i, a) + \gamma \sum_{j=1}^{i-1} p(j/i, a) (\Phi_{\omega} V)_j + \gamma \sum_{j=i}^{|S|} p(j/i, a) V_j \right\} \right] \\ + (1 - \omega) V_i \end{aligned} \quad (2.48)$$

Il est établi par Denardo [72] et Reetz [73] que  $\Phi_\omega$  est une application contractante de paramètre de contraction  $\mu := \rho(\omega^*) = 1 + (\gamma - 1)\omega^*$  et  $\rho(\omega^*) \leq \gamma < 1$ .

Puisque le vecteur fonction valeur  $J_\gamma^*$  est l'unique point fixe de  $\Phi_\omega$ , il s'en suit l'algorithme SOR :

---

**Algorithme 9 : Successive Over-Relaxation (SOR)**

---

**Input :** MDP =  $(S; A = \bigcup_{i \in S} A(i); Q = \{p(j/i, a)\}; r = \{r(i, a)\}; \gamma)$

**Output :** Le vecteur valeur optimal ; Une stratégie optimale

1 **Initialize**  $V^0 = 0 \in \mathbb{R}^{|S|}$ , **specify**  $\varepsilon > 0$

2  $n \leftarrow 0$

3 **repeat**

**for**  $i = 1$  *to*  $|S|$  **do**

**for**  $a \in A(i)$  **do**

$$Q^n(i, a) = r(i, a) + \gamma \sum_{j \prec i} p(j/i, a)V_j^{n+1} + \gamma \sum_{j \geq i} p(j/i, a)V_j^n$$

$$V_i^{n+1} = \omega^* \max_{a \in A(i)} Q^n(i, a) + (1 - \omega^*)V_i;$$

$n \leftarrow n + 1$

**until**  $\|V^{n+1} - V^n\|_\infty < \varepsilon$ ;

**for**  $i \in S$  **do**

$$\pi^*(i) \in \arg \max_{a \in A(i)} Q^n(i, a);$$

4 **return**  $V^n, \pi^*$ .

---

**Lemme 2.3.4.4** *Let  $(V^n)_{n \geq 0}$  la suite de vecteurs fournie par la méthode SOR. Alors on aura :*

$$V_i^n - \frac{\mu^n R}{1 - \mu} \leq J_\gamma(i) \leq V_i^n + \frac{\mu^n R}{1 - \mu}, \quad \text{for all } i \in S, n \geq 0. \quad (2.49)$$

**Preuve 2.3.4.8** *La preuve de ce lemme est assurée par la théorie des applications contractantes.* □

Les procédures d'élimination d'action sont un outil puissant qui est largement utilisé pour résoudre les problèmes décisionnels de Markov pour réduire la complexité des calculs. Nous proposons donc un test efficace dans le théorème suivant :



**Théorème 2.3.4.5 (Nouveaux Test de non optimalité)** Soit  $(V^n)_{n \geq 0}$  la suite de vecteurs générée par l'algorithme SOR. Si

$$r(i, b) + \gamma \sum_{j \in S} p(j/i, b) V_j^n + \frac{\mu^n (1 + \gamma) R}{1 - \mu} < V_i^n \quad (2.50)$$

à un instant  $n$ , alors toute stratégie markovienne qui sélectionne l'action  $b$  dans l'état  $i$  est non optimale.

**Preuve 2.3.4.9** Supposons que l'action  $b$  est optimale dans l'état  $i$  à l'instant  $n$ . Il est clair que le maximum dans l'équation d'optimalité de Bellman est atteint par l'action  $b$  dans l'état  $i$ . Ainsi, on obtient

$$J_\gamma^*(i) = r(i, b) + \gamma \sum_{j \in S} p(j/i, b) J_\gamma^*(j) \quad (2.51)$$

De (2.49) et (2.51), découle

$$J_\gamma^*(i) \leq r(i, b) + \gamma \sum_{j \in S} p(j/i, b) V_j^n + \frac{\mu^n R \gamma}{1 - \mu} \quad (2.52)$$

En combinant (2.49), (2.50) et (2.52), on obtient

$$J_\gamma^*(i) < V_i^n - \frac{\mu^n R}{1 - \mu} \leq J_\gamma^*(i)$$

Ce qui achève la preuve. □

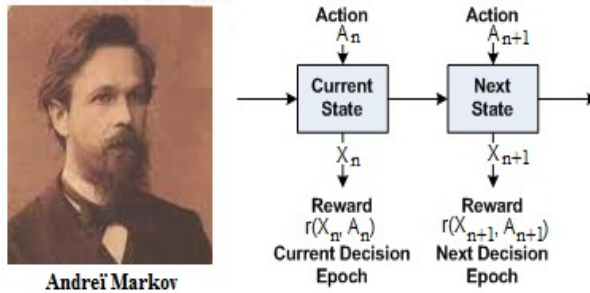


FIGURE 2.14 – Modèle du processus décisionnel de Markov

L'application pratique de notre contribution nécessite l'amélioration de l'algorithme SOR. Par conséquent, une combinaison du test (2.50) et de l'algorithme 9 (SOR) conduit à l'algorithme suivant :

---

**Algorithme 10 : Improved Successive Over-Relaxation (ISOR)**

---

**Input :** MDP =  $(S; A = \bigcup_{i \in S} A(i); Q = \{p(j/i, a)\}; r = \{r(i, a)\}; \gamma)$

**Output :** Le vecteur valeur optimal, Une stratégie optimale

```
1 Initialize  $V^0 = 0 \in \mathbb{R}^{|S|}$ , specify  $\varepsilon > 0$ 
2  $n \leftarrow 0$ 
3 repeat
4   for  $i = 1$  to  $|S|$  do
5     for  $a \in A(i)$  do
6        $Q^n(i, a) = r(i, a) + \gamma \sum_{j < i} p(j/i, a)V_j^{n+1} + \gamma \sum_{j \geq i} p(j/i, a)V_j^n$ ;
7        $S^n(i, a) = r(i, a) + \gamma \sum_{j < i} p(j/i, a)V_j^n + \frac{\mu^n(1 + \gamma)R}{1 - \mu} - V_i^n$ ;
8       if  $S^n(i, a) < 0$  then
9          $A(i) \leftarrow A(i) - \{a\}$ 
10       $V_i^{n+1} = \omega^* \max_{a \in A(i)} Q^n(i, a) + (1 - \omega^*)V_i$ ;
11     $n \leftarrow n + 1$ 
  until  $\|V^{n+1} - V^n\|_\infty < \varepsilon$ ;
12 for  $i \in S$  do
   $\pi^*(i) \in \arg \max_{a \in A(i)} Q^n(i, a)$ ;
13 return  $V^n, \pi^*$ .
```

---

**Exemple 2.3.4.4** On considère un PDM avec les données suivantes :

$$S = \{1, 2\}; A(1) = A(2) = \{a_1, a_2, a_3\}; \gamma = 0.6.$$

Les probabilités de transition :

$$p(1/1, a_1) = 0.5; p(2/1, a_1) = 0.5; p(2/2, a_1) = 0.1; p(1/2, a_1) = 0.9;$$

$$p(1/1, a_2) = 1; p(2/2, a_2) = 1; p(1/1, a_3) = 1; p(2/2, a_3) = 1.$$

Les récompenses :

$$r(1, a_1) = 7; r(1, a_2) = 5; r(1, a_3) = 8;$$

$$r(2, a_1) = 4; r(2, a_2) = 6; r(2, a_3) = 3.$$

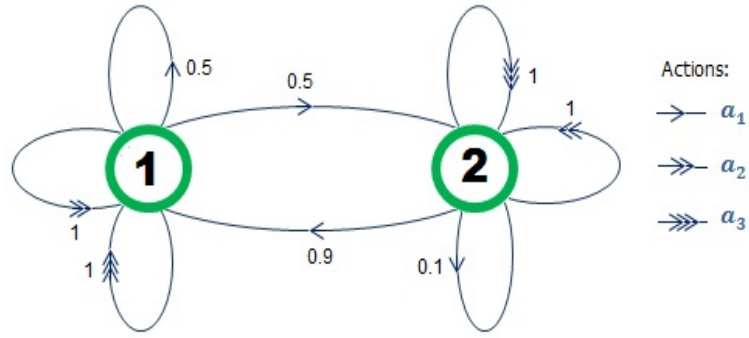


FIGURE 2.15 – Graphe probabiliste de l'exemple 2.3.4.4

It	$S^n(1, a_1)$	$S^n(1, a_2)$	$S^n(1, a_3)$	$S^n(2, a_1)$	$S^n(2, a_2)$	$S^n(2, a_3)$
1	37.526808	35.399149	38.590638	39.224409	36.462978	33.271489
5	2,076770	1,011377	4,202866	3,808447	3,491562	0,300072
6	0,162438	-0,777077	2,414413	2,187831	1,870946	-1,320543
7	-0,937285	-1,804486	1,387003	1,256839	0,939954	-2,251535
8	-1,569040	-2,394700	0,796789	0,722014	0,405129	-2,786361
9	-1,931964	-2,733759	0,457730	0,414774	0,097888	-3,093601
10	-2,140452	-2,928538	0,262951	0,238274	-0,078611	-3,270100
11	-2,260221	-3,040432	0,151057	0,136881	-0,180004	-3,371494
12	-2,329025	-3,104712	0,086777	0,078634	-0,238252	-3,429741

TABLE 2.8 – Actions éliminées par la fonction de test  $S^n$

Les résultats numériques ci-dessus sont données en appliquant l'Algorithme itérative 10 (ISOR).

Chaque valeur strictement négative de  $S^n(.,.)$  (écrite en rouge) indique que l'action  $a$  sélectionnée dans l'état  $i$  est identifiée comme une action non optimale à l'instant  $n$ . On doit l'éliminer après cette itération. Dans un premier temps, les actions  $a_2$  et  $a_1$  doivent être éliminées respectivement après la 6<sup>ème</sup> itération et à la 7<sup>ème</sup> itération dans l'état 1. Ainsi, On élimine les actions  $a_3$  et  $a_2$  respectivement après la 6<sup>ème</sup> itération et à la 10<sup>ème</sup> itération dans l'état 2.

Pour  $\gamma = 0.6$ , le vecteur valeur optimal est :

$$J_{0.6}^* = (20, 15.74468085)$$

De plus, les nouveaux espaces d'actions sont donnés par l'algorithme 10 comme suit :

$$A(1) = \{a_3\} \text{ après la } 7^{\text{ème}} \text{ itération et } A(2) = \{a_1\} \text{ après la } 10^{\text{ème}} \text{ itération.}$$

Soit  $n_1$  le plus petit nombre d'itérations nécessaires pour éliminer la première action et  $n_2$  le plus grand nombre d'itérations nécessaires pour éliminer toutes les actions non optimales. En faisant varier le facteur d'actualisation  $\gamma$ , le comportement des nombres  $n_1(\gamma)$  et  $n_2(\gamma)$  est donné par la Figure 2.16 ci-dessous :

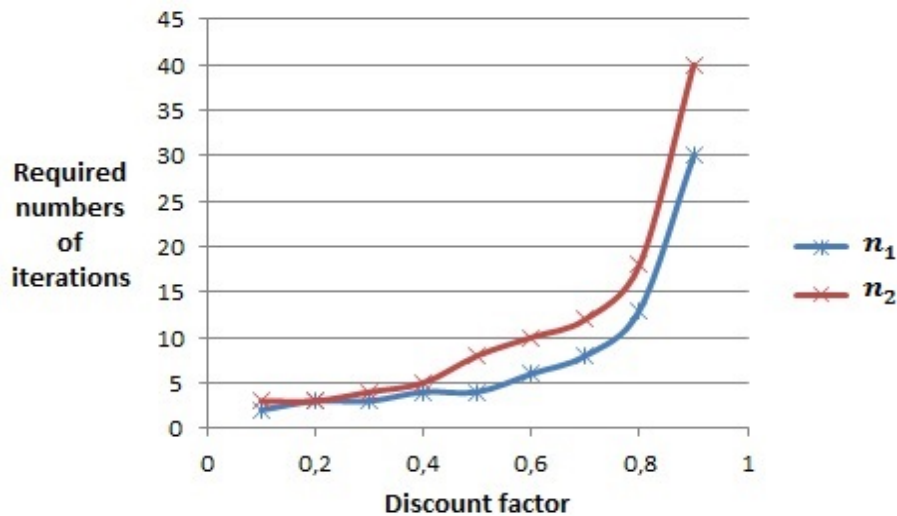


FIGURE 2.16 – Variation de  $n_1(\gamma)$  et  $n_2(\gamma)$

Pour  $0.1 \leq \gamma \leq 0.8$ , la figure 2.16 montre clairement l'efficacité de notre contribution. En fait, le nombre maximum d'itérations pour éliminer toutes les actions non optimales est faible. De plus, les valeurs des itérations  $n_1(\gamma)$  et  $n_2(\gamma)$  augmentent dès que  $\gamma$  est proche de 1.

## 2.4 Conclusion du chapitre

La programmation dynamique reste une approche efficace pour résoudre une multitude de problèmes stochastiques qui sont modélisés par des processus décisionnels de Markov. À cet égard, l'algorithme de l'itération de la valeur (VIA) est un point de concours pour de nombreux auteurs chercheurs. Chaque amélioration vise à accélérer VIA ou à réduire la complexité des calculs en espace et en temps dans un

certain sens. Ainsi, l'amélioration de VIA ne cesse d'augmenter pour établir d'importantes contributions via l'outil de programmation dynamique.

Néanmoins, ce chapitre renferme des nouveaux tests pour l'identification des actions qui ne font pas partie des décisions optimales. La combinaison de ces résultats et de l'approche itérative permet d'établir des algorithmes efficaces pour résoudre les problèmes décisionnels de Markov avec un minimum d'efforts.

Malgré ce défi, la plupart des tests de non-optimalité proposés jusqu'à présent s'inspirent des tests de MacQueen et de Porteus comme références incontournables.

## CHAPITRE 3

# LE CRITÈRE DU COÛT FLOUE $\gamma$ -ACTUALISÉ

### 3.1 Introduction

Parmi les divers changements paradigmatiques de la science et des mathématiques au cours de ce siècle, un de ces changements concerne le concept d'incertitude. En science, ce changement s'est manifesté par une transition graduelle de la vision traditionnelle, qui insiste sur le fait que l'incertitude est indésirable en science et devrait être évitée par tous les moyens possibles, à une vision alternative, qui tolère l'incertitude et insiste sur le fait que la science ne peut pas l'éviter. Selon la vision traditionnelle, la science devrait viser la certitude dans toutes ses manifestations (précision, spécificité, acuité, cohérence, etc.) ; par conséquent, l'incertitude est considérée comme non scientifique. Selon la vision alternative (ou moderne), l'incertitude est considérée comme essentielle à la science ; ce n'est pas seulement un fléau inévitable, mais il a, en fait, une grande utilité.

La première étape de la transition de la vision traditionnelle à la vision moderne de l'incertitude a commencé à la fin du *XIX*<sup>e</sup> siècle, lorsque la physique s'est intéressée aux processus au niveau moléculaire. Bien que des lois précises de la mécanique Newtonienne aient été pertinentes pour l'étude de ces processus, leur application réelle au nombre énorme d'entités impliquées aurait entraîné des demandes de calcul qui dépassaient de loin les capacités de calcul existantes et, comme nous

le réalisons maintenant, dépassent même les limites de calcul fondamentales. . Autrement dit, ces lois précises se voient refuser l'applicabilité dans ce domaine non seulement en pratique (sur la base de la technologie informatique existante) mais en principe.

La nécessité d'une approche fondamentalement différente de l'étude des processus physiques au niveau moléculaire a motivé le développement de méthodes statistiques pertinentes, qui se sont avérées applicables non seulement à l'étude des processus moléculaires (mécanique statistique), mais aussi à une foule d'autres domaines tels que la profession actuarielle, la conception de grands centraux téléphoniques, etc. Dans les méthodes statistiques, les manifestations spécifiques d'entités microscopiques (molécules, sites téléphoniques individuels, etc.) sont remplacées par leurs moyennes statistiques, qui sont liées à des variables macroscopiques appropriées. Le rôle joué en mécanique newtonienne par le calcul, qui n'implique aucune incertitude, est remplacé en mécanique statistique par la théorie des probabilités, théorie dont le but même est de saisir l'incertitude d'un certain type.

L'incertitude est donc un produit important dans le secteur de la modélisation, qui peut être échangé contre des gains dans les autres caractéristiques essentielles des modèles. Ce compromis peut ensuite être utilisé pour construire des modèles qui sont au maximum utiles par rapport à l'objectif pour lequel ils sont construits. La reconnaissance de ce rôle important de l'incertitude par certains chercheurs, qui est devenu assez explicite dans la littérature des années 1960, a commencé le deuxième âge de la transition de la vision traditionnelle à la vision moderne de l'incertitude. Cette étape est caractérisée par l'émergence de plusieurs nouvelles théories de l'incertitude, distinctes de la théorie des probabilités. Ces théories remettent en question le lien apparemment unique entre l'incertitude et la théorie des probabilités, qui était auparavant considérée comme acquise. Ils montrent que la théorie des probabilités n'est capable de représenter qu'un seul des nombreux types d'incertitude.

Il est généralement admis qu'un point important de l'évolution du concept moderne d'incertitude a été la publication d'un article fondateur de Lotfi A. Zadeh [58], même si certaines idées présentées dans l'article ont été envisagées quelque 30 ans plus tôt par le philosophe américain Max Black en 1937. Dans son article, Zadeh a

introduit une théorie dont les objets (ensembles pétillants) sont des ensembles avec des limites qui ne sont pas précises. L'appartenance à un ensemble flou n'est pas une question d'affirmation ou de déni, mais plutôt une question de degré.

La signification de l'article de Zadeh était qu'il contestait non seulement la théorie des probabilités en tant que seul agent de l'incertitude, mais les fondements mêmes sur lesquels la théorie des probabilités est basée sur la logique aristotélicienne à deux valeurs. Lorsque  $A$  est un ensemble flou et  $x$  est un objet pertinent, la proposition " $x$  est un membre de  $A$ " n'est pas nécessairement vraie ou fausse, comme l'exige la logique à deux valeurs, mais elle peut être vraie seulement dans une certaine mesure, le degré auquel  $x$  est réellement membre de  $A$ . Il est le plus courant, mais non obligatoire, d'exprimer les degrés d'appartenance à des ensembles flous ainsi que les degrés de vérité des propositions associées par des nombres dans l'intervalle fermé borné  $[0, 1]$ . Les valeurs extrêmes de cet intervalle, 0 et 1, représentent alors, respectivement, le déni total et l'affirmation de l'appartenance à un ensemble flou donné ainsi que la fausseté et la vérité de la proposition associée.

La recherche sur la théorie des ensembles flous n'a cessé de croître depuis le début de la théorie au milieu des années 1960. L'ensemble des concepts et des résultats relatifs à la théorie est désormais assez impressionnant. La recherche sur une grande variété d'applications a également été très active et a produit des résultats qui sont peut-être encore plus impressionnants.

## 3.2 Généralités sur les ensembles flous

Dans cette section, nous présentons quelques concepts de base et la terminologie des ensembles flous.

### 3.2.1 Concepts de base de la théorie floue

#### 1) Sous-ensemble flou :

**Définition 3.2.1.1 (Sous-ensemble flou)** *Si  $X$  est une collection d'objets désignés génériquement par  $x$ , alors un sous-ensemble flou  $\tilde{A}$  dans l'univers  $X$  est un*



ensemble de paires ordonnées

$$\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) / x \in \mathbb{N}\}$$

où  $\mu_{\tilde{A}} : X \rightarrow [0, 1]$  est une application appelée fonction d'appartenance. Pour tout  $x \in X$ ,  $\mu_{\tilde{A}}(x)$  est dit le degré d'appartenance (encore degré de compatibilité ou degré de vérité) de  $x$  dans  $\tilde{A}$ .

L'ensemble de tous les sous-ensembles flous de  $X$  est noté par  $\mathcal{F}$ .

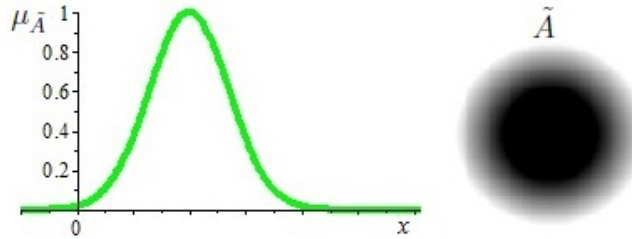


FIGURE 3.1 – Sous-ensemble flou

**Remarque 3.2.1.1** *Le sous-ensemble classique n'est en fait qu'un cas particulier de sous-ensemble flou dont la fonction d'appartenance ne prend que les valeurs 0 ou 1.*

### 2) Support d'un sous-ensemble flou :

Le support d'un sous-ensemble flou  $\tilde{A}$  est l'ensemble ordinaire défini par :

$$Supp(A) = \{x \in X / \mu_{\tilde{A}}(x) > 0\}$$

### 3) Noyau d'un sous-ensemble flou :

Le noyau d'un sous-ensemble flou  $\tilde{A}$  est l'ensemble ordinaire des éléments de  $X$  qui sont réellement dans  $A$  et défini par :

$$N(A) = \{x \in X / \mu_{\tilde{A}}(x) = 1\}$$

On dit que le sous-ensemble flou  $\tilde{A}$  est normal ou normalisé si  $N(A) \neq \emptyset$ .

### 4) Hauteur d'un sous-ensemble flou :

La hauteur d'un sous-ensemble flou  $\tilde{A}$  est défini par :

$$H(A) = \sup_{x \in X} \mu_{\tilde{A}}(x)$$

### 5) Opérations sur les sous-ensembles flous :

Les opérations classiques union et intersection des sous-ensembles ordinaires de  $X$  peuvent être étendues par les formules suivantes introduites par Zadeh [58].

#### • Inclusion et égalité de deux sous-ensembles flous :

Soient  $\tilde{A}, \tilde{B} \in \mathcal{F}(X)$ . Alors :

i)  $\tilde{A} \subset \tilde{B}$  si et seulement si  $\mu_{\tilde{A}}(x) \leq \mu_{\tilde{B}}(x)$  pour tout  $x \in X$ .

ii)  $\tilde{A} = \tilde{B}$  si et seulement si  $\mu_{\tilde{A}}(x) = \mu_{\tilde{B}}(x)$  pour tout  $x \in X$ .

#### • Intersection et réunion de deux sous-ensembles flous :

Soient  $\tilde{A}, \tilde{B} \in \mathcal{F}(X)$ . L'intersection  $\tilde{A} \cap \tilde{B}$  et la réunion  $\tilde{A} \cup \tilde{B}$  de deux sous-ensembles flous  $\tilde{A}$  et  $\tilde{B}$  de  $X$  sont définies par :

$$\mu_{\tilde{A} \cap \tilde{B}}(x) = \mu_{\tilde{A}}(x) \wedge \mu_{\tilde{B}}(x) = \min(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)), \quad \forall x \in X$$

$$\mu_{\tilde{A} \cup \tilde{B}}(x) = \mu_{\tilde{A}}(x) \vee \mu_{\tilde{B}}(x) = \max(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)), \quad \forall x \in X$$

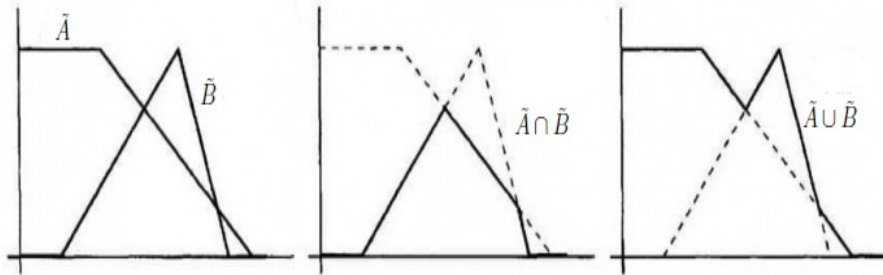


FIGURE 3.2 – Intersection & réunion floues

#### • Complémentation :

Soit  $\tilde{A} \in \mathcal{F}(X)$ . Le complémentaire de  $\tilde{A}$  est le sous-ensemble flou noté  $\tilde{A}^c$  et défini par :

$$\mu_{\tilde{A}^c}(x) = 1 - \mu_{\tilde{A}}(x), \quad \forall x \in X$$

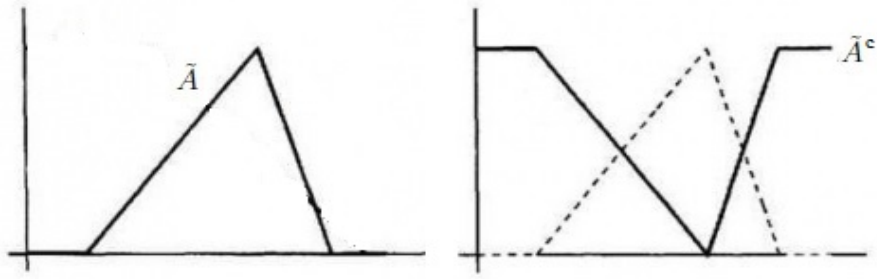


FIGURE 3.3 – Complémentaire flou

**Théorème 3.2.1.1 (Opérations algébriques sur les sous-ensembles flous)** Soient

$\tilde{A}, \tilde{B}, \tilde{C} \in \mathcal{F}(X)$ . Alors on a les propriétés suivantes :

- i)  $\tilde{A} \cap \tilde{B} = \tilde{B} \cap \tilde{A}$  et  $\tilde{A} \cup \tilde{B} = \tilde{B} \cup \tilde{A}$  (Commutativité) ;
- ii)  $(\tilde{A} \cap \tilde{B}) \cap \tilde{C} = \tilde{A} \cap (\tilde{B} \cap \tilde{C})$  et  $(\tilde{A} \cup \tilde{B}) \cup \tilde{C} = \tilde{A} \cup (\tilde{B} \cup \tilde{C})$  (Associativité) ;
- iii)  $\tilde{A} \cap \tilde{A} = \tilde{A}$  et  $\tilde{A} \cup \tilde{A} = \tilde{A}$  (Idempotence) ;
- iv)  $\tilde{A} \cap (\tilde{B} \cup \tilde{C}) = (\tilde{A} \cap \tilde{B}) \cup (\tilde{A} \cap \tilde{C})$  et  $\tilde{A} \cup (\tilde{B} \cap \tilde{C}) = (\tilde{A} \cup \tilde{B}) \cap (\tilde{A} \cup \tilde{C})$  (Distributivités mutuelles) ;
- v)  $(\tilde{A} \cap \tilde{B})^c = \tilde{A}^c \cup \tilde{B}^c$  et  $(\tilde{A} \cup \tilde{B})^c = \tilde{A}^c \cap \tilde{B}^c$  (Lois de Morgan) ;
- vi)  $(\tilde{A}^c)^c = \tilde{A}$  (Involution).

### 6) $\alpha$ -coupes d'un sous-ensemble flou :

Le concept de  $\alpha$ -coupes joue un rôle principal dans la relation entre les ensembles flous et les ensembles ordinaires. Ils peuvent être considérés comme un pont par lequel des ensembles flous et des ensembles ordinaires sont connectés. Avant d'examiner cette connexion, nous présentons d'abord quelques propriétés supplémentaires des  $\alpha$ -coupes. Toutes ces propriétés impliquent les opérations standards d'ensembles flous et l'inclusion d'ensembles flous standards.

**Définition 3.2.1.2** On appelle  $\alpha$ -coupe ou sous-ensemble de niveau  $\alpha$  du sous-ensemble flou  $\tilde{A}$ , le sous-ensemble ordinaire de  $X$  noté  $[\tilde{A}]^\alpha$  et défini par :

$$[\tilde{A}]^\alpha = \{x \in X / \mu_{\tilde{A}}(x) \geq \alpha\} \quad 0 \leq \alpha \leq 1.$$

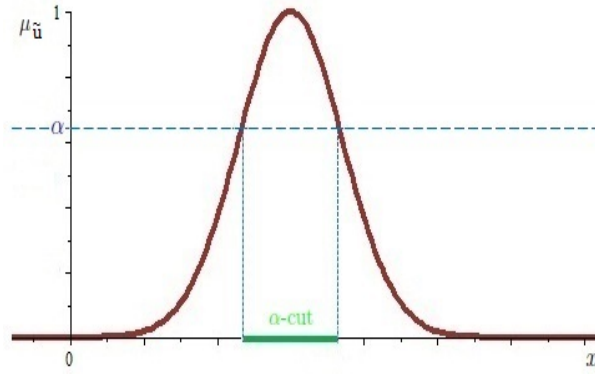


FIGURE 3.4 –  $\alpha$ -coupe d'un sous-ensemble flou

**Cas particuliers :**

Pour  $\alpha = 0$ ,  $[\tilde{A}]^\alpha = X$  et pour  $\alpha = 1$ ,  $[\tilde{A}]^\alpha = N(A)$ .

**Remarque 3.2.1.2** *La fonction d'appartenance d'un sous-ensemble flou  $\tilde{A} \in \mathcal{F}(X)$  peut être exprimée en terme des fonctions caractéristiques de ses  $\alpha$ -coupes par la formule*

$$\mu_{\tilde{A}}(x) = \sup_{\alpha \in ]0,1]} \min(\alpha, \chi_{[\tilde{A}]^\alpha}(x))$$

où

$$\forall x \in X, \quad \chi_{[\tilde{A}]^\alpha}(x) = \begin{cases} 1 & \text{si } x \in [\tilde{A}]^\alpha, \\ 0 & \text{sinon.} \end{cases}$$

**Théorème 3.2.1.2 (Opérations algébriques)** *Soient  $\tilde{A}, \tilde{B} \in \mathcal{F}(X)$ . Pour tout  $\alpha \in [0, 1]$  on a les assertions suivantes :*

- i)  $\alpha \leq \beta \Rightarrow [\tilde{A}]^\beta \subset [\tilde{A}]^\alpha$  ;
- ii)  $[\tilde{A} \cap \tilde{B}]^\alpha = [\tilde{A}]^\alpha \cap [\tilde{B}]^\alpha$  ;
- iii)  $[\tilde{A} \cup \tilde{B}]^\alpha = [\tilde{A}]^\alpha \cup [\tilde{B}]^\alpha$ .

**Preuve 3.2.1.1** i) *Soit  $x \in [\tilde{A}]^\beta$ . Donc,  $\mu_{\tilde{A}}(x) \geq \beta \geq \alpha$ . Par suite,  $x \in [\tilde{A}]^\alpha$ .*

ii) *On a :*

$$\begin{aligned} x \in [\tilde{A} \cap \tilde{B}]^\alpha &\Leftrightarrow \mu_{\tilde{A} \cap \tilde{B}}(x) \geq \alpha \Leftrightarrow \min(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)) \geq \alpha \\ &\Leftrightarrow \mu_{\tilde{A}}(x) \geq \alpha \text{ et } \mu_{\tilde{B}}(x) \geq \alpha \Leftrightarrow x \in [\tilde{A}]^\alpha \text{ et } x \in [\tilde{B}]^\alpha \\ &\Leftrightarrow x \in [\tilde{A}]^\alpha \cap [\tilde{B}]^\alpha \end{aligned}$$

iii) De même on a :

$$\begin{aligned} x \in [\tilde{A} \cup \tilde{B}]^\alpha &\Leftrightarrow \mu_{\tilde{A} \cup \tilde{B}}(x) \geq \alpha \Leftrightarrow \max(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)) \geq \alpha \\ &\Leftrightarrow \mu_{\tilde{A}}(x) \geq \alpha \text{ ou } \mu_{\tilde{B}}(x) \geq \alpha \Leftrightarrow x \in [\tilde{A}]^\alpha \text{ ou } x \in [\tilde{B}]^\alpha \\ &\Leftrightarrow x \in [\tilde{A}]^\alpha \cup [\tilde{B}]^\alpha \end{aligned}$$

□

**Théorème 3.2.1.3 (Généralisation)** Soit  $I$  un ensemble non vide et  $(\tilde{A}_i)_{i \in I}$  une famille d'éléments dans  $\mathcal{F}(X)$ . Alors :

$$\begin{aligned} i) & \bigcup_{i \in I} [\tilde{A}_i]^\alpha \subset [\bigcup_{i \in I} \tilde{A}_i]^\alpha ; \\ ii) & \bigcap_{i \in I} [\tilde{A}_i]^\alpha = [\bigcap_{i \in I} \tilde{A}_i]^\alpha . \end{aligned}$$

**Preuve 3.2.1.2** i) On a :

$$\begin{aligned} x \in \bigcup_{i \in I} [\tilde{A}_i]^\alpha &\Rightarrow \exists i \in I, x \in [\tilde{A}_i]^\alpha \Rightarrow \exists i \in I, \mu_{\tilde{A}_i}(x) \geq \alpha \Rightarrow \sup_{i \in I} \mu_{\tilde{A}_i}(x) \geq \alpha \\ &\Rightarrow \mu_{(\bigcup_{i \in I} \tilde{A}_i)}(x) \geq \alpha \Rightarrow x \in [\bigcup_{i \in I} \tilde{A}_i]^\alpha \end{aligned}$$

ii) De même on :

$$\begin{aligned} x \in \bigcap_{i \in I} [\tilde{A}_i]^\alpha &\Leftrightarrow \forall i \in I, x \in [\tilde{A}_i]^\alpha \Leftrightarrow \forall i \in I, \mu_{\tilde{A}_i}(x) \geq \alpha \Leftrightarrow \inf_{i \in I} \mu_{\tilde{A}_i}(x) \geq \alpha \\ &\Leftrightarrow \mu_{(\bigcap_{i \in I} \tilde{A}_i)}(x) \geq \alpha \Leftrightarrow x \in [\bigcap_{i \in I} \tilde{A}_i]^\alpha \end{aligned}$$

□

**Théorème 3.2.1.4 (Caractérisation d'un s-e.f par les  $\alpha$ -coupes)** Soient  $\tilde{A}, \tilde{B} \in \mathcal{F}(X)$ . Alors :

$$\begin{aligned} i) & \tilde{A} \subset \tilde{B} \Leftrightarrow \forall \alpha \in [0, 1], [\tilde{A}]^\alpha \subset [\tilde{B}]^\alpha ; \\ ii) & \tilde{A} = \tilde{B} \Leftrightarrow \forall \alpha \in [0, 1], [\tilde{A}]^\alpha = [\tilde{B}]^\alpha . \end{aligned}$$

**Preuve 3.2.1.3** i) Le sens direct est immédiat. Pour le sens réciproque, soit  $x \in X$ . On pose  $\alpha = \mu_{\tilde{A}}(x)$ . On a donc

$$\mu_{\tilde{A}}(x) = \alpha \geq \alpha \Rightarrow x \in [\tilde{A}]^\alpha = [\tilde{B}]^\alpha \Rightarrow \mu_{\tilde{B}}(x) \geq \alpha = \mu_{\tilde{A}}(x) \Rightarrow \mu_{\tilde{A}}(x) \leq \mu_{\tilde{B}}(x)$$

Donc :  $\tilde{A} \subset \tilde{B}$

ii) Cette équivalence est une conséquence de la première équivalence (i).  $\square$

### 7) Nombres flous :

Sur l'ensemble  $\mathcal{R}$  des nombres réels, on définit les sous-ensembles flous définis par leurs fonctions d'appartenance  $\mu_{\tilde{A}} : \mathbb{R} \rightarrow [0, 1]$

**Définition 3.2.1.3 (Nombre flou)** *Un nombre flou est un sous-ensemble flou  $\tilde{A}$  de  $\mathbb{R}$  vérifiant les conditions suivantes :*

- i)  $\tilde{A}$  est normalisé, i-e : il existe  $x \in \mathbb{R}$  tel que  $\mu_{\tilde{A}}(x) = 1$  ;
- ii)  $\tilde{A}$  est convexe, i-e :  $\mu_{\tilde{A}}(\lambda x + (1-\lambda)y) \geq \min(\mu_{\tilde{A}}(x), \mu_{\tilde{A}}(y))$ ,  $\forall x, y \in \mathbb{R}, \forall \lambda \in [0, 1]$  ;
- iii)  $\tilde{A}$  est semi-continue supérieurement ;
- iv) La coupe  $[\tilde{A}]^0$  de niveau 0 est compacte.

L'ensemble des nombres flous est représenté par  $\mathcal{F}(\mathbb{R})$ .

**Remarque 3.2.1.3** *Le sous-ensemble flou  $\tilde{A} \in \mathcal{F}(\mathbb{R})$  est convexe si et seulement si le sous-ensemble ordinaire  $[\tilde{A}]^\alpha$  de  $\mathbb{R}$  est convexe (intervalle) pour tout  $\alpha \in [0, 1]$ .*

### 8) Opérations arithmétiques floues : approche fonctionnelle

L'addition, la soustraction et la multiplication par un scalaire dans  $\mathcal{F}(\mathbb{R})$  sont définies de la manière suivante :

Pour tous  $\tilde{u}, \tilde{v} \in \mathcal{F}(\mathbb{R})$  et  $\lambda \geq 0$ ,

$$(\tilde{u} + \tilde{v})(x) := \sup_{x_1, x_2 \in \mathbb{R}/x_1+x_2=x} \min(\tilde{u}(x_1), \tilde{v}(x_2)) \quad (3.1)$$

$$(\tilde{u} - \tilde{v})(x) := \sup_{x_1, x_2 \in \mathbb{R}/x_1-x_2=x} \min(\tilde{u}(x_1), \tilde{v}(x_2)) \quad (3.2)$$

$$(\lambda \tilde{u})(x) := \begin{cases} \tilde{u}(x/\lambda) & \text{if } \lambda \in \mathbb{R} - \{0\}, \\ 1_{\{0\}}(x) & \text{if } \lambda = 0. \end{cases} \quad (3.3)$$

où  $1_E$  est la fonction indicatrice de l'ensemble ordinaire  $E$ .

### 9) Opérations arithmétiques floues : approche $\alpha$ -coupes

En utilisant les opérations algébriques ensemblistes  $A + B := \{x + y/x \in A, y \in B\}$  et  $\lambda A := \{\lambda x/x \in A\}$ . Pour tous  $A, B \subset \mathbb{R}$ , il vient immédiatement :

$$[\tilde{u} + \tilde{v}]^\alpha = [\tilde{u}]^\alpha + [\tilde{v}]^\alpha \quad \text{and} \quad [\lambda \tilde{u}]^\alpha = \lambda [\tilde{u}]^\alpha \quad (\forall \alpha \in [0, 1]) \quad (3.4)$$

Soient  $\rho$  la distance de Hausdorff définie sur l'ensemble  $P_K(\mathbb{R})$  des parties compactes convexes non vides de  $\mathbb{R}$  et  $d$  la distance définie sur  $\mathbb{R}$  par  $d(x, D) = \min_{y \in D} d(x, y)$  pour  $x \in \mathbb{R}$  et  $D \in P_K(\mathbb{R})$ .

Par extension de la métrique  $\rho$  sur  $\mathcal{F}(\mathbb{R})$ , on définit par abus de notation la distance de Hausdorff sur  $\mathcal{F}(\mathbb{R})$  par :

$$\rho(\tilde{u}, \tilde{v}) = \sup_{\alpha \in [0,1]} \rho([\tilde{u}]^\alpha, [\tilde{v}]^\alpha) \quad \text{for } \tilde{u}, \tilde{v} \in \mathcal{F}(\mathbb{R})$$

On rappelle que les faits suivants sont bien connus (voir Diamond et Kloeden [36]; Dubois et Prade [38]; Klir et Yuan [44]; Kurano et al. [45]; Puri et Ralesca [52]).

**Lemme 3.2.1.1** *L'espace métrique  $(\mathcal{F}(\mathbb{R}), \rho)$  est complet.*

**Lemme 3.2.1.2** *Si  $\tilde{u}, \tilde{v}, \tilde{u}', \tilde{v}'$  and  $\tilde{w} \in \mathcal{F}(\mathbb{R})$ , alors*

- i)  $\rho(\lambda\tilde{u}, \lambda\tilde{v}) = \lambda\rho(\tilde{u}, \tilde{v})$  for all  $\lambda \geq 0$ ,
- ii)  $\rho(\tilde{u} + \tilde{u}', \tilde{v} + \tilde{v}') = \rho(\tilde{u}, \tilde{v}) + \rho(\tilde{u}', \tilde{v}')$ ,
- iii)  $\rho(\tilde{w} + \tilde{u}, \tilde{w} + \tilde{v}) = \rho(\tilde{u}, \tilde{v})$

### 10) Nombre flou trapézoïdal :

**Définition 3.2.1.4 ( Nombre flou trapézoïdal)** *Un nombre flou trapézoïdal  $\tilde{a}$  est un nombre flou qui est défini par le quadruplet  $[a_L, a_U, \alpha, \beta]$ ,  $\alpha, \beta \geq 0$ . Sa fonction d'appartenance est définie par :*

$$\tilde{a}(x) = \begin{cases} 1 + \frac{x-a_L}{\alpha} & ; \quad a_L - \alpha \leq x \leq a_L, \\ 1 & ; \quad a_L \leq x \leq a_U. \\ 1 + \frac{a_U-x}{\beta} & ; \quad a_U \leq x \leq a_U + \beta. \end{cases}$$

Pour des nombres flous trapézoïdaux  $\tilde{a} = [a_L, a_U, \alpha, \beta]$  et  $\tilde{b} = [b_L, b_U, \delta, \gamma]$  et  $\lambda \in \mathbb{R}$ , on définit l'addition et la multiplication par un scalaire de la façon suivante :

$$\tilde{a} + \tilde{b} = [a_L + b_L, a_U + b_U, \alpha + \delta, \beta + \gamma] \quad (3.5)$$

$$\lambda\tilde{a} = \begin{cases} [\lambda a_L, \lambda a_U, \lambda\alpha, \lambda\beta] & \text{if } \lambda \geq 0 \\ (\lambda a_U, \lambda a_L, -\lambda\alpha, -\lambda\beta) & \text{if } \lambda < 0 \end{cases} \quad (3.6)$$

### 11) Nombre flou triangulaire :

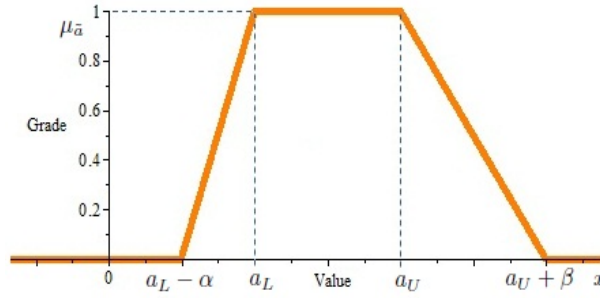


FIGURE 3.5 – Trapezoidal Fuzzy Number

**Définition 3.2.1.5 (Nombre flou triangulaire)** *Un nombre flou triangulaire  $\tilde{a}$  est un nombre flou qui est défini par le triplet  $(a_1, a_2, a_3)$  tel que  $a_1, a_2, a_3 \in \mathbb{R}$  et  $a_1 \leq a_2 \leq a_3$ . Sa fonction d'appartenance est définie par :*

$$\mu_{\tilde{a}}(x) = \begin{cases} \frac{x-a_1}{a_2-a_1} & ; a_1 \leq x \leq a_2, \\ \frac{a_3-x}{a_3-a_2} & ; a_2 \leq x \leq a_3, \\ 0 & ; otherwise \end{cases}$$

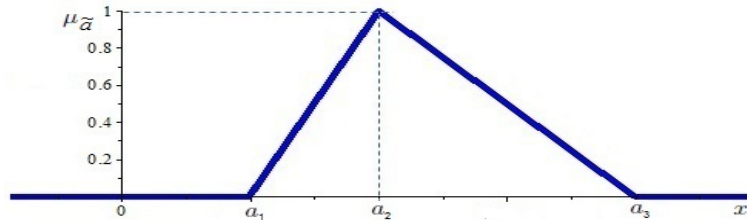


FIGURE 3.6 – Nombre flou triangulaire

Pour des nombres flous triangulaires  $\tilde{a} = (a_1, a_2, a_3)$ ,  $\tilde{b} = (b_1, b_2, b_3)$  et  $\lambda \in \mathbb{R}$ , on définit l'addition et la multiplication par un scalaire de la façon suivante :

$$\tilde{a} + \tilde{b} = (a_1 + b_1, a_2 + b_2, a_3 + b_3) \quad (3.7)$$

$$\lambda \tilde{a} = \begin{cases} (\lambda a_1, \lambda a_2, \lambda a_3) & \text{if } \lambda \geq 0 \\ (\lambda a_3, \lambda a_2, \lambda a_1) & \text{if } \lambda < 0 \end{cases} \quad (3.8)$$

## 12) Forme paramétrique d'un nombre flou :

**Définition 3.2.1.6 (Forme paramétrique)** *Dans la forme paramétrique, un nombre flou  $\tilde{u}$  est représenté par la paire  $(\underline{u}(\alpha), \bar{u}(\alpha))$ ,  $0 \leq \alpha \leq 1$ , qui satisfont les exigences*



suivantes :

- i)  $\underline{u}$  est bornée, continue à gauche et croissante sur  $[0, 1]$  ;
- ii)  $\bar{u}$  est bornée, continue à gauche et décroissante sur  $[0, 1]$  ;
- iii)  $\underline{u}(\alpha) \leq \bar{u}(\alpha), \forall \alpha \in [0, 1]$ .

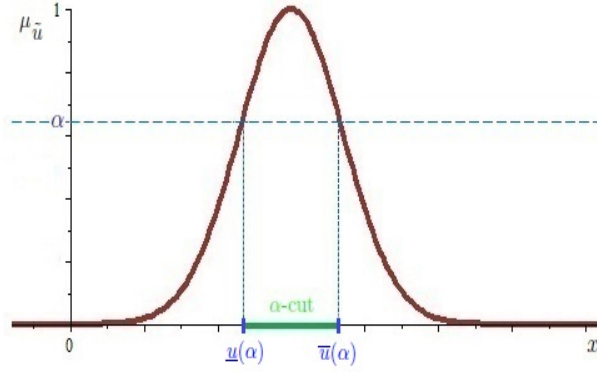


FIGURE 3.7 – Parametric form of fuzzy set

Pour des nombres flous arbitraires  $\tilde{u} = (\underline{u}, \bar{u})$ ,  $\tilde{v} = (\underline{v}, \bar{v})$  et  $\lambda \in \mathbb{R}$ , on définit l'addition et la multiplication par un scalaire de la façon suivante :

$$\underline{u} + \underline{v} = \underline{u} + \underline{v} \quad \text{and} \quad \overline{u + v} = \bar{u} + \bar{v} \quad (3.9)$$

$$\lambda \tilde{u} = \begin{cases} (\lambda \underline{u}, \lambda \bar{u}) & \text{if } \lambda \geq 0 \\ (\lambda \bar{u}, \lambda \underline{u}) & \text{if } \lambda < 0 \end{cases} \quad (3.10)$$

**13) Comparaison de deux nombres flous :** Le classement est une approche disponible pour comparer des ensembles flous sur  $\mathbb{R}$ . Divers types de fonctions de classement ont été utilisés pour résoudre des problèmes d'optimisation floue (Voir Dubois et Prade [37]; Semmouri et al. [39]; Mahdavi-Amiri et Nasseri [48]; Yager [57]). Une approche efficace pour ordonner les éléments de  $\mathcal{F}(\mathbb{R})$  est de définir une fonction de classement  $\mathcal{R} : \mathcal{F}(\mathbb{R}) \rightarrow \mathbb{R}$  mappant des ensembles flous dans  $\mathbb{R}$ . Considérons  $\tilde{u}$  et  $\tilde{v}$  dans  $\mathcal{F}(\mathbb{R})$ , on définit l'ordre sur  $\mathcal{F}(\mathbb{R})$  comme suit :

$$\tilde{u} \geq_{\mathcal{R}} \tilde{v} \quad \text{if} \quad \mathcal{R}(\tilde{u}) \geq \mathcal{R}(\tilde{v}) \quad (3.11)$$

$$\tilde{u} >_{\mathcal{R}} \tilde{v} \quad \text{if} \quad \mathcal{R}(\tilde{u}) > \mathcal{R}(\tilde{v}) \quad (3.12)$$

$$\tilde{u} =_{\mathcal{R}} \tilde{v} \quad \text{if} \quad \mathcal{R}(\tilde{u}) = \mathcal{R}(\tilde{v}) \quad (3.13)$$

On écrit aussi  $\tilde{u} \leq_{\mathcal{R}} \tilde{v}$  si  $\tilde{v} \geq_{\mathcal{R}} \tilde{u}$ . Alors, pour toute fonction de classement linéaire  $\mathcal{R}$  on peut obtenir :  $\tilde{u} \geq_{\mathcal{R}} \tilde{v}$  si et seulement si  $\tilde{u} - \tilde{v} \geq_{\mathcal{R}} \tilde{0}$  où  $\tilde{0} = 1_{\{0\}}$ , ou si et seulement si  $-\tilde{v} \geq -\tilde{u}$ . De plus, si  $\tilde{u} \geq_{\mathcal{R}} \tilde{v}$  et  $\tilde{u}' \geq_{\mathcal{R}} \tilde{v}'$ , alors  $\tilde{u} + \tilde{u}' \geq_{\mathcal{R}} \tilde{v} + \tilde{v}'$ .

**Exemple 3.2.1.1** On considère une fonction de classement linéaire sur un nombre flou trapézoïdal  $\tilde{a} = (a_L, a_U, \alpha, \beta) \in \mathcal{F}(\mathbb{R})$  définie par :

$$\mathcal{R}(\tilde{a}) = \frac{a_L + a_U}{2} + \frac{\beta - \alpha}{4}$$

La fonction de classement linéaire ci-dessus a été proposée par Yager [57].

**Exemple 3.2.1.2** Comme le nombre flou triangulaire est un cas particulier du nombre flou trapézoïdal ( $a_L = a_U = a$ ), la fonction de classement vue dans l'exemple 1., devient

$$\mathcal{R}(\tilde{a}) = a + \frac{\beta - \alpha}{4}$$

Soient  $\tilde{u}, \tilde{v} \in \mathcal{F}(\mathbb{R})$  et  $\tilde{U} = (\tilde{u}_1, \tilde{u}_2, \dots, \tilde{u}_n), \tilde{V} = (\tilde{v}_1, \tilde{v}_2, \dots, \tilde{v}_n) \in (\mathcal{F}(\mathbb{R}))^n$ . On dit que  $\max_{\mathcal{R}}(\tilde{u}, \tilde{v}) = \tilde{u}$  if  $\tilde{u} \geq_{\mathcal{R}} \tilde{v}$ , et  $\tilde{U} \geq_{\mathcal{R}} \tilde{V}$  si  $\tilde{u}_i \geq_{\mathcal{R}} \tilde{v}_i$  pour tout  $i \in S$ . Sans perte de généralité, on peut poser  $\mathcal{R}(\tilde{U}) = (\mathcal{R}(\tilde{u}_1), \mathcal{R}(\tilde{u}_2), \dots, \mathcal{R}(\tilde{u}_n))$ .

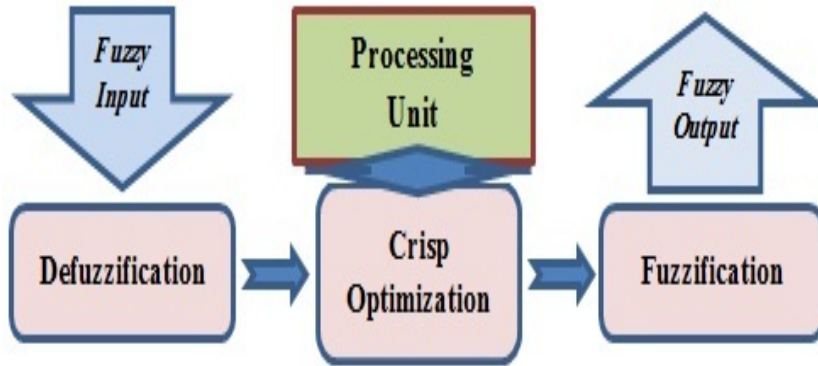


FIGURE 3.8 – Fuzzification

### 3.3 Le modèle du coût flou actualisé

#### 3.3.1 Notions préliminaires

On considère un PDM discret défini par le 5-uplet :

$$(S, A = \bigcup_{i \in S} A(i), Q = \{p(j/i, a)\}, c = \{c(i, a)\}, \gamma)$$

où  $S$  est l'espace d'états,  $A$  est l'espace d'action,  $p(j/i, a)$  est une fonction qui détermine la probabilité que l'action  $a$  dans l'état  $i$  au moment  $n$  conduise à l'état  $j$  au temps  $n$ ,  $\tilde{c}(i, a)$  est une fonction qui renvoie le coût fluu immédiat payé après le passage à l'état  $i$  à partir des états précédents et  $\gamma$  est le facteur d'actualisation. On dit qu'un PDM est fini si l'espace des états et les ensembles d'actions sont finis. Dans cet exposé, on s'intéressera au PDMs finis avec stationnarité des probabilités de transition et des coûts mis en jeu.

En utilisant la décision  $\pi^n$  à l'instant  $n$ , on définit le coût fluu immédiat espéré par le vecteur  $\tilde{c}(\pi^n)$  de  $(\mathcal{F}(\mathbb{R}))^p$  de composantes  $[\tilde{c}(i, \pi^n)]_{i \in S}$  définies par :

$$\tilde{c}(i, \pi^n) := \sum_{a \in A(i)} \pi_{i,a}^n \tilde{c}(i, a)$$

Pour que l'agent puisse minimiser le coût fluu de son interaction avec l'environnement, il doit être capable d'évaluer la valeur d'un état et de mettre en œuvre une cartographie des états aux probabilités de sélection de chaque action possible à chaque pas de temps. Ce mappage est appelé stratégie de l'agent et est noté  $\pi$ , où  $\pi_{i,a}^n$  est la probabilité que l'action  $a$  soit sélectionnée au temps  $n$  si nous sommes dans l'état  $i$  au temps  $n$ . La valeur estimée d'un état est définie en termes de coûts flous futurs auxquels on peut s'attendre. Bien entendu, les coûts flous auxquels l'agent peut s'attendre à l'avenir dépendent des actions qu'il entreprendra. En conséquence, sa fonction de valeur est définie par rapport à une stratégie particulière.

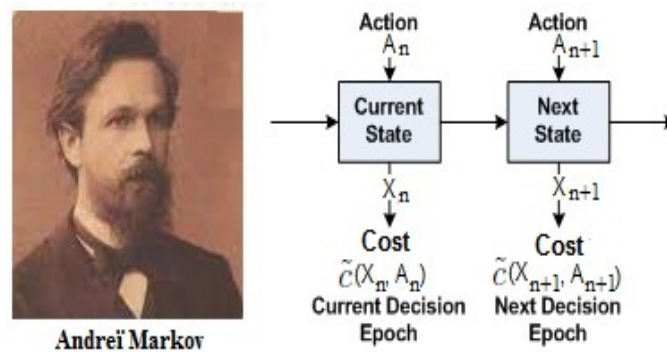


FIGURE 3.9 – Le modèle du PDM fluu

Dans ce cas on va généraliser le critère actualisé classique établi dans Bertsekas et Shreve [29], Canbolat et Rothblum [33], Derman [35], Powell [51] et Puterman [53].

Par conséquent, la valeur de l'état  $i$  sous la stratégie  $\pi$ , notée  $\tilde{\mathcal{C}}_\gamma(i, \pi)$ , est le coût espéré en partant de l'état  $i$  et suivant la stratégie  $\pi$  par la suite et peut être définie formellement à horizon fini  $T$  par :

$$\tilde{\mathcal{C}}_T(\gamma, i, \pi) := \sum_{n=1}^T \gamma^{n-1} \sum_{j \in S} \sum_{a \in A(j)} \mathbb{P}_\pi^i(X_n = j, A_n = a) \tilde{c}(j, a)$$

Il est clair que  $\tilde{\mathcal{C}}_T(\gamma, i, \pi) \in \mathcal{F}(\mathbb{R})$  pour tout  $i \in S$ ,  $\pi \in \Pi$  et  $T \geq 1$ .

Soit  $p$  le cardinal de l'espace des états  $S$  ( $p = |S|$ ). On définit :

$$(\mathcal{F}(\mathbb{R}))^p := \{\tilde{U} = (\tilde{u}_1, \tilde{u}_2, \dots, \tilde{u}_p) / \tilde{u}_i \in \mathcal{F}(\mathbb{R}), 1 \leq i \leq p\}$$

La distance de Hausdorff  $\rho$  on  $(\mathcal{F}(\mathbb{R}))^p$  est définie (par abus de notation) par :

$$\rho(\tilde{U}, \tilde{V}) = \max_{1 \leq i \leq p} (\rho(\tilde{u}_i, \tilde{v}_i))$$

où  $\tilde{U} = (\tilde{u}_1, \tilde{u}_2, \dots, \tilde{u}_p)$ ,  $\tilde{V} = (\tilde{v}_1, \tilde{v}_2, \dots, \tilde{v}_p) \in (\mathcal{F}(\mathbb{R}))^p$  et  $\rho(\tilde{u}_i, \tilde{v}_i)$  déjà définie. Ainsi, on admet que  $((\mathcal{F}(\mathbb{R}))^p, \rho)$  est un espace métrique complet.

Pour toute matrice carrée  $A = (a_{i,j})$  d'ordre  $p$  constituée de nombres réels et tout vecteur  $\tilde{U} = (\tilde{u}_1, \tilde{u}_2, \dots, \tilde{u}_p) \in (\mathcal{F}(\mathbb{R}))^p$ , le produit matriciel  $A\tilde{U}$  sera défini par sa  $i^{\text{me}}$  composante

$$(A\tilde{U})_i = \sum_{j=1}^p a_{i,j} \tilde{u}_j \quad (1 \leq i \leq p) \quad (3.14)$$

On associe à tout élément  $f$  de l'ensemble des applications  $A^S$  le vecteur colonne flou  $\tilde{c}(f) \in (\mathcal{F}(\mathbb{R}))^p$  de composante  $\tilde{c}(i, f(i))$ .

Pour toute stratégie  $\pi \in \Pi$ ,  $\tilde{\mathcal{C}}_T(\gamma, \pi)$  désigne le vecteur de composantes  $\tilde{\mathcal{C}}_T(\gamma, i, \pi)$ ,  $i \in S$  ( $T \geq 1$ ). Ainsi, on établit le lemme suivant.

**Lemme 3.3.1.1** *Pour  $\pi = (\pi^1, \pi^2, \dots) \in \Pi$ , on a :*

(i)  $\tilde{\mathcal{C}}_T(\gamma, \pi)$  est décrit en notation matricielle par

$$\tilde{\mathcal{C}}_T(\gamma, \pi) = \sum_{n=1}^T \gamma^{n-1} P(\pi^1) P(\pi^2) \dots P(\pi^{n-1}) \tilde{c}(\pi^n) \quad (3.15)$$

(ii) La suite  $\{\tilde{\mathcal{C}}_T(\gamma, \pi)\}_{T \geq 1}$  est une suite de Cauchy pour la distance de Hausdorff  $\rho$ .

**Preuve 3.3.1.1** *Pour tout entier naturel  $n \geq 1$  on a*

$$\sum_{j \in S} P_\pi^i(X_n = j) \tilde{c}(j, \pi^n(j)) = \sum_{j \in S} [P(\pi^1) P(\pi^2) \dots P(\pi^n)]_{i,j} \tilde{c}(j, \pi^n(j))$$

Pour tous  $T, N \geq 1$ , Il découle du lemme précédent que

$$\begin{aligned} \rho(\tilde{\mathcal{C}}_{T+N}(\gamma, i, \pi), \tilde{\mathcal{C}}_T(\gamma, i, \pi)) &= \rho(\tilde{0}, \sum_{n=T+1}^{T+N} \gamma^{n-1} \sum_{j \in S} \sum_{a \in A(j)} \mathbb{P}_\pi^i(X_n = j, A_n = a) \tilde{c}(j, a)) \\ &= \gamma^T \rho(\tilde{0}, \sum_{n=T+1}^{T+N} \gamma^{n-T-1} \sum_{j \in S} \sum_{a \in A(j)} \mathbb{P}_\pi^i(X_n = j, A_n = a) \tilde{c}(j, a)) \leq \frac{c}{1-\gamma} \gamma^T \end{aligned}$$

Pour tout  $i \in S$  et en posant  $c = \max_{j \in S, a \in A(j)} \rho(\tilde{0}, \tilde{c}(j, a))$

Par conséquent

$$\rho(\tilde{\mathcal{C}}_{T+N}(\gamma, \pi), \tilde{\mathcal{C}}_T(\gamma, \pi)) \leq \frac{c}{1-\gamma} \gamma^T$$

Ce qui montre (ii). □

Du lemme précédent, on peut établir la définition suivante.

**Définition 3.3.1.1** Pour toute stratégie  $\pi \in \Pi$  on définit le coût total flou actualisé à horizon infini dans lequel le coût flou future est actualisé par le facteur d'actualisation  $\gamma$  ( $0 < \gamma < 1$ ) par

$$\tilde{\mathcal{C}}(\gamma, \pi) := \lim_{T \rightarrow \infty} \tilde{\mathcal{C}}_T(\gamma, \pi)$$

i-e :

$$\tilde{\mathcal{C}}(\gamma, i, \pi) := \lim_{T \rightarrow \infty} \tilde{\mathcal{C}}_T(\gamma, i, \pi), \quad \forall i \in S$$

où  $\tilde{\mathcal{C}}(\gamma, i, \pi)$  est la  $i^{\text{me}}$  composante du vecteur valeur floue  $\tilde{\mathcal{C}}(\gamma, \pi)$ .

**Théorème 3.3.1.1** Pour toute stratégie stationnaire  $\pi$ , le vecteur valeur floue actualisé vérifie :

$$\tilde{\mathcal{C}}(\gamma, \pi) = (I - \gamma P(\pi))^{-1} \tilde{c}(\pi)$$

où  $I$  est la matrice unité d'ordre  $p$  et  $A^{-1}$  est la matrice inverse de la matrice carrée  $A$  d'ordre  $p$ .

**Preuve 3.3.1.2** Il suffit d'appliquer le lemme précédent. □

Il est clair que le vecteur  $\tilde{\mathcal{C}}(\gamma, \pi)$  est l'unique solution de l'équation

$$\tilde{\mathcal{C}}(\gamma, \pi) = \tilde{c} + \gamma P(\pi) \tilde{\mathcal{C}}(\gamma, \pi) \tag{3.16}$$

### 3.3.2 Résultats

Dans cette sous-section, nous allons transformer le problème de déterminer une politique optimale liée à la fonction de classement choisie en l'équation d'optimalité de Bellman. Ainsi, on fait appel à l'approche de programmation dynamique.

**Définition 3.3.2.1** Soit  $\mathcal{R}$  une fonction de classement linéaire sur  $\mathcal{F}(\mathbb{R})$ . Une stratégie  $\pi^* \in \Pi$  est dite optimale si

$$\tilde{\mathcal{C}}(\gamma, i, \pi^*) =_{\mathcal{R}} \inf_{\pi \in \Pi} \tilde{\mathcal{C}}(\gamma, i, \pi) := \tilde{\mathcal{C}}^*(\gamma, i), \quad \forall i \in S$$

**Théorème 3.3.2.1** Le vecteur valeur  $\tilde{\mathcal{C}}^*(\gamma)$  de composantes  $\{\tilde{\mathcal{C}}^*(\gamma, i)\}_{i \in S}$  satisfait l'équation d'optimalité de Bellman

$$\tilde{\mathcal{C}}_i =_{\mathcal{R}} \min_{a \in A(i)} \{ \tilde{c}(i, a) + \gamma \sum_{j \in S} p(j/i, a) \tilde{\mathcal{C}}_j \}, \quad i \in S$$

**Preuve 3.3.2.1** De (3.16) il découle que pour tout  $i \in S$  on a

$$\tilde{\mathcal{C}}^*(\gamma, i) = \tilde{c}(i, a_i) + \gamma \sum_{j \in S} p(j/i, a) \tilde{\mathcal{C}}^*(\gamma, j) \quad (3.17)$$

Puisque toute politique optimale obtient au moins ce montant, nous avons

$$\tilde{\mathcal{C}}^*(\gamma, i) \leq_{\mathcal{R}} \min_{a \in A(i)} \{ \tilde{c}(i, a) + \gamma \sum_{j \in S} p(j/i, a) \tilde{\mathcal{C}}^*(\gamma, j) \} \quad (3.18)$$

De (3.17), on déduit que

$$\min_{a \in A(i)} \{ \tilde{c}(i, a) + \gamma \sum_{j \in S} p(j/i, a) \tilde{\mathcal{C}}^*(\gamma, j) \} \leq_{\mathcal{R}} \tilde{\mathcal{C}}^*(\gamma, i) \quad (3.19)$$

Donc, de (3.18) et (3.19) on conclut que

$$\tilde{\mathcal{C}}^*(\gamma, i) =_{\mathcal{R}} \min_{a \in A(i)} \{ \tilde{c}(i, a) + \gamma \sum_{j \in S} p(j/i, a) \tilde{\mathcal{C}}^*(\gamma, j) \} \quad (3.20)$$

□

En conséquence, comme  $\mathcal{R}$  est une fonction de classement linéaire et à partir de (3.20), nous obtenons l'équation d'optimalité de Bellman suivante

$$W_i =_{\mathcal{R}} \min_{a \in A(i)} \{ c'(i, a) + \gamma \sum_{j \in S} p(j/i, a) W_j \}, \quad i \in S \quad (3.21)$$

où  $W_i = \mathcal{R}(\tilde{\mathcal{C}}^*(\gamma, i))$  et  $c'(i, a) = \mathcal{R}(\tilde{c}(i, a))$ .

**Théorème 3.3.2.2** Une stratégie optimale pour minimiser le coût flou total actualisé est obtenue par un algorithme standard de programmation dynamique.

**Preuve 3.3.2.2** L'équivalence entre la  $\mathcal{R}$ -minimisation de  $\tilde{\mathcal{C}}(\lambda, \pi)$  et la minimisation de  $\mathcal{R}(\tilde{\mathcal{C}}(\lambda, \pi))$  termine la preuve.  $\square$

Ainsi, nous appliquons l'algorithme d'itération de valeur ou l'algorithme d'itération de politique pour déterminer une politique optimale ou presque optimale pour l'optimisation floue.

**Exemple 3.3.2.1** On considère le problème décisionnel de Markov avec des coûts flous suivant :

$$S = \{1, 2\}; A = \{a_1, a_2\}; A(1) = A(2) = \{a_1, a_2\}$$

Probabilités de transition :  $p_{1a_11} = 0,5$ ;  $p_{1a_12} = 0,5$ ;  $p_{2a_12} = 1$ ;  $p_{1a_21} = 1$ ;

$p_{2a_21} = 0,6$ ;  $p_{2a_22} = 0,4$ .

Coûts flous triangulaires :  $\tilde{c}(1, a_1) = [10, 1, 2]$ ;  $\tilde{c}(1, a_2) = [8, 2, 2]$ ;

$\tilde{c}(2, a_1) = [6, 3, 1]$ ;  $\tilde{c}(2, a_2) = [11, 1, 3]$ .

Facteur d'actualisation :  $\gamma = 0,9$

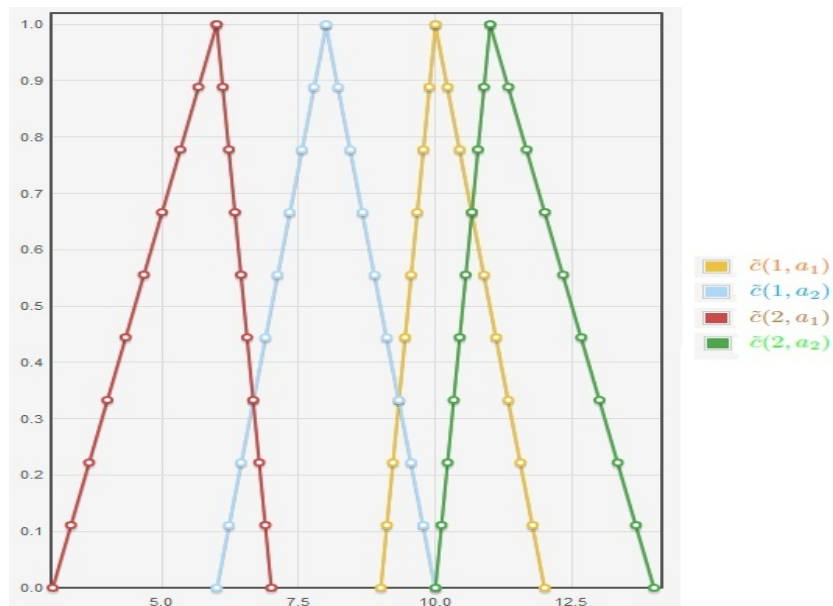


FIGURE 3.10 – Fonctions d'appartenance des coûts flous

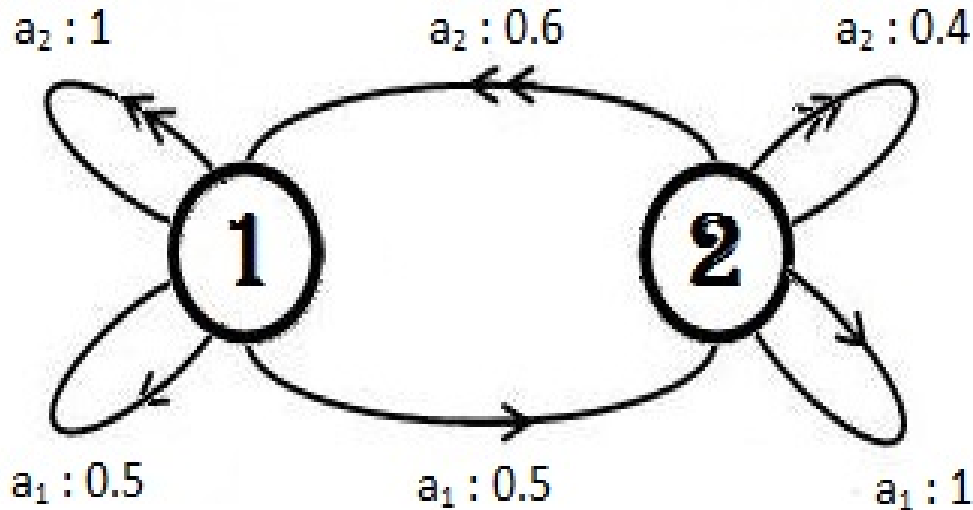


FIGURE 3.11 – Représentation graphique de l'exemple 3.3.2.1

Pour comparer les coûts flous, on utilise la fonction de classement  $\mathcal{R}$  définie sur tout nombre flou triangulaire  $\tilde{x} = [x, \alpha, \beta]$  par

$$\mathcal{R}(\tilde{x}) = x + \frac{\beta - \alpha}{4}$$

Par la fonction de classement précédente, nous transformons les coûts flous en coûts ordinaires (réels) :

$$c'(1, a_1) = 10.25, c'(1, a_2) = 8, c'(2, a_1) = 5.5 \text{ et } c'(2, a_2) = 11.5.$$

Une stratégie optimale peut être calculée par les outils de programmation dynamique comme l'algorithme d'itération de valeur. Ainsi, on obtient la politique déterministe  $\pi^*$  définie comme :

$$\pi^*(1) = \pi^*(2) = a_1$$

Par le théorème 3.3.1.1, on trouve

$$\begin{aligned} \tilde{\mathcal{C}}^*(\gamma) &= \tilde{\mathcal{C}}(\gamma, \pi^*) = (I - \gamma P(\pi^*))^{-1} \tilde{c}(\pi^*) \\ &= ([\frac{740}{11}, \frac{290}{11}, \frac{130}{11}], [60, 30, 10])' \end{aligned}$$

où  $V'$  est le vecteur transposé du vecteur  $V$ .



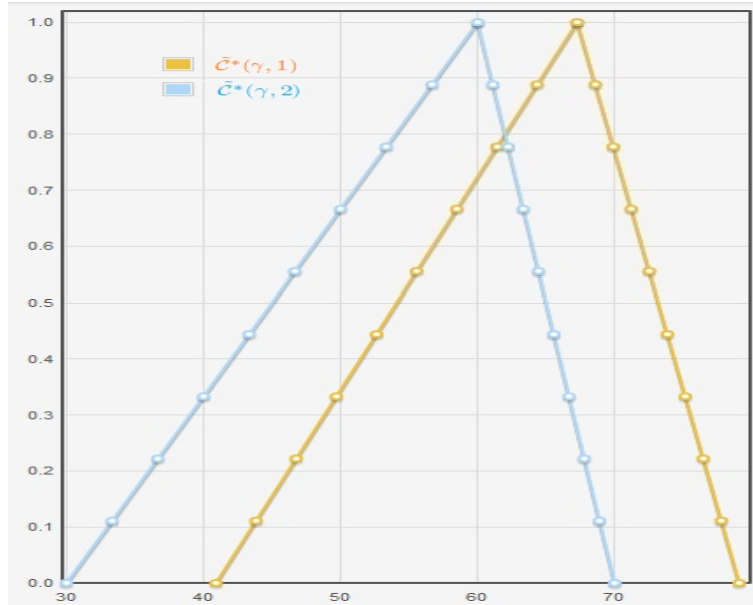


FIGURE 3.12 – Les fonctions d'appartenance du vecteur valeur optimal

**Exemple 3.3.2.2** *L'exemple suivant est inspiré de Piunovskiy [50] (p. 135). Nous supposons qu'un seul des coûts est imprécis et est donné par un nombre flou triangulaire.*

*Soit*

$$S = \{1, 2, 3, 4, 5\}; A = \{a_1, a_2\}; \gamma = 0.75$$

*et*

$p_{1a_12} = p_{1a_24} = 1, p_{2a_3} = p_{3a_2} = p_{4a_5} = p_{5a_4} = 1; \tilde{c}(1, a_1) = [10, \alpha, \beta]$  où  $\alpha = \beta = 0.5$ ,  
 $c(1, a_2) = 8, c(2, a) = 8, c(3, a) = 142/9, c(4, a) = c(5, a) = 12$  où  $a \in A$ .

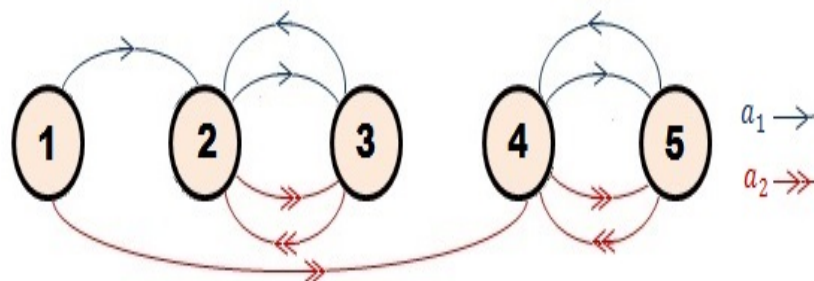


FIGURE 3.13 – Représentation graphique de l'exemple 3.3.2.2

Pour cela, nous considérons chaque coût ordinaire  $x$  ( $x \in \mathbb{R}$ ) comme un point flou représenté par le nombre flou triangulaire  $\tilde{x} = [x, 0, 0]$ .

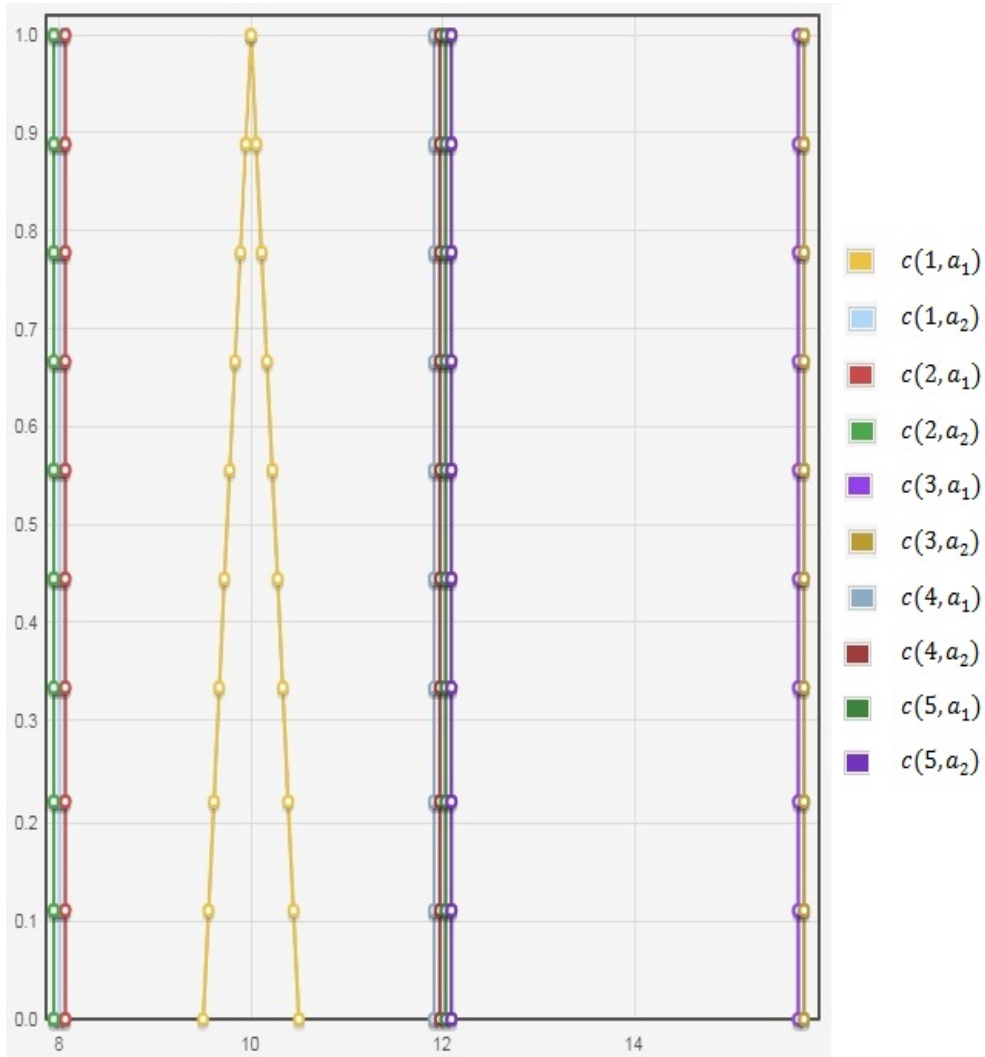


FIGURE 3.14 – Fonctions d'appartenance des coûts flous

En appliquant le même contexte utilisé dans l'exemple 3.3.2.1, nous obtenons une politique optimale déterministe  $\pi^*$  définie comme suit

$$\pi^*(1) = \pi^*(2) = \pi^*(3) = \pi^*(4) = \pi^*(5) = a_1$$

et le vecteur valeur optimale donné par

$$\tilde{C}^*(\gamma) = ([44, 0.5, 0.5], [45.333333, 0, 0], [49.777778, 0, 0], [48, 0, 0], [48, 0, 0])'$$

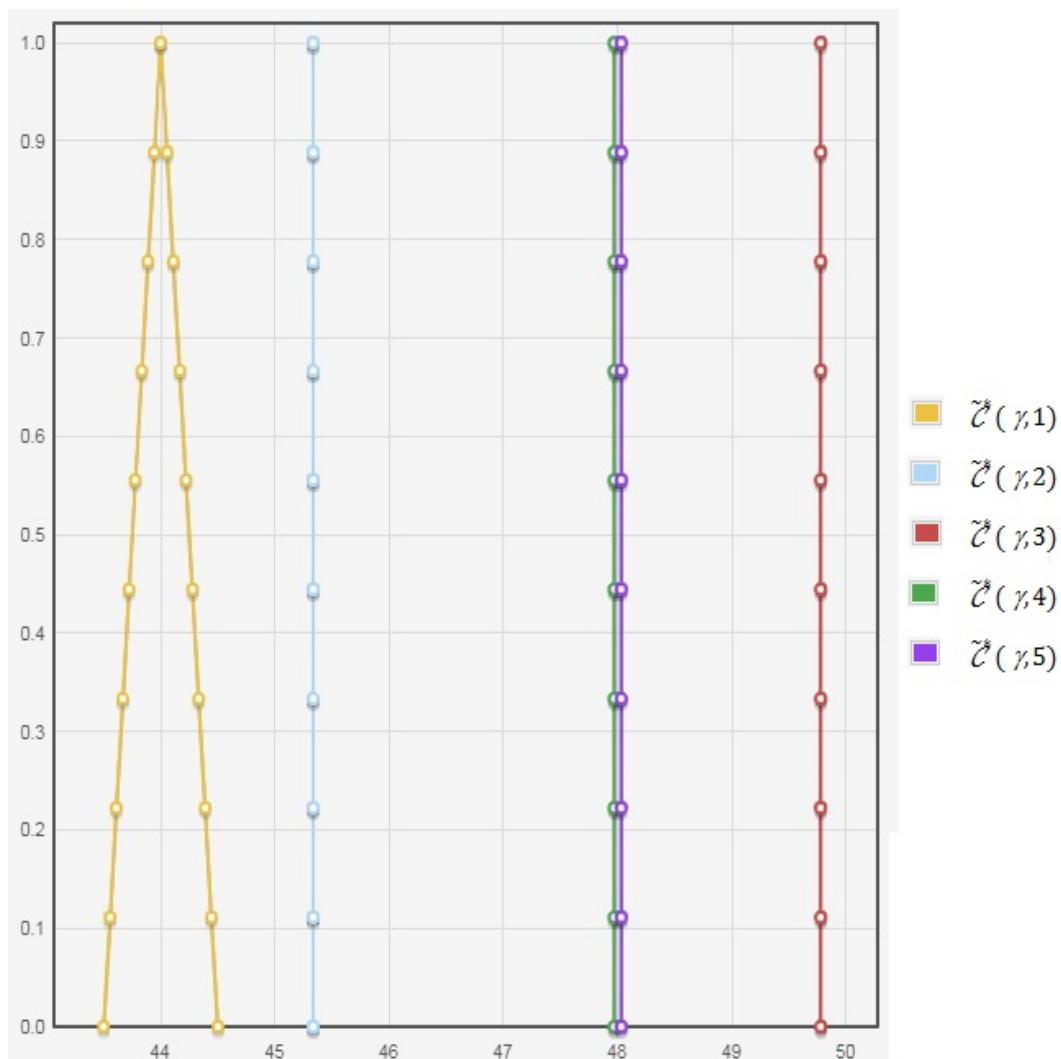


FIGURE 3.15 – Fonctions d'appartenance des composantes floues du vecteur valeur optimal.

**Remarque 3.3.2.1** *Pour  $\alpha = \beta = 0$ , nous obtenons les résultats numériques donnés par l'exemple original :*

$$\mathcal{C}^*(\gamma) = (44, 45.333333, 49.777778, 48, 48)'$$

### 3.4 Conclusion du chapitre

Par prise de décision dans un environnement flou, on entend un processus de décision dans lequel les buts et/ou les contraintes, mais pas nécessairement le système sous contrôle, sont de nature floue. Cela signifie que les buts et/ou les contraintes

constituent des classes d'alternatives dont les frontières ne sont pas clairement définies. La modélisation floue a été appliquée dans une grande variété de domaines tels que l'ingénierie et les sciences de gestion et les sciences sociales pour résoudre un certain nombre de problèmes de prise de décision qui impliquent l'incertitude des données. En particulier, les applications de cette technique de modélisation dans les problèmes de prise de décision ont une importance remarquable. Ces problèmes ont été abordés à l'aide de diverses théories telles que la théorie des probabilités, la théorie des ensembles flous, la théorie des ensembles approximatifs, la théorie des ensembles vagues, la théorie du raisonnement approximatif, etc. qui manquent de paramétrage des outils en raison desquels ils ne pouvaient pas être appliqués avec succès à de tels problèmes.

Dans ce chapitre, on a établi un nouveau modèle concernant le problème décisionnel floue avec des coûts flous actualisés. C'était l'occasion favorable d'investir l'outil de la programmation dynamique pour résoudre de tels problèmes flous.



avec

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix}; \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad \text{et} \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

Il existe deux classes fondamentales de méthodes pour résoudre le système (S). La première classe est représentée par des méthodes directes. Ils donnent théoriquement une solution exacte en un nombre fini (prévisible) d'étapes. Malheureusement, cela n'a pas à être vrai dans la pratique informatique en raison des erreurs d'arrondi : une erreur commise dans une étape se propage dans toutes les étapes suivantes. Les méthodes directes classiques sont discutées dans cette section. De plus, la résolution d'un système d'équations au moyen de décompositions matricielles, peut également être classée comme une méthode directe. La deuxième classe est appelée méthodes itératives, qui construisent une série d'approximations de solution qui (sous certaines hypothèses) converge vers la solution du système. Enfin, notez que certaines méthodes sont à la frontière entre les deux classes ; par exemple, les méthodes de gradient et le raffinement itératif.

De plus, les méthodes directes discutées dans cette section ne sont pas nécessairement optimales pour un système arbitraire. Il y a des exceptions principales. Premièrement, même s'il existe une solution unique, les méthodes numériques peuvent échouer à trouver la solution : si le nombre de variables inconnues  $n$  est grand, les erreurs d'arrondi peuvent s'accumuler et aboutir à une mauvaise solution. Il en va de même pour les systèmes avec une matrice de coefficients presque singulière. Une alternative est d'utiliser des méthodes itératives, qui sont moins sensibles à ces problèmes. Une autre approche consiste à utiliser les décompositions QR ou SVD, qui peuvent transformer certains problèmes presque singuliers en problèmes non singuliers. Deuxièmement, de très gros problèmes comprenant des centaines ou des milliers d'équations et de variables inconnues peuvent prendre beaucoup de temps à résoudre par des méthodes directes standard. D'autre part, leurs matrices de coefficients sont souvent clairsemées, c'est-à-dire que la plupart de leurs éléments sont des zéros. Des stratégies spéciales pour stocker et résoudre de tels problèmes ne cessent

d'augmenter ( voir O. Axelsson [67], D. Calvetti, L. Reichel [68] et Quarteroni, R. Sacco, F. Saleri [69]).

Dans cette section, on se limitera au cas où la matrice  $A$  est une matrice carrée inversible d'ordre  $n$  ( $n = m$ ).

## 4.2 Généralités les méthodes de résolution des systèmes linéaires

### 4.2.1 Les méthodes directes

Les systèmes linéaires sont utilisés dans de nombreux problèmes d'ingénierie et de science, ainsi que dans les applications des mathématiques aux sciences sociales et à l'étude quantitative des problèmes commerciaux et économiques. En particulier, les systèmes linéaires sont utilisés pour résoudre numériquement des équations différentielles.

Par le terme méthodes directes, nous entendons des méthodes qui donnent théoriquement la solution exacte du système en un nombre fini d'étapes telles que l'élimination gaussienne. En pratique, bien entendu, la solution obtenue sera contaminée par l'erreur d'arrondi qu'implique l'arithmétique utilisée. L'analyse de l'effet de cette erreur d'arrondi et la détermination des moyens de la maîtriser seront une composante majeure.

#### 1) La Méthode d'élimination de Gauss :

L'algorithme consiste alors à remplacer à chaque étape la matrice  $A$  par une matrice  $A^k$  dont les  $k^{\text{èmes}}$  premiers vecteurs colonnes correspondent au début d'une matrice triangulaire. À la  $(k + 1)^{\text{ème}}$  étape, on conserve les  $k$  premières lignes et les  $(k - 1)$  premières colonnes de  $A^k$ .

En pratique, si le pivot, c'est-à-dire l'élément  $a_{k,k}^k$  situé à la  $k$ -ième ligne et à la  $k$ -ième colonne, est petit ou nul, l'algorithme n'est plus valable. On emploie dans ce cas des permutations de lignes et de colonnes appelées stratégies de pivot.

---

**Algorithme 11 : Gauss Elimination Method**

---

```
1 for k = 1 to n - 1 do
2   for i = k + 1 to n do
3     for j = k + 1 to n do
4        $a_{i,j}^{(k+1)} = a_{i,j}^{(k)} - \frac{a_{i,k}^{(k)}}{a_{k,k}^{(k)}} a_{k,j}^{(k)} ; i = k + 1, \dots, n \text{ et } j = k + 1, \dots, n$ 
5      $a_{i,k}^{(k+1)} = 0 ; i = k + 1, \dots, n$ 
6      $b_i^{(k+1)} = b_i^{(k)} - \frac{a_{i,k}^{(k)}}{a_{k,k}^{(k)}} b_i^{(k)}$ 
```

---

**2) La Méthode Gauss-Jordan :**

C'est une variante de la méthode de Gauss : à la  $k^{\text{ème}}$  étape, on combine toutes les lignes (sauf la ligne  $k$ ) avec la ligne  $k$  (au lieu de ne le faire que pour les lignes d'indice supérieur à  $k$ ). On fait ainsi apparaitre des 0 sur toute la colonne sauf au niveau du pivot  $a_{k,k}^k$ .

---

**Algorithme 12 : Gauss-Jordan Elimination Method**

---

```
1 for k = 1 to n - 1 do
2   for j = k + 1 to n do
3     for i = 1 to n except k do
4        $a_{i,j}^{(k+1)} = a_{i,j}^{(k)} - \frac{a_{i,k}^{(k)}}{a_{k,k}^{(k)}} a_{k,j}^{(k)} ; i = k + 1, \dots, n \text{ et } j = k + 1, \dots, n$ 
5      $a_{i,k}^{(k+1)} = 0 ; i = k + 1, \dots, n$ 
6      $b_i^{(k+1)} = b_i^{(k)} - \frac{a_{i,k}^{(k)}}{a_{k,k}^{(k)}} b_i^{(k)}$ 
```

---

**3) La factorisation LU**

La décomposition  $LU$  consiste à écrire une matrice carrée  $A$  d'ordre  $n$  en produit de deux matrices carrées d'ordre  $n$   $L$  triangulaire inférieure ayant des 1 sur la



diagonale et  $U$  triangulaire supérieure.

$$L = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ l_{2,1} & 1 & 0 & \cdots & 0 \\ l_{3,1} & l_{3,1} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{n,1} & l_{n,2} & l_{n,3} & \cdots & 1 \end{bmatrix} \text{ et } U = \begin{bmatrix} u_{1,1} & u_{1,2} & u_{1,3} & \cdots & u_{1,n} \\ 0 & u_{2,2} & u_{2,3} & \cdots & u_{2,n} \\ 0 & 0 & u_{3,3} & \cdots & u_{3,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & u_{n,n} \end{bmatrix}$$

La méthode de Crout est fondée sur la factorisation  $LU$  qui affirme que pour une matrice carrée  $A = (a_{i,j})$  d'ordre  $n$  telle que les  $n$  sous-matrices

$$\Delta_k = \begin{bmatrix} a_{1,1} & \cdots & a_{1,k} \\ \vdots & & \vdots \\ a_{k,1} & \cdots & a_{k,k} \end{bmatrix}$$

soient inversibles, il existe un couple unique  $(L, U)$  de matrices vérifiant les propriétés précédentes telles que  $A = LU$ .

L'algorithme suivant consiste à calculer les matrices  $L$  et  $U$  :

---

**Algorithme 13 : LU Factorization Algorithm**

---

```

1 for  $k = 1$  to  $n$  do
2    $u_{1,k} = a_{1,k}$ 
3    $l_{k,1} = \frac{a_{k,1}}{u_{1,1}}$ 
4    $l_{k,k} = 1$ 
5 for  $i = 2$  to  $n$  do
6   for  $j = 1$  to  $i - 1$  do
7      $u_{i,j} = 0$ 
8   for  $j = i + 1$  to  $n$  do
9      $l_{i-1,j} = 0$ 
10 for  $i = 2$  to  $n - 1$  do
11   for  $j = i + 1$  to  $n$  do
12      $u_{i,j} = a_{i,j} - \sum_{k=1}^{i-1} l_{i,k} u_{k,j}$  and  $u_{j,i} = \left( a_{j,i} - \sum_{k=1}^{j-1} l_{j,k} u_{k,i} \right) / u_{i,i}$ 

```

---

#### 4) La factorisation de Cholesky

La factorisation de Cholesky consiste, pour une matrice symétrique définie positive  $A$ , à déterminer une matrice triangulaire inférieure  $L$  telle que :  $A = LL^T$ . Cette décomposition est unique si les coefficients diagonaux de  $L$  sont positifs.

---

**Algorithme 14 : Cholesky decomposition**

---

```
1 for  $k = 1$  to  $n$  do
2    $l_{k,k} = \sqrt{a_{k,k} - \sum_{j=1}^{k-1} l_{k,j}^2}$ 
3   for  $i = k + 1$  to  $n$  do
4      $l_{i,k} = (a_{i,k} - \sum_{j=1}^{k-1} l_{k,j} l_{i,j}) / l_{k,k}$ 
```

---

#### 4.2.2 Les méthodes itératives

En analyse numérique, une méthode itérative est une procédure mathématique qui utilise une valeur initiale pour générer une suite de solutions approximatives améliorées pour une classe de problèmes, dans laquelle la  $n^{\text{ème}}$  approximation dérive des précédentes. Une implémentation spécifique d'une méthode itérative, y compris les critères de terminaison, est un algorithme de la méthode itérative. Une méthode itérative est dite convergente si la suite correspondante converge pour des approximations initiales données. Une analyse de convergence mathématiquement rigoureuse d'une méthode itérative est généralement effectuée ; cependant, les méthodes itératives heuristiques sont également courantes.

Les méthodes itératives sont souvent utiles même pour des problèmes linéaires impliquant de nombreuses variables (parfois de l'ordre de millions), où les méthodes directes seraient prohibitives (et dans certains cas impossibles) même avec la meilleure puissance de calcul disponible.

## 1) La méthode de Jacobi :

En algèbre linéaire numérique, la méthode de Jacobi est un algorithme itératif pour déterminer les solutions d'un système d'équations linéaires à diagonale strictement dominant. Chaque élément diagonal est résolu et une valeur approximative est insérée. Le processus est ensuite itéré jusqu'à ce qu'il converge. Cet algorithme est une version allégée de la méthode de transformation de Jacobi de diagonalisation matricielle.

---

### Algorithme 15 : Jacobi Algorithm

---

```
1 Initialize  $X^{(0)} \in \mathbb{R}^n$ , specify  $\varepsilon > 0$ 
2  $k \leftarrow 0$ 
3 repeat
4   for  $i = 1$  to  $n$  do
5      $X_i^{(k+1)} = \left( b_i - \sum_{j \neq i} a_{i,j} X_j^{(k)} \right) / a_{i,i};$ 
6    $k \leftarrow k + 1$ 
   until  $\text{norm}(X^{(k+1)} - X^{(k)}) < \varepsilon;$ 
7 Return  $X^{(k+1)}$ .
```

---

## 2) La méthode de Gauss-Seidel :

En algèbre linéaire numérique, la méthode de Gauss-Seidel, également connue sous le nom de méthode de Liebmann ou méthode des déplacements successifs, est une méthode itérative utilisée pour résoudre un système d'équations linéaires. Elle porte le nom des mathématiciens allemands Carl Friedrich Gauss et Philipp Ludwig von Seidel, et est similaire à la méthode Jacobi. Bien qu'elle puisse être appliquée à des matrices avec des éléments non nuls sur les diagonales, la convergence n'est garantie que si la matrice est à diagonale strictement dominante.

---

**Algorithme 16 : Gauss-Seidel Algorithm**

---

```
1 Initialize  $X^{(0)} \in \mathbb{R}^n$ , specify  $\varepsilon > 0$ 
2  $k \leftarrow 0$ 
3 repeat
4   for  $i = 1$  to  $n$  do
5      $X_i^{(k+1)} = \left( b_i - \sum_{j < i} a_{i,j} X_j^{(k+1)} - \sum_{j > i} a_{i,j} X_j^{(k)} \right) / a_{i,i};$ 
6    $k \leftarrow k + 1$ 
  until  $norm(X^{(k+1)} - X^{(k)}) < \varepsilon;$ 
7 Return  $X^{(k+1)}$ .
```

---

**3) La méthode de Richardson :**

La méthode d'itération de Richardson modifiée est une méthode itérative de résolution d'un système d'équations linéaires. Elle est similaire à la méthode de Jacobi et Gauss-Seidel.

La méthode stationnaire du premier ordre de Richardson est une méthode itérative avec un paramètre local  $\alpha$  pour accélérer le processus d'itération.

---

**Algorithme 17 : Richardson Algorithm**

---

```
1 Initialize  $X^{(0)} \in \mathbb{R}^n$ , specify  $\varepsilon > 0$ 
2  $k \leftarrow 0$ 
3 repeat
4   for  $i = 1$  to  $n$  do
5      $r_i^{(k)} := b_i - [AX^{(k)}]_i;$ 
6      $X_i^{(k+1)} = X_i^{(k)} + \alpha r_i^{(k)};$ 
7    $k \leftarrow k + 1$ 
  until  $norm(r^{(k)}) < \varepsilon;$ 
8 Return  $X^{(k+1)}$ .
```

---

Ici  $r^{(k)}$  est le vecteur résiduel de l'itération courante.

**Théorème 4.2.2.1** *Le schéma numérique de Richardson est convergente si et seule-*

ment si  $\frac{2\text{Re}(\lambda_i)}{\alpha|\lambda_i|^2} > 1$  pour tout  $i = 1, \dots, n$ .

où  $\lambda_i \in \mathbb{C}$  sont les valeurs propres de la matrice  $A$ .

**Théorème 4.2.2.2** *Supposons que la matrice  $A$  possède des valeurs propres toutes réelles et positives telles que  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n > 0$ . Alors, la méthode stationnaire de Richardson est convergente si et seulement si  $0 < \alpha < \frac{2}{\lambda_1}$ . De plus, la valeur optimale de  $\alpha$  est  $\alpha_{opt} = \frac{2}{\lambda_1 + \lambda_n}$ .*

**Corollaire 4.2.2.1** *Soit  $A$  une matrice symétrique définie positive. Alors la méthode stationnaire de Richardson est convergente.*

**Remarque 4.2.2.1** *Malgré sa convergence, la méthode stationnaire de Richardson a l'inconvénient d'être numériquement instable.*

### 4.3 Résolution numérique des systèmes d'équations linéaires floues

Les systèmes d'équations linéaires trouvent de nombreuses applications dans le monde réel dans différents domaines scientifiques. Dans ce contexte, plusieurs méthodes itératives sont développées en raison de leur mise en œuvre la plus simple et de leur moindre coût de complexité de calcul. Le terme d'ensemble flou est apparu pour la première fois en 1965 lorsque le professeur Lotfi A. Zadeh de l'Université de Berkeley, aux États-Unis, a publié un article intitulé Ensembles flous. Il a depuis réalisé de nombreuses avancées théoriques majeures dans le domaine et a été rapidement accompagné par de nombreux chercheurs développant des travaux théoriques.

Dans cette section, on est en face d'un système linéaire flou avec un nombre flou arbitraire sous forme paramétrique et avec une matrice à coefficients réels. Il existe énormément de méthodes itératives numériques pour la résolution des systèmes linéaires flous telles que : la méthode itérative de Gauss-Seidel (G-S), Jacobi (J) et la méthode itérative de relaxation successive (SOR). Ainsi, une méthode itérative de Richardson du premier ordre sera présentée pour les systèmes linéaires flous non singuliers.



avec  $B_{ij} = \max_{i,j=1,\dots,n} (0, a_{ij})$  et  $C = B - A$

Il est facile de vérifier que la matrice carrée  $B$  contient les éléments positifs de  $A$  et la matrice carrée  $C$  contient la valeur absolue des éléments négatifs de  $A$ .

**Exemple 4.3.0.1** *Considérons un circuit résistif simple avec un courant flou et une source floue.*

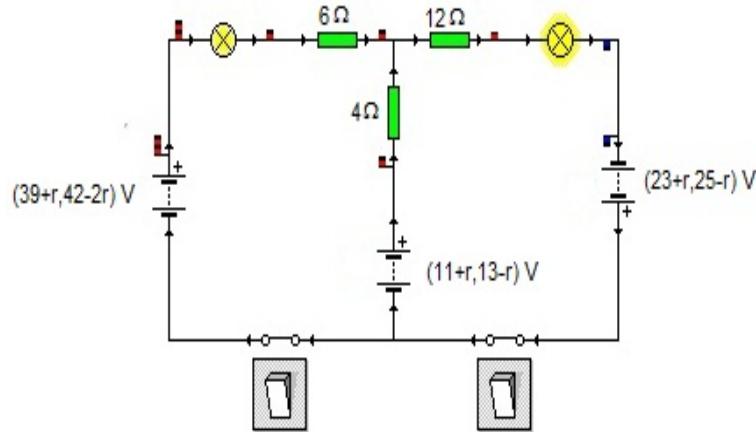


FIGURE 4.1 – Circuit flou

Ce circuit fournit un système d'équations linéaires floues donné par :

$$\begin{cases} 10\tilde{I}_1 - 4\tilde{I}_2 = (39 + r, 42 - 2r) - (11 + r, 13 - r) \\ -4\tilde{I}_1 + 16\tilde{I}_2 = (11 + r, 13 - r) - (23 + r, 25 - r) \end{cases}$$

Par simplification, on obtient

$$\begin{cases} 10\tilde{I}_1 - 4\tilde{I}_2 = (26 + 2r, 31 - 3r) \\ -4\tilde{I}_1 + 16\tilde{I}_2 = (34 + 2r, 38 - 2r) \end{cases}$$

On déduit que :

$$S = \begin{pmatrix} B & C \\ C & B \end{pmatrix} \text{ avec } B = \begin{pmatrix} 10 & 0 \\ 0 & 16 \end{pmatrix} \text{ et } C = \begin{pmatrix} 0 & 4 \\ 4 & 0 \end{pmatrix}$$

En résolvant le système (4.1), on obtient la solution exacte sous la forme paramétrique suivante :

$$\tilde{I}_1 = \left(\frac{71}{18} + \frac{1}{6}r, \frac{79}{18} - \frac{5}{18}r\right) \text{ et } \tilde{I}_2 = \left(\frac{29}{9} + \frac{1}{18}r, \frac{121}{36} - \frac{1}{12}r\right)$$

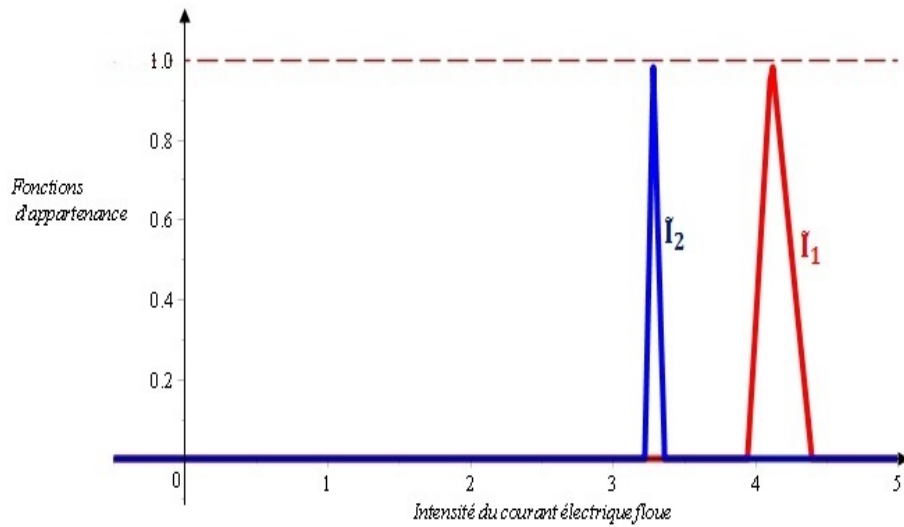


FIGURE 4.2 – La solution exacte

Dans notre cas, la matrice  $S$  a toutes ses valeurs propres 18, 8, 18 et 8 qui sont strictement positives. Ensuite, la valeur optimale du paramètre minimiseur  $\alpha$  est :

$$\alpha_{opt} = \frac{2}{\lambda_{max} + \lambda_{min}} = \frac{2}{18 + 8} = \frac{1}{13}$$

En utilisant la récursivité successive de Richardson, nous obtenons la solution approchée donnée par :

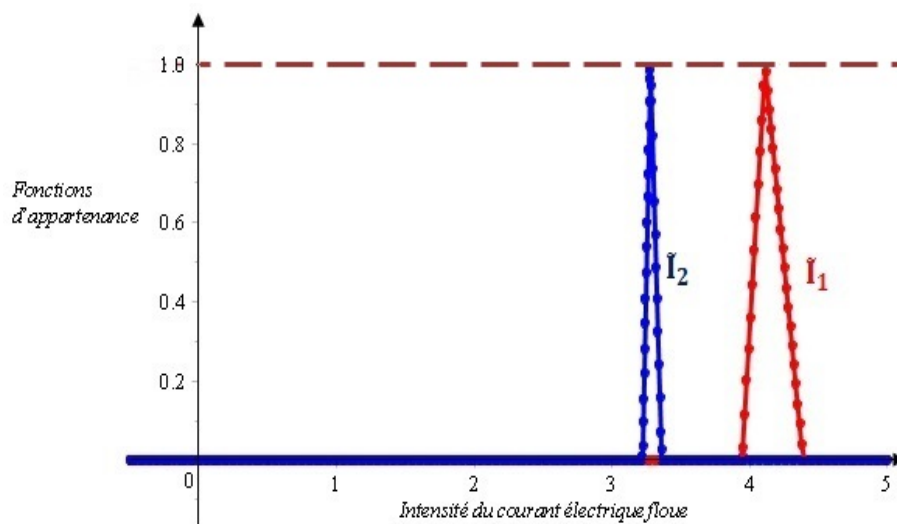


FIGURE 4.3 – La solution approchée



## 4.4 Conclusion du chapitre

Les applications réelles issues de divers domaines de l'ingénierie et des sciences comme l'électricité, le civil, l'économie, l'alimentation, etc. peuvent être modélisées à l'aide d'un système d'équations linéaires. Dans de tels modèles, il peut arriver que les valeurs des paramètres ne soient pas connues ou qu'elles ne puissent pas être énoncées avec précision, seule leur estimation due à des données expérimentales ou à des connaissances d'experts est disponible. Dans une telle situation, il est commode de représenter ces paramètres par des nombres flous.

Ce chapitre a visé l'exploitation de la méthode itérative de Richardson du premier ordre pour résoudre les systèmes d'équations linéaires flous. Cette technique a prouvé son efficacité dans plusieurs champs d'application.

## CONCLUSION GÉNÉRALE

*Le thème central de cette thèse est d'étudier les problèmes décisionnels de Markov avec des coûts actualisés. C'est un cadre mathématique qui nécessite la détermination de solutions optimales correspondant à une fonction d'utilité à minimiser. Étant donné que la programmation dynamique fournit des méthodes itératives inspirées de l'équation d'optimalité de Bellman pour résoudre de tels problèmes, nous avons présenté dans ce travail le célèbre algorithme d'itération de valeurs et ses principales variantes telles que les méthodes de Jacobi, Gauss-Seidel et Sur-relaxation. De plus, la programmation linéaire donne également une solution optimale aux problèmes ci-dessus. Dans ce contexte, nous avons donné un aperçu de quelques méthodes qui sont implantées dans l'outil de programmation linéaire.*

*Parfois, le calcul de telles solutions a lieu en présence d'opérations arithmétiques énormes et de l'utilisation d'une grande mémoire en raison de la grande taille des données du problème. Après avoir détaillé le cadre théorique de notre contribution, nous avons établi de nouveaux tests pour identifier les actions non optimales lors du calcul. À cet égard, nous avons prouvé l'efficacité de nos résultats par des exemples illustratifs.*

*Le cadre de la théorie des sous-ensembles flous a été utilisé dans le cas des environnements flous pour représenter des paramètres imprécis dans les PDMs. Ceci a lieu en adoptant le critère actualisé et en évaluant le degré de correspondance des données à l'aide de l'approche fonctionnelle basée sur la fonction d'appartenance. Ainsi, nous avons établi un modèle théorique flou pour résoudre de telles situations floues.*

*Enfin, nous avons appliqué ce cadre de la théorie floue pour résoudre des systèmes d'équations linéaires floues. Un exemple illustratif de circuit électronique flou a été considéré pour montrer la pertinence de notre approche.*

*De nombreuses perspectives s'ouvrent à la suite de cette contribution. Nous donnons les plus essentielles et les plus importantes d'entre eux :*

- Des améliorations raisonnables peuvent être apportées aux algorithmes de programmation dynamique pour réduire la complexité de calcul en temps et en mémoire.*
- Amélioration de la fonction Q-learning dans le mouvement d'un robot.*
- Enrichir le cadre de la théorie floue en permettant de résoudre les problèmes régis par la théorie floue.*
- Etude du mouvement d'un Robot avec incertitude de certains paramètres.*

## CONCLUSION AND PERSPECTIVES

*The central theme of this thesis is to study Markov decision problems with discounted costs. It is a mathematical framework that requires the determination of optimal solutions corresponding to a utility function to be minimized. Since the dynamic programming provides iterative methods inspired from the Bellman optimality equation for solving such problems, we have presented in this work the well-known Value Iteration Algorithm and its main variants such as Jacobi, Gauss-Seidel and Successive Over-relaxation methods. In addition, linear programming also gives an optimal solution to the above problems. In this context, we have given an overview on some methods which are established in the linear programming tool. Sometimes the computation of such solutions takes place in the presence of huge arithmetic operations and the use of large memory due to the large data size of the problem. After having detailed the theoretical framework of our contribution, we established new tests to identify non-optimal actions during the calculation. In this regard, we have proven the effectiveness of our results by experiments related to the real life. The framework of fuzzy subset theory has been used in case to represent imprecise parameters in MDPs by adopting the discounted criterion by evaluating the degree of correspondence of the data using the functional approach based on the membership function. Thus, we have established a fuzzy theoretical model to solve such fuzzy situations. Finally, we applied this framework of fuzzy theory to solve systems of fuzzy linear equations. Therefore, the demonstration of this work required giving an illustrative example of a fuzzy electronic circuit.*

*There are numerous perspectives for our work. We give the most essential and important of them :*

- Reasonable improvements can be made to dynamic programming algorithms for reducing computational complexity in time and memory ;*
- Improvement of the Q-learning function in the Robot motion ;*
- Enrich the framework of the fuzzy theory by allowing to solve the problems governed by fuzziness ;*
- Study of the Robot motion with uncertainty of some parameters.*

## BIBLIOGRAPHIE

- [1] Faure R. : La programmation linéaire appliquée. *Presses Universitaires de France*, Paris (1979).
- [2] Cohen V. : La recherche opérationnelle. *Presses Universitaires de France*, Paris (1995).
- [3] Conforti M., Cornuéjols G. & Zambelli G. : Integer Programming. *Springer International Publishing*, Switzerland (2014).
- [4] Schrijver A. : Theory of linear and integer programming. *John Wiley & Sons Ltd.*, England (1986).
- [5] Sacle A. : La théorie des graphes. *Presses Universitaires de France*, Paris (1974).
- [6] Diestel R. : Graph Theory. *Springer Nature*, Berlin (2017).
- [7] Jungnickel D. : Graphs, Networks and Algorithms. *Springer-Verlag*, Berlin Heidelberg (2005).
- [8] Caumel Y. : Probabilités et processus stochastiques. *Springer-Verlag France*, Paris (2011).
- [9] Graham C. : Chaînes de Markov. *Dunod*, Paris (2008).
- [10] Gallager R. G. : Stochastic Processes. *Cambridge University Press*, New York (2013).
- [11] Rosenthal J S. : A First Look at Rigorous Probability Theory. *World Scientific Publishing Co. Pte. Ltd.*, London (2006).

- [12] Ash R. B. : Basic Probability Theory. *Dover Publications, Inc. Mineola*, New York (2008).
- [13] Kannan D. : An introduction to Stochastic Processes. *Elsevier North Holland Inc.*, New York (1979).
- [14] Schuss Z. : Theory and Applications of Stochastic Processes. *Springer Science+Business Media, LLC*, London (2010).
- [15] MacQueen J. B. : A modified dynamic programming method for Markovian decision problems. *J. Math. Anal. Appl.*, pp. 38-43 (1965). doi:10.1016/0022-247X(66)90060-6
- [16] MacQueen J. B. : A test for suboptimal actions in Markovian decision problems. *Operations Research*, pp. 559-561 (1967). doi:10.1287/opre.15.3.559
- [17] Porteus, E. L. : Some bounds for discounted sequential decision processes. *Management Science*, pp. 7-11 (1971). doi:10.1287/mnsc.18.1.7
- [18] Grinold, R. C. : Elimination of suboptimal actions in Markov decision problems. *Operations Research*, pp. 848-851 (1973). doi:10.1287/opre.21.3.848
- [19] Puterman, M. L., Shin, M. C. : Modified policy iteration algorithms for discounted Markov decision problems. *Management Science*, pp. 1127-1137 (1978). doi:10.1287/mnsc.24.11.1127
- [20] White, D. J. : The determination of approximately optimal policies in Markov decision processes by the use of bounds. *Journal of the Operational Research Society*, pp. 253-259 (1982). doi:10.1057/jors.1982.51
- [21] Sladký, K. : Identification of optimal policies in Markov decision processes. *Kybernetika*, pp. 558-570 (2010). MSC: 60J10, 90C40, 93E20 | MR2676091 | Zbl1195.93148
- [22] Semmouri, A., and Jourhmane, M. : Markov Decision Processes with Discounted Cost : The action elimination procedures. In : ICCSRE 2nd International Conference of Computer Science and Renewable Energies, pp. 1-6. *IEEE Press*, Agadir, Morocco (2019) doi:10.1109/ICCSRE.2019.8807578
- [23] Semmouri, A., Jourhmane, M. : Markov Decision Processes with Discounted Costs over a Finite Horizon : Action Elimination. In : International Confe-

- rence on Artificial Intelligence & Industrial Applications, pp. 199-213. *Springer, Cham.*, Meknes, Morocco (2020) doi:10.1007/978-3-030-51186-9\_14
- [24] Semmouri, A., Jourhmane, M., Elbaghazaoui, B.E. : Markov Decision Processes with Discounted Costs : New Test of Non-optimal Actions. *Journal of Advanced Research in Dynamical and Control Systems* (05-SPECIAL ISSUE), pp. 608-616 (2020). doi:10.5373/JARDCS/V12SP5/20201796
- [25] Archibald T. W., Possani, E. : Investment and operational decisions for start-up companies : a game theory and Markov decision process approach. *Annals of Operations Research*, pp. 317-330 (2021) doi:org/10.1007/s10479-019-03426-5
- [26] Bahri O., Talbi E. G. : Robustness-based approach for fuzzy multi-objective problems. *Annals of Operations Research*, pp. 707-733 (2021). doi.org/10.1007/s10479-020-03567-y
- [27] Balbus L., Jaśkiewicz A., Nowak A. S. : Markov perfect equilibria in a dynamic decision model with quasi-hyperbolic discounting. *Annals of Operations Research*, pp. 573-591 (2020). doi:org/10.1007/s10479-018-2778-2
- [28] Bellman R.E., Zadeh L.A. : Decision making in a fuzzy environment. *Management Science*, 17, pp. 141-164 (1970). doi:org/10.1287/mnsc.17.4.B141
- [29] Bertsekas D.P., Shreve S.E. : Stochastic Optimal Control. *Academic Press*, New York (1978).
- [30] Bellman, R. E. : Dynamic Programming. *University Press*, Princeton (1957).
- [31] White, D. : Markov decision processes. *Jhon Wiley & Sons.*, England (1993).
- [32] Bhulai S., Blok H., Spieksma, F. M. : K competing queues with customer abandonment : optimality of a generalised  $c\mu$ -rule by the Smoothed Rate Truncation method. *Annals of Operations Research*, pp. 1-30 (2019). doi:org/10.1007/s10479-019-03131-3
- [33] Canbolat P.G., Rothblum U. G. : (Approximate) iterated successive approximations algorithm for sequential decision processes. *Annals of Operations Research*, 208, pp. 309-320 (2013). doi:org/10.1007/s10479-012-1073-x



- [34] Chen K. S., Chang T. C. : Fuzzy testing model for the lifetime performance of products under consideration with exponential distribution. *Annals of Operations Research*, pp. 1-12 (2020). doi.org/10.1007/s10479-020-03578-9
- [35] Derman C. : Finite State Markovian Decision Processes. *Academic Press*, New York (1970).
- [36] Diamond P., Kloeden P. : Metric Spaces of Fuzzy Sets, Theory and Applications. *World Scientific*, Singapore (1994).
- [37] Dubois D., Prade H. : Ranking of fuzzy numbers in the setting of possibility theory. *Information Sciences*, 30(3), pp. 183-224 (1983). doi:org/10.1016/0020-0255(83)90025-7
- [38] Dubois D., Prade H. : Fundamentals of Fuzzy sets. *Kluwer Academic Publishers*, Boston (2000).
- [39] Semmouri, A., Jourhmane, M., Belhallaj, Z. : Discounted Markov decision processes with fuzzy costs. *Annals of Operations Research* 295, pp. 769-786 (2020). doi:10.1007/s10479-020-03783-6
- [40] Howard, R. A. : Dynamic programming and markov processes. *John Wiley & Sons, Inc.*, New York . London (1960).
- [41] Feinberg, E.A., Shwartz, A. : Handbook of Markov Decision Processes. methods and applications, *Kluwer*, New York (2002).
- [42] James J. Buckley : Fuzzy Probabilites. *Springer-Verlag*, Berlin Heidelberg (2005).
- [43] Kaufman A., Gupta M. M. : Introduction to Fuzzy Arithmetic, Theory and applications. *Van Nostrand*, New York (1985).
- [44] Klir G., Yuan B. : Fuzzy Sets and Fuzzy Logic, Theory and Applications. *Prentice Hall, PTR*, New Jersey (1995).
- [45] Kurano M., Nakagami J., Yoshida Y. : Markov Decision Processes with Fuzzy Rewards. *Journal of Nonlinear Analysis and Convex Analysis*, pp. 105-116 (2003).
- [46] Liagkouras K., Metaxiotis K. : Improving the performance of evolutionary algorithms : a new approach utilizing information from the evolutionary process

- and its application to the fuzzy portfolio optimization. *Annals of Operations Research*, pp. 119-137 (2019). doi:10.1007/s10479-018-2876-1
- [47] Liu S. T., Lee Y. C. : Fuzzy measures for fuzzy cross efficiency in data envelopment analysis. *Annals of Operations Research*, pp. 369-398 (2021). doi:org/10.1007/s10479-019-03281-4
- [48] Mahdavi-Amiri N., Nasseri S.H. : Duality in fuzzy number linear programming by use of a certain linear ranking function. *Applied Mathematics and Computation*, pp. 206-216 (2006). doi:org/10.1016/j.amc.2005.11.161
- [49] Mohammed A. : Towards a sustainable assessment of suppliers : an integrated fuzzy TOPSIS-possibilistic multi-objective approach. *Annals of Operations Research*, pp. 1-30 (2020). doi.org/10.1007/s10479-019-03167-5
- [50] Piunovskiy A. B. : Examples in Markov decision processes (Vol. 2). *World Scientific*, London (2013).
- [51] Powell W. B. : Perspectives of approximate dynamic programming. *Annals of Operations Research*, pp. 319-356 (2016). doi:org/10.1007/s10479-012-1077-6
- [52] Puri M. L., Ralesca D. A. : Fuzzy random variable. *J. Math. Anal. Appl.*, pp. 402-422 (1986).
- [53] Puterman M.L. : *Markov decision processes : discrete stochastic dynamic programming*. New York, John Wiley & Sons (2014).
- [54] Roy J., Pamučar D., Kar S. : Evaluation and selection of third party logistics provider under sustainability perspectives : an interval valued fuzzy-rough approach. *Annals of Operations Research*, pp. 669-714 (2020). doi:org/10.1007/s10479-019-03501-x
- [55] Salari N., Makis V. : Joint maintenance and just-in-time spare parts provisioning policy for a multi-unit production system. *Annals of Operations Research*, pp. 351-377 (2020). doi:org/10.1007/s10479-019-03371-3
- [56] Valladão D., Silva T., Poggi M. : Time-consistent risk-constrained dynamic portfolio optimization with transactional costs and time-dependent re-

- turns. *Annals of Operations Research*, pp. 379-405 (2019). doi:org/10.1007/s10479-018-2991-z
- [57] Yager R.R. : A procedure for ordering fuzzy subsets of the unit interval. *Information Sciences*, pp. 143-161 (1981). doi:org/10.1016/0020-0255(81)90017-7
- [58] Zadeh L.A. : Fuzzy sets. *Information and Control*, pp. 338-353 (1965).
- [59] KAUFMAN A., GUPTA M. M. : Introduction to Fuzzy Arithmetic, Theory and applications. *Van Nostrand Reinhold Co. Inc.*, Workingham, Berkshire (1984).
- [60] Luca Amodei, Jean-Pierre Dedieu. Analyse numérique matricielle. Dunod, Paris (2008).
- [61] Demailly J. P. : Analyse numérique et équations différentielles. *EDP Sciences*, Grenoble (2006).
- [62] P. G. Ciarlet. Introduction à l'Analyse Numérique Matricielle et à l'Optimisation. MASSON, Paris (1982).
- [63] Friedman M., Kandel A. : Fuzzy linear systems. *Fuzzy Sets and Systems*, pp. 201-209 (1998). doi:org/10.1016/S0165-0114(96)00270-9
- [64] Allahviranloo T. : Numerical methods for fuzzy system of linear equations. *Appl Math Comput*, pp. 493-502 (2004). doi:org/10.1016/S0096-3003(03)00793-8
- [65] Allahviranloo T. : Successive over relaxation iterative method for fuzzy system of linear equations. *Appl Math Comput*, pp. 189-196 (2005). doi:org/10.1016/j.amc.2003.12.085
- [66] George J. K., Yuan B. : Fuzzy Sets and Fuzzy Logic, Theory and Applications. *Prentice Hall PTR*. (1995).
- [67] Axelsson O. : Iterative Solution Methods. *Cambridge University Press* (1996).
- [68] Calvetti D., Reichel L. : A hybrid iterative method for symmetric positive definite linear systems. *Numerical Algorithms*, pp. 79-98 (1996).
- [69] Quarteroni A., Sacco R., Saleri F. : Numerical Mathematics. *Springer-Verlag New York, Inc* (2000).

- [70] Saad Y., Van Der Vorst H. A. : Iterative solution of linear systems in the 20th century. *Comput. Appl. Math.*, pp. 1-33 (2000). doi:org/10.1016/S0377-0427(00)00412-X
- [71] Rahgooy T., Yazdi H. S. & Monsefi R. : Fuzzy Complex System of Linear Equations Applied to Circuit Analysis. *International Journal of Computer and Electrical Engineering*, pp. 1793-8163 (2009).
- [72] Denardo, E. V. : Contraction mappings in the theory underlying dynamic programming. **Siam Review**, pp. 165-177 (1967).
- [73] Reetz, D. : Solution of a Markovian decision problem by successive overrelaxation. **Zeitschrift für Operations Research**, pp. 29-32 (1973).
- [74] Semmouri, A., Jourhmane, M., Elbaghazaoui, B. E. : Markov Decision Processes with Discounted Rewards : Improved Successive Over-Relaxation Method. In The International Conference on Information, Communication & Cybersecurity. **Springer, Cham.**, pp. 229-238 (2021). doi:org/10.1007/978-3-030-91738-8\_22