



**HAL**  
open science

# Algorithmes pour la décision séquentielle dans l'incertain

Gildas Jeantet

► **To cite this version:**

Gildas Jeantet. Algorithmes pour la décision séquentielle dans l'incertain : optimisation de l'utilité espérée dépendant du rang et du critère de Hurwicz. Intelligence artificielle [cs.AI]. Université Pierre et Marie Curie (Paris VI), 2010. Français. NNT : . tel-03100305

**HAL Id: tel-03100305**

**<https://hal.science/tel-03100305>**

Submitted on 8 Jan 2021

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

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

UNIVERSITE PIERRE ET MARIE CURIE (PARIS VI)  
UFR INGÉNIERIE



N° attribué par la bibliothèque



## THESE

Pour l'obtention du titre de  
DOCTEUR EN INFORMATIQUE  
*spécialité: aide à la décision*

soutenue le 27 septembre 2010

***Algorithmes pour la décision séquentielle dans l'incertain :  
optimisation de l'utilité espérée dépendant du rang et du  
critère de Hurwicz***

*Candidat:* Gildas JEANTET

### *JURY*

Directeur de thèse: Patrice PERNY  
Professeur à l'Université Pierre et Marie Curie (Paris VI)

Encadrant de thèse: Olivier SPANJAARD  
Maître de Conférence à l'Université Pierre et Marie Curie (Paris VI)

Rapporteurs: Denis BOUYSSOU  
Directeur de recherche CNRS à l'Université Paris-Dauphine (Paris IX)  
Hélène FARGIER  
Directeur de recherche CNRS à l'Université Paul Sabatier (Toulouse III)

Examineurs: Philippe CHRETIENNE  
Professeur à l'Université Pierre et Marie Curie (Paris VI)  
Michel GRABISCH  
Professeur à l'Université Panthéon-Sorbonne (Paris I)  
Régis SABBADIN  
Chargé de recherche INRA (Toulouse)



*L'université n'entend donner aucune approbation ni improbation aux opinions émises dans les thèses : ces opinions doivent être considérées comme propres à leurs auteurs.*



« Une thèse sans citations, c'est un peu  
comme se coucher avec une paire d'as avant  
le flop, c'est triste. ».

*Paul Weng*



# Remerciements

Tout d'abord je tiens à remercier Olivier Spanjaard qui m'a encadré durant ces années de doctorat. Je ne sais pas si son honnêteté scientifique est plus à applaudir que sa capacité à m'avoir supporté durant ces dernières années toujours avec le sourire, mais ses grandes qualités de chercheur m'ont donné le goût des métiers de la recherche. Ses qualités pédagogiques ainsi que sa patience ont grandement facilité ma compréhension des mécanismes complexes mis en jeu dans le domaine de la théorie de la décision et son astuce m'a ouvert de nombreuses pistes qui m'ont permis de régénérer mes réflexions sur certains problèmes de recherche qui me semblaient être une impasse.

Je tiens également à remercier Patrice Perny d'avoir accepté de prendre le Ô combien très grand risque de devenir mon directeur de thèse et dont les nombreux conseils fréquents toujours pertinents m'ont souvent permis de rester dans un axe cohérent de recherche.

Enfin je tiens à saluer les grandes qualités humaines de ces deux hommes qui rendent agréable la vie quotidienne au laboratoire.

Je tiens à remercier chaleureusement Denis Bouyssou et Hélène Fargier qui ont accepté d'être les rapporteurs de cette thèse et pour l'intérêt qu'ils ont porté à ce travail de manière générale. Je tiens à remercier également Régis Sabbadin et Michel Grabisch qui ont accepté de venir examiner cette thèse, j'espère qu'ils auront apprécié les travaux présentés dans cette thèse. Enfin je remercie Philippe Chretienne qui a accepté d'assumer le rôle de président du jury et qui m'a enseigné, lors de mon master, les bases de la théorie de la complexité qui m'ont été très utiles dans ces travaux de thèse.

Je tiens à adresser un remerciement tout particulier à feu Jean-Yves Jaffray qui m'a donné le goût aux problématiques de la décision dans l'incertain. Chacun de nos entretiens a toujours été très enrichissant pour moi et il a toujours su répondre présent lorsque j'avais besoin d'aide.

Cette thèse n'aurait jamais vu le jour sans les brillantes équipes Decision et RO que j'ai côtoyées tous les jours, et mes sincères remerciements s'adressent particulièrement à Safia, Christophe, Pierre-Henri, Francis, Fanny, Paul (dont le nom est aujourd'hui rendu célèbre par la citation de la page précédente) et Pierre qui ont apporté un cadre social solide et dont les discussions autour de la machine à café ont souvent été enrichissantes. Je remercie évidemment Jean-Philippe, Lionel, Charles, Jean-Mathieu et Thomas qui m'ont grandement aidé dans les moments difficiles à coup de Chimay, je remercie d'ailleurs au passage le patron du Troll qui n'a jamais monté les prix de sa Chimay. Un remerciement

particulier à Lucie (qui a été victime d'une private running joke à son insu) pour m'avoir accueilli au laboratoire et pour ses nombreux conseils. Je tiens également à remercier Arièle, Julien, Yasmina, Nina, Anna, Alexandre, Thomas Beline, Nicolas Jouve et Stefanovitch, Hakim, Shirley et Zach, l'ensemble des doctorant qui a bercé mes années de thèses et que j'ai eu le plaisir à côtoyer tous les jours. je remercie égaleemnt Thierry Lanfroy qui a toujours su me faciliter les tâches administratives et qui ma toujours reçu dans la bonne humeur.

Enfin, je ne serais surement jamais arrivé au bout de cette thèse sans le soutien de ma famille, et c'est pourquoi je remercie du fond du coeur ma mère et Jean, ainsi que mon père et Claire et également ma mamie dont je regrette sincèrement qu'elle ne puisse venir à la soutenance.

Puis il y a mes amis, peu nombreux mais sans qui je ne serai pas ce que je suis aujourd'hui. Merci MilOo, merci Dadou, merci Raphaël, merci les Joris, merci Jean et merci Antoine. Merci également à Robin avec qui, chaque jour, on prépare la venue du jour Z.

Et enfin merci à Lucie, celle qui réside dans mon coeur et qui chaque jour doit me supporter un peu plus.

Je remercie enfin Don Mc Lean, Moriarty, The Cat Empire, Mud Flow, R.E.M, Noir Désir, Aaron, Nina Simone, Pink Floyd, Les Têtes Raides, Dolly Parton, The Clash, Thomas Fersen, La Rue Kétanou, Les V.R.P, Les Nonnes Troppo, Ijahman, Marie Kiss la Joue, Les Epis Noirs, Pigalle, Casse-Pipe, Les Princes Chameaux, Jeff Buckley, Scorpions, Trust, Queen, Malicorne, Nirvana, Java, L'attirail, Tarmac, Louise Attaque, La Tordue, Django Reinardt, Les Sales Majestés, Parabellum, Bonobo, Giora Feidman, Chet Baker, Keith Jarret, Mes Souliers sont Rouges et sûrement plein d'autres que j'oublie ici pour avoir bercé la rédaction de chacun de ces chapitres de thèse.



# Table des matières

<b>Introduction</b>	<b>1</b>
<b>1 Décision dans l'incertain</b>	<b>7</b>
1.1 Décision dans le risque . . . . .	11
1.1.1 Le développement de l'espérance d'utilité . . . . .	11
1.1.2 Mesure de l'attitude face au risque . . . . .	16
1.1.3 Faiblesses du modèle EU . . . . .	19
1.1.4 Modèles non-EU . . . . .	22
1.1.5 Du modèle EU au modèle RDU . . . . .	24
1.2 Décision dans l'incertain non risqué . . . . .	32
1.2.1 Modèles à probabilités subjectives . . . . .	32
1.2.2 Décision dans l'incertain total . . . . .	37
1.2.3 Décision dans le risque imprécis . . . . .	48
<b>2 Décision séquentielle dans l'incertain</b>	<b>57</b>
2.1 Formalisation d'un problème de décision séquentielle dans l'incertain . . . . .	59
2.2 Représentation graphique d'un problème de décision séquentielle dans l'incertain . . . . .	60
2.2.1 Arbre de décision . . . . .	61
2.2.2 Diagrammes d'influence . . . . .	64
2.3 Comparaison de stratégies . . . . .	65
2.4 Détermination d'une stratégie optimale par programmation dynamique . . . . .	66
2.5 Cohérence dynamique et conséquentialisme . . . . .	69
2.6 Choix résolu . . . . .	72
2.7 Principe du veto-process . . . . .	74
<b>3 Critère RDU dans les arbres de décision</b>	<b>79</b>
3.1 Complexité . . . . .	81
3.2 Approche par programmation mathématique . . . . .	86
3.3 Approche par énumération implicite . . . . .	91
3.4 Résultats expérimentaux . . . . .	96
3.5 Qui veut gagner des millions ? . . . . .	100
3.6 Stratégies mixtes . . . . .	103

<b>4</b>	<b>Critère RDU dans les diagrammes d'influence</b>	<b>111</b>
4.1	Diagramme d'influence . . . . .	113
4.1.1	Formalisme . . . . .	113
4.2	$\Lambda$ -relaxation de l'ensemble $\overline{\Delta}_*^{DI}$ des stratégies factorisées . . . . .	119
4.3	Algorithme par énumération implicite . . . . .	122
4.3.1	Mise en œuvre algorithmique de la $\Lambda$ -relaxation de l'ensemble $\overline{\Delta}_*^{DI}$ . . . . .	122
4.3.2	Détermination de la stratégie optimisant le critère RDU dans l'ensemble $\overline{\Delta}_\Lambda^T$ . . . . .	124
4.4	Expérimentations numériques . . . . .	128
<b>5</b>	<b>Critère de Hurwicz sous incertitude non risquée</b>	<b>135</b>
5.1	Incertitude totale . . . . .	136
5.1.1	Algorithme de résolution fondé sur la programmation dynamique biobjectif . . . . .	138
5.1.2	Expérimentations numériques . . . . .	144
5.2	Incertitude représentée par des probabilités intervalles . . . . .	145
5.2.1	Optimisation du critère de Hurwicz dans un arbre de décision séparable . . . . .	148
5.2.2	Application du critère de Hurwicz dans un arbre de décision non séparable . . . . .	153
	<b>Conclusion</b>	<b>167</b>



# Introduction

De plus en plus de problèmes étudiés en informatique relèvent de l'aide à la décision. Parmi les applications les plus connues, on peut mentionner par exemple les problèmes de plus court chemin entre deux villes en prenant en compte le coût en essence, le risque d'embouteillage et le temps de trajet, ou encore la détermination de stratégies d'investissements dans les milieux boursiers. De ce fait, le besoin de systèmes performants d'aide à la décision ou encore d'outils de prise de décision automatique ne cesse de croître.

Les systèmes d'aide à la décision sont destinés à conseiller un décideur dans des situations de choix dans lesquelles il lui est difficile de déterminer l'ensemble des meilleurs choix, du fait de la structure complexe des différentes options qui rend difficile leurs comparaisons (par exemple, dans un problème de choix de site pour une centrale nucléaire), ou encore du trop grand nombre d'options qui empêche le décideur de concevoir le problème dans son ensemble (par exemple, dans un problème d'itinéraire routier). Quant à la problématique de la prise de décision automatique, elle s'intéresse à fournir non pas des conseils mais des solutions à appliquer de manière automatique. Ce domaine est très étudié en intelligence artificielle, par exemple dans des problèmes de surveillance d'un système dans lequel il est nécessaire de réagir rapidement, ou encore dans des problématiques de navigation de robot ou de pilotage automatique. Les problématiques sont ici les mêmes que pour les systèmes d'aide à la décision, à la différence près que l'application de la solution résultante se fait sans intervention humaine.

Dans les deux cas, le propos est de fournir des outils permettant de proposer au décideur (ou d'appliquer) une ou plusieurs solutions efficaces. La théorie de la décision fournit alors des critères de décision, en tenant compte des deux dimensions suivantes :

- La dimension *descriptive*, c'est-à-dire l'aptitude du critère à faire émerger des décisions conformes aux préférences du décideur. En effet, chaque décideur possède des préférences qui lui sont propres et désire, autant que possible, rester dans le cadre de ses préférences, sous peine de rejeter la ou les solutions proposées. La prise en compte des préférences du décideur constitue donc un élément important en vue de l'adhésion du décideur à la solution qui lui sera proposée.
- La dimension *normative*, c'est-à-dire l'aptitude du critère à faire émerger des dé-

cisions qui sont conformes à des règles de bon sens communément acceptées.

Pour apprécier la pertinence d'un critère de décision, outre ces deux dimensions, il est important de s'interroger à sa mise en œuvre opérationnelle. À ce titre, l'aspect computationnel est particulièrement important : il s'agit non seulement d'être capable de reproduire (ou de construire) les préférences d'un décideur, mais aussi être capable de calculer concrètement la (ou les) solutions qui lui conviendraient le mieux afin de faire une recommandation. Néanmoins, l'aspect computationnel d'un critère n'est que très rarement pris en compte lors de l'établissement d'un critère de décision et leur application dans la pratique peuvent alors poser d'importants problèmes algorithmiques. C'est particulièrement vrai pour les problèmes abordés dans cette thèse : la décision séquentielle dans l'incertain.

### Décision dans l'incertain

Nous nous intéressons ici aux problèmes de décision sous incertitude. Cette problématique traite des situations de choix où le résultats d'une action n'est pas connue avec certitude. Knight (1921) distingue la notion d'*incertitude* et la notion de *risque*. La notion de risque traite des situations pour lesquelles il existe une distribution de probabilité sur les événements incertains et la notion d'incertitude traite de toutes les autres situations non certaines. Cette distinction entre les deux notions étant aujourd'hui bien établie, nous la conserverons. Cependant, il est d'usage de conserver le terme d'incertitude pour désigner n'importe quelle situation non certaine (i.e., risquée ou non risquée). Nous emploierons ici les termes d'incertitude risquée et d'incertitude non risquée pour distinguer le cas du risque des autres situations.

Dans le cadre du risque, les choix d'un décideur se modélisent au travers de calculs de probabilités des événements futurs. Knight (1921) indique qu'il existe deux manières d'obtenir ces probabilités. La première consiste à attacher à chaque événement une *probabilité logique*<sup>1</sup> que l'on détermine avec un calcul a priori. La seconde consiste à observer des données empiriques, ce qui permet ensuite de dégager des *probabilités statistiques*. De telles distributions de probabilité sont dites « objectives ». Lorsque cette distribution de probabilité « objective » est indisponible, on se retrouve dans le cadre de l'incertitude non risquée. En d'autres termes, on passe d'une situation d'incertitude risquée à une situation d'incertitude pour laquelle il est impossible d'établir une distribution de probabilité en raison du caractère unique et spécifique des événements. Keynes (1937) différencie égale-

---

1. La probabilité logique est définie comme la relation logique entre une proposition et un corps d'hypothèses. Cette probabilité est unique pour un énoncé donné et est indépendante des préférences personnelles. Par exemple la probabilité logique de tirer une boule bleue dans une urne opaque dans laquelle on y aurait placé 30 boules bleues et 20 boules rouges, est de 30/50.

ment ces deux notions : « Par l'expression de connaissance « incertaine », mon intention n'est pas simplement de distinguer ce qui est su avec certitude de ce qui est seulement probable. Le jeu de la roulette n'est pas sujet à l'incertitude, en ce sens, ni la perspective de tirer un bon numéro au loto. Ou encore, l'espérance de vie n'est soumise qu'à une légère incertitude. Même le temps qu'il fait n'est que modérément incertain. Le sens dans lequel j'utilise ce terme est celui selon lequel la perspective d'une guerre européenne était incertaine, ou encore le prix du cuivre et le taux d'intérêt dans vingt ans, ou la date d'obsolescence d'une invention nouvelle, ou la position des détenteurs de fortunes privées dans le système social de 1970. En ces matières, il n'y a pas de fondement scientifique sur lequel on puisse formuler, de façon autorisée, quelque raisonnement probabiliste que ce soit. Nous ne savons pas, tout simplement. ».

Toutefois, certains auteurs supposent que les individus se comportent comme s'ils attribuaient des probabilités « subjectives » aux différents événements incertains. Dans un tel cadre, ce sont les croyances propres aux décideurs qui établissent ces probabilités subjectives (Savage, 1954). Cette approche s'oppose directement à l'approche objectiviste. Une particularité de l'approche subjectiviste est alors de ne pas distinguer les cas d'incertitude risquée et d'incertitude non risquée. Dans cette thèse nous nous intéressons à trois formes d'incertitude :

- le cas de l'*incertitude totale*, traitant des situations d'incertitude pour lesquelles le décideur ne possède aucune information objective sur la vraisemblance des événements,
- le cas du *risque imprécis*, traitant des situations où pour chaque événement on ne connaît que l'intervalle dans lequel se situe la « vraie probabilité » (Jaffray, 1989),
- le cas du *risque*, traitant des situations où l'on possède pour chaque événement une probabilité objective.

Cette liste est en réalité loin d'être exhaustive et de nombreuses autres formes d'incertitude ont été étudiées dans la littérature.

### Objectif de la thèse

Dans cette thèse nous nous intéressons spécifiquement aux problèmes de décision séquentielle dans l'incertain. Ces problèmes concernent les situations d'incertitude dans lesquelles le décideur est amené à prendre plusieurs décisions étalées dans le temps (i.e., établir une *stratégie*). Cette problématique est connue en intelligence artificielle sous le nom de planification dans l'incertain et est très étudiée du fait de ses applications multiples dans de nombreux domaines (diagnostic médical, gestion de stocks, joueurs ar-

tificiels, pilotage automatique...). La théorie de la décision fournit de nombreux critères de décision permettant de comparer les stratégies entre elles. La fin du XXème siècle a vu naître de brillantes idées qui ont permis l'élaboration de critères de décision dans l'incertain très performants aussi bien sur un plan normatif que sur un plan descriptif. Les difficultés computationnelles à leur mise en œuvre conduit en pratique à utiliser des critères moins performants dans les problèmes de décision séquentiels. L'utilisation de critères sophistiqués est en effet freinée par l'absence d'algorithmes efficaces dans la littérature informatique. L'objet de cette thèse est justement de repousser ces verrous algorithmiques en proposant des algorithmes permettant leur application dans des problèmes de décision séquentielle.

### Organisation de la thèse

Cette thèse se compose de cinq chapitres. Dans le premier chapitre, nous introduisons les problèmes de décision sous incertitude et nous présentons les principaux critères de décision permettant de raisonner dans un contexte incertain. Dans le second chapitre, nous présentons la problématique plus large de la décision séquentielle sous incertitude. Nous introduisons dans ce chapitre le formalisme des arbres de décision, qui permet de représenter un problème de décision séquentielle en explicitant l'ensemble des scénarios envisageables. Dans le troisième chapitre, nous nous intéressons à l'optimisation du critère de *l'utilité espérée dépendant du rang* (RDU, *Rank Dependent Utility*) dans les situations d'incertitude risquée. Ce critère combine d'excellentes propriétés normatives et un fort pouvoir descriptif. Cependant, son optimisation pose d'importants problèmes algorithmiques. Dans ce chapitre, nous montrons que l'optimisation du critère RDU est un problème NP-difficile, puis nous proposons des procédures d'optimisation pour son optimisation. Dans le quatrième chapitre, nous nous intéressons à l'optimisation de ce même critère dans le cadre des diagrammes d'influence. A la manière d'un réseau bayésien, ce formalisme permet de tirer profit des indépendances entre les différentes variables pour rendre plus compacte la représentation d'un problème de décision séquentielle. De nouveaux problèmes algorithmiques se posent dès lors que l'on désire conserver les avantages de compacité offerts par ce formalisme. Enfin, dans le cinquième chapitre, nous nous intéressons aux problématiques de décision séquentielle dans les cadres d'incertitude totale et de risque imprécis. Dans ces deux contextes, nous nous intéresserons tout particulièrement à l'optimisation du *critère de Hurwicz*.





# Chapitre 1

## Décision dans l'incertain

**Résumé.** *Dans ce chapitre, nous rappelons les fondements de la théorie de la décision dans l'incertain. Après avoir défini la notion d'incertitude, nous passons en revue quatre modélisations classiques de l'incertitude étudiées dans la littérature : le risque (situation d'incertitude où le décideur connaît les distributions de probabilités sur les événements), l'incertitude totale (situation d'incertitude où le décideur n'a aucune information sur la vraisemblance des événements) et le risque imprécis (situation d'incertitude où le décideur dispose d'une distribution de probabilité intervalle sur les événements élémentaires). Pour chacune de ces modélisations, nous rappelons les principaux critères de décision proposés dans la littérature.*

## Introduction

La *décision dans l'incertain* est l'une des principales branches de la théorie de la décision, à l'instar de la décision multicritère et de la décision collective.

La décision dans l'incertain traite des situations de choix où le décideur est amené à prendre des décisions dans un environnement incertain. L'incertain se rencontre dans de nombreuses situations de la vie courante, ou encore dans de nombreuses problématiques scientifiques (mécanique quantique, économie, médecine, informatique...). L'incertain provient notamment de tous les événements que nous ne sommes pas capables de prédire avec certitude. Un exemple classique d'événement incertain est la météo. En effet, nul n'est capable de déterminer avec certitude le temps qu'il fera demain.

De nombreux débats existent dans la littérature scientifique sur l'existence réelle de l'incertain. Par exemple, en physique, les concepts de déterminisme basés sur les modèles Newtoniens, très à la mode aux XVIII<sup>e</sup> et XIX<sup>e</sup> siècles, maintiennent que tout peut être prédit avec certitude. Néanmoins, les physiciens en mécanique quantique ont mis en évidence que la théorie Newtonienne ne s'appliquait plus dans leur cadre. Heisenberg (1930) ajoute même que les événements qui se réalisent au niveau subatomique surviennent de façon complètement incertaine.

Sans nous étendre sur l'existence réelle ou non de l'incertain, nous adopterons ici le point de vue des économistes sur la nature de l'incertain, qui diffère du point de vue des physiciens ou des philosophes. En économie, est considéré comme faisant partie de l'incertain tout ce qui relève de l'ignorance de l'Homme. Par exemple il est tout à fait possible, en théorie, de prédire le résultat d'un jet de dé si nous disposons de tous les facteurs auxquels il est soumis. Mais à l'échelle humaine, l'ignorance de ces facteurs ainsi que la complexité des calculs à mettre en œuvre, font que le résultat a priori du jet de dé est indéterminé.

Dans la théorie des probabilités, une *épreuve aléatoire* est un phénomène pour lequel toutes les issues possibles sont connues, mais dont on ignore celle qui se réalisera vraiment. L'ensemble des *états de la nature*  $\Omega$  correspond à l'ensemble des issues possibles d'une épreuve aléatoire. On appelle *événement* toute proposition logique qui peut être vérifiée ou non à l'issue de la réalisation d'une épreuve aléatoire. Un événement est donc représentable de manière unique par un sous-ensemble de  $\Omega$ . L'événement  $A \subseteq \Omega$  est réalisé si l'état de la nature  $\omega \in \Omega$  qui se réalise appartient à  $A$ . Un *événement élémentaire* est un événement composé d'un seul élément  $\omega$  de  $\Omega$ .

**Exemple 1** *On a placé dans une urne un ensemble de boules. Une boule peut être rouge, bleue ou verte. L'épreuve aléatoire considérée est le tirage d'une boule au hasard dans cette urne. On distingue alors trois états de la nature possible :*

- $r$  : la boule tirée est rouge,
- $v$  : la boule tirée est verte,
- $b$  : la boule tirée est bleue.

L'ensemble  $\Omega = \{r, v, b\}$  est l'ensemble des états de la nature. A partir de cet ensemble d'états de la nature, on peut considérer les événements suivants :

- $\emptyset$  : l'événement nul ou vide.
- $e_r = \{r\}$  (l'événement traduisant la réalisation de l'état de la nature  $r$ ),  $e_v = \{v\}$  et  $e_b = \{b\}$  : les événements élémentaires.
- $e_1 = e_r \cup e_v$  (traduisant la réalisation d'au moins un état de la nature parmi  $r$  et  $v$ ). Suivant le même principe, toute union de deux événements élémentaires est un événement.
- $\Omega = e_r \cup e_v \cup e_b$  : l'événement certain.

Dans un problème de décision sous incertitude, les événements conduisent à différentes conséquences. Le décideur est alors amené à réaliser un choix entre plusieurs épreuves aléatoires menant à différents ensembles de conséquences. Par exemple, considérons la situation d'incertitude représentée par le tableau de gauche dans la figure 1.1. Le décideur peut choisir d'assurer sa toute nouvelle voiture contre le vol pour un an (décision  $d_1$ ) ou pas (décision  $d_2$ ). Au cours de cette année, il peut soit se faire voler sa voiture (événement  $e_1$ ) soit ne pas se la faire voler (événement  $e_2$ ). De plus, nous savons que la voiture a une valeur de 100 000 Euros et que l'assurance contre le vol coûte 5 000 Euros. Chaque entrée du tableau de gauche, dans la figure 1.1, correspond au gain du décideur selon la décision prise et l'événement qui se réalise. Ainsi la décision  $d_1$  lui assure un coût certain de 5 000 Euros tandis que la décision  $d_2$  lui coûte 100 000 Euros si  $e_1$  se réalise et ne lui coûte rien sinon. Le problème qui se pose ici au décideur est donc un choix entre les deux loteries  $L_1$  et  $L_2$  découlant des décisions  $d_1$  et  $d_2$ . Nous appelons ici une loterie un ensemble d'événements disjoints associés à des conséquences dont l'union de ces événements donne l'événement certain  $\Omega$  (nous avons représentés graphiquement les deux loteries associées aux décisions  $d_1$  et  $d_2$  dans la partie droite de la figure 1.1 page 10). Plus formellement, si  $X = \{x_1, \dots, x_n\}$  est un ensemble fini de  $n$  conséquences, on notera  $L = (x_1, e_1; x_2, e_2; \dots; x_n, e_n)$  la loterie qui associe la conséquence  $x_i$  à chaque événement  $e_i \subseteq \Omega$  pour tout  $0 < i \leq n$  avec  $\bigcup_{i=1}^n e_i = \Omega$  et  $e_i \cap e_j = \emptyset$  pour tout  $0 < i < j \leq n$ . Dans de telles conditions, notre décideur cherche alors à classer, selon ses préférences, les loteries qui lui sont proposées afin de sélectionner la décision le conduisant à la loterie qu'il préfère.

On définit la *relation de préférence*  $\succsim$  sur les loteries qui se lit *est préférée à*. La partie symétrique  $\sim$  et la partie asymétrique  $\succ$  se liront respectivement *est indifférente à* et *est strictement préférée à*. Ainsi,  $L_1 \succsim L_2$  signifie que la loterie  $L_1$  est préférée à la loterie  $L_2$ . Dans l'optique de comparer les différentes loteries d'un problème de décision, la théorie de

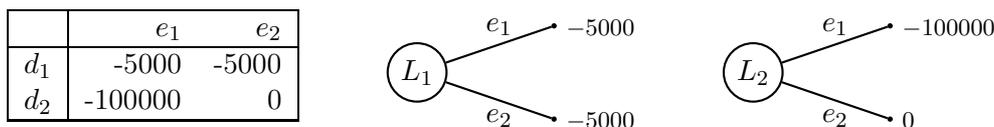


FIGURE 1.1 – Situation de décision sous incertitude et représentation graphique des loteries  $L_1$  et  $L_2$  associées aux décisions  $d_1$  et  $d_2$ .

la décision dans l'incertain propose d'utiliser une *fonction d'évaluation*  $V$  (ou *fonction de valeur* ou encore *fonction représentative des préférences*) pour caractériser une relation de préférence entre loteries : pour toutes loteries  $L_1$  et  $L_2$ ,  $L_1 \succsim L_2 \Leftrightarrow V(L_1) \geq V(L_2)$ .

Pour raisonner dans l'incertain, il est nécessaire de prendre en compte le type de données dont on dispose. En 1921, Knight distingue le « *risque* » et l'« *incertitude* ». Il entend par le terme de « *risque* » toutes les situations pour lesquelles il existe une distribution de probabilité connue sur l'ensemble des états de la nature et par le terme d'« *incertitude* » toutes les autres situations. Nous conserverons ici le sens du « *risque* » proposé par Knight, mais nous emploierons le terme d'« *incertitude* » pour désigner n'importe quelle situation où la réalisation des événements n'est pas certaine. Le terme d'« *incertitude* » est donc un terme très général puisqu'il désigne à la fois des situations très précises, comme les situations dans le risque, mais également toutes les autres situations. Nous nous intéressons ici à trois modélisations différentes de l'incertain :

- le risque : lorsque l'incertitude est représentée par une distribution de probabilité sur les événements,
- l'incertitude totale : lorsque l'on ne possède aucune information sur la vraisemblance d'un événement,
- le risque imprécis : lorsque l'incertitude est représentée par une distribution de probabilité intervalle sur les événements élémentaires.

Ces notions seront toutes détaillées plus finement dans les prochaines sections. Cette liste n'est, bien entendu, pas exhaustive et de nombreuses autres modélisations de l'incertitude peuvent être rencontrées dans la littérature. On peut notamment citer les cas où l'incertitude est représentée par :

- des ensembles de mesures de probabilités (Levi, 1980),
- des fonctions de croyances (Dempster, 1986; Shaffer, 1976),
- des mesures de possibilités et de nécessités (Zadeh, 1978; Dubois et Prade, 1985, 1988; Sabbadin, 1998),
- une relation d'ordre sur la vraisemblance des événements (Dubois et al., 2003; Bouyssou et Pirlot, 2008),
- des p-boxes (Scott et al., 2003).

Le lecteur envious d'approfondir ces notions, d'enrichir cette liste et de hiérarchiser ces

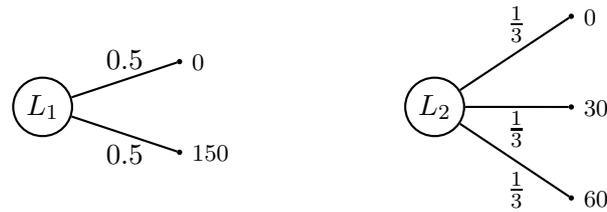


FIGURE 1.2 – Représentation graphique de deux loteries dans un contexte risqué.

représentations de l'incertitude pourra se référer à la thèse de Destercke (2008).

## 1.1 Décision dans le risque

### 1.1.1 Le développement de l'espérance d'utilité

#### Le paradoxe de Saint-Pétersbourg

Lorsque le décideur connaît la distribution de probabilité sur l'ensemble des états de la nature, on se place alors dans le cadre du risque.

Considérons l'exemple suivant afin d'illustrer une situation dans le risque. Un décideur se voit proposé de choisir entre deux jeux. Le premier jeu consiste à choisir parmi deux boîtes : dans la première il y a la somme de 150 Euros, dans la seconde il n'y a rien. Le second jeu consiste à choisir parmi trois boîtes : dans la première il y a la somme de 120 Euros, dans la seconde 60 Euros et enfin dans la dernière 30 Euros. Le décideur se trouve alors face à un choix entre deux *loteries* : une première qui offre un gain de 150 Euros avec une probabilité de  $\frac{1}{2}$  et un gain nul avec une probabilité de  $\frac{1}{2}$ , et une seconde loterie qui offre un gain de 120 euros avec une probabilité de  $\frac{1}{3}$ , un gain de 60 euros avec une probabilité de  $\frac{1}{3}$  et un gain de 30 euros avec une probabilité de  $\frac{1}{3}$  (ces loteries sont représentées graphiquement sur la figure 1.2). Plus précisément, dans le cas du risque, on appelle loterie toute distribution de probabilité sur un ensemble fini de conséquences. Si  $X = \{x_1, \dots, x_n\}$  est un ensemble fini de  $n$  conséquences, on notera  $L = (x_1, p_1; x_2, p_2; \dots; x_n, p_n)$  la loterie qui associe la probabilité  $p_i \in [0, 1]$  à chaque conséquence  $x_i$  pour tout  $0 < i \leq n$  avec  $\sum_{i=1}^n p_i = 1$ . Dans de telles conditions, notre décideur cherche alors à classer par ordre de préférence les loteries qui lui sont proposées afin de sélectionner la meilleure loterie. Pour cela, l'utilisation d'une fonction  $V$  d'évaluation des loteries est nécessaire pour comparer les loteries entre elles.

Au XVIIe siècle, au moment où se développe la théorie moderne des probabilités, des mathématiciens comme Pascal et de Fermat font l'hypothèse que l'évaluation d'une loterie  $L = (x_1, p_1; x_2, p_2; \dots; x_n, p_n)$  se fait via un calcul de son *espérance de gain* (i.e.  $V \equiv E$ ) :  $E(L) = \sum_{i=1}^n p_i \times x_i$ . Ainsi le décideur évaluera le premier jeu à  $\frac{1}{2} \times 150 + \frac{1}{2} \times 0 = 75$  et

évaluera le second jeu à  $\frac{1}{3} \times 120 + \frac{1}{3} \times 60 + \frac{1}{3} \times 30 = 70$ . Il devrait donc faire le choix de jouer au premier jeu qui maximise son espérance de gain.

Cependant, il est intéressant de remarquer qu'un décideur qui prend ses décisions en fonction de l'espérance de gain peut parfois prendre des décisions étonnantes, comme le montre le paradoxe de Saint-Pétersbourg, proposé initialement par Nicholas Bernoulli dans une lettre en 1713. Ce paradoxe a ensuite été repris et modifié par Daniel Bernoulli, son neveu, et a été discuté dans les *Transactions de l'Académie de Saint-Pétersbourg*, d'où son nom.

Le principe du jeu de Saint-Pétersbourg est le suivant : on lance en l'air une pièce de monnaie. Si Face apparaît, la banque paie 2 Euros au joueur, et on arrête le jeu. Sinon, on relance la pièce. Si Face apparaît, la banque paie 4 Euros au joueur, et on arrête le jeu. Sinon, on relance la pièce. Si Face apparaît, la banque paie 8 Euros au joueur, et ainsi de suite. Autrement dit, si Face apparaît pour la première fois au  $n$ -ième lancer, la banque paie  $2^n$  Euros au joueur. On propose maintenant au décideur de déterminer la somme qu'il serait prêt à miser pour jouer à ce jeu. Le décideur va donc déterminer la mise pour laquelle le jeu est équitable, c'est à dire la mise maximale qu'il peut mettre en jeu pour que l'espérance de gain reste positive. Dans ce jeu, pour gagner  $2^n$  euros il faut faire  $n - 1$  fois Pile avant de faire Face, la probabilité de gagner  $2^n$  euros est donc de  $\frac{1}{2^n}$ . Ainsi l'espérance de gain de ce jeu, sans prendre en compte la mise, est de :

$$\sum_{i=1}^{+\infty} \frac{1}{2^i} \times 2^i = \sum_{i=1}^{+\infty} 1 = \underbrace{1 + 1 + \dots + 1}_{\text{une infinité de fois}}$$

De cette manière, un décideur qui se fie à l'espérance de gain pour prendre ses décisions, sera prêt à payer n'importe quelle mise pour pouvoir jouer à ce jeu, puisque son espérance de gain restera infinie quelle que soit la mise qu'il acceptera de payer. Pourtant, aucun décideur raisonnable ne serait prêt à miser plus de quelques Euros pour jouer à ce jeu.

Daniel Bernoulli résout le paradoxe en introduisant l'hypothèse selon laquelle les individus ne maximisent pas l'espérance de gain mais une *espérance morale* :  $\sum_{i=1}^n p_i \times u(x_i)$  où  $u$  est une fonction logarithmique. En effet, pour Bernoulli la valeur subjective d'un jeu n'augmente pas linéairement avec la valeur monétaire de la récompense, mais avec un taux décroissant (un décideur accorde plus de poids à faire passer ses gains de 0 Euros à 100 Euros que de les faire passer de 100 000 Euros à 100 100 Euros). Avec une telle fonction  $u$ , contrairement à l'espérance de gain, l'espérance morale converge et les individus sont maintenant disposés à ne miser qu'une somme modeste pour participer à ce jeu, ce qui est compatible avec les comportements raisonnables observés.

Ainsi, sous l'hypothèse que les décideurs procèdent à un traitement non linéaire des conséquences par le biais de la fonction  $u(\cdot)$ , le paradoxe de Saint-Pétersbourg perd son

caractère paradoxal. La fonction  $u(\cdot)$  est appelée *fonction d'utilité*.

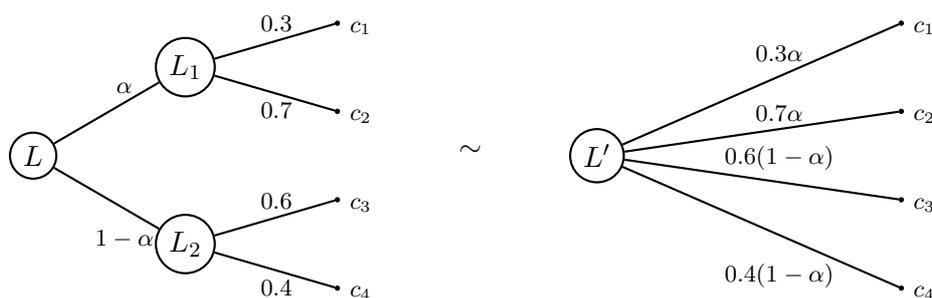
Ce sont ces travaux qui ont inspiré la théorie de l'*utilité espérée* (que l'on appellera par la suite le modèle *EU*, pour *Expected Utility*), proposée par von Neumann et Morgenstern et que nous allons aborder par la suite.

### Le modèle de von Neumann et Morgenstern

Dans l'optique de fournir une théorie solide sur la modélisation des préférences d'un décideur en situation risquée, von Neuman et Morgenstern (1947) généralisent le modèle de Bernoulli fondé sur l'usage d'une fonction d'utilité. Ils montrent, après avoir défini un ensemble d'axiomes, qu'il est toujours possible de construire une fonction d'utilité sur les conséquences, de manière à ce que l'espérance de ces utilités sur les loteries conserve l'ordre de préférence sur les loteries. La fonction d'utilité construite est alors unique à une transformation affine croissante près. En d'autres termes, si la fonction  $u$  modélise les préférences de l'individu, alors il en est de même pour la fonction  $v$  si et seulement si  $v(x) = a \times u(x) + b$  avec  $a > 0$  et  $b$  un réel quelconque.

Dans sa version de base, la théorie de l'espérance d'utilité repose sur quatre axiomes (que nous présentons par la suite). La relation  $\succsim$  doit fournir un pré-ordre total sur les loteries (*axiome de comparabilité* et *axiome de transitivité*). Elle doit également vérifier l'*axiome Archimédien* ainsi que l'*axiome d'indépendance*. Les trois premiers axiomes (comparabilité, transitivité et Archimédien) permettent la représentation des préférences sur les loteries par une fonction d'évaluation  $V$  qui associe à chaque loterie un réel afin de respecter l'équivalence  $L_1 \succsim L_2 \Leftrightarrow V(L_1) \geq V(L_2)$ . La plupart des modèles de décisions respectent ces trois premiers axiomes. Reste maintenant à définir des restrictions supplémentaires sur la fonction  $V$  afin de la rendre plus opérationnelle. Cette restriction nous est justement fournie par l'axiome d'indépendance, et de manière assez forte puisqu'elle impose la linéarité sur le traitement des probabilités de la fonction  $V$ .

Avant de présenter en détail ces quatre axiomes, il est nécessaire d'aborder un axiome qui porte sur les loteries que l'on appelle l'*axiome de réduction des loteries composées* ou encore *axiome de neutralité*. Soit  $X = \{x_1, x_2, \dots, x_n\}$  un ensemble de conséquences non vide, et  $P$  une loi de probabilité sur  $X$  telle que  $P(X) = 1$ . L'ensemble des loteries  $\mathbb{L}_X$  à support dans  $X$  est un ensemble convexe. C'est à dire que  $\forall \alpha \in [0, 1], \forall (L_1, L_2) \in \mathbb{L}_X^2, \alpha L_1 + (1 - \alpha)L_2 \in \mathbb{L}_X$ . Cette propriété de mixage entre deux loteries peut être vue comme une composition de loterie (Karni et Schmeidler, 1991). En d'autres termes, la loterie  $\alpha L_1 + (1 - \alpha)L_2$  correspond à la loterie à deux étapes qui, dans un premier temps,

FIGURE 1.3 – Équivalence entre une loterie composée  $L$  et sa forme réduite  $L'$ .

offre les loteries  $L_1$  et  $L_2$  avec les probabilités  $\alpha$  et  $1 - \alpha$  respectivement et, à la seconde étape, offre la conséquence  $x \in X$  générée par la loterie sélectionnée à la première étape. L'axiome de réduction des loteries composées considère comme équivalentes la loterie composée (à plusieurs étapes) et sa forme réduite (à une étape). Nous illustrons par un exemple graphique cet axiome dans la figure 1.3. Les deux loteries de cette figure sont équivalentes pour le décideur (la loterie composée sur la partie gauche de la figure et sa forme réduite sur la partie droite de la figure). Cet axiome est implicitement contenu dans le modèle EU et la plupart des autres modèles décisionnels dès lors que l'on désire évaluer des loteries composées. La justification de cet axiome repose sur le fait qu'un décideur n'est pas influencé dans ses choix par la modélisation de la situation de risque à laquelle il fait face. En effet, la violation de cet axiome induirait la possibilité de manipuler les choix du décideur. Ainsi, quelle que soit la représentation d'une même situation de choix dans le risque, cette propriété garantit que les préférences du décideur resteront inchangées.

Nous présentons maintenant ces axiomes portant sur la relation binaire de préférence  $\succsim$  sur les loteries.

### Axiome 1 : $\succsim$ doit fournir un pré-ordre total

Une relation binaire fournit un pré-ordre total si et seulement si elle est transitive et complète :

- $\forall (L_1, L_2, L_3) \in \mathbb{L}_X^3$ , si  $L_1 \succsim L_2$  et  $L_2 \succsim L_3$  alors on a  $L_1 \succsim L_3$ .
- $\forall (L_1, L_2) \in \mathbb{L}_X^2$  on a  $L_1 \succsim L_2$  et/ou  $L_2 \succsim L_1$ .

Cet axiome postule que le décideur est toujours capable de comparer deux loteries et de préciser celle qu'il préfère. Cet axiome assure également l'existence d'au moins une loterie que le décideur préfère sur toutes les autres.

**Axiome 2 : axiome Archimédien (ou *axiome de continuité*)**

La relation  $\succsim$  satisfait l'axiome Archimédien si et seulement si  $\forall(L_1, L_2, L_3) \in \mathbb{L}_X^3$  :

$$L_1 \succsim L_2 \succsim L_3 \Rightarrow \exists(\alpha, \beta) \in ]0, 1[^2, \alpha L_1 + (1 - \alpha)L_3 \succsim L_2 \succsim \beta L_1 + (1 - \beta)L_3$$

Cet axiome impose qu'il sera toujours possible de trouver deux réels  $\alpha$  et  $\beta$  tels que :

- $\alpha$  soit suffisamment proche de 1 pour que l'ordre des préférences entre les loteries  $L_1$  et  $L_2$  ne change pas malgré le mélange de  $L_1$  avec la loterie  $L_3$  ;
- $\beta$  soit suffisamment proche de 0 pour que l'ordre des préférences entre les loteries  $L_2$  et  $L_3$  ne change pas malgré le mélange de  $L_1$  avec la loterie  $L_3$  ;

En d'autres termes, cet axiome assure la continuité de la relation de préférence  $\succsim$  sur l'ensemble des loteries. Cet axiome possède un rôle technique et permet d'assurer que l'élicitation des préférences du décideur soit toujours possible grâce à la continuité des courbes d'indifférences. Sans nous étendre sur cette notion, nous réorientons le lecteur soucieux d'approfondir ce point vers l'article de Camerer (1989).

**Axiome 3 : axiome d'indépendance**

La relation  $\succsim$  satisfait l'axiome d'indépendance si et seulement si  $\forall(L_1, L_2, L_3) \in \mathbb{L}_X^3$  et  $\forall\alpha \in ]0, 1[$ , on a :

$$L_1 \succsim L_2 \Leftrightarrow \alpha L_1 + (1 - \alpha)L_3 \succsim \alpha L_2 + (1 - \alpha)L_3$$

Cet axiome postule que la préférence entre deux loteries ne s'inverse pas lorsqu'on les compose avec une loterie commune et une même distribution de probabilité. En effet, le terme  $(1 - \alpha)L_3$  étant commun dans les deux loteries composées obtenues, le choix entre ces deux loteries ne devrait dépendre que de la préférence entre  $L_1$  et  $L_2$ . Cet axiome est en réalité un affaiblissement de l'axiome d'additivité présent dans l'axiomatique de l'espérance de gain. Ce qui fait de l'espérance de gain un cas particulier de l'espérance d'utilité où la fonction  $u(\cdot)$  est égale à la fonction identité (i.e.,  $\forall x \in \mathbb{R}, u(x) = x$ ). L'affaiblissement de l'axiome d'additivité permet alors un traitement non linéaire des conséquences via une fonction d'utilité afin de décrire plus finement les comportements décisionnels observés. Nous verrons par la suite que cet axiome, qui semble raisonnable à première vue, reste encore trop fort et que son acceptation dans la communauté de la théorie de la décision fait l'objet de débats mouvementés.

Dès lors que les axiomes 1, 2 et 3 sont vérifiés, von Neuman et Morgenstern (1947) montrent qu'il est toujours possible de construire une fonction d'utilité  $u$  sur l'ensemble des conséquences afin de modéliser les préférences d'un décideur, et que l'évaluation d'une loterie  $L = (x_1, p_1; \dots; x_n, p_n)$  peut alors se faire via un calcul de son espérance d'utilité (i.e.,  $V \equiv EU$ ) :  $EU(L) = \sum_{i=1}^n p_i \times u(x_i)$ .

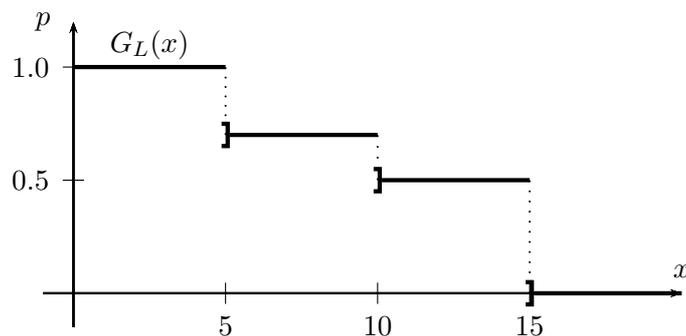


FIGURE 1.4 – Courbe représentative de la fonction décumulative  $G_L$  associée à la loterie  $L = (5, 0.3; 10, 0.2; 15, 0.5)$ .

### 1.1.2 Mesure de l'attitude face au risque

Avant de rappeler les principaux travaux sur les indicateurs qui permettent de caractériser l'attitude d'un décideur vis-à-vis du risque, il est nécessaire d'introduire des définitions sur les différentes mesures du risque.

Étant donnée une loterie  $L = (x_1, p_1; \dots; x_n, p_n)$ , on définit la fonction décumulative  $G_L$  associée à la loterie  $L$  par :

$$G_L(x) = \sum_{\substack{1 \leq i \leq n \\ x_i \geq x}} p_i$$

A la manière des loteries, on pourra noter la fonction décumulative  $G_L$  de la façon suivante :  $G_L = (x_1, G_L(x_1); \dots; x_n, G_L(x_n))$ . Ainsi, la fonction décumulative  $G_L$  associée à la loterie  $L$  mesure la probabilité de gagner au moins  $x$  avec la loterie  $L$ . Dans la figure 1.4, nous donnons la courbe représentative sur  $[0, 20]$  de la fonction décumulative  $G_L = (5, 1; 10, 0.7; 15, 0.5)$  associée à la loterie  $L = (5, 0.3; 10, 0.2; 15, 0.5)$ .

Nous allons maintenant définir la notion de dominance stochastique du premier ordre (First order Stochastic Dominance, *FSD*) entre les loteries.

**Définition 1** Soit  $(L_1, L_2) \in \mathbb{L}_X^2$ ,  $L_1$  domine stochastiquement au premier ordre  $L_2$  si et seulement si :

- $\forall x \in X, G_{L_1}(x) \geq G_{L_2}(x)$ ,
- $\exists x \in X, G_{L_1}(x) > G_{L_2}(x)$ .

On notera  $L_1$  *FSD*  $L_2$  pour dire que la loterie  $L_1$  domine stochastiquement au premier ordre la loterie  $L_2$ . En d'autre terme une loterie en domine une autre lorsqu'elle offre toujours une plus forte probabilité de gagner au moins  $x$  pour toute valeur de  $x$ .

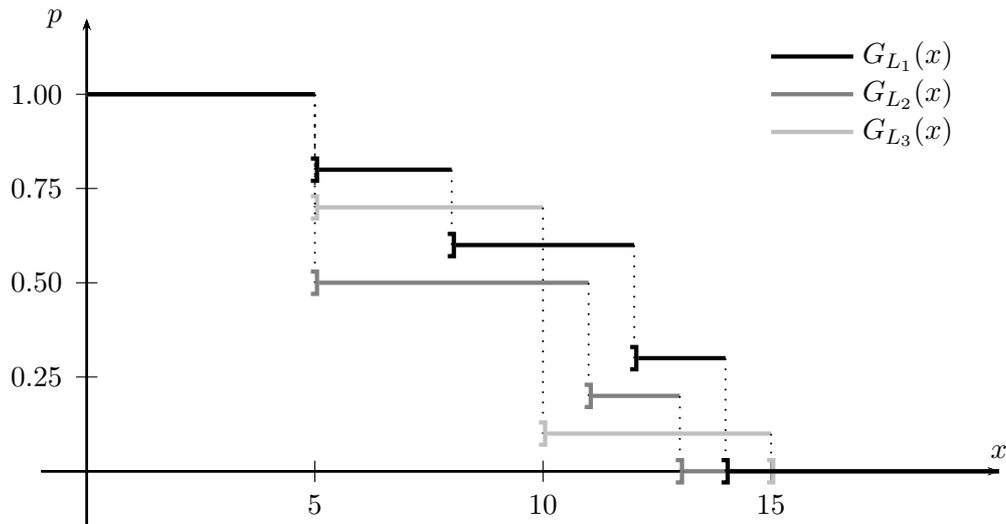


FIGURE 1.5 – Illustration de la dominance stochastique du premier ordre.

**Exemple 2** Dans le repère de la figure 1.5, nous avons représenté les fonctions décumulatives des loteries  $L_1 = (5, 0.2; 8, 0.2; 12, 0.3; 14, 0.3)$  et  $L_2 = (5, 0.5; 11, 0.3; 13, 0.2)$ . On vérifie les relations suivantes :

- $L_1$  FSD  $L_2$  car la courbe  $G_{L_1}$  est toujours au-dessous de la courbe  $G_{L_2}$ ,
- $\text{non}(L_1 \text{ FSD } L_3)$  et  $\text{non}(L_3 \text{ FSD } L_1)$  car la courbe  $G_{L_1}$  se situe parfois au-dessus de la courbe  $G_{L_3}$  et parfois au-dessous de la courbe  $G_{L_3}$ ,
- $\text{non}(L_2 \text{ FSD } L_3)$  et  $\text{non}(L_3 \text{ FSD } L_2)$  car la courbe  $G_{L_2}$  se situe parfois au-dessus de la courbe  $G_{L_3}$  et parfois au-dessous de la courbe  $G_{L_3}$ .

Cette notion est primordiale car elle définit la *rationalité* d'un décideur. On dit qu'un décideur fait des choix rationnels s'il ne préfère jamais une loterie dominée stochastiquement au premier ordre par une autre loterie de son ensemble de choix. Cette condition de rationalité est, bien entendu, fortement désirée lorsque l'on raisonne dans le risque et elle est vérifiée par le modèle EU (i.e.,  $EU(L_1) \geq EU(L_2) \Rightarrow \text{non}(L_2 \text{ FSD } L_1)$ ).

Lorsque deux loteries ne se dominent pas entre elles, le caractère risqué d'une loterie est fortement lié à sa dispersion. Par exemple, une loterie qui offre une conséquence avec une probabilité de 1 est une loterie qui offre de manière sûre une valeur certaine et est donc, par définition, sans caractère risqué. Ainsi, toute mesure de la dispersion d'une loterie est susceptible de donner une indication sur le caractère risqué d'une loterie. C'est alors tout naturellement que la variance a été pendant longtemps la mesure la plus utilisée pour évaluer le caractère risqué d'une loterie. Cependant, les arguments contre la variance, que nous n'exposerons pas ici, ne manquent pas. Pour pallier les faiblesses de la

variance, en 1970, Rothschild et Stiglitz (1970) introduisent la notion d'*accroissement de risque à moyenne constante* (*Mean Preserving Spread, MPS*). Pour définir clairement la notion de *MPS* nous devons introduire la relation de *dominance stochastique au second ordre* (*Second order Stochastic Dominance, SSD*) sur les loteries.

**Définition 2** Soit  $(L_1, L_2) \in \mathbb{L}_X^2$  avec  $X = \{x_1, \dots, x_n\}$ ,  $L_1$  domine stochastiquement au second ordre  $L_2$  si et seulement si :

- $\forall x \in X, \int_x^{+\infty} G_{L_1}(t) dt \geq \int_x^{+\infty} G_{L_2}(t) dt,$
- $\exists x \in X, \int_x^{+\infty} G_{L_1}(t) dt > \int_x^{+\infty} G_{L_2}(t) dt.$

On notera  $L_1$  *SSD*  $L_2$  pour dire que la loterie  $L_1$  domine stochastiquement au second ordre la loterie  $L_2$ . Notons qu'il est possible, en intégrant plusieurs fois la fonction décumulative d'une loterie, de définir des dominances stochastiques de n'importe quel ordre, mais nous n'aurons ici besoin que du premier ordre et du second ordre. Notons enfin que si une loterie domine au premier ordre une autre loterie, alors elle la domine également au second ordre. A partir de cette notion, on définit la relation de *MPS* comme suit :

**Définition 3** Soit  $(L_1, L_2) \in \mathbb{L}_X^2$ ,  $L_1$  est un *MPS* de  $L_2$  si et seulement si :

- $E(L_1) = E(L_2)$
- $L_1$  *SSD*  $L_2$

où  $E(L)$  désigne l'espérance de gain offerte par la loterie  $L$ . On notera  $L_1$  *MPS*  $L_2$  pour dire que la loterie  $L_1$  est un *MPS* de  $L_2$ . Si  $L_1$  est un *MPS* de  $L_2$ , alors la loterie  $L_1$  est considérée comme étant moins risquée que  $L_2$ .

### Mesure de l'aversion au risque

La théorie de la décision dans le risque fournit des propriétés intrinsèques (indépendantes du modèle de décision considéré) afin de caractériser l'*aversion au risque* d'un décideur. L'aversion au risque d'un décideur est l'indicateur le plus fréquemment utilisé en raison de son grand champs d'application. On distingue principalement l'*aversion faible au risque* et l'*aversion forte au risque*. Durant plusieurs années, ces deux notions ont été confondues au vue de l'équivalence de leurs caractérisations dans le cadre du modèle EU. C'est d'ailleurs la possibilité de distinguer ces deux caractérisations d'aversion au risque qui confère aux modèles non-EU un pouvoir de modélisation des préférences bien supérieur à celui du modèle EU.

Un décideur a une aversion faible au risque s'il préfère toujours l'espérance d'une loterie à la loterie elle-même (i.e.  $L' = (E(L), 1) \succsim L$ ). Ainsi, un décideur ayant une faible aversion au risque préfère toujours la certitude au risque. Cet indicateur est de

loin le plus utilisé dans la littérature du fait de son caractère intuitif. De manière duale, on peut dire qu'un décideur est faiblement attiré par le risque (resp. neutre) s'il préfère toujours la loterie  $L$  à la loterie  $L'$  (resp. s'il est toujours indifférent entre  $L$  et  $L'$ ).

La deuxième caractérisation de l'aversion au risque (Cohen et al., 1987), repose sur la notion d'accroissement de risque à moyenne constante. Un décideur possède une forte aversion au risque (resp. attirance forte pour le risque) si pour toutes loteries  $L_1$  et  $L_2$  de  $\mathbb{L}_X$  telles que  $L_2$  MPS  $L_1$ , il préfère toujours  $L_1$  à  $L_2$  (resp. préfère toujours  $L_2$  à  $L_1$ ).

On montre facilement que l'aversion forte au risque implique l'aversion faible au risque de même que l'attraction forte au risque implique l'attraction faible au risque. Quant à la notion de neutralité face au risque, la version faible est équivalente à la version forte. Un décideur qui possède une aversion au risque est dit *adversaire du risque* et un décideur qui possède une attirance pour le risque est dit *joueur*.

La différence entre l'aversion faible et forte est très importante. Bien que l'on montre facilement que l'aversion forte implique l'aversion faible, la relation inverse n'est que rarement vérifiée. En effet, un décideur peut être hostile à l'introduction de risque dans une situation mais être ensuite indifférent à son accroissement.

### 1.1.3 Faiblesses du modèle EU

Dans le cadre du modèle EU, l'aversion faible au risque et l'aversion forte au risque sont deux notions équivalentes. En effet, dans le modèle EU, l'aversion faible au risque se traduit par  $u(\sum_{i=1}^n p_i \times x_i) \geq \sum_{i=1}^n p_i \times u(x_i)$ . En d'autres termes, un décideur qui possède une faible aversion au risque sera doté d'une fonction d'utilité  $u$  concave. Notons que pour un décideur neutre face au risque (resp. ayant une attraction faible pour le risque) l'inégalité devient une égalité (resp. l'inégalité s'inverse) et cela revient à doter le décideur d'une fonction d'utilité linéaire (resp. convexe). De plus, pour toute fonction d'utilité  $u$  concave, les deux propositions suivantes sont équivalentes :

- $L_2$  MPS  $L_1$
- $EU(L_1) \geq EU(L_2)$

où  $L_1$  et  $L_2$  sont deux loteries. Pour résumer, dans le cadre du modèle EU, avoir une aversion faible pour le risque est une condition suffisante pour être doté d'une fonction d'utilité concave, et la concavité de la fonction d'utilité est une condition suffisante pour avoir une aversion forte pour le risque. Puisque, dans le cas général, avoir une forte aversion pour le risque est une condition suffisante pour avoir une aversion faible pour le risque, il s'ensuit que ces deux notions d'aversion au risque sont bel et bien équivalentes dans le cadre du modèle EU.

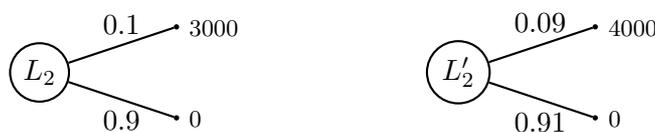
L'impossibilité de distinguer ces deux notions d'aversion dans le cadre du modèle EU se révèle très problématique dès lors que l'on cherche à modéliser finement les pré-

férences d'un décideur. Ainsi, de nombreuses préférences fréquemment rencontrées, sont impossibles à modéliser avec le critère EU. En guise d'illustration, nous présentons dans l'exemple 3 une version moderne (Kahneman et Tversky, 1979) (tableau 1.2 page 21) du célèbre *paradoxe d'Allais* (Allais, 1953) (tableau 1.1 page 21).

**Exemple 3** On propose à un décideur, dans un premier problème, de choisir entre deux loteries  $L_1$  et  $L'_1$ , et entre les loteries  $L_2$  et  $L'_2$  dans un second problème. Dans le premier problème, la loterie  $L_1$  offre un gain certain de 3 000 Euros et la loterie  $L'_1$  offre la perspective d'un gain de 4 000 Euros avec une probabilité de 0.9 et un gain nul avec une probabilité de 0.1. Les loteries  $L_1$  et  $L'_1$  sont représentées graphiquement ci-dessous :



Dans le second problème, la loterie  $L_2$  propose un gain de 3 000 Euros avec la probabilité 0.1 et un gain nul avec une probabilité de 0.9 tandis que la loterie  $L'_2$  offre le gain de 4 000 Euros avec la probabilité de 0.09 et un gain nul avec une probabilité de 0.91. Les loteries  $L_2$  et  $L'_2$  sont représentées graphiquement ci-dessous :



Dans le premier problème, la majorité des personnes soumises à cette expérience préfèrent la loterie  $L_1$  à  $L'_1$  (ils préfèrent être sûrs de gagner 3 000 Euros que de risquer de ne rien gagner pour un surplus de gain qui leur semble insuffisant), alors que dans le second problème, la majorité des personnes préfèrent la loterie  $L'_2$  à la loterie  $L_2$  (car la probabilité de gagner 4 000 Euros dans la loteries  $L'_2$  est presque la même que la probabilité de gagner 3 000 Euros dans la loterie  $L_2$ ). Si l'on traduit ces préférences, dans le cadre du modèle EU, on obtient pour le premier choix :

$$\begin{aligned}
 L_1 \succ L'_1 &\Leftrightarrow EU(L_1) > EU(L'_1) \\
 &\Leftrightarrow u(3000) > 0.1u(0) + 0.9u(4000) \\
 &\Leftrightarrow 0.1u(3000) > 0.01u(0) + 0.09u(4000) \\
 &\Leftrightarrow 0.1u(3000) + 0.9u(0) > 0.01u(0) + 0.09u(4000) + 0.9u(0) \\
 &\Leftrightarrow 0.9u(0) + 0.1u(3000) > 0.91u(0) + 0.09u(4000) \\
 &\Leftrightarrow L_2 \succ L'_2
 \end{aligned}$$

Ainsi, quelle que soit la fonction d'utilité utilisée, la préférence pour  $L_1$  sur  $L'_1$  est équivalente à la préférence pour  $L_2$  sur  $L'_2$  dans le cadre du modèle EU. Ce qui est incompatible avec la majorité des préférences des personnes soumises à cette expérience.

En fait, Allais qualifie ce comportement, loin d'être paradoxal, comme un comportement raisonnable de « preference for security in the neighbourhood of certainty » (Allais, 1997). En d'autres termes, « Un tiens vaut mieux que deux tu l'auras ». Ce phénomène est aujourd'hui bien connu sous le nom de *l'effet de certitude*. Le renversement de préférence s'explique ici simplement par l'attraction forte du passage de la quasi-certitude à la certitude, et à l'indifférence du passage du peu probable au un peu plus probable. A cela s'ajoute un phénomène de sur-pondération des résultats qui sont certains par rapport aux résultats qui ne sont que probables. Sans nous étendre sur l'effet de certitude, le lecteur enieux d'approfondir la question pourra se référer à la généralisation proposée par Starmer (2000).

De nombreux autres exemples et critiques, que nous ne détaillerons pas ici, ont été formulés dans la littérature économique à l'encontre du modèle EU et la plupart s'accordent à dire que l'axiome d'indépendance est une hypothèse bien trop forte comme le montre le paradoxe d'Allais. Gayant (1997) qualifie même de paradoxal le fait de modéliser l'aversion au risque d'un décideur en se fondant sur une fonction portant uniquement sur les conséquences (sans la prise en compte d'éléments probabilistes). Au début du millénaire, Rabin (2000) énonce son théorème d'impossibilité sur le critère EU : l'impossibilité de traduire une aversion pour le risque lorsque les enjeux sont faibles dans le cadre du modèle EU. En effet, la concavité de la fonction d'utilité permet d'expliquer l'aversion au risque lorsque les enjeux sont très importants mais implique également une quasi-neutralité vis-à-vis du risque lorsque les enjeux sont plus faibles (Arrow, 1971). En plus de rendre abusivement élevée l'aversion pour le risque lorsque les enjeux sont importants, de nombreuses expériences ont montré l'existence d'une aversion au risque sur les petits enjeux. L'existence d'une aversion pour les petits enjeux met alors en échec la capacité du modèle EU à décrire l'aversion pour le risque.

La solution suggérée pour sortir de ce théorème d'impossibilité est de décharger la

Loterie	0.01	0.1	0.89
$L_1$	100M	100M	100M
$L'_1$	0M	500M	100M
$L_2$	100M	100M	0M
$L'_2$	0M	500M	0M

TABLE 1.1 – Version d'origine du paradoxe d'Allais. Le tableau indique dans chaque case le gain offert par chaque loterie (ligne) selon une certaine probabilité (colonne). La majorité des personnes soumises à l'expérience ont préféré  $L_1$  à  $L'_1$  et  $L'_2$  à  $L_2$ .

Loterie	0	3000	4000
$L_1$	0.00	1.00	0.00
$L'_1$	0.10	0.00	0.90
$L_2$	0.90	0.10	0.00
$L'_2$	0.91	0.00	0.09

TABLE 1.2 – Version moderne du paradoxe d'Allais. Le tableau indique dans chaque case la probabilité d'obtenir chacun des gains (colonne) avec chacune des loteries (ligne). La majorité des personnes soumises à l'expérience ont préféré  $L_1$  à  $L'_1$  et  $L'_2$  à  $L_2$ .

fonction d'utilité pour qu'elle ne décrive plus l'aversion au risque mais se contente uniquement de son rôle premier de traduire le degré d'importance d'une conséquence pour le décideur. Nous détaillerons plus en détail ce point lorsque nous traiterons des modèles à transformation des probabilités.

#### 1.1.4 Modèles non-EU

Si la mise en défaut de la capacité du modèle EU à modéliser les comportements décisionnels datent des années 1950, les propositions de nouveaux modèles décisionnels font leur apparition dans les années 1970.

De façon analogue aux limites de la somme pondérée pour comparer des vecteurs de performance, il est important de souligner que l'utilisation du modèle EU peut s'avérer être problématique dans certains cas puisqu'il ne permet pas d'atteindre toutes les solutions efficaces (i.e. stratégies non dominées stochastiquement au premier ordre) dont certaines qui pourraient être préférables dans des problèmes concrets que l'on rencontre en pratique. De plus, la surcharge de la fonction d'utilité (qui porte le double rôle d'encoder les conséquences sur une échelle numérique, éventuellement déformée, et de modéliser l'attitude vis-à-vis du risque) rend difficile l'élicitation d'une fonction d'utilité pertinente pour un agent et oblige généralement les programmeurs à générer une fonction d'utilité propre au problème (qui permettra d'atteindre les solutions désirées pour ce problème précis) et non représentative des préférences que l'on voudrait donner à notre agent dans un cadre général. Une modification, même mineure, du problème peut alors devenir un vrai calvaire puisqu'il faut répercuter ces modifications sur la fonction d'utilité de chacun de nos agents.

Dans la suite, nous présentons les principales alternatives au modèle EU proposées dans la littérature. Afin de rester dans une démarche d'aide à la décision, notons que nous nous focaliserons uniquement sur les modèles qui respectent la dominance stochastique du premier ordre.

Le soucis d'efficacité d'une solution est une notion indispensable en aide à la décision. Il serait très difficile de défendre une solution qui donne toujours une probabilité inférieure de gagner la même chose qu'une autre solution. Ce point de vue est à opposer à celui des économistes ou psychologues, dont les préoccupations peuvent parfois se focaliser uniquement sur la description de comportements, quitte à rejeter la notion de rationalité. Ainsi, tous les modèles que nous aborderons par la suite dans le cadre du risque seront des modèles qui respectent la relation de dominance stochastique du premier ordre (aussi appelés *modèles conventionnels* dans la littérature).

Dans la multitude de propositions de modèles non-EU qu'aura connu la fin du XXe

siècle, nous ne retiendrons que les modèles qui ont marqué la communauté scientifique et/ou qui sont actuellement utilisés. On peut, dans un premier temps, citer le modèle de l'*utilité espérée généralisée* proposé par Machina (1982). Ce dernier part du principe que le respect de l'axiome d'indépendance n'est pas une condition nécessaire pour préserver l'hypothèse de maximisation de l'utilité espérée. Ainsi, ce modèle rejette totalement l'axiome d'indépendance et pose des conditions analytiques sur les préférences. Si ce modèle résout quelques situations qui mettaient dans l'impasse le modèle EU, il reste encore trop proche du modèle EU et de légères modifications des paradoxes classiques (comme le paradoxe d'Allais) les font redevenir insolubles.

Au lieu de rejeter totalement l'axiome d'indépendance, plusieurs modèles se sont construits autour d'un affaiblissement de l'axiome d'indépendance. Ces modèles, dit avec *betweenness*, sont des modèles conservateurs, dans le sens où leur objectif est de préserver le plus possible les propriétés normatives du modèle EU. En conséquence, ces modèles ne sont pas réellement une alternative au modèle EU car l'élargissement de leur capacité à modéliser des comportements rationnels reste très subtil. Les modèles de cette famille se basent sur un affaiblissement de l'axiome d'indépendance de la manière suivante (hypothèse de *betweenness*) :

$$\forall (L_1, L_2) \in \mathbb{L}_X^2, \forall \alpha \in ]0, 1[, L_1 \succsim L_2 \Leftrightarrow L_1 \succsim \alpha L_1 + (1 - \alpha)L_2 \succsim L_2$$

Autrement dit, lorsqu'une loterie est préférée à une seconde, alors toutes les loteries composées par ces deux loteries, sont comprises entre les loteries initiales en terme de préférence. Si cet affaiblissement permet en théorie de résoudre certaines situations où l'axiome d'indépendance est violé, il est en pratique violé dans les mêmes situations qui posaient problème avec l'axiome d'indépendance (notamment dans les problèmes mettant en jeu des petites probabilités (Camerer, 1992; Harless, 1992)). Sans rentrer dans les détails, parmi les modèles faisant l'hypothèse de *betweenness*, on peut citer l'*utilité pondérée* (Weighted utility theory, Chew et McCrimmon (1979); Chew (1983)), la *théorie de l'utilité SSB (Skew Symmetric Bilinear)* (Fishburn, 1982), l'*utilité pondérée implicite* (Fishburn, 1983; Dekel, 1986; Chew, 1989) et la *théorie de la déception* (Bell, 1985; Loomes et Sugden, 1986; Gul, 1991).

Enfin, il existe une dernière famille de modèles (modèles avec transformation des probabilités, dits *modèles à dépendance des rangs*) très présente dans la littérature économique, qui repose cette fois sur une restriction de l'axiome d'indépendance et non sur son affaiblissement. Parmi ces modèles citons l'*utilité anticipée* (Quiggin, 1982) plus connue sous le nom de l'*utilité espérée dépendant du rang (Rank Dependent Utility, RDU)*, la *théorie duale (Dual Theory, DT)* proposée par Yaari (1987), le *modèle de Allais* (Allais,

	Modèles de décision
Rejet de l'axiome d'indépendance	- <i>utilité espérée généralisée</i> (Machina, 1982)
Affaiblissement de l'axiome d'indépendance	- <i>utilité pondérée</i> (Chew et McCrimmon, 1979) - <i>théorie de l'utilité SSB</i> (Fishburn, 1982) - <i>utilité pondérée implicite</i> (Chew, 1989) - <i>théorie de la déception</i> (Gul, 1991)
Restriction de l'axiome d'indépendance	- <i>utilité espérée dépendant du rang</i> , RDU (Quiggin, 1982) - <i>théorie duale</i> , DT (Yaari, 1987) - <i>modèle de Allais</i> (Allais, 1988) - <i>théorie des prospects cumulés</i> , CPT (Tversky et Kahneman, 1992)

TABLE 1.3 – Modèles de décision non-EU dans le risque

1988) ou encore la *théorie des prospects cumulés* (*Cumulative Prospect Theory*, CPT) (Tversky et Kahneman, 1992). Contrairement aux modèles cités précédemment (i.e. le modèle de Machina et les modèles avec *betweenness*), ceux là présentent l'avantage d'élargir significativement le pouvoir descriptif du modèle EU, et de corriger efficacement les lacunes de ce dernier. Tous les modèles de cette famille reposent sur un même principe que nous illustrerons au travers du modèle RDU dans la partie suivante. Enfin, nous préciserons les différences entre ces divers modèles à dépendance des rangs après avoir présenté en détail le modèle RDU.

Le tableau 1.3 résume la classification des différents modèles de décision dans le risque. Cette liste n'est pas exhaustive et de nombreux autres modèles de décision dans le risque existent. Comme dit précédemment, nous nous sommes restreints aux modèles qui respectent la dominance stochastique du premier ordre et aux plus utilisés/cités dans la littérature économique.

### 1.1.5 Du modèle EU au modèle RDU

Buffon a proposé une seconde façon de résoudre le paradoxe de Saint-Pétersbourg, par l'introduction de l'hypothèse que les décideurs négligent les petites probabilités et considèrent comme impossible une succession de plus de  $n$  « Pile » à la suite (pour un  $n$  fixé). Le décideur se refuse alors à payer une mise supérieure à  $2^n$ . C'est justement cette idée qui a inspiré les modèles à transformation des probabilités. Cette idée de traiter non linéairement les probabilités est la vision duale du traitement non linéaire des conséquences. Ces deux idées ont le même statut et il n'y a, a priori, aucune raison de

privilégier l'une plus que l'autre d'après Gayant (2001). Cependant la tâche s'avère plus compliquée que la déformation des conséquences par une fonction d'utilité et la première tentative avec le modèle de Handa (1977) n'est pas satisfaisante. Dans son modèle, il décharge la fonction d'utilité en ajoutant une fonction croissante  $\varphi : [0; 1] \rightarrow [0; 1]$  telle que  $\varphi(0) = 0$  et  $\varphi(1) = 1$  qui permet de déformer les probabilités. Le modèle de Handa évalue alors une loterie  $L = (x_1, p_1; \dots; x_n, p_n)$  de la manière suivante :

$$V(L) = \sum_{i=1}^n \varphi(p_i)u(x_i)$$

Si l'idée semble séduisante, elle est rapidement abandonnée suite aux interventions de Fishburn (1978). En effet, ce dernier montre que le modèle de Handa peut conduire à prendre des décisions menant à des loteries dominées stochastiquement au premier ordre. Nous illustrons cet argument dans l'exemple 4 proposé par Gayant (2001).

**Exemple 4** Dans cet exemple, on pose  $\varphi(1/3) = 1/4$  et  $\varphi(2/3) = 2/3$ . On fait de plus l'hypothèse que  $\forall x \in \mathbb{R}, u(x) = x$ . Soit  $L_1 = (60, 1/3; 64, 1/3; 120, 1/3)$  et  $L_2 = (60, 2/3; 120, 1/3)$ . On a donc  $L_1$  FSD  $L_2$ . Le modèle de Handa attribue la valeur 61 à la loterie  $L_1$  et la valeur 70 à la loterie  $L_2$ . Ainsi, un décideur qui aurait des préférences qui suivent le modèle de Handa préfère la loterie  $L_2$  à la loterie  $L_1$ , alors que la loterie  $L_2$  est dominée par  $L_1$ .

### Modèle RDU

L'idée de déformer les probabilités aurait pu sombrer dans l'oubli si Quiggin (1982) n'avait pas eu l'idée clef de transformer non pas des probabilités mais des cumuls de probabilités afin d'écarter toute violation de la dominance stochastique du premier ordre.

Cette idée est le principe sur lequel repose l'espérance d'utilité avec dépendance du rang (Rank Dependent Utility, *RDU*).

En réalité, le modèle proposé par Quiggin repose sur une seconde intuition, l'hypothèse de dépendance du rang : le poids attribué à une conséquence ne dépend pas uniquement de la probabilité d'obtenir cette conséquence, mais également de l'attractivité du rang de cette conséquence par rapport aux autres conséquences possibles. Ainsi, la transformation des cumuls de probabilités et l'hypothèse de dépendance du rang ont permis l'établissement d'un modèle solide pour représenter les préférences d'un décideur dans le risque. Soit  $L = (x_1, p_1; \dots; x_n, p_n)$  telle que  $x_1 < \dots < x_n$ , la formule d'évaluation d'une loterie au sens de EU peut facilement se réarranger de la manière suivante :

$$\begin{aligned}
EU(L) &= \sum_{i=1}^n p_i u(x_i) \\
\Leftrightarrow EU(L) &= p_1 \times u(x_1) + p_2 \times u(x_2) + \dots + p_n \times u(x_n) \\
\Leftrightarrow EU(L) &= (p_1 + p_2 + \dots + p_n) \times u(x_1) + (p_2 + \dots + p_n) \times [u(x_2) - u(x_1)] \\
&\quad + \dots + p_n \times [u(x_n) - u(x_{n-1})] \\
\Leftrightarrow EU(L) &= u(x_1) + \sum_{i=2}^n \sum_{j=i}^n p_j [u(x_i) - u(x_{i-1})] \\
\Leftrightarrow EU(L) &= u(x_1) + \sum_{i=2}^n G_L(x_i) [u(x_i) - u(x_{i-1})]
\end{aligned}$$

A partir de là, le modèle RDU propose alors de déformer le cumul  $G_L(x_i)$  de probabilités présent dans chaque terme dépendant de  $i$  à l'aide d'une fonction de déformation des probabilités. La fonction représentative des préférences du modèle  $RDU$  ( $V \equiv RDU$ ) s'écrit alors :

$$RDU(L) = u(x_1) + \sum_{i=2}^n \varphi(G_L(x_i)) [u(x_i) - u(x_{i-1})]$$

où  $\varphi : [0; 1] \rightarrow [0; 1]$  est une fonction croissante telle que  $\varphi(0) = 0$  et  $\varphi(1) = 1$ .

L'introduction d'une telle fonction de transformation des probabilités est maintenant compatible avec la dominance stochastique au premier ordre. Le gain offert par une telle fonction de déformation des probabilités est considérable autant au niveau pratique qu'au niveau descriptif.

En pratique, les problèmes rencontrés en informatique ne portent pas nécessairement sur des conséquences numériques. Par exemple, pour un problème de navigation, les conséquences possibles pourraient être des positions sur une carte, ou encore, dans un problème de diagnostic médical, on pourrait avoir des conséquences dépendant de la santé du patient. Ainsi, la fonction d'utilité utilisée peut ne pas être qu'une simple déformation des valeurs des conséquences mais une fonction qui attribue des valeurs d'utilité à chacune des conséquences. Il devient alors impossible de donner une forme (concave ou même convexe) à la fonction d'utilité afin de décrire l'attitude vis-à-vis du risque du décideur, rendant inopérant le modèle EU dans ces problèmes. En effet, dans le cadre du modèle EU, la fonction d'utilité est surchargée du fait qu'elle endosse les rôles suivants :

- attribuer une utilité numérique à une conséquence dans le certain,
- modéliser l'attitude, du décideur, vis-à-vis du risque.

Dans le modèle RDU, la fonction de déformation des probabilités modélise l'attitude vis-à-vis du risque. Cette dernière décharge la fonction d'utilité, qui peut alors se consacrer pleinement à l'encodage des conséquences. Il devient, de ce fait, possible de traiter les problèmes où les conséquences ne sont pas, initialement, des valeurs numériques. D'un point de vue descriptif, la fonction  $\varphi$  permet alors de résoudre les divers paradoxes de la littérature, tel que le paradoxe d'Allais (voir l'exemple 5 qui suit).

**Exemple 5** Revenons sur l'exemple 3 page 20. On pose  $\varphi(0.09) = \varphi(0.1) = 0.2$  et  $\varphi(0.9) = 0.7$ . On fait l'hypothèse que  $\forall x \in \mathbb{R}, u(x) = x$ . Les préférences induites par le modèle RDU sont maintenant compatibles avec l'exemple de Kahneman et Tversky, en effet, on a :

$$RDU(L_1) = u(3000) = 3000$$

$$RDU(L'_1) = u(0) + \varphi(0.9)(u(4000) - u(0)) = 2800$$

Le décideur préfère donc la loterie  $L_1$  à  $L'_1$ . De même on a :

$$RDU(L_2) = u(0) + \varphi(0.1)(u(3000) - u(0)) = 600$$

$$RDU(L'_2) = u(0) + \varphi(0.09)(u(4000) - u(0)) = 800$$

Le décideur préfère donc la loterie  $L'_2$  à la loterie  $L_2$ . Il n'y a donc plus aucun paradoxe dès qu'on utilise le modèle RDU pour décrire les préférences du décideur.

Bien entendu, on peut remarquer que le modèle RDU est une généralisation du modèle EU. Le cas particulier du modèle EU se retrouve en définissant  $\varphi$  comme la fonction identité. Cette généralisation est fondée sur la restriction du *principe de la chose sûre* (*sure thing principle*) que l'on peut formuler comme suit : soit  $L_1 = (x_1, p_1; \dots; x_n, p_n)$  et  $L'_1 = (x'_1, p_1; \dots; x'_n, p_n)$  telles qu'il existe  $x_i = x'_j$ , alors la préférence de  $L_1$  sur  $L'_1$  est équivalente à la préférence de  $L_2$  sur  $L'_2$ , où  $L_2$  et  $L'_2$  sont obtenus en remplaçant la conséquence commune  $x_i$  par une autre dans  $L_1$  et  $L'_1$ .

Dans le paradoxe d'Allais (tableau 1.1 page 21), cet axiome est clairement transgressé : il y a une probabilité de 89/100 de gagner 100M d'Euros dans  $L_1$  et  $L'_1$  alors qu'il y a une probabilité de 89/100 de ne rien gagner dans  $L_2$  et  $L'_2$ , et pourtant, les préférences  $L_1 \succ L'_1$  et  $L'_2 \succ L_2$  sont observées pour la majorité des personnes interrogées.

Dans le cadre du modèle RDU, cet axiome a été restreint aux cas où la conséquence commune possède le même rang dans les deux loteries et que son remplacement n'affecte pas le rang des conséquences dans les deux loteries : soit  $L_1 = (x_1, p_1; \dots; x_n, p_n)$  et  $L'_1 = (x'_1, p_1; \dots; x'_n, p_n)$  telles qu'il existe  $x_{i_0} = x'_{i_0}$  pour un certain  $i_0$  et telles que  $x_1 < \dots < x_n$  et  $x'_1 < \dots < x'_n$ , alors la préférence de  $L_1$  sur  $L'_1$  est équivalente à la préférence de  $L_2$  sur  $L'_2$ , où  $L_2$  et  $L'_2$  sont obtenues en remplaçant la conséquence commune  $x_{i_0}$  par une conséquence  $y$  telle que  $y$  soit de rang  $i_0$  dans les deux loteries. Cette restriction est connue sous le nom de *principe de la chose sûre comonotone* (*comonotonic sure thing principle*).

En réalité cette restriction du principe de la chose sûre se transpose aisément dans le cadre des axiomes de EU en restreignant l'axiome d'indépendance aux loteries issues de variables aléatoires comonotones<sup>1</sup>. Dans le modèle RDU c'est justement une telle restriction de l'axiome d'indépendance qui est faite. Cette restriction est parfaitement

---

1. Etant donné  $\Omega$  un ensemble d'états de la nature, deux variables aléatoires réelles  $X$  et  $Y$  définies sur  $\Omega$  sont dites comonotones si et seulement si il n'existe pas  $\omega \in \Omega$  et  $\omega' \in \Omega, \omega \neq \omega'$  tels que :  $X(\omega) > Y(\omega)$  et  $X(\omega') < Y(\omega')$ .

justifiée dès lors que l'on remarque que la loterie, issue du mélange de deux variables aléatoires comonotones, ne peut qu'accroître le risque encouru tandis que la loterie issue du mélange de deux variables aléatoires non-comonotones peut accroître, faire décroître ou même annihiler le risque encouru.

### Mesure de l'attitude face au risque avec le modèle RDU

Grâce à l'introduction d'une fonction  $\varphi$  de déformation des probabilités, nous possédons maintenant un second levier pour décrire proprement le comportement d'un décideur dans une situation de risque : il est maintenant possible de focaliser le rôle de la fonction d'utilité uniquement sur l'attitude vis-à-vis des richesses (ou des conséquences en cas de conséquences non-monétaires) et de décharger son rôle de modélisation d'attitude vis-à-vis du risque sur la fonction de déformation des probabilités. Cette séparation des rôles se révèle très pratique dans de nombreux problèmes de décision.

Bien qu'il existe de nombreuses combinaisons sur les formes de la fonction  $u$  d'utilité et de la fonction  $\varphi$  de déformation des probabilités pour capturer l'attitude du décideur vis-à-vis du risque, nous supposons ici que la fonction d'utilité est linéaire pour mettre en évidence la manière dont la fonction  $\varphi$  joue sur le degré d'aversion au risque.

L'aversion faible pour le risque se traduit dans le modèle RDU par une fonction de déformation des probabilités  $\varphi$  telle que  $\forall p \in [0; 1], \varphi(p) \leq p$  (Chateauneuf et Cohen, 1994). Un décideur doté d'une fonction d'utilité linéaire est donc faiblement adverse au risque dès lors qu'il sous-estime ses probabilités de gains. De manière analogue, un décideur possède une faible attraction au risque dès lors qu'il surestime ses chances de gains (i.e.,  $\forall p \in [0; 1], \varphi(p) \geq p$ ). Quant à l'aversion forte pour le risque, elle se traduit dans le modèle RDU, par une fonction de déformation  $\varphi$  convexe (Chew et al., 1987). De façon similaire, un décideur est fortement attiré par le risque dès lors qu'il est doté d'une fonction de déformation des probabilités concave.

Dans la pratique, on utilise certaines formes remarquables pour la fonction  $\varphi$ . La plus simple, selon Quiggin (1993), est de la forme :  $\forall p \in [0; 1], \varphi(p) = p^\lambda$  avec  $\lambda \in \mathbb{R}_*^+$ . Notons que le cas particulier du modèle EU se retrouve en fixant  $\lambda$  à 1, et plus  $\lambda$  est proche de 0, plus le décideur est fortement attiré par le risque, et plus  $\lambda$  devient grand, plus le décideur affirme une aversion forte pour le risque.

Cependant la pratique a montré qu'il était difficile de modéliser l'attitude d'un décideur vis-à-vis du risque avec une fonction  $\varphi$  de cette forme. De nombreuses expériences ont mis en évidence que la plupart des décideurs sont dotés d'une fonction  $\varphi$  en S-inversé (i.e., possédant une partie concave sur les faibles probabilités, puis une partie convexe sur les fortes probabilités). Cette forme de fonction surévalue les probabilités des meilleures conséquences (ce que l'on appelle l'*effet de potentiel*, *potential effect*) et sous-évalue les

probabilités des mauvaises conséquences (ce que l'on appelle l'*effet de certitude*, *certainty effect*). Plusieurs équations de fonctions en S-inversé ont été proposées dans la littérature, la plus simple étant celle proposée par Karmarkar (1979) de la forme suivante :

$$\forall p \in [0; 1], \varphi(p) = \frac{p^\lambda}{p^\lambda + (1-p)^\lambda}$$

où  $\lambda \in ]0; 1]$ .

Notons qu'une fois de plus, le cas particulier du modèle EU se retrouve alors avec  $\lambda = 1$ , et plus  $\lambda$  est proche de 0, plus les effets de certitude et de potentiel du décideur s'intensifient. Cette forme impose une symétrie à la fonction  $\varphi$  et de nombreuses autres formes de fonctions en S-inversé ont été proposées pour briser cette symétrie. En effet il semblerait que l'effet de certitude soit généralement moins développé que l'effet de potentiel chez la plupart des décideurs. Nous ne nous étalerons pas sur ces diverses formes proposées et redirigeons le lecteur soucieux d'approfondir sa connaissance sur le sujet vers le livre de Quiggin (1993). Nous récapitulons dans la figure 1.6 les notions que nous avons brièvement exposées ici.

En ce qui concerne les autres modèles à transformation des probabilités, ils sont apparus dans la littérature après le modèle RDU. Le modèle d'Allais est identique au modèle RDU. La théorie duale est un cas particulier du modèle RDU pour lequel  $\forall x \in \mathbb{R}, u(x) = x$ . Par contre le modèle CPT présente un réel intérêt, puisqu'il généralise le modèle RDU dans les situations de gains et de pertes en même temps. En effet, si le modèle RDU se révèle très performant lorsqu'il faut décrire des préférences dans des situations de gains exclusivement ou des situations de pertes exclusivement, il est mis à mal dans les situations où l'on envisage de mélanger des mauvaises conséquences (conséquences négatives) avec des bonnes conséquences (conséquences positives). Le modèle CPT propose alors de généraliser le modèle RDU en introduisant une fonction de déformation des probabilités pour les bonnes conséquences et une seconde fonction de déformation des probabilités pour les mauvaises conséquences. Il introduit également, de manière identique, une fonction d'utilité pour les bonnes conséquences et une seconde fonction d'utilité pour les mauvaises conséquences. Bien que le modèle CPT soit une généralisation du modèle RDU, nous faisons ici le choix d'adopter le modèle RDU. Dans le but de simplifier les descriptions des mécanismes algorithmiques que nous verrons par la suite (pour lesquels la généralisation au modèle CPT reste possible). Par conséquent, nous nous placerons, par la suite, dans des situations de gains uniquement (i.e., dans lesquelles les utilités des conséquences seront toutes positives).

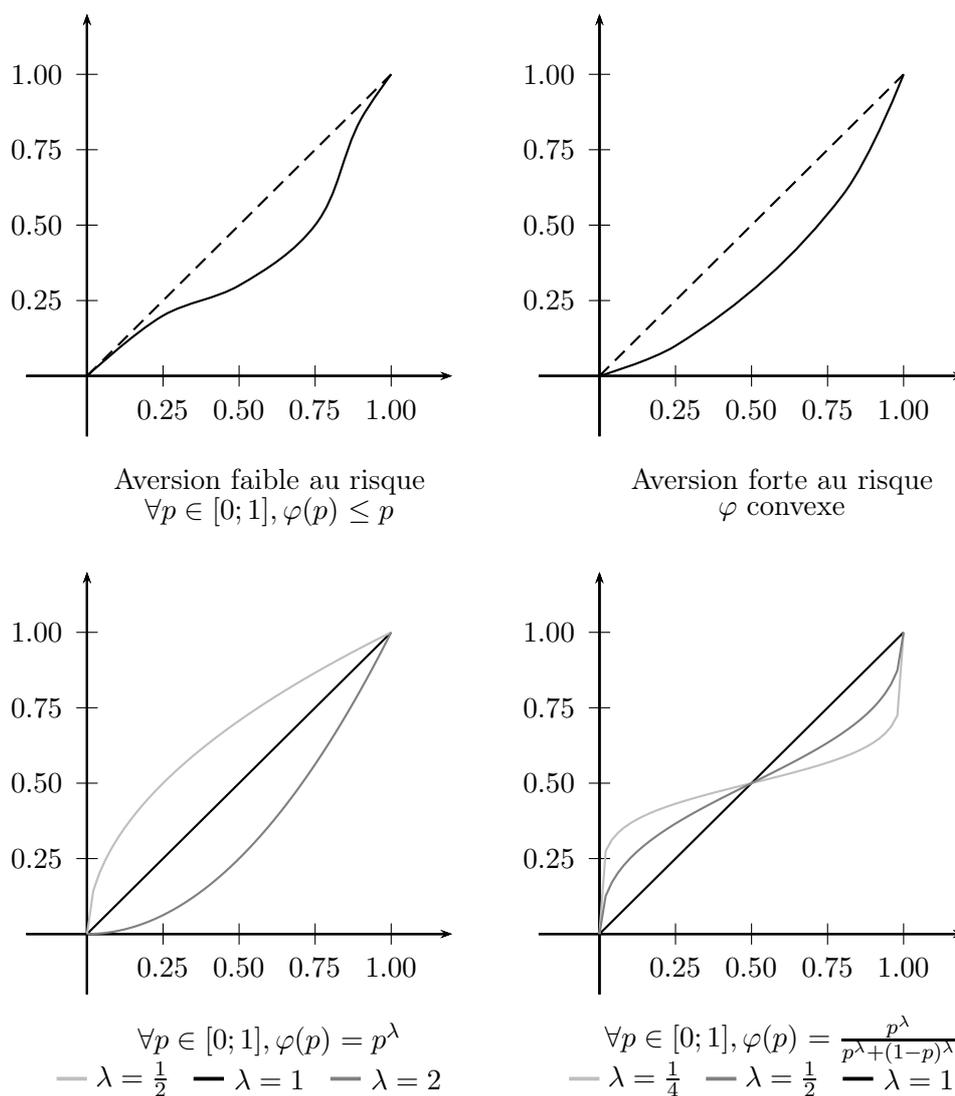


FIGURE 1.6 – Différentes formes remarquables de la fonction de déformation des probabilités dans le modèle RDU.

### Représentation graphique du modèle RDU

Nous achevons cette brève présentation du modèle RDU sur une représentation graphique très didactique proposée par Gayant (2001). Dans la figure 1.7, le quadrant nord-ouest représente la fonction  $\varphi$  (l'axe des abscisses, représentant les probabilités non déformées, est vertical et croît vers le nord, et l'axe des ordonnées, représentant les probabilités déformées, est horizontal et croît vers l'ouest). Le quadrant nord-est représente une fonction décumulative  $G_L$ . Le quadrant sud-est représente la fonction  $u$  (l'axe des abscisses, représentant les conséquences non déformées, est horizontal et croît vers l'est, et l'axe des ordonnées, représentant les conséquences déformées, est vertical et croît vers le sud).

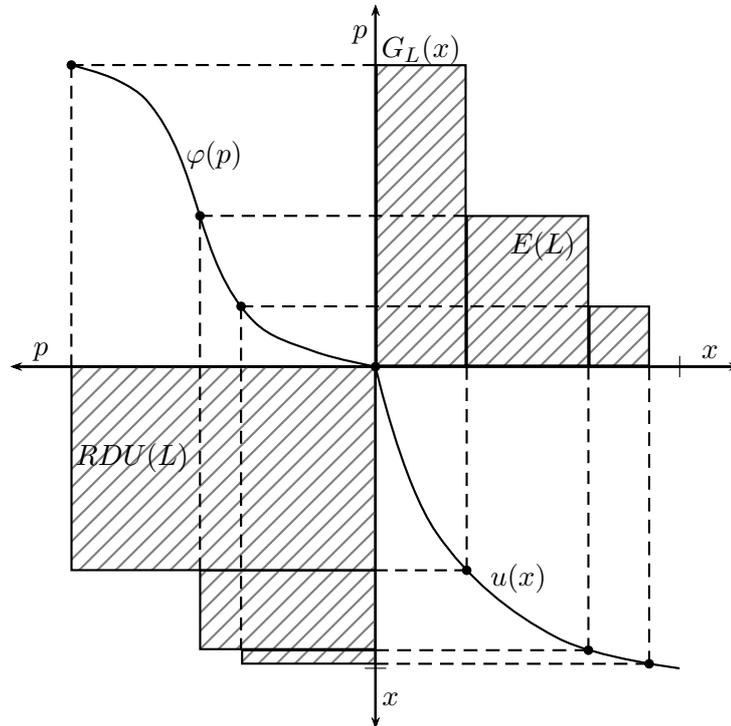


FIGURE 1.7 – Représentation graphique du modèle RDU (Gayant, 2001). La valeur de l'espérance de gain  $E(L)$  de la loterie  $L$  est égale à la surface hachurée dans le quadrant du nord-est et la valeur  $RDU(L)$  est égale à la surface hachurée dans le quadrant du sud-ouest.

La surface délimitée par la fonction décumulative  $G_L$  dans le quadrant nord-est est égale à l'espérance de gain  $E(L)$  de la loterie  $L$ . Dans le quadrant sud-ouest, la valeur  $RDU(L)$  apparaît après avoir déformé la surface du quadrant nord-est par les fonctions  $\varphi$  et  $u$  (ces dernières agissant comme des lentilles déformantes). En effet, si  $L = (x_1, p_1; x_2, p_2; x_3, p_3)$  avec  $x_1 < x_2 < x_3$ , l'aire de la surface hachurée du quadrant nord-ouest est égale à  $(p_1 + p_2 + p_3)x_1 + (p_2 + p_3)x_2 + (p_3)x_3 = E(L)$  et l'aire de la surface hachurée du quadrant sud-ouest est égale à  $\varphi(p_1 + p_2 + p_3)u(x_1) + \varphi(p_2 + p_3)u(x_2) + \varphi(p_3)u(x_3) = RDU(L)$ .

La complémentarité des fonctions  $\varphi$  et  $u$  apparaît très nettement dans cette représentation graphique et il semble alors difficilement envisageable de se passer de  $\varphi$  lorsque l'on désire décrire, de manière fidèle, des préférences dans le risque.

## 1.2 Décision dans l'incertain non risqué

Dans la pratique, il existe de nombreux problèmes pour lesquels on ne possède pas les distributions de probabilité objectives sur les états de la nature. C'est précisément le cadre de la décision dans l'incertain non risqué.

Pour raisonner dans ce cadre, on peut distinguer deux manières d'appréhender le problème. La première laisse le soin au décideur d'établir lui-même les distributions de probabilité de manière subjective. La seconde propose de raisonner, à l'aide de divers critères de décision adaptés, en exploitant la nature des données que l'on possède sur la vraisemblance des événements. La seconde méthode a le mérite de fonctionner dans tous les cas, en effet, il est souvent difficile pour un décideur d'avoir un avis sur la vraisemblance des états de la nature. Du point de vue de l'informatique, la première option a le mérite de se placer dans un cadre identique au cadre risqué, ce qui permet de réutiliser les algorithmes fonctionnant dans le risque dans le cas de l'incertain non risqué.

Dans la suite, nous présentons, sans trop nous étendre, cette première option. Après cela, nous détaillerons comment, selon la modélisation de l'incertitude, il est possible de se passer des probabilités subjectives.

### 1.2.1 Modèles à probabilités subjectives

Au début des années 1930, Ramsey et de Finetti suggèrent qu'en cas d'absence de distribution de probabilité objective, les décideurs établissent eux-même ces probabilités de manière subjective. Ces probabilités subjectives sont bien sûr positives, additives et l'événement certain doit posséder une probabilité égale à 1. Le modèle le plus simple est le modèle de l'*espérance subjective* (*Subjective Expected, SE*) (de Finetti, 1937). Dans ce modèle, les agents établissent une distribution  $Q$  de probabilité subjective sur les états de la nature et procèdent à un calcul d'espérance de gain pour évaluer la loterie  $L = (x_1, e_1; \dots; x_n, e_n)$  (i.e.,  $V \equiv SE$ ) :

$$SE(L) = \sum_{i=1}^n Q(e_i) \times x_i$$

A l'instar de l'espérance de gain dans le cas du risque, ce modèle tombe sous le coup du paradoxe de Saint-Pétersbourg. Pour pallier cela, suite à la proposition du modèle EU par von Neuman et Morgenstern (1947), Savage (1954) propose la généralisation du modèle  $SE$  en assignant une fonction d'utilité  $u$  au décideur et postule que les décideurs évaluent une loterie comme suit (i.e.,  $V \equiv SEU$ ) :

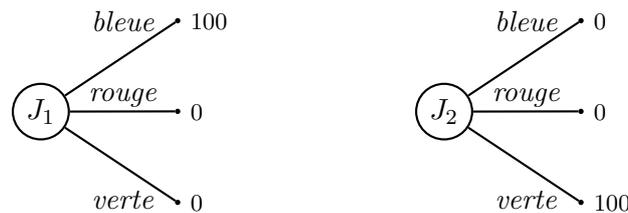
$$SEU(L) = \sum_{i=1}^n Q(e_i) \times u(x_i)$$

Ce modèle est connu sous le nom de l'*espérance subjective d'utilité* (*Subjective Expected Utility, SEU*). De façon semblable au modèle EU dans le risque, ce modèle repose sur un axiome qui impose un traitement linéaire des probabilités : le principe de la chose sûre, qui est le pendant de l'axiome d'indépendance dans le cadre de l'incertain. Malgré l'attrait sur le plan normatif du modèle *SEU*, le psychologue Daniel Ellsberg publie les résultats d'une expérience qui met en évidence l'incapacité du modèle *SEU* à modéliser les comportements dans l'incertain. Nous présentons cette expérience, bien connue sous le nom du *paradoxe d'Ellsberg* (Ellsberg, 1961), dans l'exemple 6.

**Exemple 6** *On place dans une urne opaque 30 boules bleues et 60 boules vertes ou rouges dans des proportions inconnues. Le nombre de boules vertes est donc inconnu de même que le nombre de boules rouges. On tire au hasard une boule dans cette urne (on suppose évidemment que chaque boule ici a autant de chance d'être tirée). On propose, dans un premier temps, de choisir entre deux jeux :*

- **Jeu 1** ( $J_1$ ) : *Si la boule tirée est de couleur bleue, alors vous gagnez 100 Euros.*
- **Jeu 2** ( $J_2$ ) : *Si la boule tirée est de couleur verte, alors vous gagnez 100 Euros.*

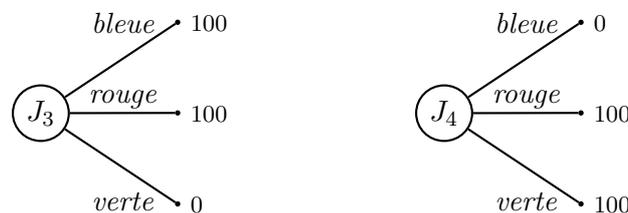
*Ce qui correspond à choisir entre les deux loteries suivantes :*



*Dans un second temps, on propose au joueur de choisir entre deux autres jeux :*

- **Jeu 3** ( $J_3$ ) : *Si la boule tirée est de couleur bleue ou rouge, alors vous gagnez 100 Euros.*
- **Jeu 4** ( $J_4$ ) : *Si la boule tirée est de couleur verte ou rouge, alors vous gagnez 100 Euros.*

*Ce qui correspond à choisir entre les deux loteries suivantes :*



Les résultats de l'expérience d'Ellsberg ont montré que la majorité des personnes qui se sont pliées à cette expérience ont préféré le jeu 1 au jeu 2 et ont préféré le jeu 4 au jeu 3. Ces préférences sont en contradiction avec les prescriptions du modèle *SEU*. En effet, si on note  $e_B$  l'événement « la boule tirée est bleue »,  $e_V$  l'événement « la boule tirée est verte » et  $e_R$  l'événement « la boule tirée est rouge »,  $Q$  la distribution de probabilité subjective établie par le décideur, alors les préférences du décideur pour le premier choix

se traduisent comme suit :

$$\begin{aligned}
J_1 \succ J_2 &\Leftrightarrow SEU(J_1) > SEU(J_2) \\
&\Leftrightarrow Q(e_V \cup e_R)u(0) + Q(e_B)u(100) > Q(e_B \cup e_R)u(0) + Q(e_V)u(100) \\
&\Leftrightarrow Q(e_B) > Q(e_V)
\end{aligned}$$

car la fonction  $u$  est unique à une transformation affine près (i.e., on peut donc supposer  $u(0) = 0$  sans perte de généralité). De la même manière les préférences du décideur pour le second choix se traduisent comme suit :

$$\begin{aligned}
J_4 \succ J_3 &\Leftrightarrow SEU(J_4) > SEU(J_3) \\
&\Leftrightarrow Q(e_B)u(0) + Q(e_V \cup e_R)u(100) > Q(e_V)u(0) + Q(e_B \cup e_R)u(100) \\
&\Leftrightarrow Q(e_V \cup e_R) > Q(e_B \cup e_R) \\
&\Leftrightarrow Q(e_V) + Q(e_R) > Q(e_B) + Q(e_R) \\
&\Leftrightarrow Q(e_V) > Q(e_B)
\end{aligned}$$

car la distribution de probabilité  $Q$  est additive (i.e.,  $Q(e_B \cup e_R) = Q(e_B) + Q(e_R)$  et  $Q(e_V \cup e_R) = Q(e_V) + Q(e_R)$ ).

Les deux inégalités obtenues sont contradictoires, ce qui rend non modélisables par le modèle SEU les préférences de la majorité des personnes soumises à cette expérience.

Le renversement des préférences, dans le paradoxe d'Ellsberg, s'explique simplement. Dans les deux choix, on propose au décideur de choisir entre un jeu où les probabilités sont parfaitement définies et un jeu où l'on ne connaît pas la probabilité de manière certaine.

En effet, le jeu 1 propose de gagner avec une probabilité de  $1/3$  la somme de 100 Euros tandis que le jeu 2 propose de gagner 100 Euros avec une probabilité imprécise comprise entre 0 et  $2/3$ . De même le jeu 3 propose de gagner 100 Euros avec une probabilité imprécise comprise entre  $1/3$  et 1 tandis que le jeu 4 propose de gagner 100 Euros avec la probabilité de  $2/3$ .

Ainsi l'expérience met en avant le fait que les décideurs préfèrent les situations où les probabilités sont parfaitement définies aux situations où les probabilités sont mal connues. Ce trait comportemental est nommé *l'aversion à l'ambiguïté*.

Afin de résoudre ce paradoxe, il est nécessaire d'établir un modèle qui puisse vérifier en même temps la propriété  $Q(e_B) > Q(e_V)$  et la propriété  $Q(e_B \cup e_R) < Q(e_V \cup e_R)$ . Ces deux propriétés sont contradictoires uniquement si l'on impose la propriété d'additivité sur le traitement des probabilités subjectives.

La problématique rencontrée ici est très similaire à la problématique rencontrée dans le cadre du risque. Une adaptation du modèle EU à la manière de Handa appliquée au modèle SEU (i.e., faire usage d'une fonction déformant les probabilités subjectives) se

heurterait au même problème rencontré dans le cadre du risque : la violation du respect de la dominance stochastique au premier ordre.

La parade se révèle être très proche de celle trouvée dans le cas du risque. Il ne faut pas établir les probabilités subjectives en les définissant uniquement sur les événements simples mais les définir également sur les unions des événements simples. C'est ce que propose de faire le modèle de l'*espérance d'utilité à la Choquet* (*Choquet Expected Utility, CEU*) proposé par Schmeidler (1989) (première version (Schmeidler, 1982)).

Avant de présenter ce modèle, nous allons définir la notion de *capacité de Choquet*. Une capacité de Choquet est une fonction de l'ensemble  $v : \mathcal{E}(\Omega) \rightarrow [0; 1]$  où  $\mathcal{E}(\Omega)$  est l'ensemble des sous-ensembles de  $\Omega$  (ou encore l'ensemble des événements de  $\Omega$ ). La fonction  $v$  doit respecter les propriétés suivantes :

- la croissance :  $\forall (A, B) \in \mathcal{E}(\Omega)^2, A \subseteq B \Rightarrow v(A) \leq v(B)$ ,
- les conditions aux bornes :  $v(\emptyset) = 0$  et  $v(\Omega) = 1$ .

Ces axiomes s'interprètent de manière très simple : la croissance correspond au fait que le degré de confiance accordé à un événement  $e$  ne peut être plus élevé que le degré de confiance accordé à un événement incluant l'événement  $e$  (qui est donc forcément plus vraisemblable), et les conditions aux bornes permettent d'assurer que le degré de confiance que l'on accorde à l'événement nul est minimal et que le degré de confiance que l'on accorde à l'événement certain est maximal.

Remarquons que si l'on ajoute à ces précédents axiomes, l'axiome d'additivité :

$$\forall (A, B) \in \mathcal{E}(\Omega)^2, A \cap B = \emptyset \Rightarrow v(A \cup B) = v(A) + v(B)$$

alors la capacité de Choquet  $v$  est bien une distribution de probabilité.

A partir d'une capacité de Choquet  $v$  on peut définir sa capacité complémentaire  $v^c$  comme suit :

$$\forall A \in \mathcal{E}(\Omega), v^c(A) = 1 - v(A^c)$$

où  $A^c$  est l'événement complémentaire à  $A$  (i.e.,  $A \cup A^c = \Omega$  et  $A \cap A^c = \emptyset$ ). Remarquons que si  $\forall A \in \mathcal{E}(\Omega), v^c(A) = v(A)$  alors la capacité de Choquet  $v$  est une distribution de probabilité.

Une capacité de Choquet est dite *convexe* ou encore *super-additive* si elle vérifie la propriété de *2-monotonie*<sup>2</sup> :

$$\forall (A, B) \in \mathcal{E}(\Omega)^2, v(A \cup B) + v(A \cap B) \geq v(A) + v(B)$$

---

2. On peut définir de manière plus générale une notion de *n-monotonie* de la manière suivante : une capacité  $v$  est *n-monotone*, avec  $n > 1$  et  $n \in \mathbb{N}$ , si et seulement si pour toute collection  $\mathcal{A} = \{A_i \in \mathcal{E}(\Omega) : i \in \mathbb{N}, 0 < i < n\}$  d'événements  $A_i$  on a :

$$v \left( \bigcup_{A_i \in \mathcal{A}} A_i \right) \geq \sum_{I \subseteq \mathcal{A}} (-1)^{|I|+1} v \left( \bigcap_{A_i \in I} A_i \right)$$

Ainsi, toute capacité de Choquet *n-monotone* est  $(n - 1)$ -monotone pour  $n > 1$  et  $n \in \mathbb{N}$ .

De la même manière on définit une capacité *concave* ou encore *sous-additive* en inversant le sens de l'inégalité. Remarquons qu'une distribution de probabilité est une capacité concave et convexe à la fois. Ainsi, les capacités super-additives modélisent l'idée de certitude (car  $v(A) + v(A^c) \leq 1$ ) et les capacités sous-additives modélisent l'idée de plausibilité (car  $v(A) + v(A^c) \geq 1$ ).

Enfin, on définit ce que l'on appelle le *cœur* d'une capacité  $v$  ou encore son *noyau* noté  $core(v)$  de la manière suivante :

$$core(v) = \{P \in \mathbb{P}_\Omega : \forall A \in \mathcal{E}(\Omega), v(A) \leq P(A)v(A^c)\}$$

où  $\mathbb{P}_\Omega$  est l'ensemble des distributions de probabilités définies sur  $\Omega$ . L'ensemble  $core(v)$  est donc l'ensemble des distributions de probabilités qui dominent sur tous les événements la capacité  $v$ .

Le modèle de l'espérance d'utilité à la Choquet propose alors d'évaluer une loterie  $L = (x_1, e_1; \dots; x_n, e_n)$  telle que  $x_1 < \dots < x_n$  comme suit (i.e.  $V \equiv CEU$ ) :

$$CEU(L) = u(x_1) + \sum_{i=2}^n v \left( \bigcup_{j=i}^n e_j \right) [u(x_i) - u(x_{i-1})]$$

où  $u$  est une fonction d'utilité et  $v$  une capacité de Choquet. Notons que, si  $v$  est additive (i.e., une distribution de probabilité), on retrouve alors le cas particulier du modèle SEU.

Cette forme fonctionnelle est quasiment la même que la forme fonctionnelle de RDU. En effet, si  $Q$  est une distribution de probabilité subjective sur les événements simples, alors il suffit de choisir chacune des capacités de Choquet de la manière suivante :

$$\forall 1 \leq i \leq n, v \left( \bigcup_{j=i}^n e_j \right) = \varphi \left( \sum_{j=i}^n Q(e_j) \right)$$

où  $\varphi$  est une fonction de déformation des probabilités.

En réalité, Waker (1990) a démontré que, sous l'hypothèse du respect de la dominance stochastique, le modèle CEU et RDU sont équivalents dans le cadre du risque. Ce qui fait du modèle CEU la généralisation du modèle RDU pour le cas de l'incertain (risqué et non risqué).

Bien entendu, le modèle CEU permet de résoudre le paradoxe d'Ellsberg en imposant aux capacités de Choquet, représentatives des préférences du décideur, de respecter :  $v(e_V \cup e_R) > v(e_B \cup e_R) > v(e_B) > v(e_V)$ .

Ainsi, dès lors que l'on s'autorise à attribuer des probabilités subjectives aux événements incertains il est possible de se retrouver dans un cadre quasiment identique au cas du risque.

Décision dans le risque	Décision dans l'incertain
Espérance de gain, $E$	Espérance subjective, $SE$ (de Finetti, 1937)
Espérance d'utilité, $EU$ (von Neuman et Morgenstern, 1947)	Espérance subjective d'utilité, $SEU$ (Savage, 1954)
Espérance d'utilité dépendant du rang, $RDU$ (Quiggin, 1982)	Espérance d'utilité à la Choquet, $CEU$ (Schmeidler, 1989)

TABLE 1.4 – Modèles de décision dans l'incertain et le risque

Dans le tableau 1.4, nous récapitulons les modèles cités précédemment dans cette section et les mettons en parallèle avec leurs homologues du risque. Cependant il n'est pas toujours possible de pouvoir établir des distributions de probabilités subjectives. D'une part parce qu'il n'est pas toujours facile (parfois même impossible) d'estimer la vraisemblance d'un événement, d'autre part, parce que le nombre d'événements à estimer est généralement très grand dans les problèmes que l'on est susceptible de rencontrer en informatique. De plus, l'établissement de données subjectives sur un problème rajoute de l'information subjective non présente initialement, et dans certains cas, on aimerait pouvoir raisonner uniquement avec les informations objectives du problème.

Dans les sections qui suivent nous allons justement voir comment se passer des probabilités subjectives selon la modélisation de l'incertitude envisagée.

### 1.2.2 Décision dans l'incertain total

Dans cette partie, nous nous plaçons dans le cas où le décideur ne possède aucune information sur la vraisemblance d'un événement.

A titre d'exemple, considérons la situation d'incertitude totale représentée par le tableau à gauche de la figure 1.8 page 38 : on a mis dans une urne opaque des boules rouges, des boules vertes et des boules bleues dans des proportions inconnues.

On note  $e_1$  l'événement « une boule rouge a été tirée »,  $e_2$  l'événement « une boule verte a été tirée » et  $e_3$  l'événement « une boule bleue a été tirée ». Le décideur doit alors choisir entre trois jeux, (décision  $d_1$  s'il choisit le premier,  $d_2$  s'il choisit le second et  $d_3$  s'il choisit le troisième) et une boule est ensuite tirée aléatoirement dans l'urne. Selon le jeu choisi et la couleur de la boule tirée, le décideur gagne alors une certaine somme.

L'ensemble des événements possibles  $E = \{e_1, e_2, e_3\}$  sont notées verticalement et l'ensemble des décisions  $D = \{d_1, d_2, d_3\}$  sont notés horizontalement. Le nombre à l'intersection de la ligne  $d_i$  et de la colonne  $e_j$  est l'utilité  $u(d_i, e_j)$  de la récompense reçue s'il choisit le jeu  $i$  et que l'événement  $e_j$  se réalise. Les loteries correspondantes aux différentes décisions  $d_i$  ont été représentées dans la partie droite de la figure 1.8 page 38.

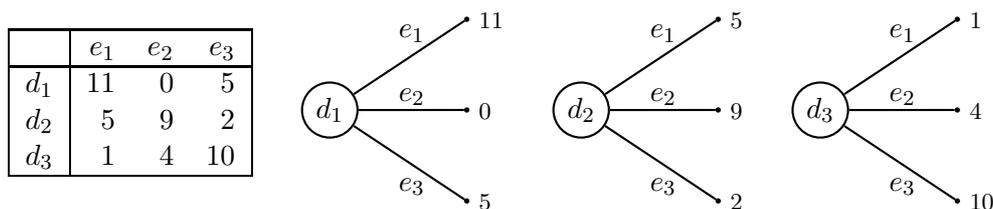


FIGURE 1.8 – Exemple de situation, sous incertitude totale, accompagné de la représentation des loteries correspondantes aux différentes décisions. Sur les loteries, ce sont les utilités qui ont été indiquées à la place des conséquences afin de respecter les valeurs du tableau.

Dans ce problème le décideur n'a donc aucune information sur la vraisemblance des événements  $e_1$ ,  $e_2$  et  $e_3$ , et doit tout de même choisir un jeu en dépit de son ignorance sur les vraisemblances. Nous allons présenter divers modèles de décision qui permettent justement de raisonner dans ce cadre précis.

### $\Gamma$ -Maximax et $\Gamma$ -Maximin

Parmi tous les critères proposés dans la littérature économique,  $\Gamma$ -Maximax et  $\Gamma$ -Maximin (appelé aussi *critère de Wald*) sont sûrement les plus « simples » que l'on puisse rencontrer.

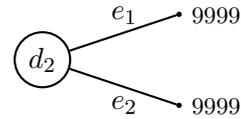
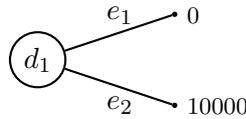
Le critère  $\Gamma$ -Maximax se propose de prendre la décision qui offre la meilleure conséquence. Plus formellement, on évalue une loterie  $L = (x_1, e_1; \dots; x_n, e_n)$  de la manière suivante ( $V \equiv \Gamma_{max}$ ) :

$$\Gamma_{max}(L) = \max_i u(x_i)$$

En guise d'exemple, plaçons-nous dans la situation du problème de la figure 1.8. Un décideur ayant des préférences qui suivent le critère  $\Gamma$ -Maximax préfère la décision  $d_1$  ( $\Gamma_{max}(d_1) = 11$ ) à la décision  $d_3$  ( $\Gamma_{max}(d_3) = 10$ ), qui est elle-même préférée à la décision  $d_2$  ( $\Gamma_{max}(d_2) = 9$ ).

Ainsi, le critère  $\Gamma$ -Maximax caractérise un décideur fortement optimiste. En effet, un décideur ayant des préférences qui suivent le critère  $\Gamma$ -Maximax ne tient compte que de la meilleure conséquence de chaque loterie pour effectuer ses choix. Ce critère, peu discriminant dans de nombreux cas, est très largement critiqué dans la littérature économique. En effet, la plupart des décideurs possèdent des préférences qui ne peuvent se décrire avec  $\Gamma$ -Maximax, comme le montre l'exemple qui suit. Considérons le problème de décision sous incertitude totale représenté ci-dessous :

	$e_1$	$e_2$
$d_1$	0	10000
$d_2$	9999	9999



Dans cette situation, la plupart des décideurs préfèrent la décision  $d_2$  à  $d_1$  alors que le critère  $\Gamma$ -Maximax impose de préférer la décision  $d_1$  à la décision  $d_2$ . Ces préférences sont donc incompatibles avec le critère  $\Gamma$ -Maximax.

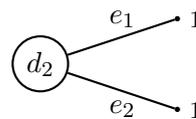
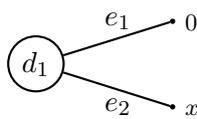
De manière symétrique, le critère  $\Gamma$ -Maximin se propose de prendre la décision qui maximise la pire conséquence possible. Plus formellement, on évalue une loterie  $L = (x_1, e_1; \dots; x_n, e_n)$  de la manière suivante ( $V \equiv \Gamma_{min}$ ) :

$$\Gamma_{min}(L) = \min_i u(x_i)$$

A titre d'exemple, replaçons nous dans le problème de la figure 1.8 page 38. Un décideur ayant des préférences qui suivent le critère  $\Gamma$ -Maximin, préfère la décision  $d_2$  ( $\Gamma_{min}(d_2) = 2$ ) à la décision  $d_3$  ( $\Gamma_{min}(d_3) = 1$ ), qui est elle-même préférée à la décision  $d_1$  ( $\Gamma_{min}(d_1) = 0$ ).

Ainsi, le critère  $\Gamma$ -Maximin caractérise un décideur fortement pessimiste. En effet, un décideur ayant des préférences qui suivent le critère  $\Gamma$ -Maximin ne tient compte que de la pire conséquence de chaque loterie pour effectuer ses choix. Encore une fois, ce critère se révèle être peu discriminant dans de nombreux cas, et la plupart des décideurs possèdent des préférences qui ne peuvent se décrire avec  $\Gamma$ -Maximin, comme le montre l'exemple de problème de décision sous incertitude totale suivant :

	$e_1$	$e_2$
$d_1$	0	$x$
$d_2$	1	1



Dans cette situation, la plupart des décideurs préfèrent la décision  $d_1$  à  $d_2$  dès lors que  $x$  devient très grand, alors que le critère  $\Gamma$ -Maximin impose de préférer la décision  $d_2$  à la décision  $d_1$ . Ces préférences sont donc incompatibles avec le critère  $\Gamma$ -Maximin.

### Minimax-Regret

Dans l'optique d'améliorer le critère  $\Gamma$ -Maximin, Savage (1951) fait l'hypothèse que les préférences d'un décideur reposent sur une notion de « regret ». Dans ce modèle, on calcule une *matrice des regrets* à partir de la matrice des conséquences. Le regret s'obtient, pour chaque utilité  $u(d_i, e_j)$ , en effectuant la différence entre l'utilité de la

meilleure conséquence pour l'événement  $e_j$  et l'utilité de la conséquence  $u(d_i, e_j)$ . Plus formellement, on calcule les coefficients de la matrice de regrets comme suit :

$$r_{i,j} = \max_k u(d_k, e_j) - u(d_i, e_j)$$

puis on évalue le regret  $r$  maximal d'une décision  $d_i$  avec la formule :

$$r(d_i) = \max_j r_{i,j}$$

Savage postule alors qu'un décideur, entre deux décisions  $d_1$  et  $d_2$ , préférera toujours celle dont le regret maximal est minimal (i.e.,  $d_1 \succ d_2 \Leftrightarrow r(d_1) < r(d_2)$ ).

En guise d'exemple, plaçons nous dans le problème de décision sous incertitude totale de la figure 1.8 page 38. La matrice des regrets est alors la suivante :

$r_{i,j}$	$e_1$	$e_2$	$e_3$	$r(d_i)$
$d_1$	0	9	5	9
$d_2$	6	0	8	8
$d_3$	10	5	0	10

Dans la dernière colonne, nous avons indiqué les regrets maximaux associés aux différentes décisions. Ainsi, un décideur ayant des préférences qui suivent le critère du Minimax-Regret préfère la décision  $d_2$  (de regret maximal 8) à la décision  $d_1$  (de regret maximal 9), qui est elle-même préférée à la décision  $d_3$  (de regret maximal 10).

Ce modèle améliore le modèle  $\Gamma$ -Maximin dans le sens où il permet maintenant de contourner la critique faite à l'encontre du critère  $\Gamma$ -Maximin, tout en conservant son caractère pessimiste (le décideur ne prend en compte que le pire regret de chaque décision pour établir ses préférences). En effet, si on considère maintenant le problème de décision sous incertitude totale suivant et sa matrice de regret (pour  $x > 1$  et très grand) :

	$e_1$	$e_2$
$d_1$	0	$x$
$d_2$	1	1

$r_{i,j}$	$e_1$	$e_2$	$r(d_i)$
$d_1$	1	0	1
$d_2$	0	$x - 1$	$x - 1$

alors le décideur préfère maintenant  $d_1$  à  $d_2$  contrairement aux préférences prédites par  $\Gamma$ -Maximin.

Néanmoins, très rapidement, de nombreuses lacunes ont été pointées par Chernoff (1954) qui anéantissent tout espoir de construire des préférences solides (i.e., des préférences indépendantes du problème dans lequel on se trouve) à partir du critère du Minimax-Regret.

La première critique émise concerne la notion de regret qui est mal définie. En effet, rien ne laisse à supposer que le regret de passer d'un état offrant une utilité de 5 à un état offrant une utilité de 3 soit équivalent au regret de passer d'un état offrant une utilité de 30 à un état offrant une utilité de 28.

Outre la naïveté de cette notion de regret, Chernoff fait remarquer que toutes les critiques que l'on pouvait faire au critère  $\Gamma$ -Maximin peuvent être facilement transposées au critère du Minimax-Regret en s'appliquant à la matrice des regrets.

Enfin, il émet une dernière critique bien plus problématique, portant sur l'ambiguïté des préférences du décideur sur les différentes décisions. Illustrons cette critique à l'aide de l'exemple suivant : considérons le problème de décision sous incertitude totale de la figure 1.8 page 38. Comme vu précédemment, sous le critère du Minimax-Regret, les préférences du décideur sont :  $d_2 \succ d_1 \succ d_3$ . Ainsi, la décision  $d_2$  est préférée à la décision  $d_3$ . Observons maintenant comment évoluent ces préférences lorsque l'on retire du problème la décision  $d_1$ . La matrice des regrets est alors la suivante :

$r_{i,j}$	$e_1$	$e_2$	$e_3$	$r(d_i)$
$d_2$	0	0	8	8
$d_3$	4	5	0	5

Ainsi, après avoir retiré la décision  $d_1$  du problème, le décideur préfère alors la décision  $d_3$  à la décision  $d_2$ . L'optimalité de la décision  $d_2$  dans le problème initial est alors remise en cause lorsque l'on retire la décision  $d_1$  : on dit que le critère n'est pas indépendant vis-à-vis d'une tierce alternative. Cette instabilité, rendant ambiguë la décision optimale, est un phénomène non souhaitable lorsque l'on cherche à décrire les préférences d'un décideur et cela aussi bien en économie que dans les problèmes rencontrés en informatique.

Notons qu'il est tout à fait possible d'envisager le critère du Maximax-Regret (améliorant le critère  $\Gamma$ -Maximax) qui serait le pendant du critère du Minimax-Regret, mais en s'appuyant sur le caractère optimiste du décideur. Bien entendu, les critiques émises à l'encontre du critère du Minimax-Regret resteraient valables pour le critère du Maximax-Regret.

### Critère de Laplace

Le critère de Laplace, aussi appelé *critère de la raison insuffisante*, postule qu'en cas d'incertitude totale sur les événements, il faut supposer a priori l'équiprobabilité des événements.

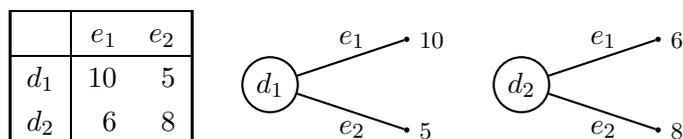
Plus formellement, on évalue une loterie  $L = (x_1, e_1; \dots; x_n, e_n)$  avec le critère de Laplace de la manière suivante (i.e.,  $V \equiv Lap$ ) :

$$Lap(L) = \frac{\sum_{i=1}^n u(x_i)}{n}$$

Notons que ce critère est un cas particulier du modèle *SEU* pour lequel on aurait choisi une distribution de probabilité subjective qui associe une probabilité identique à chaque conséquence. En guise d'illustration, dans le problème de décision sous incertitude totale de la figure 1.8 page 38, un décideur ayant des préférences qui suivent le critère de Laplace est indifférent entre la décision  $d_1$  ( $Lap(d_1) = 16/3$ ) et la décision  $d_2$  ( $Lap(d_2) = 16/3$ ) mais il préfère  $d_1$  et  $d_2$  à la décision  $d_3$  ( $Lap(d_3) = 5$ ).

Ce critère se heurte cependant à la difficulté de l'élicitation des états de la nature possibles, et au fait qu'une modélisation différente d'un même problème peut renverser totalement les préférences de l'individu.

Imaginons le jeu suivant : dans une urne opaque sont placées des boules rouges, vertes ou bleues dans des proportions inconnues. Soit  $e_1$  l'événement « la boule tirée est rouge » et  $e_2$  l'événement « la boule tirée n'est pas rouge ». On propose alors au décideur de choisir entre deux loteries :

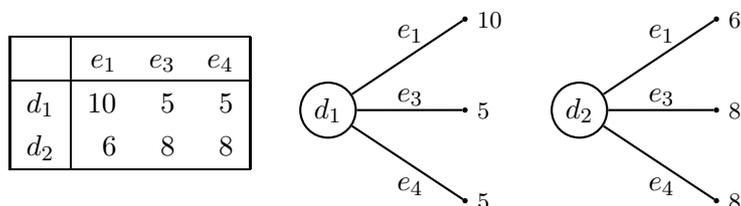


Selon le critère de Laplace le décideur préfère alors la loterie  $d_1$  ( $Lap(d_1) = 7.5$ ) à la loterie  $d_2$  ( $Lap(d_2) = 7$ ).

Maintenant, remplaçons l'événement  $e_2$  par les deux événements suivants :

- $e_3$  : « la boule tirée est verte »
- $e_4$  : « la boule tirée est bleue »

Le décideur se retrouve alors dans la situation de choix suivante :



Cette fois le décideur préfère la loterie  $d_2$  ( $Lap(d_2) = 22/3$ ) à la loterie  $d_1$  ( $Lap(d_1) = 20/3$ ).

Ainsi, le même problème présenté différemment peut totalement changer les préférences du décideur avec le critère de Laplace. Ce phénomène n'est pas désiré lorsque l'on

cherche à décrire les préférences d'un décideur : la représentation du problème ne doit pas influencer les préférences si on veut se prémunir de toute manipulation sur le décideur.

### Critère de Hurwicz

Puisque  $\Gamma$ -Maximax et  $\Gamma$ -Maximin sont des critères caractérisant des comportements pessimistes ou optimistes trop extrêmes, Hurwicz (1951) propose de les généraliser en réalisant une somme pondérée entre la meilleure conséquence et la pire conséquence. Formellement, une loterie  $L = (x_1, e_1; \dots; x_n, e_n)$  s'évalue comme suit (i.e.,  $V \equiv Hz$ ) :

$$Hz(L) = \alpha \min_i u(x_i) + (1 - \alpha) \max_i u(x_i)$$

où  $\alpha$  est un réel compris entre 0 et 1 permettant de mesurer le degré de pessimisme du décideur (i.e., plus  $\alpha$  est proche de 1, plus le décideur tient compte de la pire conséquence, et plus  $\alpha$  est proche de 0, plus le décideur tient compte de la meilleure conséquence). Notons que le cas  $\alpha = 0$  correspond au cas particulier du critère  $\Gamma$ -Maximax et que le cas  $\alpha = 1$  correspond au cas particulier du critère  $\Gamma$ -Maximin.

En guise d'illustration, considérons le problème de décision sous incertitude totale de la figure 1.8 page 38. Un décideur ayant des préférences qui suivent le critère de Hurwicz (fixons arbitrairement  $\alpha$  à 0.3) préfère la décision  $d_1$  ( $Hz(d_1) = 7.7$ ) à la décision  $d_3$  ( $Hz(d_3) = 7.3$ ), qui est elle-même préférée à la décision  $d_2$  ( $Hz(d_2) = 6.9$ ).

Ce modèle est souvent critiqué du fait qu'il ne prend pas en compte les conséquences non-extrêmes : considérons le problème de décision sous incertitude totale suivant :

	$e_1$	$e_2$	$e_3$	...	$e_i$	...	$e_{100}$
$d_1$	0	1	1	...	1	...	1
$d_2$	1	0	0	...	0	...	0

Un décideur ayant des préférences qui suivent le critère de Hurwicz est indifférent entre la décision  $d_1$  et la décision  $d_2$  quelle que soit la valeur de  $\alpha$ , alors qu'on a tendance à vouloir préférer strictement la décision  $d_2$  à la décision  $d_1$ . Mais la décision  $d_2$  est-elle vraiment meilleure que la décision  $d_1$  ?

En situation d'incertitude totale, puisque nous ne possédons aucune information sur la vraisemblance d'un événement, le problème précédent est équivalent au problème suivant :

	$e'_1$	$e'_2$
$d_1$	0	1
$d_2$	1	0

Il est alors difficilement concevable de ne pas être indifférent entre prendre la décision  $d_1$  et prendre la décision  $d_2$ .

Concernant le choix de la valeur du paramètre  $\alpha$  définissant le degré de pessimisme d'un décideur, la méthode classique consiste à déterminer la valeur de  $x$  pour laquelle le décideur est indifférent entre prendre la décision  $d_1$  et prendre la décision  $d_2$  dans le problème suivant :

	$e_1$	$e_2$
$d_1$	0	1
$d_2$	$x$	$x$

Dès lors que la valeur de  $x$  a été fixée, il suffit alors de prendre  $\alpha = 1 - x$ . L'estimation du paramètre  $\alpha$  pour un décideur donné se fait donc aisément.

Il existe d'autres critères de décision en cas d'incertitude totale : nous nous sommes cantonnés ici aux plus cités dans la littérature.

Parmi ceux que nous n'avons pas présenté ici, nous pouvons citer le *critère moyenne-variabilité* (Jokung-Nguena, 1998) qui impose la condition supplémentaire, au critère de Laplace, que l'écart entre les conséquences extrêmes soit minimal. C'est-à-dire qu'une décision  $d_1$  est préférée à une décision  $d_2$  si et seulement si elle est préférée au sens du critère de Laplace et que l'écart entre les conséquences extrêmes offertes par la décision  $d_1$  soit plus petit que l'écart entre les conséquences extrêmes de la décision  $d_2$ . Ce critère est alors incapable de comparer certains couples de loteries et n'est donc pas au même titre que les critères vus précédemment une fonction d'évaluation.

On peut également citer le critère de Starr (1966) qui repose sur une hypothèse d'équiprobabilité sur l'ensemble des mesures de probabilité que l'on peut définir sur les conséquences. Ce dernier est sensible, en plus des critiques qui lui sont spécifiques, aux mêmes critiques que celles émises à l'encontre du critère de Laplace.

Nous récapitulons dans le tableau 1.5 page 45 les préférences induites pour chaque critère de décision lorsque l'on se place dans le problème de décision sous incertitude de la figure 1.8 page 38. On constate alors l'existence de biais importants selon le choix du critère de décision (i.e., l'ordre de préférence entre les loteries est fortement dépendant du critère de décision adopté).

Notons au passage que les critères  $\Gamma$ -Maximax et  $\Gamma$ -Maximin nécessitent une information plus aisément élicitable que les autres critères. En effet, une information de nature ordinale suffit pour ces deux derniers, tandis que les autres critères ont besoin d'une mesure cardinale de l'utilité. Cependant, nous nous intéresserons ici uniquement à des problèmes dont les conséquences peuvent être représentées par des mesures cardinales.

Critères	Préférences
$\Gamma$ -Maximax	$d_1 \succ d_3 \succ d_2$
$\Gamma$ -Maximin	$d_2 \succ d_3 \succ d_1$
Maximin-Regret	$d_2 \succ d_1 \succ d_3$
Critère de Laplace	$d_1 \succ d_2 \succ d_3$
Critère d'Hurwicz ( $\alpha = 0.3$ )	$d_1 \succ d_3 \succ d_2$

TABLE 1.5 – Critères de décision pour le raisonnement sous incertitude totale, et préférences induites pour le problème de décision de la figure 1.8.

Bien que le critère de Hurwicz semble être un bon candidat pour la description des préférences en situation d'incertitude totale (de par sa simplicité et sa capacité à s'adapter aux préférences du décideur par l'utilisation du paramètre  $\alpha$ ), nous allons appuyer cette conviction par des résultats axiomatiques bien connus en économie. Pour cela nous présentons ici un ensemble d'axiomes jugés souhaitables pour décrire des préférences en situation d'incertitude totale. Cette axiomatique, initialement proposée par Milnor (1954), a été reformulée par Luce et Raiffa (1957) à la suite de Chernoff (1954). Nous avons choisi ici de présenter l'axiomatique reformulée par Luce et Raiffa.

#### Liste des axiomes encadrant les préférences (Luce et Raiffa, 1957)

**Axiome 1 :** *Tout problème de décision sous incertitude totale possède au moins une solution.*

**Axiome 2 :** *L'ensemble des meilleurs choix d'un problème de décision sous incertitude totale est indifférent à toute transformation affine de la fonction d'utilité.*

**Axiome 3 :** *L'ensemble des meilleurs choix d'un problème de décision sous incertitude totale n'est pas sensible à un changement de dénomination des événements.*

**Axiome 4 :** *Si  $d_1$  appartient à l'ensemble des meilleurs choix et  $d_2 \succsim d_1$ , alors  $d_2$  appartient à l'ensemble des meilleurs choix.*

Les axiomes 1, 2, 3 et 4 sont des axiomes d'ordre technique. Tous les critères que nous avons présentés respectent ces quatre axiomes.

**Axiome 5 :** *L'ensemble des meilleurs choix ne contient pas de décisions  $d$  dominées. C'est à dire qu'il n'existe pas de décision  $d'$  vérifiant  $u(d', e_i) \geq u(d, e_i)$  pour tout  $i$ , avec une inégalité stricte au moins pour une valeur de  $i$ .*

L'axiome 5 permet d'assurer la rationalité du décideur. Elle est le pendant de la dominance stochastique du premier ordre dans le cadre du risque. Cette propriété est bien sûr fortement désirable. Notons que cet axiome est violé par  $\Gamma$ -Maximax,  $\Gamma$ -Maximin et le critère de Hurwicz. En effet, dans le problème de décision sous incertitude suivant, toutes les décisions sont optimales au sens de  $\Gamma$ -Maximin,  $\Gamma$ -Maximax ou du critère de Hurwicz :

	$e_1$	$e_2$	$e_3$
$d_1$	0	0	2
$d_2$	0	1	2

Pourtant la décision  $d_1$  est dominée par la décision  $d_2$ . Cependant, il existe toujours une décision non dominée ayant une valeur optimale au sens de ces critères. Dans la pratique, on retire de l'ensemble des meilleurs choix les décisions dominées. Dans l'exemple précédent on aurait alors retiré de l'ensemble des meilleurs choix la décision  $d_1$ . Ainsi l'existence d'au moins une solution non dominée dans l'ensemble des meilleurs choix est une condition suffisante pour justifier le côté rationnel d'un critère de décision.

**Axiome 6 :** *L'ajout au problème de nouvelles décisions non optimales n'influence pas l'ensemble des meilleurs choix. En d'autres termes, après l'ajout d'une nouvelle décision non-optimale au problème, une décision anciennement non-optimale ne peut devenir optimale et une décision anciennement optimale ne peut devenir non-optimale.*

L'axiome 6 permet de s'assurer que le critère est indépendant vis-à-vis d'une tierce alternative. Cela permet de se prémunir des diverses manipulations qui pourraient être exercées sur un décideur par l'omission ou l'ajout de choix factices. Le critère du Maximax-Regret n'est pas compatible avec cet axiome comme nous l'avons vu précédemment.

**Axiome 7 :** *L'ajout d'une constante à toutes les utilités des conséquences associées à un même état du monde ne modifie pas l'ensemble des meilleurs choix.*

**Axiome 8 :** *L'ensemble des meilleurs choix est convexe.*

Les axiomes 7 et 8 ne sont pas respectés par le critère de Hurwicz,  $\Gamma$ -Maximin (pour l'axiome 7) et  $\Gamma$ -Maximax (pour les axiomes 7 et 8). L'axiome 7 a été l'objet de nombreux débats dans la littérature dû à son côté peu intuitif et son statut d'axiome est discutable. Quant à l'axiome 8, il est violé dans certaines situations. Illustrons l'une d'elles en considérant le problème de décision sous incertitude totale suivant :

	$e_1$	$e_2$	$e_3$
$d_1$	1	0	0
$d_2$	0	1	0
$d_3$	0.5	0.5	0

Dans ce problème, les décisions  $d_1$  et  $d_2$  sont appréciées dès lors que l'on est optimiste (dû à l'attraction de la conséquence de valeur 1) et seront donc préférées à la décisions  $d_3$  bien qu'elle soit le résultat d'un mélange convexe (par 0.5) des deux décisions précédentes. Ainsi, l'axiome 8 fournit une propriété normative qui n'est a priori pas compatible avec les comportements observés.

**Axiome 9 :** *L'ensemble des meilleurs choix d'un problème de décision sous incertitude totale ne dépend pas de la manière dont les états du monde sont étiquetés.*

**Axiome 10 :** *Retrancher à un problème de décision sous incertitude totale une colonne répétée (c'est-à-dire un état du monde qui, pour chaque acte, entraîne les mêmes conséquences qu'un autre) ne change pas l'ensemble des meilleurs choix.*

Les axiomes 9 et 10 sont des axiomes techniques qui permettent de caractériser la notion d'incertitude totale : l'axiome 9 permet de faire abstraction du contexte du problème et l'axiome 10 impose l'ignorance totale sur la vraisemblance d'un événement, en considérant comme équivalents ces deux problèmes par exemple :

	$e_1$	$e_2$
$d_1$	2	1
$d_2$	8	0

	$e_1$	$e_3$	$e_4$	$e_5$
$d_1$	2	1	1	1
$d_2$	8	0	0	0

En effet, lorsque l'on se trouve en situation d'incertitude totale, il n'y a aucune raison de supposer que la réalisation de l'événement  $e_3 \cup e_4 \cup e_5$  soit plus vraisemblable que la réalisation de l'événement  $e_2$ .

Luce et Raiffa (1957) indiquent que ces dix axiomes ne sont pas compatibles entre eux et qu'il est nécessaire d'en retirer au moins un pour créer un ensemble d'axiomes compatibles. Ils proposent alors de retirer l'axiome 7, dont la validité sur le plan normatif est discutable. Seul le critère  $\Gamma$ -Maximin est compatible avec l'ensemble des axiomes restants (sous l'hypothèse que les décisions dominées soient retirées de l'ensemble des meilleurs choix). Afin de jouir d'un plus grand éventail de comportements décisionnels envisageables, il est nécessaire de se passer d'au moins un second axiome et de tourner

le dos à une seconde propriété normative. Puisque l'axiome 8 est violé dans certaines situations, il semble alors raisonnable de le retirer afin d'accroître l'efficacité sur le plan descriptif du critère utilisé. Le critère de Hurwicz (et donc  $\Gamma$ -Maximax également) devient alors compatible avec les axiomes restants. Enfin, Arrow (1953) démontre le théorème suivant : si un critère satisfait les axiomes 1, 3, 4, 6, 9 et 10 alors il ne prend en compte que les conséquences extrêmes. Dès lors, le critère de Hurwicz apparaît comme un excellent candidat pour décrire les préférences d'un décideur en situation d'incertitude totale tout en respectant un ensemble d'axiomes raisonnables qui garantissent de bonnes propriétés décisionnelles.

### 1.2.3 Décision dans le risque imprécis

Lorsque plusieurs experts ont des points de vue divergents ou quand certaines données empiriques sont manquantes, il n'est pas toujours aisé d'établir des probabilités précises pour chacun des événements. Une manière naturelle de contourner cette difficulté est d'utiliser des intervalles de probabilités pour évaluer la vraisemblance des événements en lieu et place des valeurs scalaires de probabilités. On parle alors de *probabilités intervalles*.

Au début des années 1990, plusieurs modélisations des probabilités intervalles ont été proposées (Walley, 1991; Kuznetsov, 1991; Weichselberger, 1999). Nous considérons ici que la probabilité intervalle  $K(e)$  d'un événement élémentaire  $e \subseteq \Omega$  est l'intervalle  $K(e) = [\underline{K}(e); \overline{K}(e)]$  où  $\underline{K}(e)$  et  $\overline{K}(e)$  sont les probabilités inférieures et supérieures de l'événement  $e$ , ou plus généralement, sont les estimations minimales et maximales de la vraisemblance de l'événement  $e$  (Smith, 1961; Williams, 2007; Walley, 1991).

Soit  $K$  une distribution de probabilité intervalle sur les événements élémentaires  $e$  définis sur l'ensemble  $\Omega$  des états de la nature, on note  $\mathcal{P}_K$  l'ensemble des distributions de probabilités compatibles avec  $K$ . Plus formellement, on a :

$$\mathcal{P}_K = \{P \in \mathbb{P}_\Omega : \forall e \in \Omega, P(\{e\}) \in K(\{e\})\}$$

où  $\mathbb{P}_\Omega$  est l'ensemble des distributions de probabilités définies sur  $\Omega$ . Une condition nécessaire et suffisante pour assurer que  $\mathcal{P}_K$  ne soit pas vide a été donnée par de Campos et al. (1994) :

$$\sum_{e \in \Omega} \underline{K}(\{e\}) \leq 1 \leq \sum_{e \in \Omega} \overline{K}(\{e\})$$

Ils proposent également une condition nécessaire et suffisante pour que toutes les valeurs des intervalles, induits par les probabilités intervalles, soient atteintes par au moins une distribution de probabilité  $P \in \mathcal{P}_K$  :

$$\forall e \in \Omega, \overline{K}(\{e\}) + \sum_{f \in \Omega \setminus \{e\}} \underline{K}(\{f\}) \leq 1 \text{ et } \underline{K}(\{e\}) + \sum_{f \in \Omega \setminus \{e\}} \overline{K}(\{f\}) \geq 1$$

Cette condition permet d'assurer que les intervalles décrivant  $\mathcal{P}_K$  sont de taille minimale (on dit alors que l'hypothèse d'*accessibilité* est vérifiée). Notons que, à partir d'une distribution de probabilité intervalle  $K$  pour laquelle l'hypothèse d'accessibilité n'est pas vérifiée, il est toujours possible de construire une distribution de probabilité intervalle  $K'$  caractérisant un ensemble  $\mathcal{P}_{K'} = \mathcal{P}_K$  de distribution de probabilité et pour laquelle l'hypothèse d'accessibilité est vérifiée. Pour ce faire, il suffit de prendre  $K'$  telle que :

$$\forall e \in \Omega, \underline{K}'(\{e\}) = \inf_{P \in \mathcal{P}_K} P(\{e\}) \text{ et } \overline{K}'(\{e\}) = \sup_{P \in \mathcal{P}_K} P(\{e\})$$

Par la suite nous considérerons toujours que la distribution de probabilité intervalle  $K$  définit un ensemble  $\mathcal{P}_K$  non vide et qu'elle vérifie l'hypothèse d'accessibilité. Il est également possible de déterminer les intervalles sur les événements non élémentaires à partir de  $\mathcal{P}_K$  de la manière suivante :

$$\forall A \subseteq \Omega, K(A) = [\inf_{P \in \mathcal{P}_K} P(A); \sup_{P \in \mathcal{P}_K} P(A)]$$

Sous de telles conditions, de Campos et al. (1994) ont montré que la probabilité inférieure  $\underline{K}$  définie par  $\{\underline{K}(A) : \forall A \subseteq \Omega\}$  d'une distribution de probabilité intervalle  $K$ , est une capacité de Choquet convexe (i.e., 2-monotone).

Dans une situation de risque imprécis, le décideur est amené à réaliser des choix entre des loteries dont la vraisemblance des événements est représentée par une distribution de probabilité intervalle. Nous noterons  $EU_P(L)$  l'espérance d'utilité, de la loterie  $L = (x_1, K(e_1); \dots; x_n, K(e_n))$ , pour la distribution de probabilité  $P \in \mathcal{P}_K$ .

En guise d'exemple, considérons le problème de décision dans le risque imprécis induit par le choix entre la loterie  $L_1 = (40, K(e_1); 12, K(e_2); 3, K(e_3))$  et la loterie  $L_2 = (4, K(e_1); 23, K(e_2); 15, K(e_3))$  avec  $K(e_1) = [0.1; 0.4]$ ,  $K(e_2) = [0.3; 0.5]$  et  $K(e_3) = [0.2; 0.4]$ . Ces loteries sont représentées dans la figure 1.9 page 50. Le tableau à droite de la figure résume les données du problème de décision. On constate bien que les probabilités intervalles indiquées admettent au moins une distribution  $P$  de probabilité compatible, par exemple :  $P(e_1) = 0.4 \in K(e_1)$ ,  $P(e_2) = 0.3 \in K(e_2)$  et  $P(e_3) = 0.3 \in K(e_3)$ . Sous l'hypothèse que  $\forall x \in \mathbb{R}, u(x) = x$ , on a alors :

- $EU_P(L_1) = 40P(e_1) + 12P(e_2) + 3P(e_3) = 40 \times 0.4 + 12 \times 0.3 + 3 \times 0.3 = 20.5$
- $EU_P(L_2) = 4P(e_1) + 23P(e_2) + 15P(e_3) = 4 \times 0.4 + 23 \times 0.3 + 15 \times 0.3 = 13$

Notons que si l'on assigne, à chaque événement, la probabilité intervalle  $[0; 1]$ , on se retrouve dans un cadre d'incertitude totale. Plusieurs critères ont été proposés dans la littérature pour comparer des loteries. La plupart sont des adaptations de critères proposés dans le cadre de l'incertitude totale. Nous allons passer en revue les principaux critères de décision que l'on rencontre dans la littérature permettant de raisonner dans

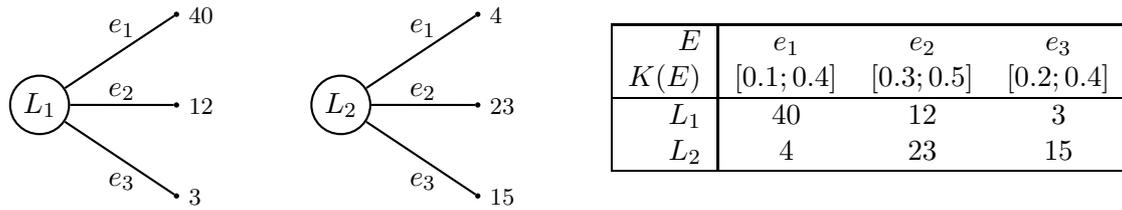


FIGURE 1.9 – Exemple de situation sous risque imprécis.

une situation de risque imprécis.

### $\Gamma$ -Maximax et $\Gamma$ -Maximin

Le critère  $\Gamma$ -Maximax impose qu'une loterie  $L_1$  soit préférée à une loterie  $L_2$  si et seulement si il existe une distribution de probabilité  $P$  compatible avec les probabilités intervalles, telle que l'espérance d'utilité de  $L_1$  soit au moins aussi bonne que l'espérance d'utilité de  $L_2$  pour toute distribution de probabilité compatible avec les probabilités intervalles.

De manière plus formelle, on évalue la loterie  $L = (x_1, K(e_1); \dots; x_n, K(e_n))$  comme suit (i.e.,  $V \equiv \Gamma_{max}$ ) :

$$\Gamma_{max}(L) = \max_{P \in \mathcal{P}_K} EU_P(L)$$

En guise d'exemple, considérons les loteries de la figure 1.9. Les valeurs assignées aux loteries, par le critère  $\Gamma$ -Maximax, sont les suivantes (avec  $\forall x \in \mathbb{R}, u(x) = x$ ) :

- $\Gamma_{max}(L_1) = EU_{P_1}(L_1) = 21.4$  avec  $P_1(e_1) = 0.4$ ,  $P_1(e_2) = 0.4$  et  $P_1(e_3) = 0.2$
- $\Gamma_{max}(L_2) = EU_{P_2}(L_2) = 17.9$  avec  $P_2(e_1) = 0.1$ ,  $P_2(e_2) = 0.5$  et  $P_2(e_3) = 0.4$

Au sens du critère  $\Gamma$ -Maximax, la loterie  $L_1$  est donc préférée à la loterie  $L_2$ . A l'instar du critère  $\Gamma$ -Maximax proposé dans le cadre de l'incertitude totale, ce critère traduit des préférences fortement optimistes puisque l'on ne prend en compte que le meilleur cas (i.e., la distribution de probabilité optimisant l'espérance d'utilité) pour évaluer une loterie.

De façon analogue, on évalue une loterie  $L = (x_1, K(e_1); \dots; x_n, K(e_n))$  avec le critère  $\Gamma$ -Maximin comme suit (i.e.,  $V \equiv \Gamma_{min}$ ) :

$$\Gamma_{min}(L) = \min_{P \in \mathcal{P}_K} EU_P(L)$$

En guise d'exemple, considérons les loteries représentées sur la figure 1.9. Les valeurs assignées aux loteries par le critère  $\Gamma$ -Maximin sont les suivantes (avec  $\forall x \in \mathbb{R}, u(x) = x$ ) :

- $\Gamma_{min}(L_1) = EU_{P_1}(L_1) = 11.2$  avec  $P_1(e_1) = 0.1$ ,  $P_1(e_2) = 0.5$  et  $P_1(e_3) = 0.4$
- $\Gamma_{min}(L_2) = EU_{P_2}(L_2) = 13$  avec  $P_2(e_1) = 0.4$ ,  $P_2(e_2) = 0.3$  et  $P_2(e_3) = 0.3$

Au sens du critère  $\Gamma$ -Maximin, la loterie  $L_2$  est donc préférée à la loterie  $L_1$ . Encore une fois, on reproche à ce critère de traduire des comportements trop pessimistes puisque seul le pire cas (i.e., la distribution de probabilité minimisant l'espérance d'utilité) est envisagé pour évaluer une loterie.

Notons que le critère  $\Gamma$ -Maximin peut s'interpréter comme une intégrale de Choquet. En effet, lorsque la probabilité inférieure  $\underline{K}$  de la distribution de probabilité intervalle  $K$  est une capacité de Choquet convexe et que l'hypothèse d'accessibilité est vérifiée (ce qui est le cas ici), alors pour toute loterie  $L = (x_1, K(e_1); \dots; x_n, K(e_n))$  on a (Denneberg, 1997) :

$$\Gamma_{min}(L) = CEU(L) \text{ avec } v(A) = \inf_{P \in \mathcal{P}_K} P(A)$$

Autrement dit, maximiser le critère  $\Gamma$ -Maximin revient à maximiser l'espérance d'utilité à la Choquet en prenant comme capacité la probabilité inférieure d'une distribution de probabilité intervalle. Ceci rappelle l'interprétation de Schmeidler (1986) de l'aversion au risque en terme de pessimisme. En effet, ce dernier a montré qu'une interprétation possible de l'aversion au risque pouvait se traduire par une capacité de Choquet convexe et a montré l'équivalence des deux propositions suivantes :

- $v$  est une capacité convexe
- $core(v) \neq \emptyset$  et  $\forall L \in \mathbb{L}_X, CEU(L) = \min_{P \in core(v)} EU_P(L)$

Ainsi, à travers le critère  $\Gamma$ -Maximin, on retrouve cette même idée d'aversion au risque, en terme de pessimisme, proposée par Schmeidler, consistant à évaluer une loterie par la pire espérance d'utilité.

Toutefois, il est important de souligner que les ensembles  $\mathcal{P}_K$  et  $core(v)$  (lorsque  $v$  est défini par la probabilité inférieure  $\underline{K}$  de  $K$ ) sont deux ensembles qui ne sont généralement pas égaux.

### Critère de Hurwicz

Les critères  $\Gamma$ -Maximax et  $\Gamma$ -Maximin traduisant des comportements décisionnels trop extrêmes, Jaffray et Jeleva (2007) proposent d'adapter le critère de Hurwicz au cadre des probabilités intervalles, en réalisant la somme pondérée par  $\alpha \in [0; 1]$  entre les valeurs retournées par les critères  $\Gamma$ -Maximax et  $\Gamma$ -Maximin.

Autrement dit, l'évaluation d'une loterie  $L = (x_1, K(e_1); \dots; x_n, K(e_n))$  se réalise via la formule suivante (i.e.,  $V \equiv Hz$ ) :

$$Hz(L) = \alpha \min_{P \in \mathcal{P}_K} EU_P(L) + (1 - \alpha) \max_{P \in \mathcal{P}_K} EU_P(L) \text{ avec } \alpha \in [0; 1]$$

En guise d'exemple, considérons les loteries de la figure 1.9 page 50. Les valeurs assignées aux loteries par le critère de Hurwicz sont les suivantes (avec  $\forall x \in \mathbb{R}, u(x) = x$ ) :

- pour  $\alpha = 0.2$ 
  - $H_z(L_1) = 0.2 \times \Gamma_{min}(L_1) + 0.8 \times \Gamma_{max}(L_1) = 19.36$
  - $H_z(L_2) = 0.2 \times \Gamma_{min}(L_2) + 0.8 \times \Gamma_{max}(L_2) = 16.92$
- pour  $\alpha = 0.8$ 
  - $H_z(L_1) = 0.8 \times \Gamma_{min}(L_1) + 0.2 \times \Gamma_{max}(L_1) = 13.24$
  - $H_z(L_2) = 0.8 \times \Gamma_{min}(L_2) + 0.2 \times \Gamma_{max}(L_2) = 13.98$

Lorsque  $\alpha$  est proche de 0 (i.e. lorsque le décideur est optimiste) la loterie  $L_1$  est donc préférée à la loterie  $L_2$ , et inversement, lorsque  $\alpha$  est proche de 1 (i.e., lorsque le décideur est pessimiste) la loterie  $L_2$  est préférée à la loterie  $L_1$ . Comme dans le cadre de l'incertitude totale, le paramètre  $\alpha$  permet de faire varier le degré de pessimisme du décideur. Pour déterminer la valeur de  $\alpha$ , il suffit d'employer la même méthode que dans le cadre de l'incertitude totale puisqu'il s'agit là d'un cas particulier de risque imprécis (i.e., toutes les probabilités intervalles sont  $[0; 1]$ ).

### E-admissibilité

L'E-admissibilité n'est pas, à proprement dit, un critère de décision. Elle se rapproche davantage d'une idée de dominance puisqu'elle permet de classer les loteries en deux sous-ensembles disjoints : les loteries E-admissibles et les loteries non E-admissibles.

La notion d'E-admissibilité a été introduite par Levi (1974) et discutée par Seidenfeld (2004). Le caractère E-admissible d'une loterie offre certaines propriétés intéressantes qui assurent l'efficacité d'une loterie sur les loteries non E-admissibles. On dit qu'une loterie est E-admissible si et seulement si elle est optimale, pour au moins une distribution de probabilité au sens de l'espérance d'utilité. Plus formellement, la loterie  $L_i = (x_1, K(e_1); \dots; x_n, K(e_n))$  est dite E-admissible dans un ensemble de loteries  $A = \{L_1; \dots; L_k\}$  si et seulement si  $\exists P \in \mathcal{P}_K, \forall j \neq i, EU_P(L_i) \geq EU_P(L_j)$ . L'E-admissibilité permet ainsi d'assurer qu'il existe au moins un cas (i.e., une distribution de probabilité) dans lequel cette loterie fait partie des meilleurs choix.

En guise d'exemple considérons le problème de décision dans le risque imprécis suivant :

$E$	$e_1$	$e_2$
$K(E)$	$[0.3; 0.6]$	$[0.4; 0.7]$
$L_1$	40	12
$L_2$	12	40
$L_3$	15	15

La loterie  $L_1$  est E-admissible puisqu'aucune des autres loteries ne possède une valeur d'espérance d'utilité strictement supérieure à la sienne sur la distribution de probabilité  $P_1$  définie par  $P_1(e_1) = 0.6$  et  $P_1(e_2) = 0.4$ . De même, la loterie  $L_2$  est E-admissible puisqu'elle est imbattue sur la distribution de probabilité  $P_2$  définie par  $P_2(e_1) = 0.3$  et  $P_2(e_2) = 0.7$ . Par contre la loterie  $L_3$  n'est pas E-admissible puisqu'il n'existe aucune distribution de probabilité pour laquelle elle possède un score d'espérance d'utilité supérieure à celui de toutes les autres loteries. En effet, quelle que soit la distribution de probabilité considérée, l'espérance d'utilité de la loterie  $L_3$  vaut 15 alors que l'on a  $\Gamma_{min}(L_1) = 20.4$  et  $\Gamma_{min}(L_2) = 23.2$ . Soulignons le fait qu'un décideur qui maximise le critère  $\Gamma$ -Maximax prend toujours une décision E-admissible contrairement à un décideur maximisant  $\Gamma$ -Maximin ou le critère de Hurwicz.

## Conclusion

La notion d'incertitude est une notion très large et englobe toute information non déterministe à l'échelle du décideur.

De nombreuses modélisations de l'incertitude existent et nous n'avons passé en revue qu'un sous-ensemble de celles proposées dans la littérature. Nous avons commencé par faire la distinction entre les cas risqués (i.e., incertitude représentée par une distribution de probabilité, connue du décideur) et les cas non risqués (toutes les autres formes d'incertitude).

Dans les cas risqués, de nombreux modèles issus de la littérature économique existent, et la pauvreté du pouvoir descriptif de l'espérance de gain n'est plus sujette à débat depuis plusieurs siècles.

L'ajout d'une fonction d'utilité, qui conduit au modèle de l'espérance d'utilité, est un premier pas vers une modélisation solide des préférences d'un décideur en situation risquée, mais ne suffit toujours pas à décrire la grande diversité des comportements observés. Son incapacité à modéliser de manière fidèle l'aversion au risque est maintenant admise dans la littérature économique.

L'ajout d'une fonction de déformation des probabilités dans le cadre du modèle RDU permet alors de fournir le levier manquant à la description des préférences d'un décideur en situation risquée, tout en préservant le décideur des décisions menant à des loterie dominées stochastiquement au premier ordre.

Les cas non risqués, aujourd'hui très étudiés dans les communautés des économistes et des informaticiens, semblent plus difficile à traiter.

Nous avons présenté une première approche consistant à se ramener à un cadre quasi-identique au cadre du risque. Pour ce faire, le décideur établit, de manière subjective, l'ensemble des probabilités sur les états de la nature. Cependant l'établissement de telles probabilités peut se révéler problématique, voire même impossible, dans de nombreuses situations.

Une seconde approche consiste à travailler uniquement avec les données présentes dans l'énoncé initial, sans éliciter une distribution de probabilité subjective. Les cadres de l'incertitude totale et du risque imprécis (i.e., incertitude représentée par une distribution de probabilité intervalle sur les états de la nature) s'inscrivent dans ce type d'approches. Ces deux formes d'incertitude non risquée, loin d'être les seules rencontrées dans la pratique, seront celles qui détermineront les cadres d'incertitude qui nous intéresseront dans les prochains chapitres. Pour cette raison, nous avons passé en revue les différents critères de décision praticables dans ces deux cadres.

Le chapitre suivant traite des problèmes de décision séquentielle dans l'incertain. Dans ces problèmes, on ne s'intéresse plus à prendre une unique décision, mais à prendre un ensemble de décisions étalées dans le temps. Dans ce chapitre, nous nous intéresserons à la représentation graphique d'une telle problématique. Enfin, nous verrons comment il est théoriquement possible d'appliquer les critères de décision vu dans ce chapitre dans un cadre de décision séquentielle.





## Chapitre 2

# Décision séquentielle dans l'incertain

**Résumé.** *Dans ce chapitre, nous nous intéressons aux problèmes de décision séquentielle dans l'incertain. Après avoir formalisé cette problématique, nous nous intéressons à la représentation graphique d'un problème de décision séquentielle sous incertitude sous forme d'un arbre de décision. La résolution d'un problème de décision séquentielle dans l'incertain est un problème combinatoire en soi, de par le grand nombre de stratégies potentielles. La programmation dynamique est valide lorsque le critère de décision appliqué respecte l'axiome d'indépendance (ou le principe de la chose sûre). Dans tous les autres cas, l'échec de la programmation dynamique est directement lié au fait que ces critères permettent de décrire des comportements non-conséquentialistes. Machina (1989) montre, au travers de nombreux exemples, qu'il est tout à fait naturel de rejeter le conséquentialisme lorsque les préférences du décideur ne vérifient pas l'axiome d'indépendance. Suite à cette observation, McClennen (1990) propose la méthode du choix résolu permettant d'appliquer, dans un cadre séquentiel, les critères de décision ne respectant pas l'axiome d'indépendance. Jaffray (1999) perfectionne cette théorie et la rend opérationnelle avec sa procédure de veto-process.*

## Introduction

Jusqu'alors nous n'avions considéré que des problèmes dans lesquels le décideur avait à choisir entre plusieurs actions ponctuelles. Une problématique plus large, que l'on rencontre fréquemment dans la pratique, est de considérer des situations dans lesquelles le décideur ne devra non pas prendre une unique décision, mais plusieurs décisions étalées dans le temps (une *stratégie* ou une *politique*).

En guise d'illustration, considérons le jeu de type *quitte ou double* de l'exemple 7. Dans ce problème, les gains (ou les pertes) du décideur dépendent des décisions prises et des événements qui se sont effectivement réalisés.

**Exemple 7** *Dans un premier temps, le décideur est amené à choisir entre l'événement Pile et l'événement Face. On jette alors une pièce de monnaie et on regarde quel événement se réalise : Pile ou Face. Si le joueur a choisi le mauvais événement, il repart avec la somme de 0 Euro. Sinon, le décideur doit choisir entre repartir avec la somme de 50 Euros ou relancer à nouveau la pièce. S'il choisit de relancer la pièce, il repart avec la somme de 100 Euros si la pièce tombe une seconde fois sur le bon événement, et il repart avec la somme de 0 Euro dans le cas contraire.*

Ce type de problème est communément appelé *problème de décision séquentielle dans l'incertain*. Ces problèmes sont très fréquents en informatique : on les rencontre dès lors que l'on a besoin de planifier des actions dans le temps dans un environnement incertain. Par exemple, on rencontre ce type de problématique dans l'élaboration de joueurs artificiels pour des jeux multi-joueurs (l'incertitude étant alors liée aux décisions de jeu des autres joueurs), ou encore dans des problèmes de navigation de robot en terrain inconnu ou des problèmes de diagnostic médical.

Dans ce chapitre, en premier lieu, nous définissons de manière formelle un problème de décision séquentielle dans l'incertain et nous détaillons comment représenter un tel problème sous la forme d'un arbre de décision.

Nous montrons ensuite que résoudre un problème de décision séquentielle dans l'incertain est un problème algorithmique en soi. Enfin nous présentons l'algorithme classique de programmation dynamique proposé pour l'utilisation du critère de l'utilité espérée. Cependant, à l'aide d'un exemple emprunté à Jaffray et Nielsen (2006), nous mettons en évidence l'échec de la programmation dynamique lorsque l'axiome d'indépendance (ou le principe de la chose sûre) n'est pas vérifié. Suite à ce constat, nous présentons l'idée du choix résolu, proposée par McClennen (1990), et celle de veto-process, proposée par Jaffray (1999). Ces formalismes posent des bases décisionnelles solides permettant d'appliquer des critères de décision ne vérifiant pas l'axiome d'indépendance dans un cadre séquentiel.

## 2.1 Formalisation d'un problème de décision séquentielle dans l'incertain

Dans un problème de décision séquentielle dans l'incertain, pour une décision à prendre à la date  $t$ , le décideur peut utiliser toute l'information disponible à cette date. Par exemple, s'il réfléchit à la date  $t_1 = 0$  à ce qu'il fera à la date  $t_2 > t_1$ , il peut envisager de prendre des décisions différentes à la date  $t_2$  selon l'information qui sera alors disponible à cette date. En d'autres termes, la décision à prendre à la date  $t_2$  est conditionnée par les événements connus à la date  $t_2$ . De manière classique, on fait l'hypothèse qu'une information connue à la date  $t$  reste connue aux dates ultérieures (le décideur a une mémoire). Cette hypothèse entraîne que l'ensemble  $\mathfrak{P}_{t+1} = \{e_{t+1}^1; \dots; e_{t+1}^n\}$  des événements connus à la date  $t + 1$  forme une sous-partition de l'ensemble  $\mathfrak{P}_t = \{e_t^1; \dots; e_t^m\}$  des événements connus à la date  $t$ .

Dans un tel cadre, on peut formaliser un problème de décision séquentielle dans l'incertain comme suit. L'ensemble des événements  $\mathfrak{E}$  est constitué d'une suite de partitions  $\mathfrak{P}_t$  avec  $t = 1, \dots, T$  de l'événement certain  $\Omega$ ,  $\mathfrak{P}_{t+1}$  étant une sous-partition de  $\mathfrak{P}_t$  pour tout  $t$ . Une *stratégie*  $\Delta$  est constituée par la donnée, pour chaque  $t = 1, \dots, T - 1$ , d'une décision conditionnelle  $d_t^i$  pour chaque événement  $e_t^i \in \mathfrak{P}_t$ , ainsi que d'une décision initiale à  $t = 0$ .

Au terme de la résolution des différents événements et des décisions prises conditionnellement à ces événements, le décideur atteint une conséquence  $x \in X$  ( $X$  désignant l'ensemble des conséquences possibles). Puisque l'on dispose d'une structure emboîtée sur l'ensemble des événements  $\mathfrak{E}$ , il suffit de préciser l'événement réalisé à la date  $T$  pour déterminer les événements qui se sont réalisés aux dates précédentes (par la condition  $e_1^i \supset e_2^j \supset \dots \supset e_T^k$  reliant les événements réalisés<sup>1</sup>). On utilise alors une fonction résultat  $\Psi$  qui, à toute suite de décisions  $(d_0, d_1, \dots, d_{T-1})$  et tout événement  $e_T \in \mathfrak{P}_T$ , associe  $x = \Psi(d_0, d_1, \dots, d_{T-1}, e_T)$  où  $x$  est la conséquence atteinte lorsque  $e_T$  se réalise et que la suite de décisions conditionnelles  $(d_0, d_1, \dots, d_{T-1})$  est prise (où  $d_t$  est la décision conditionnelle à l'événement  $e_t \in \mathfrak{P}_t$ ,  $e_t \supset e_T$ ).

Il apparaît alors qu'une stratégie  $\Delta$  peut conduire à autant de conséquences différentes qu'il y a d'événements dans  $\mathfrak{P}_T$ .

**Exemple 8** Soit  $\mathfrak{E}$  l'ensemble des événements constitué par  $\mathfrak{P}_1 = \{e_1^1, e_1^2, e_1^3\}$  et  $\mathfrak{P}_2 = \{e_2^1, e_2^2, e_2^3, e_2^4, e_2^5, e_2^6\}$  tels que  $e_1^1 = e_2^1 \cup e_2^2 \cup e_2^3$ ,  $e_1^2 = e_2^4 \cup e_2^5$  et  $e_1^3 = e_2^6$ . L'ensemble des décisions que peut prendre le décideur à la date  $t = 0$  est  $\{d_0^1, d_0^2\}$  et l'ensemble des décisions que le décideur peut prendre à la date  $t = 1$  est  $\{d_1^1, d_1^2, d_1^3\}$ . Une stratégie possible

---

1. Les événements de  $\mathfrak{P}_T$  peuvent alors être considérés comme des événements élémentaires.

est  $\Delta = (d_0^1, d_1^2 \text{ si } e_1^1, d_1^1 \text{ si } e_1^2, d_1^3 \text{ si } e_1^3)$ . Elle peut conduire aux six conséquences suivantes :  $\Psi(d_0^1, d_1^2, e_2^1)$ ,  $\Psi(d_0^1, d_1^2, e_2^2)$ ,  $\Psi(d_0^1, d_1^2, e_2^3)$ ,  $\Psi(d_0^1, d_1^1, e_2^4)$ ,  $\Psi(d_0^1, d_1^1, e_2^5)$ ,  $\Psi(d_0^1, d_1^3, e_2^6)$ .

Notons que dans la pratique, l'ensemble des décisions disponibles à une date  $t$ , dépend directement des décisions prises et des événements qui se sont réalisés aux dates  $t' < t$ . Pour illustration reprenons le jeu de Quitte ou Double de l'exemple 7. Si on note :

- $e_p$  : l'événement « la pièce tombe sur Pile »,
- $e_f$  : l'événement « la pièce tombe sur Face »,
- $e_{pp}$  : l'événement « la pièce tombe sur Pile au premier lancer et sur Pile au second lancer »,
- $e_{pf}$  : l'événement « la pièce tombe sur Pile au premier lancer et sur Face au second lancer »,
- $e_{fp}$  : l'événement « la pièce tombe sur Face au premier lancer et sur Pile au second lancer »,
- $e_{ff}$  : l'événement « la pièce tombe sur Face au premier lancer et sur Face au second lancer »,

alors on a  $\mathfrak{E}$  constitué par  $\mathfrak{P}_1 = \{e_p, e_f\}$  et  $\mathfrak{P}_2 = \{e_{pp}, e_{pf}, e_{fp}, e_{ff}\}$ . On vérifie bien que  $e_p = e_{pp} \cup e_{pf}$  et  $e_f = e_{fp} \cup e_{ff}$ . L'ensemble des décisions disponibles à la date  $t = 0$  est  $\{d_0^1 = \text{Pile}, d_0^2 = \text{Face}\}$  et l'ensemble des décisions disponibles à la date  $t = 1$  est  $\{d_1^1 = \text{Quitte}, d_1^2 = \text{Double}\}$  si l'événement  $e_p$  se réalise et que la décision  $d_0^1 = \text{Pile}$  a été prise, ou bien si l'événement  $e_f$  s'est réalisé et que la décision  $d_0^1 = \text{Face}$  a été prise. Dans tous les autres cas, l'ensemble des décisions disponibles est  $\{d_1^1 = \text{Ne rien faire}\}$ . Une stratégie alors possible est  $\Delta = \{d_0^1 = \text{Pile}, d_1^2 = \text{Double}$  si  $e_p, d_1^2 = \text{Ne rien faire}$  si  $e_f, \}$ . Elle peut conduire aux quatre conséquences suivantes :  $\Psi(d_0^1 = \text{Pile}, d_1^2 = \text{Double}, e_{pp}) = 100$ ,  $\Psi(d_0^1 = \text{Pile}, d_1^2 = \text{Double}, e_{pf}) = 0$ ,  $\Psi(d_0^1 = \text{Pile}, d_1^1 = \text{Ne rien faire}, e_{fp}) = 0$  et  $\Psi(d_0^1 = \text{Pile}, d_1^1 = \text{Ne rien faire}, e_{ff}) = 0$ .

## 2.2 Représentation graphique d'un problème de décision séquentielle dans l'incertain

Il est commode de représenter de manière graphique le problème que nous venons de formaliser. Pour ce faire, plusieurs *modèles graphiques* ont été proposés dans la littérature. Le choix d'un modèle graphique dépend de la structure du problème de décision que l'on désire modéliser, de la manière dont les données ont été obtenues ou encore des ressources informatiques dont on dispose.

Dans ce chapitre, nous nous intéressons essentiellement au formalisme des *arbres de décision* (Raiffa, 1968), qui permettent une représentation simple et explicite d'un problème de décision séquentielle dans l'incertain. Dans le chapitre 4, nous évoquons

également une seconde représentation graphique d'un problème de décision séquentielle sous incertitude : les *diagrammes d'influence* (Howard et Matheson, 1984), qui possèdent la particularité d'exploiter les diverses indépendances du problème pour fournir une représentation plus compacte que celle offerte par les arbres de décision. Nous en ferons une présentation beaucoup plus détaillée dans le chapitre 4 qui y est consacré.

Bien entendu, ces formalismes ne sont pas les seuls proposés dans la littérature, et d'autres existent, dont certains très utilisés comme *les processus décisionnels de Markov* (Bellman, 1957; Howard, 1960). Cependant, les avantages de ce dernier sont principalement exploités lorsque l'on s'intéresse à des problèmes de décision séquentielle « répétée ».

### 2.2.1 Arbre de décision

Un *graphe orienté*  $G = (\mathcal{N}, \mathcal{E})$  est composé d'un ensemble fini  $\mathcal{N}$  de nœuds et d'un ensemble fini  $\mathcal{E}$  d'*arcs*. Un arc est un couple de nœuds  $(N_1, N_2)$  avec  $\{N_1, N_2\} \subseteq \mathcal{N}$ . Si  $(N_1, N_2) \in \mathcal{E}$ , on dit alors que le nœud  $N_1$  est le *père* du nœud  $N_2$  et que le nœud  $N_2$  est le *fil* du nœud  $N_1$ . On notera par la suite  $\text{fils}(N)$  l'ensemble des fils du nœud  $N$  et  $\text{pere}(N)$  le père du nœud  $N$ .

Un *chemin* du graphe orienté  $G = (\mathcal{N}, \mathcal{E})$  allant d'un nœud  $N \in \mathcal{N}$  à un nœud  $N' \in \mathcal{N}$  est un ensemble d'arcs  $\mathcal{E}' = \{(N_1, N_2), (N_2, N_3), \dots, (N_{n-1}, N_n)\}$  où  $\mathcal{E}' \subseteq \mathcal{E}$ ,  $N_1 = N$  et  $N_n = N'$ . La longueur d'un chemin est donnée par le nombre d'arcs de  $\mathcal{E}'$ .

Le *sous-graphe*  $G'$  d'un graphe orienté  $G = (\mathcal{N}, \mathcal{E})$ , induit par l'ensemble de nœuds  $\mathcal{N}' \subseteq \mathcal{N}$ , est le graphe  $G' = (\mathcal{N}', \mathcal{E}')$  où  $\mathcal{E}' = \{(N, N') : \{N, N'\} \subseteq \mathcal{N}', (N, N') \in \mathcal{E}\}$ .

Un *arbre*  $T = (\mathcal{N}, \mathcal{E})$  est un graphe orienté dans lequel il existe exactement un nœud, que l'on appelle *racine*, à partir duquel il existe exactement un chemin allant à chaque nœud. Dans un arbre, s'il existe un chemin allant de  $N$  à  $N'$ , on dit que le nœud  $N$  est un *ancêtre* du nœud  $N'$  et on dit que  $N'$  est un *descendant* du nœud  $N$ . Enfin, on note  $N_r$  la racine d'un arbre. Enfin, on appelle la *profondeur* (ou *hauteur*) d'un arbre la longueur (en nombre d'arcs) du chemin le plus long depuis la racine vers une feuille.

Le *sous-arbre*  $T(N)$  de  $T$ , est l'arbre de racine  $N \in \mathcal{N}$  qui est le sous-graphe de l'arbre  $T$  induit par le nœud  $N$  et les descendants de  $N$  dans l'arbre  $T$ .

A la manière de Jaffray et Nielsen (2006), on définit un *arbre de décision*  $T$  de la manière suivante : un arbre de décision  $T$  est un arbre dont l'ensemble des nœuds  $\mathcal{N}$  est partitionné en trois sous-ensembles disjoints :

- $\mathcal{N}_D$  : l'ensemble des *nœuds de décision*, que l'on représente graphiquement par des rectangles. Ils représentent les états où le décideur est amené à prendre une décision. Les arcs issus d'un nœud de décision représentent les décisions que peut prendre le décideur en ce nœud.

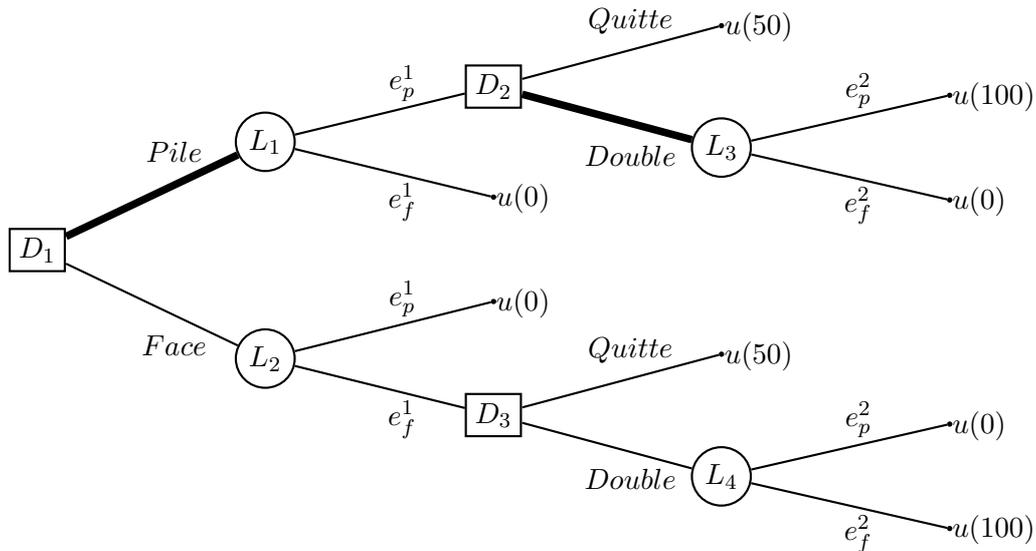


FIGURE 2.1 – Arbre de décision représentant le problème de décision séquentielle dans l'incertain de l'exemple 7. La stratégie  $\{(D_1, L_1), (D_2, L_3)\}$  est représentée en gras.

- $\mathcal{N}_C$  : l'ensemble des *nœuds de chance*, que l'on représente graphiquement par des cercles. Ils représentent les états où la nature intervient (autrement dit, les incertitudes du problème). Les arcs issus d'un nœud de chance représentent les différents événements qui peuvent se réaliser à l'issue de ce nœud.
- $\mathcal{C}$  : l'ensemble des *nœuds terminaux* (ou conséquences), qui sont les feuilles de l'arbre (i.e., nœuds n'ayant pas de fils). Ils représentent les états terminaux d'un problème de décision séquentielle. Une valeur d'utilité  $u(N)$  est associée à chaque nœud  $N$ , correspondant à l'utilité de la conséquence représentée par le nœud  $N$ .

Afin d'alléger certaines notations, nous ferons ici l'hypothèse que la racine  $N_r$  d'un arbre de décision  $T$  est toujours un nœud de décision (i.e.,  $N_r \in \mathcal{N}_D$ ).

Dans un tel formalisme, une *stratégie* pour un arbre de décision  $T = (\mathcal{N}, \mathcal{E})$  est un ensemble d'arcs  $\Delta = \{(N, N') : N \in \mathcal{N}_D^\Delta, N' \in \mathcal{N}^\Delta\} \subseteq \mathcal{E}$ , où  $\mathcal{N}_D^\Delta = \mathcal{N}_D \cap \mathcal{N}^\Delta$  et  $\mathcal{N}^\Delta \subseteq \mathcal{N}$  est un ensemble de nœuds qui contient :

- la racine  $N_r$ ,
- un fils exactement pour chaque nœud  $N \in \mathcal{N}_D^\Delta = \mathcal{N}_D \cap \mathcal{N}^\Delta$ ,
- tous les fils pour chaque nœud  $N \in \mathcal{N}_C^\Delta = \mathcal{N}_C \cap \mathcal{N}^\Delta$ .

On notera  $\overline{\Delta}^T$  l'ensemble de toutes les stratégies dans un arbre de décision  $T$ . La restriction d'une stratégie à un sous-arbre  $T(N)$ , qui est une stratégie dans le sous-arbre  $T(N)$ , est appelée une *sous-stratégie*.

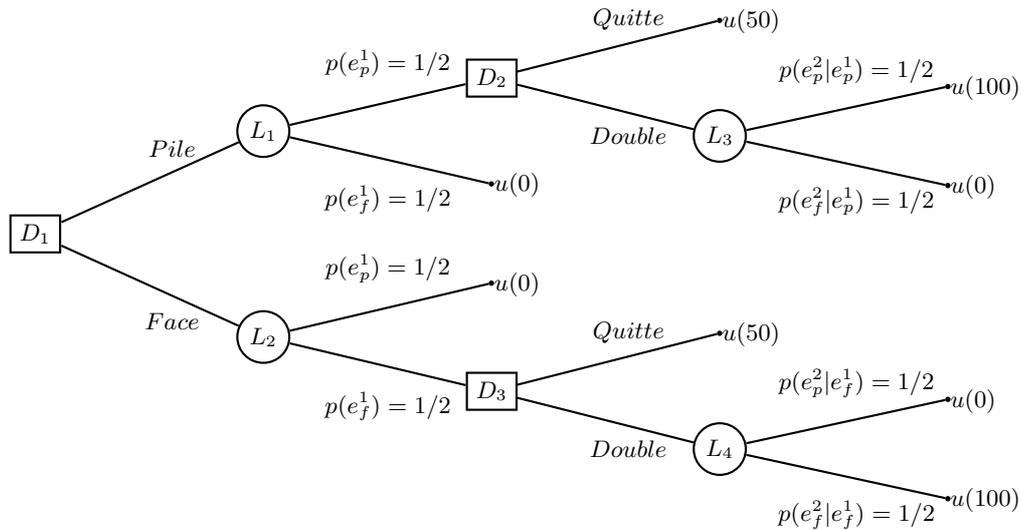


FIGURE 2.2 – Arbre de décision représentant le problème de décision séquentielle dans l'incertain de l'exemple 7 dans un cadre risqué.

Généralement, un tel arbre est représenté horizontalement de manière à ce que les différents nœuds suivent l'ordre chronologique depuis la gauche vers la droite. Notons également que, dans la pratique, on omet de préciser l'orientation des arcs lorsque l'on représente graphiquement un arbre de décision, l'orientation se déduisant naturellement à partir de l'ordre chronologique.

En guise d'illustration, considérons le jeu de type *quitte ou double* détaillé dans l'exemple 7 page 58. Si l'on nomme  $e_f^i$ , l'événement « la pièce tombe sur Face au  $i$ -ème lancer » et  $e_p^i$ , l'événement « la pièce tombe sur Pile au  $i$ -ème lancer », alors on peut représenter ce problème de décision séquentielle dans l'incertain à l'aide de l'arbre de décision représenté dans la figure 2.1 page 62. Le nœud  $D_1$  représente l'état où le décideur doit prendre sa première décision (i.e., choisir entre *Pile* ou *Face*). Les loteries (ou nœuds de chance)  $L_1$  et  $L_2$  représentent le premier lancer de pièce. Si le décideur, en  $D_1$ , n'a pas deviné le résultat du premier lancer, il obtient alors la conséquence qui lui rapporte  $u(0)$ , sinon il se retrouve en  $D_2$  ou  $D_3$  selon la décision prise en  $D_1$ . Il doit alors prendre une seconde décision : quitter ou tenter de doubler ses gains. S'il décide de quitter le jeu il se retrouve à la conséquence qui lui rapporte  $u(50)$ . Sinon les loteries  $L_3$  et  $L_4$  représentent le second lancer et, selon l'événement qui s'y réalise, le décideur se retrouve confronté aux conséquences qui lui rapportent  $u(0)$  ou  $u(100)$ .

Une stratégie possible dans ce problème est de choisir *Pile* en  $D_1$  et de choisir *Double* en  $D_2$  si  $e_p^1$  s'est réalisé en  $L_1$  (i.e.  $\Delta = \{(D_1, L_1), (D_2, L_3)\}$ ). Nous avons indiqué en gras une telle stratégie sur la figure 2.1 page 62.

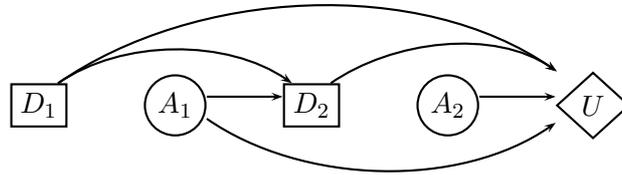


FIGURE 2.3 – Diagramme d'influence représentant le problème de décision séquentielle dans l'incertain de l'exemple 7.

Généralement, les données relatives à la vraisemblance des événements sont explicitées sur les événements conditionnels et sont associées aux arcs issus des nœuds de chance. Par exemple, si l'on se place dans un cadre risqué et que l'on suppose la pièce non truquée, alors on dispose des informations :  $p(e_f^1) = p(e_p^1) = p(e_f^2|e_f^1) = p(e_f^2|e_p^1) = p(e_p^2|e_f^1) = p(e_p^2|e_p^1) = 1/2$ . Nous pouvons ainsi en déduire l'arbre de décision donné dans la figure 2.2 page 63.

### 2.2.2 Diagrammes d'influence

Une autre manière de représenter graphiquement un problème de décision séquentielle dans l'incertain est le formalisme des diagrammes d'influence (Howard et Matheson, 1984). Ces derniers tirent avantage des propriétés d'indépendances entre les différentes variables du problème pour factoriser sa représentation. Ils ont été fortement inspirés par le formalisme des réseaux bayésiens, proposés initialement par Pearl (1985), qui permettent de représenter une loi jointe de manière compacte en tirant profit des indépendances entre les différentes variables.

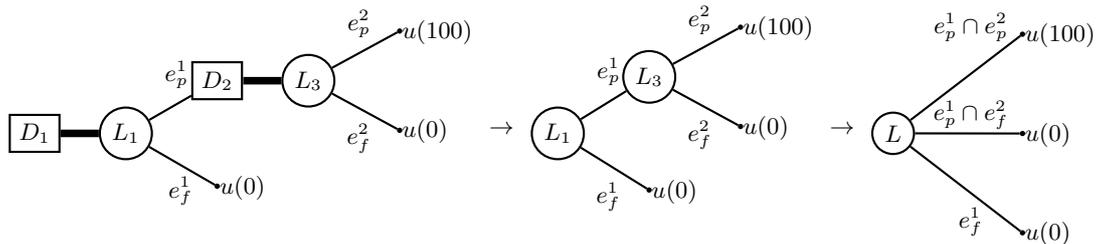
Sans détailler le formalisme des diagrammes d'influence, nous donnons dans la figure 2.3 la représentation du jeu de Quitte ou Double de l'exemple 7 page 58 sous la forme d'un diagramme d'influence. Chaque nœud est associé à une variable et les arcs aboutissants aux nœuds de chance ou d'utilité représentent les dépendances entre les différentes variables, tandis que les autres arcs représentent une notion de temporalité. Dans la figure 2.3, le gain (représenté par le nœud  $U$ ) dépend des variables de décision  $D_1$  et  $D_2$  ainsi que des variables aléatoires  $A_1$  et  $A_2$  qui représentent les lancers de pièces, c'est pourquoi tous ces nœuds sont pères du nœud  $U$ . La probabilité d'obtenir Face ou Pile lors d'un lancer est indépendant de tout le reste, aussi aucun arc de dépendance n'aboutit sur ces nœuds. Enfin, la décision prise en  $D_2$  doit se faire après avoir pris sa décision en  $D_1$  et jeté la pièce en  $A_1$ , ce qui justifie les arcs  $(A_1, D_2)$  et  $(D_1, D_2)$ . Nous reviendrons plus en détail sur ce formalisme dans le chapitre 4.

### 2.3 Comparaison de stratégies

Dans l'optique de pouvoir comparer des stratégies entre elles, il est important de remarquer que, dans un problème de décision séquentielle sous incertitude, chaque stratégie se réduit de manière simple sous la forme d'une loterie.

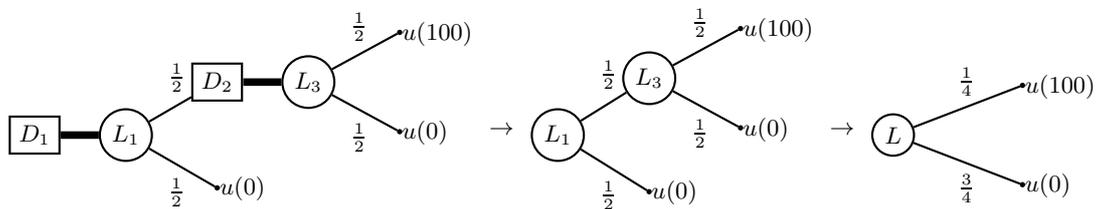
En effet, à partir de la stratégie  $\Delta$  on peut élaguer tous les sous-arbres descendants des arcs issus des nœuds de décision et n'appartenant pas à la stratégie  $\Delta$ . L'arbre obtenu ne comporte que des nœuds de décision ayant un unique fils. Ces derniers peuvent donc être supprimés en rattachant ce fils à leur père. Les nœuds restants forment alors une loterie composée.

En guise d'exemple, considérons l'arbre de décision représenté dans la figure 2.1 page 62 et construisons la loterie composée associée à la stratégie  $\Delta = \{(D_1, L_1), (D_2, L_3)\}$  représentée en gras sur la figure. En supprimant tous les sous-arbres succédant à des décisions n'appartenant pas à la stratégie  $\Delta$ , on obtient l'arbre représenté à gauche ci-dessous qui se réduit aisément en l'arbre représenté au centre ci-dessous. Ce dernier se trouve être une loterie composée et peut bien évidemment se réduire en une loterie simple (arbre représenté à droite ci-dessous).



Dans le but de pouvoir comparer les stratégies entre elles, pour évaluer une stratégie  $\Delta$  on évalue la loterie composée qui en résulte. L'évaluation d'une telle loterie se réalise à l'aide d'un critère de décision (voir chapitre 1).

Par exemple, pour évaluer la stratégie  $\Delta = \{(D_1, L_1), (D_2, L_3)\}$  de l'arbre de décision représenté sur la figure 2.2 page 63 par le modèle EU, il suffit d'évaluer la loterie suivante :



ce qui donne une espérance d'utilité de  $1/4 \times u(100) + 3/4 \times u(0) = 25$  (avec  $\forall x \in \mathbb{R}, u(x) = x$ ).

Par conséquent, on est capable de comparer des stratégies selon un critère de décision donné. Pour cette raison, on se permettra d'appliquer les critères de décision de manière indifférente sur des loteries ou des stratégies.

## 2.4 Détermination d'une stratégie optimale par programmation dynamique

Notons que, dans la pratique, le nombre de stratégies d'un arbre de décision devient très rapidement élevé lorsqu'on augmente le nombre de nœuds. En effet, considérons un arbre  $T$  de décision binaire complet de profondeur  $2h$  tel que, les nœuds de profondeur paire soient des nœuds de décision ou des nœuds d'utilité, et les nœuds de profondeur impaire soient des nœuds de chance. Nous nous intéressons ici à comptabiliser le nombre de stratégies en fonction du nombre de nœuds de décision. Le nombre de nœuds de décision est une quantité représentative (du même ordre de grandeur) de la taille de l'arbre  $T$  (nombre de nœuds).

Il y a 1 nœud de décision à la profondeur 0, 4 nœuds de décision à la profondeur 2, 16 nœuds de décision à la profondeur 4, ...,  $4^i$  nœuds de décision à la profondeur  $2i$ . Le nombre total de nœuds de décision dans  $T$  est donc égal à la somme des termes d'une suite géométrique de raison 4 :  $n = |\mathcal{N}_D| = \sum_{i=0}^{p-1} 4^i = \frac{4^p-1}{3}$ .

Exprimons maintenant le nombre de stratégies en fonction de la profondeur. Pour cela, on procède par induction arrière sur  $T$ , en remontant le nombre de stratégies jusqu'à la racine. On commence par étiqueter à 2 les nœuds de décision qui ne possèdent aucun nœud de décision dans leur descendance. On applique ensuite les relations de récurrence suivantes : le nombre de stratégies  $|\overline{\Delta}^{T(N_C)}|$ , à partir d'un nœud de chance  $N_C$ , est égal au produit des nombres de stratégies à partir de ses successeurs ; et le nombre de stratégies  $|\overline{\Delta}^{T(N_D)}|$ , à partir d'un nœud de décision  $N_D$ , est égal à la somme des nombres de stratégies à partir de ses successeurs. Ainsi, le nombre total de stratégies à partir d'un nœud de décision  $N_D$  peut se calculer à l'aide de la suite récurrente  $(u_k)$  suivante :  $u_0 = 2, u_k = 2u_{k-1}^2$ , où  $k$  indique le nombre de nœuds de décision ( $N_D$  exclu) sur un chemin quelconque de  $N_D$  vers un nœud d'utilité. Le terme général de cette suite est  $2^{(2^{k+1}-1)}$ . On peut facilement vérifier qu'on a  $k = p - 1$  à la racine. Par conséquent, le nombre total de stratégies dans  $T$  est  $|\overline{\Delta}^T| = u_{p-1} = 2^{(2^p-1)} \in \Theta(2^{\sqrt{n}})$  puisque  $n = \frac{4^p-1}{3}$ .

Dans ce cas précis d'arbre de décision, le nombre de stratégies est donc bien exponentiel en la taille de l'instance.

Face à un tel constat, il devient alors inconcevable d'énumérer l'ensemble des stratégies d'un arbre de décision afin de déterminer celle qui optimise un critère de décision donné. La recherche d'une stratégie optimale, au sens d'un critère de décision donné, est donc un problème algorithmique en soi.

Une manière naturelle d'envisager le problème est de construire la stratégie optimale par une méthode de programmation dynamique consistant à remonter, par induction arrière dans l'arbre de décision, la meilleure stratégie de chaque sous-arbre. Le principe

**Algorithme 1 : ProgDyn**(nœud  $N$ )

**Sortie :** Loterie construite par une méthode de programmation dynamique suivant l'optimisation du critère  $V$  dans l'arbre  $T(N)$ .

**si**  $N \in \mathcal{C}$  **alors**

    /\* Cas des nœuds d'utilité \*/  
    **Retourner** la loterie  $(u(N), 1)$

**si**  $N \in \mathcal{N}_D$  **alors**

    /\* Cas des nœuds de décision \*/  
    **Retourner** la loterie  $\underset{L = \text{ProgDyn}(N')}{\underset{N' \in \text{fils}(N)}{\text{argmax}}} V(L)$

**si**  $N \in \mathcal{N}_C$  **alors**

    /\* Cas des nœuds de chance \*/  
    /\* On note  $P((N, N'))$  la probabilité de l'événement associé à l'arc  $(N, N')$  \*/  
    **Retourner** la loterie  $\sum_{N' \in \text{fils}(N)} P((N, N')) \times \text{ProgDyn}(N')$

de cette méthode est de remonter, à partir des feuilles de l'arbre, et de prendre en chaque nœud de décision, la décision menant à la sous-stratégie ayant la meilleure évaluation par un critère de décision  $V$  donné. La procédure est détaillée dans l'algorithme 1 pour le cas du risque.

Il est bien connu qu'une telle méthode est valide pour le critère de l'espérance d'utilité (i.e.,  $V \equiv EU$ ) lorsque l'on se trouve dans une situation de risque. Autrement dit, la loterie retournée est celle de la stratégie optimisant le critère de l'espérance d'utilité. Notons qu'en apportant de légères modifications à l'algorithme 1, il est possible de retourner une stratégie optimale au lieu de sa loterie. En guise d'illustration, déroulons cet algorithme sur l'arbre de décision représenté dans la figure 2.2 page 63. Les loteries remontées en chaque nœud sont (avec  $\forall x \in \mathbb{R}, u(x) = \sqrt{x}$ ) :

- en  $L_3$  :  $(u(0), 0.5; u(100), 0.5)$ .
- en  $L_4$  :  $(u(0), 0.5; u(100), 0.5)$ .
- en  $D_2$  :  $\arg \max\{EU((u(0), 0.5; u(100), 0.5)), EU((u(50), 1))\} = (u(50), 1)$  puisque  $EU((u(0), 0.5; u(100), 0.5)) = 1/2 \times 10 + 1/2 \times 0 = 5$  et  $EU((u(50), 1)) = \sqrt{50}$ . La décision *Quitte* est donc sélectionnée en  $D_2$ .
- en  $D_3$  :  $\arg \max\{EU((u(0), 0.5; u(100), 0.5)), EU((u(50), 1))\} = (u(50), 1)$  puisque  $EU((u(0), 0.5; u(100), 0.5)) = 1/2 \times 10 + 1/2 \times 0 = 5$  et  $EU((u(50), 1)) = \sqrt{50}$ . La décision *Quitte* est donc sélectionnée en  $D_3$ .
- en  $L_1$  :  $(u(0), 1 \times 0.5; u(50), 1 \times 0.5) = (u(0), 0.5; u(50), 0.5)$ .
- en  $L_2$  :  $(u(0), 1 \times 0.5; u(50), 1 \times 0.5) = (u(0), 0.5; u(50), 0.5)$ .
- en  $D_1$  :  $\arg \max\{EU((u(0), 0.5; u(50), 0.5)), EU((u(0), 0.5; u(50), 0.5))\} = (u(0), 0.5; u(50), 0.5)$ . Les deux décisions sont optimales au sens du modèle EU.

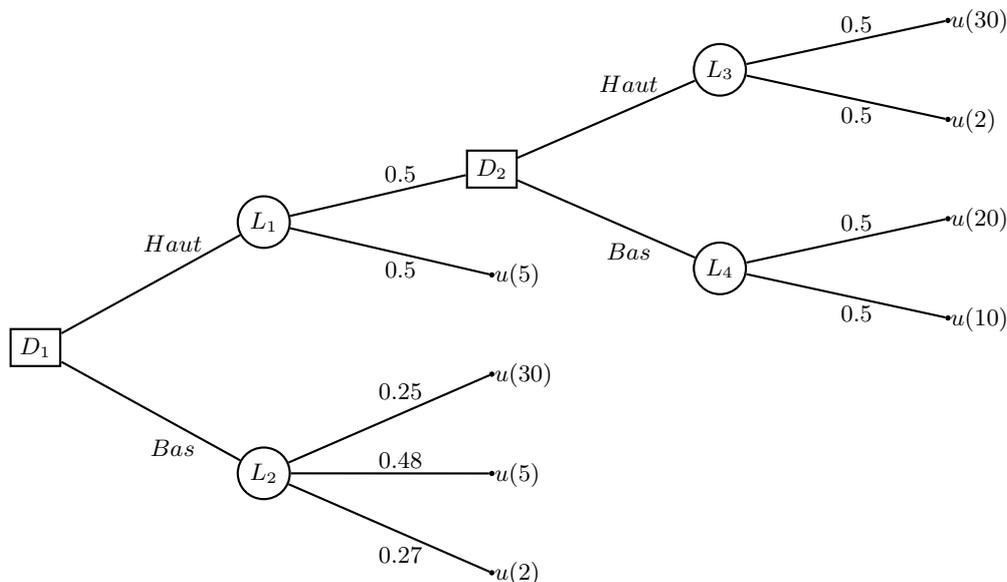


FIGURE 2.4 – Arbre de décision proposé par Jaffray et Nielsen (2006) représentant une situation dans le risque qui invalide la programmation dynamique pour la détermination d'une stratégie optimisant le modèle RDU.

Une des stratégies optimales, au sens du modèle EU, pouvant être retournée par une méthode de programmation dynamique, consiste donc à prendre la décision *Pile* en  $D_1$ , et la décision *Quitte* en  $D_2$  si  $e_p^1$  se réalise.

La validité du principe de programmation dynamique repose sur le respect de la *propriété de monotonie* (Morin, 1982) qui se formule comme suit dans un contexte de décision dans le risque :

**Propriété de monotonie.** La fonction d'évaluation  $V$  des loteries, respecte la propriété de monotonie si et seulement si  $\forall(L_1, L_2, L_3) \in \mathbb{L}_X^3$  et  $\forall\alpha \in ]0; 1[$  on a :

$$V(L_1) \geq V(L_2) \Rightarrow V(\alpha L_1 + (1 - \alpha)L_3) \geq V(\alpha L_2 + (1 - \alpha)L_3)$$

Cette propriété étant vérifiée, dans le cadre du risque, pour une fonction d'évaluation  $V$  dès lors que la relation  $\succsim$  satisfait l'axiome d'indépendance, il est alors immédiat que l'algorithme 1 page 67 est valide pour le modèle EU.

Cependant, la validité de la programmation dynamique pour les critères ne vérifiant pas l'axiome d'indépendance n'est que rarement vérifiée, comme le montre l'exemple 9 qui suit, proposé initialement par Jaffray et Nielsen (2006).

**Exemple 9** *Considérons l'arbre de décision  $T$  représenté dans la figure 2.4 page 68. La fonction de déformation des probabilités considérée ici est  $\forall p \in [0; 1], \varphi(p) = e^{-\sqrt{-\ln(p)}}$  et la fonction d'utilité est  $\forall x \in \mathbb{R}, u(x) = x$ . Dans cet arbre de décision, les différentes stratégies disponibles en  $D_1$  sont  $\{(D_1, L_1), (D_2, L_3)\}$ ,  $\{(D_1, L_1), (D_2, L_4)\}$  et  $\{(D_1, L_2)\}$ . L'évaluation de ces stratégies avec le modèle RDU donne :*

- La stratégie  $\Delta_1 = \{(D_1, L_1), (D_2, L_3)\}$  conduit à la loterie  $(u(2), 0.25; u(5), 0.5; u(30), 0.25)$ . On a donc  $RDU(\Delta_1) = 2 + \varphi(0.75) \times (5 - 2) + \varphi(0.25) \times (30 - 5) = 11.46$ .
- La stratégie  $\Delta_2 = \{(D_1, L_1), (D_2, L_4)\}$  conduit à la loterie  $(u(5), 0.5; u(10), 0.25; u(20), 0.25)$ . On a donc  $RDU(\Delta_2) = 5 + \varphi(0.5) \times (10 - 5) + \varphi(0.25) \times (20 - 10) = 10.26$ .
- La stratégie  $\Delta_3 = \{(D_1, L_2)\}$  conduit à la loterie  $(u(2), 0.27; u(5), 0.48; u(30), 0.25)$ . On a donc  $RDU(\Delta_3) = 2 + \varphi(0.73) \times (5 - 2) + \varphi(0.25) \times (30 - 5) = 11.41$ .

En  $D_2$ , les différentes sous-stratégies disponibles sont  $\{(D_2, L_3)\}$  et  $\{(D_2, L_4)\}$  et ont comme valeurs d'évaluation par le modèle RDU :

- La sous-stratégie  $\Delta_4 = \{(D_2, L_3)\}$  conduit à la loterie  $(u(2), 0.5; u(30), 0.5)$ . On a donc  $RDU(\Delta_4) = 2 + \varphi(0.5) \times (30 - 2) = 14.18$ .
- La sous-stratégie  $\Delta_5 = \{(D_2, L_4)\}$  conduit à la loterie  $(u(10), 0.5; u(20), 0.5)$ . On a donc  $RDU(\Delta_5) = 10 + \varphi(0.5) \times (20 - 10) = 14.35$ .

Une procédure par programmation dynamique impose donc de prendre la décision  $(D_2, L_4)$  en  $D_2$  puisque  $RDU(\{(D_2, L_4)\}) > RDU(\{(D_2, L_3)\})$ , et par suite de prendre la décision  $(D_1, L_2)$  en  $D_1$  puisque  $RDU(\{(D_1, L_2)\}) > RDU(\{(D_1, L_1), (D_2, L_4)\})$ . La stratégie retournée par programmation dynamique est alors  $\{(D_1, L_2)\}$  qui n'est pas optimale puisque  $RDU(\{(D_1, L_2)\}) < RDU(\{(D_1, L_1), (D_2, L_3)\})$ . De plus, remarquons que la loterie associée à la stratégie retournée par programmation dynamique est dominée stochastiquement au premier ordre par la loterie associée à la stratégie qui optimise le critère RDU. Ainsi, non seulement la stratégie construite par programmation dynamique peut ne pas être optimale, mais il se peut qu'elle soit également dominée, ce qui rend complètement inexploitable une telle stratégie.

## 2.5 Cohérence dynamique et conséquentialisme

D'un point de vue algorithmique, l'échec de la programmation dynamique pour la détermination d'une stratégie optimisant un critère de décision donné est la raison principale du non usage de ce critère dans un contexte de décision séquentielle. En effet, le problème algorithmique qui en découle devient généralement assez complexe, et décourage l'utilisation du critère.

D'un point de vue décisionnel, l'échec de la programmation dynamique est révélateur de certaines propriétés normatives qui sont violées, et a été le sujet de nombreux débats qui ont relancé la question de la légitimité de l'axiome d'indépendance (ou du principe de la chose sûre). Les arguments exposés ont fait l'objet d'une littérature conséquente mais assez disparate dans le sens où les différences terminologiques et conceptuelles d'un auteur à l'autre rendent très difficile la lisibilité (Etchart, 2002). Ainsi, il en est ressorti que le rejet de l'axiome d'indépendance pose des difficultés normatives dans un cadre de décision séquentielle. En effet, en décision séquentielle, un axiome qui se pose naturellement est celui de la *cohérence dynamique*. Cet axiome stipule que les préférences d'un décideur sont identiques à tout instant. Ainsi, un décideur est dit cohérent dynamiquement s'il est cohérent dans le temps et ne change pas d'avis alors que l'ensemble d'information reste inchangé (car la résolution de l'incertitude est prise en compte ex-ante dans la détermination de la stratégie optimale).

L'exemple 9 page 68 met en avant la violation de l'axiome de cohérence dynamique par le modèle RDU. En effet, en  $D_1$  le décideur décide d'appliquer la stratégie  $\{(D_1, L_1), (D_2, L_3)\}$  et en  $D_2$  il préfère la sous-stratégie  $\{(D_2, L_4)\}$  qui n'est pas compatible avec la stratégie fixée en  $D_1$ . Cet exemple montre de plus, que la violation de cet axiome peut entraîner des conséquences indésirables sur le plan normatif, comme la violation de la dominance stochastique du premier ordre, dès lors que l'on décide de construire une stratégie par programmation dynamique.

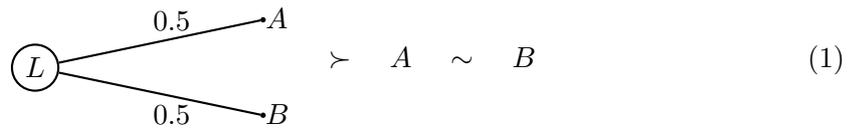
Hammond (1988, 1989) a démontré que l'axiome de cohérence dynamique ne peut être vérifié que si l'axiome d'indépendance est vérifié. Dans sa théorie du *conséquentialisme*, il établit un théorème qui fait découler l'axiome d'indépendance de trois axiomes de rationalité dynamique : l'axiome de cohérence dynamique, l'axiome *conséquentialiste* et l'axiome de réduction des loteries composées (voir chapitre 1). L'axiome conséquentialiste stipule que les préférences sur les sous-stratégies d'un sous-arbre ne dépendent que du sous-arbre en question. En d'autres termes, un décideur est dit conséquentialiste si à chaque instant ses préférences ne dépendent que de l'avenir (donc ne dépendent pas de ce qui s'est passé ou de ce qui aurait pu se passer).

Un modèle de décision qui relâche l'axiome d'indépendance doit alors violer au moins un de ces trois axiomes de rationalité dynamique. Hammond considérant comme intangibles l'axiome conséquentialiste et l'axiome de réduction des loteries composées, il en conclut que tout modèle qui ne respecte pas l'axiome d'indépendance doit relâcher l'axiome de cohérence dynamique.

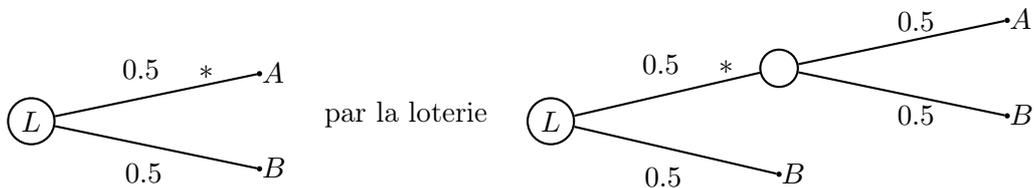
La démonstration de Hammond reposant sur le caractère intangible de l'axiome conséquentialiste et de l'axiome de réduction des loteries composées, l'unique moyen de sortir

de son raisonnement est de remettre en cause ces intangibilités. De cette manière, il devient alors possible de conserver l'axiome de cohérence dynamique tout en rejetant l'axiome d'indépendance. Machina (1989) considère justement que l'abandon de l'axiome conséquentialiste est parfaitement naturel lorsque l'on ne vérifie pas l'axiome d'indépendance.

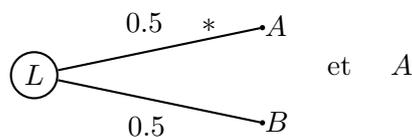
Afin d'illustrer son argumentation, il propose, entre autres, de reprendre un exemple classique de la littérature mettant en scène une mère et ses deux enfants Amandine et Benjamin. Ne possédant qu'un seul bonbon, un seul de ses enfants pourra en avoir un. On fait l'hypothèse ici, que la mère n'a aucune préférence entre ses deux enfants, ce qui signifie qu'elle est indifférente entre donner le bonbon à Amandine ou le donner à Benjamin. Par contre, par soucis d'équité, elle se refuse de le donner de manière arbitraire et préfère strictement s'en reposer sur le hasard. C'est pourquoi elle joue à pile ou face pour savoir à qui donner le bonbon afin de donner à chacun de ses enfants *la même chance* d'avoir le bonbon. Si on note  $A$  la conséquence « Amandine a le bonbon » et  $B$  la conséquence « Benjamin a le bonbon », alors les préférences de la mère peuvent se traduire comme suit :



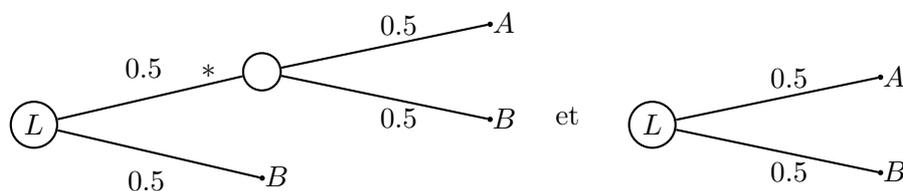
Supposons que le lancer de la pièce soit en faveur d'Amandine. Benjamin, conformément à l'axiome conséquentialiste, dit à sa mère que, puisqu'elle préfère la loterie  $(A, 0.5; B; 0.5)$  à la loterie  $(A, 1)$ , au lieu de donner le bonbon à Amandine, elle doit relancer la pièce une seconde fois pour déterminer à qui donner le bonbon (puisque'un risque déjà encouru n'influence pas les préférences), c'est à dire remplacer la loterie (le symbole « \* » indique l'événement déjà réalisé) :



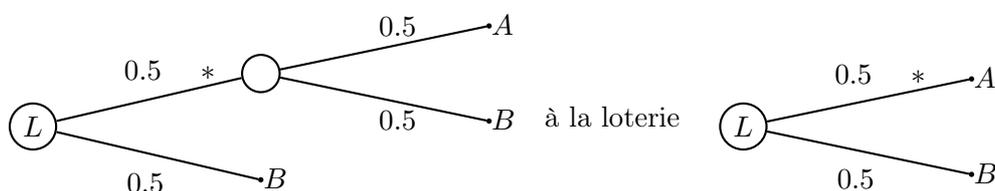
La mère réplique alors : « non, tu as déjà eu ta chance ! ». Pourtant, l'axiome conséquentialiste force à considérer comme équivalentes les loteries :



De même que les loteries suivantes sont considérées comme équivalentes au regard du conséquentialisme :



Par conséquent, d'après les préférences indiquées en (1), un décideur qui adopte un comportement conséquentialiste aurait dû préférer la loterie :



contrairement aux préférences de la mère.

Machina, à travers cet exemple, met en avant qu'un risque déjà encouru influence encore les préférences. Il fournit de nombreux autres exemples que nous ne détaillerons pas ici. Nous retiendrons seulement la conclusion que l'axiome conséquentialiste n'est pas intangible lorsque le décideur ne respecte pas l'axiome d'indépendance.

## 2.6 Choix résolu

McClellenn (1990) montre que le rejet de l'axiome conséquentialiste permet de vérifier l'axiome de cohérence dynamique sans avoir à conserver l'axiome d'indépendance. En effet, la principale différence entre un décideur  $a$  qui adopte un comportement conséquentialiste et un décideur  $b$  qui adopte un comportement non-conséquentialiste, vient du fait que le décideur  $b$  accorde encore de l'importance à un risque déjà encouru contrairement au décideur  $a$  qui ne tient plus compte d'un risque une fois qu'il a été encouru. C'est à dire qu'à tout instant, le décideur  $b$  compare toujours les loteries « vues de la racine » tandis que le décideur  $a$  compare les loteries « mises à jour » avec l'information de l'événement qui s'est réalisé.

Par exemple, dans l'arbre de décision représenté sur la figure 2.4 page 68, le décideur  $a$  compare les loteries suivantes :

- en  $D_1$  :  $L_{D_1} = (2, 0.25; 5, 0.5; 30, 0.25)$ ,  $L'_{D_1} = (5, 0.5; 10, 0.25; 20, 0.25)$  et  $L''_{D_1} = (2, 0.27; 5, 0.48; 30, 0.25)$ ,
- en  $D_2$  :  $L_{D_2} = (2, 0.5; 30, 0.5)$  et  $L'_{D_2} = (10, 0.5; 20, 0.5)$ ,

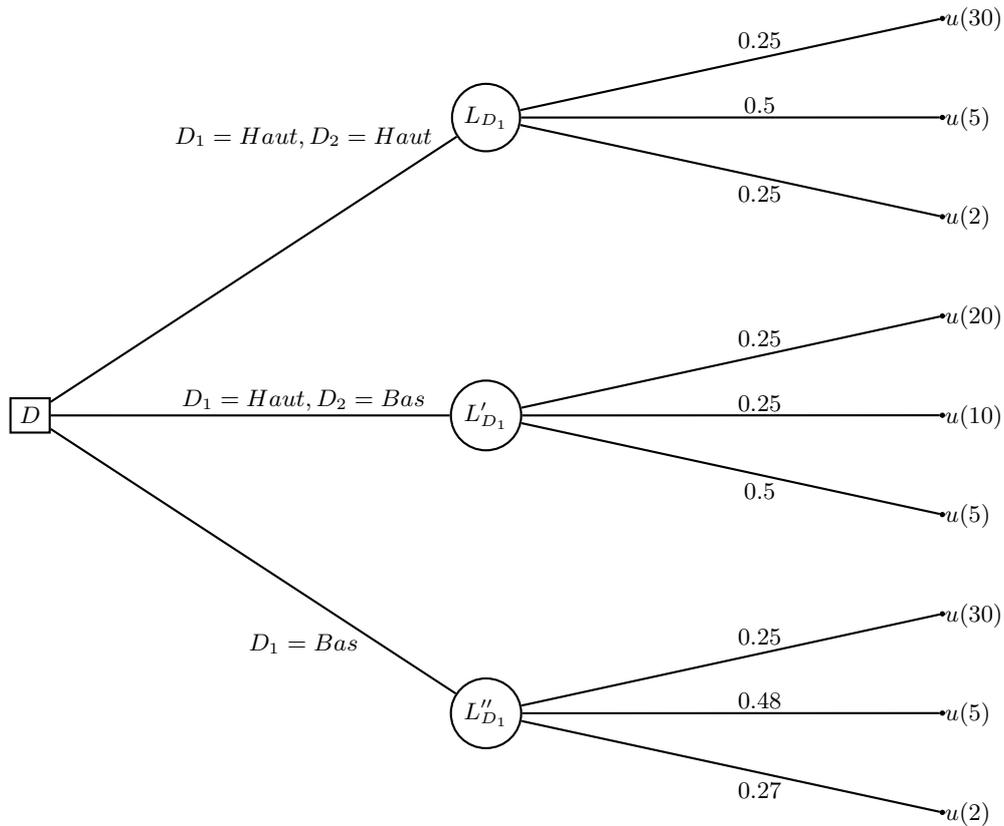


FIGURE 2.5 – Version réduite de l'arbre de décision représenté sur la figure 2.4. Ces arbres sont considérés comme équivalents pour un décideur ayant un comportement non-conséquentialiste.

tandis que le décideur  $b$  compare les loteries suivantes :

- en  $D_1$  :  $L_{D_1}$ ,  $L'_{D_1}$  et  $L''_{D_1}$ ,
- en  $D_2$  :  $L_{D_1}$  et  $L'_{D_1}$ .

C'est à dire que le décideur  $b$  considère comme équivalent le problème de décision séquentielle représenté par l'arbre de décision représenté sur la figure 2.4 page 68 et le problème de décision non séquentielle représenté par l'arbre de décision représenté sur la figure 2.5.

De cette manière, dès lors que l'on se permet de rejeter l'axiome conséquentialiste, il est tout à fait possible de conserver l'axiome de cohérence dynamique. Ce type de comportement, formalisé initialement par McClennen (1990), est bien connu sous le nom de *choix résolu*.

Le choix résolu désigne le comportement d'un décideur qui se fixe un plan de décision et n'en dévie plus par la suite (i.e., qui s'impose de rester cohérent dynamiquement). L'idée du choix résolu résulte alors dans le fait que mon *ego présent* doit fixer une stratégie et que mes *egos futurs* applique cette stratégie. Nous détaillons la notion d'ego dans la

partie suivante.

Ainsi, si l'on rejette l'axiome conséquentialiste, et que l'on décide de faire du choix résolu, on raisonne comme si l'on se plaçait dans l'arbre de décision réduit. De ce fait, l'*ego présent* dicte ses préférences aux *egos futurs*.

## 2.7 Principe du veto-process

La principale objection émise à l'encontre du choix résolu est la question de sa faisabilité d'un point de vue psychologique.

Cette objection est pertinente aussi bien dans le cadre descriptif que dans le cadre normatif qui nous intéresse tout particulièrement en informatique. La stratégie fixée au départ, pour être applicable d'un point de vue psychologique, ne doit pas être considérée comme inacceptable par l'un des egos (présent ou futurs) intervenant dans la stratégie, au risque que l'un d'entre eux choisisse de dévier de la stratégie initialement fixée.

Les questions soulevées alors par Jaffray (1999) sont les suivantes : les egos peuvent-ils collaborer ? si oui, le peuvent-ils de manière efficace ? Ces questions amènent tout naturellement à définir précisément la nature d'un ego.

A chaque nœud de décision  $N \in \mathcal{N}_D$ , on associe un ego. Cet ego *n'existe* qu'à l'instant de prendre la décision au nœud  $N$ . Ainsi, lorsque l'on dit « *les préférences de l'ego attaché au nœud  $N$  sont* », cela équivaut à dire « *les préférences du décideur au nœud  $N$  sont* ». Chaque ego est maître de la décision à prendre au nœud auquel il est attaché.

Il est important de noter que chaque ego sert ses propres intérêts, et prendra donc une décision qui va dans le sens de ces derniers. Néanmoins, les egos restent guidés par un objectif commun : retourner une stratégie non dominée. Cela signifie que si une stratégie dominée est envisagée par un ego, alors ils devraient rejeter collectivement cette stratégie.

On suppose également que chaque ego se connaît parfaitement. C'est-à-dire qu'il connaît ses propres préférences ainsi que les préférences de tous les autres egos. De cette manière, chaque ego sait exactement ce qu'il est prêt à abandonner, en déviant de ses meilleurs choix, pour aller dans le sens d'un consensus global. Il est alors envisageable de consulter l'ensemble de ces egos afin qu'ils puissent se mettre d'accord sur une stratégie de compromis satisfaisante pour chacun des egos existant lors de cette stratégie (un ego existe lors d'une stratégie si le nœud de décision auquel il se rattache existe dans le sous-arbre induit par la stratégie), et qui ne soit pas une stratégie dominée. Évidemment, si ce que chaque ego est prêt à sacrifier, en faveur d'un consensus global, est trop bas, il se peut qu'aucune stratégie de compromis ne voit le jour.

Contrairement au choix résolu où l'ego présent dicte ses préférences aux egos futurs, la stratégie déterminée ici est réalisable d'un point de vue psychologique car elle garantit qu'aucun ego futur n'aura envie de dévier de la stratégie collectivement établie.

Suivant ce principe, Jaffray (1999) propose une procédure nommée *veto-process* dans le but de rendre opérationnelle la détermination de cette stratégie de compromis.

Le principe de l'algorithme est de remonter, par programmation dynamique, en chaque nœud de décision  $N$ , un ensemble de sous-stratégies résultant d'un compromis entre l'ego associé à  $N$  et les egos présents dans le sous-arbre induit par  $N$ .

Pour ce faire, chaque ego reçoit un ensemble de sous-stratégies estimées acceptables par l'ensemble des egos appartenant à son futur et met à jour ces sous-stratégies en y préfixant les décisions dont il dispose. Une première condition nécessaire à l'adhésion d'un ego à une stratégie de compromis est le caractère non-dominée de la sous-stratégie induite. Ainsi toutes les stratégies dominées sont donc retirées de l'ensemble des stratégies envisagé par l'ego du nœud  $N$ . Chaque sous-stratégie restante est ensuite évaluée selon le critère de décision  $v$ .

Soient  $V_{max}$  la meilleure évaluation de l'ensemble des stratégies encore envisagées par l'ego au nœud  $N$ , et  $\Theta$  la tolérance de perte d'utilité pour chaque ego ( $\Theta$  représente la quantité d'utilité qu'est prêt à abandonner chaque ego en faveur d'un consensus général). Toutes les sous-stratégies dont l'évaluation est strictement inférieure à  $V_{max} - \Theta$  sont rejetées.

Enfin, parmi les sous-stratégies non-dominées, dont la valeur est supérieure ou égale à  $V_{max} - \Theta$ , on sélectionne les  $k$  meilleures<sup>2</sup> afin de se préserver d'une explosion combinatoire de la taille des ensembles de sous-stratégies remontés en chaque nœud. Une fois l'induction arrière terminée, on sélectionne la stratégie de meilleure évaluation, par le critère  $v$ , dans l'ensemble remonté à la racine de l'arbre et on la retourne.

Notons que, dans certains cas, cet ensemble peut être vide et l'algorithme ne retourne alors aucune solution. Dans ce cas, il est nécessaire de réviser les valeurs des paramètres  $\Theta$  et/ou  $k$  si les ressources physiques de calcul le permettent.

La procédure est formalisée dans l'algorithme 2 page 76, où  $p$  désigne la profondeur de l'arbre,  $\overline{\Delta}_N$  l'ensemble des sous-stratégies remontées au nœud  $N$ , et  $v(\Delta)$  la valeur de l'évaluation par le critère de décision  $v$  de la stratégie  $\Delta$ . L'algorithme 2 parcourt chaque nœud  $N$  en partant des feuilles de l'arbre jusqu'à la racine. En chacun de ces nœuds, un ensemble  $\overline{\Delta}_N$  de stratégies est construit à partir des ensembles  $\overline{\Delta}_{N'}$  de ses fils. Chaque stratégie est ensuite évaluée avec le critère de décision  $v$ . Enfin, on ne conserve, dans l'ensemble  $\overline{\Delta}_N$ , que les  $k$  meilleures stratégies qui ne sont pas dominées et dont l'évaluation par  $v$  est supérieure à  $V_{max} - \Theta$ . Pour finir, une fois que l'ensemble de stratégies est construit à la racine, on retourne celle d'évaluation maximale.

---

2. Dans l'algorithme d'origine, le paramètre  $k$  n'existe pas. Il a été introduit dans une version retravaillée par Jaffray et Nielsen (2006).

---

**Algorithme 2 : VetoProcess**(arbre de décision  $T$ , réel  $\Theta$ , entier  $k$ )
 

---

**Sortie** : une stratégie de compromis entre les différents egos

**pour chaque** nœud  $N$  de profondeur  $p$  à 0 dans  $T$  **faire**

<b>si</b> $N \in \mathcal{N}_D$ <b>alors</b> /* Cas des nœuds de décision */ $\bar{\Delta}_N \leftarrow \bigcup_{N' \in \text{fils}(N)} \{ \{(N, N')\} \cup \Delta : \Delta \in \bar{\Delta}_{N'} \}$	
<b>si</b> $N \in \mathcal{N}_C$ <b>alors</b> /* Cas des nœuds de chance */ $\bar{\Delta}_N \leftarrow \prod_{N' \in \text{fils}(N)} \bar{\Delta}_{N'}$	
<b>pour chaque</b> stratégie $\Delta \in \bar{\Delta}_N$ <b>faire</b> $\lfloor V_\Delta \leftarrow v(\Delta)$ $V_{max} \leftarrow \max_{\Delta \in \bar{\Delta}_N} \{V_\Delta\}$	
<b>pour chaque</b> stratégie $\Delta \in \bar{\Delta}_N$ <b>faire</b> <table style="border-collapse: collapse; margin-left: 1em;"> <tr> <td style="border-left: 1px solid black; padding-left: 0.5em;"> <b>si</b> (<math>\Delta</math> est dominée) OU (<math>V_\Delta &lt; V_{max} - \Theta</math>) <b>alors</b>  <math>\lfloor \bar{\Delta}_N \leftarrow \bar{\Delta}_N \setminus \{\Delta\}</math> </td> </tr> </table>	<b>si</b> ( $\Delta$ est dominée) OU ( $V_\Delta < V_{max} - \Theta$ ) <b>alors</b> $\lfloor \bar{\Delta}_N \leftarrow \bar{\Delta}_N \setminus \{\Delta\}$
<b>si</b> ( $\Delta$ est dominée) OU ( $V_\Delta < V_{max} - \Theta$ ) <b>alors</b> $\lfloor \bar{\Delta}_N \leftarrow \bar{\Delta}_N \setminus \{\Delta\}$	
<b>tant que</b> $ \bar{\Delta}_N  > k$ <b>faire</b> $\lfloor \bar{\Delta}_N \leftarrow \bar{\Delta}_N \setminus \{ \arg \min_{\Delta \in \bar{\Delta}_N} \{V_\Delta\} \}$	

**Retourner**  $\arg \max_{\Delta \in \bar{\Delta}_{N_r}} \{V_\Delta\}$ 


---

## Conclusion

En informatique, de nombreux problèmes peuvent se réduire à un problème de décision séquentielle sous incertitude. Dans ces problèmes, le décideur est amené à prendre plusieurs décisions étalées dans le temps, contrairement à la problématique que nous avons vu dans le chapitre 1. Dans le présent chapitre, nous avons formalisé cette problématique et nous avons vu comment représenter un problème de décision séquentielle dans l'incertain de manière graphique sous la forme d'un arbre de décision. Cette représentation explicite possède le grand avantage d'être simple à appréhender et à expliciter. En effet, nul besoin de connaissances poussées pour comprendre le problème représenté par un arbre de décision, ou encore pour modéliser son propre problème à l'aide de cette représentation.

Nous avons également pu constater que le nombre de stratégies pouvait grandir de manière exponentielle en fonction de la taille de l'arbre de décision. La détermination d'une stratégie optimale, au sens d'un critère de décision donné, devient alors un problème algorithmique en soi. En outre, dès lors que le critère de décision considéré ne

respecte pas l'axiome d'indépendance, les approches algorithmiques qui reposent sur la programmation dynamique se révèlent inefficaces. Ce type de critères a été très peu utilisé jusqu'à aujourd'hui dans un contexte séquentiel du fait de la difficulté algorithmique qui en découle. Nous nous intéresserons dans les prochains chapitres à repousser ce verrou algorithmique.

D'un point de vue décisionnel, nous avons constaté que l'échec de la programmation dynamique était lié à la violation de l'axiome conséquentialiste. A l'issue de nombreux débats dans la littérature économique, il est ressorti qu'il était naturel de rejeter l'axiome conséquentialiste dès lors que l'on rejette l'axiome d'indépendance (Machina, 1989). Suite à cela, McClennen (1990) a introduit la notion de choix résolu afin de pouvoir appliquer les critères "non-conséquentialistes" dans un contexte séquentiel.

Ce paradigme est devenu une référence dès lors que l'on désire utiliser des critères violant l'axiome d'indépendance. Jaffray (1999) a ensuite proposé une sophistication de cette théorie avec le concept de veto-process, qui permet de garantir le bon déroulement de la stratégie fixée initialement.

Dans les prochains chapitres, nous allons justement nous intéresser à la détermination de stratégies optimales lorsque l'on utilise des critères « non-conséquentialistes ». Les problèmes algorithmiques qui se posent alors sont généralement assez complexes. Afin d'y voir plus clair dans ces problèmes, nous établissons des résultats de complexité, et nous nous efforçons de proposer des algorithmes efficaces pour déterminer la solution optimale. Ces algorithmes seront illustrés par des résultats expérimentaux que nous interpréterons.

Dans cette optique, dans le chapitre 3, nous nous intéresserons à l'application du critère RDU dans le cas du risque lorsque l'on cherche à faire du choix résolu. Nous travaillerons dans ce chapitre dans le formalisme des arbres de décisions. Ensuite, dans le chapitre 4, nous nous pencherons sur l'utilisation du critère RDU dans les diagrammes d'influence (modèles graphiques que nous détaillerons). Cette modélisation permet une représentation beaucoup plus compacte, mais nous verrons que cet atout pose de nouveaux problèmes algorithmiques qui ne se posaient pas dans le formalisme des arbres de décision. Enfin, le chapitre 5 se place dans le cadre où l'incertitude n'est pas risquée. Les solutions proposées vont alors dépendre de la modélisation de l'incertitude, et nous serons amenés à distinguer les cas d'incertitude totale et de risque imprécis.



## Chapitre 3

# Critère RDU dans les arbres de décision<sup>1</sup>

**Résumé.** Dans ce chapitre, nous nous intéressons à l'application du critère RDU dans un arbre de décision. Dans un premier temps, nous démontrons que la recherche d'une stratégie pure optimale dans un arbre de décision est un problème NP-difficile. Suite à ce constat, nous proposons deux approches pour résoudre le problème. Une première, mettant en œuvre une formulation par un programme linéaire en variables mixtes, valide uniquement pour certaines formes de la fonction  $\varphi$ , et une seconde, fondée sur le principe du « branch and bound », valide quelle que soit la fonction  $\varphi$  envisagée. Nous comparons ensuite ces deux approches via diverses expérimentations numériques. La méthode de branch and bound se révélant plus efficace et plus générale, nous l'appliquons sur un problème réel tiré du célèbre jeu télévisé « Qui veut gagner des millions ? ». L'arbre qui en résulte est d'une taille très importante (plus de 75 millions de nœuds de décision). Malgré cela, l'algorithme retourne une stratégie optimale assez rapidement. Enfin, nous nous focalisons sur le problème de la détermination d'une stratégie mixte qui optimise le critère RDU dans un arbre de décision. Afin de résoudre ce problème, nous proposons une formulation par un programme linéaire qui n'est valide que pour une certaine forme de la fonction  $\varphi$  : concave et linéaire par morceaux. Nous concluons par un résultat de polynomialité dans ce dernier cas.

---

1. Ce chapitre s'appuie en partie sur des résultats présentés dans (Jeantet et Spanjaard, 2008a) et (Jeantet et Spanjaard, 2008b).

## Introduction

Le critère de l'espérance d'utilité domine la littérature en décision dans le risque. Son attrait sur le plan normatif et la simplicité algorithmique qui en découle (i.e., la possibilité de calculer une stratégie optimale par programmation dynamique) constituent les principales raisons de ce succès. En effet, l'absence d'algorithme pour le calcul d'une stratégie optimale au sens d'un critère non-EU constitue un solide verrou à son utilisation dans la pratique.

Dans ce chapitre, nous nous consacrons au développement d'algorithmes permettant de déterminer une stratégie optimale au sens du critère de l'espérance d'utilité dépendant du rang (RDU). En effet, comme nous l'avons vu dans le chapitre 1, ce critère relâche l'axiome d'indépendance, ce qui lui confère un pouvoir descriptif très élevé. En contrepartie, les approches par programmation dynamique ne permettent pas de déterminer une stratégie optimale (voir chapitre 2) et peuvent même conduire à une stratégie dominée stochastiquement au premier ordre.

Comme nous l'avons vu dans le chapitre précédent, il est alors nécessaire de renoncer à l'axiome conséquentialiste si l'on veut vérifier l'axiome de cohérence dynamique. Une première méthode consiste à faire du choix résolu depuis la racine. Autrement dit, on détermine la stratégie qui optimise le critère RDU à la racine et on applique cette stratégie sans jamais en dévier quels que soient les événements qui se réalisent. Une seconde méthode consiste à appliquer le principe du veto-process, c'est-à-dire que l'on cherche à déterminer une stratégie qui tend à satisfaire les différents egos du décideur, au sens du critère RDU, tout en s'efforçant d'assurer le caractère non-dominée de la stratégie. Cette approche a déjà été traitée récemment dans la littérature par Jaffray et Nielsen (2006).

Nous nous intéressons ici à faire du choix résolu depuis la racine. Notre objectif diverge donc de l'approche par veto-process dans le sens où la stratégie que l'on cherche à déterminer n'est pas la même, bien que les deux approches se basent sur le critère RDU pour modéliser les préférences du décideur.

Dans un premier temps, nous présentons un résultat de complexité pour le problème de la détermination d'une stratégie optimisant le critère RDU dans un arbre de décision. Dans un second temps, afin de résoudre ce problème, nous proposons une première approche par programmation mathématique. Cependant, cette approche est valide uniquement pour certaines formes particulières de la fonction  $\varphi$  de déformation des probabilités : les fonctions concaves et linéaires par morceaux. Suite à cela, nous développons une seconde approche, par énumération implicite, permettant de traiter le problème pour n'importe quelle forme de la fonction  $\varphi$  de transformation des probabilités. Nous présentons ensuite les résultats des expérimentations numériques que nous avons menées, permettant de comparer et d'évaluer les performances de ces deux approches. Nous ex-

périmentons également l'algorithme d'énumération implicite sur un problème réel tiré du célèbre jeu télévisé « Qui veut gagner des millions ? ». Enfin, nous nous intéressons au cas des *stratégies mixtes*. Les stratégies mixtes divergent des *stratégies pures*<sup>2</sup> par le fait que l'on ne prend plus nos décisions de manière certaine mais en suivant une loi de probabilité que l'on détermine.

### 3.1 Complexité

On appelle RDU-OPT le problème de la détermination d'une stratégie qui optimise le critère RDU depuis la racine dans un arbre de décision. Sous la condition que la taille d'un arbre de décision correspond au nombre de nœuds de décision impliqués, nous prouvons dans la suite que ce problème est NP-difficile :

**Proposition 1** *Le problème RDU-OPT est un problème NP-Difficile.*

*Preuve.* On s'appuie sur une réduction polynomiale du problème 3-SAT vers le problème RDU-OPT. Le problème 3-SAT se formule comme suit :

INSTANCE : un ensemble  $X$  de variables booléennes, une collection  $C$  de clauses sur  $X$  telle que  $|c| = 3$  pour toute clause  $c \in C$ .

QUESTION : existe-t-il une instanciation des variables booléennes de  $X$  qui satisfait simultanément toutes les clauses de  $C$  ?

Soient  $X = \{x_1, \dots, x_n\}$  et  $C = \{c_1, \dots, c_m\}$ . La construction polynomiale d'un arbre de décision à partir d'une instance du problème 3-SAT se réalise comme suit.

On définit un nœud de décision pour chaque variable de  $X$ . Etant donnée  $x_i$  une variable de  $X$ , le nœud de décision associé dans l'arbre de décision, noté également  $x_i$ , possède deux fils :

- le premier (nœud de chance noté  $V_i$ ) correspond à l'instanciation *vrai* de  $x_i$ ,
- et le second (nœud de chance noté  $F_i$ ) correspond à l'instanciation *faux* de  $x_i$ .

Soient  $\{c_{i_1}, \dots, c_{i_j}\} \subseteq C$  le sous-ensemble de clauses dans lesquelles figurent le littéral positif  $x_i$ , et  $\{c_{i'_1}, \dots, c_{i'_k}\} \subseteq C$  le sous-ensemble de clauses dans lesquelles figurent le littéral négatif  $\bar{x}_i$ .

Pour chaque clause  $c_{i_h}$  ( $1 \leq h \leq j$ ) on crée, comme fils de  $V_i$ , un nœud terminal noté  $c_{i_h}$ , correspondant à la clause  $c_{i_h}$ . On crée, de plus, un fils supplémentaire de  $V_i$  noté  $c_0$ , correspondant à une conséquence  $c_0$  fictive. De même, on crée un fils de  $F_i$  pour chaque clause  $c_{i'_h}$  ( $1 \leq h \leq k$ ), ainsi qu'un fils supplémentaire correspondant à la conséquence  $c_0$

---

2. Les "stratégies pures" sont celles que nous avons définies dans le chapitre 2.

fictive. De cette manière, le nœud  $V_i$  possède  $j + 1$  fils, tandis que le nœud  $F_i$  possède  $k + 1$  fils.

Afin de constituer un arbre de décision, on ajoute un nœud de chance  $H$  père de tous les nœuds de décision  $x_i$  ( $1 \leq i \leq n$ ). Enfin, on ajoute un nœud de décision à la racine, ayant  $H$  comme unique fils (ce qui permet de respecter l'hypothèse que la racine d'un arbre de décision est un nœud de décision). L'arbre de décision ainsi construit possède  $n + 1$  nœuds de décision,  $2n + 1$  nœuds de chance et au plus  $2n(m + 1)$  nœuds terminaux. Sa taille est donc en  $O(nm)$ , ce qui garantit bien la polynomialité de la transformation. A titre d'illustration, sur la figure 3.1, nous donnons l'arbre de décision obtenu pour l'instance suivante de 3-SAT :  $(x_1 \vee x_2 \vee x_3) \wedge (\overline{x_1} \vee x_3 \vee x_4) \wedge (\overline{x_2} \vee \overline{x_3} \vee \overline{x_4})$ .

Remarquons que l'on peut établir une bijection entre l'ensemble des stratégies de l'arbre de décision construit et l'ensemble des instanciations réalisables dans le problème 3-SAT initial. Pour ce faire, il suffit de poser  $x_i = 1$  dans le problème 3-SAT si et seulement si l'arc  $(x_i, V_i)$  figure dans la stratégie, et  $x_i = 0$  si et seulement si l'arc  $(x_i, F_i)$  figure dans la stratégie. Une instanciation satisfaisante du problème 3-SAT (i.e., qui satisfait simultanément toutes les clauses) correspond alors à une stratégie où toute clause  $c_i$  ( $1 \leq i \leq m$ ) figure comme conséquence possible (figurant donc de une à trois fois).

Pour compléter la réduction, il s'agit maintenant d'établir, d'une part, les probabilités assignées aux arcs issus des nœuds de chance  $H$ ,  $V_i$  et  $F_i$ , et d'autre part, les utilités des conséquences et la fonction  $\varphi$ . La réduction va consister à les définir de façon à ce que seules les stratégies correspondant à des instanciations satisfaisantes maximisent le critère RDU. Plus précisément, nous visons à ce que :

- (i) la valeur RDU d'une stratégie ne dépende que de l'ensemble (et non du multi-ensemble) de ses conséquences possibles (autrement dit l'ensemble des clauses satisfaites par l'instanciation correspondante),
- (ii) la valeur RDU d'une stratégie correspondant à une instanciation satisfaisante vaille exactement  $m$ ,
- (iii) si une stratégie est susceptible de conduire à un ensemble de conséquences possibles qui est strictement inclus dans l'ensemble des conséquences d'une autre stratégie, la valeur RDU de cette dernière soit strictement supérieure.

Pour ce faire, après avoir affecté la probabilité  $\frac{1}{n}$  aux arcs issus de  $H$ , on définit les autres probabilités et les utilités de la façon suivante (pour  $i \neq 0$ ) :

$$\begin{cases} p_i = \left(\frac{1}{10}\right)^i \\ u(c_i) = \sum_{j=1}^i 10^{j-1} \end{cases}$$

où  $p_i$  désigne la probabilité associée aux arcs conduisant à une conséquence  $c_i$ . Pour les arcs de type  $(V_j, c_0)$  (resp.  $(F_j, c_0)$ ), on pose  $u(c_0) = 0$  et on affecte la probabilité qui

complémentaire à 1 l'ensemble des probabilités affectées aux arcs issus du nœud de chance  $V_j$  (resp.  $F_j$ ). Notons que cette dernière probabilité est bien positive puisque la somme des  $p_i$  est strictement inférieure à 1. Enfin, la fonction  $\varphi$  est définie comme suit<sup>3</sup> :

$$\varphi(p) = \begin{cases} 0 & \text{si } p \in [0; \frac{p_m}{n}[ \\ p_i & \text{si } p \in [\frac{p_{i+1}}{n}; \frac{p_i}{n}[ \text{ pour } i < m \\ 1 & \text{si } p \in [\frac{p_1}{n}; 1[ \end{cases}$$

A titre d'illustration, les probabilités et la fonction d'utilité obtenues pour l'instance de 3-SAT indiquée plus haut ont été indiquées sur l'arbre de décision de la figure 3.1 page 84 et la fonction  $\varphi$  obtenue est la suivante :

$$\varphi(p) = \begin{cases} 0, & \text{si } p \in [0; \frac{1}{4 \times 1000}[ \\ \frac{1}{100}, & \text{si } p \in [\frac{1}{4 \times 1000}; \frac{1}{4 \times 100}[ \\ \frac{1}{10}, & \text{si } p \in [\frac{1}{4 \times 100}; \frac{1}{4 \times 10}[ \\ 1, & \text{si } p \in [\frac{1}{4 \times 10}; 1[ \end{cases}$$

Dans la suite, nous considérons une stratégie quelconque  $\Delta$ , induisant une loterie notée  $L$ , et on note  $I \subseteq \{0, \dots, m\}$  l'ensemble des indices des conséquences possibles de  $\Delta$ . Remarquons que la conséquence  $c_0$  est toujours présente dans une stratégie  $\Delta$ . On appelle  $\alpha_i \in \{1, 2, 3\}$  le nombre d'occurrences de la conséquence  $c_i$  dans  $\Delta$ . Par abus de notation, nous confondons ci-dessous  $c_i$  et  $u(c_i)$ .

*Preuve de (i).* La valeur RDU d'une stratégie  $\Delta$  quelconque vaut :

$$RDU(L) = c_0 \times \varphi(1) + \sum_{i \in I} (c_i - c_{prec_I(i)}) \varphi \left( \sum_{\substack{j \in I \\ j \geq i}} \alpha_j \frac{p_j}{n} \right), \text{ où } prec_I(i) = \max\{j \in I : j < i\}.$$

Montrons que :

$$\forall i \in I, \varphi \left( \sum_{\substack{j \in I \\ j \geq i}} \alpha_j \frac{p_j}{n} \right) = \varphi \left( \sum_{\substack{j \in I \\ j \geq i}} \frac{p_j}{n} \right).$$

Par croissance de  $\varphi$ , on a :

$$\varphi \left( \sum_{\substack{j \in I \\ j \geq i}} \frac{p_j}{n} \right) \leq \varphi \left( \sum_{\substack{j \in I \\ j \geq i}} \alpha_j \frac{p_j}{n} \right) \leq \varphi \left( \sum_{\substack{j \in I \\ j \geq i}} 3 \frac{p_j}{n} \right).$$

---

3. Notons que la fonction  $\varphi$  choisie est croissante seulement au sens large. Bien que cela suffise, dans la pratique la fonction  $\varphi$  est strictement croissante. Cependant, en adaptant légèrement la fonction, il est possible d'obtenir une fonction  $\varphi$  strictement croissante. De même, la fonction définie ici n'est pas continue, tandis qu'en pratique la fonction  $\varphi$  est continue. Encore une fois on peut adapter facilement cette construction afin de la rendre continue.

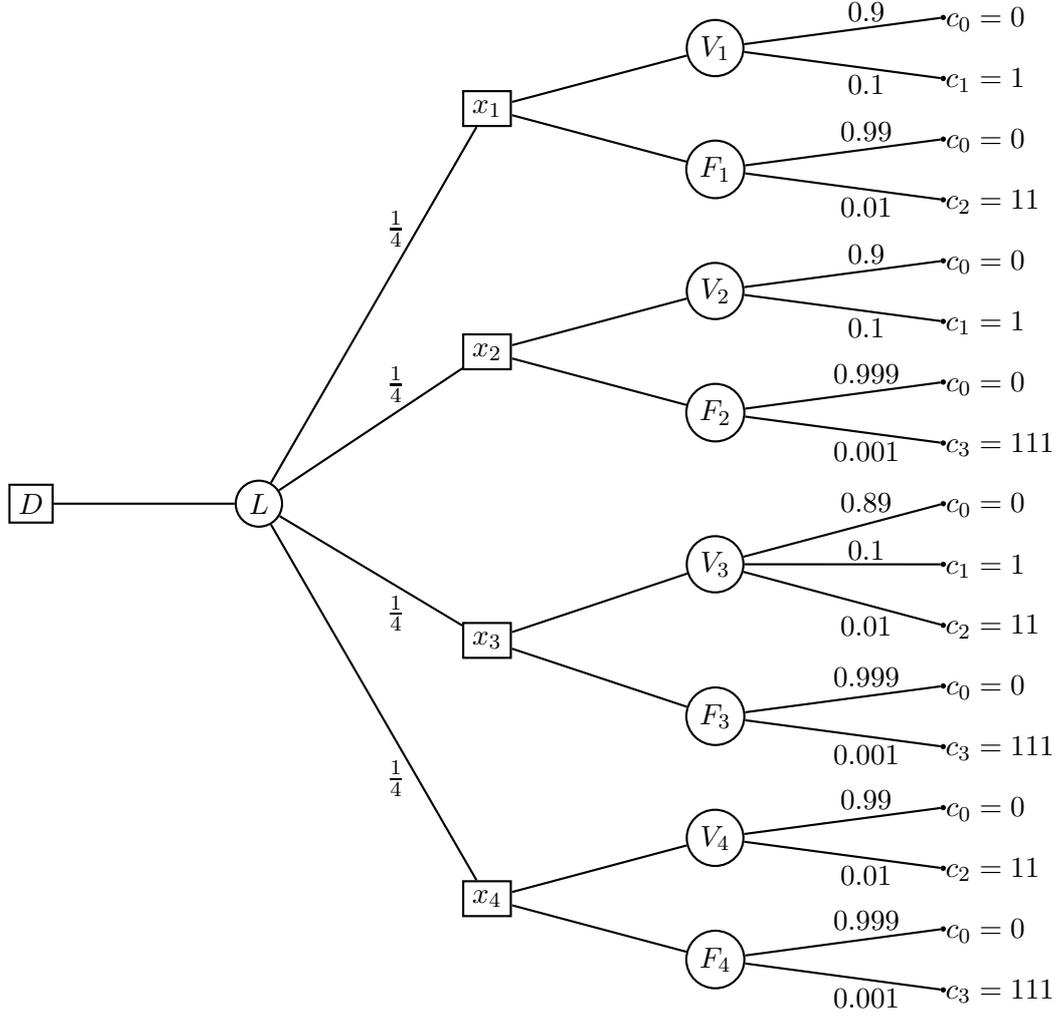


FIGURE 3.1 – Arbre de décision correspondant à la réduction polynomiale de l'instance 3-SAT suivante :  $(x_1 \vee x_2 \vee x_3) \wedge (\bar{x}_1 \vee x_3 \vee x_4) \wedge (\bar{x}_2 \vee \bar{x}_3 \vee \bar{x}_4)$ .

On a donc :

$$\varphi \left( \sum_{\substack{j \in I \\ j \geq i}} \frac{1}{n} \left( \frac{1}{10} \right)^j \right) \leq \varphi \left( \sum_{\substack{j \in I \\ j \geq i}} \alpha_j \frac{p_j}{n} \right) \leq \varphi \left( \sum_{\substack{j \in I \\ j \geq i}} \frac{3}{n} \left( \frac{1}{10} \right)^j \right).$$

Puisque

$$\varphi \left( \sum_{\substack{j \in I \\ j \geq i}} \frac{1}{n} \left( \frac{1}{10} \right)^j \right) = \varphi \left( \sum_{\substack{j \in I \\ j \geq i}} \frac{3}{n} \left( \frac{1}{10} \right)^j \right) = p_{i-1},$$

on a par encadrement :

$$\varphi \left( \sum_{\substack{j \in I \\ j \geq i}} \alpha_j \frac{p_j}{n} \right) = \varphi \left( \sum_{\substack{j \in I \\ j \geq i}} \frac{p_j}{n} \right).$$

Or  $c_0 \times \varphi(1) = 0$ . On conclut donc que :

$$RDU(L) = \sum_{i \in I} (c_i - c_{prec_I(i)}) \varphi \left( \sum_{\substack{j \in I \\ j \geq i}} \frac{p_j}{n} \right).$$

*Preuve de (ii).* Considérons une stratégie  $\Delta^*$  correspondant à une instanciation satisfaisante, et la loterie induite  $L^*$  où toutes les conséquences  $c_i$  de  $C$  sont possibles. D'après (i), on a :

$$RDU(L^*) = \sum_{i=1}^m (c_i - c_{i-1}) \varphi \left( \sum_{j=i}^m \frac{p_j}{n} \right).$$

On remarque que pour tout  $i \leq m$  :

$$(c_i - c_{i-1}) \varphi \left( \sum_{j=i}^m \frac{p_j}{n} \right) = 10^{i-1} \times p_{i-1} = 10^{i-1} \times \left( \frac{1}{10} \right)^{i-1} = 1.$$

Par conséquent,  $RDU(L^*) = m$ .

*Preuve de (iii).* Soient  $\Delta$  (resp.  $\Delta'$ ) une stratégie quelconque de loterie induite  $L$  (resp.  $L'$ ) et  $I \subseteq \{0, \dots, m\}$  (resp.  $J = I \cup \{k\}$ ) l'ensemble des indices de ses conséquences possibles. On suppose ici que  $k < \max I$ , le cas  $k = \max I$  étant évident. Par définition,  $\{i \in I : i \neq k\} = \{i \in J : i \neq k\}$ . On peut donc écrire la valeur RDU de  $\Delta$  comme une somme de trois termes :

$$\begin{aligned} RDU(L) &= \sum_{\substack{i \in J \\ i \leq k-1}} (c_i - c_{prec_J(i)}) \varphi \left( \sum_{\substack{j \in I \\ j \geq i}} \frac{p_j}{n} \right) \\ &+ (c_k - c_{prec_J(k)}) \varphi \left( \sum_{\substack{j \in I \\ j \geq k}} \frac{p_j}{n} \right) \\ &+ \sum_{\substack{i \in J \\ i \geq k+1}} (c_i - c_{prec_J(i)}) \varphi \left( \sum_{\substack{j \in J \\ j \geq i}} \frac{p_j}{n} \right) \end{aligned}$$

De la même manière, la valeur RDU de la stratégie  $\Delta'$  s'écrit également comme une somme de trois termes :

$$\begin{aligned}
RDU(L) &= \sum_{\substack{i \in J \\ i \leq k-1}} (c_i - c_{prec_J(i)}) \varphi \left( \sum_{\substack{j \in J \\ j \geq i}} \frac{p_j}{n} \right) \\
&+ (c_k - c_{prec_J(k)}) \varphi \left( \sum_{\substack{j \in J \\ j \geq k}} \frac{p_j}{n} \right) \\
&+ \sum_{\substack{i \in J \\ i \geq k+1}} (c_i - c_{prec_J(i)}) \varphi \left( \sum_{\substack{j \in J \\ j \geq i}} \frac{p_j}{n} \right)
\end{aligned}$$

Par croissance de  $\varphi$ , on a :

$$I \subseteq J \Rightarrow \forall i \leq k-1, \varphi \left( \sum_{\substack{j \in I \\ j \geq i}} \frac{p_j}{n} \right) \leq \varphi \left( \sum_{\substack{j \in J \\ j \geq i}} \frac{p_j}{n} \right).$$

Ainsi le premier terme de  $RDU(L)$  est inférieur ou égal au premier terme de  $RDU(L')$ . On vérifie facilement que :

$$\varphi \left( \sum_{\substack{j \in I \\ j \geq k}} \frac{p_j}{n} \right) = p_{succ_I(k)-1} \text{ et } \varphi \left( \sum_{\substack{j \in J \\ j \geq k}} \frac{p_j}{n} \right) = p_{prec_J(k)} = p_{k-1}$$

où  $succ_I(i) = \min\{j \in I : j > i\}$ . Or  $p_{succ_I(k)-1} < p_{k-1}$  car  $succ_I(k) - 1 > k - 1$ . Donc le second terme de  $RDU(L)$  est strictement inférieur au second terme de  $RDU(L')$ . Enfin, le troisième terme de  $RDU(L)$  est bien évidemment égal au troisième terme de  $RDU(L')$ . Par conséquent  $RDU(L) < RDU(L')$ .

On conclut de (i),(ii) et (iii) que toute stratégie correspondant à une instantiation non-satisfaisante présente une valeur RDU strictement inférieure à  $m$ , et que toute stratégie correspondant à une instantiation satisfaisante présente une valeur RDU exactement égale à  $m$ . Trouver une instantiation satisfaisante dans 3-SAT revient donc à trouver une stratégie valant  $m$  dans RDU-OPT. ■

Puisque le problème RDU-OPT est NP-difficile, il est nécessaire de développer des algorithmes exponentiels performants pour résoudre le problème. C'est justement le propos des deux prochaines parties.

### 3.2 Approche par programmation mathématique

Nous présentons ici une première approche, par programmation linéaire en variables mixtes, pour résoudre le problème RDU-OPT. Notons que l'approche présentée ici n'est

valide que pour une classe précise de fonctions  $\varphi$  de déformation des probabilités : la fonction  $\varphi$  doit être linéaire par morceau et concave. Rappelons qu'une fonction concave de déformation des probabilités traduit un comportement de goût pour le risque (en supposant affine la fonction d'utilité).

Afin d'alléger les notations et la formulation du programme mathématique, nous supposons ici que l'arbre de décision  $T$  est de profondeur paire, et que tout chemin de la racine (nœud de décision  $D_1$ ) vers un nœud terminal alterne les nœuds de décision (notés  $D_i$  avec  $i \in \{1, \dots, |\mathcal{N}_D|\}$ ) et les nœuds de chance (notés  $L_i$  avec  $i \in \{1, \dots, |\mathcal{N}_C|\}$ ). Cela nous permet alors, sans ambiguïté, de noter  $(i, j)$  les arcs  $(D_i, L_j)$  issus des nœuds de décision, appelés aussi *arcs décisionnels* par la suite.

Dans un premier temps, nous détaillons un ensemble de contraintes permettant de définir l'ensemble de toutes les stratégies d'un arbre de décision  $T$ . Pour ce faire, chaque arc  $(i, j)$  issu d'un nœud de décision donne naissance à une variable booléenne  $y_{(i,j)}$ . Les  $|\mathcal{N}_D|$  contraintes linéaires suivantes définissent l'ensemble des stratégies réalisables de l'arbre de décision  $T$  :

$$\begin{aligned} \sum_j y_{(1,j)} &= 1 \\ \sum_j y_{(i,j)} &= y_{prev_D(i)} \quad \forall i \in \{2, \dots, |\mathcal{N}_D|\} \end{aligned}$$

où  $y_{(i,j)} = 1$  (resp.  $y_{(i,j)} = 0$ ) si la décision  $(i, j)$  appartient (resp. n'appartient pas) à la stratégie. La fonction  $prev_D(i)$  retourne l'arc issu du plus proche nœud de décision ancêtre de  $D_i$ .

De cette manière, toute affectation des variables  $y_{(i,j)}$  qui vérifie ces contraintes correspond à une stratégie de  $T$ , et toute stratégie de  $T$  correspond à une affectation des variables  $y_{(i,j)}$  qui vérifie ces contraintes.

**Exemple 10** Pour l'arbre de décision représenté sur la figure 3.2 page 88, les contraintes qui définissent l'ensemble des stratégies réalisables de l'arbre sont les suivantes :

$$\begin{aligned} y_{(1,1)} + y_{(1,2)} &= 1 \\ y_{(2,3)} + y_{(2,4)} &= y_{(1,1)} \\ y_{(3,5)} + y_{(3,6)} &= y_{(1,1)} \\ y_{(4,7)} + y_{(4,8)} &= y_{(1,2)} \\ y_{(5,9)} + y_{(5,10)} &= y_{(1,2)} \end{aligned}$$

Nous détaillons maintenant la formulation de la fonction objectif que l'on désire maximiser. L'ensemble des utilités de l'arbre de décision  $T$  est noté  $U = \{u_1, u_2, \dots, u_n\}$ , avec  $u_1 \leq u_2 \leq \dots \leq u_n$ , et la probabilité d'obtenir l'utilité  $u_h$  est notée  $p_h$ . La probabilité  $p_h$

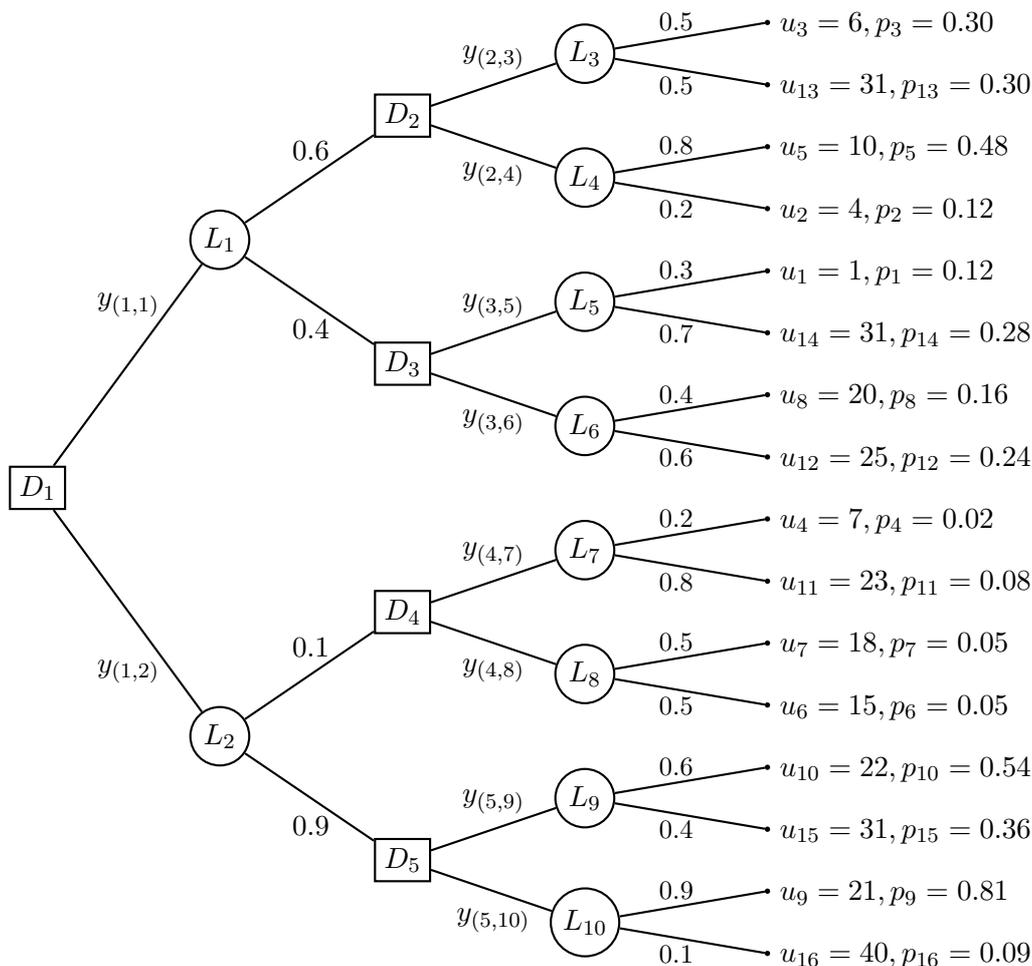


FIGURE 3.2 – Arbre de décision et les différentes variables et paramètres figurant dans le programme mathématique.

correspond au produit de toutes les probabilités rencontrées sur le chemin de la racine au nœud d'utilité offrant  $u_h$ . Le critère RDU peut alors s'écrire comme suit :

$$u_1 + \sum_{h=2}^n (u_h - u_{h-1}) \varphi \left( \sum_{j \geq h} p_j y_{prev_u(j)} \right)$$

où  $prev_u(j)$  est la fonction qui retourne l'arc issu du plus proche nœud de décision ancêtre du nœud d'utilité offrant l'utilité  $u_j$ . L'introduction de  $(n - 1)$  variables  $\varphi_h$ , permet alors de reformuler cette fonction objectif de la façon suivante :

$$u_1 + \sum_{h=2}^n (u_h - u_{h-1}) \varphi_h, \text{ où } \forall h \in \{2, \dots, n\}, \varphi_h = \varphi \left( \sum_{j \geq h} p_j y_{prev_u(j)} \right)$$

**Exemple 11** Considérons l'arbre de décision représenté sur la figure 3.2, la fonction objectif vaut  $1 + (4 - 1)\varphi_2 + \dots + (31 - 25)\varphi_{13} + (31 - 31)\varphi_{14} + (31 - 31)\varphi_{15} + (40 - 31)\varphi_{16}$ ,

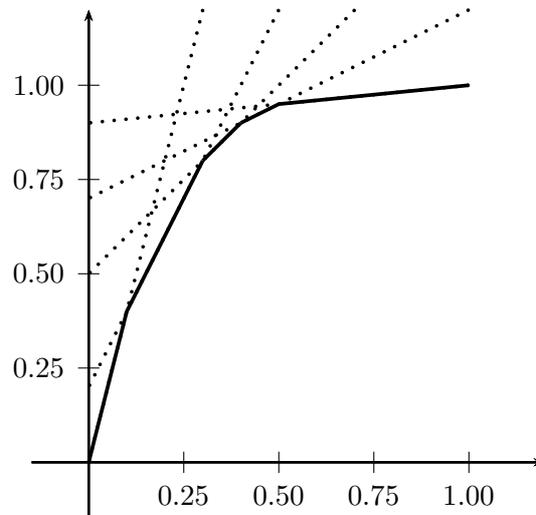


FIGURE 3.3 – L’enveloppe inférieure d’un ensemble de fonction affines est une fonction concave et linéaire par morceaux.

ce qui est équivalent à :

$$1 + 3\varphi_2 + 2\varphi_3 + \varphi_4 + 3\varphi_5 + 5\varphi_6 + 3\varphi_7 + 2\varphi_8 + \varphi_9 + \varphi_{10} + 2\varphi_{11} + 2\varphi_{12} + 6\varphi_{13} + 9\varphi_{16}$$

Notons que les variables  $\varphi_{14}$  et  $\varphi_{15}$ , n’apparaissant pas dans la fonction objective, peuvent être retirées du programme mathématique.

Les contraintes liées à  $\varphi_h$  ne sont bien sûr pas linéaires, du fait de la présence de la fonction  $\varphi$  qui n’est pas affine. Nous nous intéressons maintenant à la linéarisation de la fonction  $\varphi$ .

La linéarisation que nous proposons ici traite uniquement le cas des fonctions  $\varphi$  qui sont concaves et linéaires par morceaux. Une fonction concave et linéaire par morceaux peut être décrite par l’enveloppe inférieure d’un ensemble de fonctions affines. En effet, soit un ensemble  $\{f_1, f_2, \dots, f_m\}$  de fonctions avec  $\forall k \in \{1, 2, \dots, m\}, f_k(p) = a_k p + b_k$  où  $(a_k, b_k) \in \mathbb{R}^2$ , la fonction  $\varphi(p) = \min\{f_1(p), f_2(p), \dots, f_m(p)\}$  est une fonction concave et linéaire par morceaux. En guise d’illustration, nous avons représenté en pointillés sur la figure 3.3 les fonctions de l’ensemble  $\{f_1(x), \dots, f_5(x)\}$  avec :  $f_1(x) = 4x$ ,  $f_2(x) = 2x + 0.2$ ,  $f_3(x) = x + 0.5$ ,  $f_4(x) = \frac{1}{2}x + 0.7$ ,  $f_5(x) = \frac{1}{9}x + 0.9$ . L’enveloppe inférieure de cet ensemble de fonction a été représentée en gras et correspond bien à une fonction concave et linéaire par morceaux.

Lorsque la fonction  $\varphi$  est concave et linéaire par morceaux, la valeur de  $\varphi_h$  peut alors

s'obtenir en optimisant le programme linéaire suivant :

$$\begin{aligned}\varphi_h &= \max_{\alpha} \alpha \\ \alpha &\leq f_k \left( \sum_{j \geq h} p_j y_{prev_u(j)} \right) \quad \forall k \in \{1, \dots, m\} \\ \alpha &\geq 0\end{aligned}$$

**Exemple 12** Supposons que  $\varphi(p) = 1.8p$  pour  $p \leq 0.5$ , et  $\varphi(p) = 0.4p + 0.6$  pour  $p > 0.5$ . On a alors  $\varphi(p) = \min\{f_1(p), f_2(p)\}$  avec  $f_1(p) = 1.8p$  et  $f_2(p) = 0.4p + 0.6$ . Etant donnée une instanciation valide des variables booléennes  $y_{(i,j)}$  de l'arbre de décision représenté sur la figure 3.2 page 88, la valeur de  $\varphi_{13}$  peut alors s'exprimer comme l'optimum du programme linéaire suivant :

$$\begin{aligned}\varphi_{13} &= \max_{\alpha} \alpha \\ \alpha &\leq 1.8(0.30y_{(2,3)} + 0.28y_{(3,5)} + 0.36y_{(5,9)} + 0.09y_{(5,10)}) \\ \alpha &\leq 0.4(0.30y_{(2,3)} + 0.28y_{(3,5)} + 0.36y_{(5,9)} + 0.09y_{(5,10)}) + 0.6 \\ \alpha &\geq 0\end{aligned}$$

Puisque  $u_h - u_{h-1} \geq 0$ , la maximisation de la fonction objectif implique la maximisation de chacun des  $\varphi_h$ . La formulation du programme linéaire en variables mixtes pour optimiser RDU dans un arbre de décision s'écrit donc comme suit :

$$\begin{aligned}\max_{\varphi} & u_1 + \sum_{h=2}^n (u_h - u_{h-1}) \varphi_h \\ \varphi_h &\leq f_k \left( \sum_{j \geq h} p_j y_{prev_u(j)} \right) \quad \forall h \in \{2, \dots, n\}, \forall k \in \{1, \dots, m\} \\ \sum_j y_{(1,j)} &= 1 \\ \sum_j y_{(i,j)} &= y_{prev_D(i)} \quad \forall i \in \{2, \dots, |\mathcal{N}_D|\} \\ y_{(i,j)} &\in \{0, 1\} \quad \varphi_h \geq 0\end{aligned}$$

Ce programme mathématique comporte  $(n - 1)$  variables réelles,  $|\mathcal{N}_D|$  variables booléennes et  $(n - 1)m + |\mathcal{N}_D|$  contraintes (puisque  $m$  contraintes sont créées pour chaque variable  $\varphi_h$ ). La taille de ce programme mathématique est donc linéaire en la taille de l'arbre de décision  $T$  pour un nombre  $m$  fixé de morceaux caractérisant la fonction  $\varphi$ . Rappelons cependant que la complexité, dans le pire des cas, de l'optimisation d'un programme linéaire en variables mixtes, est exponentielle dans le nombre de variables entières.

### 3.3 Approche par énumération implicite

Dans cette partie, nous présentons une méthode basée sur le principe du *branch and bound* pour résoudre le problème RDU-OPT. Contrairement à l'approche précédente, cette méthode reste valide pour n'importe quelle fonction  $\varphi$  de déformation des probabilités (i.e., la fonction  $\varphi$  n'est plus restreinte aux fonctions concaves et linéaires par morceaux).

Le principe de cette méthode consiste à partitionner l'ensemble des stratégies en plusieurs sous-ensembles de stratégies. Chaque sous-ensemble est différencié par la décision  $(N, N')$  prise au nœud de décision  $N$ . De manière plus formelle, les nœuds de l'arbre d'énumération sont caractérisés par une *stratégie partielle*. Une stratégie partielle permet de définir un sous-ensemble de stratégies. Considérons un arbre de décision  $T$  et un ensemble  $\mathcal{N}^\Gamma$  de nœuds incluant :

- la racine  $N_r \in \mathcal{N}_D$  de  $T$ ,
- un fils exactement pour chaque nœud  $N \in \mathcal{N}_D^\Gamma = \mathcal{N}_D \cap \mathcal{N}^\Gamma$ .

L'ensemble des arcs  $\Gamma = \{(N, N') : N \in \mathcal{N}_D^\Gamma, N' \in \mathcal{N}^\Gamma\} \subseteq \mathcal{E}$  définit une *stratégie partielle* de l'arbre  $T$  dès lors que le sous-graphe induit par  $\mathcal{N}^\Gamma$  est un arbre. Une stratégie  $\Delta$  est dite *compatible* avec la stratégie partielle  $\Gamma$  si et seulement si  $\Gamma \subseteq \Delta$ . Le sous-ensemble des stratégies caractérisées par la stratégie partielle  $\Gamma$  correspond à l'ensemble des stratégies compatibles avec  $\Gamma$ .

En chaque nœud de l'arbre d'énumération, on fixe un arc parmi les arcs issus d'un nœud de décision donné. L'ordre dans lequel les nœuds de décision sont parcourus est donné par une fonction de rang  $rg : \mathcal{N}_D \rightarrow \{1, 2, \dots, |\mathcal{N}_D|\}$ . Si plusieurs nœuds de décision sont candidats<sup>4</sup> pour compléter  $\mathcal{N}^\Gamma$ , on choisit celui dont le rang est le plus bas. La fonction  $rg$  est définie comme suit :

$$\begin{cases} rg(N_r) = 1 \\ |past(N)| > |past(N')| \Rightarrow rg(N) > rg(N') \\ |past(N)| = |past(N')| \text{ et } EU(\mathcal{T}(N)) > EU(\mathcal{T}(N')) \Rightarrow rg(N) < rg(N') \end{cases}$$

où  $EU(\mathcal{T}(N))$  est la valeur optimale du critère EU dans le sous-arbre  $\mathcal{T}(N)$  (rappelons que  $\mathcal{T}(N)$  est le sous-arbre de  $T$  ayant pour racine  $N$ ) et  $past(N)$  est l'ensemble des nœuds rencontrés sur le chemin de la racine à  $N$  ( $N$  exclu).

**Exemple 13** *Considérons l'arbre de décision représenté sur la figure 3.2, il existe une unique fonction de rang  $rg$  définie par :  $rg(D_1) = 1$ ,  $rg(D_2) = 5$ ,  $rg(D_3) = 3$ ,  $rg(D_4) = 4$  et  $rg(D_5) = 2$  (car  $EU(\mathcal{T}(D_5)) < EU(\mathcal{T}(D_3)) < EU(\mathcal{T}(D_4)) < EU(\mathcal{T}(D_2))$ ). Nous avons représenté l'arbre d'énumération correspondant dans la figure 3.4 qui suit.*

---

4. Un nœud  $N^c$  est candidat pour compléter une stratégie partielle  $\Gamma$  si et seulement si l'ensemble  $\{(N, N') : N \in \mathcal{N}_D^\Gamma \cup \{N^c\}, N' \in \mathcal{N}^\Gamma \cup \{N^c\}\}$  est une stratégie partielle.

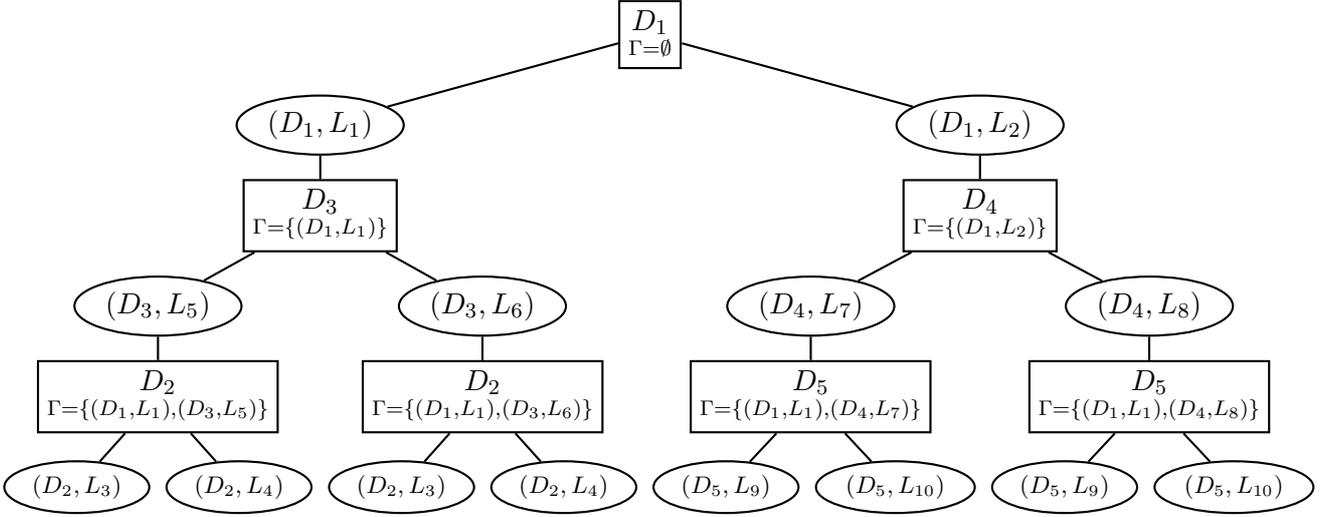


FIGURE 3.4 – Arbre d'énumération correspondant à l'arbre de décision représenté sur la figure 3.2.

L'algorithme 3 page 93 décrit formellement, sous forme récursive, la procédure d'énumération implicite que nous proposons ici. Elle prend en argument une stratégie partielle  $\Gamma$ , ainsi que la meilleure valeur selon RDU trouvée jusqu'à présent (notée  $RDU_{opt}$ ). La recherche se fait en profondeur d'abord dans l'arbre d'énumération. Les nœuds de décision candidats pour venir compléter la stratégie partielle  $\Gamma$  (i.e., pour entrer dans  $\mathcal{N}^\Gamma$ ) sont notés  $\mathcal{N}_1$ . Parmi les nœuds de  $\mathcal{N}_1$ , on note  $N_{min}$  celui dont le rang est minimal.

L'ensemble des arcs décisionnels issus de  $N_{min}$  est noté  $\mathcal{E}_{min}$ . Cet ensemble définit l'ensemble des extensions possibles de  $\Gamma$  que l'on envisage dans la recherche (en d'autres termes, les fils du nœud associé à  $\Gamma$  dans l'arbre d'énumération). Pour chacune de ces extensions possibles de  $\Gamma$  (i.e., en chaque fils du nœud courant de l'arbre d'énumération), on calcule, par l'intermédiaire de la fonction  $ev^-$  (resp.  $ev^+$ ), une borne inférieure (resp. supérieure) de la meilleure évaluation, selon le critère RDU, des stratégies compatibles avec l'extension de  $\Gamma$  envisagée. La valeur retournée par la fonction  $ev^-$  correspond à l'évaluation, selon le critère RDU, d'une stratégie réalisable compatible avec l'extension de  $\Gamma$  envisagée, tandis que celle retournée par  $ev^+$  ne correspond à aucune stratégie réalisable particulière. On met donc à jour la valeur de  $RDU_{opt}$  si celle retournée par  $ev^-$  est supérieure. Comme il est d'usage dans un branch and bound, la valeur retournée par la fonction  $ev^+$  nous indique s'il faut continuer à explorer, en profondeur, l'arbre d'énumération, ou si l'on peut couper l'ensemble des stratégies compatibles avec l'extension de la stratégie partielle (i.e., la stratégie optimisant le critère RDU ne fait pas partie de l'ensemble de stratégies caractérisé par cette extension). L'optimalité de la valeur  $RDU_{opt}$

**Algorithme 3 : BB**(Stratégie partielle  $\Gamma$ , Réel  $RDU_{opt}$ )

**Sortie :** La valeur RDU optimale parmi les stratégies compatibles avec  $\Gamma$

$\mathcal{N}_1 \leftarrow \{N_1 \in \mathcal{N}_D : N_1 \text{ est candidat}\};$

$N_{min} \leftarrow \arg \min_{N \in \mathcal{N}_1} rg(N);$

$\mathcal{E}_{min} \leftarrow \{(N_{min}, A) \in \mathcal{E} : A \in \text{fils}(N_{min})\};$

**pour chaque**  $(N, A) \in \mathcal{E}_{min}$  **faire**

$RDU_{opt} \leftarrow \max \{RDU_{opt}, ev^-(\Gamma \cup \{(N, A)\})\};$

**si**  $ev^+(\Gamma \cup \{(N, A)\}) > RDU_{opt}$  **alors**

$RDU_{temp} \leftarrow \mathbf{BB}(\Gamma \cup \{(N, A)\}, RDU_{opt});$

**si**  $RDU_{temp} > RDU_{opt}$  **alors**

$RDU_{opt} \leftarrow RDU_{temp};$

**Retourner**  $RDU_{opt}$

retournée est garantie puisque seules les sous-stratégies sous-optimales sont ignorées dans la recherche (car  $ev^+$  retourne un borne supérieure).

Il nous reste maintenant à détailler les aspects clefs de la méthode de branch and bound proposée ici : l'initialisation, le calcul de la borne inférieure et le calcul de la borne supérieure.

**Initialisation**

L'initialisation consiste à déterminer une stratégie initiale. La valeur de cette stratégie selon le critère RDU permet d'initialiser la valeur de  $RDU_{opt}$ . Il est bien connu que plus la solution initiale est de bonne qualité, plus la méthode de branch and bound s'avère efficace. En effet, une bonne valeur initiale, permet de couper très rapidement d'importants sous-ensembles de solutions. Pour initialiser notre algorithme de branch and bound, nous proposons ici de démarrer avec la stratégie optimisant le critère EU. De cette manière, la valeur de  $RDU_{opt}$  est initialement égale à la valeur RDU de la stratégie optimisant le critère EU.

**Calcul de la borne inférieure**

En chaque nœud de l'arbre d'énumération (caractérisé par la stratégie partielle  $\Gamma$ ), on calcule la valeur  $V$ , selon le critère RDU, de la stratégie optimisant le critère EU parmi celles qui sont compatibles avec  $\Gamma$ . Ce calcul est réalisé par la fonction  $ev^-$  dans l'algorithme. Lorsque la valeur RDU de cette stratégie est plus élevée que la meilleure valeur de  $RDU_{opt}$  trouvée jusqu'alors, on met à jour la valeur de  $RDU_{opt}$ .

### Calcul de la borne supérieure

La fonction permettant de calculer une borne supérieure est notée  $ev^+$ . Elle retourne une borne supérieure de toutes les évaluations par le critère RDU des stratégies compatibles avec la stratégie partielle  $\Gamma$ , qu'elle prend en argument. Le principe de cette évaluation est de construire une loterie, que l'on nommera par la suite *loterie dominante*, qui domine stochastiquement au premier ordre toutes les loteries correspondantes à des stratégies compatibles avec  $\Gamma$ . Ensuite, on évalue, avec le critère RDU, la valeur de cette loterie dominante. Cette valeur fournit bien une borne supérieure puisque le critère RDU est compatible avec la dominance stochastique au premier ordre : si la loterie  $L$  domine stochastiquement au premier ordre la loterie  $L'$ , alors  $RDU(L) \geq RDU(L')$ .

Afin de construire une telle loterie, on procède par programmation dynamique dans l'arbre de décision. Notons que l'on peut manipuler indifféremment une loterie et sa fonction décumulative puisqu'à chaque loterie correspond une unique fonction décumulative et que seule la fonction décumulative nous intéresse lorsque l'on évalue la valeur d'une loterie avec le critère RDU. Ainsi, nous ne construisons pas une loterie dominante mais une fonction décumulative dominante qui correspond à une loterie dominante.

L'initialisation est réalisée comme suit : à chaque nœud d'utilité  $N_T \in \mathcal{C}$  on associe la fonction décumulative  $G_{LN_T} = (1, u(N_T))$ . Puis, par induction arrière, on remonte à la racine une fonction décumulative correspondant à une loterie dominante.

En chaque nœud de chance  $A \in \mathcal{N}_C$ , on associe la fonction décumulative  $G_{LA}$  induite par les fonctions décumulatives associées à chacun des fils de  $A$ . Plus précisément,  $G_{LA}$  se calcule comme suit :

$$\forall u, G_{LA}(u) = \sum_{N \in \text{fils}(A)} P((A, N)) G_{LN}(u)$$

où  $G_{LN}$  correspond à la fonction décumulative assignée au nœud  $N \in \mathcal{N}$  et  $P((A, N))$  est la probabilité associée à l'arc  $(A, N)$  dans l'arbre de décision.

En chaque nœud de décision  $D \in \mathcal{N}_D$ , on calcule une fonction décumulative  $G_{LD}$  selon les relations de récurrence suivantes :

$$\begin{cases} \forall u, G_{LD}(u) = G_{LN}(u) \text{ si } \exists N \in \text{fils}(D) : (D, N) \in \Gamma \\ \forall u, G_{LD}(u) = \max_{N \in \text{fils}(D)} G_{LN}(u) \text{ sinon} \end{cases}$$

Enfin, la valeur retournée par la fonction  $ev^+$  est  $RDU(G_{LN_r})$  où  $G_{LN_r}$  correspond à la fonction décumulative construite en la racine de l'arbre.

La complexité de cette procédure récursive pour  $\Gamma = \emptyset$  est en  $O(|\mathcal{N}| \cdot |U|)$  (où  $|U|$  est le nombre de valeurs d'utilités distinctes présentes dans l'arbre de décision) puisque chaque nœud de  $\mathcal{N}$  est examiné une fois et que le "support" des fonctions décumulatives

calculées est borné par  $|U|$ . Néanmoins, notons que durant la procédure de branch and bound, lorsque l'on ajoute un arc  $(D_i, N_j)$  à  $\Gamma$ , il n'est pas nécessaire de recalculer les fonctions décumulatives  $G_{LN}$  en chaque nœud  $N \in \mathcal{N}$ . En effet, il est possible d'utiliser les fonctions décumulatives  $G_{LN}$  déjà calculées lors du calcul de  $ev^+(\Gamma)$ . Il suffit pour cela de mettre à jour uniquement les fonctions décumulatives susceptibles d'avoir changées, c'est-à-dire celles assignées aux nœuds rencontrés sur le chemin de la racine au nœud  $D_i$ . Puisque la longueur d'un chemin de  $T$  est bornée par la profondeur  $p$  de l'arbre, la complexité de la procédure du calcul de la valeur de  $ev^+(\Gamma)$  pour  $\Gamma \neq \emptyset$  est alors en  $O(p \cdot |U|)$ .

Pour conclure, il nous reste à prouver la validité de cette procédure récursive. Pour cela, on procède par induction arrière dans l'arbre de décision :

- *En un nœud chance*  $A \in \mathcal{N}_C$  : considérons un ensemble  $\{L_N : N \in \text{fils}(A)\}$  de loteries tel que  $\forall N \in \text{fils}(A)$ ,  $L^N$  domine stochastiquement au premier ordre  $L_N$ . On a alors :

$$\forall u, G_{LA}(u) = \sum_{N \in \text{fils}(A)} p((A, N)) G_{LN}(u) \geq \sum_{N \in \text{fils}(A)} p((A, N)) G_{LN}(u)$$

Ainsi, on assure que la loterie  $L^A$  domine stochastiquement au premier ordre toutes les loteries induites par une stratégie de  $T(A)$ .

- *En un nœud de décision*  $D$  : s'il existe  $N \in \text{fils}(D)$  tel que  $(D, N) \in \Gamma$ , alors on est dans un cas trivial. Sinon, considérons un ensemble  $\{L_N : N \in \text{fils}(A)\}$  de loteries tel que  $\forall N \in \text{fils}(A)$ ,  $L^N$  domine stochastiquement au premier ordre  $L_N$ . Par définition on a :

$$\forall u, G_{LD}(u) = \max_{N \in \text{fils}(D)} G_{LN}(u)$$

On en déduit alors que

$$\forall N \in \text{fils}(D), \forall u, G_{LD}(u) \geq G_{LN}(u) \geq G_{LN}(u)$$

Ainsi, on assure que la loterie  $L^D$  domine stochastiquement au premier ordre toutes les loteries induites par une stratégie de  $T(A)$ .

Par conséquent, la loterie  $L^{N_r}$  domine stochastiquement au premier ordre toutes les loteries induites par une stratégie compatible avec  $\Gamma$ .

**Exemple 14** *Considérons l'arbre de décision représenté sur la figure 3.2. Supposons que  $\Gamma = \{(D_1, A_1), (D_2, A_3)\}$ . Les fonctions décumulatives assignées aux différents nœuds sont :*

- *En  $A_3$  :  $G_{LA_3} = (6, 1; 31, 0.5)$ .*

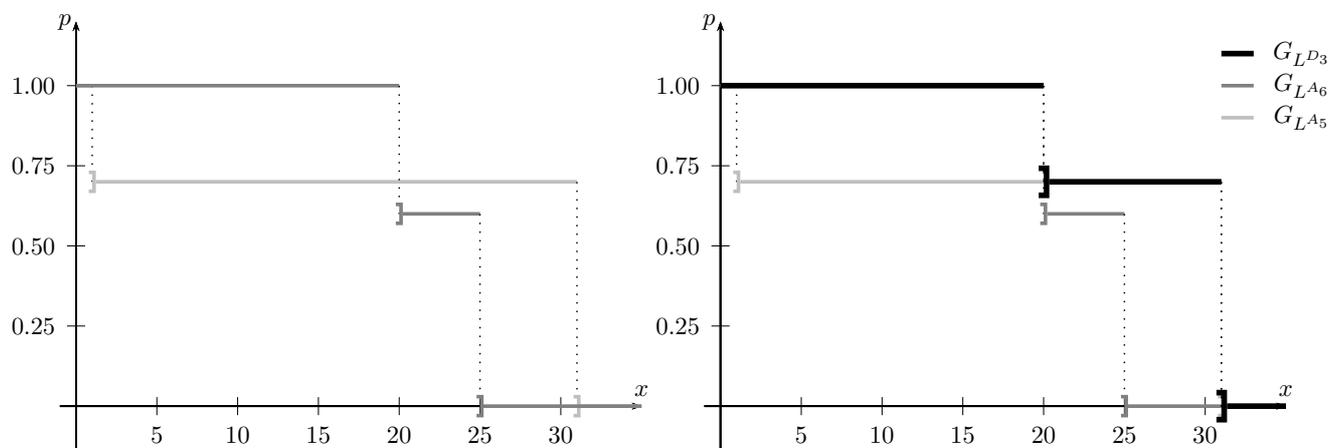


FIGURE 3.5 – Représentation graphique des fonctions décumulatives  $G_{L^{A_5}}$ ,  $G_{L^{A_6}}$  et  $G_{L^{D_3}}$  lors du calcul de la borne supérieure dans l'arbre de décision représenté sur la figure 3.2 avec  $\Gamma = \{(D_1, A_1), (D_2, A_3)\}$  (voir exemple 14).

- En  $A_4$  :  $G_{L^{A_4}} = (4, 1; 10, 0.8)$ .
- En  $A_5$  :  $G_{L^{A_5}} = (1, 1; 31, 0.7)$ . Cette fonction décumulative est représentée en gris clair sur la figure 3.5.
- En  $A_6$  :  $G_{L^{A_6}} = (20, 1; 25, 0.6)$ . Cette fonction décumulative est représentée en gris sur la figure 3.5.
- En  $D_2$  :  $G_{L^{D_2}} = G_{L^{A_3}}$  car  $(D_2, A_3) \in \Gamma$ .
- En  $D_3$  : la fonction décumulative assignée au nœud  $D_3$  est l'enveloppe supérieure des fonctions décumulatives  $G_{L^{A_5}}$  et  $G_{L^{A_6}}$ . Plus précisément, elle est égale à :

$$\forall u, G_{L^{D_3}}(u) = \max\{G_{L^{A_5}}(u), G_{L^{A_6}}(u)\}$$

Cette fonction décumulative est représentée en noir sur la figure 3.5. On a donc  $G_{L^{D_3}} = (20, 1; 31, 0.7)$ .

- En  $A_1$  :  $G_{L^{A_1}} = 0.6G_{L^{D_2}} + 0.4G_{L^{D_3}} = (6, 1; 20, 0.7; 31, 0.59)$ .
- En  $D_1$  :  $G_{L^{D_1}} = G_{L^{A_1}}$  car  $(D_1, A_1) \in \Gamma$ .

La borne supérieure de l'ensemble des stratégies caractérisées par  $\Gamma$  vaut alors  $ev^+(\Gamma) = RDU(L^{D_1})$ .

### 3.4 Résultats expérimentaux

Pour les expérimentations numériques, tous les algorithmes ont été programmés avec le langage C++. Ces algorithmes ont été exécutés sur un ordinateur doté d'un processeur

Profondeur (nœuds) \ Méthode	MIP			B&B		
	min	moy	max	min	moy	max
4 (31)	< 0.1	< 0.1	< 0.1	< 0.1	< 0.1	< 0.1
6 (127)	< 0.1	< 0.1	< 0.1	< 0.1	< 0.1	< 0.1
8 (511)	< 0.1	< 0.1	< 0.1	< 0.1	< 0.1	< 0.1
10 (2047)	2.1	2.3	2.8	< 0.1	2.5	5.0
12 (8191)	79.4	86.0	107.3	0.3	5.6	20.2

TABLE 3.1 – Temps CPU (en secondes) d'exécution des algorithmes pour une fonction  $\varphi$  concave et linéaire par morceaux.

Pentium IV CPU 2.13GHz et de 3.5Go de RAM.

Dans un premier temps, nous avons comparé les performances de l'approche par programmation linéaire en variables mixtes et de l'approche par énumération implicite.

Les instances, ayant servies à la comparaison des performances entre les deux approches, ont été générées aléatoirement mais la structure des arbres de décision est restée toujours la même.

Plus précisément, nous avons imposé une structure d'arbre binaire complet pour les arbres de décision. C'est à dire que, pour un arbre de profondeur  $p$ , les nœuds de profondeur strictement inférieure à  $p$  ont exactement deux fils et les nœuds de profondeur  $p$  n'ont aucun fils (ce sont les feuilles de l'arbre).

Nous avons de plus imposé que la profondeur  $p$  de l'arbre soit toujours paire, que les nœuds de profondeur paire strictement inférieure à  $p$  soient des nœuds de décision, que les nœuds de profondeur impaire soient des nœuds de chance et que les nœuds de profondeur  $p$  soient des nœuds d'utilité. Cette alternance de nœuds de décision et de nœuds de chance nous assure de générer des arbres de décision dont le nombre de stratégies est exponentiel en la taille de l'arbre.

Les valeurs d'utilité aux feuilles sont des nombres réels, générés aléatoirement, compris dans l'intervalle  $[1; 1000]$ . Les probabilités conditionnelles, assignées à chaque arc issu d'un nœud de chance, sont des réels positifs, générés aléatoirement, qui somment bien à 1 au sein d'un même nœud de chance.

La fonction  $\varphi$  utilisée pour faire les tests est la suivante :  $\varphi(p) = \min\{f_1(x), \dots, f_5(x)\}$  avec :  $f_1(x) = 4x$ ,  $f_2(x) = 2x + 0.2$ ,  $f_3(x) = x + 0.5$ ,  $f_4(x) = \frac{1}{2}x + 0.7$ ,  $f_5(x) = \frac{1}{9}x + 0.9$ . La fonction  $\varphi$  est bien concave et linéaire par morceaux (conditions requises pour la formulation en programme linéaire en nombre entier).

Pour la résolution des programmes linéaires en nombre entier nous avons utilisé le solveur ILOG CPLEX v11.1.0.

Le tableau 3.1 référence les temps CPU d'exécution (en secondes) des algorithmes ob-

Prof. (nœuds)	$\gamma = 0.2$			$\gamma = 0.5$			$\gamma = 0.8$		
	min	moy	max	min	moy	max	min	moy	max
4 (31)	< 0.1	< 0.1	< 0.1	< 0.1	< 0.1	< 0.1	< 0.1	< 0.1	< 0.1
6 (127)	< 0.1	< 0.1	< 0.1	< 0.1	< 0.1	< 0.1	< 0.1	< 0.1	< 0.1
8 (511)	< 0.1	< 0.1	< 0.1	< 0.1	< 0.1	< 0.1	< 0.1	< 0.1	< 0.1
10 (2047)	< 0.1	0.4	1.1	< 0.1	0.1	0.5	< 0.1	0.1	0.5
12 (8191)	0.3	4.3	13.8	0.1	2.7	7.4	0.1	1.6	8.0
14 (32767)	3.0	62.1	512.9	1.2	17.3	99.4	0.6	10.9	101.4

TABLE 3.2 – Temps CPU (en secondes) d'exécution de l'algorithme par énumération implicite en fonction de la profondeur et du paramètre  $\gamma$ .

tenus par les deux approches en fonction de la profondeur des arbres de décision générés : colonnes « MIP » (pour *Mixed Integer Programming*) pour l'approche par programmation linéaire en variables mixtes et colonnes « B&B » pour la méthode par énumération implicite.

Pour chaque profondeur (4, 6, 8, 10 et 12), 50 instances ont été générées. Les temps représentés, dans les colonnes intitulées « moy », représentent la moyenne, sur les 50 instances, des temps d'exécution obtenus. Dans les colonnes intitulées « min » (resp. « max »), nous avons indiqué le temps minimal (resp. maximal) atteint par au moins une instance parmi les 50. Précisons que les temps de résolution indiqués pour l'approche par formulation avec un programme linéaire en nombre entier, ne tiennent pas compte des temps de *pré-traitement* (preprocessing time) de CPLEX.

Au regard de ces résultats, nous pouvons constater que les deux approches fournissent instantanément la stratégie optimale pour les arbres de décision de profondeur inférieure ou égale à 8. Cependant, lorsque cette profondeur est supérieure ou égale à 12, la méthode par énumération implicite fournit des temps nettement plus compétitifs que la formulation en programme linéaire en variables mixtes.

Dans un second temps, nous avons voulu étudier plus spécifiquement les performances de l'algorithme d'énumération implicite. Nous nous sommes donc intéressés à observer l'influence de la forme de la fonction  $\varphi$  de déformation des probabilités, ainsi que l'influence de la profondeur de l'arbre de décision, sur le temps CPU de résolution. Nous avons choisi la forme suivante pour paramétrer la fonction de déformation des probabilités :  $\forall p \in [0; 1], \varphi(p) = p^\gamma / (p^\gamma + (1 - p)^\gamma)$  avec  $\gamma \in ]0; 1]$ . Cette forme fonctionnelle est l'une des formes proposées dans la littérature (Karmarkar, 1979) pour la description d'un décideur (voir chapitre 1). Elle permet de caractériser un effet de certitude par la sous-évaluation des faibles conséquences et un effet de potentiel par la surévaluation des très bonne conséquences. Nous avons ensuite fait varier la valeur de  $\gamma$ . Plus la valeur de  $\gamma$  est faible, plus on « s'éloigne » d'un traitement linéaire des probabilités (i.e., plus on

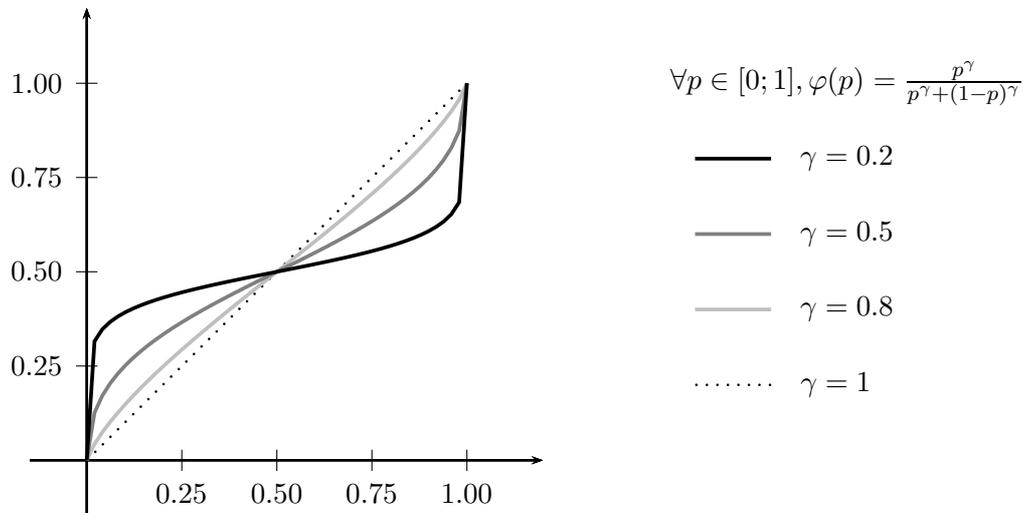


FIGURE 3.6 – Allure de la courbe représentative, pour différentes valeurs de  $\gamma$ , de la fonction  $\varphi$  de déformation des probabilités suivante :  $\forall p \in [0; 1], \varphi(p) = p^\gamma / (p^\gamma + (1-p)^\gamma)$ .

s'éloigne du modèle EU). En effet, le cas  $\gamma = 1$  correspond à une fonction  $\varphi$  identité, et le cas où  $\gamma$  tend vers 0 à valeur positive correspond à une fonction s'écartant énormément de la fonction identité. Afin que le lecteur puisse être à même de se faire une opinion sur la distance de cette fonction avec la fonction identité, nous avons représenté l'allure de cette fonction dans la figure 3.6 pour plusieurs valeurs de  $\gamma$ .

Les différentes valeurs de  $\gamma$  que nous avons sélectionnées pour faire nos expérimentations numériques sont  $\gamma = 0.2$ ,  $\gamma = 0.5$  et  $\gamma = 0.8$ . Les arbres de décision utilisés pour ces expérimentations numériques ont été générés de la même manière que ceux de l'expérience précédente (arbres binaires complets avec alternance des nœuds de décision et des nœuds de chance). Pour chaque jeu de paramètres, 50 arbres ont été générés.

Le tableau 3.2 page 98 présente les performances de l'algorithme en fonction de la profondeur de l'arbre de décision et de la valeur du paramètre  $\gamma$ . Les colonnes intitulées « moy » donnent la moyenne, sur 50 instances générées aléatoirement, du temps de résolution (en secondes). Les colonnes intitulées « min » (resp. « max ») donnent le temps minimal (resp. maximal) rencontré pour au moins une instance parmi les 50. Sans surprise, on constate que, plus  $\gamma$  est proche de 1 (i.e., plus le critère RDU est proche du critère EU), plus les temps sont bas. Il est également important de noter que l'algorithme traite les instances de profondeur plus grande que 14 mais qu'à partir de cette profondeur, certaines instances pathologiques (i.e., avec un temps de résolution très long) apparaissent, ce qui rend difficile les mesures de performances.

Enfin, dans l'optique de mesurer la qualité des bornes inférieures et supérieures uti-

$\gamma = 0.2$					
Ratio \ Prof. (nœuds)	4 (31)	6 (127)	8 (511)	10 (2047)	12 (8191)
$RDU(L_{EU^*})/RDU^*$	95.1%	92.46%	90.0%	91.2%	90.9%
$RDU(L_{SD})/RDU^*$	104.4%	105.1%	106.9%	112.4%	117.5%
$\gamma = 0.5$					
Ratio \ Prof. (nœuds)	4 (31)	6 (127)	8 (511)	10 (2047)	12 (8191)
$RDU(L_{EU^*})/RDU^*$	99.6%	99.3%	90.0%	98.8%	98.2%
$RDU(L_{SD})/RDU^*$	105.5%	109.6%	109.4%	109.1%	110.2%
$\gamma = 0.8$					
Ratio \ Prof. (nœuds)	4 (31)	6 (127)	8 (511)	10 (2047)	12 (8191)
$RDU(L_{EU^*})/RDU^*$	99.8%	99.5%	99.4%	98.9%	98.6%
$RDU(L_{SD})/RDU^*$	106.4%	106.3%	108.8%	107.8%	107.2%

TABLE 3.3 – Qualités (en %) de la borne inférieure (i.e.  $RDU(L_{EU^*})$ ) et de la borne supérieure (i.e.  $RDU(L_{SD})$ ) en fonction de la profondeur de l'arbre et de la valeur de  $\gamma$ .

lisées dans l'approche par énumération implicite, nous avons observé, en fonction de la valeur du paramètre  $\gamma$  et de la profondeur de l'arbre de décision, la valeur des ratios  $RDU(L_{EU^*})/RDU^*$  et  $RDU(L_{SD})/RDU^*$ , où  $L_{EU^*}$  est la loterie correspondante à la stratégie qui optimise le critère EU,  $L_{SD}$  est la loterie dominante que l'on construit lors du calcul de la borne supérieure et  $RDU^*$  est la valeur RDU de la stratégie qui optimise le critère RDU. Le tableau 3.3 présente les observations faites sur ces ratios. Chaque valeur du tableau est une moyenne réalisée sur 50 instances. Les arbres de décision ont été générés de la même manière que dans les expériences précédentes (arbres binaires complets avec alternance des nœuds de décision et des nœuds de chance).

Ainsi, on constate que la valeur  $RDU(L_{EU^*})$  fournit une excellente borne inférieure, qui, sans surprise, tend à se détériorer lorsque la valeur de  $\gamma$  tend vers 0. Quant à la valeur  $RDU(L_{SD})$  de la borne supérieure, elle fournit une valeur à moins de 10% de la valeur optimale sur la plupart des instances.

### 3.5 Qui veut gagner des millions ?

Contrairement à ce que pourrait laisser suggérer le titre de cette partie, nous ne proposons pas au lecteur de gagner des millions. Dans cette partie, nous nous intéressons à expérimenter l'algorithme d'énumération implicite sur un problème concret. En effet, les expérimentations menées dans la partie précédente ont été réalisées sur des structures d'arbres très denses, ce qui, du fait de l'alternance des nœuds de décision et des nœuds de chance, augmente l'aspect combinatoire du problème. De plus, les valeurs des différentes stratégies, du fait de la façon de générer les probabilités et les utilités, sont dans l'ensemble

très proches, ce qui ne facilite pas les coupes lors de l'exécution de branch and bound. Les instances traitées précédemment peuvent donc être considérées, a priori, comme difficiles.

C'est pourquoi nous proposons ici de mesurer les performances de cet algorithme sur une structure d'arbre plus « naturelle » et sur des valeurs de probabilités et d'utilités reflétant une problématique réelle. Le problème que nous proposons de traiter ici est issu d'un célèbre jeu télévisé, bien connu en France sous le nom de « Qui veut gagner des millions ? ».

Le but de ce jeu est de répondre à une suite de 15 questions à choix multiples (quatre réponses possibles) de culture générale. Dès qu'une mauvaise réponse est donnée par le joueur, le jeu s'arrête et le joueur part avec un gain correspondant au dernier palier franchi (il y a un palier toutes les cinq questions). Le joueur peut également refuser de répondre à une question, dans ce cas le jeu s'arrête et le joueur part avec un gain dépendant du numéro de la question précédente.

A la suite de Perea et Puerto (2007), nous traitons ici le jeu dans sa version espagnole de 2003. Dans cette version, les gains correspondant aux numéros de questions sont respectivement (dans l'ordre des questions) : 150, 300, 450, 900, 1800, 2100, 2700, 3600, 4500, 9000, 18000, 36000, 72000, 144000 et 300000 Euros. Quant aux paliers, ils se situent à la question 5 et à la question 10. Ainsi, si le candidat donne la mauvaise réponse à la troisième question, il ne gagne rien, s'il décide de ne pas répondre à la troisième question, il gagne 300 Euros et s'il répond mal à la septième question, il gagne 1800 Euros (gain correspondant au dernier palier franchi). Enfin, le candidat dispose de trois jokers qu'il peut utiliser durant le jeu : *l'appel à un ami* (lui permettant de téléphoner à un ami pour lui demander conseil), *le 50 :50* (deux mauvaises réponses sont retirées aléatoirement des choix possibles) et *l'avis du public* (chaque membre du public vote pour la réponse qu'il pense juste et la répartition des votes est indiquée au candidat). Bien entendu, chaque joker n'est utilisable qu'une seule fois, par contre il est possible d'utiliser plusieurs jokers sur une même question.

Dans un premier temps, nous avons construit un arbre de décision à partir de la modélisation de ce jeu proposée par Perea et Puerto (2007). Dans leur modélisation, une stratégie est complètement définie par la donnée des numéros des questions où sont utilisés les jokers, ainsi que le numéro de la question où l'on décide de quitter le jeu. Les calculs sont basés sur la connaissance de la probabilité de bien répondre à la question  $k$ . L'utilisation d'un joker augmente cette probabilité. Les probabilités utilisées par Perea et Puerto sont issues de statistiques réalisées sur un large échantillon d'émissions télévisées.

Nous avons mené nos expériences avec trois fonctions  $\varphi$  de déformation des probabilités : la fonction identité (i.e.,  $\forall p \in [0, 1], \varphi(p) = p$ ), la fonction racine carré (i.e.,

$\varphi(p)$	50 : 50	<i>Appel</i>	<i>Public</i>	<i>Quitter</i>	$E(L)$	<i>Max</i>	$G_L(2.7K)$
$p$	9	10	12	13	2387	36K	0.10
$p^2$	4	5	5	8	1536	2.7K	0.35
$\sqrt{p}$	14	15	13	X	1987	300K	0.06

TABLE 3.4 – Stratégies optimales obtenues pour différentes fonctions  $\varphi$  de déformation des probabilités.

$\forall p \in [0; 1], \varphi(p) = \sqrt{p}$ ) et la fonction carré (i.e.,  $\forall p \in [0; 1], \varphi(p) = p^2$ ). La fonction d'utilité utilisée étant linéaire, ces fonctions de déformation des probabilités caractérisent respectivement une attitude neutre vis-à-vis du risque, un goût pour le risque et une aversion pour le risque. Nous avons reporté les résultats obtenus dans le tableau 3.4. Pour chaque fonction  $\varphi$  expérimentée nous avons indiqué la stratégie résultante (colonnes 50 : 50, *Appel*, *Public* et *Quitter*) ainsi que l'espérance de la loterie  $L$  correspondant à cette stratégie (colonne  $E(L)$ ), la valeur maximale d'utilité figurant dans la loterie  $L$  (colonne *Max*) ainsi que la probabilité de gagner au moins 2,7K avec la loterie  $L$  (colonne  $G_L(2.7K)$ ). Notons que, dans tous les cas, le calcul de la stratégie optimale à été instantané bien que l'arbre de décision contienne plus de 14400 nœuds de décision et que la profondeur de cet arbre soit de 30.

La principale limite de la modélisation introduite par Perea et Puerto (2007) vient du fait que l'utilisation d'un joker ne dépend pas de la connaissance du candidat, mais uniquement du numéro de la question. Autrement dit, dans leur modélisation, une stratégie impose d'utiliser un joker à la question  $k$ , et cela même si l'on connaît déjà la réponse à la question. Pour cette raison, nous avons raffiné leur modélisation de la manière suivante : si le candidat connaît la réponse à la question  $k$ , alors il répond directement, sinon il doit prendre la décision d'utiliser un ou plusieurs jokers, de quitter le jeu sans répondre ou de répondre sans être certain.

Dans la figure 3.7 page 104, nous avons représenté un « morceau » de l'arbre de décision que nous avons construit. Ce « morceau » correspond à une partie de la situation qui se présente au décideur à la première question. Si le candidat connaît la réponse à la première question (événement « Sait » au nœud de chance  $Q_1$ ), alors il donne sa réponse et on passe à la question 2. Si le candidat ne connaît pas la réponse à la première question (événement « Ne sait pas » au nœud de chance  $Q_1$ ), alors le décideur est soumis au choix suivant (nœud de décision  $D_1^1$ ) :

- quitter le jeu,
- utiliser un des trois jokers,
- répondre directement.

S'il décide de quitter le jeu, alors il ne gagne rien puisque le gain lié à la question 0 est

de 0 Euro. S'il décide de répondre directement, il y a une probabilité qu'il réponde juste ou qu'il réponde faux (nœud de chance  $A_1^1$ ). S'il répond juste, on passe à la question suivante, sinon le jeu s'arrête et le candidat part avec 0 Euro (gain correspondant au premier palier). S'il décide de prendre un joker, alors le candidat est soumis à un nouveau choix (nœud de décision  $D_1^2$ ,  $D_1^3$  et  $D_1^4$ ) :

- quitter le jeu,
- utiliser un second joker
- répondre à la question (la probabilité de répondre juste est plus élevée que précédemment du fait de l'utilisation du premier joker).

Le morceau d'arbre se poursuit de la même manière jusqu'à que chacune des issues de l'arbre soit une feuille ou un passage à la question suivante. Ce motif est répété pour chaque question, en prenant bien sûr le soin de retirer les décisions liées à l'utilisation des jokers déjà utilisés auparavant. Concernant les probabilités de connaître la réponse à la question  $k$  ou de répondre correctement à la question  $k$  selon les jokers utilisés, nous nous sommes basés sur les données fournies par Perea et Puerto (2007).

L'arbre de décision construit contient plus de 75 millions de nœuds et le nombre de stratégies est exponentiel en la taille de l'arbre. Contrairement aux expériences précédentes, pour exécuter l'algorithme, nous avons utilisé un ordinateur avec 64Go de RAM afin de pouvoir stocker l'arbre de décision et effectuer nos calculs sans soucis de mémoire. Nous avons utilisé une fonction d'utilité linéaire pour encoder les conséquences. Malgré la taille très importante de l'arbre de décision, l'algorithme d'énumération implicite a retourné en 2992 secondes une stratégie qui optimise le critère RDU lorsque  $\forall p \in [0; 1], \varphi(p) = p^2$  (ce qui correspond à un comportement d'aversion pour le risque), et en 4026 secondes lorsque  $\varphi(p) = p^{2/3}$  (ce qui correspond à un comportement de goût pour le risque). Notons que plus la concavité de la fonction  $\varphi$  est forte, plus les temps de résolution sont longs.

### 3.6 Stratégies mixtes

Dans cette partie, nous nous intéressons à la détermination d'une *stratégie mixte* optimisant le critère RDU. Dans une stratégie mixte, le décideur choisit aléatoirement (en suivant une distribution de probabilité) la décision à prendre en chaque nœud de décision. Ainsi, la détermination d'une stratégie mixte consiste à déterminer une distribution de probabilité en chaque nœud de décision. L'ensemble des stratégies mixtes inclut donc l'ensemble des *stratégies pures* (dans cette partie, nous donnons le nom de stratégies pures aux stratégies que nous manipulons dans les parties précédentes afin de pouvoir les différencier des stratégies mixtes). Par exemple, une stratégie mixte possible pour l'arbre de décision représenté sur la figure 3.2 page 88 est l'ensemble des distribution de

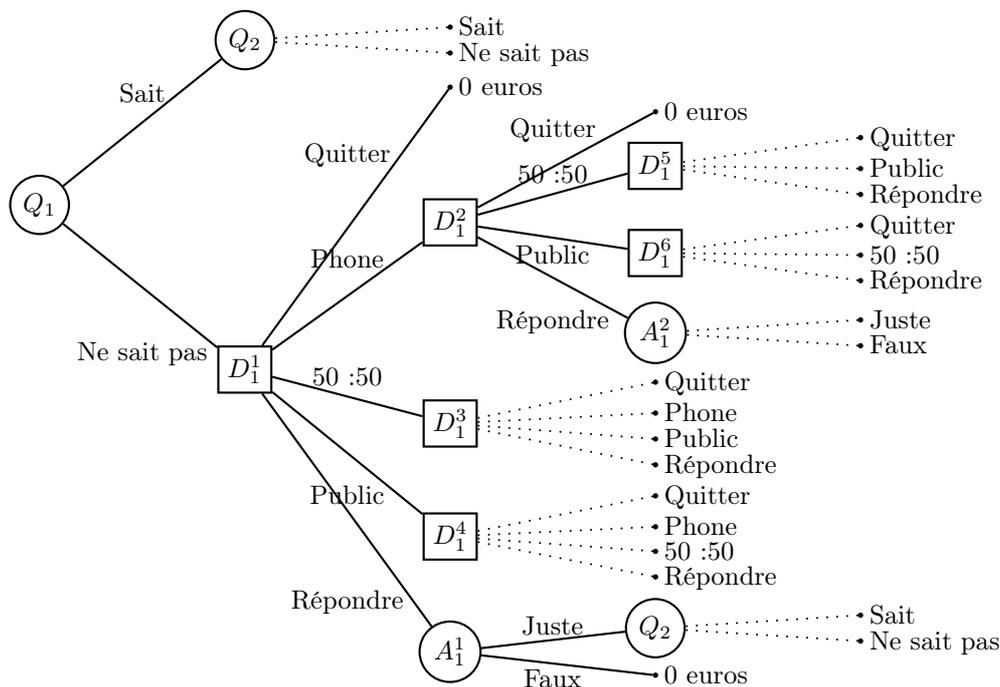


FIGURE 3.7 – Représentation du « morceau » de l'arbre de décision raffinant la modélisation de Perea et Puerto (2007).

probabilité, associées aux nœuds de décision, suivantes :

- $P(D_1 = (D_1, L_1)) = 0.6$  et  $P(D_1 = (D_1, L_2)) = 0.4$
- $P(D_2 = (D_2, L_3)) = 0.3$  et  $P(D_2 = (D_2, L_4)) = 0.7$
- $P(D_3 = (D_3, L_5)) = 0.2$  et  $P(D_3 = (D_3, L_6)) = 0.8$
- $P(D_4 = (D_4, L_7)) = 0.5$  et  $P(D_4 = (D_4, L_8)) = 0.5$
- $P(D_5 = (D_5, L_9)) = 0.9$  et  $P(D_5 = (D_5, L_{10})) = 0.1$

L'application de cette stratégie consiste alors à tirer au sort la décision à prendre en suivant ces distributions de probabilité.

Lorsque le critère de décision utilisé est le critère EU, il existe toujours une stratégie pure qui soit optimale. De ce fait, la question de la détermination d'une stratégie mixte ne se pose pas dans le cadre du modèle EU. Cependant, dans le cadre du modèle RDU, selon la forme de la fonction  $\varphi$  de déformation des probabilités, il peut arriver qu'une stratégie mixte possède une meilleure évaluation qu'une stratégie pure. En guise d'exemple, considérons un arbre de décision avec un unique nœud de décision pouvant conduire à deux options différentes : la loterie certaine  $(5, 1)$ , et la loterie  $(1, 0.5; 10, 0.5)$ . Faisons l'hypothèse que la fonction  $\varphi$  de déformation des probabilités soit la suivante :  $\varphi(0) = 0$ ,  $\varphi(p) = 0.45$  si  $p \in ]0, 0.7]$  et  $\varphi(p) = 1$  si  $p > 0.7$ . Les valeurs RDU des deux stratégies pures sont respectivement 5 et  $1 + 9\varphi(0.5) = 5.05$ . Considérons maintenant la stratégie mixte qui consiste à choisir, avec une probabilité de 0.6, la décision menant à la loterie certaine,

et de choisir, avec une probabilité de 0.4, la seconde loterie. Cette stratégie mixte conduit alors à la loterie suivante :  $(1, 0.5 \times 0.4; 5, 1 \times 0.6; 10, 0.5 \times 0.4) = (1, 0.2; 5, 0.6; 10, 0.2)$ . L'évaluation de cette loterie, avec le critère RDU, donne alors  $1 + 5\varphi(0.8) + 5\varphi(0.2) = 7.25$ . Ainsi, une stratégie mixte peut effectivement posséder une valeur RDU strictement supérieure à toutes les valeurs des stratégies pures.

Néanmoins, lorsque la fonction  $\varphi$  de déformation des probabilités est convexe, alors il existe toujours une stratégie pure qui soit optimale au sens du critère RDU. Pour s'en convaincre, il suffit de remarquer que : soit  $X = \{u_1, \dots, u_n\}$  avec  $u_1 \leq u_2 \leq \dots \leq u_n$  et  $\lambda \in [0; 1]$ , alors  $\forall (L_1, L_2) \in \mathbb{L}_X^2$ , on a :

$$RDU(\lambda L_1 + (1 - \lambda)L_2) = u_1 + \sum_{i=2}^n (u_i - u_{i-1})\varphi(\lambda G_{L_1}(u_i) + (1 - \lambda)G_{L_2}(u_i))$$

car  $\forall u \in X, G_{\lambda L_1 + (1-\lambda)L_2}(u) = \lambda G_{L_1}(u) + (1 - \lambda)G_{L_2}(u)$ . On a de plus :

$$\begin{aligned} & u_1 + \sum_{i=2}^n (u_i - u_{i-1})\varphi(\lambda G_{L_1}(u_i) + (1 - \lambda)G_{L_2}(u_i)) \\ & \leq u_1 + \sum_{i=2}^n (u_i - u_{i-1})(\lambda\varphi(G_{L_1}(u_i)) + (1 - \lambda)\varphi(G_{L_2}(u_i))) \end{aligned}$$

par convexité de la fonction  $\varphi$ . Or on a :

$$u_1 + \sum_{i=2}^n (u_i - u_{i-1})(\lambda\varphi(G_{L_1}(u_i)) + (1 - \lambda)\varphi(G_{L_2}(u_i))) = \lambda RDU(L_1) + (1 - \lambda)RDU(L_2)$$

et on a évidemment :

$$\lambda RDU(L_1) + (1 - \lambda)RDU(L_2) \leq \max\{RDU(L_1), RDU(L_2)\}$$

Lorsque  $\varphi$  est une fonction convexe, la recherche d'une stratégie mixte optimisant le critère RDU revient donc à déterminer une stratégie pure optimisant le critère RDU. Pour ce faire, on peut toujours utiliser l'algorithme d'énumération implicite introduit précédemment.

Lorsque la fonction  $\varphi$  est concave et linéaire par morceaux, on peut montrer que la recherche d'une stratégie mixte optimisant le critère RDU est un problème polynomial. Pour cela, nous allons adapter le programme linéaire en variables mixtes introduit dans la partie 3.2 page 86. Rappelons que ce programme permet de déterminer une stratégie pure optimale au sens de RDU lorsque la fonction  $\varphi$  est concave et linéaire par morceaux. Cette adaptation consiste à transformer les variables booléennes en variables réelles, ainsi qu'à modifier les contraintes qui sont liées à ces variables.

L'ensemble des utilités de l'arbre de décision  $T$  est noté  $U = \{u_1, u_2, \dots, u_n\}$ , avec  $u_1 \leq u_2 \leq \dots \leq u_n$ . Dans une stratégie mixte, la probabilité d'obtenir une conséquence

$u_h$  est égale au produit  $p_h$  des probabilités associées aux événements rencontrés sur le chemin allant de la racine au nœud offrant la conséquence  $u_h$ , multiplié par le produit des probabilités assignées aux décisions rencontrées sur le même chemin. Dans cette optique, pour chaque arc décisionnel  $(D_i, L_j)$ , nous substituons une variable réelle  $p_{(i,j)}$  à la variable booléenne  $y_{(i,j)}$ . Cependant, afin d'obtenir des contraintes linéaires, cette variable ne doit pas représenter la probabilité  $P((D_i, L_j)|D_i)$  (probabilité de prendre la décision  $(D_i, L_j)$  conditionnellement au fait d'atteindre le nœud  $D_i$ ), mais le produit des probabilités assignées aux décisions rencontrées sur le chemin allant de la racine au nœud de chance  $L_j$ . Il y a alors une bijection entre les valeurs des probabilités  $P((D_i, A_j)|D_i)$  et les valeurs des probabilités  $p_{(i,j)}$ , puisque  $P((D_i, A_j)|D_i) = p_{(i,j)}/p_{prev_D(i)}$ . Cette mise en correspondance est illustrée sur la figure 3.8 page 107.

Dans cette formulation, la probabilité d'obtenir l'utilité  $u_h$  vaut  $p_h p_{prev_u(h)}$  car  $p_{prev_u(h)}$  représente le produit des probabilités assignées aux décisions rencontrées sur le chemin allant de la racine au nœud offrant la conséquence  $u_h$ . La fonction objectif s'écrit donc de la façon suivante :

$$u_1 + \sum_{h=2}^n (u_h - u_{h-1}) \varphi \left( \sum_{j \geq h} p_j p_{prev_u(j)} \right)$$

Intéressons-nous maintenant aux contraintes liées aux variables  $p_{(i,j)}$ . D'après la formule de Bayes, nous avons :

$$\sum_j p_{(i,j)} = \sum_j p_{prev_D(i)} P((D_i, A_j)|D_i)$$

D'autre part, on a :

$$\sum_j p_{prev_D(i)} P((D_i, A_j)|D_i) = p_{prev_D(i)} \sum_j P((D_i, A_j)|D_i) = p_{prev_D(i)}$$

car les probabilités conditionnelles somment à 1. On en déduit que :

$$\sum_j p_{(i,j)} = p_{prev_D(i)}$$

Les contraintes sur les variables  $p_{(i,j)}$  sont donc très similaires aux contraintes que nous avons sur les variables  $y_{(i,j)}$  :

$$\begin{aligned} \sum_j p_{(1,j)} &= 1 \\ \sum_j p_{(i,j)} &= p_{prev_D(i)} \quad \forall i \in \{2, \dots, |\mathcal{N}_D|\} \end{aligned}$$

où  $p_{(i,j)} \in [0; 1]$ . Toutes les autres contraintes sont identiques à celles de la formulation pour les stratégies pures (avec  $y_{(i,j)}$  remplacé par  $p_{(i,j)}$ ).

Le nombre de contraintes et de variables étant polynomial en la taille de l'arbre de décision, et étant donné que toutes les variables sont réelles, on en déduit que la recherche d'une stratégie mixte optimisant le critère RDU peut se faire en un temps polynomial en la taille de l'arbre de décision lorsque la fonction  $\varphi$  est concave et linéaire par morceaux.

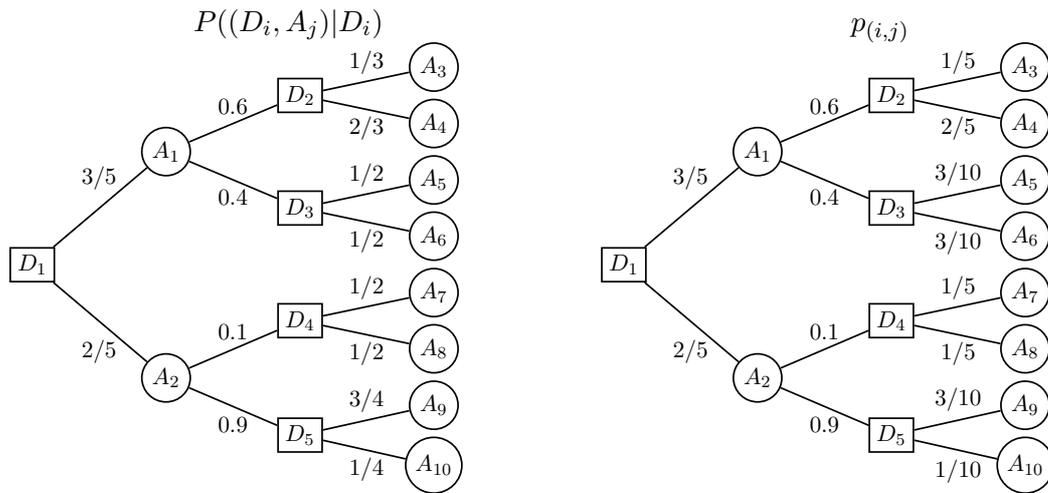


FIGURE 3.8 – Correspondance entre les valeurs des probabilités  $P((D_i, A_j)|D_i)$  et les valeurs des probabilités  $p(i,j)$ .

## Conclusion

Dans ce chapitre, nous nous sommes consacrés à l'optimisation du critère RDU dans les arbres de décision.

Dans un premier temps, nous avons démontré que le problème de la recherche d'une stratégie pure optimisant le critère RDU dans un arbre de décision est un problème NP-difficile. Face à ce constat, il est alors nécessaire de développer des méthodes d'optimisation combinatoire si l'on désire faire usage du critère RDU pour modéliser les préférences du décideur.

Pour ce faire, nous avons présenté une première méthode consistant à modéliser le problème sous la forme d'un programme linéaire en variables mixtes. Le nombre de variables et de contraintes de ce programme linéaire est polynomial en la taille de l'arbre. Cependant, cette formulation n'est exploitable que lorsque la fonction  $\varphi$  de transformation des probabilités est concave et linéaire par morceaux. Pour pallier cet inconvénient, nous avons développé une méthode de branch and bound, restant valide quelle que soit la fonction  $\varphi$  de transformation des probabilités utilisée.

Suite à cela, nous avons comparé les deux méthodes sur des instances aléatoires. L'approche par branch and bound s'est révélée être beaucoup plus efficace. La formulation par un programme linéaire en variables mixtes ne perd cependant pas tout son intérêt. En effet, nous avons pu constater qu'une modification mineure de cette formulation permettrait de traiter en temps polynomial le problème de la recherche d'une stratégie mixte qui optimise le critère RDU.

Nous avons ensuite présenté des expérimentations numériques pour l'approche basée sur le branch and bound. Comme on pouvait s'y attendre, plus la fonction  $\varphi$  s'éloigne

de la fonction identité (i.e., plus le critère RDU s'éloigne du critère EU), plus le temps de résolution devient important. Nous avons également constaté la qualité des bornes utilisées dans cet algorithme au travers d'une autre série d'expérimentations numériques.

La taille des arbres de décision que l'algorithme peut traiter en un temps raisonnable, bien que déjà conséquente dans les expérimentations numériques menées, est limitée par la « difficulté » des instances générées aléatoirement. En effet, l'alternance systématique des nœuds de chance et des nœuds de décision est en faveur d'une combinatoire très élevée. De plus, les arbres générés étant équilibrés, les stratégies réalisables conduisent à des loteries pouvant être assez similaires, ce qui n'est pas un cadre favorisant les coupes lors de l'exécution du branch and bound. Afin de se convaincre qu'en situation réelle cet algorithme peut traiter de plus grandes instances, nous avons appliqué l'algorithme à un problème concret : celui issu du jeu télévisé « Qui veut gagner des millions ? ». L'arbre obtenu est d'une taille très importante. En effet, l'arbre de décision qui en résulte comporte pas moins de 75 millions de nœuds. Malgré cela, l'algorithme fournit la stratégie optimisant le critère RDU en des temps raisonnables, et ce pour diverses formes de la fonction  $\varphi$ .

Les résultats des expérimentations numériques menées sont assez encourageants. De plus, notons que lorsqu'une stratégie est calculée, il n'est plus nécessaire de faire aucun calcul pour sa mise en application. Il suffit juste d'appliquer la stratégie retournée par l'algorithme. Ainsi, même si dans certaines situations, le temps de résolution peut prendre un temps important, ce calcul ne se fait qu'une seule fois et l'application de la stratégie est ensuite instantané au moment de prendre les décisions puisqu'il suffit alors de suivre la stratégie calculée précédemment qui anticipe déjà toutes les situations possibles.

Ainsi le verrou algorithmique qui empêche l'application du critère RDU dans un cadre séquentiel a été repoussé et de nombreux problèmes sont maintenant traitables avec le critère RDU. Notons que la méthode par énumération implicite proposée fonctionne avec n'importe quel critère compatible avec la dominance stochastique du premier ordre.

Le travail sur les stratégies mixtes reste cependant encore incomplet. En effet, nous n'avons pas proposé de méthode pour les cas où la fonction  $\varphi$  n'était ni convexe ni concave. Il est d'autant plus intéressant de se pencher sur ces différents cas puisqu'ils concernent les formes remarquables de la fonction  $\varphi$  permettant de décrire, de manière fidèle, les préférences de la plupart des décideurs.

Dans le prochain chapitre nous allons nous intéresser à l'application du critère RDU dans un autre modèle graphique : les diagrammes d'influence. Ces derniers présentent des avantages indéniables de compacité, mais induisent de nouveaux problèmes algorithmiques qui ne se posaient pas dans le formalisme des arbres de décision.





## Chapitre 4

# Critère RDU dans les diagrammes d'influence<sup>1</sup>

**Résumé.** Dans ce chapitre, nous nous intéressons à l'utilisation du critère RDU dans les diagrammes d'influence. Dans un premier temps, nous présentons le formalisme des diagrammes d'influence. Ce formalisme permet de représenter de manière compacte un problème de décision séquentielle dans l'incertain en tirant profit des indépendances entre les différentes variables du problème. Cependant, comme nous le faisons remarquer dans la suite, le stockage en mémoire d'une stratégie peut devenir de taille exponentielle en la taille du diagramme d'influence. Néanmoins, lorsque l'on utilise le critère EU il est possible de tirer profit de l'axiome conséquentialiste dans le but de factoriser la représentation en mémoire d'une stratégie. Malheureusement, comme nous l'avons vu dans le chapitre 2, le critère RDU ne respecte pas cet axiome. Nous proposons donc dans ce chapitre un algorithme approché visant à déterminer une stratégie à la fois compacte et proche de l'optimum au sens de RDU. Pour cela, nous proposons de réaliser une  $\Lambda$ -relaxation de l'ensemble des stratégies factorisées (i.e., représentées de manière plus compacte en mémoire). Cette relaxation consiste à ajouter  $\Lambda$  dépendances sur chaque variable de décision. Enfin, nous déterminons la stratégie qui optimise le critère RDU dans cet ensemble de stratégies moins gourmandes en mémoire. Pour ce faire, nous adaptons l'algorithme d'énumération implicite proposé dans le cadre des arbres de décision (voir chapitre 3). Pour finir, nous présentons les performances de cette approche au travers d'expérimentations numériques et mettons en avant la qualité de la stratégie obtenue au sens de RDU.

---

1. Ce chapitre s'appuie en partie sur des résultats présentés dans (Jeantet et Spanjaard, 2009a).

## Introduction

Dans le chapitre précédent, nous nous sommes intéressés à déterminer une stratégie optimisant le critère RDU dans un arbre de décision. Les arbres de décision permettent de représenter graphiquement un problème de décision séquentielle de manière simple et explicite. Cependant, l'explicitation de tous les scénarios envisageables du problème, qui simplifie l'interprétation du modèle d'une part, est également ce qui rend les arbres de décision gourmands en place mémoire d'autre part. En effet, comme nous l'avons vu dans le chapitre précédent, l'arbre obtenu pour le problème de « Qui veut gagner des millions ? » comporte un nombre de nœuds très important, et nous avons dû utiliser un ordinateur doté de 64Go de RAM afin de pouvoir stocker l'arbre de décision et effectuer nos calculs.

Certains problèmes de décision séquentielle sous incertitude étant, de par leur nature, de tailles encore plus conséquentes, il devient alors inenvisageable d'utiliser le formalisme des arbres de décision pour les représenter. Face à ce constat, d'autres représentations graphiques ont vu le jour dans la littérature. Ces dernières permettent de représenter des problèmes de décision séquentielle de manière plus compacte. Dans ce chapitre, nous nous intéressons à l'une d'entre elles plus spécifiquement, connue sous le nom de *diagramme d'influence*, introduite initialement par Howard et Matheson (1984). Cette représentation tire profit des indépendances entre les différentes variables du problème pour factoriser sa représentation graphique, et permet donc de réduire la place occupée en mémoire.

L'usage d'un diagramme d'influence se heurte néanmoins au problème de la représentation d'une stratégie en mémoire. En effet, la taille d'une stratégie est généralement exponentielle en la taille du diagramme d'influence. Ainsi, lorsque l'on explicite une stratégie (i.e., on énumère toutes les décisions qui la composent conditionnellement aux événements passés), on perd les avantages de compacité offerts par les diagrammes d'influence. Néanmoins, avec le critère EU, il est possible de tirer profit des propriétés induites par l'axiome conséquentialiste pour factoriser la représentation d'une stratégie en mémoire. En revanche, aucune factorisation n'est a priori envisageable lorsque l'on utilise un critère ne respectant pas l'axiome conséquentialiste et l'intérêt d'utiliser un diagramme d'influence plutôt qu'un arbre de décision devient alors quasiment nul. Cependant, comme dit précédemment, il est des cas où la taille du problème de décision séquentielle est telle qu'elle ne permet pas de se passer du formalisme des diagrammes d'influence. Une façon de sortir de cette impasse est alors de réaliser un compromis entre la taille des stratégies envisagées et leur pouvoir descriptif. C'est justement ce que l'on s'efforce de faire ici en réalisant une  $\Lambda$ -relaxation de l'ensemble des stratégies dont la représentation en mémoire est factorisable.

Dans un premier temps, nous présentons le formalisme des diagrammes d'influence. Nous détaillons ensuite l'idée de la  $\Lambda$ -relaxation de l'espace des stratégies factorisées. Enfin, nous présentons une mise en œuvre algorithmique, dans un diagramme d'influence, de cette relaxation. Ceci nous permet de proposer un algorithme d'énumération implicite permettant de déterminer une stratégie approchée au sens de RDU. Les performances de ce dernier algorithme sont évaluées via des expérimentations numériques.

## 4.1 Diagramme d'influence

Un diagramme d'influence (Howard et Matheson, 1984) est un modèle graphique permettant de représenter, de manière compacte, un problème de décision séquentielle. Contrairement à la représentation fournie par un arbre de décision, dans un diagramme d'influence, l'accent est mis sur les dépendances entre les différentes variables. Tirant profit des indépendances entre les variables, le diagramme d'influence permet alors de factoriser l'information redondante.

### 4.1.1 Formalisme

Un diagramme d'influence est un graphe orienté  $\mathcal{G} = (\mathcal{N}, \mathcal{E})$  sans circuit. L'ensemble des nœuds  $\mathcal{N}$  est partitionné en trois sous-ensembles disjoints :

- $\mathcal{N}_D = \{D_1, \dots, D_n\}$  : l'ensemble des nœuds de décision, que l'on représente graphiquement par des rectangles. Les nœuds de décision représentent les variables de décision du problème.
- $\mathcal{N}_C = \{L_1, \dots, L_p\}$  : l'ensemble des nœuds de chance, que l'on représente graphiquement par des cercles. Les nœuds de chance représentent les variables aléatoires du problème.
- $\mathcal{N}_U = \{U_1, \dots, U_m\}$  : l'ensemble des nœuds d'utilité, que l'on représente graphiquement par des losanges. Les nœuds d'utilité représentent les variables d'utilité du problème, et sont donc les puits du graphe (i.e., aucun arc n'est issu de ces nœuds).

Par abus de langage, on notera de la même manière un nœud et la variable représentée par ce nœud. On notera  $D(X)$  le domaine de la variable  $X$ . On notera  $pere(N)$  l'ensemble des nœuds qui sont les pères du nœud  $N \in \mathcal{N}$ . Par ailleurs, l'ensemble des arcs  $\mathcal{E}$  est partitionné en deux sous-ensembles disjoints :

- $\mathcal{E}_D = \{(N, N') : N' \in \mathcal{N}_C \cup \mathcal{N}_U\}$  : l'ensemble des arcs de dépendance. Un arc de dépendance  $(N, N')$  représente une dépendance entre la variable  $N$  et la variable  $N'$ .
- $\mathcal{E}_T = \{(N, N') : N' \in \mathcal{N}_D\}$  : l'ensemble des arcs temporels. Un arc temporel  $(N, N')$  représente le fait que la variable  $N$  est observée avant de prendre la décision  $N'$ .

Ensuite, à chaque nœud de chance, on associe une table de probabilités conditionnelles. Cette table indique la probabilité qu'un événement se réalise, conditionnellement aux valeurs des variables représentées par chacun de ses pères. De même, à chaque nœud d'utilité est associée une table d'utilités retournant la valeur d'utilité offerte en fonction des valeurs des variables représentées par chacun de ses pères. Enfin, on impose une condition structurelle sur ce graphe : il existe au moins un chemin orienté qui passe par tous les nœuds de décision. Cette condition permet de s'assurer qu'il existe un ordre stricte entre les différents nœuds de décision et donc de déterminer dans quel ordre chaque décision doit être prise.

**Exemple 15** *La figure 4.1 page 115 représente un diagramme d'influence. Ce diagramme comporte deux nœuds de décision :  $D_1$  et  $D_2$ . Les décisions possibles en  $D_1$  sont  $\alpha$  et  $\beta$ , et les décisions possibles en  $D_2$  sont  $\gamma$  et  $\delta$ . Il comporte également trois nœuds de chance :  $A_1$  (d'événements  $\theta_1$  et  $\theta_2$ ),  $A_2$  (d'événements  $\theta'_1$  et  $\theta'_2$ ) et  $A_3$  (d'événements  $\theta''_1$  et  $\theta''_2$ ). Enfin il comporte un unique nœud d'utilité :  $U$ . Les arcs  $(D_1, D_2)$ ,  $(A_1, D_2)$  et  $(A_2, D_2)$  sont des arcs temporels puisqu'ils pointent vers un nœud de décision. Les arcs  $(D_1, U)$ ,  $(A_1, A_2)$ ,  $(A_2, A_3)$ ,  $(D_2, A_3)$ ,  $(D_2, U)$  et  $(A_3, U)$  sont des arcs de dépendances puisqu'ils pointent vers un nœud de chance ou vers un nœud d'utilité. De ce fait, on associe, au nœud  $A_1$  la table de probabilités représentant  $P(A_1)$ , au nœud  $A_2$  la table de probabilités  $P(A_2|A_1)$ , au nœud  $A_3$  la table de probabilités  $P(A_3|D_2, A_2)$  et au nœud  $U$  la table d'utilités  $u(D_1, D_2, A_3)$ .*

Afin d'illustrer la compacité de la représentation d'un diagramme d'influence, nous mettons en parallèle deux représentations d'un même problème dans la figure 4.2 page 117. A gauche est donnée la représentation du problème sous la forme d'un arbre de décision et à droite sa représentation sous la forme d'un diagramme d'influence.

Dans le premier cas, selon la décision prise au nœud  $D$ , les probabilités aux deux nœuds  $L$  sont différentes. On peut donc déduire une dépendance entre la variable  $D$  et la variable  $L$  dans le diagramme d'influence. De même, les conséquences étant toutes différentes, on en déduit que la décision prise au nœud  $D$ , ainsi que l'événement de  $L$  qui se réalise, conditionnent la valeur de l'utilité. Cela se traduit par deux arcs de dépendance vers le nœud  $U$  dans le diagramme d'influence. Dans ce cas de figure, on ne

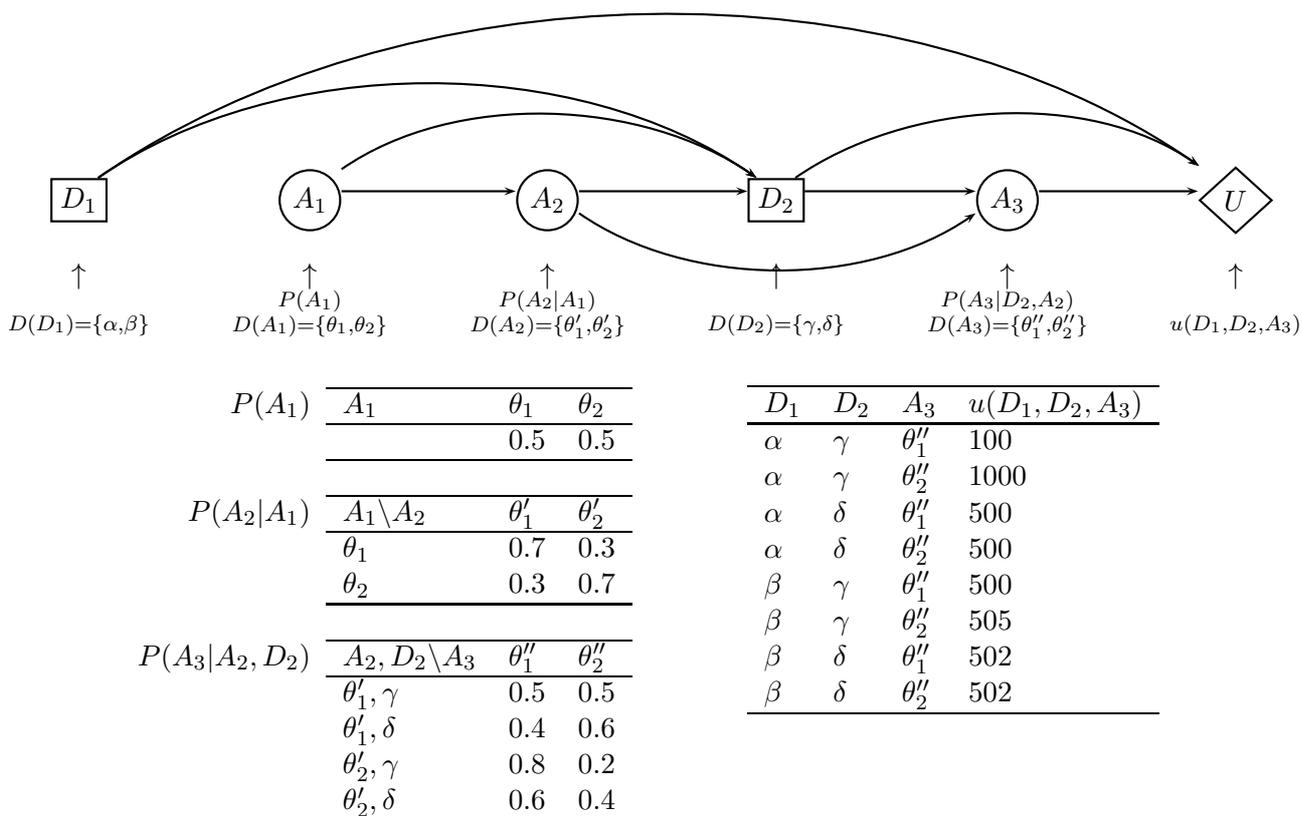


FIGURE 4.1 – Exemple de diagramme d'influence.

gagne aucune place mémoire car tous les conditionnements possibles sont présents. En effet, les tables stockées au nœud  $L$  et au nœud  $U$  dépendent des valeurs des variables  $L$  et  $D$  et comportent donc chacune  $|D(D)| \times |D(L)| = 2 \times 3 = 6$  entrées. Nous retenons donc 12 valeurs tout comme dans l'arbre de décision où l'on explicite également 12 valeurs (6 valeurs de probabilités et 6 valeurs d'utilités).

Dans le second cas, on déduit, comme précédemment, une dépendance entre la variable  $D$  et la variable  $L$ . Par contre, les utilités ne dépendent que de l'événement qui se réalise au nœud  $L$  : la décision prise au nœud  $D$  ne conditionne pas la valeur d'utilité obtenue. Cela se traduit par l'absence d'arc entre les nœuds  $D$  et  $U$  dans le diagramme d'influence. Cette fois la table stockée au nœud  $L$  comporte  $|D(D)| \times |D(L)| = 2 \times 3 = 6$  entrées et la table stockée au nœud  $U$  comporte seulement  $|D(L)| = 3$  entrées. On ne conserve donc que 9 valeurs au lieu des 12 explicitées dans l'arbre de décision.

Dans le troisième cas, on déduit, une fois encore, une dépendance entre la variable  $D$  et la variable  $L$ . Par contre, la valeur d'utilité ne dépend pas de l'événement qui se réalise au nœud  $L$  (ce qui se traduit par l'absence d'arc entre  $L$  et  $U$  dans le diagramme d'influence), mais cette valeur est différente selon la décision prise au nœud  $D$  (ce qui se traduit par la présence d'un arc entre le nœud  $D$  et le nœud  $U$ ). Cette fois la table

stockée au nœud  $L$  comporte  $|D(D)| \times |D(L)| = 2 \times 3 = 6$  entrées et la table stockée au nœud  $U$  comporte seulement  $|D(D)| = 2$  entrées. On ne conserve donc que 8 valeurs au lieu des 12 explicitées dans l'arbre de décision.

Dans le dernier cas, les utilités étant toutes différentes, on en déduit qu'elles dépendent conjointement de l'événement qui se réalise au nœud  $L$  et de la décision prise au nœud  $D$ . En revanche, les probabilités des événements au nœud  $L$  restent identiques quelle que soit la décision prise au nœud  $D$ , d'où l'absence d'arc entre la variable  $D$  et la variable  $L$  dans le diagramme d'influence. Cette fois la table stockée au nœud  $L$  comporte seulement  $|D(L)| = 3$  entrées et la table stockée au nœud  $U$  comporte  $|D(D)| \times |D(L)| = 2 \times 3 = 6$  entrées. On ne conserve donc que 9 valeurs au lieu des 12 explicitées dans l'arbre de décision.

Ainsi, le formalisme des diagrammes d'influence, à la manière des *réseaux bayésiens*, exploite les indépendances conditionnelles entre les différentes variables pour représenter le problème de décision séquentielle de manière plus compacte. Evidemment, comme nous l'avons vu dans le premier cas, si tous les conditionnements possibles sont présents, le diagramme d'influence ne permettra aucun gain sur la taille de la représentation.

Dans l'optique de définir une stratégie, il est nécessaire de connaître l'ensemble des variables déjà observées lorsque l'on prend une décision. Une stratégie consiste alors à fixer une valeur à chaque variable de décision conditionnellement aux événements passés et aux décisions déjà prises. Dans un arbre de décision, les événements passés et les décisions déjà prises se déduisent naturellement de la sémantique du modèle : ce sont les arcs sur le chemin allant de la racine au nœud actuel. Dans un diagramme d'influence, l'ordre d'observation des variables est moins naturel.

Pour déterminer l'ordre d'observation des variables, on partitionne l'ensemble  $\mathcal{N}_C$  en  $|\mathcal{N}_D| + 1$  sous-ensembles disjoints :  $I_0, I_1, \dots, I_n$ . L'ensemble  $I_0$  contient les nœuds de chance observés avant le nœud de décision  $D_1$  (i.e., l'ensemble des nœuds de chance qui sont pères du nœud  $D_1$ ),  $I_k$  contient les nœuds de chance observés entre les nœuds de décision  $D_k$  et  $D_{k+1}$  (i.e., l'ensemble des nœuds de chance qui sont pères du nœud  $D_{k+1}$ ), et  $I_n$  contient les nœuds de chance jamais observés ou observés après la décision prise en  $D_n$  (i.e., l'ensemble des nœuds de chance qui ne sont pas pères d'un nœud de décision). Ce partitionnement induit alors un ordre partiel  $\ll$  sur  $\mathcal{N}_D \cup \mathcal{N}_C$  :

$$I_0 \ll D_1 \ll I_1 \ll \dots \ll I_{n-1} \ll D_n \ll I_n$$

Tout ordre sur les variables compatible avec cet ordre partiel est un ordre valide d'observation des variables.

**Exemple 16** *Considérons le diagramme d'influence représenté sur la figure 4.1 page 115.*

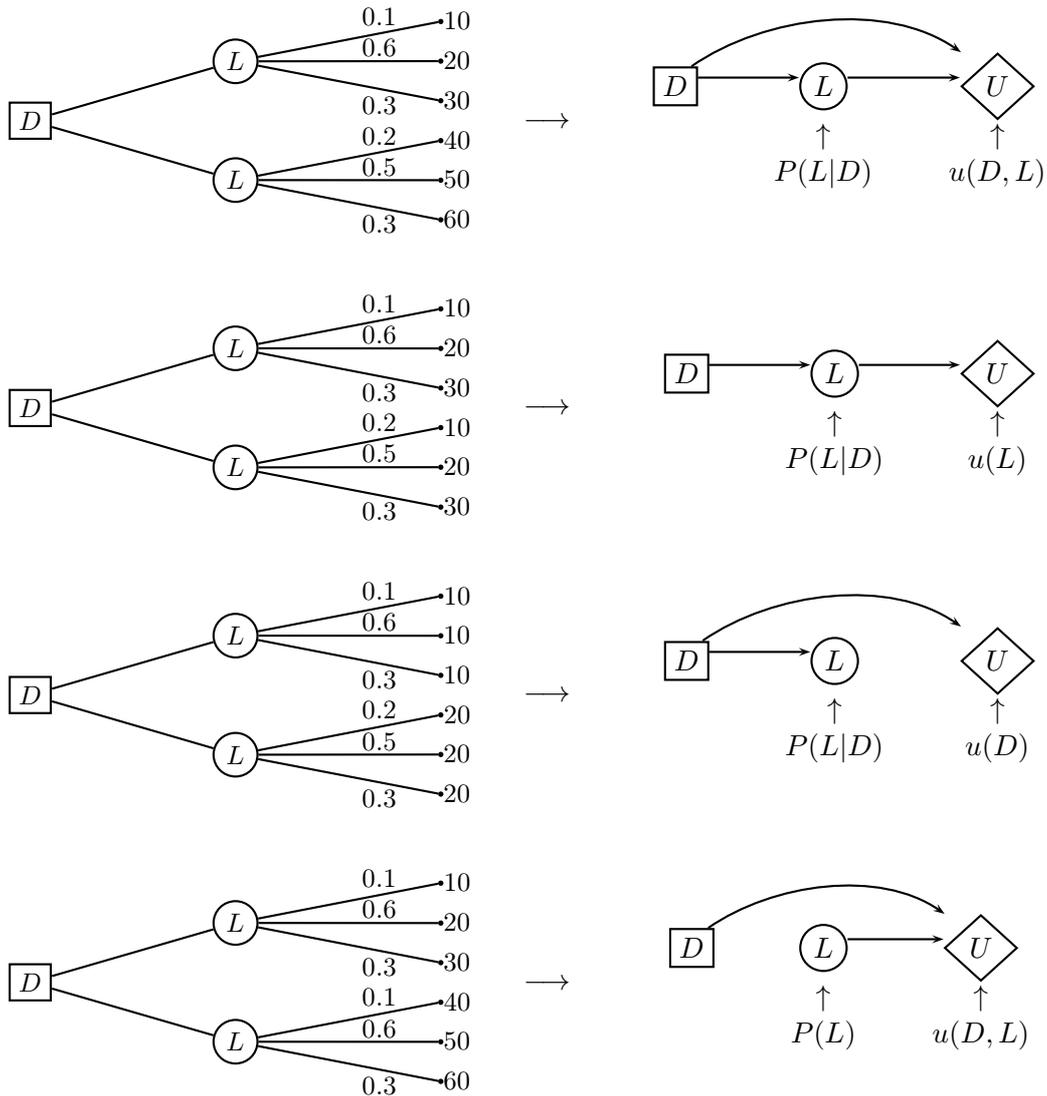


FIGURE 4.2 – Passage d’une structure d’arbre de décision à une structure de diagramme d’influence.

L’ordre partiel induit par les arcs temporels est le suivant :

$$D_1 \ll \{A_1, A_2\} \ll D_2 \ll \{A_3\}$$

De cette manière, les ordres :

$$D_1 \ll A_1 \ll A_2 \ll D_2 \ll A_3 \text{ et } D_1 \ll A_2 \ll A_1 \ll D_2 \ll A_3$$

sont tous deux des ordres valides d’observation des variables puisqu’ils sont compatibles avec le précédent ordre partiel.

Pour chaque nœud de décision  $D \in \mathcal{N}_D$ , on note  $\mathcal{V}_{\text{passe}}(D)$  (resp.  $\mathcal{V}_{\text{futur}}(D)$ ) l’ensemble de toutes les variables  $X \in \mathcal{N}_D \cup \mathcal{N}_C$  vérifiant  $X \ll D$  (resp.  $D \ll X$ ). On

note de plus  $V_{\text{passe}}(D)$  (resp.  $V_{\text{futur}}(D)$ ) le vecteur  $\langle X_1, \dots, X_k \rangle$  de toutes les variables de  $\mathcal{V}_{\text{passe}}(D)$  (resp.  $\mathcal{V}_{\text{futur}}(D)$ ).

Soit  $v = \langle x_1, \dots, x_k \rangle$  un vecteur d'affectations des variables  $X_1, \dots, X_k$  tels que  $\forall 1 \leq i \leq k, X_i \in \mathcal{V}_{\text{passe}}(D), x_i \in D(X_i)$ . On note  $\text{futur}(D|V_{\text{passe}}(D) = v)$  le futur de la variable  $D$  conditionnellement aux événements passés et aux décisions déjà prises (i.e., le vecteur d'affectation  $v$ ). De manière plus formelle,  $\text{futur}(D|V_{\text{passe}}(D) = v)$  est le couple :

$$\langle P(V_{\text{futur}}(D)|V_{\text{passe}}(D) = v, D), u(V_{\text{futur}}(D), D, V_{\text{passe}}(D) = v) \rangle$$

où  $P(V_{\text{futur}}(D)|V_{\text{passe}}(D) = v, D)$  représente la distribution de probabilité jointe sur les variables de  $\mathcal{V}_{\text{futur}}(D)$  conditionnellement au vecteur d'affectation  $v$  du vecteur  $V_{\text{passe}}(D)$  et à la valeur de la variable  $D$ , et  $u(V_{\text{passe}}(D) = v, D, V_{\text{futur}}(D))$  est la valeur des utilités selon les valeurs des variables de  $\mathcal{V}_{\text{futur}}(D)$  et la valeur de la variable  $D$ , conditionnellement au vecteur d'affectation  $v$  du vecteur  $V_{\text{passe}}(D)$ .

**Exemple 17** *Considérons le diagramme d'influence représenté sur la figure 4.1 page 115. On a  $V_{\text{passe}}(D_1) = \langle \rangle$  où  $\langle \rangle$  est le vecteur vide, et  $V_{\text{futur}}(D_1) = \langle A_1, A_2, D_2, A_3 \rangle$ . On a donc  $\text{futur}(D_1) = \langle P(A_1, A_2, D_2, A_3|D_1), u(D_1, A_1, A_2, D_2, A_3) \rangle$  où  $P(A_1, A_2, D_2, A_3|D_1) = P(A_1) \times P(A_2|A_1) \times P(A_3|A_2, D_2)$  et  $u(D_1, A_1, A_2, D_2, A_3) = u(D_1, D_2, A_3)$ . De même, on a  $V_{\text{passe}}(D_2) = \langle D_1, A_1, A_2 \rangle$ , et  $V_{\text{futur}}(D_2) = \langle A_3 \rangle$ . Le couple  $\text{futur}(D_2|D_1 = \alpha, A_1 = \theta_1, A_2 = \theta'_1)$  s'écrit alors comme suit :*

$$\langle P(A_3|D_1 = \alpha, A_1 = \theta_1, A_2 = \theta'_1, D_2), u(D_1 = \alpha, A_1 = \theta_1, A_2 = \theta'_1, D_2, A_3) \rangle$$

où  $P(A_3|D_1 = \alpha, A_1 = \theta_1, A_2 = \theta'_1, D_2) = P(A_1 = \theta_1) \times P(A_2 = \theta'_1|A_1 = \theta_1) \times P(A_3|A_2 = \theta'_1, D_2)$ , et  $u(D_1 = \alpha, A_1 = \theta_1, A_2 = \theta'_1, D_2, A_3) = u(D_1 = \alpha, D_2, A_3)$ .

Enfin, une stratégie dans un diagramme d'influence est un ensemble de *règles de décision* pour les variables  $D_k$ , où une règle de décision pour  $D_k$  associe une valeur de son domaine  $D(D_k)$  à chaque vecteur d'affectation des variables de  $\mathcal{V}_{\text{passe}}(D_k) \setminus \mathcal{N}_D$ .

**Exemple 18** *Considérons le diagramme d'influence représenté sur la figure 4.1 page 115. Il y a exactement 32 stratégies possibles. Une stratégie possible parmi ces 32 est :*

$$\begin{aligned} \Delta_1 = \{ & D_1 = \alpha, \quad D_2 = \gamma \text{ si } A_1 = \theta_1 \text{ et } A_2 = \theta'_1, \\ & D_2 = \delta \text{ si } A_1 = \theta_1 \text{ et } A_2 = \theta'_2, \\ & D_2 = \delta \text{ si } A_1 = \theta_2 \text{ et } A_2 = \theta'_1, \\ & D_2 = \delta \text{ si } A_1 = \theta_2 \text{ et } A_2 = \theta'_2 \} \end{aligned}$$

Une autre stratégie possible est :

$$\begin{aligned} \Delta_2 = \{ & D_1 = \beta, \quad D_2 = \gamma \text{ si } A_1 = \theta_1 \text{ et } A_2 = \theta'_1, \\ & D_2 = \delta \text{ si } A_1 = \theta_1 \text{ et } A_2 = \theta'_2, \\ & D_2 = \gamma \text{ si } A_1 = \theta_2 \text{ et } A_2 = \theta'_1, \\ & D_2 = \delta \text{ si } A_1 = \theta_2 \text{ et } A_2 = \theta'_2 \} \end{aligned}$$

## 4.2 $\Lambda$ -relaxation de l'ensemble $\overline{\Delta}_*^{DI}$ des stratégies factorisées

Il est important de remarquer qu'une stratégie peut alors avoir une taille exponentielle en la taille du diagramme d'influence (qui dépend directement du nombre de variables et de la taille des tables). En effet, lorsque l'on exprime une stratégie de la façon indiquée dans la section précédente, on construit un ensemble de règles de décision du même ordre de grandeur que l'arbre de décision que l'on obtiendrait si l'on « développait » le diagramme d'influence. La place mémoire gagnée se trouve ainsi mise en défaut dès lors que l'on cherche à expliciter une stratégie. Néanmoins, lorsque l'on utilise le modèle EU, il est possible de tirer profit de l'axiome conséquentialiste pour factoriser l'expression de la stratégie. En effet, cet axiome impose que seul le couple  $futur(D|\mathcal{V}_{passe}(D))$  influence la décision prise en  $D|\mathcal{V}_{passe}(D)$ . Ainsi, si  $futur(D|\mathcal{V}_{passe}(D) = v) = futur(D|\mathcal{V}_{passe}(D) = v')$  avec  $v \neq v'$ , alors on prend la même décision en  $D|\mathcal{V}_{passe}(D) = v'$  et en  $D|\mathcal{V}_{passe}(D) = v$  (c'est-à-dire que l'on prend la même décision dans tous les « futurs » identiques). Cependant, si une variable  $X$  de  $\mathcal{V}_{passe}(D)$  ne modifie jamais le vecteur  $futur(D|\mathcal{V}_{passe}(D))$ , quelle que soit la valeur prise par  $X$ , alors la variable  $X$  n'est pas requise pour définir le futur de  $D|\mathcal{V}_{passe}(D)$ . C'est justement ce qui permet de réduire la taille des stratégies lorsque l'on utilise un critère compatible avec l'axiome conséquentialiste. Il suffit pour cela d'exprimer chaque règle de décision non plus en fonction de toutes les variables de  $\mathcal{V}_{passe}(D)$ , mais uniquement en fonction des variables de  $\mathcal{V}_{passe}(D)$  ayant une influence sur le couple  $futur(D|\mathcal{V}_{passe}(D))$ . Bien évidemment, plus les indépendances entre les variables sont nombreuses, plus le gain en taille mémoire est important pour représenter le problème, et plus l'expression d'une stratégie peut se factoriser. Par la suite, on notera  $\overline{\Delta}^{DI}$  l'ensemble des stratégies d'un diagramme d'influence  $DI$  et  $\overline{\Delta}_*^{DI}$  l'ensemble des stratégies factorisées. On a bien évidemment  $\overline{\Delta}_*^{DI} \subseteq \overline{\Delta}^{DI}$ .

**Exemple 19** Dans la figure 4.3 page 120, nous avons représenté l'arbre de décision correspondant au diagramme d'influence représenté sur la figure 4.1 page 115. Les sous-arbres identiques, qui sont issus des nœuds de décision  $D_2$ , sont formatés de la même manière (gras, pointillés...). En accord avec l'axiome conséquentialiste, dans le modèle EU il existe toujours une stratégie optimale où l'on fixe la même sous-stratégie dans chaque sous-arbre identique. Ainsi, la stratégie  $\Delta_1$  de l'exemple 18 page 118 peut être omise lorsque l'on cherche à optimiser EU puisqu'elle impose de prendre une décision différente dans deux sous-arbres identiques (en pointillés sur la figure). En revanche, la stratégie  $\Delta_2$  de l'exemple 18 page 118 sera envisagée et peut être factorisée de la manière suivante :

$$\Delta_2 = \{D_1 = \beta, D_2 = \gamma \text{ si } A_2 = \theta'_1, D_2 = \delta \text{ si } A_2 = \theta'_2\}$$

En effet, quelle que soit la valeur prise par la variable  $A_1$ , cette valeur n'influence jamais le futur du nœud  $D_2$ . La décision prise au nœud  $D_2$  peut alors s'exprimer uniquement en

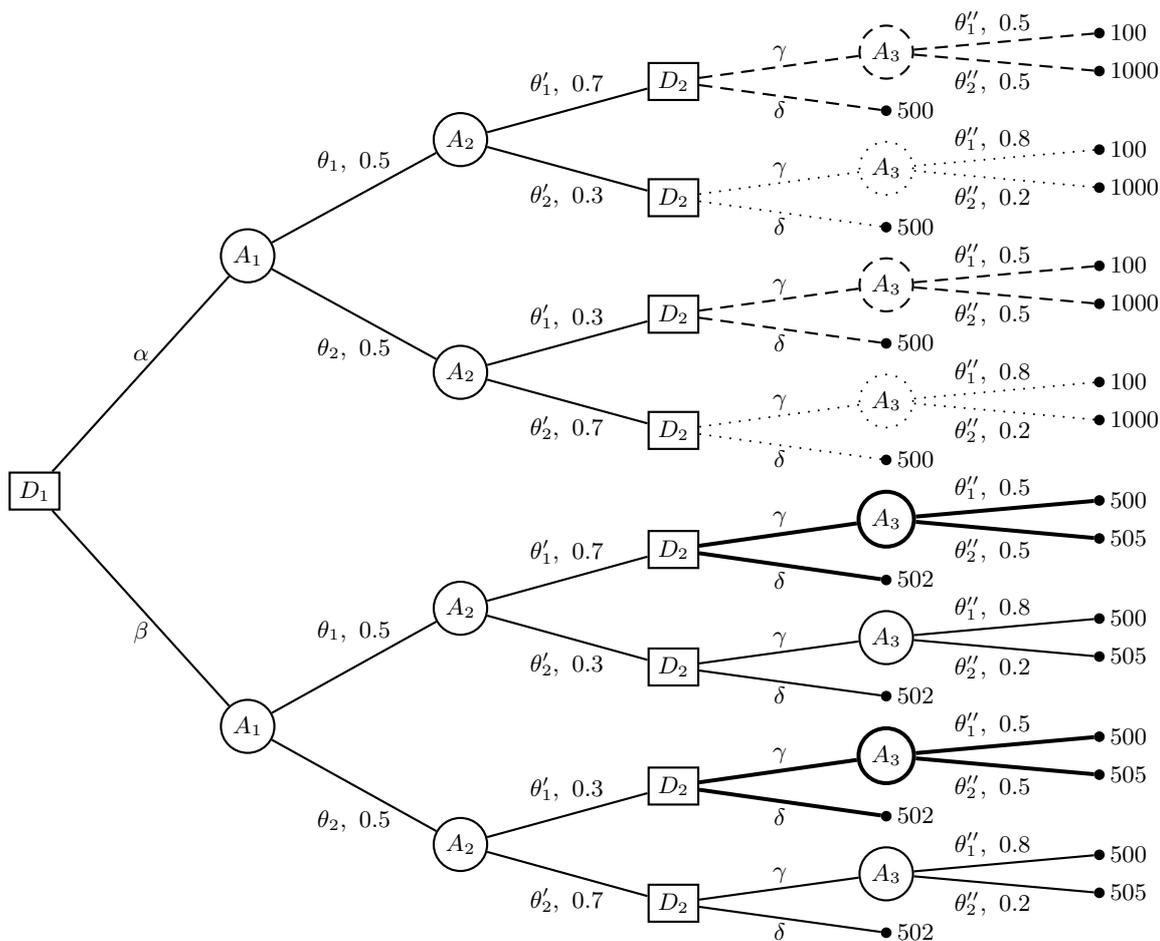


FIGURE 4.3 – Arbre de décision correspondant au diagramme d'influence représenté sur la figure 4.1.

*fonction de la valeur prise par la variable  $A_2$ .*

Lorsque l'on utilise un critère ne respectant pas l'axiome conséquentialiste, il n'est plus possible de représenter la stratégie sous sa forme factorisée puisqu'il devient envisageable de prendre une décision différente en  $D|V_{passe}(D) = v$  et en  $D|V_{passe}(D) = v'$  même si  $futur(D|V_{passe}(D) = v) = futur(D|V_{passe}(D) = v')$ . La taille des stratégies n'est donc plus bornée par la taille d'une stratégie factorisée mais est du même ordre de grandeur que l'arbre de décision que l'on obtient lorsque l'on « développe » le diagramme d'influence. La question qui nous intéresse ici se pose alors tout naturellement : comment peut-on faire usage d'un critère de décision qui ne respecte pas l'axiome conséquentialiste lorsque l'on utilise un diagramme d'influence pour représenter un problème de décision séquentielle ?

Si la mémoire disponible en machine permet le développement d'une stratégie non factorisée, on peut alors utiliser une *méthode en deux phases*, consistant dans un premier temps à transformer le diagramme d'influence sous la forme d'un arbre de décision, puis

dans un second temps à utiliser les techniques existantes dans les arbres de décision (voir chapitre 3). Ce type d'approche en deux phases fait partie des premières méthodes proposées pour optimiser l'espérance d'utilité dans un diagramme d'influence (Howard et Matheson, 1984), et a été repris récemment pour optimiser d'autres critères, comme par exemple les utilités qualitatives dans les diagrammes d'influence possibilistes (Garcia et Sabbadin, 2008).

Cependant, si l'on ne dispose pas de suffisamment de mémoire pour représenter une stratégie sous sa forme non factorisée, il est nécessaire d'envisager une approche différente. Une première approche envisageable pourrait être de déterminer une stratégie optimale parmi les stratégies de l'ensemble  $\overline{\Delta}_*^{DI}$  (i.e., parmi l'ensemble des stratégies factorisées). Néanmoins, une telle approche, sans aucun pré-traitement, rendrait alors caduque tous les avantages descriptifs dont on pouvait bénéficier en utilisant un critère non-EU. Afin de bénéficier tout à la fois des avantages descriptifs d'un critère non-EU et de la compacité de la représentation sous forme d'un diagramme d'influence, il est nécessaire de réaliser un compromis entre la taille des stratégies envisagées et la qualité descriptive de la stratégie. C'est justement ce que nous nous proposons de faire par la suite en réalisant une  $\Lambda$ -relaxation de l'ensemble  $\overline{\Delta}_*^{DI}$ .

L'idée de la  $\Lambda$ -relaxation de l'ensemble  $\overline{\Delta}_*^{DI}$  consiste à rajouter  $\Lambda$  dépendances « factices » sur chacune des variables de décision afin d'augmenter le nombre de stratégies factorisées (i.e., augmenter  $|\overline{\Delta}_*^{DI}|$ ). Si le couple  $futur(D|V_{passe}(D))$  ne dépend pas de la valeur prise par la variable aléatoire  $A \in \mathcal{V}_{passe}(D)$ , alors l'ajout d'une dépendance « factice » de la variable  $A$  sur la variable  $D$  revient à considérer maintenant que le couple  $futur(D|V_{passe}(D))$  dépend de la valeur de  $A$ . De cette manière, dans sa forme factorisée, la règle de décision associée à la variable  $D$  doit maintenant prendre en compte les valeurs prises par la variable  $A$ . On note  $\overline{\Delta}_\Lambda^{DI}$  l'ensemble des stratégies factorisées après une  $\Lambda$ -relaxation de l'ensemble  $\overline{\Delta}_*^{DI}$ . On a donc  $\overline{\Delta}_*^{DI} = \overline{\Delta}_0^{DI} \subseteq \overline{\Delta}_1^{DI} \subseteq \overline{\Delta}_2^{DI} \subseteq \dots \subseteq \overline{\Delta}^{DI}$  si on suppose qu'une  $(k+1)$ -relaxation de l'ensemble  $\overline{\Delta}_*^{DI}$  est obtenue après une 1-relaxation de l'ensemble obtenu à partir d'une  $k$ -relaxation de l'ensemble  $\overline{\Delta}_*^{DI}$ . En effet, notons que les éléments composant l'ensemble  $\overline{\Delta}_\Lambda^{DI}$ , dépend bien évidemment de la procédure qui gouverne l'ajout de dépendances « factices ».

Dans le diagramme d'influence représenté sur la figure 4.1 page 115, quelle que soit la valeur de la variable aléatoire  $A_1$ ,  $futur(D_2|D_1, A_1, A_2)$  conserve toujours la même valeur. La variable  $A_1$  n'influence donc pas le futur de  $D_2|D_1, A_1, A_2$ . Le rajout d'une dépendance « factice » de la variable aléatoire  $A_1$  sur la variable de décision  $D_2$  permet alors d'élargir le champs des stratégies factorisées que l'on note  $\overline{\Delta}_1^{DI}$  (en effet, aucun ajout de dépendance fictive ne peut être fait sur la variable  $D_1$ ). En l'occurrence, après l'ajout de la dépendance de la variable  $A_1$  sur la variable  $D_2$ , on a  $\overline{\Delta}^{DI} = \overline{\Delta}_1^{DI}$  puisque

$futur(D_2|V_{passe}(D_2))$  dépend alors de toutes les variables aléatoires de  $\mathcal{V}_{passe}(D_2)$ .

Il est important de préciser que le rajout de  $\Lambda$  dépendances « factices » sur chaque variable de décision ne modifie pas le problème de décision d'origine, mais uniquement l'ensemble  $\overline{\Delta}_*^{DI}$  des stratégies factorisées nouvellement noté  $\overline{\Delta}_\Lambda^{DI}$ .

### 4.3 Algorithme par énumération implicite

Dans cette partie, nous proposons une approche opérationnelle permettant d'appliquer le critère de décision RDU dans le formalisme des diagrammes d'influence. Ce que l'on se propose de faire ici est de réaliser une  $\Lambda$ -relaxation de l'ensemble  $\overline{\Delta}_*^{DI}$ , puis de déterminer la stratégie optimale au sens du critère RDU parmi les stratégies de l'ensemble  $\overline{\Delta}_\Lambda^{DI}$ . La stratégie que nous cherchons à déterminer n'est donc pas nécessairement la stratégie qui optimise le critère RDU dans l'ensemble  $\overline{\Delta}^{DI}$  de toutes les stratégies, mais celle qui optimise le critère RDU parmi les stratégies dont la taille n'est pas trop grande pour pouvoir être représentées en mémoire.

#### 4.3.1 Mise en œuvre algorithmique de la $\Lambda$ -relaxation de l'ensemble $\overline{\Delta}_*^{DI}$

Dans la suite, nous proposons une méthode opérationnelle permettant de mettre en œuvre une  $\Lambda$ -relaxation de l'ensemble  $\overline{\Delta}_*^{DI}$ .

Dans un diagramme d'influence  $DI = (\mathcal{N}, \mathcal{E})$ , étant donnée une variable de décision  $D \in \mathcal{N}_D$ , pour qu'une variable aléatoire  $A \in \mathcal{N}_C$  conditionne le couple  $futur(D|V_{passe}(D))$ , il suffit que  $A \in \mathcal{V}_{passe}(D)$  et qu'elle conditionne une variable  $A' \in \mathcal{V}_{futur}(D)$ . Ainsi, pour créer une dépendance de la variable  $A$  sur la variable  $D$  (i.e. imposer à la variable  $A$  de conditionner le couple  $futur(D|V_{passe}(D))$ ), on peut ajouter un arc de dépendance  $(A, A')$  à  $\mathcal{E}$  tel que  $A' \in \mathcal{N}_C \cup \mathcal{N}_U, A \ll D \ll A'$ .

Il convient maintenant d'éclaircir les conséquences induites par le fait d'« ajouter un arc de dépendance  $(A, A')$  ». L'ajout d'un arc  $(A, A')$  s'accompagne d'une mise à jour de la table associée à la variable  $A'$  : la table de probabilité conditionnelle de  $A'$  doit maintenant tenir compte de la variable  $A$  (i.e., on duplique la table pour chaque modalité de  $A$ ). Ainsi, ajouter un arc  $(A, A')$  incrémente de  $(|D(A)| - 1) \cdot t(A')$  la taille  $t(A') = \prod_{B \in \text{pere}(A')} |D(B)| \cdot |D(A')|$  de la table de probabilité conditionnelle associée au nœud  $A'$ .

De cette manière, pour réaliser une  $\Lambda$ -relaxation de l'ensemble  $\overline{\Delta}_*^{DI}$ , il suffit de créer  $\Lambda$  arcs au plus<sup>2</sup> pour chaque variable de décision, soit au plus  $\Lambda \cdot |\mathcal{N}_D|$  arcs.

---

2. Un arc de dépendance  $(A, A')$  crée une dépendance de  $A$  sur chaque nœud de décision  $D$  qui vérifie  $A \ll D \ll A'$ .

Nous présentons ici une procédure gloutonne afin de déterminer un ensemble d'arcs de dépendance à ajouter au diagramme d'influence.

Cette procédure gloutonne vise tout d'abord à limiter au maximum la taille des stratégies en localisant les dépendances nouvellement créées autour des nœuds de décision (i.e., la tête et la queue de l'arc sont choisis dans les ensembles  $I_j$  et  $I_i$  avec  $j < i$ ,  $j$  étant le plus grand possible et  $i$  étant l'indice du nœud de décision sur lequel on désire créer une dépendance). En effet, les algorithmes les plus efficaces d'inférences dans les diagrammes d'influence se font généralement dans un *arbre de jonction* construit à partir du diagramme d'influence. Pour construire un tel arbre, il est nécessaire de trianguler le diagramme d'influence. Or il se trouve que la taille de l'arbre de jonction dépend directement de la taille des cliques dans le graphe triangulé. Cette taille est toujours supérieure à la taille des stratégies factorisées. Il apparaît alors que diminuer au maximum la taille des cliques dans le graphe triangulé permet de borner la taille des stratégies envisagées. Nous faisons ici l'hypothèse intuitive que la taille des cliques, dans le graphe triangulé, est fortement dépendante de « l'éparpillement » des voisins de chaque nœud du graphe, et que plus ces derniers seront « groupés » plus la taille des cliques résultant de la triangulation devrait être basse.

Dans un second temps, cette procédure vise à limiter l'augmentation de la taille du diagramme d'influence. Pour ce faire, lorsque l'on ajoute un arc  $(A, A')$ , on s'efforce de choisir  $A$  et  $A'$  de manière à ce que la taille de la table associée au nœud  $A'$  augmente le moins possible. L'ajout d'un arc de dépendance  $(A, A')$  incrémente la taille du diagramme d'influence de  $(|D(A)| - 1) \cdot t(A')$ , où  $t(A')$  est la taille de la table en  $A'$ . De ce fait, lorsqu'il y a plusieurs candidats pour jouer le rôle du nœud  $A$ , on choisit celui de modalité la plus basse. Le nœud  $A$  étant maintenant imposé, pour diminuer cet incrément il est alors nécessaire de choisir la variable  $A'$  qui minimise la valeur de  $t(A')$ . Ainsi, s'il y a plusieurs candidats pour jouer le rôle de  $A'$ , on sélectionne celui dont la table associée possède la plus petite taille.

Le détail de la procédure est donné dans l'algorithme 4 page 125. La procédure *ajoutArcs* prend en argument un diagramme d'influence et lui ajoute des arcs jusqu'à augmenter de  $\Lambda$  (lorsque c'est possible) le nombre de variables conditionnant au moins une variable aléatoire de  $\mathcal{V}_{futur}(D)$ , pour chaque nœud de décision  $D$ . Dans un premier temps, pour chaque nœud de décision  $D_i$ , on calcule le nombre  $n_i$  de variables qui conditionnent au moins une variable de  $\mathcal{V}_{futur}(D_i)$ . Ensuite, pour chaque nœud de décision  $D_i$ , tant que le nombre de variables qui conditionnent au moins une variable aléatoire de  $\mathcal{V}_{futur}(D_i)$  est strictement inférieur à  $n_i + \Lambda$  et qu'il est possible d'ajouter des dépendances sur le nœud  $D_i$ , on continue d'ajouter des arcs. L'arc ajouté débute d'un nœud de chance de  $\mathcal{V}_{passe}(D_i)$  qui ne conditionne aucune variable aléatoire de  $\mathcal{V}_{futur}(D_i)$  (variable  $A$  dans

l'algorithme), et aboutit à un nœud de chance de  $\mathcal{V}_{futur}(D_i)$  (variable  $A'$  dans l'algorithme). La variable  $A$  est choisie dans l'ensemble  $I_j$  tel que  $j$  soit le plus grand possible tout en étant strictement inférieur à  $i$ . Si plusieurs variables sont candidates, on en choisit une parmi celles possédant la modalité la plus basse. Quant à la variable  $A'$ , on la choisit dans l'ensemble  $I_i$ . Si plusieurs variables restent encore candidates, on en choisit une parmi celles qui possèdent la table de plus petite taille. Une fois l'arc  $(A, A')$  ajouté, on met à jour la table en  $A'$  pour y inclure un conditionnement selon les valeurs prises par la variable  $A$ . Lorsque l'on ne peut plus ajouter d'arcs (ou que l'on a ajouté suffisamment d'arcs pour réaliser une  $\Lambda$ -relaxation de l'ensemble  $\overline{\Delta}_*^{DI}$ ), l'algorithme s'arrête.

### 4.3.2 Détermination de la stratégie optimisant le critère RDU dans l'ensemble $\overline{\Delta}_\Lambda^T$

Dans cette partie, nous proposons un algorithme permettant de déterminer une stratégie, appartenant à l'ensemble  $\overline{\Delta}_\Lambda^T$ , qui optimise le critère RDU. Pour ce faire, nous adaptons, dans le cadre des diagrammes d'influence, la méthode par énumération implicite proposée dans le formalisme des arbres de décision (voir la section 3.3 page 91). Le principe est le même que dans les arbres de décision. Nous présentons ci-dessous les éléments clefs de l'algorithme.

#### Principe de branchement

Nous partitionnons l'ensemble des stratégies en plusieurs sous-ensembles, chacun d'eux étant différencié par une règle de décision. Le principe de séparation est donc sensiblement le même que celui proposé dans le cadre des arbres de décision. Par exemple, si l'on considère le diagramme d'influence représenté sur la figure 4.1 page 115, l'ensemble de stratégies caractérisé par l'ensemble de règles de décision  $\{D_1 = \alpha\}$  (i.e., l'ensemble des stratégies respectant ces règles de décision) peut être partitionné en deux sous-ensembles de stratégies, respectivement caractérisés par les ensembles de règles de décision suivants :

- $\{D_1 = \alpha, D_2 = \gamma \text{ si } A_2 = \theta'_1\}$
- $\{D_1 = \alpha, D_2 = \delta \text{ si } A_2 = \theta'_1\}$

L'arbre d'énumération correspondant est représenté sur la figure 4.4. A titre d'illustration, nous avons mis en gras le chemin menant au nœud représentant l'ensemble de stratégies caractérisé par l'ensemble de règles de décision  $\{D_1 = \alpha, D_2 = \gamma \text{ si } A_2 = \theta'_1\}$ .

Ainsi, on explore des sous-ensembles de stratégies que l'on raffine par des règles de décision. Si plusieurs variables de décision sont susceptibles d'être instanciées pour étendre l'ensemble de règles de décision considéré, on sélectionne la variable de décision  $D_i$  minimisant l'indice  $i$ . En d'autres termes, on n'étend pas l'ensemble des règles de décision avec un règle de décision sur la variable de décision  $D_j$  s'il reste des règles de décision compatibles pour la variable de décision  $D_i, i < j$ . Enfin, parmi les différentes

---

**Algorithme 4 : ajoutArcs**(Diagramme d'influence  $DI = (\mathcal{N}, \mathcal{E})$ , réel  $\Lambda$ )
 

---

**Sortie :** Un diagramme d'influence sur lequel a été appliqué une  $\Lambda$ -relaxation de l'ensemble  $\overline{\Delta}_*^{DI}$ 

/\* L'ordre d'observation des variables est le suivant :  $I_0 \ll D_1 \ll I_1 \ll \dots \ll D_n \ll I_n$  \*/  
 /\* On dit qu'une variable aléatoire  $A$  conditionne la variable  $D_i$  si  $A \in \mathcal{V}_{passe}(D_i)$  et  $\exists A' \in \mathcal{V}_{futur}(D_i), (A, A') \in \mathcal{E}$  \*/

**pour chaque**  $D_i \in \mathcal{N}_D$  **faire**

/\* **nb**( $D_i$ ) retourne  $|\{V : V \in \mathcal{V}_{passe}(D_i), \exists V' \in \mathcal{V}_{futur}(D_i), (V, V') \in \mathcal{E}\}|$   
 i.e. **nb**( $D_i$ ) retourne le nombre de variables aléatoires qui conditionnent  $D_i$  \*/

/\*  $n_i$  est le nombre initial de variables aléatoires qui conditionnent  $D_i$  \*/  
 $n_i \leftarrow \mathbf{nb}(D_i);$

**pour**  $i$  variant de  $|\mathcal{N}_D|$  à 1 avec un pas de  $-1$  **faire**

/\* Cas du nœud de décision  $D_i$  \*/

**pour**  $j$  variant de  $i - 1$  à 1 avec un pas de  $-1$  **faire**

/\* Cas de l'ensemble  $I_j$  \*/

**tant que**  $(n_i + \Lambda > \mathbf{nb}(D_i))$  ET  $(\exists A \in I_j, \forall A' \in \mathcal{V}_{futur}(D_i), (A, A') \notin \mathcal{E})$ 
**faire**

/\* On est ici uniquement si on a pas encore ajouté  $\Lambda$  dépendances sur le nœud  $D_i$   
 ET qu'il existe au moins une variable de  $I_j$  qui ne conditionne pas  $D_i$  \*/

/\*  $A$  est la variable de  $I_j$  de nombre de modalités minimal parmi celles qui ne conditionnent pas  $D_i$  \*/

$$A \leftarrow \arg \min_{V \in I_j} \{|V| : \forall V' \in \mathcal{V}_{futur}(D_i), (V, V') \notin \mathcal{E}\};$$

/\*  $A'$  est la variable de  $I_i$  ayant la plus petite table de probabilité conditionnelle \*/

$$A' \leftarrow \arg \min_{V \in I_i} \{|V| \cdot \prod_{V' \in \text{pere}(V)} |V'|\};$$

/\* On ajoute l'arc  $(A, A')$  au diagramme \*/

$$\mathcal{E} \leftarrow \mathcal{E} \cup \{(A, A')\};$$

Mettre à jour la table associée à  $A'$ ;

**Retourner**  $DI$ ;
 

---

règles de décision compatibles de la variable  $D_i$ , on choisit celle conduisant au sous-ensemble de stratégies qui comporte la stratégie ayant la plus haute valeur d'utilité. Chaque table du diagramme d'influence est alors mise à jour en fonction des règles de décision qui sont ajoutées à l'ensemble des règles de décision considérées, c'est-à-dire que

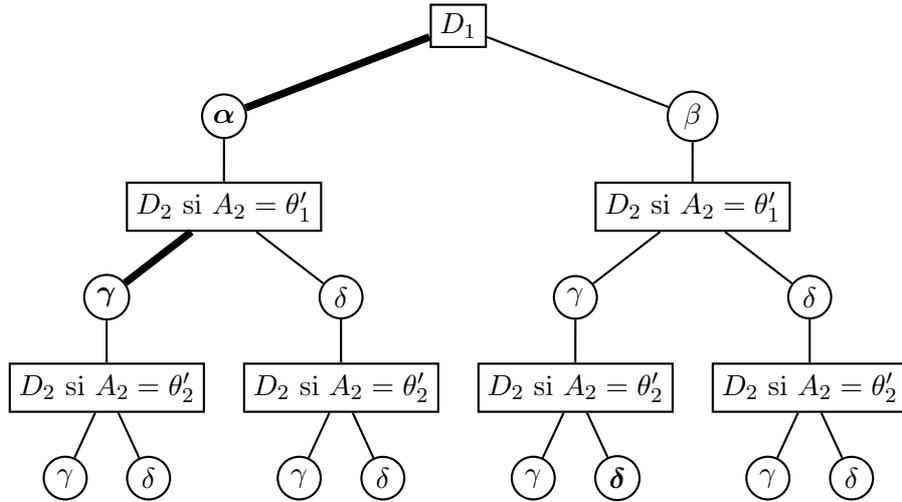


FIGURE 4.4 – Arbre d'énumération correspondant au diagramme d'influence représenté sur la figure 4.1 page 115. Les arcs en gras décrivent l'ensemble de règles de décision suivant  $\{D_1 = \alpha, D_2 = \gamma \text{ si } A_2 = \theta'_1\}$ .

l'on supprime les entrées incompatibles avec l'ensemble des règles de décision considérées.

### Initialisation

On initialise la procédure de branch and bound avec la stratégie qui optimise le critère EU. La valeur selon le critère RDU de la stratégie optimisant le critère EU constitue la valeur initiale de la borne inférieure. Par soucis de précision, nous rappelons maintenant comment déterminer la stratégie qui optimise le critère EU dans un diagramme d'influence. On procède par programmation dynamique à l'image de ce qui se fait dans les arbres de décision. Sans perte de généralité, par soucis de simplicité nous faisons l'hypothèse qu'un unique nœud d'utilité est présent dans le diagramme d'influence. Soit  $U$  cette variable d'utilité, à valeur dans  $\{u_1, \dots, u_m\}$  avec  $u_1 \leq \dots \leq u_m$ . L'initialisation se fait à partir de la variable  $U$  suivant la formule :

$$\forall i \geq 1, \quad P(U = u_i | I_0, \dots, I_n, D_1, \dots, D_n) = \begin{cases} 1 & \text{si } U(I_0, \dots, I_n, D_1, \dots, D_n) = u_i \\ 0 & \text{sinon} \end{cases}$$

où  $U(I_0, \dots, I_n, D_1, \dots, D_n)$  est la valeur prise par la variable  $U$  en fonction des valeurs prises par les variables  $I_0, \dots, I_n, D_1, \dots, D_n$ . L'utilité espérée maximale  $EU_n$  pour la décision  $D_n$  est donnée par la formule suivante :

$$EU_n = \max_{D_n} \sum_{j=i}^m \sum_{I_n} P(I_n | I_0, \dots, I_{n-1}, D_1, \dots, D_n) \times P(U = u_i | I_0, \dots, I_n, D_1, \dots, D_n)$$

Par suite, pour tout  $k \in \{1, \dots, n-1\}$ , l'utilité espérée maximale  $EU_k$  pour la décision  $D_k$ , est donnée par la formule suivante :

$$EU_k = \max_{D_k} \sum_{j=i}^m \sum_{I_k} P(I_k | I_0, \dots, I_{k-1}, D_1, \dots, D_k) \times EU_{k-1}$$

Pour réaliser ces calculs de manière efficace, il existe de nombreux algorithmes proposés dans la littérature. Parmi les plus célèbres, on peut citer l'algorithme de Shachter (1986) qui effectue tous les calculs directement dans le diagramme d'influence, ainsi que l'algorithme de Jensen et al. (1994), s'inspirant fortement des techniques d'inférence dans les réseaux bayésiens, qui fait appel au formalisme des *arbres de jonction* (Robertson et Seymour, 1984) pour réaliser les différents calculs. Notre contribution ne se situant pas dans les techniques d'inférences au sein d'un diagramme d'influence, nous ne détaillerons pas plus ce point.

#### Calcul de la borne inférieure

De la même manière que dans les arbres de décision, en chaque nœud de l'arbre d'énumération, on détermine une stratégie qui optimise le critère EU dans le sous-ensemble de stratégies considéré, puis on calcule la valeur RDU de cette dernière.

#### Calcul de la borne supérieure

Le principe, une fois de plus, est le même que dans les arbres de décision. En chaque nœud de l'arbre d'énumération, on construit une loterie qui domine stochastiquement au premier ordre l'ensemble des loteries correspondant au sous-ensemble de stratégies considéré. La valeur selon le critère RDU de cette loterie constitue alors une borne supérieure puisque le critère RDU est compatible avec la dominance stochastique du premier ordre.

Pour déterminer une telle loterie, on procède par programmation dynamique. A nouveau, sans perte de généralité, par soucis de simplicité nous faisons l'hypothèse qu'il y a un unique nœud d'utilité. Soit  $U$  cette variable d'utilité à valeurs dans  $\{u_1, \dots, u_m\}$  avec  $u_1 \leq \dots \leq u_m$ . L'initialisation se fait à partir de la variable  $U$  suivant la formule :

$$\forall i \geq 1, \quad P(U = u_i | I_0, \dots, I_n, D_1, \dots, D_n) = \begin{cases} 1 & \text{si } U(I_0, \dots, I_n, D_1, \dots, D_n) = u_i \\ 0 & \text{sinon} \end{cases}$$

où  $U(I_0, \dots, I_n, D_1, \dots, D_n)$  est la valeur prise par la variable  $U$  en fonction des valeurs prises par les variables  $I_0, \dots, I_n, D_1, \dots, D_n$ . Considérons maintenant la variable de décision  $D_k$ . Etant donnée les valeurs prises par les variables  $I_0, \dots, I_{k-1}, D_1, \dots, D_{k-1}$ , la loterie calculée en  $D_k$  est obtenue par la formule suivante, pour tout  $i \geq 1$  :

$$\begin{aligned} & P(U \geq u_i | I_0, \dots, I_{k-1}, D_1, \dots, D_{k-1}) \\ &= \max_{D_k} \sum_{j=i}^m \sum_{I_k} P(I_k | I_0, \dots, I_{k-1}, D_1, \dots, D_k) \times P(U = u_j | I_0, \dots, I_{k-1}, I_k, D_1, \dots, D_k) \end{aligned}$$

En effet, la loterie que l'on désire construire se déduit alors facilement comme suit :

$$\begin{aligned} & P(U = u_i | I_0, \dots, I_{k-1}, D_1, \dots, D_{k-1}) \\ = & P(U \geq u_i | I_0, \dots, I_{k-1}, D_1, \dots, D_{k-1}) - P(U \geq u_{i+1} | I_0, \dots, I_{k-1}, D_1, \dots, D_{k-1}) \end{aligned}$$

La valeur qui est finalement retournée est la valeur selon le critère RDU de la loterie construite en  $D_1$ .

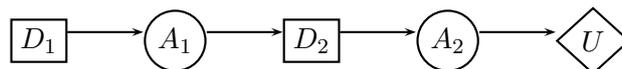
Pour réaliser ces calculs de manière efficace, il suffit de réaliser une simple adaptation des algorithmes de la littérature qui permettent de calculer une stratégie EU optimale. En effet, les calculs sont sensiblement les mêmes (voir la partie précédente sur l'initialisation), sauf qu'au lieu de calculer une espérance, on construit une loterie. Ainsi, les algorithmes de Shachter (1986) et de Jensen et al. (1994) permettent de réaliser ces calculs avec efficacité.

#### 4.4 Expérimentations numériques

Tous les algorithmes ont été implémentés avec le langage C++. Les expérimentations numériques ont été réalisées sur un ordinateur doté d'un processeur Pentium IV CPU 2.13GHz et de 3.5Go de RAM. Nous présentons dans cette partie les résultats que nous avons obtenu sur des diagrammes d'influence générés aléatoirement. Pour ces expérimentations numériques nous avons choisi la fonction  $\varphi$  de déformation des probabilités suivante :  $\forall p \in [0; 1], \varphi(p) = p^\lambda / (p^\lambda + (1 - p)^\lambda)$ , avec  $\lambda = 0.2$ . Cette forme de fonction  $\varphi$  est classique dans la littérature économique (voir chapitre 1) et la valeur 0.2 pour le paramètre  $\lambda$  permet de ne pas être trop proche du critère EU.

##### Génération des diagrammes d'influence

Concernant la génération aléatoire des diagrammes d'influence, nous avons procédé de la manière suivante. Etant donné un nombre de nœuds de décision  $n$ , nous avons dans un premier temps créé une chaîne de nœuds de longueur  $2n + 1$  qui alterne les nœuds de décision et les nœuds de chance pour s'achever par un nœud d'utilité. Par exemple, pour  $n = 2$  nœuds de décision, nous obtenons le graphe suivant :

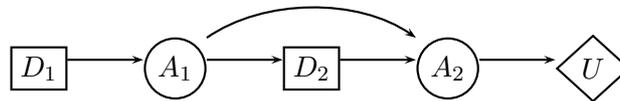


Ensuite, nous avons ajouté des arcs de manière à ce que chaque nœud de chance soit relié à un autre nœud de chance. Cela permet de rendre plus dense le diagramme

	$n = 4$	$n = 5$	$n = 6$	$n = 7$	$n = 8$	$n = 9$	$n = 10$
$\Lambda = 0$	0	0	0	0.21	0.85	2.78	9.11
$\Lambda = 1$	0	0	0	0.23	0.62	3.94	12.15
$\Lambda = 2$	0	0	0	0.38	1.10	4.79	23.36
$\Lambda = 3$	0	0	0.10	0.72	2.33	9.25	29.97
$\Lambda = 4$	0	0	0.56	1.07	4.46	18.61	93.52
$\Lambda = 5$	0	0.08	1.91	3.43	19.84	87.40	—
2 ph.	0.17	0.93	5.02	17.38	—	—	—

TABLE 4.1 – Temps d'exécution (sec.).

d'influence et de s'assurer que la décision prise en chaque nœud de décision soit influente sur la valeur de la stratégie. Ce qui nous donne le graphe suivant (pour  $n = 2$ ) :



Enfin, nous avons ajouté aléatoirement des nœuds de chance et des arcs entre les différents nœuds. Dans le graphe obtenu, nous avons généré en chaque nœud de chance et chaque nœud d'utilité, de manière aléatoire, des tables de probabilités conditionnelles et d'utilités (les valeurs d'utilités sont des réels compris entre 0 et 500).

### Résultats numériques

Tout d'abord, nous avons comparé les temps d'exécution entre la méthode en deux phases<sup>3</sup> et l'algorithme d'énumération implicite proposé ici. Pour ce dernier algorithme, il convient de préciser que nous avons effectué les calculs de la borne supérieure et de la borne inférieure suivant le principe de l'algorithme proposé par Jensen et al. (1994). Cet algorithme consiste à transformer le diagramme d'influence en un arbre de jonction, qui est alors une structure adaptée pour faire de la programmation dynamique.

Pour la comparaison, nous avons généré 40 instances aléatoirement pour chaque valeur de  $n$  (nombre de nœuds de décision). Le tableau 4.1 donne les temps de résolution que nous avons obtenus (le symbole '-' apparaît lorsqu'il y a eu au moins une instance qui a mis plus de 30 minutes à s'exécuter). Pour chaque colonne, nous avons fait varier le nombre  $n$  de nœud de décision de 4 à 10 et pour chaque ligne nous avons fait varier la valeur de  $\Lambda$  (le nombre de dépendances créées sur chaque variable de décision).

Sans surprise, nous constatons que plus  $n$  et  $\Lambda$  augmentent, plus les temps d'exécutions deviennent élevés. Notons que la méthode en deux phases ne permet pas de traiter

3. La méthode en deux phases consiste à transformer, dans un premier temps, le diagramme d'influence en un arbre de décision, puis, dans un second temps, à déterminer une stratégie qui optimise le critère RDU dans l'arbre de décision à l'aide de l'algorithme de branch and bound proposé dans le chapitre 2.

	$\Lambda = 0$	$\Lambda = 2$	$\Lambda = 3$	$\Lambda = 4$	$\Lambda = 5$
$n = 2$	97%	100%	100%	100%	100%
$n = 3$	93%	96%	98%	100%	100%
$n = 4$	92%	93%	95%	97%	97%
$n = 5$	93%	94%	95%	95%	97%
$n = 6$	92%	93%	93%	95%	96%

TABLE 4.2 – Influence du paramètre  $\Lambda$  sur la qualité de la stratégie.

dans un temps raisonnable les instances ayant plus de 7 nœuds de décision, contrairement à la seconde approche qui est capable de traiter de plus grosses instances (au prix de l'optimalité bien entendu).

Dans un second temps nous nous sommes intéressés à mesurer la qualité de la solution optimale dans l'ensemble  $\overline{\Delta}_\Lambda^{DI}$  avec la mise en œuvre que nous proposons ici. Pour ce faire, nous avons comparé la valeur selon le critère RDU de la stratégie retournée avec la valeur optimale de la stratégie RDU déterminée à l'aide de la méthode en deux phases. Etant donné un diagramme d'influence  $DI$ , nous notons  $RDU^*(DI)$  la valeur de la stratégie optimale dans  $\overline{\Delta}_*^{DI}$  et  $RDU_\Lambda(DI)$  la valeur de la stratégie optimale dans  $\overline{\Delta}_\Lambda^{DI}$ .

Nous avons observé la valeur du ratio  $RDU_\Lambda(DI)/RDU^*(DI)$  pour différentes valeurs de  $\Lambda$  et de  $n$ . Pour chaque valeur de  $n$ , 200 diagrammes d'influence ont été générés aléatoirement. Nous avons ensuite appliqué la méthode proposée pour réaliser une  $\Lambda$ -relaxation de l'ensemble  $\overline{\Delta}_*^{DI}$  sur chacune des 200 instances en faisant varier la valeur de  $\Lambda$ . Autrement dit, pour  $n$  fixé, ce sont les 200 même instances qui ont subi une 0-relaxation, une 1-relaxation, et ainsi de suite.

Le tableau 4.2 donne la valeur du ratio  $RDU_\Lambda(DI)/RDU^*(DI)$  moyen sur les 200 diagrammes d'influence pour chaque jeu de paramètres. Il semble qu'une 5-relaxation fournit d'excellentes stratégies, du moins pour les instances générées ici. Notons que, du fait que les tables soient générées de manière aléatoire en suivant une loi uniforme, l'éparpillement des valeurs des stratégies selon le critère EU ou le critère RDU est relativement faible.

## Conclusion

Les diagrammes d'influence présentent l'avantage d'offrir une représentation bien plus compactes que les arbres de décision en tirant profit des indépendances entre les différentes variables intervenant dans le problème de décision séquentielle. Cependant, si le problème peut être modélisé de manière compacte, les stratégies restent elles de taille exponentielle en la taille du diagramme d'influence. L'utilisation du critère EU permet

de factoriser l'expression des stratégies en tirant profit de l'axiome conséquentialiste. Par contre, dès lors que l'on désire utiliser un critère qui ne respecte pas cet axiome pour modéliser les préférences du décideur, comme le critère RDU, il n'est plus possible de factoriser la représentation des stratégies. Il apparaît alors nécessaire de réaliser un compromis entre le pouvoir descriptif du critère utilisé et la taille des stratégies pour tirer profit de la représentation compacte d'un diagramme d'influence.

Nous avons proposé dans ce chapitre l'idée de la  $\Lambda$ -relaxation de l'ensemble  $\overline{\Delta}_*^{DI}$  pour réaliser un tel compromis. Cette relaxation consiste à ajouter  $\Lambda$  dépendances à chaque variable de décision afin d'élargir l'ensemble  $\overline{\Delta}_*^{DI}$  des stratégies factorisées. Nous avons proposé une façon de mettre œuvre une telle relaxation à l'aide d'un algorithme glouton qui rajoute des arcs de dépendance au sein du diagramme d'influence. Une fois cette relaxation réalisée, nous avons adapté l'algorithme d'énumération implicite proposé pour les arbres de décision aux diagrammes d'influence, afin de déterminer une stratégie optimisant le critère RDU dans l'ensemble des stratégies factorisées du diagramme d'influence obtenu. Enfin, à travers une série d'expérimentations numériques, nous avons montré que l'approche proposée était viable au niveau opérationnel.

Il est légitime de s'interroger sur la pertinence du compromis réalisé entre le pouvoir descriptif du critère utilisé (RDU en l'occurrence) et la taille des stratégies explorée. En effet, nous avons vu que le critère RDU permettait de combler les lacunes descriptives du modèle EU. Cependant, ces lacunes existent-elles toujours lorsque l'on restreint l'espace des stratégies ? La réponse est bien entendu positive, puisque certaines stratégies non-dominées de l'ensemble ne sont pas atteignables lorsque l'on cherche à maximiser le critère de l'utilité espérée, et que ces mêmes stratégies sont atteignables avec le critère RDU. Néanmoins, une voie de recherche prometteuse pour améliorer notre approche serait d'ajouter moins d'arcs (i.e., réduire la valeur de  $\Lambda$ ), mais de guider le choix de ces arcs en se fondant sur l'impact de leur ajout sur la qualité de la solution. Autrement dit, il s'agirait d'identifier les arcs dont l'ajout est critique. A l'aide d'une telle méthode, on pourrait espérer obtenir une stratégie factorisée de taille moindre tout en améliorant significativement la qualité de la stratégie. Des approches heuristiques dans ce but nous semblent envisageables.

Une autre limite, imputable directement à la structure des diagrammes d'influence, est que la taille des stratégies, même factorisées, reste encore trop grande dans de nombreux problèmes. Une piste à envisager, est alors d'adapter la méthode proposée ici aux *Limited Memory Influence Diagrams*, abrégés en LIMIDs dans la littérature (Lauritzen et Nilsson, 2001). Cette représentation graphique, fortement inspirée des diagrammes d'in-

fluence, permet de représenter un problème de décision de manière compacte et d'assurer une taille raisonnable pour les stratégies factorisées (i.e., du même ordre de grandeur que la taille de l'instance d'origine). Cependant, l'optimisation exacte du critère EU dans ces derniers est déjà un problème algorithmique complexe.

D'un point de vue algorithmique, le passage du critère EU au critère RDU (ou tout autre critère qui ne respecte pas l'axiome conséquentialiste) a un vrai coût en terme de complexité. Dans un problème de décision séquentielle dans le risque, nous avons le loisir de pouvoir choisir un critère qui respecte l'axiome conséquentialiste, et donc de choisir si l'on peut sacrifier l'efficacité computationnelle au profit de l'efficacité descriptive. Dans le prochain chapitre, nous allons voir que dès lors que nous quittons le cadre de l'incertitude risquée, ce choix n'existe plus, car même les critères les plus simples ne respectent pas l'axiome conséquentialiste.





## Chapitre 5

# Critère de Hurwicz dans les problèmes de décision séquentielle sous incertitude non risquée<sup>1</sup>

**Résumé.** *Dans ce chapitre nous traitons les problèmes de décision séquentielle dans les contextes d'incertitude totale et de risque imprécis. Plus précisément, nous nous intéressons à l'optimisation du critère de Hurwicz dans ces cadres. Dans un premier temps, nous traitons les situations d'incertitude totale et proposons un algorithme polynomial pour l'optimisation du critère de Hurwicz. Dans un second temps nous nous traitons les situations de risque imprécis et nous distinguons les cas où l'arbre de décision est séparable (i.e., toutes les loteries sont mutuellement indépendantes deux à deux) et les cas où il ne l'est pas. Dans le cas où l'hypothèse de séparabilité est faite, nous proposons un algorithme fondé sur la programmation dynamique biobjectif pour optimiser le critère de Hurwicz depuis la racine. Enfin, si l'hypothèse de séparabilité n'est pas faite, on démontre que le problème de la détermination d'une stratégie optimisant le critère de Hurwicz est un problème NP-difficile, et nous adaptons la procédure de veto-process proposée par Jaffray et Nielsen (2006) pour déterminer une stratégie de compromis lorsque les préférences du décideur suivent le critère de Hurwicz.*

---

1. Ce chapitre s'appuie en partie sur des résultats présentés dans (Jeantet et Spanjaard, 2009b), (Jeantet, 2009) et (Jeantet et Spanjaard, 2010).

## Introduction

Dans les chapitres précédents, nous nous sommes focalisés sur les situations d'incertitude risquée, c'est-à-dire les cas où l'incertitude est modélisée sous la forme de distributions de probabilité. Cependant, lorsque plusieurs experts ont des points de vue qui divergent, ou quand certaines données empiriques (parfois toutes) sont manquantes, il n'est pas possible d'éliciter des distributions de probabilités sur chacun des événements. Nous avons vu, dans le chapitre 1, qu'il était possible de se ramener à un cas très proche du risque en établissant une distribution de probabilité sur les événements de manière subjective. Néanmoins, de la même manière que lorsque l'on transforme une utilité ordinale en utilité cardinale, ces données subjectives créent de l'information qui n'existait pas initialement dans le problème. En effet, certains problèmes exigent que l'on différencie les données récoltées de manière objective, qui sont propres au problème de décision mais indépendantes du profil du décideur qui y sera confronté, et les paramètres préférentiels propres au décideur.

Dans ce chapitre, nous traitons le cas où la modélisation de l'incertitude s'appuie exclusivement sur les données objectives fournies par les experts ou les appareils de mesure. Dans un premier temps, nous nous intéressons aux problèmes pour lesquels on ne dispose d'aucune information sur la vraisemblance des événements. Ces problèmes de décision, dits sous incertitude totale, ont été très étudiés dans la littérature économique, mais très peu dans un cadre séquentiel. Nous avons vu, dans le chapitre 1, que le critère de Hurwicz semblait un être critère intéressant pour raisonner en incertitude totale. Aussi, nous nous efforçons ici d'appliquer ce critère dans un cadre séquentiel. Dans un second temps, nous nous intéressons aux problèmes pour lesquels on ne dispose que d'une distribution de probabilité intervalle sur les événements élémentaires des loteries considérées. Une fois encore, nous nous efforçons à appliquer le critère de Hurwicz dans un cadre séquentiel. Cependant, nous différencierons les arbres de décision *séparables* et les arbres de décision *non séparables*. La première famille d'arbres de décision, plus simples à traiter, suppose l'indépendance mutuelle de toutes les variables aléatoires présentes dans le problème. Une telle hypothèse va rendre plus commode la mise en œuvre du critère de Hurwicz. Dans la seconde famille d'arbres de décision, nous montrons tout d'abord que le problème est NP-difficile, puis nous proposons un algorithme de veto-process (Jaffray, 1999) pour appliquer le critère de Hurwicz. Tous les algorithmes proposés ont été soumis à des expérimentations numériques que nous présentons en fin de chaque partie.

### 5.1 Incertitude totale

Dans cette partie nous nous efforçons d'appliquer le critère de Hurwicz dans un arbre de décision lorsque l'incertitude est totale (i.e., le décideur ne détient aucune information

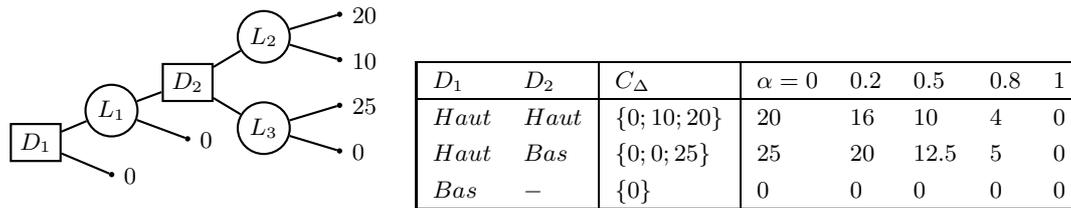


FIGURE 5.1 – Exemple d’arbre de décision et évaluation de ses stratégies par le critère de Hurwicz.

sur la vraisemblance des événements).

Notons que dans ce cadre, une stratégie peut alors se réduire à l’ensemble des conséquences qu’elle peut potentiellement atteindre. En guise d’exemple, considérons l’arbre de décision  $T$  représenté sur la figure 5.1. Dans cet arbre, on note  $\mathcal{C}_T = \{0, 0, 0, 10, 20, 25\}$  l’ensemble des conséquences. Il y a exactement trois stratégies :  $\Delta_1 = \{D_1 = \textit{Haut}, D_2 = \textit{Haut}\}$ ,  $\Delta_2 = \{D_1 = \textit{Haut}, D_2 = \textit{Bas}\}$  et  $\Delta_3 = \{D_1 = \textit{Bas}\}$ . Chaque stratégie  $\Delta$  conduit à un ensemble de conséquences potentielles  $\mathcal{C}_\Delta \subseteq \mathcal{C}_T$ . Par exemple, la stratégie  $\Delta_1 = \{D_1 = \textit{Haut}, D_2 = \textit{Haut}\}$  conduit à l’ensemble de conséquences potentielles  $\mathcal{C}_{\Delta_1} = \{0, 10, 20\}$ . De cette manière, le critère de Hurwicz se définit alors comme suit : soit  $\Delta$  une stratégie menant à l’ensemble de conséquences potentielles  $\mathcal{C}_\Delta = \{u_1, u_2, \dots, u_n\}$  de sorte que  $u_1 \leq u_2 \leq \dots \leq u_n$ , l’évaluation de la stratégie  $\Delta$  par le critère de Hurwicz est  $\alpha u_1 + (1 - \alpha)u_n$  où  $\alpha$  est un réel compris entre 0 et 1. Le tableau de la figure 5.1 donne la valeur de chaque stratégie de l’arbre de décision de gauche en fonction de quelques valeurs de  $\alpha$ . Les colonnes  $D_1$  et  $D_2$  indiquent les décisions prises en chacun des nœuds de décision, la colonne  $\mathcal{C}_\Delta$  donne l’ensemble des conséquences potentielles offertes par la stratégie correspondante.

Dans cette partie, nous nous intéressons à faire du choix résolu depuis la racine. Dans un tel cadre, un décideur ayant des préférences qui suivent le critère de Hurwicz sélectionne alors la stratégie dont l’évaluation est maximale suivant ce critère. Néanmoins, le critère de Hurwicz ne respectant pas l’axiome conséquentialiste, la détermination d’une stratégie par programmation dynamique peut conduire à une stratégie sous-optimale. Nous illustrons ici, à l’aide d’un exemple initialement proposé par Jaffray et Jeleva (2007), que l’algorithme classique, procédant par programmation dynamique, n’est pas valide lorsque l’on cherche à déterminer une stratégie optimale au sens du critère de Hurwicz (pour  $\alpha \in ]0; 1[$ ). Considérons l’arbre de décision représenté sur la figure 5.1 et prenons la valeur 0.5 pour le paramètre  $\alpha$ . En  $D_2$  le décideur préfère la décision *Haut* à la décision *Bas* (pour *Haut* l’ensemble des conséquences potentielles est  $\{10, 20\}$ , ce qui donne une évaluation de 15 contre 12.5). Le sous-arbre issu de la décision *Bas* en  $D_2$  est donc supprimé. En  $D_1$  le décideur préfère la décision *Haut* qui offre une évaluation

de 10 contre 0. La stratégie retournée par programmation dynamique est donc  $\{D_1 = Haut, D_2 = Haut\}$ . Cette stratégie est sous-optimale puisque, comme nous pouvons le voir dans le tableau de la figure 5.1 page 137, la stratégie optimisant le critère d'Hurwicz pour  $\alpha = 0.5$  est  $\{D_1 = Haut, D_2 = Bas\}$ . L'approche classique par programmation dynamique est donc caduque puisqu'une sous-stratégie d'une stratégie optimale n'est pas nécessairement optimale. Dans l'optique de contourner cette difficulté algorithmique, nous proposons, par la suite, un algorithme de programmation dynamique biobjectif.

### 5.1.1 Algorithme de résolution fondé sur la programmation dynamique biobjectif

Dans cette partie nous nous intéressons à déterminer une stratégie optimale pour un décideur ayant des préférences qui suivent le critère de Hurwicz.

Pour toute stratégie  $\Delta$  (ou sous-stratégie), de l'arbre de décision  $T$ , menant à l'ensemble de conséquences  $\mathcal{C}_\Delta = \{u_1, u_2, \dots, u_n\} \subseteq \mathcal{C}_T$  telles que  $u_1 \leq u_2 \leq \dots \leq u_n$ , on définit le vecteur  $\langle \underline{u}; \bar{u} \rangle \in \mathcal{C}_T^2$  de telle sorte que  $\underline{u} = u_1$  et  $\bar{u} = u_n$ . L'évaluation d'une stratégie, par le critère de Hurwicz, peut alors se faire en évaluant ce vecteur comme suit (pour  $\alpha \in [0; 1]$ ) :

$$Hz(\langle \underline{u}; \bar{u} \rangle) = \alpha \underline{u} + (1 - \alpha) \bar{u}$$

Le critère de Hurwicz est bien entendu compatible avec la dominance de Pareto au sens large (notée  $\succsim_p$  dans la suite) sur l'ensemble des vecteurs de  $\mathcal{C}_T^2$ . Pour rappel, le vecteur  $V_1 = \langle \underline{u}_1; \bar{u}_1 \rangle$  Pareto-domine au sens large le vecteur  $V_2 = \langle \underline{u}_2; \bar{u}_2 \rangle$  si et seulement si  $\underline{u}_1 \geq \underline{u}_2$  et  $\bar{u}_1 \geq \bar{u}_2$ . En effet, pour tout  $(V_1, V_2) \in \mathcal{C}_T^2 \times \mathcal{C}_T^2$ , on a bien :

$$V_1 \succsim_p V_2 \Leftrightarrow \begin{cases} \underline{u}_1 \geq \underline{u}_2 \\ \bar{u}_1 \geq \bar{u}_2 \end{cases} \Rightarrow \forall \alpha \in [0; 1], \alpha \underline{u}_1 + (1 - \alpha) \bar{u}_1 \geq \alpha \underline{u}_2 + (1 - \alpha) \bar{u}_2$$

où  $V_1 = \langle \underline{u}_1; \bar{u}_1 \rangle$  et  $V_2 = \langle \underline{u}_2; \bar{u}_2 \rangle$ .

L'idée ici, est alors de procéder en deux phases :

1. construire l'ensemble  $\Omega$  des vecteurs non-dominés au sens de Pareto (si plusieurs vecteurs sont égaux un seul est conservé),
2. évaluer avec le critère de Hurwicz chaque vecteur de  $\Omega$  et retenir celui d'évaluation maximale.

Notons que, dans la pratique, pour chaque vecteur non-dominé de  $\Omega$ , on conserve la stratégie correspondante. Cependant, par soucis de simplicité et sans perte de généralité, nous ne travaillons ici que sur les vecteurs induits.

Nous allons maintenant voir comment il est possible de construire l'ensemble  $\Omega$  de manière efficace. Pour ce faire, nous procédons par programmation dynamique. En effet, la relation  $\succsim_p$  sur les vecteurs de  $\mathcal{C}_T^2$  vérifie le principe de monotonie (condition suffisante pour faire de la programmation dynamique, Morin (1982)) :

**Proposition 2** *Pour tout vecteur  $V_1 = \langle \underline{u}_1; \overline{u}_1 \rangle$ ,  $V_2 = \langle \underline{u}_2; \overline{u}_2 \rangle$  et  $V_3 = \langle \underline{u}_3; \overline{u}_3 \rangle$  on a :*

$$V_1 \succsim_p V_2 \Rightarrow V_1 \oplus V_3 \succsim_p V_2 \oplus V_3$$

où  $\oplus$  est l'opérateur défini par  $V_1 \oplus V_2 = \langle \min\{\underline{u}_1; \underline{u}_2\}; \max\{\overline{u}_1; \overline{u}_2\} \rangle$ .

*Preuve.* On vérifie :

$$V_1 \succsim_p V_2 \Leftrightarrow \begin{cases} \underline{u}_1 \geq \underline{u}_2 \\ \overline{u}_1 \geq \overline{u}_2 \end{cases} \Rightarrow \begin{cases} \min\{\underline{u}_1; \underline{u}_3\} \geq \min\{\underline{u}_2; \underline{u}_3\} \\ \max\{\overline{u}_1; \overline{u}_3\} \geq \max\{\overline{u}_2; \overline{u}_3\} \end{cases} \Leftrightarrow V_1 \oplus V_3 \succsim_p V_2 \oplus V_3$$

■

Puisque  $\succsim_p$  respecte le principe de monotonie, il est alors possible de construire  $\Omega$  par programmation dynamique. En chaque nœud  $N \in \mathcal{N}$  on calcule un ensemble  $\Omega_N$  de vecteurs non-dominés au sens de Pareto sur l'ensemble des sous-stratégies à partir de  $N$  (i.e., l'ensemble des stratégies dans le sous-arbre  $T(N)$ ). L'ensemble  $\Omega$  que l'on cherche à construire est donc l'ensemble construit en la racine de l'arbre de décision. Chacun de ces ensembles est construit par induction arrière dans l'arbre de décision. En d'autres termes, on construit l'ensemble  $\Omega_N$  à partir des ensembles construits en chaque fils de  $N$ . On note  $\succsim_{LEX}$  la relation d'ordre lexicographique sur les vecteurs de  $\mathcal{C}_T^2$  telle que  $\langle \underline{u}_1; \overline{u}_1 \rangle \succsim_{LEX} \langle \underline{u}_2; \overline{u}_2 \rangle$  si et seulement si ( $(\underline{u}_1 > \underline{u}_2)$  ou ( $(\underline{u}_1 = \underline{u}_2)$  et  $(\overline{u}_1 \geq \overline{u}_2)$ )). Remarquons qu'une famille de  $n$  vecteurs  $\langle a_i; b_i \rangle$  non dominés classés par ordre croissant lexicographique vérifie  $a_i < a_{i+1}$  et  $b_i > b_{i+1}$  (*propriété lexicographique*). Tous les ensembles  $\Omega_N$  que nous manipulerons par la suite seront supposés ordonnés lexicographiquement de manière croissante et vérifieront donc cette propriété. On notera  $\min_{LEX} \Omega_N$  le plus petit élément au sens lexicographique de l'ensemble  $\Omega_N$ .

On initialise l'induction arrière en construisant les ensembles  $\Omega_N$  en chaque nœud de conséquence : pour tout nœud de conséquence  $N \in \mathcal{C}$  muni de la conséquence  $c \in \mathcal{C}_T$ , on fixe  $\Omega_N = \{ \langle c; c \rangle \}$ . Les ensembles construits sont bien des ensembles de vecteurs non dominés et classés lexicographiquement puisqu'ils ne possèdent qu'un seul élément.

Dans la suite nous allons voir comment on construit, lors de la programmation dynamique, les ensembles  $\Omega_N$  lorsque  $N$  est un nœud de décision ou un nœud de chance.

**Construction de  $\Omega_N$  si  $N$  est un nœud de décision**

Si  $N \in \mathcal{N}_D$ , alors  $\Omega_N$  est l'ensemble des vecteurs non-dominés  $\mathcal{V}_N^D$  de l'ensemble défini par :

$$\mathcal{V}_N^D = \bigcup_{N' \in \text{fils}(N)} \Omega_{N'}$$

où  $\text{fils}(N)$  est l'ensemble des fils de  $N$ .

Afin de construire efficacement  $\Omega_N$ , il est nécessaire d'introduire une propriété clef. Notons que l'approche naïve consistant à comparer tous les vecteurs deux à deux est en :

$$O \left( \left( \sum_{N' \in \text{fils}(N)} |\Omega_{N'}| \right) \times \left( \sum_{N' \in \text{fils}(N)} |\Omega_{N'}| \right) \right)$$

**Proposition 3** Soit  $N_1$  et  $N_2$  deux ensembles de vecteurs dans  $\mathcal{C}_T^2$ , on pose  $\langle \underline{u}_1; \overline{u}_1 \rangle = \min_{LEX} \Omega_{N_1}$  et  $\langle \underline{u}_2; \overline{u}_2 \rangle = \min_{LEX} \Omega_{N_2}$  et on suppose que  $\langle \underline{u}_1; \overline{u}_1 \rangle \succsim_{LEX} \langle \underline{u}_2; \overline{u}_2 \rangle$ . Sous de telles conditions on vérifie la propriété suivante : si  $\langle \underline{u}_2; \overline{u}_2 \rangle$  n'est pas dominé par  $\langle \underline{u}_1; \overline{u}_1 \rangle$  alors aucun vecteur de  $N_1$  ne domine  $\langle \underline{u}_2; \overline{u}_2 \rangle$

*Preuve.* On a  $\underline{u}_2 \geq \underline{u}_1$  ou  $\overline{u}_2 \geq \overline{u}_1$ . De plus, grâce à la propriété lexicographique, on sait que pour tout  $\langle \underline{u}; \overline{u} \rangle \in N_1 \setminus \{\langle \underline{u}_1; \overline{u}_1 \rangle\}$  on a  $\underline{u}_1 < \underline{u}$  et  $\overline{u}_1 > \overline{u}$ . On déduit que  $\underline{u}_2 > \underline{u}$  ou  $\overline{u}_2 > \overline{u}$ . Le vecteur  $\langle \underline{u}_2; \overline{u}_2 \rangle$  n'est donc dominé par aucun vecteur de  $N_1$ .

On peut étendre facilement la démonstration au cas où il y a plus de deux ensembles. ■

Cette proposition nous permet d'introduire le lemme de complexité suivant :

**Lemme 1** Pour tout nœud de décision  $N$ , on peut construire l'ensemble  $\Omega_N$  à partir des ensembles  $\Omega_{N'}, N' \in \text{fils}(N)$  en :

$$O \left( \sum_{N' \in \text{fils}(N)} |\Omega_{N'}| \right)$$

*Preuve.* Pour chaque ensemble  $\Omega_{N'}$  on récupère le plus petit élément lexicographique. Plus formellement on construit l'ensemble :

$$\Omega_{min} = \bigcup_{N' \in \text{fils}(N)} \min_{LEX} \Omega_{N'}$$

Parmi ces éléments récupérés, on note  $V_{min} \in \min_{LEX} \Omega_{min}$  et  $M \in \text{fils}(N)$  tel que  $V_{min} \in \Omega_M$ . On compare  $V_{min}$  avec tous les vecteurs de l'ensemble  $\Omega_{min}$  privé de  $V_{min}$ . Si  $V_{min}$  est non dominé au sens de Pareto alors on peut l'ajouter à l'ensemble  $\Omega_N$ , en effet, il est alors le plus petit élément (au sens lexicographique) non-dominé de l'ensemble  $\mathcal{V}_N^D$  (voir la proposition 3). Il suffit ensuite de répéter l'opération jusqu'à que tous les ensembles  $\Omega_{N'}$  soient vides en prenant soin de supprimer  $V_{min}$  de l'ensemble  $\Omega_M$  avant

de réitérer les opérations. ■

Nous illustrons la construction de l'ensemble  $\Omega_N$  pour  $N \in \mathcal{N}_D$  au travers de l'exemple suivant.

**Exemple 20** Soit  $\Omega_{N_1} = \{\langle 1; 7 \rangle; \langle 2; 4 \rangle\}$ ,  $\Omega_{N_2} = \{\langle 2; 6 \rangle; \langle 3; 5 \rangle\}$  et  $\Omega_{N_3} = \{\langle 1; 6 \rangle; \langle 4; 4 \rangle\}$  avec  $\text{fils}(N) = \{N_1, N_2, N_3\}$ . Nous décrivons ci-dessous la suite des comparaisons effectuées pour construire l'ensemble  $\Omega_N$  lorsque  $N$  est un nœud de décision.

- $V_{min} = \langle 1; 6 \rangle$  est dominé dans l'ensemble  $\Omega_{min} = \{\langle 1; 7 \rangle; \langle 2; 6 \rangle; \langle 1; 6 \rangle\}$ , on le supprime de  $\Omega_{N_3}$ .
- $V_{min} = \langle 1; 7 \rangle$  n'est pas dominé dans l'ensemble  $\Omega_{min} = \{\langle 1; 7 \rangle; \langle 2; 6 \rangle; \langle 4; 4 \rangle\}$ , on le supprime de  $\Omega_{N_1}$  et on l'ajoute à  $\Omega_N$ .
- $V_{min} = \langle 2; 4 \rangle$  est dominé dans l'ensemble  $\Omega_{min} = \{\langle 2; 4 \rangle; \langle 2; 6 \rangle; \langle 4; 4 \rangle\}$ , on le supprime de  $\Omega_{N_1}$ .
- $V_{min} = \langle 2; 6 \rangle$  n'est pas dominé dans l'ensemble  $\Omega_{min} = \{\langle 2; 6 \rangle; \langle 4; 4 \rangle\}$ , on le supprime de  $\Omega_{N_2}$  et on l'ajoute à  $\Omega_N$ .
- $V_{min} = \langle 3; 5 \rangle$  n'est pas dominé dans l'ensemble  $\Omega_{min} = \{\langle 3; 5 \rangle; \langle 4; 4 \rangle\}$ , on le supprime de  $\Omega_{N_2}$  et on l'ajoute à  $\Omega_N$ .
- $V_{min} = \langle 4; 4 \rangle$  n'est pas dominé dans l'ensemble  $\Omega_{min} = \{\langle 4; 4 \rangle\}$ , on le supprime de  $\Omega_{N_3}$  et on l'ajoute à  $\Omega_N$ .
- $\Omega_{N_1}$ ,  $\Omega_{N_2}$  et  $\Omega_{N_3}$  sont vides on s'arrête.

On obtient donc l'ensemble  $\Omega_N = \{\langle 1; 7 \rangle; \langle 2; 6 \rangle; \langle 3; 5 \rangle; \langle 4; 4 \rangle\}$ .

### Construction de $\Omega_N$ si $N$ est un nœud de chance

Si  $N \in \mathcal{N}_C$ , alors  $\Omega_N$  est l'ensemble des vecteurs non-dominés de l'ensemble  $\mathcal{V}_N^C$  défini par :

$$\mathcal{V}_N^C = \{V_1 \oplus V_2 \oplus \dots \oplus V_k : (V_1, V_2, \dots, V_k) \in \Omega_{N_1} \times \Omega_{N_2} \times \dots \times \Omega_{N_k}\}$$

où  $N_1, N_2, \dots, N_k$  sont les fils de  $N$ .

Afin de construire efficacement  $\Omega_N$ , il est nécessaire d'introduire une propriété clef. Notons que l'approche naïve consistant à comparer tous les vecteurs deux à deux est en :

$$O\left(\left(\prod_{N' \in \text{fils}(N)} |\Omega_{N'}|\right) \times \left(\prod_{N' \in \text{fils}(N)} |\Omega_{N'}|\right)\right)$$

**Proposition 4** Soit  $N_1$  et  $N_2$  deux ensembles de vecteurs de  $\mathcal{C}_T^2$ , on pose  $\langle \underline{u}_1; \overline{u}_1 \rangle = \min_{LEX} \Omega_{N_1}$  et  $\langle \underline{u}_2; \overline{u}_2 \rangle = \min_{LEX} \Omega_{N_2}$  et on suppose que  $\langle \underline{u}_1; \overline{u}_1 \rangle \succsim_{LEX} \langle \underline{u}_2; \overline{u}_2 \rangle$ . Sous de telles conditions on vérifie la propriété suivante :

$$\langle \underline{u}_2; \overline{u}_2 \rangle \oplus \langle \underline{u}_1; \overline{u}_1 \rangle \succsim_p \langle \underline{u}_2; \overline{u}_2 \rangle \oplus \langle \underline{u}; \overline{u} \rangle \text{ pour tout vecteur } \langle \underline{u}; \overline{u} \rangle \in \Omega_{N_1} \setminus \{\langle \underline{u}_1; \overline{u}_1 \rangle\}$$

*Preuve.* On a  $\underline{u}_2 < \underline{u}_1$  ou ( $\underline{u}_2 = \underline{u}_1$  et  $\overline{u}_2 \leq \overline{u}_1$ ). De plus, grâce à la propriété lexicographique, on sait que pour tout  $\langle \underline{u}; \overline{u} \rangle \in N_1 \setminus \{\langle \underline{u}_1; \overline{u}_1 \rangle\}$ ,  $\underline{u}_1 < \underline{u}$  et  $\overline{u}_1 > \overline{u}$ . On en déduit que  $\max\{\overline{u}_1; \overline{u}_2\} \geq \max\{\overline{u}; \overline{u}_2\}$  et  $\min\{\underline{u}_1; \underline{u}_2\} = \min\{\underline{u}; \underline{u}_2\} = \underline{u}_2$ , autrement dit  $\langle \underline{u}_2; \overline{u}_2 \rangle \oplus \langle \underline{u}_1; \overline{u}_1 \rangle \succ_p \langle \underline{u}_2; \overline{u}_2 \rangle \oplus \langle \underline{u}; \overline{u} \rangle$ .

On peut étendre facilement la démonstration au cas où il y a plus de deux ensembles. ■

Cette proposition nous permet d'introduire le lemme de complexité suivant :

**Lemme 2** *Pour tout nœud de chance  $N$ , on peut construire l'ensemble  $\Omega_N$  à partir des ensembles  $\Omega_{N'}, N' \in \text{fils}(N)$  en :*

$$O \left( \sum_{N' \in \text{fils}(N)} |\Omega_{N'}| \right)$$

*Preuve.* Pour chaque ensemble  $\Omega_{N'}$  on récupère le plus petit élément lexicographique. Plus formellement on construit l'ensemble :

$$\Omega_{min} = \bigcup_{N' \in \text{fils}(N)} \min_{LEX} \Omega_{N'}$$

Parmi ces éléments récupérés, on note  $V_{min} \in \min_{LEX} \Omega_{min}$  et  $M \in \text{fils}(N)$  tel que  $V_{min} \in \Omega_M$ . On construit le vecteur  $V$  en appliquant l'opérateur  $\oplus$  sur l'ensemble des plus petits vecteurs de chaque ensemble (i.e.  $V = V_1 \oplus V_2 \oplus \dots \oplus V_k$  tels que  $\bigcup_{i=1}^k V_i = \Omega_{min}$ ) et on ajoute  $V$  à  $\Omega_N$ . On supprime  $V_{min}$  de l'ensemble  $\Omega_M$  car on sait que toutes agrégation avec l'opérateur  $\oplus$  de l'élément  $V_{min}$  fournira alors un vecteur dominé par  $V$  (voir la proposition 4). Enfin on répète ce processus jusqu'à qu'au moins un des ensembles  $\Omega_N$  soit vide. Pour finir il suffit de supprimer tous les éléments dominés de  $\Omega_N$  (lorsque les éléments sont triés dans l'ordre lexicographique, ce qui est le cas ici, cette étape se fait linéairement). ■

Nous illustrons la construction de l'ensemble  $\Omega_N$  lorsque  $N \in \mathcal{C}$  au travers de l'exemple suivant.

**Exemple 21** *Soit  $\Omega_{N_1} = \{\langle 1; 7 \rangle; \langle 2; 4 \rangle\}$ ,  $\Omega_{N_2} = \{\langle 2; 6 \rangle; \langle 3; 5 \rangle\}$  et  $\Omega_{N_3} = \{\langle 1; 6 \rangle; \langle 4; 4 \rangle\}$  avec  $N_1, N_2, N_3 \in \text{fils}(N)$ . Nous décrivons ci-dessous la suite des opérations effectuées pour construire l'ensemble  $\Omega_N$  lorsque  $N$  est un nœud de chance.*

- $V = \langle 1; 7 \rangle \oplus \langle 2; 6 \rangle \oplus \langle 1; 6 \rangle = \langle 1; 7 \rangle$  ; on l'ajoute à  $\Omega_N$  et on supprime  $V_{min} = \langle 1; 6 \rangle$  de  $\Omega_{N_3}$ .
- $V = \langle 1; 7 \rangle \oplus \langle 2; 6 \rangle \oplus \langle 4; 4 \rangle = \langle 1; 7 \rangle$  ; on l'ajoute à  $\Omega_N$  et on supprime  $V_{min} = \langle 1; 7 \rangle$  de  $\Omega_{N_1}$ .
- $V = \langle 2; 4 \rangle \oplus \langle 2; 6 \rangle \oplus \langle 4; 4 \rangle = \langle 2; 6 \rangle$  ; on l'ajoute à  $\Omega_N$  et on supprime  $V_{min} = \langle 2; 4 \rangle$  de  $\Omega_{N_1}$ .

—  $\Omega_{N_1}$  est vide on s'arrête.

On obtient alors l'ensemble  $\Omega_N = \{\langle 1; 7 \rangle; \langle 1; 7 \rangle; \langle 2; 6 \rangle\}$ . Après suppression des éléments dominés on déduit  $\Omega_N = \{\langle 1; 7 \rangle; \langle 2; 6 \rangle\}$ .

Une fois la programmation dynamique achevée, il suffit alors d'évaluer tous les vecteurs de l'ensemble construit en la racine avec le critère de Hurwicz.

En guise d'illustration nous appliquons l'algorithme sur l'arbre de la figure 5.1 page 137. Les ensembles construits sont successivement :

- $\Omega_{L_2} = \{\langle 10; 20 \rangle\}$
- $\Omega_{L_3} = \{\langle 0; 25 \rangle\}$
- $\Omega_{D_2} = \{\langle 10; 20 \rangle; \langle 0; 25 \rangle\}$
- $\Omega_{L_1} = \{\langle 0; 25 \rangle\}$
- $\Omega_{D_1} = \{\langle 0; 25 \rangle\}$

La stratégie optimale au sens de Hurwicz pour tout  $\alpha$  est donc la stratégie caractérisée par le vecteur  $\langle 0; 25 \rangle$ , ce qui correspond à la stratégie  $\{D_1 = Haut; D_2 = Bas\}$ .

Enfin, nous sommes maintenant en mesure d'énoncer le résultat de complexité suivant :

**Proposition 5** *Le calcul d'une stratégie optimisant le critère de Hurwicz peut se réaliser via un algorithme en  $O(n^2)$  si  $n$  est les nombre de nœuds dans l'arbre de décision.*

*Preuve.* Pour prouver ce théorème, remarquons que  $|\Omega| \leq |\mathcal{C}_T|$ . En effet, pour chaque valeur de la première composante vectorielle il ne peut y avoir qu'un seul non dominé au plus. Ainsi le nombre de conséquences d'un arbre de décision étant en  $O(n)$ , le nombre maximum de vecteurs non dominés est alors en  $O(n)$  également. Un ensemble de vecteurs de non dominés est construit en chaque nœud de l'arbre de décision avec des méthodes en :

$$O\left(\sum_{N' \in \text{fils}(N)} |\Omega_{N'}|\right)$$

(d'après le lemme 1 et le lemme 2). D'autre part, on a :

$$\sum_{N \in \mathcal{N} \setminus \mathcal{N}_U} \sum_{N' \in \text{fils}(N)} |\Omega_{N'}| \leq \sum_{N \in \mathcal{N}} |\Omega_N|$$

On en déduit une complexité globale de :

$$O\left(\sum_{N \in \mathcal{N}} |\Omega_N|\right) = O(n^2)$$

puisque  $|\Omega_N|$  est en  $O(n)$ . ■

Dans la section suivante nous illustrons les performances de l'algorithme proposé au travers d'expérimentations numériques.

Prof. (nœuds)		Moyenne	Maximum
4 (31)	Cardinal ( $ \Omega $ )	2	5
	Temps (sec.)	0	0
6 (127)	Cardinal ( $ \Omega $ )	4	6
	Temps (sec.)	0	0
8 (511)	Cardinal ( $ \Omega $ )	7	14
	Temps (sec.)	0	0
10 (2047)	Cardinal ( $ \Omega $ )	13	22
	Temps (sec.)	0	0
12 (8191)	Cardinal ( $ \Omega $ )	24	39
	Temps (sec.)	0	0
14 (32767)	Cardinal ( $ \Omega $ )	47	68
	Temps (sec.)	0.02	1
16 (131071)	Cardinal ( $ \Omega $ )	90	115
	Temps (sec.)	0.14	1
18 (524287)	Cardinal ( $ \Omega $ )	174	216
	Temps (sec.)	0.58	1
20 (2097151)	Cardinal ( $ \Omega $ )	348	570
	Temps (sec.)	2.09	3
22 (8388607)	Cardinal ( $ \Omega $ )	714	1164
	Temps (sec.)	8.31	9

TABLE 5.1 – Temps d'exécution CPU (en sec.) et cardinal des ensembles  $\Omega_N$  construits lors de la détermination de la stratégie qui optimise le critère de Hurwicz dans un arbre de décision sous incertitude totale.

### 5.1.2 Expérimentations numériques

L'algorithme proposé dans la partie précédente a été implémenté avec le langage C++. Les expérimentations numériques ont été réalisées sur un ordinateur doté d'un processeur Pentium IV CPU 2.13GHz et de 3.5Go de RAM.

Nous présentons dans cette partie les résultats obtenus sur des arbres de décision générés aléatoirement. Pour ce faire, nous avons imposé une structure d'arbre binaire complet pour les arbres de décision. Autrement dit, pour un arbre de profondeur  $p$ , les nœuds de profondeur strictement inférieure à  $p$  ont exactement deux fils et les nœuds de profondeur  $p$  n'ont aucun fils (ce sont les feuilles de l'arbre).

De plus, nous avons imposé que la profondeur  $p$  de l'arbre soit toujours paire, que les nœuds de profondeur paire strictement inférieure à  $p$  soient des nœuds de décision, que les nœuds de profondeur impaire soient des nœuds de chance et que les nœuds de profondeur  $p$  soit des nœuds d'utilité. Les valeurs d'utilités aux feuilles sont des nombres réels générés aléatoirement dans l'intervalle  $[1; 1000]$ .

Le tableau 5.1 donne pour chaque profondeur d'arbre (500 instances ont été géné-

rées pour chaque profondeur) le cardinal moyen et le cardinal maximal des ensembles construits en chaque nœud de l'arbre. Nous donnons également le temps moyen d'exécution de l'algorithme (en secondes) ainsi que le temps maximum observé pour au moins une instance. Comme le montre le tableau 5.1 page 144, nous sommes capables de traiter des instances de profondeur 22 (plus de 8 000 000 nœuds). Si nous ne sommes pas allés plus loin c'est essentiellement dû au fait que la mémoire requise pour stocker les arbres de décision de plus grande profondeur devient trop grande. Les temps de résolution restent en effet toujours inférieurs à 10 secondes.

## 5.2 Incertitude représentée par des probabilités intervalles

Dans cette partie nous nous efforçons à appliquer le critère de Hurwicz dans un arbre de décision lorsque l'incertitude de chaque événement  $e$ , en chaque nœud de chance, est représentée par une probabilité intervalle  $K(e) = [\underline{K}(e), \overline{K}(e)]$  où  $\underline{K}(e)$  est la probabilité inférieure de l'événement  $e$  et  $\overline{K}(e)$  la probabilité supérieure de l'événement  $e$ .

Notons que nous supposons ici que la distribution  $K$  définit toujours un ensemble de distributions de probabilité  $\mathcal{P}_K$  non vide (i.e., il existe toujours au moins une distribution de probabilité compatible avec les probabilités intervalles du problème).

En guise d'exemple, considérons l'arbre de décision  $T$  représenté sur la figure 5.2 page 146 reprenant un exemple célèbre de forage pétrolier, introduit initialement par Raiffa (1968). La sémantique de cet arbre de décision est détaillée dans l'exemple 22.

**Exemple 22** *Une compagnie pétrolière doit décider de forer ou non un site donné. Dans ce but, elle doit d'abord décider si elle sonde ou non la structure géologique du site (décision  $D_1$ ), ce qui coûte 10 000 Euros et donne une meilleure estimation de la quantité de pétrole qui s'y trouve. Le résultat du sondage peut être vu comme une variable aléatoire  $X_1$  qui peut prendre trois valeurs possibles : aucune si la sonde ne trouve aucune trace de pétrole, faible si la sonde trouve une faible trace de pétrole, ou forte si la sonde trouve une forte trace de pétrole. Ensuite, la compagnie doit décider si elle réalise un forage ou non (décision  $D_2$ ), ce qui coûte 70 000 Euros. Finalement, si la compagnie décide de forer, le résultat du forage peut être vu comme une variable aléatoire  $X_2$  (associée aux nœuds  $X_2^i$  pour  $i = 1..4$ ) qui peut prendre trois valeurs possibles : le puits est sec (le revenu est nul), humide (120 000 Euros) ou imbibé (270 000 Euros). Ce problème peut être représenté par l'arbre de décision sur la gauche de la figure 5.2 page 146. Remarquons que la décision  $D_2$  est dupliquée en plusieurs nœuds (nœuds  $D_2^1, D_2^2, D_2^3$  et  $D_2^4$ ) car elle peut être prise dans différents contextes (un sondage a été réalisé ou non, le résultat du sondage est encourageant ou non...).*

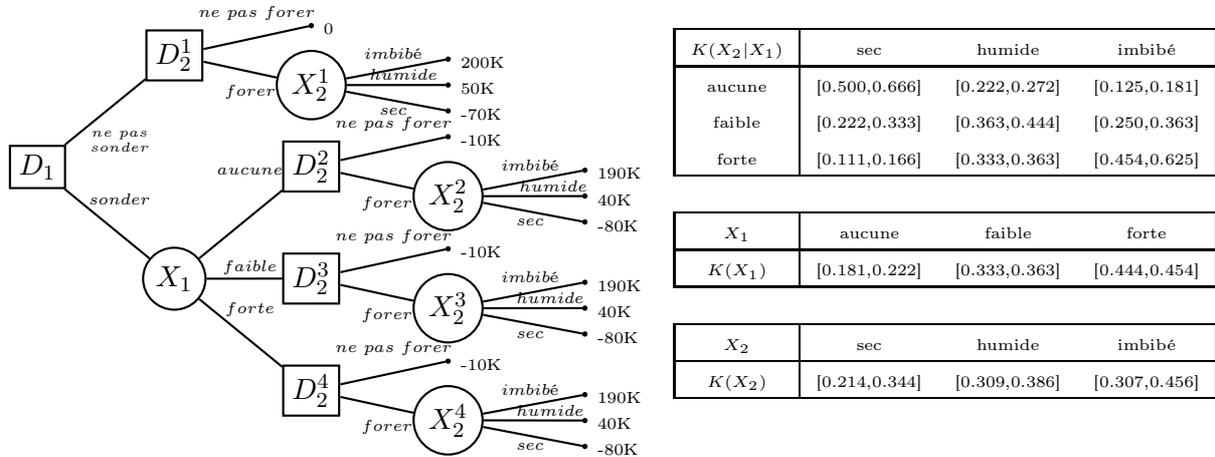


FIGURE 5.2 – Arbre de décision pour le problème du forage pétrolier.

Dans la suite, nous allons distinguer les arbres de décision dits *séparables* des arbres de décision dits *non-séparables*.

Avant cela, nous introduisons la notation suivante. Soit  $E_N = \{e_1, e_2, \dots, e_n\}$  l'ensemble des événements issus du nœud de chance  $N$ . On note  $\mathcal{P}_N$  l'ensemble des distributions de probabilité compatibles avec les probabilités intervalles définies sur  $E_N$ . Plus formellement, on a :

$$\mathcal{P}_N = \left\{ P \in \mathbb{P}_{E_N} : \sum_{e \in E_N} P(e) = 1 \text{ et } \forall e \in E_N, P(e) \in K(e) \right\}$$

où  $\mathbb{P}_{E_N}$  est l'ensemble de toutes les distributions de probabilité définies sur  $E_N$ .

De manière similaire, étant donnée une loterie  $L = (x_1, e_1; \dots; x_n, e_n)$ , on note  $E_L = \{e_1, e_2, \dots, e_n\}$  l'ensemble des événements de la loterie  $L$  et  $\mathcal{P}_L$  l'ensemble des distributions de probabilité compatibles avec les probabilités intervalles  $K(e_i)$ . Plus formellement, on a :

$$\mathcal{P}_L = \left\{ P \in \mathbb{P}_{E_L} : \sum_{e \in E_L} P(e) = 1 \text{ et } \forall e \in E_L, P(e) \in K(e) \right\}$$

On a donc :

$$\underline{EU}(L) = \min_{P \in \mathcal{P}_L} EU_P(L)$$

$$\overline{EU}(L) = \max_{P \in \mathcal{P}_L} EU_P(L)$$

où  $EU_P(L)$  désigne l'espérance de la loterie  $L$  avec la distribution de probabilité  $P$ .

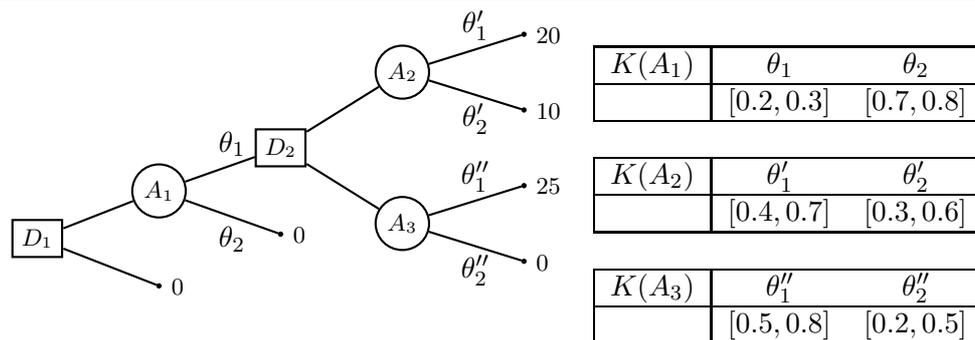


FIGURE 5.3 – Exemple d'arbre de décision dit séparable.

### Arbre de décision séparable

On dit qu'un arbre de décision est séparable dès lors que :

$$\mathcal{P}_K = \prod_{N \in \mathcal{N}_C} \mathcal{P}_N$$

En d'autres termes, une distribution de probabilité est cohérente dès lors qu'elle somme à 1 en chaque nœud de chance et qu'elle vérifie les probabilités intervalles définies aux nœuds de chance. Par exemple, si les variables  $A_1$ ,  $A_2$ , et  $A_3$  sont mutuellement indépendantes, alors l'arbre de décision représenté sur la figure 5.3 est un arbre de décision dit séparable, et une distribution de probabilité cohérente est par exemple :  $P(A_1 = \theta_1) = 0.2 = 1 - P(A_1 = \theta_2)$ ,  $P(A_2 = \theta'_1) = 0.7 = 1 - P(A_2 = \theta'_2)$  et  $P(A_3 = \theta''_1) = 0.6 = 1 - P(A_3 = \theta''_2)$ .

### Arbre de décision non-séparable

On dit qu'un arbre de décision est non-séparable dès lors que :

$$\mathcal{P}_K \subset \prod_{N \in \mathcal{N}_C} \mathcal{P}_N$$

En d'autres termes, pour qu'une distribution de probabilité appartienne à  $\mathcal{P}_K$  il ne suffit pas qu'elle somme à 1 en chaque nœud de chance. Par exemple, considérons l'arbre de décision représenté sur la figure 5.2 page 146 et la distribution partielle de probabilité suivante :  $P(X_2 = \text{sec} | X_1 = \text{aucune}) = 0.55$ ,  $P(X_2 = \text{sec} | X_1 = \text{faible}) = 0.33$ ,  $P(X_2 = \text{sec} | X_1 = \text{forte}) = 0.12$ ,  $P(X_1 = \text{aucune}) = 0.20$ ,  $P(X_1 = \text{faible}) = 0.35$ ,  $P(X_1 = \text{forte}) = 0.45$  et  $P(X_2 = \text{sec}) = 0.22$ . Quelle que soit la manière dont on complètera cette distribution partielle de probabilité, elle ne peut appartenir à  $\mathcal{P}_K$  (i.e., elle ne peut être globalement cohérente) puisque le théorème des probabilités totales n'est pas respecté :  $P(X_2 = \text{sec} | X_1 = \text{aucune})P(X_1 = \text{aucune}) + P(X_2 = \text{sec} | X_1 = \text{faible})P(X_1 = \text{faible}) + P(X_2 = \text{sec} | X_1 = \text{forte})P(X_1 = \text{forte}) = 0.2795 \neq 0.22 = P(X_2 = \text{sec})$ .

### 5.2.1 Optimisation du critère de Hurwicz dans un arbre de décision séparable

Dans cette partie nous nous intéressons à optimiser le critère de Hurwicz dans un arbre de décision séparable. Notons que le critère de Hurwicz ne respecte pas l'axiome conséquentialiste, et cela même lorsque l'arbre de décision est séparable. En effet, pour s'en convaincre, il suffit de remarquer que le cadre de l'incertitude totale est un cas particulier de risque imprécis pour lequel chaque probabilité intervalle est  $[0, 1]$ . Comme nous l'avons vu précédemment (voir la section 5.1 page 136), lorsque l'incertitude est totale, le critère de Hurwicz ne respecte pas l'axiome conséquentialiste, et donc il en est de même lorsque le risque est imprécis. Puisque le critère de Hurwicz ne respecte pas l'axiome conséquentialiste, nous proposons ici de faire du choix résolu depuis la racine (i.e., on sélectionne une des stratégies optimisant le critère de Hurwicz depuis la racine, et on n'en dévie pas lors de son application).

Dans un premier temps, nous allons voir comment déterminer une stratégie optimisant les critères  $\Gamma$ -Maximin et  $\Gamma$ -Maximax qui sont des cas particuliers du critère de Hurwicz (pour  $\alpha = 1$  et  $\alpha = 0$ ). Dans un arbre de décision séparable, l'axiome d'indépendance est respecté pour ces deux critères :

**Proposition 6** *Soit  $L_1$ ,  $L_2$  et  $L_3$  trois loteries incertaines et  $\lambda$  un réel compris entre 0 et 1. Si l'ensemble  $\mathcal{P}_{\lambda L_1 + (1-\lambda)L_3}$  (resp.  $\mathcal{P}_{\lambda L_2 + (1-\lambda)L_3}$ ) des distributions de probabilité compatibles avec la loterie composée  $\lambda L_1 + (1-\lambda)L_3$  (resp.  $\lambda L_2 + (1-\lambda)L_3$ ) est le produit cartésien de  $\mathcal{P}_{L_1}$  (resp.  $\mathcal{P}_{L_2}$ ) et  $\mathcal{P}_{L_3}$  (condition de séparabilité) alors on a les propriétés suivantes :*

$$\underline{EU}(L_1) \geq \underline{EU}(L_2) \Rightarrow \underline{EU}(\lambda L_1 + (1-\lambda)L_3) \geq \underline{EU}(\lambda L_2 + (1-\lambda)L_3)$$

$$\overline{EU}(L_1) \geq \overline{EU}(L_2) \Rightarrow \overline{EU}(\lambda L_1 + (1-\lambda)L_3) \geq \overline{EU}(\lambda L_2 + (1-\lambda)L_3)$$

*Preuve.* Montrons que si la condition de séparabilité est vérifiée on a :

$$\underline{EU}(\lambda L_1 + (1-\lambda)L_3) = \lambda \underline{EU}(L_1) + (1-\lambda) \underline{EU}(L_3).$$

Nous avons par définition :

$$\underline{EU}(\lambda L_1 + (1-\lambda)L_3) = \min_{P \in \mathcal{P}_{\lambda L_1 + (1-\lambda)L_3}} EU_P(\lambda L_1 + (1-\lambda)L_3).$$

La linéarité de l'espérance nous donne :

$$\min_{P \in \mathcal{P}_{\lambda L_1 + (1-\lambda)L_3}} EU_P(\lambda L_1 + (1-\lambda)L_3) = \min_{P \in \mathcal{P}_{\lambda L_1 + (1-\lambda)L_3}} \lambda EU_P(L_1) + (1-\lambda) EU_P(L_3).$$

D'après la condition de séparabilité, on a donc :

$$\min_{P \in \mathcal{P}_{\lambda L_1 + (1-\lambda)L_3}} \lambda EU_P(L_1) + (1-\lambda)EU_P(L_3) = \min_{\substack{P_1 \in \mathcal{P}_{L_1} \\ P_3 \in \mathcal{P}_{L_3}}} \lambda EU_{P_1}(L_1) + (1-\lambda)EU_{P_3}(L_3).$$

On en conclut :

$$\begin{aligned} \min_{\substack{P_1 \in \mathcal{P}_{L_1} \\ P_3 \in \mathcal{P}_{L_3}}} \lambda EU_{P_1}(L_1) + (1-\lambda)EU_{P_3}(L_3) &= \lambda \min_{P_1 \in \mathcal{P}_{L_1}} EU_{P_1}(L_1) + (1-\lambda) \min_{P_3 \in \mathcal{P}_{L_3}} EU_{P_3}(L_3) \\ &= \lambda \underline{EU}(L_1) + (1-\lambda) \underline{EU}(L_3). \end{aligned}$$

Par un raisonnement identique on obtient :

$$\underline{EU}(\lambda L_2 + (1-\lambda)L_3) = \lambda \underline{EU}(L_2) + (1-\lambda) \underline{EU}(L_3).$$

Ce qui nous donne de manière immédiate :

$$\underline{EU}(L_1) \geq \underline{EU}(L_2) \Rightarrow \underline{EU}(\lambda L_1 + (1-\lambda)L_3) \geq \underline{EU}(\lambda L_2 + (1-\lambda)L_3).$$

Un raisonnement analogue nous permet également de montrer que :

$$\overline{EU}(L_1) \geq \overline{EU}(L_2) \Rightarrow \overline{EU}(\lambda L_1 + (1-\lambda)L_3) \geq \overline{EU}(\lambda L_2 + (1-\lambda)L_3).$$

■

Dans un arbre de décision séparable, la proposition 6 est valable en chaque nœud de chance. Il est donc possible de retourner une stratégie qui optimise le critère  $\Gamma$ -Maximin (ou le critère  $\Gamma$ -Maximax) en procédant par programmation dynamique dans l'arbre de décision.

Comme nous l'avons vu précédemment, une approche par programmation dynamique ne permet cependant pas de déterminer une stratégie optimisant le critère de Hurwicz dès lors que  $\alpha \in ]0, 1[$ . Néanmoins, il est possible d'exploiter la propriété suivante : si une sous-stratégie est dominée sur le critère  $\Gamma$ -Maximax et le critère  $\Gamma$ -Maximin (i.e., possède une évaluation moins élevée au sens des deux critères qu'une autre sous-stratégie) alors il n'existe pas de stratégie optimale au sens de Hurwicz qui étend cette sous-stratégie. En effet,  $\forall (L_1, L_2, L_3) \in \mathbb{L}_X$ , si :

$$\langle \underline{EU}(L_1), \overline{EU}(L_1) \rangle \succ_p \langle \underline{EU}(L_2), \overline{EU}(L_2) \rangle$$

alors on a (grâce à la propriété 6) :

$$\langle \underline{EU}(\lambda L_1 + (1-\lambda)L_3), \overline{EU}(\lambda L_1 + (1-\lambda)L_3) \rangle \succ_p \langle \underline{EU}(\lambda L_2 + (1-\lambda)L_3), \overline{EU}(\lambda L_2 + (1-\lambda)L_3) \rangle$$

où  $\lambda \in [0, 1]$ . Puisque le critère de Hurwicz est compatible<sup>2</sup> avec la dominance de Pareto sur les vecteurs de composantes  $\Gamma$ -Maximin et  $\Gamma$ -Maximax, nous proposons ici de construire l'ensemble des vecteurs non dominés au sens de Pareto. Dans cet ensemble, il existe alors au moins un vecteur associé à une stratégie optimisant le critère de Hurwicz.

Dans l'optique de construire cet ensemble, nous proposons une approche par programmation dynamique biobjectif. Pour ce faire, en chaque nœud  $N \in \mathcal{N}$ , on construit l'ensemble  $\Omega_N$  des vecteurs non dominés qui sont associés aux stratégies du sous-arbre  $T(N)$ .

On initialise la programmation dynamique en chaque nœud d'utilité  $N \in \mathcal{C}$  avec l'ensemble  $\Omega_N = \{\langle u(N), u(N) \rangle\}$ , où  $u(N)$  est la valeur de l'utilité associée au nœud  $N$ . En chaque nœud de décision  $N \in \mathcal{N}_D$ , on calcule l'ensemble  $\Omega'_N$  comme suit :

$$\Omega'_N = \bigcup_{N' \in \text{fils}(N)} \Omega_{N'}$$

Le sous-ensemble des vecteurs non-dominés de  $\Omega'_N$ , constitue l'ensemble  $\Omega_N$ .

En chaque nœud de chance  $N \in \mathcal{N}_C$ , on calcule l'ensemble  $\Omega'_N$  comme suit :

$$\left\{ \left\langle \min_{P \in \mathcal{P}_N} \left\{ \sum_{i=1}^n P(e_i) \underline{EU}_i \right\}, \max_{P \in \mathcal{P}_N} \left\{ \sum_{i=1}^n P(e_i) \overline{EU}_i \right\} \right\rangle \right\}$$

pour tout  $(\langle \underline{EU}_1, \overline{EU}_1 \rangle, \dots, \langle \underline{EU}_n, \overline{EU}_n \rangle) \in \Omega_{N_1} \times \dots \times \Omega_{N_n}$  où  $|\text{fils}(N)| = n$ ,  $\{e_1, \dots, e_n\}$  est l'ensemble des événements issus du nœud  $N$  et pour tout  $i \in \{1, \dots, n\}$ ,  $N_i$  est le fils de  $N$  issu de l'événement  $e_i$ . Le sous-ensemble des vecteurs non-dominés de  $\Omega'_N$ , constitue l'ensemble  $\Omega_N$ . Notons que pour déterminer la distribution de probabilité  $P \in \mathcal{P}_N$  qui minimise la somme  $\sum_{i=1}^n P(e_i) \underline{EU}_i$ , on peut utiliser le résultat suivant :

**Proposition 7** *Considérons la loterie  $L = (x_1, e_1; x_2, e_2, \dots, x_n, e_n)$  tel que  $x_1 \leq x_2 \leq \dots \leq x_n$ . La distribution de probabilité  $P \in \mathcal{P}_L$  telle que  $EU_P(L) = \underline{EU}(L)$  est définie récursivement par :*

$$\begin{cases} P(e_1) = \min \left\{ 1 - \sum_{j=2}^n \underline{K}(e_j), \overline{K}(e_1) \right\} \\ P(e_i) = \min \left\{ 1 - \sum_{j=1}^{i-1} P(e_j) - \sum_{j=i+1}^n \underline{K}(e_j), \overline{K}(e_i) \right\} \quad \forall i \in \{2, \dots, n\} \end{cases}$$

*Preuve.* Montrons que  $\forall P' \in \mathcal{P}_L, P' \neq P$  on a :

$$EU_P(L) \leq EU_{P'}(L)$$

2. En effet,  $\forall (\Delta_1, \Delta_2) \in \overline{\Delta}^T \times \overline{\Delta}^T$ ,  $(\Gamma_{\min}(\Delta_1), \Gamma_{\max}(\Delta_1)) \succ_p (\Gamma_{\min}(\Delta_2), \Gamma_{\max}(\Delta_2)) \Rightarrow Hz(\Delta_1) \geq Hz(\Delta_2)$ .

On note  $i_0$  l'indice tel que  $P'(e_i) = P(e_i)$  pour  $i < i_0$  et  $P'(e_{i_0}) < P(e_{i_0})$  ( $P'(e_{i_0}) > P(e_{i_0})$  étant un cas impossible). On a alors :

$$\sum_{i=i_0}^n P'(e_i) = 1 - \sum_{i=1}^{i_0-1} P(e_i)$$

Par conséquent, on a :

$$P'(e_{i_0}) < P(e_{i_0}) \Rightarrow \exists i > i_0, P'(e_i) > \underline{K}(e_i)$$

Soit  $i_1 = \min\{i : i > i_0 \text{ et } P'(e_i) > \underline{K}(e_i)\}$  et  $\varepsilon = \min\{P(e_{i_0}) - P'(e_{i_0}), P'(e_{i_1}) - \underline{K}(e_{i_1})\} > 0$ . On note  $P_1$  la distribution de probabilité définie par  $P_1(e_{i_0}) = P'(e_{i_0}) + \varepsilon$ ,  $P_1(e_{i_1}) = P'(e_{i_1}) - \varepsilon$  et  $P_1(e_i) = P'(e_i)$  pour tout  $i \neq i_0, i_1$ . Nous avons alors  $EU_{P_1}(L) \leq EU_{P'}(L)$  puisque  $EU_{P_1}(L) - EU_{P'}(L) = \varepsilon(x_{i_0} - x_{i_1}) \leq 0$ .

Si  $P_1 \neq P$ , par un raisonnement similaire, on peut construire la distribution de probabilité  $P_2$  telle que  $EU_{P_2}(L) \leq EU_{P_1}(L)$ . On génère ainsi une suite  $P_1, \dots, P_k$  de distributions de probabilité telles que  $EU_{P_{i+1}}(L) \leq EU_{P_i}(L)$  et  $P_k = P$ . On déduit alors que  $EU_P(L) \leq EU_{P'}(L)$ . ■

Notons que pour déterminer la distribution de probabilité  $P \in \mathcal{P}_N$  qui maximise l'expression  $\sum_{i=1}^n P(e_i) \overline{EU}_i$ , on procède de manière symétrique.

L'exemple 23 déroule la programmation dynamique biobjectif proposée ici sur l'arbre de décision représenté sur la figure 5.3 page 147.

**Exemple 23** *Considérons l'arbre de décision représenté sur la figure 5.3 page 147. Nous prenons la fonction identité pour la fonction d'utilité dans cet exemple. En chaque nœud de l'arbre, on calcule les ensembles de vecteurs comme suit :*

- En  $A_2$  : l'événement  $\theta'_1$  conduit à l'ensemble  $\{\langle 20, 20 \rangle\}$  et l'événement  $\theta'_2$  conduit à l'ensemble  $\{\langle 10, 10 \rangle\}$ . On en déduit alors :

$$\begin{aligned} \Omega'_{A_2} &= \left\{ \left\langle \min_{P \in \mathcal{P}_{A_2}} \{20 \times P(\theta'_1) + 10 \times P(\theta'_2)\}, \max_{P \in \mathcal{P}_{A_2}} \{20 \times P(\theta'_1) + 10 \times P(\theta'_2)\} \right\rangle \right\} \\ &= \{\langle 20 \times 0.4 + 10 \times 0.6, 20 \times 0.7 + 10 \times 0.3 \rangle\} \\ &= \{\langle 14, 17 \rangle\} \end{aligned}$$

L'ensemble des vecteurs non dominés de  $\Omega'_{A_2}$  constitue l'ensemble  $\Omega_{A_2}$ . On en déduit que  $\Omega_{A_2} = \{\langle 14, 17 \rangle\}$ .

- En  $A_3$  : l'événement  $\theta''_1$  conduit à l'ensemble  $\{\langle 25, 25 \rangle\}$  et l'événement  $\theta''_2$  conduit à l'ensemble  $\{\langle 0, 0 \rangle\}$ . On en déduit alors :

$$\begin{aligned} \Omega'_{A_3} &= \left\{ \left\langle \min_{P \in \mathcal{P}_{A_3}} \{25 \times P(\theta''_1) + 0 \times P(\theta''_2)\}, \max_{P \in \mathcal{P}_{A_3}} \{25 \times P(\theta''_1) + 0 \times P(\theta''_2)\} \right\rangle \right\} \\ &= \{\langle 25 \times 0.5, 25 \times 0.8 \rangle\} \\ &= \{\langle 12.5, 20 \rangle\} \end{aligned}$$

L'ensemble des vecteurs non dominés de  $\Omega'_{A_3}$  constitue l'ensemble  $\Omega_{A_3}$ . On en déduit que  $\Omega_{A_3} = \{\langle 12.5, 20 \rangle\}$ .

- En  $D_2$  : la décision "haut" conduit à l'ensemble  $\Omega_{A_2} = \{\langle 14, 17 \rangle\}$  et la décision "bas" conduit à l'ensemble  $\Omega_{A_3} = \{\langle 12.5, 20 \rangle\}$ . On en déduit alors :

$$\begin{aligned}\Omega'_{D_2} &= \Omega_{A_2} \cup \Omega_{A_3} \\ &= \{\langle 14, 17 \rangle\} \cup \{\langle 12.5, 20 \rangle\} \\ &= \{\langle 14, 17 \rangle, \langle 12.5, 20 \rangle\}\end{aligned}$$

L'ensemble des vecteurs non dominés de  $\Omega'_{D_2}$  constitue l'ensemble  $\Omega_{D_2}$ . On en déduit que  $\Omega_{D_2} = \{\langle 14, 17 \rangle, \langle 12.5, 20 \rangle\}$ .

- En  $A_1$  : l'événement  $\theta_1$  conduit à l'ensemble  $\Omega_{D_2} = \{\langle 14, 17 \rangle, \langle 12.5, 20 \rangle\}$  et l'événement  $\theta_2$  conduit à l'ensemble  $\{\langle 0, 0 \rangle\}$ . On en déduit alors :

$$\begin{aligned}\Omega'_{A_1} &= \left\langle \left\langle \min_{P \in \mathcal{P}_{A_1}} \{14 \times P(\theta_1) + 0 \times P(\theta_2)\}, \max_{P \in \mathcal{P}_{A_1}} \{17 \times P(\theta_1) + 0 \times P(\theta_2)\} \right\rangle, \right. \\ &\quad \left. \left\langle \min_{P \in \mathcal{P}_{A_1}} \{12.5 \times P(\theta_1) + 0 \times P(\theta_2)\}, \max_{P \in \mathcal{P}_{A_1}} \{20 \times P(\theta_1) + 0 \times P(\theta_2)\} \right\rangle \right\rangle \\ &= \{\langle 2.8, 5.1 \rangle, \langle 2.5, 6 \rangle\}\end{aligned}$$

L'ensemble des vecteurs non dominés de  $\Omega'_{A_1}$  constitue l'ensemble  $\Omega_{A_1}$ . On en déduit que  $\Omega_{A_1} = \{\langle 2.8, 5.1 \rangle, \langle 2.5, 6 \rangle\}$ .

- En  $D_1$  : la décision "haut" conduit à l'ensemble  $\Omega_{A_1} = \{\langle 2.8, 5.1 \rangle, \langle 2.5, 6 \rangle\}$  et la décision "bas" conduit à l'ensemble  $\{\langle 0, 0 \rangle\}$ . On en déduit alors :

$$\begin{aligned}\Omega'_{D_1} &= \Omega_{A_1} \cup \{\langle 0, 0 \rangle\} \\ &= \{\langle 2.8, 5.1 \rangle, \langle 2.5, 6 \rangle\} \cup \{\langle 0, 0 \rangle\} \\ &= \{\langle 2.8, 5.1 \rangle, \langle 2.5, 6 \rangle, \langle 0, 0 \rangle\}\end{aligned}$$

L'ensemble des vecteurs non dominés de  $\Omega'_{D_1}$  constitue l'ensemble  $\Omega_{D_1}$ . On en déduit que  $\Omega_{D_1} = \{\langle 14, 17 \rangle, \langle 12.5, 20 \rangle\}$ .

Enfin, chaque vecteur de l'ensemble  $\Omega_{D_1}$  est évalué selon le critère de Hurwicz et on sélectionne la stratégie associée au vecteur ayant la meilleure évaluation.

### Expérimentations numériques

L'algorithme proposé dans la partie précédente a été implémenté avec le langage C++. Les expérimentations numériques ont été réalisées sur un ordinateur doté d'un processeur Pentium IV CPU 2.13GHz et de 3.5Go de RAM.

Nous présentons dans cette partie, les résultats obtenus sur des arbres de décision générés aléatoirement. La structure des arbres de décision ont été générés de la même manière que pour les expérimentations dans le cas de l'incertitude totale : arbres binaires complets avec alternance des nœuds de décision et des nœuds de chance. Les valeurs

Prof. (nœuds)		Moyenne	Maximum
4 (31)	Cardinal ( $ \Omega $ )	2	4
	Temps (sec.)	0	0
6 (127)	Cardinal ( $ \Omega $ )	4	12
	Temps (sec.)	0	0
8 (511)	Cardinal ( $ \Omega $ )	9	28
	Temps (sec.)	0	0
10 (2047)	Cardinal ( $ \Omega $ )	32	101
	Temps (sec.)	0	0
12 (8191)	Cardinal ( $ \Omega $ )	144	600
	Temps (sec.)	0	0
14 (32767)	Cardinal ( $ \Omega $ )	940	3290
	Temps (sec.)	2	44
16 (131071)	Cardinal ( $ \Omega $ )	7182	40930
	Temps (sec.)	1189	3595

TABLE 5.2 – Temps d’exécution CPU (en sec.) et cardinal des ensembles  $\Omega_N$  construits lors de la détermination de la stratégie qui optimise le critère de Hurwicz dans un arbre de décision avec probabilités intervalles.

d’utilité aux feuilles sont des nombres réels générés aléatoirement dans l’intervalle  $[1; 500]$ . Concernant les probabilités intervalles, nous avons dans un premier temps généré aléatoirement des distributions de probabilité en chaque nœud de chance, puis nous avons tiré aléatoirement une borne inférieure et supérieure autour de chacune de ces probabilités. Les bornes générées constituent nos probabilités intervalles. De cette manière nous nous assurons qu’il existe toujours au moins une distribution de probabilité compatible avec les probabilités intervalles (la distribution de probabilité générée initialement).

Le tableau 5.2 donne pour chaque profondeur d’arbre (500 instances ont été générées pour chaque profondeur) le cardinal moyen et le cardinal maximal des ensembles construits en chaque nœud de l’arbre. Nous donnons également le temps moyen d’exécution de l’algorithme (en secondes) ainsi que le temps maximum observé pour au moins une instance. Comme le montre le tableau 5.2, nous sommes capables de traiter des instances de profondeur 16 (plus de 130000 nœuds). Notons qu’au-delà de cette profondeur la mémoire disponible n’est pas suffisante pour calculer l’ensemble des vecteurs non dominés.

### 5.2.2 Application du critère de Hurwicz dans un arbre de décision non séparable

Dans cette partie, nous nous intéressons à l’application du critère de Hurwicz dans un arbre de décision non séparable.

Puisque la programmation dynamique ne permet pas de déterminer une stratégie optimisant le critère de Hurwicz dans un arbre de décision séparable, elle ne permet pas non plus, à plus forte raison, de déterminer une stratégie optimisant le critère de Hurwicz dans un arbre de décision non séparable. Cependant le problème algorithmique qui en découle se révèle être plus complexe que dans le cas des arbres de décision séparables. En effet, nous montrons par la suite que la détermination d'une stratégie optimisant le critère de Hurwicz dans un arbre de décision séparable est un problème NP-difficile. Puisque le critère de Hurwicz généralise le critère  $\Gamma$ -Maximax, la démonstration découle directement de la proposition suivante :

**Proposition 8** *La détermination d'une stratégie optimisant le critère  $\Gamma$ -Maximax est dans un arbre de décision séparable est un problème NP-difficile.*

*Preuve.* On s'appuie sur une réduction polynomiale depuis le problème 3-SAT. Rappelons que le problème 3-SAT se formule comme suit :

INSTANCE : un ensemble  $X$  de variables booléennes, une collection  $C$  de clauses sur  $X$  telle que  $|c| = 3$  pour toute clause  $c \in C$ .

QUESTION : Existe-t-il une instantiation des variables booléennes de  $X$  qui satisfait simultanément toutes les clauses de  $C$  ?

Soient  $X = \{x_1, \dots, x_n\}$  et  $C = \{c_1, \dots, c_m\}$ . La construction polynomiale d'un arbre de décision à partir d'une instance du problème 3-SAT se réalise comme suit.

On définit un nœud de décision pour chaque clause de  $C$ . Etant donné  $c_i$  une clause de  $C$ , le nœud de décision associé dans l'arbre de décision, noté également  $c_i$ , possède trois fils (nœuds de chance) : un pour chaque littéral de la clause  $c_i$ . Ces nœuds de chance sont étiquetés par le nom du littéral correspondant. Tous les nœuds de chance  $x_i$  (resp.  $\bar{x}_i$ ) possèdent deux fils :

- le nœud d'utilité offrant une valeur d'utilité 1 avec la probabilité  $p_i \in [0, 1]$  (resp.  $1 - p_i$ ),
- le nœud d'utilité offrant une valeur d'utilité 0 avec la probabilité  $1 - p_i \in [0, 1]$  (resp.  $p_i$ ).

Afin de constituer un arbre de décision, on ajoute un nœud de chance  $A$  père de tous les nœuds de décision  $c_i$  ( $1 \leq i \leq m$ ) permettant d'atteindre chacun de ses fils avec la probabilité  $p_0 \in [1/m, 1/m]$ . Enfin, on rajoute un nœud de décision  $D$  à la racine ayant pour unique fils le nœud de chance  $A$  (pour conserver la convention que la racine d'un arbre de décision est un nœud de décision).

L'arbre de décision ainsi construit comporte  $m + 1$  nœuds de décision,  $3m + 1$  nœuds de chance et  $6m$  nœuds d'utilité. On distingue de plus  $n + 1$  variables représentant

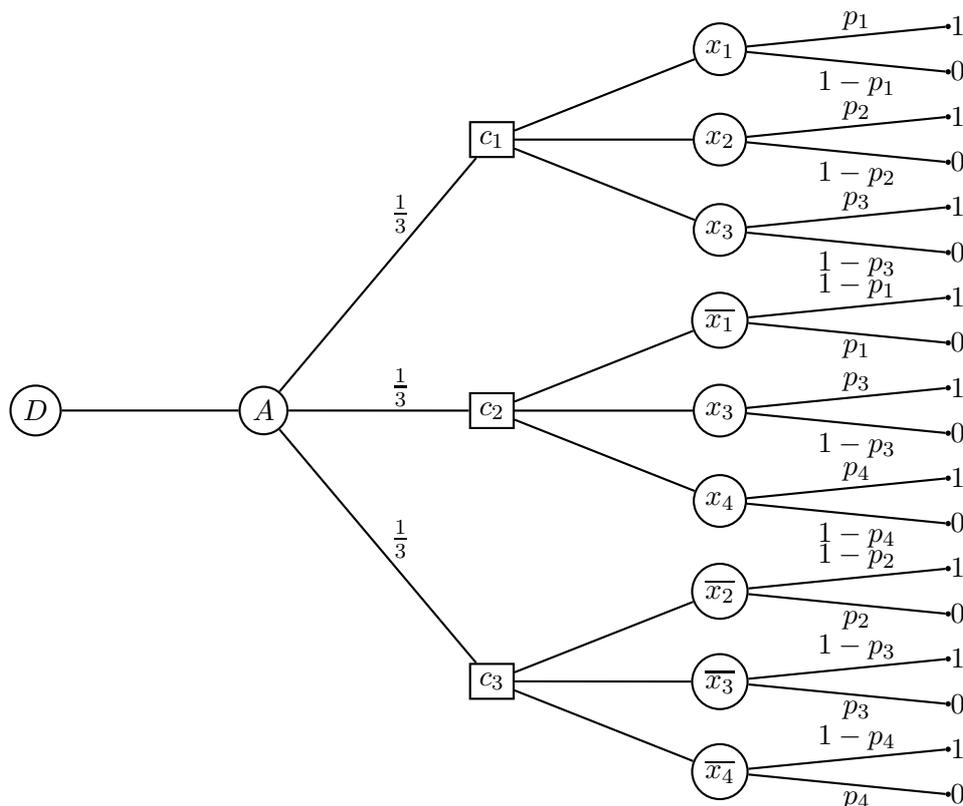


FIGURE 5.4 – Exemple de réduction en arbre de décision pour l'instance suivante de 3-SAT :  $(x_1 \vee x_2 \vee x_3) \wedge (\bar{x}_1 \vee x_3 \vee x_4) \wedge (\bar{x}_2 \vee \bar{x}_3 \vee \bar{x}_4)$ .

une probabilité (i.e.,  $p_i$  pour  $0 \leq i \leq n$ ). Ceci garantit bien la polynomialité de la transformation.

A titre d'illustration, sur la figure 5.4, nous donnons l'arbre de décision obtenu pour l'instance suivante de 3-SAT :  $(x_1 \vee x_2 \vee x_3) \wedge (\bar{x}_1 \vee x_3 \vee x_4) \wedge (\bar{x}_2 \vee \bar{x}_3 \vee \bar{x}_4)$ .

Remarquons que la valeur d'une stratégie optimisant le critère  $\Gamma$ -Maximax dans l'arbre de décision construit est bornée par 1.

Etant donnée une instantiation satisfaisant l'expression 3-SAT, on peut construire une stratégie dont l'évaluation par le critère  $\Gamma$ -Maximax vaut 1. En effet, pour chaque clause  $c_i$ , il existe un littéral qui a la valeur « Vrai ». Notons  $k_i$  l'indice du littéral de valeur « Vrai » appartenant à la clause  $c_i$ . En chaque nœud de décision  $c_i$ , on fixe la décision conduisant au nœud de chance  $x_{k_i}$ . Si, pour tout  $1 \leq i \leq m$ , on pose  $p_{k_i} = 1$  (resp. 0) si le littéral est positif (resp. négatif), l'espérance d'utilité de la stratégie vaut alors 1.

Inversement, à partir de toute stratégie dont l'évaluation avec le critère  $\Gamma$ -Maximax vaut 1, on peut construire une instantiation qui satisfait l'expression 3-SAT. En effet,

en chaque nœud de décision  $c_i$ , la décision fixée dans la stratégie conduit nécessairement à un nœud de chance retournant une utilité de 1 avec la probabilité 1. Notons  $k_i$  l'indice du nœud de chance choisi en  $c_i$ . Pour tout  $1 \leq i \leq m$ , on fixe la valeur du littéral  $x_{k_i}$  à « Vrai » (resp. « Faux ») si  $p_{k_i} = 1$  (resp. 0). L'instanciation qui en résulte satisfait l'expression 3-SAT. ■

### Algorithme de veto-process

Contournant la difficulté liée à la complexité du problème, dans le cas des critères  $\Gamma$ -Maximin et  $\Gamma$ -Maximax, Kikuti et al. (2005) ont présenté des algorithmes qui retournent des stratégies calculées par induction arrière dans l'arbre de décision. Cette approche consiste à remonter le temps du futur vers le présent, et à prendre systématiquement la décision qui maximise le critère choisi étant donné les anticipations sur les actions futures. D'un point de vue opérationnel, cette approche permet de déterminer une stratégie en un temps très compétitif. En contrepartie, la stratégie retournée est susceptible d'être sous-optimale et même « dominée » en un certain sens (Jaffray, 1999). Pour cette raison, à la suite de Jaffray, nous proposons un algorithme de veto-process. Une telle approche, bien que moins performante qu'une induction arrière classique, permet de garantir des propriétés solides sur la qualité de la stratégie retournée. De plus, contrairement à Kikuti et al. (2005), nous proposons ici d'appliquer le critère de Hurwicz, qui permet de modéliser des comportements moins extrêmes que les critères  $\Gamma$ -Maximin et  $\Gamma$ -Maximax.

Pour ce faire, nous adaptons, dans l'algorithme 5 page 157, la procédure de veto-process proposée par Jaffray et Nielsen (2006) (voir chapitre 2). Rappelons ici le principe guidant cette procédure : chaque nœud de décision représente un ego, dont les préférences suivent le critère de Hurwicz, seul maître de la décision à prendre au nœud où il est associé. On cherche alors à déterminer une stratégie qui satisfait autant que possible tous les egos, afin qu'aucun d'entre eux ne soit tenté de dévier lors de l'application de la stratégie. Une condition nécessaire à l'adhésion d'un ego à une sous-stratégie est le caractère E-admissible de cette dernière (i.e., qu'elle soit optimale au sens de l'espérance d'utilité pour au moins une distribution de probabilité). De ce fait, la stratégie globale calculée est E-admissible, ainsi que chaque sous-stratégie qui la compose. En effet, il est difficilement concevable qu'un ego puisse appliquer une stratégie qui n'est jamais optimale quelle que soit la vraie distribution de probabilité. C'est pourquoi on rejette toutes les sous-stratégies qui ne sont pas E-admissibles.

La procédure se déroule comme suit. L'algorithme parcourt chaque nœud  $N$  en partant des feuilles de l'arbre jusqu'à la racine. En chacun de ces nœuds, un ensemble  $\mathcal{S}_N$  de stratégies est construit à partir des ensemble  $\mathcal{S}_{N'}$  de ses fils. Ensuite, chaque stratégie est évaluée avec le critère de Hurwicz. Enfin, on ne conserve dans l'ensemble  $\mathcal{S}_N$  que les  $k$  meilleures stratégies parmi celles qui sont E-admissibles, et dont l'évaluation est

---

**Algorithme 5 : VetoProcess**(arbre de décision  $T$ , réel  $\Theta$ , entier  $k$ )
 

---

**Sortie** : une stratégie de compromis entre les différents egos

**pour chaque** nœud  $N$  de profondeur  $p$  à 0 dans  $T$  **faire**

<b>si</b> $N \in \mathcal{N}_D$ <b>alors</b> /* Cas des nœuds de décision */ $\bar{\Delta}_N \leftarrow \bigcup_{N' \in \text{fils}(N)} \{(N, N')\} \cup \Delta : \Delta \in \bar{\Delta}_{N'}$		
<b>si</b> $N \in \mathcal{N}_C$ <b>alors</b> /* Cas des nœuds de chance */ $\bar{\Delta}_N \leftarrow \prod_{N' \in \text{fils}(N)} \bar{\Delta}_{N'}$		
<b>pour chaque</b> stratégie $\Delta \in \bar{\Delta}_N$ <b>faire</b> $V_\Delta \leftarrow Hz(\Delta)$ $V_{max} \leftarrow \max_{\Delta \in \bar{\Delta}_N} \{V_\Delta\}$		
<b>pour chaque</b> stratégie $\Delta \in \bar{\Delta}_N$ <b>faire</b> <table style="border-collapse: collapse; margin-left: 20px;"> <tr> <td style="border-left: 1px solid black; padding-left: 5px; vertical-align: top;"> <b>si</b> (<math>\Delta</math> n'est pas E-admissible) OU (<math>V_\Delta &lt; V_{max} - \Theta</math>) <b>alors</b>  <table style="border-collapse: collapse; margin-left: 20px;"> <tr> <td style="border-left: 1px solid black; padding-left: 5px; vertical-align: top;"> <math>\bar{\Delta}_N \leftarrow \bar{\Delta}_N \setminus \{\Delta\}</math> </td> </tr> </table> </td> </tr> </table>	<b>si</b> ( $\Delta$ n'est pas E-admissible) OU ( $V_\Delta < V_{max} - \Theta$ ) <b>alors</b> <table style="border-collapse: collapse; margin-left: 20px;"> <tr> <td style="border-left: 1px solid black; padding-left: 5px; vertical-align: top;"> <math>\bar{\Delta}_N \leftarrow \bar{\Delta}_N \setminus \{\Delta\}</math> </td> </tr> </table>	$\bar{\Delta}_N \leftarrow \bar{\Delta}_N \setminus \{\Delta\}$
<b>si</b> ( $\Delta$ n'est pas E-admissible) OU ( $V_\Delta < V_{max} - \Theta$ ) <b>alors</b> <table style="border-collapse: collapse; margin-left: 20px;"> <tr> <td style="border-left: 1px solid black; padding-left: 5px; vertical-align: top;"> <math>\bar{\Delta}_N \leftarrow \bar{\Delta}_N \setminus \{\Delta\}</math> </td> </tr> </table>	$\bar{\Delta}_N \leftarrow \bar{\Delta}_N \setminus \{\Delta\}$	
$\bar{\Delta}_N \leftarrow \bar{\Delta}_N \setminus \{\Delta\}$		
<b>tant que</b> $ \bar{\Delta}_N  > k$ <b>faire</b> <table style="border-collapse: collapse; margin-left: 20px;"> <tr> <td style="border-left: 1px solid black; padding-left: 5px; vertical-align: top;"> <math>\bar{\Delta}_N \leftarrow \bar{\Delta}_N \setminus \{\arg \min_{\Delta \in \bar{\Delta}_N} \{V_\Delta\}\}</math> </td> </tr> </table>	$\bar{\Delta}_N \leftarrow \bar{\Delta}_N \setminus \{\arg \min_{\Delta \in \bar{\Delta}_N} \{V_\Delta\}\}$	
$\bar{\Delta}_N \leftarrow \bar{\Delta}_N \setminus \{\arg \min_{\Delta \in \bar{\Delta}_N} \{V_\Delta\}\}$		

**Retourner**  $\arg \max_{\Delta \in \bar{\Delta}_{N_r}} \{V_\Delta\}$ 


---

supérieure à  $V_{max} - \theta$ . Enfin, une fois l'ensemble de stratégies construit en la racine, on retourne celle d'évaluation maximale.

Pour mettre un œuvre un tel algorithme, il est nécessaire de pouvoir évaluer une stratégie au sens du critère de Hurwicz, ainsi que de pouvoir déterminer le caractère E-admissible de la stratégie. C'est précisément le propos des sections qui suivent.

### Evaluation d'une stratégie selon le critère de Hurwicz

Avant toute chose, il nous faut introduire quelques notations qui nous seront utiles par la suite. Soit  $X_1, \dots, X_n$  l'ensemble des variables aléatoires figurant dans l'arbre de décision  $\mathcal{T}$ , et  $X = \langle X_1, \dots, X_n \rangle$  le vecteur aléatoire correspondant. Nous notons  $\mathcal{D}(Y)$  le domaine d'une variable aléatoire (ou vecteur aléatoire)  $Y$ . De plus, étant donné un sous-ensemble  $I \subseteq \{1, \dots, n\}$  d'indices, nous désignons par  $X_I$  le vecteur aléatoire ayant pour composantes les  $X_i$  pour  $i \in I$ . En outre, étant donné un nœud de chance  $N$ , nous désignons par  $id(N)$  l'indice  $i$  de la variable aléatoire  $X_i$  associée au nœud  $N$ . De même, étant donné un nœud  $N$ , nous désignons par  $\pi(N)$  l'ensemble des *indices* des variables aléatoires sur le chemin de la racine à  $N$  ( $id(N)$  exclu si  $N \in \mathcal{N}_C$ ). L'ensemble

complémentaire de ce dernier est noté  $\pi^c(N) = \{1, \dots, n\} \setminus \pi(N)$ . Enfin, nous désignons par  $x(N)$  le vecteur des *valeurs affectées* aux variables aléatoires sur le chemin de la racine à  $N$ .

Par exemple, sur l'arbre de la figure 5.2 page 146,  $id(X_2^2) = 2$  car la variable aléatoire associée à  $X_2^2$  est  $X_2$ , et  $\pi(X_2^2) = \{1\}$  car seule la variable aléatoire  $X_1$  figure sur le chemin menant de  $D_1$  à  $X_2^2$ . Par conséquent, on a  $X_{\pi(X_2^2)} = \langle X_1 \rangle$ , et  $x(X_2^2) = \langle aucune \rangle$  car il est nécessaire que  $X_1$  prenne la valeur *aucune* pour arriver en  $X_2^2$ .

Nous nous intéressons maintenant à l'évaluation d'une stratégie  $\Delta$  selon le critère de Hurwicz. Il est important de remarquer qu'une stratégie est caractérisée par l'ensemble  $\mathcal{U} \subset \mathcal{C}$  des nœuds d'utilité qu'elle permet d'atteindre. L'évaluation  $H_z(\Delta)$  d'une stratégie  $\Delta$  selon le critère de Hurwicz peut s'écrire comme suit :

$$\alpha \min_{P \in \mathcal{P}_K} \sum_{N \in \mathcal{U}} P(X_{\pi(N)} = x(N)) \times u(N) + (1 - \alpha) \max_{P \in \mathcal{P}_K} \sum_{N \in \mathcal{U}} P(X_{\pi(N)} = x(N)) \times u(N)$$

où  $\alpha \in [0, 1]$  et  $P(X_{\pi(N)} = x(N))$  représente (conformément à nos notations) le produit des probabilités assignées aux branches le long du chemin de la racine à  $N$ .

Face à la nature combinatoire du problème, nous allons formuler un programme linéaire pour évaluer la valeur de  $\Delta$  selon le critère  $\Gamma$ -Maximin (le terme de coefficient  $\alpha$ ), et un autre programme linéaire pour l'évaluer selon le critère  $\Gamma$ -Maximax (le terme de coefficient  $1 - \alpha$ ). Il suffira ensuite de combiner les valeurs obtenues avec le paramètre  $\alpha$  pour obtenir l'évaluation selon le critère de Hurwicz.

La principale difficulté, dans un arbre de décision non séparable, consiste alors à caractériser l'ensemble des distributions  $P \in \mathcal{P}_K$ . Afin de caractériser l'ensemble  $\mathcal{P}_K$  des distributions de probabilité compatibles avec  $K$ , nous allons définir un jeu de contraintes linéaires (que nous appellerons *contraintes de cohérence* par la suite) portant sur l'ensemble des variables  $P(X = x)$  pour tout  $x$ . En effet, cet ensemble de variables suffit à caractériser une distribution de probabilité. Dans le but de vérifier les axiomes de Kolmogorov, chacune de ces variables doit être comprise entre 0 et 1 et la somme de ces dernières doit être égale à 1 puisqu'elles représentent les probabilités sur les événements élémentaires. Ce que l'on traduit par la contrainte suivante :

$$\sum_{x \in \mathcal{D}(X)} P(X = x) = 1 \tag{5.1}$$

Par ailleurs, afin de lier les probabilités sur les événements élémentaires  $P(X = x)$  aux probabilités  $P(X_{\pi(N)} = x(N))$  il est nécessaire d'introduire les contraintes linéaires suivantes pour tout  $N \in \mathcal{N}_C \cup \mathcal{N}_U$  :

$$P(X_{\pi(N)} = x(N)) = \sum_{y \in \mathcal{D}(X_{\pi^c(N)})} P(X = (x(N), y)) \quad (5.2)$$

où  $(x(N), y)$  désigne l'affectation du vecteur aléatoire  $X$  tel que  $X_{\pi(N)} = x(N)$  et  $X_{\pi^c(N)} = y$ .

Enfin, pour garantir que les intervalles de probabilités imprécises indiqués dans l'arbre sont bien respectés, il est nécessaire d'introduire les contraintes suivantes pour tout nœud de chance  $N$  en utilisant les règles de Bayes :

$$P(X_{\pi(N) \cup \{id(N)\}} = (x(N), y)) \leq P(X_{\pi(N)} = x(N))P^+(X_{id(N)} = y | X_{\pi(N)} = x(N)) \quad (5.3)$$

$$P(X_{\pi(N) \cup \{id(N)\}} = (x(N), y)) \geq P(X_{\pi(N)} = x(N))P^-(X_{id(N)} = y | X_{\pi(N)} = x(N)) \quad (5.4)$$

Notons que la probabilité  $P(X_{\pi(N) \cup \{id(N)\}} = (x(N), y))$  n'est autre que la variable  $P(X_{\pi(N')} = (x(N), y))$  dès lors que  $N$  est le nœud de chance précédant immédiatement  $N'$  et que l'affectation  $X_{id(N)} = y$  figure sur le chemin menant à  $N'$ .

**Exemple 24** *Considérons l'arbre de décision  $\mathcal{T}$  de la figure 5.2 page 146. Dans un premier temps, nous définissons l'ensemble des contraintes de cohérence permettant de caractériser  $\mathcal{P}_{\mathcal{T}}$ . L'ensemble des variables définissant les probabilités sur les événements élémentaires est le suivant :*

$$\{p_{as}, p_{ah}, p_{ai}, p_{fs}, p_{fh}, p_{fi}, p_{Fs}, p_{Fh}, p_{Fi}\}$$

où  $p_{as}$  correspond à la probabilité  $P(X_1 = \underline{aucune}, X_2 = \underline{sec})$  et ainsi de suite (il est à noter que le symbole  $f$  est utilisé pour faible, et  $F$  pour forte). Ces variables ont pour domaine de définition l'intervalle  $[0; 1]$ . La contrainte 5.1 est alors la suivante :

$$p_{as} + p_{ah} + p_{ai} + p_{fs} + p_{fh} + p_{fi} + p_{Fs} + p_{Fh} + p_{Fi} = 1$$

Ensuite, en chaque nœud de chance, l'ensemble des contraintes de cohérences restantes est le suivant :

Au nœud de chance  $X_2^1$  :

- $p_i = p_{ai} + p_{fi} + p_{Fi}$  (cont. 5.2)
- $p_h = p_{ah} + p_{fh} + p_{Fh}$  (cont. 5.2)
- $p_s = p_{as} + p_{fs} + p_{Fs}$  (cont. 5.2)
- $0.307 \leq p_i \leq 0.456$  (cont. 5.3 et 5.4)
- $0.309 \leq p_h \leq 0.386$  (cont. 5.3 et 5.4)
- $0.214 \leq p_s \leq 0.344$  (cont. 5.3 et 5.4)

Au nœud de chance  $X_1$  :

- $p_a = p_{ai} + p_{ah} + p_{as}$  (cont. 5.2)
- $p_f = p_{fi} + p_{fh} + p_{fs}$  (cont. 5.2)
- $p_F = p_{Fi} + p_{Fh} + p_{Fs}$  (cont. 5.2)
- $0.181 \leq p_a \leq 0.222$  (cont. 5.3 et 5.4)
- $0.333 \leq p_f \leq 0.363$  (cont. 5.3 et 5.4)
- $0.444 \leq p_F \leq 0.454$  (cont. 5.3 et 5.4)

Au nœud de chance  $X_2^2$  :

- $0.125p_a \leq p_{ai} \leq 0.181p_a$  (cont. 5.3 et 5.4)
- $0.222p_a \leq p_{ah} \leq 0.272p_a$  (cont. 5.3 et 5.4)
- $0.500p_a \leq p_{as} \leq 0.666p_a$  (cont. 5.3 et 5.4)

Au nœud de chance  $X_2^3$  :

- $0.250p_f \leq p_{fi} \leq 0.363p_f$  (cont. 5.3 et 5.4)
- $0.363p_f \leq p_{fh} \leq 0.444p_f$  (cont. 5.3 et 5.4)
- $0.222p_f \leq p_{fs} \leq 0.333p_f$  (cont. 5.3 et 5.4)

Au nœud de chance  $X_2^4$  :

- $0.454p_F \leq p_{Fi} \leq 0.625p_F$  (cont. 5.3 et 5.4)
- $0.333p_F \leq p_{Fh} \leq 0.363p_F$  (cont. 5.3 et 5.4)
- $0.111p_F \leq p_{Fs} \leq 0.166p_F$  (cont. 5.3 et 5.4)

Une fois définies les contraintes de cohérences, l'évaluation d'une stratégie selon le critère d'Hurwicz peut être réalisée en définissant les deux programmes linéaires donnant les valeurs selon  $\Gamma$ -Maximin et  $\Gamma$ -Maximax. Considérons par exemple la stratégie  $\Delta = \{D_1 = \text{sonder}, D_2^2 = \text{ne pas forer}, D_2^3 = \text{forer}, D_2^4 = \text{forer}\}$ . Les programmes linéaires se formulent comme suit :

$$\Gamma_{\min}(\Delta) = \min \quad -10p_a + 190p_{fi} + 40p_{fh} - 80p_{fs} + 190p_{Fi} + 40p_{Fh} - 80p_{Fs}$$

s.c : contraintes de cohérence

$$\Gamma_{\max}(\Delta) = \max \quad -10p_a + 190p_{fi} + 40p_{fh} - 80p_{fs} + 190p_{Fi} + 40p_{Fh} - 80p_{Fs}$$

s.c : contraintes de cohérence

Enfin la valeur de la stratégie  $\Delta$  selon le critère d'Hurwicz est alors :

$$Hz(\Delta) = \alpha\Gamma_{\min}(\Delta) + (1 - \alpha)\Gamma_{\max}(\Delta)$$

### E-admissibilité d'une stratégie

Rappelons qu'une stratégie  $\Delta$  est dite *E-admissible* s'il existe au moins une distribution de probabilité  $P \in \mathcal{P}_K$  pour laquelle la stratégie  $\Delta$  est maximale au sens de l'espérance d'utilité. Nous montrons maintenant comment réduire le test d'E-admissibilité

d'une stratégie à un problème de satisfaction de contraintes linéaires. Plus précisément, la stratégie  $\Delta$  sera E-admissible si et seulement si le polyèdre défini par l'ensemble des contraintes est non-vide. L'ensemble des contraintes comporte bien évidemment les contraintes de cohérence (sur les probabilités) définies précédemment. Aux contraintes de cohérence s'ajoute un ensemble de contraintes dont la satisfaction permet de vérifier qu'il existe une distribution de probabilité  $P$  compatible avec les contraintes de cohérence telle que la stratégie considérée est optimale au sens de l'espérance d'utilité. Pour ce faire, on introduit une variable réelle  $y_N$  pour chaque nœud  $N$  (cette variable est destinée à représenter la valeur de l'espérance d'utilité atteinte par la stratégie  $\Delta$  dans l'arbre  $T(N)$ ) et on définit les contraintes suivantes :

$$y_N = P\left(X_{\pi(N)} = x(N)\right) \times u(N) \quad \forall N \in \mathcal{C} \quad (5.5)$$

$$y_N = \sum_{N' \in \text{fils}(N)} y_{N'} \quad \forall N \in \mathcal{N}_C \quad (5.6)$$

Enfin, un dernier jeu de contraintes impose que la stratégie  $\Delta$  soit maximale au sens de l'espérance d'utilité. Pour chaque arc  $(N, N')$  issue d'un nœud de décision  $N$  :

- si  $(N, N')$  appartient à la stratégie  $\Delta$ , on ajoute la contrainte

$$y_N \leq y_{N'} \quad (5.7)$$

- si  $(N, N')$  n'appartient pas à la stratégie  $\Delta$ , on ajoute la contrainte

$$y_N \geq y_{N'} \quad (5.8)$$

Ce dernier jeu de contraintes impose que  $y_N$  vaut au plus l'espérance d'utilité de la sous-stratégie induite par  $\Delta$  dans le sous-arbre de racine  $N$ , et au moins l'espérance d'utilité maximum dans le sous-arbre de racine  $N$ . On en conclut que si une telle espérance existe en chaque nœud de décision, alors il existe une distribution de probabilité  $P \in \mathcal{P}_K$  (caractérisée par les variables  $P(X = x)$ ) telle que l'espérance d'utilité de  $\Delta$  soit de valeur maximale. Pour illustration, nous indiquons ci-dessous les contraintes obtenues pour une stratégie particulière dans l'exemple du forage pétrolier.

**Exemple 25** *Considérons l'arbre de décision de la figure 5.2 et la stratégie  $\Delta = \{D_1 = \text{sonder}, D_2^2 = \text{ne pas forer}, D_2^3 = \text{forer}, D_2^4 = \text{forer}\}$ . Les contraintes de cohérence pour cet arbre sont données dans l'exemple 24. Par souci de simplicité, nous n'indiquons pas ici les contraintes et les variables associées aux nœuds d'utilité. Les contraintes en chacun des autres nœuds sont les suivantes :*

- Au nœud  $X_2^1$  :  $y_{X_2^1} = 200p_i + 50p_h - 70p_s$
- Au nœud  $D_2^1$  :  $y_{D_2^1} \geq 0$  et  $y_{D_2^1} \geq y_{X_2^1}$

- Au nœud  $X_2^2$  :  $y_{X_2^2} = 200p_{ai} + 50p_{ah} - 70p_{as}$
- Au nœud  $D_2^2$  :  $y_{D_2^2} \leq -10p_a$  et  $y_{D_2^2} \geq y_{X_2^2}$
- Au nœud  $X_2^3$  :  $y_{X_2^3} = 200p_{fi} + 50p_{fh} - 70p_{fs}$
- Au nœud  $D_2^3$  :  $y_{D_2^3} \geq -10p_f$  et  $y_{D_2^3} \leq y_{X_2^3}$
- Au nœud  $X_2^4$  :  $y_{X_2^4} = 200p_{Fi} + 50p_{Fh} - 70p_{Fs}$
- Au nœud  $D_2^4$  :  $y_{D_2^4} \geq -10p_F$  et  $y_{D_2^4} \leq y_{X_2^4}$
- Au nœud  $X_1$  :  $y_{X_1} = y_{D_2^2} + y_{D_2^3} + y_{D_2^4}$
- Au nœud  $D_1$  :  $y_{D_1} \geq y_{D_2^1}$  et  $y_{D_1} \leq y_{X_1}$

### Expérimentations numériques

L'algorithme proposé dans la partie précédente a été implémenté avec le langage C++. Les expérimentations numériques ont été réalisées sur un ordinateur doté d'un processeur Pentium IV CPU 2.13GHz et de 3.5Go de RAM.

Nous présentons dans cette partie, les résultats obtenus sur des arbres de décision générés aléatoirement. La structure des arbres de décision ont été générés de la même manière que pour les expérimentations dans le cas de l'incertitude totale et du risque imprécis pour les arbres de décision séparables : arbres binaires complets avec alternance des nœuds de décision et des nœuds de chance. Les valeurs d'utilité aux feuilles sont des nombres réels générés aléatoirement dans l'intervalle  $[1; 500]$ . Chaque nœud de chance est associé à une variable aléatoire. Concernant les probabilités intervalles, nous avons dans un premier temps généré aléatoirement une distribution de probabilité sur les événements élémentaires. A partir de cette distribution, en suivant les règles de Bayes, nous avons calculé les probabilités conditionnelles précises en chaque nœud de chance, puis nous avons tiré aléatoirement une borne inférieure et supérieure autour de chacune de ces probabilités. De cette manière nous nous assurons qu'il existe toujours au moins une distribution de probabilité compatible avec les probabilités intervalles (celle générée initialement).

Dans un premier temps, nous avons traité des arbres de décision où chaque niveau de nœud de chance est associé à une même variable aléatoire. Remarquons que, dans ce cas, le nombre de probabilités portant sur les événements élémentaires ( $\in O(2^v)$  si  $v$  est le nombre de variables aléatoires) est linéaire en la taille de l'arbre de décision puisque  $v$  est alors une valeur logarithmique du nombre  $n$  de nœuds dans l'arbre. Ceci implique une complexité polynomiale de l'algorithme en fonction de  $k$  et  $n$ . Pour cette série d'expérimentations nous avons fait varier  $k$ ,  $\theta$  et  $p$  (la profondeur de l'arbre). Nous nous sommes intéressés aux temps d'exécution de l'algorithme en fonction de ces paramètres. Pour chaque jeu de paramètres où  $p \leq 12$ , 100 arbres ont été générés aléatoirement. Nous avons indiqué sur la table 5.3 page 163 la moyenne des temps d'exécution en secondes.

Prof. (nœuds)	4 (31)	6 (127)	8 (511)	10 (2047)	12 (8191)	14 (32767)
$k = 2$ $\theta = 10$	< 0.01	0.04	0.39	6.22	111.73	2518.95
$k = 2$ $\theta = 50$	< 0.01	0.04	0.38	6.24	112.78	2487.56
$k = 2$ $\theta = 100$	< 0.01	0.04	0.39	6.18	112.34	2660.13
$k = 2$ $\theta = 300$	< 0.01	0.04	0.37	6.23	110.15	2532.40
$k = 5$ $\theta = 10$	< 0.01	0.04	0.38	6.34	114.10	2711.24
$k = 5$ $\theta = 50$	< 0.01	0.04	0.40	6.42	115.21	2689.83
$k = 5$ $\theta = 100$	< 0.01	0.04	0.38	6.39	115.75	2802.88
$k = 5$ $\theta = 300$	< 0.01	0.04	0.38	6.41	117.60	2765.12
$k = 10$ $\theta = 10$	< 0.01	0.04	0.39	6.78	121.35	3398.91
$k = 10$ $\theta = 50$	< 0.01	0.04	0.37	6.82	126.03	2991.33
$k = 10$ $\theta = 100$	< 0.01	0.04	0.38	6.91	124.53	3132.48
$k = 10$ $\theta = 300$	< 0.01	0.04	0.38	6.87	125.84	3255.04

TABLE 5.3 – Moyenne du temps d’exécution, en seconde, en fonction de la profondeur  $p$  de l’arbre et des paramètres  $k$  et  $\theta$ .

Notons que, pour la profondeur 14, seulement cinq instances ont été générées aléatoirement pour les différents paramètres. Les temps d’exécution dépendent évidemment de  $k$  et  $\theta$  puisqu’ils conditionnent le nombre de stratégies retenues en chaque nœud. Avec ce type d’instances, nous sommes capables de résoudre rapidement (en moins d’une heure) des instances jusqu’à la profondeur 14 (soit plus de 32000 nœuds).

Dans un second temps, nous nous sommes intéressés aux arbres de décision où le nombre  $v$  de variables aléatoires distinctes dans l’arbre de décision est indépendant de la taille de l’instance. Pour cette série d’expérimentations, nous avons fait varier la profondeur  $p$  et le nombre de variable  $v$ , en fixant le paramètre  $k$  à 5 et le paramètre  $\theta$  à 50. Pour chaque jeu de paramètres, nous avons généré 100 instances en attribuant de manière aléatoire les variables aléatoires aux nœuds de chance. Les temps indiqués sur la table 5.4 représentent la moyenne du temps d’exécution, en secondes, pour chaque jeu de paramètres (le symbole “–” indique que l’instance est impossible (i.e., moins de variables distinctes que « d’étages » de nœud de chance dans l’arbre, ou plus de variables distinctes que de nœuds de chance dans l’arbre) et le symbole “X” correspond aux jeux de paramètres impliquant un temps de résolution supérieur à 1 000 secondes). Nous constatons qu’à partir d’un certain nombre  $v$  de variables aléatoires, l’ajout d’une variable aléatoire supplémentaire à l’instance double le temps d’exécution moyen de l’algorithme.

Prof. (nœuds)	4 (31)	6 (127)	8 (511)	10 (2047)	12 (8191)
$v = 2$	< 0.01	—	—	—	—
$v = 3$	< 0.01	0.04	—	—	—
$v = 4$	< 0.01	0.04	0.40	—	—
$v = 5$	< 0.01	0.05	0.66	6.42	—
$v = 6$	< 0.01	0.05	0.85	7.65	115.21
$v = 7$	< 0.01	0.08	1.18	10.87	135.28
$v = 8$	< 0.01	0.13	1.41	13.71	214.21
$v = 9$	0.02	0.17	1.86	21.96	287.68
$v = 10$	0.03	0.25	3.44	35.61	485.16
$v = 11$	—	0.56	6.25	62.95	698.81
$v = 12$	—	0.94	11.40	94.35	X
$v = 13$	—	1.92	22.10	165.61	X
$v = 14$	—	4.03	42.03	342.91	X
$v = 15$	—	9.58	80.01	680.95	X
$v = 16$	—	20.98	162.46	X	X
$v = 17$	—	49.11	347.77	X	X
$v = 18$	—	98.06	786.34	X	X

TABLE 5.4 – Moyenne du temps d’exécution en seconde en fonction de la profondeur de l’arbre et du nombre de variables aléatoires  $v$ .

## Conclusion

Dans ce chapitre, nous nous sommes intéressés à appliquer le critère de Hurwicz dans des contextes d’incertitude totale ou de risque imprécis.

Dans le cadre de l’incertitude totale, nous avons montré que le problème de la détermination d’une stratégie optimisant le critère de Hurwicz pouvait être résolu en temps polynomial. L’algorithme proposé, fondé sur la programmation dynamique biobjectif, s’est révélé être très efficace. En effet, lors des expérimentations numériques que nous avons menées, nous avons été capables de traiter, en temps raisonnable, des arbres de décision de très grande taille (plus de 8 000 000 nœuds, les arbres de plus grandes tailles ne tenant pas en mémoire).

Dans le cadre du risque imprécis, le problème algorithmique induit est plus complexe. Dans un premier temps, nous avons distingué le cas des arbres de décision séparables et le cas des arbres de décision non-séparables. Lorsque l’arbre de décision est séparable, le problème algorithmique est très proche de celui proposé dans le cadre de l’incertitude totale. En effet, l’adaptation de l’algorithme par programmation dynamique biobjectif permet de résoudre le problème de manière assez efficace.

Concernant le cas des arbres de décision non-séparables, nous avons montré que le problème algorithmique induit est NP-difficile. Nous avons alors présenté une approche de type veto-process pour résoudre ce problème. L’algorithme proposé est une adaptation

de l'algorithme de Jaffray et Nielsen (2006) proposé dans le cadre du risque pour le critère RDU. Cet algorithme s'est révélé être efficace et permet, de plus, d'assurer de bonnes propriétés sur la stratégie déterminée, contrairement à une approche directe par induction arrière.

Dans des travaux futurs, il pourrait être intéressant de proposer un algorithme qui détermine la stratégie optimisant le critère de Hurwicz depuis la racine. Du fait de la complexité du problème, une approche par énumération implicite semble être une piste à privilégier. La difficulté sera alors de déterminer une procédure efficace de calcul d'une borne supérieure. L'initialisation et le calcul de la borne inférieure peuvent eux être calqués sur le principe algorithmique de veto-process que nous proposons ici.



# Conclusion

## Synthèse

Dans cette thèse, nous nous sommes intéressés à établir des algorithmes pour mettre en œuvre des critères de décision non-EU dans un cadre séquentiel.

Tout d'abord, nous avons traité le cas de l'incertitude risquée. Dans ce cadre, le critère EU domine très largement la littérature. En effet, le respect de l'axiome d'indépendance lui confère de bonnes propriétés algorithmiques pour le développement d'algorithmes d'optimisation. Cependant, ses faiblesses sur le plan descriptif découlent de ce même axiome. Nous nous sommes donc tout naturellement portés sur l'application, dans un cadre séquentiel, d'un autre critère de décision : le modèle RDU. Ce dernier relâche l'axiome d'indépendance, ce qui lui confère un fort pouvoir descriptif. Cependant, la relaxation de l'axiome d'indépendance force à renoncer aux propriétés algorithmiques dont bénéficie le modèle EU, et le problème algorithmique lié à l'optimisation du critère RDU devient alors bien plus complexe. L'absence d'algorithme efficace dans la littérature pour l'optimisation du modèle RDU constitue le frein principale à son utilisation dans la pratique.

Dans un premier temps, nous nous sommes intéressés à l'optimisation du critère RDU dans le formalisme des arbres de décision. Les arbres de décision permettent la représentation d'un problème de décision séquentielle en explicitant les différents scénarios. Nous avons montré que le problème induit par l'optimisation du critère RDU à la racine d'un arbre de décision était un problème NP-difficile. Face à ce constat, nous avons développé des algorithmes exponentiels exacts pour résoudre ce problème. Parmi les différentes approches algorithmiques proposées, l'une d'entre elles, basée sur le principe du branch and bound, s'est révélée être particulièrement efficace. Cette méthode nous a permis, entre autres, de résoudre en temps raisonnable le problème de « Qui veut gagner des millions ? », qui se modélise sous la forme d'un arbre de décision comportant plus de 75 millions de noeuds.

Le formalisme des arbres de décision est cependant très gourmand en mémoire. En effet, le nombre de scénarios distincts est généralement très grand. Pour pallier cet inconvénient, d'autres représentations graphiques ont vu le jour, dont les diagrammes d'influence.

Ce modèle graphique tire profit des indépendances présentes dans le problème pour obtenir une représentation définie « en compréhension » d'un arbre de décision. Cependant, ce formalisme reste confronté au problème de la représentation en mémoire d'une stratégie. En effet, ces dernières sont généralement exponentielles en la taille du diagramme d'influence. Néanmoins, le modèle EU peut tirer profit de l'axiome conséquentialiste pour factoriser la représentation d'une stratégie. En revanche, la représentation d'une stratégie lorsque l'on utilise le critère RDU devient problématique puisque ce dernier ne respecte pas l'axiome conséquentialiste. Nous proposons alors de réaliser un  $\Lambda$ -affaiblissement de l'axiome conséquentialiste afin d'obtenir un compromis entre le pouvoir descriptif du modèle et la compacité de la représentation d'une stratégie en mémoire. Pour ce faire, nous avons détaillé une procédure de pré-traitement du diagramme d'influence qui conduit à accroître l'ensemble des stratégies factorisables. Enfin, nous avons proposé un algorithme qui recherche une stratégie optimisant le critère RDU parmi celles-ci.

Nous nous sommes ensuite intéressés à l'optimisation du critère de Hurwicz dans un contexte d'incertitude totale. Nous avons proposé un algorithme polynomial, basé sur la programmation dynamique biobjectif, pour déterminer une stratégie optimisant le critère de Hurwicz dans un arbre de décision dans un contexte d'incertitude totale. Nous nous sommes par ailleurs penchés sur l'optimisation du critère de Hurwicz dans le cas du risque imprécis. Nous avons distingué deux types d'arbres : les arbres séparables et les arbres non-séparables. Pour les arbres séparables nous avons proposé un algorithme à nouveau basé sur la programmation dynamique biobjectif, en revanche, pour le cas des arbres non-séparables le problème se révèle être bien plus complexe. En effet, nous avons montré que le problème est NP-difficile. Nous avons alors proposé un algorithme de veto-process permettant d'appliquer le critère de Hurwicz dans ce dernier cas. La stratégie retournée n'est pas nécessairement optimale vue de la racine, mais se justifie pleinement d'un point de vue décisionnel.

## Discussion

Quelle que soit la modélisation de l'incertitude à laquelle nous avons été confronté, nous avons toujours considéré qu'il était possible de travailler sur les loteries réduites. Cette hypothèse classique, si elle est pertinente dans un cadre de décision non séquentielle, semble beaucoup plus discutable dans un cadre de décision séquentielle. En effet, dans un cadre non séquentiel, sa justification repose sur le fait que la représentation de la structure d'incertitude ne doit pas influencer le décideur dans ses décisions. Ceci implique donc une indifférence de la part du décideur sur l'ordre dans lequel les événements surviennent. Ainsi, on considère qu'un décideur reste indifférent entre jeter un dé à six faces puis jeter une pièce au fait jeter d'abord la pièce puis ensuite le dé à six faces, ou encore au fait

de lancer un dé à douze faces (où chacune des combinaisons entre la valeur d'un dé à six faces et la valeur Pile ou Face est représentée sur une face du dé). Dans un cadre de décision séquentielle, la validité d'une telle hypothèse semble bien plus fragile. En effet, si il semble naturel que l'ordre des événements n'ait aucune importance lorsque l'on ne peut intervenir entre ces derniers, cette hypothèse semble beaucoup plus discutable lorsque l'on peut prendre des décisions entre ces événements. Autrement dit, si entre chaque décision la réduction des loteries se justifie aisément, la justification de la réduction des loteries qui précèdent une décision avec celles qui suivent cette même décision est un problème bien plus délicat. Kreps et Porteus (1979) proposent des exemples de situations séquentielles dans le risque où l'axiome de réduction des loteries composées est violé par des décideurs dont les préférences suivent le modèle EU. Cependant ils montrent que le classement des loteries induites par les préférences d'un tel décideur reste toujours le résultat d'une maximisation d'une espérance d'utilité. A notre connaissance, dans un cadre de décision séquentielle aucune justification de l'axiome de réduction des loteries n'a été fournie lorsque l'on utilise un modèle ne respectant pas l'axiome d'indépendance (qui découle directement de l'axiome de réduction des loteries composées, de l'axiome conséquentialiste et de l'axiome de cohérence dynamique (Hammond, 1989)).

La remise en cause de la possibilité de réduire les loteries, ainsi que la remise en cause de l'axiome conséquentialiste, sont deux arguments en faveur d'une procédure de veto-process (Jaffray, 1999) lorsque l'on désire appliquer un critère de décision dans un cadre séquentiel. En effet, cette procédure relâche de manière implicite l'axiome conséquentialiste et l'axiome de réduction des loteries composées. En effet, chaque ego ne peut construire une stratégie qu'en complétant les sous-stratégies proposés par ses egos futurs. Ceci est précisément une manière de relâcher l'axiome conséquentialiste, puisque chaque ego prend sa décision non pas en fonction de la sous-stratégie optimale induite par son sous-arbre futur mais parmi toutes celles ne s'éloignant pas excessivement de l'optimale. De même, chaque loterie composée est validée « morceau par morceau » par chacun des egos intervenant dans la stratégie associée à cette loterie. Ainsi, les loteries composées associées aux stratégies qui on « survécues » jusqu'à l'ego racine, sont issus d'une validation en chaque instant où une décision est à prendre, et de manière implicite, l'ordre des événements est pris en compte dès lors qu'ils sont séparés par des décisions. On s'autorise ainsi à ne réduire que les loteries pour lesquelles chaque « morceau » a été validé et non toutes les loteries composées envisageables. Ceci constitue un relâchement (qui nous semble souhaitable) de l'axiome de réduction des loteries composées. Nous avons également vu que, d'un point de vue psychologique, une stratégie retournée par une procédure de veto-process était aisée à mettre en œuvre, ce qui est moins le cas pour une stratégie retournée par une méthode de choix résolu depuis la racine.

Ainsi, tant d'un point de vue décisionnel que d'un point de vue psychologique, le

principe de veto-process semble parfaitement justifié. De plus, d'un point de vue informatique, la mise en œuvre du principe de veto-process, proposée par Jaffray et Nielsen (2006), est bien plus efficace que la plupart des algorithmes de choix résolu depuis la racine. Ceci constitue un argument de poids dans son adoption lorsque l'on désire appliquer un critère de décision dans un problème de décision séquentielle de très grande taille.

Ce constat ne remet cependant pas en cause l'utilité des procédures permettant de réaliser un choix résolu depuis la racine, puisque dans de nombreux cas en informatique, l'application d'une telle stratégie ne pose pas de problème, notamment dans tous les cas où elle ne sera pas mise en œuvre par un « humain » mais par une machine. De plus, de telles stratégies permettent de fournir d'excellentes heuristiques lorsque l'on cherche à fournir au décideur plusieurs « bonnes » stratégies afin d'élaguer toutes celles s'éloignant trop des préférences du décideur (i.e., dont la valeur s'éloigne trop de la valeur optimale).

### Perspectives de recherche

**Compacité d'une stratégie.** Un problème très peu étudié dans la littérature reste celui de la représentation compacte d'une stratégie. Dans le chapitre 4, nous évoquons brièvement la possibilité de factoriser des stratégies lorsque l'on respecte l'axiome conséquentialiste. Cependant, même les stratégies factorisées sont parfois de très grande taille. Le problème se pose dès lors qu'il est nécessaire d'explicitier chaque règle de décision appartenant à la stratégie. Un moyen commode de contourner le problème, est de se focaliser sur le développement d'algorithmes incrémentaux, procédant « à la volée » pour déterminer la décision requise lorsque l'on tombe dans un état donné du problème. On évite ainsi l'écueil de devoir stocker une stratégie complète en mémoire.

Par exemple, l'optimisation du critère de Hurwicz ne prend en compte que la valeur de la pire conséquence et la valeur de la meilleure conséquence pour évaluer une stratégie. Par conséquent, il suffit de connaître la pire conséquence et de la meilleure conséquence (conséquences extrêmes) pour caractériser une classe de stratégies ayant la même évaluation selon le critère de Hurwicz. Pour déterminer les conséquences extrêmes caractérisant la classe de stratégies optimisant le critère de Hurwicz, nous pensons ici qu'il est possible d'élaborer un modèle graphique dédié s'appuyant sur la programmation dynamique biobjectif. Après avoir calculé ces conséquences extrêmes, une seconde procédure devrait être proposée, exploitant cette information pour calculer dynamiquement les règles de décision à l'instant où elles sont requises. En d'autres termes, on déduit la règle de décision à l'instant  $t$  seulement au moment où on en a besoin sans jamais conserver les autres règles de décision.

**Cas de l'ambiguïté.** Dans le chapitre 1, nous avons vu une situation d'incertitude, dans le cadre du paradoxe d'Ellsberg (voir exemple 6 page 33), assez intéressante. Ce contexte d'incertitude est connu sous le nom de l'*ambiguïté* dans la littérature. Plus précisément, elle traite des situations dans lesquelles l'incertitude est définie sur des sous-ensembles d'événements non nécessairement disjoints ou élémentaires. De nombreux cas d'applications rentrent dans le cadre de cette généralisation. Par exemple, pour une croyance sur la localisation d'une cible sur une carte, on peut avoir des croyances qui se décrivent comme suit :

- la probabilité que la cible se trouve dans un rayon de 1 kilomètre autour du point  $A$  est comprise dans l'intervalle  $[a_1, b_1]$  avec  $0 \leq a_1 \leq b_1 \leq 1$ ,
- la probabilité que la cible se trouve dans un rayon de 2 kilomètres autour du point  $A$  est comprise dans l'intervalle  $[a_2, b_2]$  avec  $0 \leq a_2 \leq a_1 \leq b_1 \leq b_2 \leq 1$ ,
- ...,
- la probabilité que la cible se trouve dans un rayon de  $n$  kilomètres autour du point  $A$  est comprise dans l'intervalle  $[a_n, b_n]$  avec  $0 \leq a_n \leq \dots \leq a_2 \leq a_1 \leq b_1 \leq b_2 \leq \dots \leq b_n \leq 1$ .

D'autres formes remarquables de croyances, fréquemment rencontrées dans les problèmes de décision, ne peuvent être abordées dans le cadre du risque imprécis que nous avons traité dans cette thèse. Une perspective de recherche intéressante consiste à adapter les algorithmes vus dans le cadre du risque imprécis (voir chapitre 5) pour les étendre à des problèmes de décision séquentielle sous ambiguïté.

**Approche par cographes.** Enfin, il est légitime de se demander si les représentations graphiques actuelles d'un problème de décision séquentielle sont adéquates pour leur résolution. Autant ils permettent une représentation attrayante pour le décideur, autant ils se peut qu'ils soient inadaptés pour la réalisation des calculs lors de l'optimisation d'un critère de décision donné. Ainsi, il nous semble intéressant de modifier la représentation du problème pour mieux le résoudre par la suite. Par exemple, une façon d'appréhender un problème de décision séquentielle est d'identifier un ensemble de conséquences compatibles permettant d'optimiser un critère de décision donné, et de calculer ensuite les décisions qui mènent à ces conséquences. A titre d'exemple, considérons le graphe construit de la manière suivante. Soit  $U = \{u_1, u_2, \dots, u_n\}$  (avec  $u_1 < u_2 < \dots < u_n$ ) l'ensemble des utilités distinctes associées aux différentes conséquences et  $E = \{e_1, e_2, \dots, e_n\}$  l'ensemble des événements élémentaires tels que l'événement  $e_i$  permette d'atteindre la conséquence  $u_i$  (sous réserve de prendre les décisions adéquates). Pour chaque valeur d'utilité  $u \in U$ , on crée un sommet. Pour chaque paire de sommets, on crée une arête si et seulement si les deux valeurs d'utilités représentées par les sommets apparaissent ensemble dans au moins une stratégie (une condition nécessaire et suffisante pour assurer

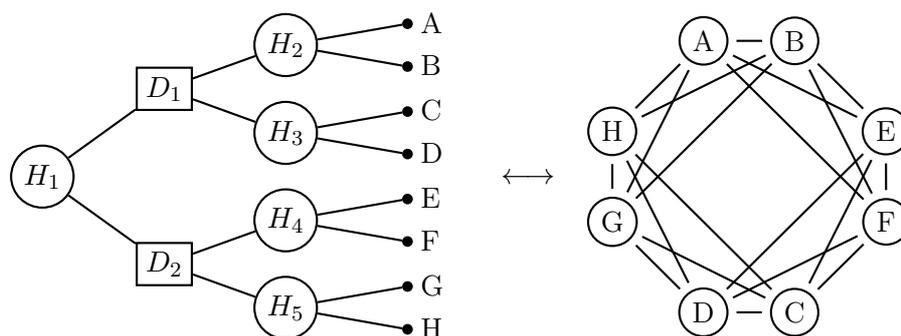


FIGURE 5.5 – Exemple de réduction d'un arbre de décision en cographe.

la coexistence de deux conséquences dans au moins une stratégie, est que leur ancêtre commun le plus proche soit un noeud de chance). Enfin, on associe à chaque sommet le couple  $(u_i, e_i)$  correspondant. Le graphe ainsi construit appartient à la famille des *cographe*s. L'ensemble des cographe

s est en bijection avec l'ensemble des arbres ayant des noeuds séries et parallèles (noeuds de chance et noeuds de décision dans notre cadre), et le passage d'une structure à l'autre se fait en temps linéaire. En guise d'illustration, nous représentons dans la figure 5.5 un arbre de décision et le cographe associé.

Dans un tel graphe, il existe une bijection entre les *cliques maximales*<sup>3</sup> et les stratégies, c'est-à-dire que la stratégie menant à l'ensemble des conséquences  $U$  correspond à la clique formée par les noeuds associés aux conséquences de  $U$  (par exemple, la stratégie  $\{(D_1, H_2), (D_2, H_4)\}$  correspond à la clique maximale  $\{A, B, E, F\}$ ). On évalue un ensemble de sommets à l'aide d'un critère  $V$  en construisant une loterie à partir des couples  $(u_i, e_i)$  associés à chaque noeud de l'ensemble de sommets, puis en évaluant cette dernière loterie selon le critère  $V$ . De cette manière, la recherche d'une stratégie optimisant un critère  $V$  donné se ramène à la recherche d'une clique maximale d'évaluation optimale selon le critère  $V$ . La représentation d'un problème de décision séquentielle sous cette forme permettrait de bénéficier des nombreux outils algorithmiques existants sur les graphes. Ceci nous semble être une piste de recherche intéressante pour des travaux futurs.

3. Une clique  $C$  d'un graphe  $\mathcal{G} = (\mathcal{N}, \mathcal{E})$  est une clique maximale si pour tout  $x \in \mathcal{N} \setminus C$ ,  $C \cup \{x\}$  n'est pas une clique.



# Bibliographie

- M. Allais (1953) *Le comportement de l'homme rationnel devant le risque : critique des postulats de l'école américaine*, pp. 503–546.
- M. Allais (1988) *The General Theory of Random Choices in Relation to the Invariant Cardinal Utility Function and the Specific Probability Function : The (Uq) Model, A General Overview*, Reidel Publishing Company.
- M. Allais (1997) “An outline of my main contributions to economic science”, *The American Economic Review*, tm. 87, no. 6, pp. 3–12.
- K. J. Arrow (1953) *Hurwicz's optimality criterion for decision-making under ignorance*, Technical report 6 - Department of Economics and Statistics - Stanford University.
- K. J. Arrow (1971) *Essays in the Theory of Risk-Bearing*, North-Holland Pub. Co.
- D. Bell (1985) “Disappointment in decision making under uncertainty”, *Operations Research*, tm. 33, pp. 1–27.
- R. Bellman (1957) “A markovian decision process”, *Journal of Mathematics and Mechanics*, tm. 6.
- D. Bouyssou et M. Pirlot (2008) “On some ordinal models for decision making under uncertainty”, *Annals of Operations Research*, tm. 163, no. 1, pp. 19–48.
- C. Camerer (1989) “An experimental test of several generalized utility theories”, *Journal of Risk and Uncertainty*, tm. 2, no. 1, pp. 61–104.
- C. Camerer (1992) *Recent Tests of Generalizations of Expected Utility Theory*, Kluwer Academic Publishers, Boston.
- A. Chateauneuf et M. Cohen (1994) “Risk-seeking with diminishing marginal utility in a non expected utility model”, *Journal of Risk and Uncertainty*, , no. 9, pp. 77–91.
- H. Chernoff (1954) “Rational selection of decision functions”, *Econometrica*, tm. 22, pp. 422–443.

- S. Chew (1983) "A generalization of the quasi linear mean with applications to the measurement of income inequality and decision theory resolving the allais paradox", *Econometrica*, tm. 51, pp. 1065–1092.
- S. Chew (1989) "Axiomatic utility theories with the betweenness property", *Annals of Operational Research*, tm. 19, pp. 273–298.
- S. Chew, E. Karni et Z. Safra (1987) "Risk aversion in the theory of expected utility with rank dependent probabilities", *Journal of Economic Theory*, tm. 42, pp. 370–380.
- S. Chew et K. McCrimmon (1979) *Alpha-Nu Choice Theory : A Generalization of Expected Utility Theory*, 669, Working Paper, University of British Columbia.
- M. Cohen, J.-Y. Jaffray et T. Said (1987) "Experimental comparison of individual behavior under risk and under uncertainty", *Organizational Behavior and Human Decision Processes*, tm. 39, pp. 1–22.
- L. de Campos, J. Huete et S. Moral (1994) "Probability intervals : a tool for uncertain reasoning", *I. J. of Uncertainty, Fuzziness and Knowledge-Based Systems*, , no. 2, pp. 167–196.
- B. de Finetti (1937) "La prévision : ses lois logiques, ses sources subjectives", *Annales de l'institut Henri Poincaré*, tm. 7, pp. 1–68.
- E. Dekel (1986) "An axiomatic characterization of preferences under uncertainty : Weakening the independence axiom", *Journal of Economic Theory*, tm. 40, pp. 304–318.
- A. Dempster (1986) "Upper and lower probabilities induced by a multivalued mapping", *The Annals of Mathematical Statistics*, tm. 38, no. 2, pp. 325–339.
- D. Denneberg (1997) *Non-additive measure and integral*, Kluwer Academic Publishers Dordrecht.
- S. Destercke (2008) *Représentation et combinaison d'informations incertaines : applications aux études de sûreté nucléaires*, Thèse de Doctorat de l'Université de Toulouse.
- D. Dubois, H. Fargier et P. Perny (2003) "Qualitative decision theory with preference relations and comparative uncertainty : an axiomatic approach", *Artificial Intelligence*, tm. 148, no. 1-2, pp. 219–260.
- D. Dubois et H. Prade (1985) *Théorie des possibilités*, Paris : Masson.
- D. Dubois et H. Prade (1988) *Possibility Theory - An Approach to Computerized Processing of Uncertainty*, New York : Plenum Press.

- D. Ellsberg (1961) "Risk, ambiguity and the savage axioms", *Quarterly Journal of Economics*, tm. 75, pp. 643–669.
- N. Etchart (2002) "Adequate moods for non-eu decision making in a sequential framework", *Theory and Decision*, tm. 52, no. 1, pp. 1–28.
- P. Fishburn (1978) "On handa's "new theory of cardinal utility" and the maximization of expected return", *Journal of Political Economy*, tm. 86, pp. 321–324.
- P. Fishburn (1982) *Dominance in SSB utility theory*, Mimeographed, Bell Laboratories.
- P. Fishburn (1983) "Transitive measurable utility", *Journal of Economic Theory*, tm. 31, pp. 293–317.
- L. Garcia et R. Sabbadin (2008) "Complexity results and algorithms for possibilistic influence diagrams", *Artificial Intelligence*, tm. 172, no. 8-9, pp. 1018–1044.
- J.-P. Gayant (1997) "Décroissance de l'utilité marginale et aversion probabiliste pour le risque : Une remise en cause de l'interprétation classique", *Revue d'économie politique*, tm. 107, no. 3, pp. 331–342.
- J.-P. Gayant (2001) *Risque et Décision*, Imprimerie France Quercy.
- F. Gul (1991) "A theory of disappointment aversion", *Econometrica*, tm. 59, no. 3, pp. 667–686.
- P. Hammond (1988) "Consequentialist foundations for expected utility", *Theory and Decision*, tm. 25, pp. 25–78.
- P. Hammond (1989) "Consistent plans, consequentialism, and expected utility", *Econometrica*, tm. 57, no. 6, pp. 1445–1449.
- J. Handa (1977) "Risk, probabilities and a new theory of cardinal utility", *Journal of Political Economics*, tm. 85, pp. 97–122.
- D. W. Harless (1992) "Predictions about indifference curves inside the unit triangle : A test of variants of expected utility theory", *Journal of Economic Behavior & Organization*, tm. 18, no. 3, pp. 391–414.
- R. Howard (1960) *Dynamic Programming and Markov Processes*, The M.I.T. Press.
- R. Howard et J. Matheson (1984) *Readings on the Principles and Applications of Decision Analysis Vol. II*, Menlo Park CA : Strategic Decisions Group.
- L. Hurwicz (1951) "Optimality criteria for decision making under ignorance", *Cowles Commission Discussion Paper, Statistics*, tm. 370.

- J.-Y. Jaffray (1989) “Généralisation du critère de l'utilité espérée aux choix dans l'incertain régulier”, *Recherche opérationnelle*, tm. 23, pp. 237–267.
- J.-Y. Jaffray (1999) “Rational decision making with imprecise probabilities”, *1st International Symposium on Imprecise Probability : Theories and Applications*, pp. 183–188.
- J.-Y. Jaffray et M. Jeleva (2007) “Information processing under imprecise risk with the hurwicz criterion”, *5th International Symposium on Imprecise Probability : Theories and Applications*, pp. 233–242.
- J.-Y. Jaffray et T. Nielsen (2006) “An operational approach to rational decision making based on rank dependent utility”, *European Journal of Operational Research*, tm. 169, no. 1, pp. 226–246.
- G. Jeantet (2009) “Optimisation du critère d'hurwicz pour les arbres de décision hasard en situation d'incertain total”, *7ème Manifestation de JEunes Chercheurs STIC*.
- G. Jeantet et O. Spanjaard (2008a) “Approche algorithmique de la recherche d'une stratégie rdu-optimale dans un arbre de décision”, *recueil des articles longs du 9ème Congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision*, pp. 79–94.
- G. Jeantet et O. Spanjaard (2008b) “Rank-dependent probability weighting in sequential decision problems under uncertainty”, *International Conference on Automated Planning and Scheduling*, pp. 148–155.
- G. Jeantet et O. Spanjaard (2009a) “Choix résolu et utilité espérée dépendant du rang dans les diagrammes d'influence”, *Cinquièmes Journées Francophones Modèles formels de l'interaction*, pp. 181–192.
- G. Jeantet et O. Spanjaard (2009b) “Optimizing the hurwicz criterion in decision trees with imprecise probabilities”, *1st International Conference on Algorithmic Decision Theory*, pp. 340–352.
- G. Jeantet et O. Spanjaard (2010) “Une approche de choix résolu au sens de jaffray dans les arbres de décision munis de probabilités imprécises”, *Onzième Congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision*, pp. 141–154.
- F. Jensen, F. V. Jensen et S. L. Dittmer (1994) “From influence diagrams to junction trees”, *10th Conference on Uncertainty in Artificial Intelligence*.
- O. Jokung-Nguena (1998) *Micro-économie de l'incertain*, Dunod.
- D. Kahneman et A. Tversky (1979) “Prospect theory : An analysis of decision under risk”, *Econometrica*, tm. 47, pp. 263–291.

- U. Karmarkar (1979) "Subjectively weighted utility and the allais paradox", *Organizational Behavior and Human Performance*, tm. 24, no. 1, pp. 67–72.
- E. Karni et D. Schmeidler (1991) *Utility theory with uncertainty*, tm. 4, pp. 1763–1831.
- J. M. Keynes (1937) "La théorie générale de l'emploi", *Quarterly Journal of Economics* - traduit par Nicolas Jabko dans la *Revue française d'économie*.
- D. Kikuti, F. G. Cozman et C. P. de Campos (2005) "Partially ordered preferences in decision trees : computing strategies with imprecision in probabilities", dans *IJCAI Workshop on Advances in Preference Handling*.
- F. Knight (1921) *Risk, uncertainty and profit*, Houghton Mifflin Company published.
- D. M. Kreps et E. L. Porteus (1979) "Dynamic choice theory and dynamic programming", *Econometrica*, tm. 47, pp. 91–100.
- V. Kuznetsov (1991) *Interval Statistical Models*, Moscow : Radio i Svyaz Publ.
- S. Lauritzen et D. Nilsson (2001) "Representing and solving decision problems with limited information", *Management Science*, tm. 47, pp. 1238–1251.
- I. Levi (1974) "On indeterminate probabilities", *Journal of Philosophy*, tm. 71, no. 13, pp. 391–418.
- I. Levi (1980) *The enterprise of knowledge : an essay on knowledge, credal probability, and chance*, Cambridge : MIT Press.
- G. Loomes et R. Sugden (1986) "Disappointment and dynamic consistency in choice under uncertainty", *Review of Economic Studies*, tm. 53, pp. 271–282.
- R. D. Luce et H. Raiffa (1957) *Games and decisions : introduction and critical survey*, John Wiley & Sons.
- M. J. Machina (1982) "'expected utility" analysis without the independence axiom", *Econometrica*, tm. 50, pp. 277–323.
- M. J. Machina (1989) "Dynamic consistency and non-expected utility models of choice under uncertainty", *Journal of Economic Literature*, tm. 27, pp. 1622–1668.
- E. McClennen (1990) *Rationality and Dynamic choice : Foundational Explorations*, Cambridge University Press.
- J. Milnor (1954) *Games Against Nature*, In R. M. Thrall and C. Coombs and R. Davis. John Wiley & Sons.

- T. Morin (1982) “Monotonicity and the principle of optimality”, *Journal of Mathematical Analysis and Applications*, tm. 86, pp. 665–674.
- J. Pearl (1985) “Bayesian network : A model of self-activated memory for evidential reasoning”, *Technical Report 850021 (R-43) - UCLA Computer Science Department Technical Report and in Cognitive Science Society - UC Irvine*.
- F. Perea et J. Puerto (2007) “Dynamic programming analysis of the tv game “who wants to be a millionaire ?””, *European Journal of Operational Research*, tm. 183, pp. 805–811.
- J. Quiggin (1982) “A theory of anticipated utility”, *Journal of Economic Behavior and Organization*, tm. 3, pp. 323–343.
- J. Quiggin (1993) *Generalized Expected Utility Theory : The Rank-Dependent Model*, Kluwer Academic Publishers.
- M. Rabin (2000) “Risk aversion and expected-utility theory : A calibration theorem”, *Econometrica*, tm. 68, no. 5, pp. 1281–1292.
- H. Raiffa (1968) *Decision Analysis : Introductory Lectures on Choices under Uncertainty*, Addison-Wesley.
- N. Robertson et P. D. Seymour (1984) “Graph minors iii : Planar tree-width”, *Journal of Combinatorial Theory*, tm. 36, pp. 49–64.
- M. Rothschild et J. Stiglitz (1970) “Increasing risk i : A definition”, *Journal of Economic Theory*, tm. 2, pp. 225–243.
- R. Sabbadin (1998) *Une approche ordinale de la décision dans l'incertain : Axiomatization, représentation logique et application à la décision séquentielle*, Thèse de doctorat, Université Paul Sabatier, Toulouse.
- L. J. Savage (1951) “The theory of statistical decision”, *Journal of the American Statistical Association*, tm. 46, pp. 56–67.
- L. J. Savage (1954) *The Foundations of Statistics*, New-York, John Willey and Sons.
- D. Schmeidler (1982) “Subjective expective utility without additivity”, *Forder Institute Working Paper*.
- D. Schmeidler (1986) “Integral representation without additivity”, *Proceedings of the American Mathematical Society*, tm. 97, pp. 255–261.
- D. Schmeidler (1989) “Subjective probability and expected utility without additivity”, *Econometrica*, tm. 57, pp. 571–587.

- F. Scott, V. Kreinovich, L. Ginzburg, D. Myers et K. Sentz (2003) *Constructing Probability Boxes and Dempster-Shafer Structures*, Sandia National Laboratories.
- T. Seidenfeld (2004) "A contrast between two decision rules for use with (convex) set of probabilities : G-maximin versus e-admissibility", *Synthèse*, tm. 140, pp. 69–88.
- R. Shachter (1986) "Evaluating influence diagrams", *Operations Research*, tm. 34, pp. 871–882.
- G. Shaffer (1976) *A Mathematical Theory of Evidence*, Princeton University Press.
- C. A. B. Smith (1961) "Consistency in statistical inference and decision", *Journal of the Royal Statistical Society*, tm. B, no. 23, pp. 1–37.
- C. Starmer (2000) "Developments in non-expected utility theory : The hunt for a descriptive theory of choice under risk", *Journal of Economic Literature*, tm. 38, pp. 332–382.
- M. K. Starr (1966) "Planning models", *Management Science*, tm. 13, no. 4, pp. B115–B141.
- A. Tversky et D. Kahneman (1992) "Advances in prospect theory : Cumulative representation of uncertainty", *Journal of Risk and Uncertainty*, tm. 5, pp. 297–323.
- J. von Neuman et O. Morgenstern (1947) *Theory of games and economic behaviour*, Princeton University Press, 2nd edition.
- P. Waker (1990) "Under stochastic dominance choquet expected utility and anticipated utility are identical", *Theory and Decision*, tm. 29, pp. 119–132.
- P. Walley (1991) *Statistical reasoning with imprecise probabilities*, Chapman and Hall.
- K. Weichselberger (1999) "The theory of interval-probability as a unifying concept for uncertainty", *1st International Symposium on Imprecise Probability : Theories and Applications*, pp. 387–396.
- P. M. Williams (2007) "Notes on conditional previsions", *International Journal of Approximate Reasoning*, tm. 44.
- M. Yaari (1987) "The dual theory of choice under risk", *Econometrica*, tm. 55, no. 1, pp. 95–115.
- L. Zadeh (1978) "Fuzzy sets as a basis for a theory of possibility", *Fuzzy Sets and Systems*, tm. 1, pp. 3–28.



# Index

- Adversaire du risque, 19
- Arbre de décision, 61
  - non-séparable, 147
  - séparable, 147
- Aversion
  - au risque, 18
  - faible au risque, 18
  - forte au risque, 18
- Axiome
  - Archimédien, 13, 15
  - cohérence dynamique, 70
  - comparabilité, 13, 14
  - conséquentialiste, 70
  - continuité, 15
  - indépendance, 13, 15
  - neutralité, 13
  - réduction des loteries composées, 13, 70
  - transitivité, 13, 14
- Capacité de Choquet, 35
- Choix résolu, 73
- Choquet expected utility, 35
- Cohérence dynamique, 69
- Conséquentialisme, 69
- Diagramme d'influence, 64, 113
- Dominance stochastique
  - au premier ordre, 16
  - au second ordre, 18
- E-admissible, 52
- Effet de certitude, 21
- Epreuve aléatoire, 8
- Espérance de gain, 11
- Etat de la nature, 8
- Evénement, 8
- Evénement élémentaire, 8
- Expected Utility, 13
- Fonction d'évaluation, 10
- Fonction d'utilité, 13
- Fonction décumulative, 16
- Fonction de valeur, 10
- Incertain, 8–11
- Joueur, 19
- Loterie, 9, 11
- Mean preserving spread, 18
- Modèle
  - $\Gamma$ -Maximax
    - Incertude totale, 38
    - Risque imprécis, 50
  - $\Gamma$ -Maximin
    - Incertude totale, 38
    - Risque imprécis, 50
  - CEU, 35
  - E, 11
  - EU, 13
  - Hurwicz
    - Incertitude totale, 43
    - Risque imprécis, 51
  - Laplace, 41
  - Minimax-regret, 39
  - RDU, 24

- 
- SE, 32
  - SEU, 33
  
  - Paradoxe
    - Allais, 20
    - Ellsberg, 33
  - Pré-ordre total, 14
  - Principe de la chose sûre, 27
  - Principe de la chose sûre comonotone, 27
  - Probabilités intervalles, 48
  - Problème de décision séquentielle, 59
  - Propriété de monotonie, 68
  
  - Règle de décision, 118
  - Rank dependent utility, 24
  - Relation de préférence, 9
  
  - Séparabilité, 146
  - Stratégie, 59, 62, 118
  - Stratégie mixte, 103
  - Stratégie partielle, 91
  - Stratégie pure, 103
  - Subjective expectation, 32
  - Subjective expected utility, 33
  
  - Utilité espérée, 13
  - Utilité espérée dépendant du rang, 24
  
  - Veto-process, 74



Vu :  
Le Président  
M.

Vu :  
Les Suffragants  
MM.

Vu et permis d'imprimer :  
Le Vice-Président du Conseil Scientifique chargé de la Recherche de l'Université Pierre  
et Marie Curie (PARIS VI)





### *Résumé*

Dans cette thèse, nous nous intéressons aux problèmes de décision séquentielle dans l'incertain. Ces problèmes concernent les situations d'incertitude dans lesquelles le décideur est amené à prendre plusieurs décisions étalées dans le temps (i.e., établir une *stratégie*). Cette problématique est bien connue en intelligence artificielle, sous le nom de planification dans l'incertain, et est très étudiée du fait de ses applications multiples dans de nombreux domaines (diagnostic médical, joueurs artificiels, pilotage automatique, gestion de stocks...). La communauté des économistes a fourni de nombreux critères de décision permettant de raisonner dans l'incertitude afin de pouvoir comparer des stratégies entre elles. Cependant, les difficultés liées à leur mise en œuvre conduisent en pratique à utiliser des critères moins performants dans les problèmes de décision séquentielle. L'utilisation des critères performants est en effet freinée par l'absence d'algorithmes efficaces dans la littérature informatique. L'objet de cette thèse est justement de repousser ces verrous algorithmiques en proposant des algorithmes permettant leur application dans des problèmes de décision séquentielle.

*Mots-clés* : Aide à la décision, décision séquentielle, planification, incertain, risque, arbres de décision, diagrammes d'influence, optimisation combinatoire, espérance d'utilité, utilité espérée dépendant du rang, Hurwicz.

### *Abstract*

In this thesis, we are interested in sequential decision problems under uncertainty. These problems concern situations of uncertainty where the decision maker has to make several decisions spread over time (i.e., establish a *strategy*). This problem is much studied in artificial intelligence, known as planning under uncertainty, because of its several applications in many fields (medical diagnosis, artificial players, autopilot, inventory management, ...). The economist community has provided many decision criteria for reasoning under uncertainty in order to compare strategies. However, the difficulties associated with their implementation leads in practice to use criteria less efficient in sequential decision problems. The use of performant criteria is indeed hindered by the lack of efficient algorithms in the computer science literature. The purpose of this thesis is precisely to tackle these algorithmic locks by providing algorithms for optimizing these criteria in sequential decision problems.

*Keywords* : Decision aiding, sequential decision problems, planification, uncertainty, risk, decision trees, influence diagrams, combinatorial optimization, expected utility, rank dependent expected utility, Hurwicz.