

Centre de Recherche en Informatique Signal et Automatique de Lille  
Université Lille 1

Thèse en vue de l'obtention du titre de  
Docteur en Informatique de l'Université Lille 1

# Dialoguer pour décider

**Recommandation experte proactive et prise de décision  
multi-agents équitable**

**Fabien DELECROIX**

Soutenue le 23 mars 2015

**Jury :**

*Rapporteurs :*

Flavien BALBO

Professeur à l'École des mines de Saint-Étienne

Nicolas SABOURET

Professeur à Supélec

*Examineurs :*

Wassila OUERDANE

Maître de Conférences à l'École Centrale Paris

Jean-Stéphane VARRÉ

Professeur à l'Université Lille 1

*Directeurs :*

Maxime MORGE

Maître de Conférences à l'Université Lille 1

Jean-Christophe ROUTIER

Professeur à l'Université Lille 1



# Remerciements

---

Comme dans cette thèse il est question de décision, je ne résiste pas à définir le remerciement relativement à cette idée. Remercier, c'est prendre la décision de montrer de la gratitude à l'autre pour sa ou ses propres décisions. Si cet acte de langage ne coûte pas de ressource matérielle à celui qui l'exprime et n'en rapporte pas à celui qui le reçoit, sa valeur symbolique est pour moi loin d'être négligeable. C'est pourquoi je compte bien profiter de cet espace pour remercier certaines décisions et à travers elles leurs auteurs. Ceci étant, une thèse est un document public et j'aime que certaines choses ne le soient pas. Je décide donc de ne pas remercier ici directement mes proches dont l'aide, sous les formes les plus diverses, m'a été extrêmement précieuse et dans son ensemble essentielle. D'autant qu'il me semble que si je le faisais vraiment, le document pourrait doubler de volume. Je ne les oublie pas pour autant, tout au contraire.

Mes premiers remerciements nominatifs vont à Jean-Christophe Routier et Maxime Morge pour avoir choisi d'encadrer cette thèse et joué un rôle majeur dans sa fabrication. J'exprime ma gratitude en particulier pour leur disponibilité irréprochable, une pédagogie et une patience à toute épreuve ainsi qu'un niveau scientifique impressionnant. J'adresse ensuite des remerciements à Flavien Balbo et Nicolas Sabouret pour avoir accepté de rapporter ma thèse. Me savoir lu par vous est gratifiant et j'apprécie le contenu, le détail et le style de chacun de vos rapports. Je souhaite aussi remercier Wassila Ouerdane et Jean-Stéphane Varré d'avoir accepté d'examiner mes travaux et de faire partie de mon jury.

Un cliché répandu du thésard est celui d'un ermite solitaire plongé dans son ordinateur et des piles de papiers. S'il est vrai qu'une partie de l'activité de recherche se fait de manière solitaire et que certains spécimens sont fidèles à ce cliché, c'est loin d'être le cas me concernant. Au-delà de mes encadrants, une partie de l'activité de recherche a été réalisée en coopération avec de nombreuses personnes, en particulier sur le plan applicatif. Parmi elles, je souhaite remercier Adrien Nouveau et Isabelle Pialaprat pour leur contribution directe au prototype du Vendeur Virtuel Ubiquitaire (cf. chapitre 8) sur le plan de l'ingénierie mais aussi sur celui de la réflexion. Sur ce même projet, je remercie spécialement Bruno Beaufiles pour son rôle de coordinateur mais aussi Laurent Deslandres, Muriel Lauvergne, Ingrid Poncin et Slim Ben Mimoun. Concernant la troisième partie de cette thèse, l'implémentation du démonstrateur logiciel pour le problème de recherche d'un point de rencontre dans un labyrinthe (cf. chapitre 11) a été réalisée par Thomas Nachtergaele, que je remercie vivement.

L'environnement principal dans lequel cette thèse a été concoctée est celui de l'équipe Système Multi-Agents et Comportements (SMAC). Je tiens à remercier les agents qui, comme moi, y ont été situés et avec lesquels j'y ai eu des interactions bénéfiques. Chronologiquement, je remercie Antoine Nongaillard, Jean-Christophe Routier, Maxime Morge, Bruno Beaufiles, Iryna Veryzhenko, David Panzoli, Yoann Kubera, Tony Dujardin, Marc-Antoine Dupré dit Marcus, Philippe Matthieu, Yann Secq, Sébastien Picault, Jean-Paul Delahaye, Patricia Everaere, Adrien Nouveau, Isabelle Pialaprat, Matthis Gaciarz, Lisa Rougetet, Guillaume Dauster, Éric Piette et Guillaume Dubuisson Duplessis.

Ayant été hébergé dans le bureau 14A du M3, je tiens aussi à remercier mes voisines d'en face secrétaires pédagogiques pour leur bonne humeur et les services qu'elles m'ont rendus : Patricia Caron dite Patou, Aline Peers, Marie-Claire Vagelli, Isabelle Delaire et la charmante Brunette Mahiddine ;-).

J'ai suivi l'ensemble de mes études supérieures à l'Université Lille 1, pour une très large part au sein des Formations en Informatique de Lille 1 (FIL), où j'ai pu bénéficier d'enseignements de qualité. Nombreux sont les noms qui me viennent en tête à cet instant, je tiens à remercier en particulier Éric Wegrzynowski et Jean-Christophe Routier pour les connaissances fondamentales qu'ils m'ont transmises et m'ont ensuite permis de transmettre à mon tour.

Je n'oublie pas de remercier parmi les personnels de l'Université celles et ceux qui m'ont rendu service, que ce soit ponctuellement ou au quotidien. Je pense notamment à Christophe Mata, Sébastien Martelleur, Christelle Copin, Annie Marescaux, Michèle Mayer, Thi Nguyen et Bruno Boursier.

Enfin et peut-être surtout, je tiens à remercier les gens qui se sont impliqués bénévolement dans la vie universitaire. En particulier, un grand merci à Yoann Dufresne, Agathe Deswazière, Damien Riquet, Alexandre Gombert et Matthis Gaciarz pour leur rôle joué dans l'Association des Étudiants en Informatique. Merci et bravo à Alexandre Feugas, Julie Hamon, Guillaume Madelaine et Antoine Bertout pour ce qu'ils ont réalisé avec l'association de doctorants TILDA. Je termine en remerciant chaleureusement celles et ceux qui luttent pour défendre les intérêts matériels et moraux des étudiants logés en résidence universitaire du CROUS de Lille comme cela fut mon cas, en particulier merci à Camille Lihouck et Pacôme Hollestelle.

# Table des matières

---

<b>Introduction</b>	<b>1</b>
<b>Notations</b>	<b>7</b>
<b>I État de l'art</b>	<b>9</b>
<b>1 Décision et modélisation des préférences</b>	<b>11</b>
1.1 Introduction . . . . .	11
1.2 Choix . . . . .	12
1.2.1 Fonction de choix . . . . .	12
1.2.2 Rationalité . . . . .	13
1.3 Modèle de préférence ordinal . . . . .	14
1.4 Modèle de préférence cardinal . . . . .	18
1.5 Modèles de préférence multi-critères . . . . .	19
1.5.1 Décision multi-critères . . . . .	20
1.5.2 Préférences ordinales . . . . .	21
1.5.3 Préférences cardinales . . . . .	23
1.6 Synthèse . . . . .	24
<b>2 Communication entre les agents</b>	<b>27</b>
2.1 Introduction . . . . .	27
2.2 Langage . . . . .	28
2.2.1 Théorie des actes de langage . . . . .	28
2.2.2 Langage de communication d'agents . . . . .	29
2.3 Dialogue . . . . .	31
2.3.1 Typologie du dialogue . . . . .	31
2.3.2 Protocoles d'interaction . . . . .	33
2.3.3 Jeu de dialogue . . . . .	35
2.4 Synthèse . . . . .	36

---

<b>3</b>	<b>Prise de décision assistée</b>	<b>37</b>
3.1	Introduction	37
3.2	Méthodes et outils d'aide à la décision	37
3.3	Systèmes experts	39
3.4	Agents conversationnels	39
3.4.1	Perception et analyse des entrées de l'utilisateur	40
3.4.2	Expression de l'agent	40
3.4.3	Gestion du dialogue et raisonnement	41
3.5	Systèmes de recommandation	42
3.6	Synthèse	43
<b>4</b>	<b>Agrégation des préférences</b>	<b>45</b>
4.1	Introduction	45
4.2	Agrégation de préférences ordinales	45
4.2.1	Consensus	46
4.2.2	Préférences de Pareto	46
4.3	Agrégation de préférences cardinales	50
4.3.1	Bien-être utilitaire	50
4.3.2	Bien-être égalitaire	51
4.3.3	Bien-être de Nash	52
4.4	Synthèse	53
<b>5</b>	<b>Processus de prise de décision collective</b>	<b>55</b>
5.1	Introduction	55
5.2	Systèmes de vote	56
5.2.1	Critère de la majorité	56
5.2.2	Méthode Condorcet	57
5.2.3	Méthode Borda	58
5.2.4	Caractérisation des systèmes de vote	59
5.3	Négociation	60
5.3.1	Objet de négociation	61
5.3.2	Protocoles	61
5.3.3	Modèles décisionnels d'agents	62

5.3.4	Propriétés des protocoles de négociation . . . . .	63
5.4	Synthèse . . . . .	65
<b>6</b>	<b>Conclusions de l'état de l'art</b>	<b>67</b>
6.1	Vers un agent dialogique proactif . . . . .	67
6.1.1	Modèle de préférence du décideur . . . . .	67
6.1.2	Mode de dialogue homme-agent . . . . .	68
6.1.3	Processus d'assistance à la décision . . . . .	68
6.2	Vers un processus de décision collective équitable . . . . .	69
6.2.1	Modèle de préférences des participants . . . . .	69
6.2.2	Agrégation des préférences et évaluation sociale . . . . .	70
6.2.3	Processus de décision collective . . . . .	71
<b>II</b>	<b>Un agent dialogique proactif pour assister la prise de décision</b>	<b>73</b>
<b>7</b>	<b>Modèle d'agent dialogique proactif pour la recommandation experte</b>	<b>75</b>
7.1	Introduction . . . . .	75
7.2	Approche . . . . .	76
7.2.1	Problématique . . . . .	76
7.2.2	Caractéristiques de l'agent dialogique proactif . . . . .	77
7.3	Modèle de dialogue . . . . .	80
7.3.1	Approche du dialogue . . . . .	80
7.3.2	Jeu dialogique d'assistance à la décision . . . . .	82
7.4	Architecture d'agent . . . . .	84
7.4.1	Représentation des connaissances et raisonnement . . . . .	84
7.4.2	Gestion des buts et du dialogue . . . . .	86
7.5	Conclusion . . . . .	87
7.5.1	Synthèse . . . . .	87
7.5.2	Discussion . . . . .	88
7.5.3	Perspectives . . . . .	88
<b>8</b>	<b>Application : Vendeur Virtuel Ubiquitaire</b>	<b>91</b>
8.1	Introduction . . . . .	91
8.2	Problématique applicative . . . . .	92

8.2.1	Contexte et objectifs . . . . .	92
8.2.2	Analyse des mécanismes de vente . . . . .	93
8.3	Représentation des connaissances et raisonnement . . . . .	93
8.3.1	Produit . . . . .	93
8.3.2	Client . . . . .	94
8.3.3	Calcul du vecteur courant . . . . .	96
8.4	Gestions des buts, stratégie et tactique . . . . .	97
8.4.1	Identification du profil du client . . . . .	98
8.4.2	Identification du produit . . . . .	101
8.4.3	Comportement d'agent . . . . .	105
8.5	Implémentation . . . . .	105
8.6	Conclusion . . . . .	107
8.6.1	Synthèse . . . . .	107
8.6.2	Perspectives . . . . .	107
<b>III</b>	<b>Négociation multi-agents équitable</b>	<b>109</b>
<b>9</b>	<b>Système délibératif multilatéral pour un accord équitable</b>	<b>111</b>
9.1	Introduction . . . . .	111
9.2	Hypothèses sur les préférences . . . . .	112
9.3	Définition d'une alternative équitable . . . . .	112
9.3.1	Modélisation de l'effort . . . . .	113
9.3.2	Critère pour l'agrégation des efforts . . . . .	114
9.4	Résolution centralisée de la recherche des alternatives équitables . . . . .	117
9.5	Négociation multilatérale pour atteindre un accord équitable . . . . .	118
9.5.1	Protocole . . . . .	119
9.5.2	Stratégies . . . . .	121
9.6	Conclusion . . . . .	126
9.6.1	Synthèse . . . . .	126
9.6.2	Discussion . . . . .	126
9.6.3	Perspectives . . . . .	127
<b>10</b>	<b>Expérimentation du protocole de négociation pour un accord équitable</b>	<b>129</b>



---

10.1	Introduction . . . . .	129
10.2	Protocole expérimental . . . . .	129
10.2.1	Génération des préférences . . . . .	130
10.2.2	Méthode de comparaison des ensembles d'accord . . . . .	131
10.3	Résultats . . . . .	132
10.3.1	Gain à temporeriser . . . . .	132
10.3.2	Équité des accords . . . . .	134
10.3.3	Recours à l'arbitraire . . . . .	134
10.4	Conclusion . . . . .	136
10.4.1	Synthèse . . . . .	136
10.4.2	Perspectives . . . . .	136
<b>11</b>	<b>Application : Coopération pour la recherche d'un point de rencontre</b>	<b>139</b>
11.1	Introduction . . . . .	139
11.2	Problème . . . . .	139
11.3	Négociation pour la recherche d'un point de rencontre . . . . .	141
11.3.1	Modèle de préférences . . . . .	141
11.3.2	Recherche collective du point de rencontre optimal . . . . .	142
11.4	Démonstrateur logiciel . . . . .	144
11.5	Évaluation expérimentale . . . . .	145
11.5.1	Efficacité de la négociation . . . . .	147
11.5.2	Coût communicationnel . . . . .	147
11.5.3	Efficacité des tactiques . . . . .	149
11.6	Conclusion . . . . .	149
11.6.1	Synthèse . . . . .	149
11.6.2	Perspectives . . . . .	149
	<b>Conclusion</b>	<b>153</b>
<b>A</b>	<b>Éléments d'expertise dans le domaine de la vente</b>	<b>157</b>
A.1	Analyse des mécanismes de vente . . . . .	157
A.2	Données du VVU pour le domaine des APN . . . . .	159
	<b>Index</b>	<b>167</b>

**Bibliographie**

**169**

---

# Introduction

---

Toute la vie est une affaire de choix. Cela commence par : « la tétine ou le téton? » Et cela s'achève par : « Le chêne ou le sapin? »

---

Pierre Desproges

## Motivations

Décider, c'est choisir, sélectionner une alternative parmi un ensemble de possibilités. La décision la plus élémentaire consiste à choisir entre deux alternatives : « fromage ou dessert ? ». Dans le cas général, la décision peut s'opérer dans un ensemble d'alternatives plus vaste comme la carte d'un restaurant ou les plages horaires d'un agenda. Si la prise de décision peut être purement individuelle, ce processus peut aussi revêtir des aspects sociaux qui le rendent plus complexe. Ainsi, dans cette thèse, nous considérons des décisions impliquant au moins deux acteurs. Plus précisément, on s'intéresse à deux objets d'études : la prise de décision assistée et la prise de décision collective.

Par **prise de décision assistée**, on désigne le champ des décisions prises avec l'assistance d'un interlocuteur. Les acteurs ont alors des rôles distincts : décideur et assistant. Selon les applications, on cherche à satisfaire différents critères, que ce soit au niveau de la décision ou du processus décisionnel en lui-même. Avec l'essor du Web et en particulier du commerce en ligne, l'assistance à l'internaute devient un enjeu important et prend par exemple la forme d'agents conversationnels animés. Cependant, si l'apparence de ces assistants a considérablement été améliorée, leurs capacités dialogiques restent souvent limitées et décevantes.

La **prise de décision collective** considère une société d'acteurs qui prennent ensemble une décision. Il s'agit d'établir un processus permettant d'aboutir à un accord qui engage l'ensemble des acteurs participants. Ce champ traite notamment de deux grandes problématiques :

- *la coopération* où l'ensemble des acteurs poursuit un objectif commun et s'organise pour l'atteindre ;
- *la compétition* où chaque acteur a ses propres préférences individuelles et où il s'agit d'aboutir à un accord en dépit des divergences.

Le processus décisionnel peut alors prendre une forme plus ou moins interactive et relever soit de la *négociation*, soit du *système de vote*. Une problématique majeure de ce domaine est associée

à la notion de justice et vise à élaborer des processus de délibération collective aussi équitables que possible.

## Contexte scientifique

La prise de décision constitue un objet d'étude majeur dans le domaine de l'Intelligence Artificielle. Dans notre cadre, les acteurs sont confrontés à des problèmes de décisions dans lesquels sont impliqués d'autres acteurs. Or, les **S**ystèmes **M**ulti-**A**gents (SMA) représentent un paradigme de premier ordre pour l'analyse, la conception et l'implémentation de systèmes composés d'entités autonomes en interaction. Par conséquent, nous prenons le parti de considérer les acteurs comme de telles entités, c'est-à-dire comme des agents.

## Objectifs et contributions

Comme indiqué plus avant, nos travaux se situent d'une part dans le champ de la prise de décision assistée et d'autre part dans celui des modèles informatiques de prise de décision collective.

Concernant la prise de décision assistée, l'objectif est la modélisation d'un agent assistant permettant un dialogue cohérent et efficace. L'agent doit donc être crédible d'un point de vue conversationnel et pertinent sur le plan du conseil et de la recommandation. Pour répondre à ce double objectif, nous concevons un agent dialogique proactif : il dirige la conversation en posant les questions adéquates pour acquérir les préférences du décideur afin de lui recommander les alternatives les plus pertinentes. Notre proposition trouve une de ses applications dans le champ du *e-commerce* avec notre participation au projet PICOM<sup>1</sup> intitulé Vendeur Virtuel Ubiquitaire.

Concernant la décision collective, l'objectif est l'élaboration d'un processus de délibération collective compatible avec des préférences incomplètes et présentant un caractère équitable. Pour ce faire, nous définissons les accords équitables en appliquant le critère du *leximax* sur le rang des alternatives. Nous proposons ensuite un protocole de négociation permettant d'atteindre de tels accords et procédons à son évaluation au regard de la stratégie employée par les agents participant. Le protocole est ensuite appliqué sur un mode distribué au problème de recherche coopérative d'un point de rencontre dans un labyrinthe.

## Organisation du document

Cette thèse est structurée en trois parties principales. La première partie dresse un état de l'art de la décision individuelle, collective et assistée et des modes d'interaction entre agents. La deuxième partie présente un modèle d'agent dialogique proactif pour la décision assistée et l'applique au domaine du commerce électronique. La troisième partie propose un processus de décision collective pour atteindre un accord socialement équitable et l'illustre par la résolution distribuée d'un problème de rencontre dans un labyrinthe. Une vue d'ensemble des liens entre les chapitres décrits ci-après est proposée dans la figure 1.

---

1. Pôle de compétitivité Industries du Commerce

---

**Partie I : État de l'art.** Dans cette première partie, nous proposons une synthèse et une analyse de travaux réalisés dans le champ de la modélisation des préférences, du dialogue, de l'aide à la décision et de la négociation.

**Chapitre 1 : Décision et modélisation des préférences.** Dans l'ensemble des problématiques abordés dans cette thèse, la modélisation des préférences du ou des agents décideurs constitue une question préalable. En effet, on considère généralement qu'une décision est basée sur des préférences qui la fondent. Ainsi, plusieurs modèles de préférences existent et présentent des propriétés différentes. Nous présentons donc dans ce chapitre les modèles principaux et leurs caractéristiques.

**Chapitre 2 : Communication entre les agents.** Que la décision soit assistée ou collective, l'interaction entre les agents joue un rôle essentiel dans le processus. Pour que les agents dialoguent, il est nécessaire qu'ils partagent un langage et suivent un protocole.

**Chapitre 3 : Prise de décision assistée.** La prise de décision assistée fait l'objet d'approches nombreuses et diverses allant de l'aide à la décision aux systèmes de recommandation en passant par les systèmes experts et les agents conversationnels. Ces approches empruntent différentes hypothèses sur les connaissances de chacun des agents, leur mode d'interaction ainsi que leur mode de raisonnement.

**Chapitre 4 : Agrégation des préférences.** Agréger des préférences consiste à évaluer globalement des alternatives à partir de préférences locales ou individuelles. Nous nous intéressons d'une part, aux travaux relatifs à l'agrégation de préférences ordinales avec les préférences unanimes encore dites de Pareto, puis d'autre part, à l'agrégation de préférences cardinales avec la théorie du choix social.

**Chapitre 5 : Processus de prise de décision collective.** L'évaluation d'un processus de décision collective est une question centrale. Différents critères ont été envisagés pour évaluer d'une part le résultat et d'autre part le processus qui permet de l'atteindre. Nous caractérisons les différents mécanismes et les situons relativement à ces critères. Dans un cadre de coordination, le critère est clair et constitutif du problème à résoudre. Dans un cadre de recherche d'accord, un critère souvent retenu est celui de l'équité qui fait l'objet d'interprétations diverses.

**Chapitre 6 : Conclusions de l'état de l'art.** Ce chapitre est à la fois une synthèse de notre état de l'art et un pont avec nos contributions. Nous y tirons des conclusions à partir de notre réflexion sur les travaux existants et indiquons et justifions nos approches et partis pris pour les contributions présentées dans les chapitres suivants.

**Partie II : Un agent dialogique proactif pour assister la prise de décision.** Cette deuxième partie a vocation à présenter notre contribution sur la prise de décision assistée.

**Chapitre 7 : Modèle d'agent dialogique proactif pour la recommandation experte.** Notre contribution prend la forme d'un Agent Dialogique Proactif (ADP) [Delecroix *et al.* 2011] qui dirige le dialogue avec l'utilisateur humain afin de lui faire part de recommandations fondées sur son expertise et les informations acquises au cours du dialogue. Nous présentons les différentes couches de l'architecture de notre agent à savoir sa représentation des connaissances et son raisonnement, puis sa gestion du dialogue et des buts [Delecroix *et al.* 2012b].

**Chapitre 8 : Application : Vendeur Virtuel Ubiquitaire.** Un des domaines d'application de notre contribution est celui du commerce électronique. Notre proposition d'agent dialogique

proactif a été appliquée dans le cadre du projet Vendeur Virtuel Ubiquitaire labellisé par le Pôle de compétitivité des industries et du commerce (PICOM) de Lille. Ce projet vise à l'amélioration de l'expérience d'achat en ligne via l'intégration d'un agent virtuel guidant l'utilisateur dans sa recherche de produit [Delecroix *et al.* 2013a].

**Partie III : Négociation multi-agents équitable.** Dans cette troisième partie, nous présentons notre contribution sur la prise de décision collective.

**Chapitre 9 : Système délibératif multilatéral pour un accord équitable.** Notre contribution prend la forme d'un protocole équitable visant à une solution la plus satisfaisante possible pour l'ensemble des agents [Delecroix *et al.* 2013b]. Le critère que choisissons est celui du *lexi-max* sur le rang des alternatives. Deux stratégies accompagnent le protocole, l'une coopérative, permettant notamment la résolution distribuée de la recherche d'alternatives optimales ; l'autre, temporisatrice, dans laquelle chaque agent défend ses propres intérêts.

**Chapitre 10 : Expérimentation du protocole de négociation pour un accord équitable.** Ce chapitre propose une évaluation expérimentale de notre protocole [Delecroix *et al.* 2014b] ayant plusieurs objectifs : premièrement, établir les conditions dans lesquelles un agent égoïste à effectivement intérêt à appliquer la stratégie temporisatrice plutôt que la coopérative ; deuxièmement, étudier l'influence sur la qualité du résultat de l'emploi de la stratégie temporisatrice par tout ou partie des agents ; troisièmement, montrer que notre processus ne nécessite que rarement de recourir à l'arbitraire pour finaliser la décision.

**Chapitre 11 : Application : Coopération pour la recherche d'un point de rencontre.** Nous appliquons notre protocole avec la stratégie coopérative à la recherche d'un point de rencontre dans un labyrinthe méconnu des agents. En négociant de manière répétée, les agents établissent dynamiquement une solution efficace pour se rejoindre rapidement [Delecroix *et al.* 2014a].

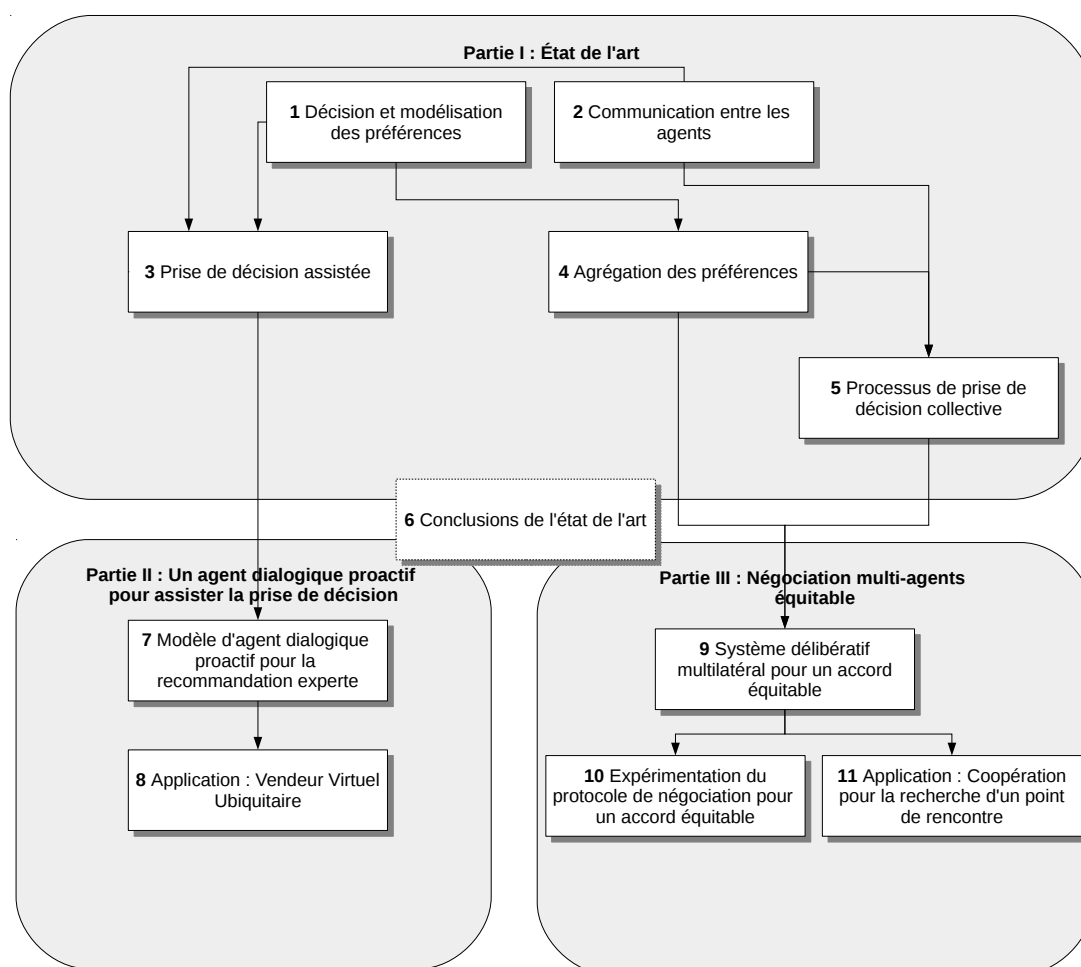


FIGURE 1 : Organisation du manuscrit





---

# Notations

---

Notation	Description	Page
$\mathcal{X}$	ensemble des alternatives	11
$\mathbf{C}^I$	concept	20
$\mathcal{X}_i$	attribut	20
$\Omega$	ensemble des agents	
$c$	fonction de choix	12
$c_{max}$	fonction de choix maximal	19
$c_{ND}$	fonction de choix des non-dominés	18
$c_{opt}$	fonction de choix optimal	12
$u_i(x)$	utilité de l'alternative $x$ pour l'agent $i$	18
$r(x, \mathcal{X}, \succsim_i)$	rang de l'alternative $x$ dans $\mathcal{X}$ pour l'agent $i$	113
$\bar{r}(x, \mathcal{X}, \Omega)$	rang moyen de l'alternative $x$ dans $\mathcal{X}$ pour les agents dans $\Omega$	113
$r_{max}(x, \mathcal{X}, \Omega)$	rang maximal de l'alternative $x$ dans $\mathcal{X}$ pour les agents dans $\Omega$	113
$\succ$	relation de préférence stricte	14
$\succsim$	relation de préférence large	15
$\sim$	relation d'indifférence	14
$\parallel$	relation d'incomparabilité	16
$\succsim^L$	relation de préférence lexicographique	22
$\succsim^P$	relation de préférence de Pareto	47
$\succ^{PN}$	relation de préférence nécessaire de Pareto	48
$\succ^{PN}$	relation de préférence potentielle de Pareto	49
$\succsim^{lm}$	relation de préférence <i>leximin</i>	51
$\succsim^{lex,r}$	relation de préférence <i>leximax</i>	116
$\succsim^U$	relation de préférence unanime	21
$\vec{r}_\Omega(x)$	vecteur décroissant des rangs de l'alternative $x$ pour les agents dans $\Omega$	116
$\vec{t}_\Omega(x)$	vecteur décroissant des tours auxquels l'alternative $x$ est proposée par les agents dans $\Omega$	120

<b>Notation</b>	<b>Description</b>	<b>Page</b>
$p_i(t)$	l'ensemble des alternatives proposées par l'agent $i$ au tour $t \geq 1$	120
$\mathcal{P}_i(t)$	l'ensemble des alternatives proposées par l'agent $i$ avant le tour $t \geq 1$	120
$\mathcal{S}$	profil de stratégies où l'agent	121
$C_\Omega$	profil de stratégies où tous les agents appliquent la stratégie coopérative	121
$T_\Omega$	profil de stratégies où tous les agents appliquent la stratégie temporisatrice	121
$C_{\Omega-i}$	profil de stratégies où tous les agents appliquent la stratégie coopérative sauf $i$ qui temporise	121
$T_{\Omega-i}$	profil de stratégies où tous les agents appliquent la stratégie temporisatrice sauf $i$ qui coopère	121
$\sigma_\Omega(\mathcal{X}, \mathcal{S})$	ensemble d'accords trouvé par les agents $\Omega$ en appliquant le profil $\mathcal{S}$	121
$Adj$	ensemble des paires de cellules adjacentes d'un labyrinthe	140
$M$	ensemble des murs d'un labyrinthe	140

Première partie

État de l'art



---

# Décision et modélisation des préférences

---

## Sommaire

---

<b>1.1</b>	<b>Introduction</b>	<b>11</b>
<b>1.2</b>	<b>Choix</b>	<b>12</b>
1.2.1	Fonction de choix	12
1.2.2	Rationalité	13
<b>1.3</b>	<b>Modèle de préférence ordinal</b>	<b>14</b>
<b>1.4</b>	<b>Modèle de préférence cardinal</b>	<b>18</b>
<b>1.5</b>	<b>Modèles de préférence multi-critères</b>	<b>19</b>
1.5.1	Décision multi-critères	20
1.5.2	Préférences ordinales	21
1.5.3	Préférences cardinales	23
<b>1.6</b>	<b>Synthèse</b>	<b>24</b>

---

## 1.1 Introduction

Ce chapitre a pour objet principal la modélisation des préférences et la prise de décision d'un agent. Il n'aborde que très peu les aspects sociaux qui font l'objet des chapitres suivants. Ici, nous considérons donc un unique agent, muni de préférences statiques et faisant face à un problème de décision défini par un ensemble d'alternatives, noté  $\mathcal{X}$ , tel que :

- $\mathcal{X}$  n'est pas l'ensemble vide, c'est-à-dire qu'il y a au moins une alternative ;
- $\mathcal{X}$  est un ensemble fini, c'est-à-dire que le nombre d'alternatives est borné ;
- les éléments de  $\mathcal{X}$  sont des alternatives mutuellement exclusives, c'est-à-dire que le choix d'une alternative exclut celui d'une autre.

Pour commencer, nous définissons formellement la notion de fonction de choix dans la section 1.2. Ensuite, nous présentons les deux principales approches de modélisation des préférences présents dans la littérature :

- la modélisation ordinaire, fondée sur un postulat qualitatif et articulée par une logique relationnelle, présentée dans la section 1.3 ;

- la modélisation cardinale, fondée sur un postulat quantitatif et s'appuyant sur des outils numériques, présentée dans la section 1.4.

Dans la section 1.5, nous considérons des problèmes de décision où interviennent plusieurs dimensions et raffinons les modèles de préférence pour exprimer des préférences multi-critères. Nous concluons ce chapitre en proposant dans la section 1.6 une analyse des différents modèles de préférence et des fonctions de choix associées.

## 1.2 Choix

Prendre une décision, c'est faire un choix, opérer une sélection parmi les alternatives. La littérature propose une définition formelle de la notion de choix sous la forme d'une fonction [Hansson & Grüne-Yanoff 2012].

### 1.2.1 Fonction de choix

On appelle fonction de choix toute fonction permettant de sélectionner un sous-ensemble non-vide d'alternatives au sein d'un ensemble donné.

#### Définition 1.1 (Fonction de choix).

Soit  $\mathcal{X}$  un ensemble fini et non vide d'éléments. On appelle **fonction de choix** toute fonction  $c : 2^{\mathcal{X}} \rightarrow 2^{\mathcal{X}}$  définie pour tout sous-ensemble  $\mathcal{Y} \subseteq \mathcal{X}$  avec  $\mathcal{Y} \neq \emptyset$  telle que :

1.  $c(\mathcal{Y}) \subseteq \mathcal{Y}$  ;
2.  $c(\mathcal{Y}) \neq \emptyset$ .

La première condition signifie que la ou les alternatives choisies doivent appartenir à l'ensemble auquel s'applique la fonction. La seconde condition indique la nécessité que le résultat de la fonction comporte au moins un élément pour qu'il y ait à proprement parler choix.

**Remarque 1.** De prime abord, on peut penser que la seconde condition ôte la possibilité de modéliser le refus. Imaginons par exemple que l'on vous propose un thé ou un café mais que vous ne souhaitez ni l'un ni l'autre. Dans ce cas de figure, il y a en fait une troisième alternative implicite dans la description du problème et on peut modéliser l'ensemble d'alternatives de ce problème de décision comme suit :  $\mathcal{X} = \{\text{the, cafe, refus}\}$ . Ainsi, le refus comme l'abstention peuvent constituer des alternatives à part entière.

Dans cette définition classique de la fonction de choix, on considère que le résultat de la fonction peut contenir plusieurs alternatives. Autrement dit, cela signifie que l'ensemble des alternatives choisies conviennent <sup>1</sup>.

La question du choix va généralement de pair avec celle de la rationalité, on trouve d'ailleurs les deux termes associés dans la *Théorie du choix rationnel*. La question qui nous intéresse dans la section qui suit est « Comment définir, au regard de ses choix, si un agent est rationnel ou non ? »

1. On note cependant que s'il y a eu choix, la décision n'est pour autant pas finalisée. Pour ce faire, il reste encore à sélectionner un unique élément dans cet ensemble, par exemple en procédant à un tirage aléatoire.

### 1.2.2 Rationalité

Dans cette section, nous étudions la notion de rationalité d'un agent et ses différentes interprétations dans le cadre décisionnel. La rationalité est abordée à la fois dans les sciences humaines – notamment en Économie, Psychologie et Sociologie – ainsi que dans l'Intelligence Artificielle (IA), la Théorie des Jeux et les Statistiques.

Dans le champ de l'IA, [Russell 1997] énonce qu'un agent est rationnel si « *ses actions sont sensées étant données l'information dont dispose l'agent et ses buts* ». Autrement dit, dans notre contexte, un agent est rationnel si ses actions sont issues d'un raisonnement en adéquation avec ses préférences. Ici, la rationalité consiste à choisir une alternative perçue comme étant la plus satisfaisante possible.

Ainsi, on peut caractériser les fonctions de choix par des propriétés de cohérence (cf. [Hansson & Grüne-Yanoff 2012, Sen 1977]). Dans la littérature, ces propriétés sont généralement appelées  $\alpha$ ,  $\beta$ , etc.

La propriété  $\alpha$  établit que si un élément est choisi dans un ensemble d'alternatives, il l'est aussi dans tout sous-ensemble auquel il appartient.

**Propriété 1.1** (Propriété  $\alpha$  (ou de Chernoff)). *Soit  $c$  une fonction de choix définie sur un ensemble d'alternatives  $\mathcal{X}$ . On dit que  $c$  respecte la propriété  $\alpha$  si et seulement si, pour tout sous-ensemble  $\mathcal{Y} \subseteq \mathcal{X}$ , on a  $[\mathcal{Y} \cap c(\mathcal{X})] \subseteq c(\mathcal{Y})$ .*

Cette propriété traduit l'indépendance vis-à-vis des alternatives non-pertinentes (souvent exprimée en anglais, *independence of irrelevant alternatives*). Elle est communément admise comme nécessaire à la rationalité d'un agent.

**Exemple 1.1 (Fonction de choix ne respectant pas la propriété  $\alpha$ ).**

*Considérons un décideur qui choisit l'alternative `refus` parmi l'ensemble  $\mathcal{X} = \{\text{the}, \text{cafe}, \text{refus}\}$  et l'alternative `cafe` parmi l'ensemble  $\mathcal{X}' = \{\text{cafe}, \text{refus}\}$  ( $c(\mathcal{X}') = \{\text{cafe}\}$ ). Une telle fonction de choix ne respecte pas la propriété  $\alpha$ . On remarque en effet que ces deux choix ne semblent pas cohérents, la seule absence de `the` des options disponibles ne permettant pas de justifier le choix de `refus` plutôt que `cafe`.*

La propriété  $\beta$  établit que si deux éléments sont choisis parmi un ensemble d'alternatives, la présence de l'un dans un sur-ensemble implique celle de l'autre.

**Propriété 1.2** (Propriété  $\beta$ ). *Soit  $c$  une fonction de choix définie sur un ensemble d'alternatives  $\mathcal{X}$ . On dit que  $c$  respecte la propriété  $\beta$  si et seulement si, pour tout sous-ensemble  $\mathcal{Y} \subseteq \mathcal{X}$ , pour toute paire d'alternatives  $\{x, y\} \subseteq c(\mathcal{Y})$ , si  $x \in c(\mathcal{X})$  alors  $y \in c(\mathcal{X})$ .*

**Exemple 1.2 (Fonction de choix ne respectant pas la propriété  $\beta$ ).**

*Considérons un décideur qui choisit les alternatives `the` et `cafe` parmi  $\mathcal{Y} = \{\text{the}, \text{cafe}, \text{refus}\}$  et les alternatives `cafe` et `chocolat` parmi  $\mathcal{X} = \{\text{chocolat}, \text{the}, \text{cafe}, \text{refus}\}$ . La fonction de choix employée ne respecte pas la propriété  $\beta$ . Au sens de cette propriété, ces deux choix sont incohérents, la seule présence d'une alternative supplémentaire (ici `chocolat`) dans les options disponibles ne constituant*

|| pas un critère rationnel pour disqualifier *the tout* en conservant *café*.

D'autres propriétés ont été envisagées pour modéliser la cohérence d'une fonction de choix, toutefois celles-ci ne font pas l'objet d'un consensus et nous choisissons de ne pas les présenter en détails ici.

Un agent rationnel vise à tirer la plus grande satisfaction des décisions qu'il prend. Pour ce faire, il s'agit pour lui de choisir les alternatives qu'il préfère. Ainsi, après avoir défini ce qu'était une fonction de choix, nous consacrons les sections suivantes aux préférences sur lesquelles vont se baser de telles fonctions.

### 1.3 Modèle de préférence ordinal

Dans l'approche ordinaire de modélisation des préférences, on considère que l'agent est capable de procéder à des comparaisons sur des paires d'alternatives. À l'issue d'une comparaison, on peut affirmer soit une préférence stricte pour l'une des deux alternatives, soit une indifférence entre les deux alternatives. Ces deux notions sémantiques peuvent être modélisées formellement par des relations binaires sur les alternatives et articulées dans une logique des préférences. Les premiers systèmes complets de logiques des préférences ont été proposés par [Halldén 1957] et [Von Wright 1963]<sup>2</sup>.

Différentes notations ont été employées dans la littérature pour désigner les relations de préférence. Dans ce document, nous choisissons de noter la préférence stricte par le symbole  $\succ$ .

**Notation 1.1** (Relation de préférence stricte). *Soit  $\mathcal{X}$  un ensemble d'alternatives. On appelle **relation de préférence stricte** la relation binaire  $\succ \subseteq \mathcal{X} \times \mathcal{X}$ . Si  $x \succ y$ , on dit que «  $x$  est meilleure que  $y$  » ou que « l'agent préfère strictement  $x$  à  $y$  ».*

La relation d'indifférence est quant à elle notée  $\sim$ .

**Notation 1.2** (Relation d'indifférence). *Soit  $\mathcal{X}$  un ensemble d'alternatives. On appelle **relation d'indifférence** la relation binaire  $\sim \subseteq \mathcal{X} \times \mathcal{X}$ . Si  $x \sim y$ , on dit que «  $x$  est aussi bonne que  $y$  » ou que « l'agent est indifférent entre  $x$  et  $y$  ».*

Ces deux relations forment un système généralement caractérisé par un ensemble d'axiomes :

$$\neg(x \succ y \wedge y \succ x) \text{ (asymétrie de } \succ) \tag{1.1}$$

$$x \sim y \Rightarrow y \sim x \text{ (symétrie de } \sim) \tag{1.2}$$

$$x \sim x \text{ (réflexivité de } \sim) \tag{1.3}$$

$$\neg(x \succ y \wedge x \sim y) \text{ (incompatibilité de } \sim \text{ et } \succ) \tag{1.4}$$

$$x \succ y \wedge y \succ z \Rightarrow x \succ z \text{ (transitivité de } \succ) \tag{1.5}$$

$$x \sim y \wedge y \sim z \Rightarrow x \sim z \text{ (transitivité } \sim) \tag{1.6}$$

---

2. Dans la littérature, on présente souvent en premier lieu la relation de préférence large, de laquelle on extrait ensuite les relations de préférence stricte et d'indifférence (voir par exemple [Dubus 2010]). Ici, le parti pris est d'opter pour une présentation davantage constructiviste pour mieux mettre en évidence les axiomes fondant la logique des préférences et les hypothèses sous-jacentes.



Les quatre premiers axiomes sont élémentaires pour fonder ce qu'est la préférence et l'indifférence et ne font l'objet d'aucune controverse. La transitivité des relations d'indifférence et de préférence stricte (axiomes 1.5 et 1.6) sont communément admises. Elles ont cependant été discutées, notamment par [Armstrong 1950] et [Luce 1956], qui posent le problème du saut qualitatif dans un ensemble d'alternatives continu. [Tversky 1969] étudie expérimentalement les préférences de décideurs humains et remarque qu'elles peuvent, sous des conditions spécifiques, être intransitives. En général, on considère cependant que la transitivité assure une cohérence dans les préférences de l'agent. Si l'agent préfère  $x$  à  $y$  et  $y$  à  $z$ , alors il préfère nécessairement  $x$  à  $z$ . De la même manière, si l'agent est indifférent entre  $x$  et  $y$  et entre  $y$  et  $z$ , alors il l'est aussi entre  $x$  et  $z$ . Par la suite, on suppose admis les axiomes 1.1 à 1.6.

On peut synthétiser ces deux relations de comparaison en une unique relation de préférence large.

**Définition 1.2 (Relation de préférence large).**

Soit  $\mathcal{X}$  un ensemble d'alternatives,  $\succ$  une relation de préférence stricte sur  $\mathcal{X}$  et  $\sim$ , une relation d'indifférence sur  $\mathcal{X}$ . On définit la **relation de préférence large** comme une relation binaire  $\succsim \subseteq \mathcal{X} \times \mathcal{X}$  telle que :

$$\forall (x, y) \in \mathcal{X}^2, x \succsim y \Leftrightarrow (x \succ y \vee x \sim y)$$

On dit que «  $x$  est au moins aussi bonne que  $y$  ».

Une relation de préférence large est un pré-ordre.

**Définition 1.3 (Pré-ordre).**

Soit  $R \subseteq E \times E$ , une relation binaire définie sur un ensemble  $E$ .  $R$  est un pré-ordre si et seulement si :

1.  $R$  est réflexive, c'est-à-dire  $\forall x \in E, xRx$  ;
2.  $R$  est transitive, c'est-à-dire  $\forall (x, y, z) \in E^3, xRy \text{ et } yRz \Rightarrow xRz$ .

**Propriété 1.3** (Une relation de préférence large est un pré-ordre).

**Démonstration 1.1** (Une relation de préférence large est un pré-ordre). *La réflexivité est obtenue par (1.3). La transitivité est obtenue par (1.5) et (1.6).*

Une relation binaire est dite totale si pour toute paire d'éléments, elle établit au moins une association entre eux deux.

**Définition 1.4 (Relation totale).**

Soit  $R$  une relation sur un ensemble  $\mathcal{X}$ ,  $R$  est une relation totale si et seulement si,  $\forall (x, y) \in \mathcal{X}^2$ , on a  $(xRy)$  ou  $(yRx)$ .

Lorsqu'une relation de préférence n'est pas totale, certaines alternatives ne sont pas comparables, et on dit alors que la relation de préférence est partielle. L'incomparabilité est elle aussi représentable sous la forme d'une relation binaire, déductible de la relation de préférence large.

**Définition 1.5 (Relation d'incomparabilité).**

Soit  $\mathcal{X}$  un ensemble d'alternatives et  $\succsim$  la relation de préférence large sur ces alternatives.

On appelle **relation d'incomparabilité** la relation binaire notée  $\parallel \subseteq \mathcal{X} \times \mathcal{X}$  et définie telle que :

$$\forall (x, y) \in \mathcal{X}^2, x \parallel y \Leftrightarrow \neg(x \succsim y) \text{ et } \neg(y \succsim x)$$

si  $x \parallel y$ , on dit que «  $x$  et  $y$  sont incomparables ».

On peut interpréter l'incomparabilité comme le refus du décideur de s'engager sur un jugement incertain. En effet, la notion d'incomparabilité se distingue de celle d'indifférence. D'un point de vue formel, on remarque que contrairement à la relation d'indifférence, la relation d'incomparabilité n'est pas transitive : deux alternatives peuvent être comparables à une troisième mais incomparables entre elles (voir 1.4). Ainsi, là où l'indifférence modélise une sémantique d'appréciation égale d'un agent pour deux alternatives distinctes, l'incomparabilité décrit l'incapacité du décideur à les comparer l'une à l'autre.

Comme le montre l'exemple qui suit, la modélisation de l'incomparabilité dans les préférences du décideur peut présenter une incompatibilité avec la propriété 1.2 dite propriété  $\beta$ .

**Exemple 1.3 (Incompatibilité entre propriété  $\beta$  et préférences partielles).**

Dans l'exemple 1.2, on considère un décideur qui choisit, d'une part,  $\{the, cafe\}$  parmi  $\{the, cafe, refus\}$  et d'autre part,  $\{cafe, chocolat\}$  parmi  $\{chocolat, the, cafe, refus\}$ .

Or, le décideur peut considérer, premièrement, que *the* et *cafe* sont incomparables ; deuxièmement, que *chocolat* et *cafe* sont incomparables ; troisièmement, qu'il préfère *chocolat* à *the*. Dans une telle situation, bien que ne respectant pas la propriété  $\beta$ , choisir  $\{chocolat, cafe\}$  semble rationnel.

Les préférences ordinales sont souvent représentées sous la forme d'un graphe orienté où chaque sommet correspond à une alternative.

**Définition 1.6 (Graphe de préférence).**

Soit  $(\mathcal{X}, \succsim)$  un ensemble d'alternatives fini non vide muni d'une relation de préférence large, on appelle **graphe de préférence** le graphe orienté  $G = (\mathcal{X}, A)$  tel que :  $A = \{(x, y) \in \mathcal{X}^2 \mid x \neq y, x \succsim y, \nexists z \in \mathcal{X} \setminus \{x, y\}, x \succsim z \wedge z \succsim y\}$ .

Dans un graphe de préférence tel que celui de l'exemple 1.4, le fait qu'une alternative  $x$  est au moins aussi bonne qu'une alternative  $y$  est représenté par un chemin depuis  $x$  vers  $y$ . Ainsi, l'ensemble des arcs du graphe correspond à la réduction réflexo-transitive de la relation de préférence.

**Exemple 1.4 (Préférences larges partielles d'un agent sur un ensemble de 7 alternatives).**

Nous considérons ici la relation de préférence d'un agent exprimée sur un ensemble d'alternatives  $\mathcal{X} = \{a, b, c, d, e, f, g\}$ . Le graphe de préférence de la figure 1.1 définit la relation de préférence large  $\succsim$  de l'agent décideur.

À la figure 1.1, correspondent les relations binaires suivantes :

- $\succ \equiv \{(a, b), (a, c), (a, d), (a, e), (a, f), (a, g), (b, f), (b, g), (c, d), (c, e), (c, g), (d, g), (e, g)\}$
- $\sim \equiv \{(d, e), (e, d)\}$
- $\succsim \equiv \succ \cup \sim$
- $\parallel \equiv \{(b, c), (b, d), (b, e), (c, b), (c, f), (d, b), (d, f), (e, b), (e, f), (f, c), (f, d), (f, e), (f, g), (g, f)\}$

On peut remarquer que la relation de préférence large est transitive :  $c \succsim e$  et  $e \succsim d \Rightarrow c \succsim d$ .

Ce n'est en revanche pas le cas de la relation d'incomparabilité :  $b \parallel c$  et  $c \parallel f \not\Rightarrow b \parallel f$ .

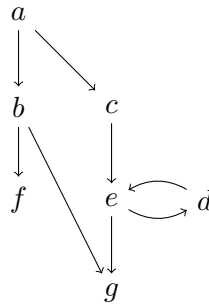


FIGURE 1.1 : Graphe de préférence correspondant à la relation  $\succsim$  d'un agent

Les propriétés des relations de comparaison sont récapitulées dans la table 1.1.

Relation de comparaison	Propriétés
préférence stricte $\succ$	asymétrique et transitive
indifférence $\sim$	réflexive, symétrique et transitive
préférence large $\succsim$	réflexive et transitive
incomparabilité $\parallel$	symétrique

TABEAU 1.1 : Propriétés des relations de comparaison

En résumé, avec une relation de préférence stricte uniquement, il n'est pas possible de modéliser l'équivalence de deux alternatives pour un agent. Pour y remédier, on peut, si on le souhaite, l'assortir d'une relation d'indifférence et donc utiliser une relation de préférence large. Enfin, si on veut enrichir davantage le modèle et exprimer l'incomparabilité entre deux alternatives, on peut s'autoriser à ne pas avoir une relation totale mais partielle.

Pour sélectionner les meilleures alternatives, on définit différentes fonctions de choix selon les hypothèses retenues sur les préférences.

Une alternative est dite optimale si elle est au moins aussi bonne que toute autre alternative.

#### Définition 1.7 (Optimalité).

Soit  $(\mathcal{X}, \succsim)$  un ensemble d'alternatives muni d'une relation de préférence large sur  $\mathcal{X}$ . On définit la fonction de choix optimal  $c_{opt}$  telle que  $c_{opt}(\mathcal{X}, \succsim) = \{x \in \mathcal{X} \mid \forall y \in \mathcal{X}, x \succsim y\}$ .

Si la relation de préférence est partielle, il n'existe pas nécessairement d'alternative optimale (on a alors  $c_{opt}(\mathcal{X}, \succsim) = \emptyset$ ). L'optimalité ne peut donc constituer une fonction de choix qu'avec

une relation de préférence totale. Sous cette condition, la fonction de choix optimal vérifie les propriétés  $\alpha$  et  $\beta$ . La notion de non-dominance, moins restrictive, permet de distinguer, dans un ensemble d'alternatives, celles auxquelles on ne préfère aucune autre.

**Définition 1.8 (Ensemble des non-dominés).**

Soit  $(\mathcal{X}, \succsim)$  un ensemble d'alternatives muni d'une relation de préférence large sur  $\mathcal{X}$ . La fonction de choix des **alternatives non-dominées** sur  $\mathcal{X}$  pour  $\succsim$  est notée  $c_{ND}(\mathcal{X}, \succsim)$  et définie telle que :

$$c_{ND}(\mathcal{X}, \succsim) = \{x \in \mathcal{X} \mid \forall y \in \mathcal{X}, \neg(y \succ x)\}$$

**Exemple 1.5 (Optimalité et non-dominance).**

Dans les préférences représentées dans la figure 1.1, on peut noter que :

- pour l'ensemble d'alternatives  $\mathcal{X}$  :
  - $a$  est optimale sur  $\mathcal{X}$  pour  $\succsim$ , écrit formellement,  $c_{opt}(\mathcal{X}, \succsim) = \{a\}$
  - et a fortiori  $a$  est non dominée,  $c_{ND}(\mathcal{X}, \succsim) = \{a\}$  ;
- pour l'ensemble d'alternatives  $\mathcal{Y} = \mathcal{X} \setminus \{a\}$  :
  - il n'y a pas d'alternative optimale,  $c_{opt}(\mathcal{Y}, \succsim) = \emptyset$
  - mais  $b$  et  $c$  sont non-dominées,  $c_{ND}(\mathcal{Y}, \succsim) = \{b, c\}$ .

De plus, on peut aussi remarquer que comme  $c_{ND}(\succsim, \{c, f\}) = \{c, f\}$  et  $c_{ND}(\succsim, \{b, c, f\}) = \{b, c\}$ . Ainsi, une fonction de choix non-dominée ne respecte pas nécessairement la propriété  $\beta$  (cf. propriété 1.2).

En conclusion, on remarque que l'optimalité ( $c_{opt}$ ) ne constitue pas nécessairement une fonction de choix pour une relation de préférence partielle. Cependant, la sélection de l'ensemble des non-dominés ( $c_{ND}$ ) constitue en toute hypothèse une fonction de choix car il existe toujours au moins une alternative non dominée. En effet, l'optimalité est plus restrictive que la non-dominance. Ainsi, toute alternative optimale est aussi non-dominée alors que l'inverse n'est pas nécessairement vrai.

## 1.4 Modèle de préférence cardinal

Contrairement à l'approche ordinale, l'approche cardinale modélise les penchants d'un agent sous la forme d'une fonction d'utilité associant à chaque alternative une valeur numérique. Nous faisons le choix de n'intégrer à notre étude ni le champ de la théorie de la décision dans le risque ni celui de l'utilité espérée de [Von Neumann & Morgenstern 1944]. Autrement dit, on ne considère pas que l'environnement est incertain et que différents événements ont une certaine probabilité de survenir. On peut envisager l'utilité comme une évaluation de la satisfaction attribuée par le décideur à l'alternative, cette approche est notamment formalisée dans [Fishburn 1970].

**Définition 1.9 (Fonction d'utilité).**

Soit  $\succsim$  une relation de préférence large totale sur  $\mathcal{X}$  un ensemble d'alternatives. On appelle fonction d'utilité toute fonction  $u : \mathcal{X} \rightarrow \mathbb{R}$  telle que  $\forall (x, y) \in \mathcal{X}^2, x \succsim y \Leftrightarrow u(x) \geq u(y)$ .

Considérons par exemple les préférences d'un décideur modélisées par la fonction d'utilité définie dans le tableau 1.2. On constate que l'utilité associée à l'alternative  $a$  est supérieure à celle associée à l'alternative  $b$ , ce qui signifie que  $a$  est préférable à  $b$  pour le décideur. De plus, on peut noter que les utilités de  $b$  et  $d$  sont égales : le décideur trouve les alternatives  $b$  et  $d$  aussi bonnes l'une que l'autre.

$\mathbf{x}$	$a$	$b$	$c$	$d$
$\mathbf{u}(\mathbf{x})$	4	2	1	2

TABLEAU 1.2 : Fonction d'utilité  $u$  définie sur les alternatives  $a, b, c, d$ .

Les valeurs numériques introduites par les fonctions d'utilité permettent diverses interprétations. Dans le chapitre 6 de [Fishburn 1970], la différence d'utilité indique l'intensité des préférences. En suivant cette hypothèse, étant donnée la fonction d'utilité définie dans le tableau 1.2, on peut par exemple affirmer que le degré de préférence de  $a$  sur  $b$  est deux fois supérieur à celui de  $b$  sur  $c$ . D'autres travaux excluent au contraire toute interprétation de degré. C'est le cas par exemple dans [Dubus 2010], où l'auteur souligne qu'« *il ne faut pas attacher de sémantique aux valeurs de la fonction d'utilité autre qu'une sémantique ordinale : les nombres ne servent qu'à se comparer entre eux et n'indiquent aucunement un niveau de préférence.* »

On retrouve notamment les fonctions d'utilité employées par des économistes néoclassiques (cf. [Walras 1874]) dans la théorie du consommateur, pour mesurer la satisfaction du décideur. Avec des valeurs numériques, l'alternative optimale pour le décideur est celle qui maximise l'utilité. Autrement dit, l'argument – au sens mathématique – du maximum rassemble les meilleures alternatives.

**Définition 1.10 (Utilité maximale).**

Soient  $\mathcal{X}$  un ensemble d'alternatives et  $u : \mathcal{X} \rightarrow \mathbb{R}$  une fonction d'utilité. La fonction de choix maximal  $c_{max}$  est définie telle que

$$c_{max}(\mathcal{X}, u) = \arg \max_{x \in \mathcal{X}} [u(x)] = \{x \in \mathcal{X} \mid \forall y \in \mathcal{X}, u(x) \geq u(y)\}.$$

On remarque qu'il existe toujours au moins une alternative qui maximise l'utilité, potentiellement plusieurs,  $c_{max}$  constitue donc toujours une fonction de choix qui vérifie  $\alpha$  et  $\beta$ .

## 1.5 Modèles de préférence multi-critères

Jusqu'ici, et quelque soit l'approche - ordinale ou cardinale, nous avons envisagé les préférences de l'agent sur des alternatives considérées comme atomiques. Dans cette section, nous étudions les préférences construites par un calcul opéré en amont sur d'autres données. Dans la littérature, on parle généralement de problèmes de *décision multi-critères*, où les alternatives sont comparées les unes aux autres à l'aune d'un ensemble de critères.

La section 1.5.1 définit ce qu'est un problème de décision impliquant plusieurs critères. Dans le cas général, aucune alternative ne sera optimale sur tous les critères à la fois, il s'agit alors d'agrèger les différents critères pour rendre tout ou partie des alternatives comparables. On

retrouve ainsi les deux types de modèles de préférence adaptés en conséquence : le modèle ordinal avec les préférences lexicographiques et les CP-nets, en section 1.5.2 ; le modèle cardinal avec les fonctions d'agrégation linéaires et non linéaires en section 1.5.3.

### 1.5.1 Décision multi-critères

Dans un problème de décision, un agent peut prendre en compte plusieurs paramètres pour comparer les alternatives et effectuer un choix. Typiquement, il arrive que lors d'une prise de décision, on souhaite minimiser un coût et maximiser la qualité. Dans la littérature, on retrouve cette problématique dans les champs proches de la prise de décision multi-critères (ou *Multiple-Criteria Decision-Making* souvent abrégée MCDM) et de la prise de décision multi-attributs (ou *Multiple-Attribute Utility-Theory* souvent abrégée MAUT). La distinction décision multi-critères et multi-attributs n'est pas toujours évidente ni unanime et comporte une part d'interprétation. Dans un problème de décision multi-attributs, ce sont les alternatives, et donc le problème en lui-même, qui est objectivement défini sur plusieurs attributs. Dans un problème de décision multi-critères, on considère que les critères représentent les désirs du décideur sur différentes facettes des alternatives. On peut généraliser en considérant des alternatives composites, définies dans un même espace multi-dimensionnel appelé concept, chaque dimension correspondant à un critère ou à un attribut selon l'approche.

**Définition 1.11 (Concept).**

Soit  $(\mathcal{X}_i)_{i \in I}$  une famille d'ensembles finis non-vides indicés dans  $I$ , on appelle **concept**  $C^I$  le produit cartésien  $C^I = \prod_{i \in I} \mathcal{X}_i$

Un concept est composé d'un ensemble d'attributs ayant chacun un domaine de définition. Ces domaines de définition peuvent être divers. Dans le cas le plus simple, un attribut peut prendre ses valeurs dans un ensemble binaire (par exemple, {oui, non}), plus généralement, il peut s'agir d'un ensemble discret (par exemple, {rouge, vert, bleu, jaune}) d'un intervalle (par exemple  $[[0, 20]]$ ) ou même d'un ensemble infini (par exemple  $\mathbb{N}$ ).

**Exemple 1.6 (Description du concept voyage).**

Illustrons la notion de concept avec le concept de voyage qu'on choisit de définir par :

- un mode de transport,  $Mode = \{avion, train, bus\}$  ;
- un temps de trajet en heure,  $Temps = [0, 100]$  ;
- un prix en euros,  $Prix = [0, 10000]$  ;
- une consommation de  $CO_2$  en kilogrammes  $CO_2 = [0, 1000]$ .

Formellement, on définit  $Voyage = Mode \times Temps \times Prix \times CO_2$ .

Étant donnée la définition de concept, une alternative est alors une *instance*, c'est-à-dire un objet typé qui prend ses valeurs dans les domaines de définition des attributs du concept.

**Définition 1.12 (Instance).**

Soit un concept  $C^I = \prod_{i \in I} \mathcal{X}_i$ , une instance  $x \in C^I$  est un vecteur de valeurs  $(x_1, \dots, x_n)$  où  $\forall i \in I, x_i \in \mathcal{X}_i$ .

**Exemple 1.7 (Instance vol Lille-Toulouse du concept voyage).**

Étant donné la définition du concept de voyage dans l'exemple 1.6, on définit l'instance  $v$  du vol Lille-Toulouse par  $v = (\text{avion}, 2, 70, 133)$ .

L'ensemble des alternatives  $\mathcal{X}^I$  composé d'instances du concept  $C^I$  forme un sous-ensemble du produit cartésien ( $\mathcal{X}^I \subseteq C^I$ ). Dit moins formellement, toute combinaison de valeurs ne constitue pas nécessairement une alternative. Avec un tel modèle, on peut considérer que les préférences d'un agent portent sur certains aspects des alternatives – la valeur d'un attribut, ou plus généralement un  $n$ -uplet de valeurs de différents attributs – et que c'est à partir de ces fragments de préférence, après agrégation, que sont construites les préférences entières sur les alternatives. La fonction de décision de l'agent peut ainsi être définie sur l'ensemble des ensembles d'instances imaginables – que celles-ci soient disponibles ou non – et donc, en particulier sur les alternatives du problème.

Que ce soit dans le champ de la décision multi-critères ou du choix social (cf. chapitre 4), la problématique de l'agrégation des préférences est centrale. D'un côté, il s'agit d'agréger des préférences affiliées à des critères, de l'autre à des individus. Une comparaison entre ces deux problématiques ainsi que les approches et résultats qui leurs sont associées est proposée dans [Bouyssou *et al.* 2009]. Après avoir défini un problème multi-critères, étudions maintenant quelles formes peuvent prendre les préférences.

**1.5.2 Préférences ordinales**

Dans une approche ordinale de modélisation de préférences multi-critères, on considère que chaque critère du décideur est représenté sous la forme d'une relation de préférence définie sur un ou plusieurs attributs. Le problème à résoudre est alors de synthétiser une relation de préférence globale à partir des relations de préférence locales afin de lui appliquer une fonction de choix et de prendre une décision.

Une règle d'agrégation prudente consiste à établir une préférence globale seulement si elle est avérée localement sur chaque attribut.

**Définition 1.13 (Préférences unanimes).**

Soient  $C^I = \prod_{i \in I} \mathcal{X}_i$  un concept et  $(\succsim_i)_{i \in I}$  une famille de préférences sur ses attributs. On définit la relation de **préférence unanime**  $\succsim^U$  telle que  $\forall (x, y) \in \mathcal{X}$  :

$$x \succsim^U y \Leftrightarrow \forall i \in I, x \succsim_i y$$

Cette même règle est souvent présentée comme règle de Pareto (cf. définition 4.2 du chapitre 5). On remarque que dans le cas général, la relation de préférence unanime n'est pas totale. En

effet, il suffit que deux alternatives soient meilleures l'une que l'autre sur un critère différent pour qu'aucune ne soit unanimement préférée à l'autre.

Si les préférences unanimes peuvent constituer une règle à respecter, elles ne sont généralement pas suffisantes pour permettre une décision car trop incomplètes. En effet, l'instance idéale qui rassemble les valeurs préférées de l'agent sur chaque attribut n'existe généralement pas dans la pratique. De nombreuses alternatives peuvent alors être non-dominées et incomparables au sens de l'unanimité. De plus, ces préférences ne permettent pas de prioriser certains critères par rapport à d'autres.

### Préférences lexicographiques

Afin de raffiner le modèle de préférence, on peut souhaiter hiérarchiser les différents critères en leur conférant une importance relative. Pour modéliser le fait qu'un critère est plus important qu'un autre, on peut, utiliser une relation de préférence, cette fois sur l'ensemble des attributs que l'on appellera par commodité *relation d'importance*. On enrichit donc notre modèle en lui ajoutant une relation de préférence définie sur les critères. Dès lors, il est possible d'appliquer la règle lexicographique qui est une généralisation de l'ordre du dictionnaire.

#### Définition 1.14 (Préférences lexicographiques).

Soient un concept  $C = \prod_{i \in I}$ , la relation d'importance stricte  $\succ_I$  et la famille des relations de préférence totales  $(\succsim_i)_{i \in I}$  sur chacun des attributs. La relation de préférence lexicographique sur les instances du concept est la relation de préférence  $\succsim^L$  définie telle que pour tout couple d'alternatives  $(x, y) \in \mathcal{X}^2$  :

- $(x_1, \dots, x_n) \succ^L (y_1, \dots, y_n) \Leftrightarrow \exists l \in I, x_l \succ_l y_l \text{ et } \forall k \succ_I l, (x_k \sim_k y_k)$
- $(x_1, \dots, x_n) \sim^L (y_1, \dots, y_n) \Leftrightarrow \forall k \in I, x_k \sim_k y_k$

Dans la définition ci-avant, on considère une relation d'importance stricte et des relations de préférence totales. Sous ces hypothèses, les préférences lexicographiques ainsi produites sont totales.

**Propriété 1.4** (Complétude de la relation de préférence lexicographique). *Si la relation de préférence sur l'ensemble des critères  $\succsim_I$  est stricte et totale et que les relations  $(\succsim_i)_{i \in I}$  sont totales, alors la relation de préférence lexicographique sur les instances  $\succsim_C$  est totale.*

Dans une relation de préférence lexicographique, on considère qu'une perte, même infinitésimale, sur un critère ne peut être compensée par un gain, même large, sur un critère moins important.

### CP-nets

Proposés par [Boutilier *et al.* 1999], les CP-nets ou *Conditional Preference Networks* sont un modèle de représentation graphique et compacte de préférences qualitatives. Cette approche se fonde sur l'hypothèse *ceteris paribus*<sup>3</sup> en considérant qu'il existe, au niveau des valeurs des

3. « Toutes choses étant égales par ailleurs ».



attributs (ici appelés variables), des *préférences indépendantes*, valables « toutes choses étant égales par ailleurs ». Ainsi, on représente les relations de préférence internes aux attributs. En outre, ce modèle rend possible la modélisation de *préférences conditionnelles*. Elle prennent la forme d'une *table conditionnelle de préférence* dans laquelle on exprime des préférences au sein d'un attribut mais cette fois-ci conditionnées par les valeurs prises par d'autres attributs dits variables parentes.

**Définition 1.15 (CP-net).**

Un CP-net sur un ensemble de variables  $V = \{\mathcal{X}_1, \dots, \mathcal{X}_n\}$  est un graphe orienté  $G$  sur  $V$ . Chaque nœud  $\mathcal{X}_i \in V$  est annoté par une table conditionnelle de préférence  $CPT(\mathcal{X}_i)$ . Une table  $CPT(\mathcal{X}_i)$  définit un ordre total sur les valeurs de  $\mathcal{X}_i$  pour chaque combinaison de valeurs des parents de  $\mathcal{X}_i$  notés  $Pa(\mathcal{X}_i)$ .

### 1.5.3 Préférences cardinales

La modélisation utilitaire de préférences multi-critères vise à établir une fonction d'utilité définie pour chaque élément de l'espace des instances. Cette question a notamment fait l'objet des travaux des chapitres 4 et 5 de [Fishburn 1970] et est présentée en détails dans [Keeney & Raiffa 1976].

#### Fonctions d'agrégation linéaires

Une première approche est de considérer que l'utilité d'une instance est égale à la somme des utilités de chaque valeur qui compose son vecteur, c'est-à-dire de proposer une fonction d'agrégation additive.

**Définition 1.16 (Fonction d'agrégation additive).**

Soit un concept  $C^I = \prod_{i \in I} \mathcal{X}_i$ . Soit  $(u_i)_{i \in I}$  un ensemble de fonctions d'utilité telles que  $u_i : \mathcal{X}_i \rightarrow \mathbb{R}$ . On définit la **fonction d'agrégation additive**  $S_I : \mathcal{X} \rightarrow \mathbb{R}$  telle que  $\forall x = (x_1, \dots, x_n) \in \mathcal{X}$  :

$$S_I(x) = \sum_{i \in I} u_i(x_i)$$

Avec une telle fonction d'agrégation, on fait l'hypothèse que chaque critère a une importance équivalente. On peut s'abstraire de cette hypothèse et généraliser davantage en pondérant les critères.

**Définition 1.17 (Fonction d'agrégation additive pondérée).**

Soit un concept  $C^I = \prod_{i \in I} \mathcal{X}_i$ . Soient  $(u_i)_{i \in I}$  un ensemble de fonctions d'utilité telles que  $u_i : \mathcal{X}_i \rightarrow \mathbb{R}$ . Soient  $(\lambda_i)_{i \in I}$  un ensemble de poids tels que  $\forall i, \lambda_i \in [0; 1]$ . On définit la **fonction**

d'agrégation additive pondérée  $SP_I : \mathcal{X} \rightarrow \mathbb{R}$  telle que  $\forall x = (x_1, \dots, x_n) \in \mathcal{X}$  :

$$SP_I(x) = \sum_{i \in I} u_i(x) \times \lambda_i$$

De nombreux autres modèles de préférences existent (pour un état de l'art, voir [Figueira *et al.* 2005]), nous choisissons de ne pas les présenter en détails ici mais nommons toutefois la moyenne pondérée ordonnée (souvent abrégé *OWA* en anglais) (voir [Yager 1988]), les intégrales de Choquet, les réseaux GAI (voir [Gonzales *et al.* 2007]) ou encore les UCP-net (voir [Boutilier *et al.* 2001]).

## 1.6 Synthèse

Dans ce premier chapitre consacré à la décision, nous avons formellement défini et caractérisé les fonctions de choix. Puis, nous avons étudié les deux approches de modélisation des préférences, ordinale et cardinale, ainsi que leurs extensions pour des problèmes de décision multi-critères.

Les approches ordinales et cardinales de modélisation des préférences ont chacune leurs spécificités. Une relation de préférence stricte totale est un ordre strict : elle est symétrique et transitive et correspond à un classement de l'ensemble des alternatives sans ex-æquo. Une relation de préférence large totale est un pré-ordre : elle est réflexive et transitive et permet de modéliser l'indifférence du décideur entre deux alternatives pour le décideur. Dans le cas où la relation de préférence est partielle, l'absence de préférence ou d'indifférence entre deux alternatives dénote leur caractère incomparable pour le décideur.

Dans [Fishburn 1970], l'auteur définit le modèle de préférence cardinal relativement au modèle ordinal. Il établit que toute relation de préférence large totale et, *a fortiori*, stricte totale, est représentable sous la forme d'une fonction d'utilité. Réciproquement, on peut représenter numériquement un pré-ordre total  $\succsim$  par une infinité de fonctions d'utilité. La condition nécessaire et suffisante à remplir est  $\forall (x, y) \in \mathcal{X}^2, x \succsim y \Leftrightarrow u(x) \geq u(y)$ .

Comme une fonction d'utilité (cf. définition 1.9) permet d'obtenir une relation de préférence totale (cf. définition 1.4), elle en revêt l'ensemble des propriétés. Cependant, une fonction d'utilité rend obligatoire la comparaison entre toutes les alternatives et ne permet donc pas de modéliser l'incomparabilité contrairement aux relations de préférence partielles. En bref, le modèle de préférence ordinal subsume le modèle de préférence cardinal.

Par ailleurs, on note que l'utilité semble indiquer un degré de préférence. La pertinence de cette dernière sémantique et son fondement dans un cadre décisionnel restent néanmoins discutables et controversés. Plus généralement, le caractère numérique de l'utilité ouvre à de nombreuses opérations mathématiques à la sémantique indéfinie et pouvant faire l'objet d'interprétations erronées dans une optique comparative. Enfin, avec une valeur numérique, on peut s'abstraire des autres alternatives et faire référence à une échelle pour évaluer une alternative et par exemple sa distance à un idéal qui n'est pas nécessairement une alternative disponible. On sort alors de l'évaluation comparative inhérente à la notion de préférence et on se situe davantage dans une évaluation intrinsèque des alternatives et de la satisfaction qu'elles peuvent apporter au décideur. Le tableau 1.3 résume les sémantiques exprimables par chaque modèle de préférence.

Sémantiques → Modèles ↓	indifférence	incomparabilité	intensité
stricte totale	✗	✗	✗
large totale	✓	✗	✗
stricte partielle	✗	✓	✗
large partielle	✓	✓	✗
fonction d'utilité	✓	✗	✓

TABLEAU 1.3 : Sémantiques possibles pour les différents modèles de préférence

Avec une relation de préférence stricte totale, il est facile de proposer une fonction de choix respectant les propriétés  $\alpha$ ,  $\beta$  et garantissant l'unicité du résultat. Sans le caractère strict des préférences, des ex-æquo sont possibles et on perd l'unicité du résultat, tout comme avec une fonction d'utilité. Enfin, si les préférences sont partielles, la propriété  $\beta$  peut s'avérer trop restrictive et n'est pas nécessairement garantie (cf. exemples 1.3 et 1.5).

Préférences	Fonction de choix	$\alpha$	$\beta$	Unicité
stricte totale	$c_{opt}$	✓	✓	✓
large totale	$c_{opt}$	✓	✓	✗
fonction d'utilité	$c_{max}$	✓	✓	✗
stricte partielle	$c_{ND}$	✓	✗	✗
large partielle	$c_{ND}$	✓	✗	✗

TABLEAU 1.4 : Propriétés des fonctions de choix selon leur modèle de préférence

Les différents modèles de préférence multi-critères peuvent quant à eux être comparés à l'aune de leur expressivité et leur compacité. En effet, pour atteindre une expressivité totale, il s'agirait de décrire exhaustivement l'ensemble des relations de préférence sur chaque couple d'alternatives. Or, quand l'espace des alternatives est très grand, un tel modèle n'est pas exploitable. De plus, de par leur structure, les problèmes de décision multi-critères induisent généralement des motifs dans les préférences du décideur. Ainsi, on peut chercher à exprimer les préférences sur les alternatives en fonction de préférences sur les valeurs d'un même attribut. C'est notamment le cas avec les préférences unanimes, cependant celles-ci laissent incomparables de nombreuses alternatives. Pour être plus expressif et complet, on peut utiliser des préférences lexicographiques qui consistent à classer les attributs. Les fonctions d'agrégation additive pondérée constituent le pendant utilitaire de cette approche en pondérant chaque critère. Enfin, des modèles plus expressifs comme les CP-net s'affranchissent de l'hypothèse de séparabilité en considérant que certaines préférences au sein d'un attribut sont conditionnées par les valeurs prises par d'autres attributs.



---

# Communication entre les agents

---

## Sommaire

<b>2.1</b>	<b>Introduction</b>	<b>27</b>
<b>2.2</b>	<b>Langage</b>	<b>28</b>
2.2.1	Théorie des actes de langage	28
2.2.2	Langage de communication d'agents	29
<b>2.3</b>	<b>Dialogue</b>	<b>31</b>
2.3.1	Typologie du dialogue	31
2.3.2	Protocoles d'interaction	33
2.3.3	Jeu de dialogue	35
<b>2.4</b>	<b>Synthèse</b>	<b>36</b>

---

## 2.1 Introduction

Pour rappel, notre étude a pour objet les décisions dans lesquelles interviennent plusieurs agents. Dans cette perspective, la communication entre les agents est donc un point clé auquel ce chapitre est consacré.

Le paradigme agent propose de considérer des entités capables de percevoir un environnement et d'agir sur celui-ci. Avec plusieurs agents, cette boucle de rétro-action permet des *interactions* entre les agents constitutives du *système complexe* formé par l'ensemble [Kubera *et al.* 2011]. On distingue deux types d'interactions :

- les *interactions directes*, c'est-à-dire l'échange de *messages* via un *langage* et un *protocole* communs ;
- les *interactions indirectes*, c'est-à-dire l'altération de l'environnement par un agent perçue par un autre qui agit en conséquence, l'environnement est alors le support de l'interaction.

Contrairement à l'interaction directe qui est intentionnelle – un agent s'adresse directement à un autre pour communiquer – l'interaction indirecte peut tout à fait ne pas l'être.

Ici, nous nous intéressons spécifiquement aux interactions directes et ne formulons aucune hypothèse quant à l'environnement dans lequel évoluent les agents. Dans ce cadre, percevoir, c'est recevoir des messages ; agir, c'est envoyer des messages. La communication entre les agents se base donc sur un échange d'informations via des messages comme dans le modèle défini

dans [Shannon 1948]. En conformité avec ce dernier, on appelle l'agent qui envoie un message son *émetteur* et celui qui le reçoit son *récepteur*.

Plus qu'un échange de signaux, la communication vise à ce que le récepteur d'un message interprète le contenu que son émetteur a souhaité exprimer. Il est donc nécessaire que les deux agents emploient un langage partagé pour communiquer, ce point fait l'objet de la section 2.2. Dans nos travaux, nous nous intéressons en particulier au dialogue. La section 2.3 est consacrée à cette forme de communication. Enfin, la section 2.4 propose une synthèse de ce chapitre.

## 2.2 Langage

Pour que l'on puisse à proprement parler de dialogue, il est nécessaire que les agents comprennent le contenu des messages qu'ils reçoivent. C'est ici qu'intervient la notion de langage. Si un **langage naturel** peut convenir pour le dialogue entre deux agents humains, il pose par contre de nombreuses difficultés dès lors qu'un des acteurs est un agent logiciel. Cet état de fait est notamment illustré par notre échec jusqu'alors à passer le test décrit par Turing dans [Turing 1950]. En effet, malgré les avancées dans le domaine du Traitement Automatique des Langues (TAL), les difficultés d'interprétation et de génération posées par le langage naturel, sur les plans syntaxiques, sémantiques et pragmatiques n'ont pas été totalement surmontées. Pour contourner ce problème, des langages aux règles explicites et strictes, réduisant l'ambiguïté, ont été conçus. Ces langages artificiels constituent des solutions plus formelles et permettent à des agents logiciels de dialoguer.

### 2.2.1 Théorie des actes de langage

De nombreux langages d'agents se fondent sur la *théorie des actes de langage*. Cette branche de la philosophie du langage repose sur le principe « Dire c'est faire », elle a été fondée par [Austin 1962] et notamment développée par [Searle 1969]. Cette approche propose de considérer les énoncés des agents comme des actes. Un acte de langage est composite, il est lui-même constitué :

- d'un *acte locutoire* ou *locution*, relatif à ce qui est dit par le locuteur de l'énoncé, c'est le contenu propositionnel de l'acte de langage ;
- d'un *acte illocutoire* ou *illocution*, relatif à la signification de ce qui est dit par l'émetteur de l'énoncé, c'est le sens de la proposition ;
- d'un *acte perlocutoire* ou *perlocution*, relatif aux conséquences de l'acte sur les états mentaux du récepteur de l'énoncé. En d'autres termes, il s'agit des modifications dans l'état mental du récepteur entraînées par la prise de connaissance du message.

Dans une optique analytique, Austin a proposé une classification de ces actes en fonction de leur but illocutoire. Par la suite, Searle a lui aussi proposé une classification qui fait désormais référence [Searle 1969]. Il y distingue cinq types d'actes :

- *assertifs* ou *représentatifs*, ils expriment une affirmation, un fait sur le monde ;
- *directifs*, ils orientent le récepteur vers l'accomplissement d'un acte ;
- *commissifs* ou *promissifs*, ils engagent l'émetteur à accomplir un acte ;
- *expressifs*, ils expriment une affirmation sur l'état mental de l'émetteur ;

- *déclaratifs*, ils accomplissent l'action contenue dans la proposition de l'acte par leur énonciation.

Le tableau 2.1 illustre les différents types d'actes de langage en associant quelques verbes à chacun d'eux ainsi qu'un exemple.

Type d'acte de langage	Verbes	Exemple
assertif	penser, affirmer, informer, signaler	« Il pleut dehors. »
directif	suggérer, proposer, demander, ordonner, supplier	« Vous prendrez un café ? »
commissif	s'engager, accepter, promettre, menacer	« Je m'engage à dire la vérité. »
expressif	s'excuser, s'indigner, protester, féliciter	« Bravo ! »
déclaratif	déclarer, approuver, renoncer, dénoncer	« Je vous déclare innocent. »

TABLEAU 2.1 : Illustration des différents types d'actes de langage

La typologie définie précédemment a servi de base pour l'élaboration de langages de communication d'agents.

### 2.2.2 Langage de communication d'agents

Dans les SMA, lorsqu'on souhaite que les agents puissent communiquer et se comprendre, on utilise généralement un langage de communication d'agents ou *ACL* (pour *Agent Communication Language*). Contrairement au langage naturel, un *ACL* est généralement plus explicite, et donc plus formel, afin d'éviter l'ambiguïté dans la communication entre les agents. Ainsi, il est pourvu :

- d'une *syntaxe*, qui définit la structure à laquelle doit se conformer tout énoncé du langage ;
- d'une *sémantique*, qui définit le sens à conférer à un énoncé du langage ;
- d'une *pragmatique*, qui définit le contexte dans lequel formuler un énoncé du langage.

Parmi les *ACL* les plus notoires proposés par la communauté scientifique, on peut citer chronologiquement KQML [Finin *et al.* 1994] et FIPA-ACL [FIPA 2002b]. Nous concentrons ici notre étude sur FIPA-ACL, largement répandu et considéré comme un standard. Nous présentons ici sa syntaxe et sa sémantique, la pragmatique est abordée dans la section 2.3.2.

#### Syntaxe

Proche de la syntaxe de KQML, FIPA-ACL structure les messages par un ensemble de champs définis dans [FIPA 2002a] et listés dans le tableau 2.2. Les 22 actes de communication permis par FIPA-ACL sont exprimés par des performatifs et classés en catégorie dans la table 2.3.

Paramètre	Description
<code>performative</code>	type d'acte de communication
<code>sender</code>	émetteur du message
<code>receiver</code>	destinataire(s) du message
<code>reply-to</code>	destinataire à indiquer en réponse à ce message
<code>content</code>	contenu du message
<code>language</code>	langage du contenu du message
<code>encoding</code>	mode d'encodage du contenu du message
<code>ontology</code>	ontologie(s) utilisée(s) pour la sémantique du contenu
<code>protocol</code>	protocole employé par l'émetteur du message
<code>conversation-id</code>	identifiant de la conversation
<code>reply-with</code>	identifiant du message pour référence ultérieure
<code>in-reply-to</code>	réponse avec référence à un message antérieur
<code>reply-by</code>	date butoir de réponse à ce message

TABLEAU 2.2 : Paramètres des messages FIPA

## Sémantique

À chaque champ ou paramètre du message est associée une sémantique décrite dans la table 2.2. Plus en détails :

- le champ `performative` est essentiel et requis pour tout message. On lui associe un des 22 actes performatifs de la table 2.3, chacun ayant sa sémantique précisée dans [FIPA 2002b]. Parmi l'ensemble des performatifs, 4 sont primitifs, les autres peuvent être obtenus par composition.
- 3 champs concernent la communication : `sender` et `receiver` indiquent respectivement l'émetteur et le destinataire du message ; `reply-to` indique à qui adresser une réponse au message.
- 4 champs concernent le contenu : `content` qui contient le contenu lui-même ; `language`, `encoding` et `ontology` qui permettent son interprétation.
- 5 champs concernent l'enchaînement des messages : `protocol`, définit les règles de la conversation ; `conversation-id`, renseigne l'identifiant de la conversation ; `reply-with`, indique l'identifiant du message ; `in-reply-to` indique le message auquel on répond et `reply-by` fixe une date butoir pour la réponse.

### Exemple 2.1 (Exemple d'acte de langage *inform* d'après [FIPA 2002b]).

*Ci-dessous est décrit un exemple d'acte de langage *inform* auquel est associé 4 paramètres. Le premier indique que l'agent *i* est l'émetteur de l'acte, le second que l'agent *j* est son destinataire. Le paramètre *content* décrit le contenu duquel l'agent *i* souhaite informer l'agent *j*, en l'occurrence une prévision météorologique de pluie pour le lendemain. Enfin, le paramètre *language* indique dans quel langage ce contenu est exprimé, ici en Prolog.*

```
(inform
:sender i
:receiver j
```



Catégorie	Performatifs	Primitif
Transmission d'information	inform	✓
	inform-if	✗
	inform-ref	✗
	confirm	✓
	disconfirm	✓
Demande d'information	query-if	✗
	query-ref	✗
	subscribe	✗
Négociation	accept-proposal	✗
	cfp	✗
	propose	✗
	reject proposal	✗
Distribution de tâche	request	✓
	request-when	✗
	request-whenever	✗
	agree	✗
	cancel	✗
	propagate	✗
	proxy	✗
	refuse	✗
Manipulation d'erreurs	failure	✗
	not-understood	✗

TABLEAU 2.3 : Classification des actes de communication de FIPA-ACL

```

:content "meteo(demain, pluie)"
:language Prolog
)

```

## 2.3 Dialogue

Le mode de communication le plus élémentaire est le *monologue* : un agent a le rôle d'émetteur, le ou les autres ont le rôle de récepteur. Dans notre contexte, nous considérons au contraire que l'ensemble des agents impliqués dans le processus décisionnel sont susceptibles d'adopter tantôt le rôle d'émetteur, tantôt celui de récepteur, on parle alors de *dialogue*. Le dialogue ne se distingue pas seulement par un échange de message mais surtout par "*la recherche d'une inter-compréhension*" [Pasquier 2005].

### 2.3.1 Typologie du dialogue

Dans [Walton & Krabbe 1995], une typologie de six types de dialogues de base est décrite, nous la présentons dans le tableau 2.4.

Chaque type de dialogue est défini par sa situation initiale, le but de ses participants et l'objectif

Type de dialogue	Situation initiale	But des participants	But du dialogue
<i>Demande d'information</i>	Manque d'information	Acquérir ou donner des informations	Partage d'informations
<i>Enquête</i>	Désir d'une preuve	Démontrer un fait	Vérifier/Réfuter des hypothèses
<i>Délibération</i>	But commun à atteindre	Trouver le meilleur plan d'action	Coordination des participants
<i>Persuasion</i>	Conflit d'opinion	Persuader l'autre	Résolution ou clarification
<i>Négociation</i>	Conflit de désirs	Arriver à la situation la plus favorable	Parvenir à un accord
<i>Éristique</i>	Conflit personnel	Attaquer verbalement l'interlocuteur	Expression d'un antagonisme

TABLEAU 2.4 : Description des types de dialogues d'après [Walton 2010]

du dialogue en lui-même.

Dans un dialogue de *demande d'information*, un participant fait l'hypothèse d'une asymétrie de l'information avec son interlocuteur. Le dialogue permet de réduire cette asymétrie par le partage de l'information.

Exemple : *Alice a oublié sa montre et demande l'heure à Bob qui la lui donne.*

Dans un dialogue d'*enquête*, les participants ont pour but commun la recherche de nouvelles connaissances. Partant d'un problème ouvert, il s'agit à l'issue du dialogue de pouvoir répondre à une question en démontrant la véracité ou la fausseté de propositions.

Exemple : *Alice et Bob cherchent à établir ensemble qui était le quatrième président de la cinquième république française.*

Un dialogue de *délibération* est, comme une enquête, coopératif. Il se distingue par son objet : établir un plan d'action pour atteindre un objectif commun. L'enjeu n'est pas ici de savoir ce qui est vrai mais de décider quoi faire. Pour McBurney *et al.*, dans un cadre délibératif, les agents sont prêts à partager leurs croyances et préférences et une proposition peut être optimale pour le groupe sans l'être pour aucun des agents qui le compose [McBurney *et al.* 2007].

Exemple : *Alice et Bob conviennent ensemble d'un rendez-vous.*

Dans un dialogue de *négociation*, les participants ont là aussi à prendre une décision mais contrairement à une délibération, ils ne désirent pas une même situation finale. Négocier constitue alors un moyen pour les agents d'atteindre un compromis.

Exemple : *Alice veut regarder un match de football tandis que Bob souhaite regarder un documentaire animalier. Après d'âpres négociations, ils regarderont finalement un épisode de Monty Python's Flying Circus.*

Un dialogue de *persuasion* peut être perçu comme un jeu où les participants s'affrontent : chacun a une opinion différente et souhaite la voir adopter par son interlocuteur. Pour ce faire, ils échangent des arguments sous la forme de propositions afin de défendre leur point de vue ou d'attaquer le point de vue adverse.

Exemple : *Alice est pro-nucléaire et essaie de faire changer d'opinion Bob qui lui est anti-nucléaire.*

Le dialogue *éristique* correspond à une situation où chacun des participants a pour but d'attaquer l'autre verbalement, nous ne nous y intéressons pas plus en détails ici.

Tandis que les dialogues de demande d'information, d'enquête et de délibération partent d'une situation ouverte, les dialogues de persuasion, de négociation et d'éristique ont pour origine un conflit. Dans les premiers, les participants au dialogue partagent généralement un but commun et sont coopératifs, dans les seconds, ils ont des buts différents voire incompatibles et sont davantage en opposition. La figure 2.1 propose une vue analytique des différents types de dialogue à travers le prisme de deux axes : l'axe vertical (coopération/conflict) décrit si les agents sont alliés ou au contraire opposés ; l'axe horizontal (épistémique/pratique) situe le but du dialogue, soit du côté de la recherche de la connaissance, soit de celui de la prise de décision.

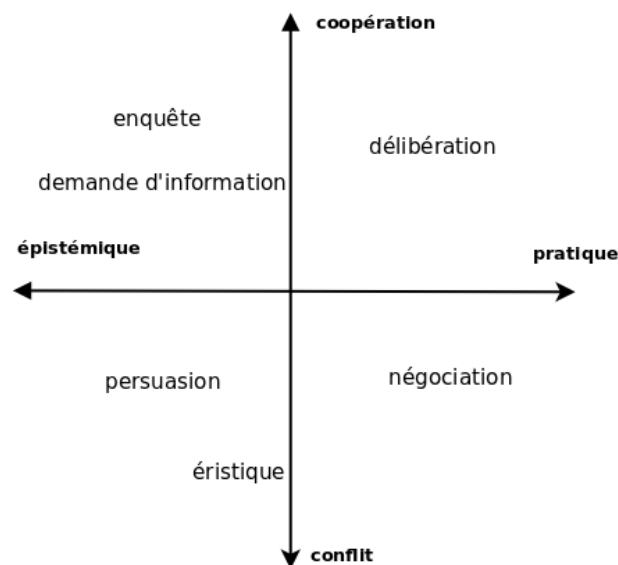


FIGURE 2.1 : Vue analytique des dialogues

Quand l'objectif visé est une prise de décision collective, on se situe soit dans un dialogue de négociation s'il y a conflit, soit dans un dialogue de délibération si le problème est ouvert.

Cette typologie de dialogue a une visée normative, ses auteurs ne prétendent pas qu'elle corresponde strictement à la réalité ni qu'elle est exhaustive. De plus, un dialogue n'est pas nécessairement d'un type unique, il peut être mixte et être décomposé en sous-dialogues de type différents ; les agents peuvent passer d'un type de dialogue à un autre, c'est-à-dire procéder à des changements dialectiques.

### 2.3.2 Protocoles d'interaction

Un protocole de dialogue décrit les règles d'enchaînements des actes de langage. Il est partagé par les participants auxquels ils doivent conformer leurs interactions pour le bon déroulement du dialogue. Un protocole affecte des rôles aux participants et décrit les séquences de messages autorisées. Dans la littérature, on trouve différentes formes de représentations des protocoles [Huget & Koning 2001] parmi lesquelles notamment le diagramme d'interaction UML et l'automate à états fini.

Le langage UML (acronyme de *Unified Modeling Language*) est, comme son nom l'indique, un langage de modélisation unifié. Souvent employé pour concevoir des logiciels, il propose un ensemble de conventions graphiques pour la modélisation d'objets notamment. Afin de répondre à certaines problématiques spécifiques de l'approche agent, une extension d'UML nommée AUML, pour *Agent UML* a été proposée [Bauer et al. 2001], elle permet de représenter les protocoles d'interactions et leurs instances sous la forme de diagrammes. En particulier, on trouve les protocoles FIPA représentés sous ce formalisme [FIPA 2002b].

Un diagramme d'interaction est l'extension agent d'un diagramme de séquence UML. Il prend la forme d'un graphique dans lequel :

- les rôles des participants du protocole sont symbolisés par des rectangles alignés horizontalement et annotés par le nom du rôle ;
- à chaque rôle est associé une ligne de vie, symbolisée par un trait interrompu fin et décrivant un axe temporel ;
- les échanges de messages entre deux rôles sont symbolisés par des flèches allant de l'émetteur du message vers son récepteur et annotées par un acte de langage dont le contenu n'est généralement pas spécifié.

**Exemple 2.2 (Diagramme d'interaction AUML du protocole Propose).**

La figure 2.2 décrit le protocole de proposition sous la forme d'un diagramme d'interaction. Le nom du protocole (FIPA-Propose-Protocol) est indiqué dans l'encart en haut à gauche. Le protocole décrit deux rôles : l'initiateur (Initiator dans le premier rectangle) et le participant (Participant dans le second rectangle). Dans ce protocole, l'initiateur du dialogue envoie un acte propose au participant qui y répond soit par un acte reject-proposal, soit par un acte accept-proposal. Le losange indique le choix du participant entre ces deux possibilités.

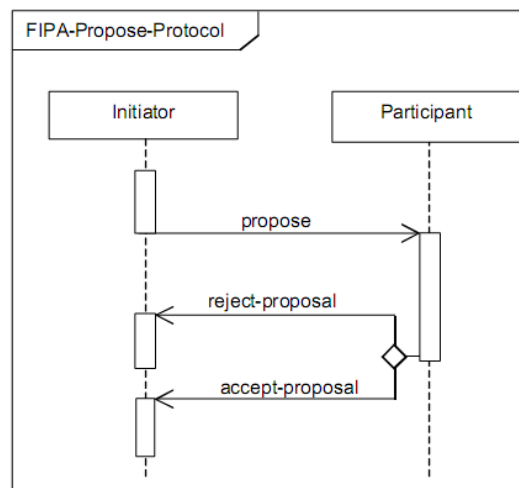


FIGURE 2.2 : Diagramme d'interaction du protocole de proposition FIPA

La description d'un protocole peut aussi prendre la forme d'un *automate à états fini*. Chaque action d'un agent est alors modélisée par une transition d'un état de la conversation vers un autre. Ainsi, un automate à états fini décrivant un protocole peut être défini par :

- un ensemble fini d'états de dialogue, parmi lesquels certains sont caractéristiques :

- un unique état initial auquel commence le dialogue,
- un ensemble d'états finaux dans lesquels se termine le dialogue ;
- un ensemble fini d'actes de langage ;
- un ensemble de transitions, c'est-à-dire de triplets associant chacun un état de départ, un acte de langage qu'un émetteur réalise auprès de récepteur(s) et un état d'arrivée.

L'automate est généralement représenté par un graphe orienté dans lequel chaque état est un nœud représenté par un cercle étiqueté par son nom. L'état initial se distingue par la présence d'un arc dont il est la destination et qui n'a pas d'origine. Les états finaux sont indiqués par un double cerclage. Chaque transition est représentée par un arc allant d'un état de départ à un état d'arrivée et annotée par une condition, portant généralement sur l'énonciation d'un acte.

**Exemple 2.3 (Automate à états fini décrivant le protocole de proposition FIPA).**

La figure 2.3 décrit le protocole de proposition sous la forme d'un **diagramme d'interaction**. À l'état initial, l'agent initiateur envoie un acte propose au participant. À l'état suivant (numéroté 2), le participant peut soit répondre par un acte reject-proposal soit par un acte accept-proposal et atteindre dans un cas comme dans l'autre un état final (états numérotés 3 et 4).

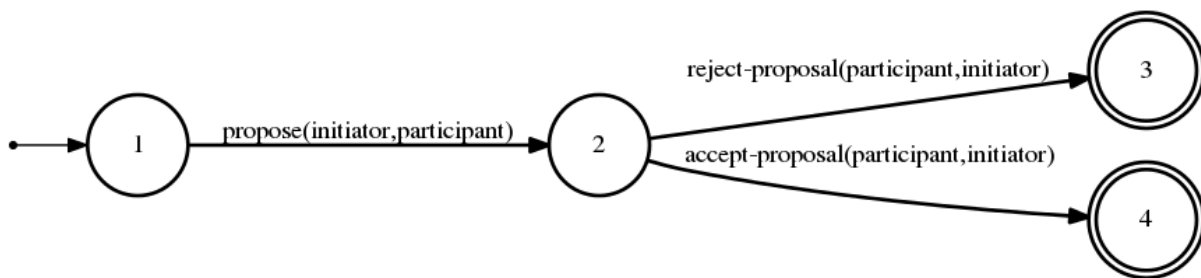


FIGURE 2.3 : Automate à états fini du protocole de proposition FIPA

Il existe d'autres modèles de représentation des protocoles parmi lesquelles les réseaux de Petri [Petri 1979], généralement présentés sous formes de graphes orientés avec deux types de nœuds : les places et les transitions. Nous choisissons de ne pas entrer dans le détail de leur présentation ici et renvoyons le lecteur au chapitre 6 de [Ferber & Perrot 1995] ou à [Huget & Koning 2001].

### 2.3.3 Jeu de dialogue

On peut voir le processus dialogique à travers la métaphore d'un jeu [Morge 2005]. Les participants au dialogue sont alors des joueurs qui se voient attribuer un rôle et dont les coups sont des messages qui doivent respecter des règles spécifiés par un protocole. Une fois le jeu terminé, on appelle historique la séquence des coups de dialogue.

**Définition 2.1 (Jeu de dialogue).**

Soit  $ACL$  un langage de communication d'agents (cf. définition 2.2.2). Un **jeu de dialogue** est un tuple  $JD = (P, \mathcal{M}, T, protocol, H)$  avec :

- $P$ , l'ensemble des joueurs qui prennent part au jeu de dialogue ;

- $\mathcal{M} \subseteq \mathcal{ACL}$ , l'ensemble des actes de langage bien formés ;
- $T: H \rightarrow \mathcal{P}$ , la fonction de distribution du tour de parole ;
- $\mathit{protocol}: H \rightarrow 2^{\mathcal{M}}$ , la fonction correspondant au protocole qui détermine les coups légaux qui sont autorisés pour étendre l'historique ;
- $H$ , l'ensemble des historiques possibles, c'est-à-dire des séquences de coups en conformité avec la fonction de tour de parole  $T$  et le protocole  $\mathit{protocol}$ .

## 2.4 Synthèse

Dans ce chapitre consacré à la communication entre agents, nous avons identifié différentes fins et moyens du dialogue. Dans cette thèse, nous nous intéressons en particulier aux prises de décision collective et assistée.

Dans le cadre d'une prise de décision assistée, le dialogue s'apparente davantage à la *demande d'information*, l'*enquête* et éventuellement la *persuasion* et la *délibération* sur certains aspects dans des cas spécifiques.

Dans le cadre de la prise de décision collective, les dialogues sont du type de la  *négociation* ou de la *délibération* selon que le but soit ou non commun aux agents.

Si les langages naturels sont utilisés quotidiennement par les sociétés humaines, ils posent des difficultés majeures pour leur interprétation par des agents logiciels, notamment par leur caractère ambigu. Ainsi, il est nettement plus facile de concevoir des agents employant un langage artificiel dont la syntaxe, la sémantique et la pragmatique reposent sur des bases plus formelles. En conclusion, il n'y a pas de mode de communication supérieur dans l'absolu, tout dépend de la nature des agents impliqués et du but poursuivi. Dans le chapitre suivant, nous nous intéressons à un cas de dialogue entre un agent humain qui souhaite prendre une décision et son interlocuteur, un agent logiciel, qui l'assiste dans cette tâche.

---

# Prise de décision assistée

---

## Sommaire

---

<b>3.1 Introduction</b>	<b>37</b>
<b>3.2 Méthodes et outils d'aide à la décision</b>	<b>37</b>
<b>3.3 Systèmes experts</b>	<b>39</b>
<b>3.4 Agents conversationnels</b>	<b>39</b>
3.4.1 Perception et analyse des entrées de l'utilisateur	40
3.4.2 Expression de l'agent	40
3.4.3 Gestion du dialogue et raisonnement	41
<b>3.5 Systèmes de recommandation</b>	<b>42</b>
<b>3.6 Synthèse</b>	<b>43</b>

---

## 3.1 Introduction

Ce chapitre a pour objet la prise de décision assistée et vise à dresser un état de l'art des approches et solutions informatiques existantes en la matière. Pour ce faire, nous prenons appui à la fois sur le chapitre 1 et 2 consacrés respectivement à la prise de décision et à la communication.

Nous commençons notre passage en revue des différentes approches et technologies par les méthodes d'aide à la décision en section 3.2. Puis, nous continuons avec les systèmes experts, présentés dans la section 3.3. Dans la section 3.4, nous nous intéressons aux agents conversationnels. Nous étudions ensuite les systèmes de recommandation dans la section 3.5. Enfin, la section 3.6 propose une synthèse comparative des différentes approches.

## 3.2 Méthodes et outils d'aide à la décision

Le champ de l'aide à la décision regroupe l'ensemble des méthodes et procédés mis en œuvre dans le but d'accompagner le décideur dans sa prise de décision. [Roy 1985] propose une définition précise de l'aide à la décision : « L'aide à la décision est l'activité de celui qui, prenant appui sur des modèles clairement explicités mais non nécessairement complètement formalisés, aide à obtenir des éléments de réponses aux questions que se pose un intervenant dans le processus de décision, éléments concourant à éclairer la décision et normalement à prescrire, ou simplement

à favoriser un comportement de nature à accroître la cohérence entre l'évolution du processus d'une part, les objectifs et le système de valeurs au service desquels cet intervenant se trouve placé d'autre part. »

Dans un dialogue d'aide à la décision, on distingue ainsi au moins deux participants ayant chacun un rôle distinct :

- le **décideur**, qui est confronté à un problème de décision qu'il tente de résoudre, c'est lui qui prend la décision ;
- l'**analyste**, qui adopte les objectifs du décideur et cherche à l'aider en dialoguant avec lui sans toutefois pouvoir prendre la décision à sa place.

Initialement, l'aide à la décision prend typiquement la forme d'avis de tiers sur un problème de décision ayant trait à un domaine duquel ils sont experts. Par la suite, les outils mathématiques ont été introduits pour formaliser davantage la prise de décision. Tsoukiàs présente ainsi l'aide à la décision comme issue de la recherche opérationnelle et caractérisée par une approche formelle, abstraite et rationaliste [Tsoukiàs 2008]. Avec l'essor de l'Informatique, l'aide à la décision a intégré des outils logiciels permettant d'aller plus loin notamment dans la modélisation et la simulation. Enfin, dans une perspective à rapprocher de l'IA, l'aide à la décision s'automatise et prend la forme de logiciels interagissant directement avec le décideur pour l'assister [Ouedane 2009].

[Roy 1985] catégorise 4 problématiques traitées par l'aide à la décision :

- la *problématique du choix*, où il s'agit de restreindre autant que possible l'ensemble des alternatives en gardant les meilleures d'entre elles ;
- la *problématique du tri*, où il s'agit de catégoriser les alternatives ;
- la *problématique du rangement*, où il s'agit de regrouper les alternatives en classe d'équivalence ;
- la *problématique de la description*, où il s'agit de décrire les alternatives et leurs conséquences.

Ici, nous nous intéressons en particulier à la problématique du choix et, à son préalable, la problématique de la description. Pour prendre ses décisions, l'agent décideur raisonne sur sa base de croyances au sens large (que ce soit sur les alternatives, sur ses buts, sur ses préférences, etc.), autrement dit, sur sa représentation du problème de décision. Pour l'agent analyste, l'aide à la décision passe par une interaction avec l'agent décideur afin de venir modifier ou compléter les croyances du décideur.

Parmi les contributions les plus notables sur ces problématiques, les méthodes ELECTRE (pour ELimination Et Choix Traduisant la REalité) [Roy & Bouyssou 1993] constituent une panoplie d'outils méthodologiques spécifiques pour résoudre les problématiques de choix, de tri et de rangement. Ces méthodes reposent sur une approche dite de *surclassement* qui consiste à identifier les paramètres du problème, en particulier les critères, pour ensuite agréger les comparaisons des alternatives pour chacun de ces critères afin de rendre les alternatives comparables et enfin de procéder à une recommandation. L'incomparabilité entre deux alternatives est généralement prise en compte dans ces méthodes. Certaines de ces méthodes sont cependant complexes à l'utilisation et peuvent nécessiter de passer par une évaluation numérique du problème par le décideur.



### 3.3 Systèmes experts

Un *système expert* est un logiciel visant à capturer une expertise dans un domaine particulier afin de répondre à des questions. Par exemple, le système expert MYCIN présenté dans [Shortliffe et al. 1975] a été développé dans le champ médical pour l'aide au diagnostic et à la prescription pour certaines infections. On peut décomposer un système expert en deux parties :

- une *base de connaissance*, elle même composée :
  - d'une *base de règles*, qui regroupe l'ensemble des règles constituant l'expertise dans le domaine ;
  - d'une *base de faits*, qui rassemble les éléments décrivant un cas concret particulier ;
- un *moteur d'inférences* qui permet d'inférer de nouveaux faits à partir des règles et des faits connus.

Les systèmes experts sont donc conçus en suivant une architecture cognitive. La base de connaissance correspond à la mémoire du système : elle comporte une partie dynamique, la base de faits, propre à un problème donné et évoluant au cours du temps, ainsi qu'une partie statique, la base de règles, valable quelque soit le cas concret étudié. Une règle décrit l'implication d'une action par une condition. La condition porte sur un ou plusieurs faits, appelés prémisses de la règle, tandis que l'action consiste généralement à modifier la base de faits. Le moteur d'inférences se base sur différents mécanismes de chaînage pour raisonner, on distingue notamment :

- le chaînage avant, qui consiste à effectuer toutes les déductions possibles ;
- le chaînage arrière, qui consiste à vérifier si un fait est vrai ou non en remontant les règles ;
- le chaînage mixte, qui combine les deux précédents : l'application du chaînage avant jusqu'à saturation de la base, puis l'identification de faits encore déductibles par chaînage arrière pour identifier les questions pertinentes à poser à l'utilisateur.

On distingue différentes classes de systèmes experts selon l'ordre de la logique associée. Ainsi, un système expert peut être qualifié :

- d'ordre 0, s'il relève uniquement du calcul propositionnel, les faits sont alors à valeur booléenne ;
- d'ordre 0+ ou 0.5 si les faits sont à valeur symbolique ou réelle ;
- d'ordre 1, s'il relève du calcul des prédicats.

De plus, on peut raffiner encore le modèle en y incluant des coefficients de vraisemblance sur les faits, comme c'est le cas pour MYCIN.

### 3.4 Agents conversationnels

Les premiers agents conversationnels sont apparus au milieu des années 60. ELIZA [Weizenbaum 1966] en est un exemple célèbre. Son fonctionnement consiste à simuler le comportement d'un psychologue rogiérien en formulant des questions à partir des énoncés saisis par l'utilisateur. Depuis, on trouve des agents conversationnels sous des formes diverses et variées. Au-delà de la dimension conversationnelle ludique, de nombreux agents développés ont un rôle plus pragmatique d'assistants personnels, avec pour but d'aider l'utilisateur dans ses tâches [Maes et al. 1994]. Depuis l'essor d'internet, certaines entreprises se sont spécialisées dans la conception

et l'intégration d'agents conversationnels aux sites web<sup>1</sup>. Dans la plupart des cas, ils constituent un interlocuteur pour répondre aux questions que se pose l'utilisateur. Ils constituent alors une alternative interactive à la classique rubrique FAQ (*Frequently Asked Questions*). Le web a ainsi vu apparaître de nombreux assistants virtuels parmi lesquels on peut par exemple citer les agents Anna d'Ikea<sup>2</sup> et Laura d'ERDF<sup>3</sup>.

Le dénominateur commun de ces agents est la recherche d'une interaction intuitive, similaire à celle que leur interlocuteur aurait avec un humain. Dans la communauté scientifique française, les problématiques inhérentes à la conception d'agents conversationnels sont notamment l'objet d'étude du groupe de travail « Affects, Compagnon Artificiels et Interactions » (ACAI)<sup>4</sup>. On distingue notamment trois grandes problématiques :

- la perception et analyse des entrées de l'utilisateur (section 3.4.1) ;
- la production de sorties par l'agent (section 3.4.2) ;
- la gestion du dialogue et le raisonnement de l'agent conversationnel (section 3.4.3).

### 3.4.1 Perception et analyse des entrées de l'utilisateur

Dans le cas d'ELIZA comme pour beaucoup d'agents conversationnels, l'interface avec l'utilisateur est similaire à celle employée dans une conversation entre deux humains via un logiciel de messagerie instantanée avec :

- un champ de saisie, permettant à l'utilisateur d'entrer des énoncés au clavier ;
- et un historique de conversation, où est affiché tout ou partie des derniers énoncés de l'agent conversationnel et de l'utilisateur.

Ainsi, les entrées détectées par l'agent sont les messages saisis au clavier par l'utilisateur en langage naturel. Les entrées de l'utilisateur peuvent prendre d'autres formes que celles de chaînes de caractères. C'est notamment de plus en plus le cas depuis les années 1980 et le développement des interfaces multimodales [Bolt 1980]. Ainsi, d'autres agents prennent en entrée des sons ce qui vient ajouter une problématique de reconnaissance automatique de la parole. Enfin, les entrées peuvent être des signaux vidéo, il s'agit alors de reconnaître l'utilisateur et de détecter chez lui des expressions non verbales comme des gestes ou encore des émotions [Devillers *et al.* 2005]. Si la possibilité laissée à l'utilisateur de s'exprimer librement dans sa langue naturelle permet une interface intuitive et conviviale, elle provoque cependant chez lui de fortes attentes dans la compréhension de ses énoncés par l'agent conversationnel.

### 3.4.2 Expression de l'agent

Contrairement à ELIZA, de nombreux agents sont *incarnés* dans des avatars et parfois même *animés*. Selon Lester *et al.*, la personnification d'un interlocuteur crédibilise davantage le système, ce phénomène est appelé le *Persona Effect* [Lester *et al.* 1997].

Par exemple, Riviere *et al.* proposent une librairie d'actes de langage adaptés à l'expression

---

1. voir <http://www.chatbots.org/> pour un panorama des agents conversationnels existants

2. [http://www.ikea.com/ms/fr\\_FR/customer\\_service/page\\_anna.html](http://www.ikea.com/ms/fr_FR/customer_service/page_anna.html)

3. <http://bleuciel.edf.com>

4. <http://acai.limsi.fr>

multimodale d'un agent conversationnel animé [Rivière & Pesty 2010]. S'appuyant sur la théorie des actes de discours de Searle et Vanderveken (cf. section 2.2.1, page 28), ces travaux font le lien entre des *émotions* dites complexes et des actes de langages au travers des conditions de sincérité, du mode d'accomplissement et du degré de puissance. Les actes de langages proposés ont été testés à l'aide de l'agent conversationnel GRETA développé par l'équipe de Catherine Pelachaud [Poggi *et al.* 2005].

Dans [Braffort *et al.* 2010], l'outil DIVA permet de munir les agents de capacités déictiques afin de guider l'utilisateur dans l'interface sur une page web. Les agents sont alors directement intégrés à la structure de la page.

Les attentes fortes de l'utilisateur évoquées dans la section 3.4.1 sont renforcées par l'incarnation dans un avatar souvent anthropomorphique. Pour y répondre, il faut que l'agent soit capable de produire des énoncés en cohérence avec ceux de l'utilisateur, idéalement, semblables à ceux d'un alter ego humain.

### 3.4.3 Gestion du dialogue et raisonnement

L'une des problématiques principales concernant les agents conversationnels est leur compétence dialogique. En effet, la détection d'entrées et la production de sorties ne suffisent pas à garantir la crédibilité de l'agent, pour cela encore faut-il que l'agent fasse preuve de cohérence dans son comportement. L'une des problématiques est alors celle du traitement computationnel d'énoncés en langage naturel abordée en section 2.2 (page 28). L'approche adoptée dans le cas d'ELIZA est celle du *filtrage par motif*, c'est-à-dire que l'agent reconnaît un certain nombre de motifs prédéfinis. Il n'y a pas à proprement parler de module de gestion du dialogue ni de but défini.

Parmi les contributions plus récentes, les travaux de Barthès [Barthès 2007] se focalisent le problème de l'interprétation du langage naturel pour l'élaboration d'un agent assistant. Le principe est de circonscrire un ensemble de tâches pour définir une ontologie du domaine et un ensemble de gabarits (en anglais *pattern*) qui permettent d'identifier la tâche demandée à l'agent. Différents outils sont envisageables dans l'optique de favoriser cette adéquation entre entrées reçues et sorties produites, on peut notamment citer le langage NabuTalk créé par la société Dialonics<sup>5</sup> qui inclut du traitement sémantique et permet la construction de motifs évolués. D'autres travaux proposent de spécifier le dialogue sous la forme d'un jeu en s'appuyant sur des corpus de dialogue Homme-Homme afin de mieux définir le comportement de l'agent [Dubuisson Duplessis 2014]. On peut citer également [Sansonnet *et al.* 2013] qui propose une architecture d'agent visant à doter leur raisonnement de traits psychologiques pour guider leur comportement social.

Bien que de nombreuses recherches vont dans ce sens, les capacités de l'agent sur le plan dialogique sont actuellement très éloignées de celle des humains et dans de nombreux cas, ne satisfont pas les attentes que leur interface suggère.

---

5. depuis acquise par Akio : <http://www.akio.fr>

### 3.5 Systèmes de recommandation

De plus en plus répandus depuis plusieurs années, les systèmes de recommandation visent à suggérer à l'utilisateur des alternatives pertinentes dans un contexte spécifique (livre, film, ressource web, service, etc.). Ils sont typiquement employés quand l'utilisateur fait face à un grand catalogue d'alternatives ou qu'il ne connaît pas suffisamment le domaine pour parvenir à évaluer la satisfaction qu'il peut en attendre. Le rôle du système de recommandation est alors d'aider l'utilisateur à faire face à la surcharge informationnelle pour faciliter sa prise de décision et favoriser des alternatives plus satisfaisantes [Resnick & Varian 1997]. Pour que les recommandations soient personnalisées, il est nécessaire de disposer d'informations sur l'utilisateur. Ces données peuvent être collectées selon différentes modalités, il peut s'agir :

- d'une collecte implicite, par exemple l'historique de navigation de l'utilisateur ;
- d'une collecte explicite, par exemple sous la forme d'une évaluation comparative ou absolue d'alternatives.

Une collecte explicite peut être plus ou moins interactive. Il existe des systèmes passifs, où c'est l'utilisateur qui fournit un ensemble de données au système, par exemple en notant les films de son choix. Au contraire, d'autres systèmes sont plus interactifs et la collecte de données prend la forme d'une conversation.

À partir des données collectées, le système génère un modèle de l'utilisateur et de ses actions afin de prédire quels éléments sont susceptibles de l'intéresser. Il existe de nombreux algorithmes de filtrage employés pour le calcul des recommandations, on peut généralement les classer en deux catégories : collaboratif ou par contenu.

Dans le cas du *filtrage collaboratif*, il s'agit, au sein d'un environnement social, d'identifier des profils présentant des similarités avec celui de l'utilisateur. Prenons par exemple la recommandation d'articles de recherche, le système peut établir le profil en fonction des lectures des utilisateurs et mettre en avant les articles lus par des pairs ayant de nombreuses lectures communes. On qualifie de telles recommandations de sociales.

Dans le cas du *filtrage par contenu*, il s'agit d'identifier des alternatives présentant des similarités avec celles pour lesquels le décideur a montré un intérêt. Cette approche requiert donc de disposer d'un modèle assez riche des alternatives, par exemple décrivant leur caractéristiques (comme décrit dans la section 1.5, page 19) ou leur associant des mots clés. Avec cette méthode, la recommandation d'un article de recherche peut par exemple se baser sur le nombre de mots clés communs avec ceux des articles lus par l'utilisateur.

Tandis que le filtrage par contenu nécessite d'avoir accès aux caractéristiques des alternatives, le filtrage collaboratif présente l'avantage de ne requérir aucune description des alternatives. Cependant, pour être efficace et pertinent, le filtrage collaboratif demande de disposer d'une base de profils d'utilisateurs large et diverse alors que le filtrage par contenu peut être opérationnel et pertinent dès le premier utilisateur. Afin de pallier aux limites de chacune des approches et de gagner en robustesse et en pertinence, des *systèmes hybrides* ont été développés. Avec leur développement et leur popularisation sur le web, les systèmes de recommandation continuent à faire l'objet de nombreux travaux sur des problématiques comme leur évaluation, leur interface ou leur scalabilité [Ricci *et al.* 2011].

## 3.6 Synthèse

Dans ce chapitre, nous avons étudié la problématique d'un agent logiciel assistant un humain dans sa prise de décision. Une synthèse des différentes approches répondant à cette problématique et des objectifs poursuivis dans chacune d'elle est présenté dans le tableau 3.1.

Outil	Approche	But
Méthodes d'aide à la décision	numérique	éclairer le décideur
Systèmes experts	logique	automatiser un raisonnement
Agents conversationnels	symbolique	permettre une interaction intuitive
Systèmes de recommandation	numérique	proposer un item pertinent

TABLEAU 3.1 : Grille d'analyse des objectifs poursuivis dans les différentes approches

Alors que l'aide à la décision suppose dans les termes une intention bienveillante de la part de l'assistant, aucune hypothèse préalable n'est faite en la matière dans le cadre des systèmes de recommandation. Les systèmes experts sont des outils de production de raisonnements logiques qui ont notamment été utilisés pour l'aide au diagnostic. En ce qui concerne les agents conversationnels, c'est le sentiment de convivialité qui est souvent recherché, généralement en essayant de mimer les comportements humains. L'interface et l'interaction sont alors primordiales.

Les intérêts et limites de chacune de ces approches sont résumés dans le tableau 3.2.

Approche	Avantages	Limites
Méthodes d'aide à la décision	Explicite les données du problème, prend en compte l'incomparabilité	parfois complexe, peuvent requérir une évaluation numérique du décideur
Systèmes experts	explicable, permet d'argumenter	pas conviviaux, requièrent la saisie de règles par des experts
Agents conversationnels	<i>Persona Effect</i> , conviviaux	vallée dérangeante, déceptifs dans leurs capacités dialogiques et/ou métiers
Systèmes de recommandation	permettent de produire des recommandations pertinentes	nécessitent un accès aux données utilisateurs, pas adaptés à tous les domaines, peu explicatifs

TABLEAU 3.2 : Grille d'analyse des différentes approches logicielles pour assister la décision

Alors que le champ de l'aide à la décision vise à expliciter le processus décisionnel, celui des systèmes de recommandation se préoccupe généralement davantage du résultat, c'est-à-dire de la production de recommandations suivies par l'utilisateur. De nombreux systèmes de recommandation font l'hypothèse que l'on dispose préalablement d'informations sur le décideur, ou sur un ensemble de décideurs, et se focalisent davantage sur la façon d'utiliser cette information que sur celle de l'obtenir [Montaner *et al.* 2003]. De plus, les systèmes de recommandation suscitent des réserves sur des aspects de vie privée et d'anonymat des données personnelles de l'utilisateur [Ramakrishnan *et al.* 2001]. En effet, dans de nombreuses applications, la personnalisation de l'expérience pour l'utilisateur est réalisée au prix de la perte de la confidentialité

de ses données par l'utilisateur [Sheng *et al.* 2008].

L'hypothèse d'un accès à des données personnelles est beaucoup moins répandue dans le champ des agents conversationnels où c'est l'interaction qui est au centre des débats, notamment lorsqu'elle est multimodale comme dans [Poggi *et al.* 2005]. Cependant, l'incarnation de l'agent peut être à l'origine d'attentes fortes chez son interlocuteur, en particulier quand l'ACA prend un aspect anthropomorphe. Cet état de fait vient renforcer des attentes déjà fortes induites par la formulation libre et peut conduire à la frustration de l'utilisateur, faute à des compétences (métiers et linguistiques) insuffisantes [Ben Mimoun & Poncin 2010]. Ce décalage entre l'apparence de l'agent et son intelligence est à l'origine du phénomène de la vallée de l'étrange [Mori *et al.* 2012], ou autrement dit, du malaise ou de la perplexité de l'utilisateur face à de tels agents. Au contraire, les systèmes experts se concentrent sur la logique, la représentation des connaissances et les mécanismes de raisonnement. Cependant, ils relèguent souvent le dialogue avec l'utilisateur au second plan.

---

# Agrégation des préférences

---

## Sommaire

---

<b>4.1</b>	<b>Introduction</b>	<b>45</b>
<b>4.2</b>	<b>Agrégation de préférences ordinales</b>	<b>45</b>
4.2.1	Consensus	46
4.2.2	Préférences de Pareto	46
<b>4.3</b>	<b>Agrégation de préférences cardinales</b>	<b>50</b>
4.3.1	Bien-être utilitaire	50
4.3.2	Bien-être égalitaire	51
4.3.3	Bien-être de Nash	52
<b>4.4</b>	<b>Synthèse</b>	<b>53</b>

---

## 4.1 Introduction

Dans le chapitre 1, nous avons présenté les différents modèles de préférences. Ici, nous nous intéressons au calcul de préférences globales à partir d'un ensemble de préférences locales. Cette problématique d'agrégation est commune à différents domaines dont notamment la Théorie du Choix Social et la Théorie de la Décision Multi-critères. Ici, c'est en premier lieu sous l'angle de la Théorie du Choix social que nous abordons cette question. On retrouve la problématique spécifique de l'agrégation de préférences de différents individus, sous le terme de *règle de choix collectif* (en anglais, « *collective choice rule* ») employé dans [Sen 1977]. Il s'agit de proposer des fonctions d'agrégation de préférences afin d'évaluer socialement les alternatives à partir des évaluations individuelles. Nous nous intéressons aux travaux relatifs à l'agrégation sociale de préférences ordinales dans la section 4.2, puis cardinales dans la section 4.3. Dans un cas comme dans l'autre, nous faisons, comme dans le chapitre 1, l'hypothèse que les préférences des agents sont statiques. La section 4.4 conclut ce chapitre en discutant des intérêts et limites des deux approches.

## 4.2 Agrégation de préférences ordinales

Considérons ici un ensemble d'alternatives  $\mathcal{X}$  et un ensemble d'agents  $\Omega$ . Afin de ne pas formuler d'hypothèses restrictives, nous supposons que tout agent  $i \in \Omega$  est muni d'une relation de

préférence large sur  $\mathcal{X}$ , non nécessairement totale, notée  $\succsim_i$  (cf. définition 1.2 page 11).

Pour illustrer notre propos, nous nous appuyerons sur l'exemple 4.1 tout au long de cette section.

**Exemple 4.1 (Préférences larges partielles de deux agents sur un ensemble de 7 alternatives).**

Nous considérons ici un problème de décision portant sur un ensemble d'alternatives  $\mathcal{X} = \{a, b, c, d, e, f, g\}$ . Les graphes de préférences de la figure 4.1 décrivent les préférences de deux agents conformément à la définition 1.6 (page 16).

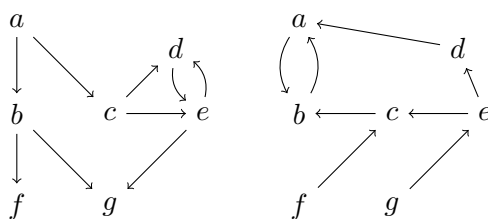


FIGURE 4.1 : Deux relations de préférence :  $\succsim_1$  (à gauche) et  $\succsim_2$  (à droite).

Intéressons nous pour commencer aux préférences qui ne font l'objet d'aucune contradiction au sein du groupe d'agents, autrement dit, aux préférences unanimes. Dans le cas trivial où il existe une ou plusieurs alternatives non-dominées pour l'ensemble des agents (cf. définition 1.8 page 18), alors on est dans un cas de consensus.

### 4.2.1 Consensus

**Définition 4.1 (Consensus).**

Soit un ensemble d'agents  $\Omega$  tel que tout agent  $i \in \Omega$  est muni d'une relation de préférences notée  $\succsim_i$  sur  $\mathcal{X}$ . On appelle **consensus** toute alternative  $x \in \mathcal{X}$  telle que :

$$x \in \bigcap_{i \in \Omega} c_{ND}(\mathcal{X}, \succsim_i)$$

L'intersection des alternatives non-dominées capture la notion de consensus, c'est-à-dire un choix multi-agents réalisé en l'absence de conflits. Cependant, l'existence d'un consensus n'est pas garantie. En effet, dans le cas général, aucune alternative ne fait l'unanimité, les agents ont des préférences plus ou moins divergentes, et sont en désaccord sur l'alternative à choisir. Une illustration de cette situation courante est proposée dans l'exemple 4.1 où l'intersection des alternatives non-dominées est vide, il n'existe donc aucun consensus dans ce cas.

### 4.2.2 Préférences de Pareto

Une approche moins restrictive pour distinguer les alternatives socialement satisfaisantes consiste à agréger les préférences des agents. Autrement dit, il s'agit de définir une unique relation de préférence à partir d'un ensemble donné de relations de préférence.



Dans le cadre de ses travaux sur le choix social, [Pareto 1897] a défini une relation de préférence sociale agréant des relations de préférence individuelles totales (voir aussi [Sen 1977], chapitre 2) qui permet de comparer des alternatives deux à deux du point de vue d'un groupe d'agents.

Selon le critère de Pareto, pour qu'une alternative  $x$  Pareto-domine une alternative  $y$ , il faut que  $x$  soit au moins aussi bonne que  $y$  pour tous les agents et qu'au moins un agent préfère strictement  $x$  à  $y$ .

**Définition 4.2 (Préférences de Pareto).**

Soit  $\Omega$  un ensemble d'agents tel que tout agent  $i \in \Omega$  est muni d'une relation de préférence totale notée  $\succsim_i$  sur  $\mathcal{X}$ . On définit la relation de préférence large de **Pareto**  $\succsim_\Omega \subseteq \mathcal{X} \times \mathcal{X}$  telle que  $\forall (x, y) \in \mathcal{X}^2$ ,

$$x \succsim_\Omega y \Leftrightarrow (\forall i \in \Omega, x \succsim_i y)$$

De cette relation de préférence, on peut facilement extraire :

- la préférence stricte de Pareto, ou **Pareto-dominance**,

$$\begin{aligned} x \succ_\Omega y &\Leftrightarrow (x \succsim_\Omega y) \wedge \neg(y \succsim_\Omega x) \\ &\Leftrightarrow (\forall i \in \Omega, x \succsim_i y) \wedge (\exists i \in \Omega, x \succ_i y) \end{aligned}$$

- et l'indifférence de Pareto,

$$\begin{aligned} x \sim_\Omega y &\Leftrightarrow (x \succsim_\Omega y) \wedge (y \succsim_\Omega x) \\ &\Leftrightarrow \forall i \in \Omega, x \sim_i y \end{aligned}$$

La relation de préférences de Pareto ne demande aucune hypothèse forte supplémentaire, elle considère simplement une alternative socialement sous-optimale s'il en existe une meilleure pour un agent sans qu'elle ne soit pire pour aucun des autres.

**Exemple 4.2 (Pareto-dominance d'une alternative sur une autre).**

Dans les préférences présentées dans la figure 4.1, l'alternative  $a$  est meilleure que l'alternative  $b$  pour l'agent 1 et l'agent 2 est indifférent à choisir l'une plutôt que l'autre. Ainsi, au niveau du groupe d'agents  $\Omega = \{1, 2\}$  l'alternative  $a$  Pareto-domine l'alternative  $b$  ( $a \succ_\Omega b$ ).

On peut remarquer que la relation de préférence de Pareto est un pré-ordre (cf. définition 1.3, page 15).

**Propriété 4.1** (Les préférences de Pareto constituent un pré-ordre). *La relation de préférences de Pareto  $\succsim_\Omega \subseteq \mathcal{X} \times \mathcal{X}$  est une relation réflexive et transitive.*

Le pré-ordre que constitue les préférences de Pareto étant en général partiel, la notion d'incomparabilité est particulièrement pertinente au niveau social. Elle exprime les conflits au sein de l'ensemble d'agents qui contredisent l'affirmation d'une préférence collective d'une alternative sur une autre.

Une alternative est considérée comme Pareto-optimale si elle n'est pas Pareto-dominée.

**Définition 4.3 (Pareto-optimalité).**

L'ensemble des alternatives **Pareto-optimales** pour  $\Omega$  sur  $\mathcal{X}$  est l'ensemble  $c_{ND}(\mathcal{X}, \succ_{\Omega})$  tel que :

$$c_{ND}(\mathcal{X}, \succ_{\Omega}) = \{x \in \mathcal{X} \mid \forall y \in \mathcal{X} \neg(y \succ_{\Omega} x)\}$$

**Exemple 4.3 (Pareto-optimalité d'une alternative).**

Dans la figure 4.1, considérons uniquement l'ensemble d'alternatives  $\mathcal{X}' = \{a, b, f\}$  de manière à avoir des relations totales. Aucune alternative n'est au moins aussi bonne que l'alternative  $a$  pour l'agent 1, par conséquent  $a$  est Pareto-optimale.

La Pareto-optimalité est un critère moins fort que le consensus. Autrement dit, tout consensus est Pareto-optimal.

**Propriété 4.2 (Pareto-optimalité des consensus).**

$$\bigcap_{i \in \Omega} c_{ND}(\mathcal{X}, \succ_i) \subseteq c_{ND}(\mathcal{X}, \succ_{\Omega})$$

La Pareto-optimalité capture la notion de rationalité multi-agents. En effet, si une alternative est Pareto-dominée, c'est parce qu'il en existe une qui lui est préférable du point de vue du groupe d'agents. De plus, l'existence d'une alternative Pareto-optimale est garantie. La Pareto-optimalité constitue donc une fonction de choix (cf. définition 1.1, page 12).

**Propriété 4.3 (Existence d'un Pareto-optimal).**

$$c_{ND}(\mathcal{X}, \succ_{\Omega}) \neq \emptyset$$

Comme indiqué, les préférences de Pareto sont définies sur des relations totales. Or, comme abordé dans le chapitre 1, certains travaux évitent de faire cette hypothèse et considèrent qu'il peut exister des paires d'alternatives incomparables pour un agent. Dans la littérature, Bouveret *et al.* proposent d'étendre les préférences de Pareto à des relations partielles [Bouveret *et al.* 2010]. Deux extensions sont alors envisagées : les *préférences nécessaires* et les *préférences potentielles*.

Les préférences nécessaires sont une extension stricte de la règle de Pareto. En effet, dans le cas où deux alternatives sont incomparables pour au moins un agent, la place est laissée au doute et les alternatives sont considérées Pareto-incomparables.

**Définition 4.4 (Préférences de Pareto nécessaires ( $\succ^{PN}$ )).**

Soit un ensemble d'agents  $\Omega$  tel que tout agent  $i \in \Omega$  est muni d'une relation de préférence notée  $\succ_i$  sur  $\mathcal{X}$ . On définit la relation de **préférence de Pareto nécessaire**  $\succ^{PN} \subseteq \mathcal{X} \times \mathcal{X}$  telle que  $\forall (x, y) \in \mathcal{X}^2$  :

$$x \succ^{PN} y \Leftrightarrow (\forall i \in \Omega, x \succ_i y) \wedge (\exists i \in \Omega, x \succ_i y)$$

Une autre extension des préférences de Pareto consiste à établir des préférences potentielles, c'est-à-dire à définir si une préférence au sens de Pareto est envisageable dans le cas où l'ensemble des agents lèvent leurs incomparabilités.

**Définition 4.5 (Préférences de Pareto potentielles ( $\succ^{PP}$ )).**

Soit un ensemble d'agents  $\Omega$  tel que tout agent  $i \in \Omega$  est muni d'une relation de préférence notée  $\succsim_i$  sur  $\mathcal{X}$ . On définit la relation de **préférence de Pareto potentielle**  $\succ^{PP} \subseteq \mathcal{X} \times \mathcal{X}$  telle que  $\forall (x, y) \in \mathcal{X}^2$  :

$$x \succ^{PP} y \Leftrightarrow (\forall i \in \Omega, \neg(y \succ_i x)) \wedge (\exists i \in \Omega, \neg(y \succsim_i x))$$

On remarque que la préférence nécessaire implique la préférence potentielle. Ces deux définitions étant posées, il est maintenant possible de la même manière de décliner deux modes de Pareto-optimalité : la Pareto-optimalité potentielle et la Pareto-optimalité nécessaire.

Pour qu'une alternative soit nécessairement Pareto-optimale, il faut qu'aucune autre ne la Pareto-domine potentiellement.

**Définition 4.6 (Pareto-optimalité nécessaire).**

L'ensemble des alternatives **nécessairement Pareto-optimales** pour  $\Omega$  sur  $\mathcal{X}$  est l'ensemble  $c_{ND}(\mathcal{X}, \succ^{PP}) = \{x \in \mathcal{X} \mid \forall y \in \mathcal{X} \neg(y \succ^{PP} x)\}$

De manière symétrique, pour qu'une alternative soit potentiellement Pareto-optimale, il faut qu'aucune autre ne la Pareto-domine nécessairement.

**Définition 4.7 (Pareto-optimalité potentielle).**

L'ensemble des alternatives **potentiellement Pareto-optimales** pour  $\Omega$  sur  $\mathcal{X}$  est l'ensemble  $c_{ND}(\mathcal{X}, \succ^{PN}) = \{x \in \mathcal{X} \mid \forall y \in \mathcal{X} \neg(y \succ^{PN} x)\}$

**Exemple 4.4 (Illustration des Pareto-optimalité nécessaires et potentielles).**

Dans l'exemple 4.1, on peut remarquer que  $b$  n'est pas potentiellement Pareto-optimale car nécessairement dominée par  $a$  ( $a \succ_1 b$  et  $a \succ_2 b$ ) et donc a fortiori,  $b$  n'est pas nécessairement Pareto-optimale. L'alternative  $c$  est potentiellement Pareto-optimale mais pas nécessairement car  $f$  la Pareto-domine potentiellement ( $c \parallel_1 f$  et  $f \succ_2 c$ ). Pour résumer, l'ensemble des alternatives potentiellement Pareto-optimales est  $\{a, c, e, f, g\}$ , soit 5 alternatives sur 7; l'ensemble des alternatives nécessairement Pareto-optimales est  $\{a, e, f, g\}$ , soit 4 alternatives sur 7.

Remarquons que tandis que l'existence d'une alternative potentiellement Pareto-optimale est garantie, ça n'est pas le cas pour la Pareto-optimalité nécessaire.

**Exemple 4.5 (Existence d'une alternative potentiellement Pareto-optimale et absence d'une alternative nécessairement Pareto-optimale).**

Soit un ensemble de deux alternatives  $\mathcal{X} = \{a, b\}$  incomparables à la fois pour l'agent 1 et l'agent 2. Aucune des deux alternatives n'est nécessairement Pareto-optimale  $c_{ND}(\mathcal{X}, \succ^{PP}) = \emptyset$  alors que toutes deux le sont potentiellement  $c_{ND}(\mathcal{X}, \succ^{PN}) = \{a, b\}$ .

### 4.3 Agrégation de préférences cardinales

Dans cette section, nous considérons un ensemble d'alternatives  $\mathcal{X}$  et un ensemble d'agents  $\Omega$  tel que tout agent  $i \in \Omega$  est muni d'une fonction d'utilité définie sur  $\mathcal{X}$ , notée  $u_i$ .

Pour illustrer notre propos, nous nous appuyerons sur l'exemple 4.6 tout au long de cette section.

**Exemple 4.6 (Fonctions d'utilité de deux agents sur un ensemble de 7 alternatives).**

Nous considérons ici un problème de décision portant sur un ensemble d'alternatives  $\mathcal{X} = \{a, b, c, d, e, f, g\}$ . Les fonctions d'utilité de deux agents 1 et 2 définies sur ces alternatives sont décrites par le tableau 4.1.

$x$	$a$	$b$	$c$	$d$	$e$	$f$
$u_1(x)$	10	5	7	6	5	7
$u_2(x)$	3	5	2	4	6	2

TABLEAU 4.1 : Fonctions d'utilité des agents 1 et 2

Nous nous intéressons à présent à l'agrégation de préférences modélisées sous la forme de fonctions d'utilité. Autrement dit, il s'agit ici d'établir une fonction de bien-être social à partir des fonctions d'utilité individuelles. Plusieurs fonctions d'agrégation des utilités des individus ont été proposées dans la littérature (voir [Moulin 2002]).

#### 4.3.1 Bien-être utilitaire

Les philosophes Bentham et Mill ont posé les fondements de la pensée utilitariste, celle-ci a pour principe « le plus grand bonheur du plus grand nombre » [Bentham 1789, Mill 1863]. Dans cette approche, classiquement, l'opérateur d'agrégation est la somme. Le bien-être social utilitaire est ainsi égal à l'addition des utilités individuelles.

**Définition 4.8 (Bien-être social utilitaire).**

Le **bien-être social utilitaire**, noté  $u_{\Omega}^{\Sigma}$ , est défini tel que :

$$u_{\Omega}^{\Sigma}(x) = \sum_{i \in \Omega} u_i(x)$$

Une alternative qui maximise le bien-être social utilitaire est Pareto-optimale.

**Théorème 4.1 (Pareto-optimalité d'un optimum utilitaire).**

Soit  $\Omega$  un ensemble d'agents et  $u_i$  la fonction d'utilité de chaque agent  $i \in \Omega$ , on a :

$$c_{max}(\mathcal{X}, u_{\Omega}^{\Sigma}) \subseteq c_{ND}(\mathcal{X}, \succ_{\Omega})$$

Comme le montre l'exemple 4.7, si un optimum social utilitaire est Pareto-optimal, il ne prend cependant aucunement en considération les inégalités entre les agents.

**Exemple 4.7 (Un optimal social utilitaire mais inégalitaire).**

Soient  $u_1$  et  $u_2$  les fonctions d'utilités respectives de deux agents définies dans le tableau 4.1. L'alternative  $a$  maximise le bien être utilitaire avec  $u_{\Omega}^{\Sigma}(a) = u_1(a) + u_2(a) = 10 + 3 = 13$  et est Pareto-optimale – aucune n'est au moins aussi satisfaisante pour l'agent 1 – cependant, elle présente le plus gros écart d'utilité entre les agents.

Une approche plus égalitaire du bien-être social est relative à l'utilité de l'agent le moins satisfait. Elle est notamment défendue dans les travaux de [Rawls 1971]. L'objectif est alors de maximiser l'utilité minimale, c'est-à-dire d'appliquer la règle dite du *maximin*.

**4.3.2 Bien-être égalitaire****Définition 4.9 (Bien-être social égalitaire).**

Le **bien-être social égalitaire**, noté  $u_{\Omega}^{-}$ , est défini tel que :

$$u_{\Omega}^{-}(x) = \min_{i \in \Omega} u_i(x)$$

**Exemple 4.8 (Illustration du critère égalitaire).**

Soient  $u_1$  et  $u_2$  les fonctions d'utilités respectives de deux agents définies dans le tableau 4.1. Les alternatives qui maximisent le bien être égalitaire sont  $b$  et  $e$  avec  $u_{\Omega}^{-}(b) = \min(u_1(b), u_2(b)) = 5$  et  $u_{\Omega}^{-}(e) = \min(u_1(e), u_2(e)) = 5$ .

Le bien-être social égalitaire se focalise uniquement sur l'utilité d'un seul agent. Ainsi, dans notre exemple,  $b$  et  $e$  sont des optimums égalitaires à utilité minimale égale. Pourtant, on remarque, que si les deux alternatives sont équivalentes pour l'agent 1,  $b$  est préférable à  $e$  pour l'agent 2. Le critère égalitaire n'est donc pas suffisamment discriminant et ne garantit pas la Pareto-optimalité. On peut cependant le raffiner en appliquant la règle du *maximin* lexicographique, encore appelée *leximin* et défendue notamment par Sen dans [Sen 1970, Sen 1976].

**Définition 4.10 (Préférences *leximin*).**

Soient  $x, y \in \mathcal{X}$  deux alternatives et  $\Omega$  un ensemble de  $n$  agents. On considère  $((1(x), \dots, n(x)))$  le vecteur des utilités des agents ordonnées de manière croissante, c'est-à-dire tel que  $1(x)$  soit l'utilité minimale pour l'alternative  $x$ ,  $2(x)$  la deuxième moins élevée et plus généralement  $j(x)$  la  $(j - i)$ ème utilité la moins élevée. On dit que :

- l'alternative  $x$  est strictement *leximin*-préférée à  $y$  (on note  $x \succ^{lm} y$ ) si et seulement si

$$\exists k \leq n, \forall i < k, i(x) = i(y) \text{ et } k(x) > k(y)$$

- l'alternative  $x$  est *leximin*-équivalente à  $y$  (on note  $x \sim^{lm} y$ ) si et seulement si

$$\forall i, i(x) = i(y)$$

Une alternative *leximin*-optimale maximise au sens lexicographique les utilités des agents les plus pauvres. Les préférences *leximin* permettent de comparer toute paire d'alternatives.

**Propriété 4.4** (La relation de préférences  $(\succ^{lm}, \mathcal{X})$  est un pré-ordre total).

Comme la relation de préférences *leximin* est un pré-ordre total, il y a toujours au moins une alternative *leximin*-optimale et la *leximin*-optimalité constitue bien une fonction de choix.

**Définition 4.11 (Leximin-optimalité).**

L'ensemble des alternatives **leximin-optimales** pour  $\mathcal{X}$  est  $c_{ND}(\mathcal{X}, \succ^{lm})$ .

Si le bien-être utilitaire agrège les utilités sans prendre en compte les potentielles inégalités, la règle du *leximin* quant à elle se focalise uniquement sur le ou les agents les plus pauvres.

**Exemple 4.9 (Illustration de la leximin-optimalité).**

Soient  $u_1$  et  $u_2$  les fonctions d'utilités respectives de deux agents définies dans le tableau 4.1. Comme montré dans l'exemple 4.8, les alternatives maximisant l'utilité minimale sont  $b$  et  $e$ . Concernant la deuxième utilité minimale, on a  $2(b) = 5$  et  $2(e) = 6$ . Par conséquent, on a  $e \succ^{lm} b$  et donc  $e$  est l'unique alternative *leximin*-optimale dans notre exemple.

Une alternative qui maximise lexicographiquement les utilités minimales est Pareto-optimale.

**Théorème 4.2 (Pareto-optimalité d'un leximin-optimal).**

Soit  $\Omega$  un ensemble d'agents et  $u_i$  la fonction d'utilité de chaque agent  $i \in \Omega$ , on a :

$$c_{max}(\mathcal{X}, \succ^{lm}) \subseteq c_{ND}(\mathcal{X}, \succ_{\Omega})$$

### 4.3.3 Bien-être de Nash

Dans [Harsanyi & Selten 1972], Harsanyi et Selten s'inspirent des travaux de Nash pour proposer un compromis entre les approches utilitaires et égalitaires avec le produit comme opérateur d'agrégation.

**Définition 4.12 (Bien-être social de Nash).**

Le **bien-être social de Nash**, noté  $u_{\Omega}^{\Pi}$ , est défini tel que :

$$u_{\Omega}^{\Pi}(x, \Omega) = \prod_{i \in \Omega} u_i(x)$$

Pour être pertinent, le calcul du bien-être de Nash suppose que les utilités exprimées par les agents sont strictement positives.

**Exemple 4.10 (Illustration du bien-être de Nash).**

Soient  $u_1$  et  $u_2$  les fonctions d'utilités respectives de deux agents définies dans le tableau 4.1. Dans ce cas de figure, deux alternatives maximisent le produit des utilités :  $a$  avec  $u_{\Omega}^{\Pi}(a, \Omega) =$

||  $u_1(x) \times u_2(x) = 10 \times 3 = 30$  et  $e$  avec  $u_{\Omega}^{\Pi}(e, \Omega) = u_1(x) \times u_2(x) = 5 \times 6 = 30$ .

Une alternative qui maximise le bien-être social de Nash est Pareto-optimale.

**Théorème 4.3 (Pareto-optimalité d'un optimum de Nash).**

Soit  $\Omega$  un ensemble d'agents et  $u_i$  la fonction d'utilité de chaque agent  $i \in \Omega$ , on a :

$$c_{max}(\mathcal{X}, u_{\Omega}^{\Pi}) \subseteq c_{ND}(\mathcal{X}, \succ_{\Omega})$$

Le tableau 4.2 résume les optimums sociaux sur les différents critères pour les fonctions d'utilité définies dans le tableau 4.1.

$x$	$a$	$b$	$c$	$d$	$e$	$f$
$u_1(x)$	<b>10</b>	5	7	6	5	7
$u_2(x)$	3	5	2	4	<b>6</b>	2
$u_{\Omega}^{\Sigma}(x)$	<b>13</b>	10	9	10	11	9
$u_{\Omega}^{-}(x)$	3	<b>5</b>	2	4	<b>5</b>	2
$\vec{u}_{\Omega}(x)$	(3, 10)	(5, 5)	(2, 7)	(4, 6)	<b>(5, 6)</b>	(2, 7)
$u_{\Omega}^{\Pi}(x)$	<b>30</b>	25	14	24	<b>30</b>	14

TABLEAU 4.2 : Coût social des alternatives selon les fonctions d'agrégation avec en gras les optimums.

## 4.4 Synthèse

Dans ce chapitre, nous avons présenté différentes visions du bien-être social. La plupart des travaux s'accordent à dire que la Pareto-optimalité est un critère qu'il faut chercher à satisfaire. La Pareto-optimalité constitue un critère d'efficacité sociale. En effet, si une alternative n'est pas Pareto-optimale, il existe une autre alternative pour laquelle le bien-être d'au moins un des agents est supérieur sans qu'aucun agent n'en pâtisse. Ainsi, dans une optique de bien-être social, la recherche d'une solution Pareto-optimale est impliquée par l'idée de plus grand bien pour le plus grand nombre, indépendamment d'autres considérations. Cependant, dès que deux agents ont des préférences opposées, la totalité ou presque des alternatives sont Pareto-optimales. Ainsi, si la Pareto-optimalité est un critère pertinent qui permet généralement d'écarter certaines alternatives sans léser aucun agent pour autant, elle est rarement suffisamment discriminante pour prendre une décision sans recourir fortement à l'arbitraire. Différentes écoles s'affrontent alors pour définir une alternative socialement bonne. Tandis que certains considèrent le bien-être social comme la somme des bien-être individuels, d'autres considèrent ce critère inéquitable et préfèrent évaluer le bien-être de la société au regard de l'agent le moins bien loti ou encore en calculant le produit des utilités individuelles. Le tableau 4.3 lesquelles des agrégateurs correspondant à ces principes mènent à une alternative Pareto-optimale ou non.

En outre, certains considèrent que les comparaisons interpersonnelles sont à proscrire. En effet, la notion de bien-être est par définition relative et donc non objectivable. Cependant, pour établir une règle de choix collectif, en particulier équitable, il peut être tentant d'utiliser des opérateurs mathématiques pour comparer et agréger des préférences numériques (utilité). Ainsi, il devient

Agrégateur	Pareto-optimalité
$u_{\Omega}^{\Sigma}$	✓
$u_{\Omega}^{-}$	✗
$\succsim^{lm}$	✓
$u_{\Omega}^{\Pi}$	✓

TABLEAU 4.3 : Agrégateurs d'utilité et Pareto-optimalité.

possible de mettre en balance les utilités des différents agents et d'établir des critères comme le critère utilitaire, égalitaire, du *leximin* ou encore le produit de Nash. Cette approche, fréquemment adoptée dans les problèmes d'allocation de ressource, se fonde sur des comparaisons interpersonnelles (voir notamment [Gilardone 2007] chap. IV sec. III). Or, on peut sérieusement s'interroger sur le bien-fondé de celles-ci. Ainsi, cette hypothèse a été source de controverses au sein des théoriciens du choix social. Arrow, Sen et Rawls notamment refusent de procéder à des comparaisons interpersonnelles et privilégient l'emploi de relations de préférence pour éviter cet écueil. De plus, comme présenté dans le chapitre 1, les relations de préférence permettent la modélisation de l'incomparabilité auquel le principe de Pareto peut-être étendu (cf. définitions 4.4 à 4.7).

Après avoir étudié ici la caractérisation des alternatives dans le cadre de préférences multiples, nous nous intéressons dans le chapitre suivant aux moyens permettant à un ensemble d'agents de prendre une décision collectivement.



---

# Processus de prise de décision collective

---

## Sommaire

---

<b>5.1</b>	<b>Introduction</b>	<b>55</b>
<b>5.2</b>	<b>Systèmes de vote</b>	<b>56</b>
5.2.1	Critère de la majorité	56
5.2.2	Méthode Condorcet	57
5.2.3	Méthode Borda	58
5.2.4	Caractérisation des systèmes de vote	59
<b>5.3</b>	<b>Négociation</b>	<b>60</b>
5.3.1	Objet de négociation	61
5.3.2	Protocoles	61
5.3.3	Modèles décisionnels d'agents	62
5.3.4	Propriétés des protocoles de négociation	63
<b>5.4</b>	<b>Synthèse</b>	<b>65</b>

---

## 5.1 Introduction

Après avoir disserté dans le chapitre 4 des différents critères d'évaluation sociale des alternatives, nous nous intéressons à présent aux moyens proposés dans la littérature afin de permettre à des agents de décider ensemble effectivement. Ainsi, dans ce chapitre, nous considérons un ensemble d'agents, deux ou plus, prenant une décision collectivement.

On distingue généralement deux familles de processus mis en œuvre par un groupe d'agents pour prendre une décision collectivement :

- les *systèmes de vote*, dans lesquels l'ensemble des acteurs se prononcent sur un ensemble d'alternatives fixé. Une règle est ensuite appliquée de manière centralisée sur l'ensemble des votes pour décider de l'alternative retenue ;
- les *processus de négociation*, dans lesquels les acteurs s'échangent des offres suivant un protocole jusqu'à ce qu'un accord soit trouvé ou que la négociation échoue.

Pour raffiner davantage, dans la littérature, la négociation est souvent divisée en deux catégories : négociation *compétitive* et négociation *coopérative*.

La négociation *compétitive* correspond à la définition donnée dans la section 2.3 (page 31), c'est-à-dire à une situation où des agents ayant des intérêts conflictuels dialoguent dans l'optique d'atteindre un accord mutuel. Un cas typique est celui d'un vendeur souhaitant tirer le plus grand profit d'une vente tandis que l'acheteur veut dépenser le moins possible.

La négociation *coopérative* est davantage à rapprocher du dialogue de délibération où il s'agit alors de dialoguer dans l'optique de résoudre un problème de manière distribuée [Davis & Smith 1983]. C'est le cas notamment quand des agents s'organisent ensemble pour essayer d'accomplir une tâche – par exemple l'exploration de l'environnement – de la manière la plus rapide.

Ainsi, tandis que dans le paradigme compétitif, chaque agent poursuit ses propres intérêts [Rosenstein & Zlotkin 1994], dans le paradigme coopératif, les agents adoptent un but commun et unique.

Selon [Wooldridge 2009], on peut distinguer deux types d'agents en fonction des objectifs visés :

- dans les cadres de la *résolution distribuée de problème* et de la *négociation coopérative*, l'objectif est commun et les agents sont dits *bienveillants* ;
- dans les cadres du *choix social* et de la *négociation compétitive*, les agents agissent uniquement selon leur propre intérêt (on parle parfois d'agents *égo-centrés* voire d'agents *égoïstes*). Il s'agit de prendre une décision de groupe malgré les divergences potentielles.

Dans ce chapitre, nous nous situons dans le second cas de figure et nous nous intéressons au choix social. Dans cette perspective, de nombreux travaux ont visé à établir un processus délibératif « socialement satisfaisant ». On distingue alors deux aspects à l'aune desquels évaluer un processus : d'une part, les caractéristiques qu'il présente et d'autre part, la qualité de la décision qu'il produit.

Nous consacrons la section 5.2 à l'étude des systèmes de vote et la section 5.3 à la négociation. Enfin, nous concluons ce chapitre par une synthèse (section 5.4), des différents processus étudiés et de leurs caractéristiques.

## 5.2 Systèmes de vote

Les systèmes de vote ont fait l'objet de nombreux travaux, notamment de part l'importance qu'ils revêtent dans les systèmes de scrutin électoraux sur le plan politique. Généralement, ces travaux font des hypothèses simplificatrices sur les préférences des agents en les considérant comme des relations strictes totales [de Borda 1781, Caritat 1785]. Dans cette section, nous utilisons aussi bien le terme d'alternative que celui de *candidat*, souvent consacré dans la Théorie du Vote.

### 5.2.1 Critère de la majorité

Un critère souvent retenu dans les systèmes de vote est le *critère de la majorité*. Pour cause, dans le choix le plus élémentaire où il y a lieu de choisir entre deux alternatives, il paraît tout à fait légitime de choisir celle qui est préférée par la majorité des agents. C'est typiquement le cas du référendum. Cependant, dans le cas général, le nombre des alternatives est supérieur à deux, par exemple dans le cas d'élections législatives ou présidentielles. Diverses règles sont pratiquées dans ces scrutins à travers le monde et les époques.

**Exemple 5.1 (Scrutin uninominal majoritaire à un tour).**

Une règle simple consiste à permettre à chaque électeur de choisir un unique candidat parmi ceux en lice puis de proclamer élu celui ayant reçu le plus grand nombre de voix. C'est par exemple le cas actuellement pour l'élection du président de la République d'Islande. Contrairement au cas élémentaire où l'alternative choisie bénéficiait d'une majorité absolue, ici, la majorité est relative. Il est alors possible qu'il existe un candidat préféré au vainqueur. Un tel cas de figure est présenté dans la table 5.1. Dans cet exemple, si les agents suivent la fonction de choix optimal (cf. définition 1.7, page 17),  $d$  est vainqueur avec une majorité relative de 3 voix contre 2 pour  $a$ , 1 pour  $b$  et 1 pour  $c$ . Cependant, une majorité absolue d'agents (4 sur 7) préfèrent à la fois  $a$ ,  $b$  et  $c$  à  $d$ .

$i$	$\succ_i$	$c_{opt}(\{a, b, c, d\}, \succ_i)$	$c_{opt}(\{a, d\}, \succ_i)$
1	$a \succ b \succ c \succ d$	$a$	$a$
2	$a \succ c \succ b \succ d$	$a$	$a$
3	$b \succ a \succ c \succ d$	$b$	$a$
4	$c \succ b \succ a \succ d$	$c$	$a$
5	$d \succ b \succ a \succ c$	$d$	$d$
6	$d \succ b \succ c \succ a$	$d$	$d$
7	$d \succ c \succ b \succ a$	$d$	$d$

TABLEAU 5.1 : Préférences d'une population de 7 agents sur l'ensemble d'alternatives  $\{a, b, c, d\}$ .

Le scrutin uninominal majoritaire à un tour est susceptible de voir élire le candidat considéré comme le plus mauvais pour une majorité.

**Exemple 5.2 (Scrutin uninominal majoritaire à deux tours).**

En France, depuis le référendum de 1962, l'élection présidentielle prend la forme d'un scrutin uninominal majoritaire à deux tours au suffrage universel direct. Au premier tour, chaque électeur peut voter pour un unique candidat, les deux candidats ayant recueilli le plus de voix sont qualifiés pour le second tour. Au second tour, chaque électeur peut voter pour un unique candidat, le candidat ayant recueilli le plus de voix est élu. Si on reprend les préférences de la table 5.1 et en considérant, là encore, que les agents suivent la fonction de choix optimal, les candidats  $d$  et  $a$  arrivent en tête avec respectivement 3 et 2 voix. Au second tour,  $a$  l'emporte avec 4 voix contre 3 pour  $d$ .

Avec le scrutin uninominal majoritaire à deux tours, on évite le pire candidat. Cependant, on peut remarquer que dans l'exemple 5.1, l'alternative  $b$  était majoritairement préférée à toute autre. Dans la seconde moitié du XVIIIème siècle, Condorcet propose de retenir ce critère.

**5.2.2 Méthode Condorcet**

Pour qu'une alternative soit un gagnant de Condorcet, il faut qu'elle soit préférée par une majorité d'agents à toute autre.

**Définition 5.1 (Gagnant de Condorcet).**

Soit un ensemble d'agents  $\Omega$  tel que tout agent  $i \in \Omega$  est muni d'une relation de préférence sur  $\mathcal{X}$  notée  $\succsim_i$ . Soit une alternative  $a \in \mathcal{X}$ , on dit que  $a$  est un gagnant de Condorcet si et seulement si  $\forall b \in \mathcal{X}, \text{card}(\{i \mid a \succ_i b\}) \geq \text{card}(\{i \mid b \succ_i a\})$

Dans [Caritat 1785], Condorcet propose un processus dans lequel chaque participant donne son classement des candidats afin de déduire le gagnant de Condorcet : la *méthode Condorcet*. Est désigné gagnant le candidat qui recueille une majorité de préférences sur tout autre.

**Exemple 5.3 (Situation avec un gagnant de Condorcet).**

Avec les préférences décrites dans le tableau 5.1, il existe, pour toute alternative, une majorité qui lui préfère l'alternative  $b$  qui est donc un gagnant de Condorcet.

Un processus de décision qui choisit le gagnant de Condorcet quand il existe est dit *Condorcet cohérent*. Condorcet remarque que le critère qu'il propose n'induit pas nécessairement une relation de préférence transitive. En effet, en effectuant le tournoi, il est possible d'arriver à une situation cyclique généralement appelée *paradoxe de Condorcet*.

**Exemple 5.4 (Illustration du paradoxe de Condorcet).**

Comme le montre la situation décrite dans le tableau 5.2, il est possible qu'au sens de Condorcet,  $a$  gagne contre  $b$ ,  $b$  gagne contre  $c$  et  $c$  gagne contre  $a$ .

Ainsi, si le gagnant de Condorcet est par définition unique, son existence n'est cependant pas garantie.

$i$	$\succsim_i$
1	$a \succ b \succ c \succ d$
2	$b \succ c \succ d \succ a$
3	$c \succ a \succ d \succ b$

TABLEAU 5.2 : Préférences d'une population de 3 agents sur l'ensemble d'alternatives  $\{a, b, c, d\}$ .

**5.2.3 Méthode Borda**

Contemporain de Condorcet, Borda a lui aussi proposé une méthode pour la prise de décision collective appelée de nos jours *méthode Borda* [de Borda 1781]. Elle consiste en un vote pondéré où, à partir de chaque classement d'électeur on calcule un score pour chaque candidat. Soit  $n$  le nombre de candidats, pour chaque classement d'électeur, on ajoute alors  $n$  points au premier candidat classé,  $n - 1$  point au second, etc. Le vainqueur est le candidat totalisant le score le plus élevé.

**Exemple 5.5 (Illustration de la méthode Borda).**

Nous considérons que chaque agent énonce son classement décrit en table 5.1. Le classement de l'agent 1 par exemple attribue 4 points à  $a$ , 3 points à  $b$ , 2 points à  $c$  et 1 point à  $d$ . En

procédant de même pour chaque classement, on arrive à un score total de 17 points pour  $a$ , 21 points pour  $b$ , 16 points pour  $c$  et 16 points pour  $d$ . Le candidat  $b$  est donc désigné vainqueur par la méthode Borda.

Il existe un grand nombre d'autres systèmes de vote. Sans entrer dans le détail de chacun d'entre eux, il nous semble cependant pertinent d'évoquer ici le *vote simple transférable* [Hare 1873], la *méthode de Coombs* [Coombs 1964] ainsi que le *vote par approbation* [Brams & Fishburn 1978].

### 5.2.4 Caractérisation des systèmes de vote

Étant donné le grand nombre de méthodes existantes, une des préoccupations principales dans la théorie du choix social est de définir un ensemble de bonnes propriétés pour définir une relation de préférence sociale<sup>1</sup>. Dans ses travaux, Arrow propose d'en retenir 5 : l'*universalité*, la *non-dictature*, l'*unanimité*, la *transitivité* et l'*indépendance*. Arrow démontre dans [Arrow 1951] que si le choix porte sur au moins 3 alternatives, il est impossible de réunir l'ensemble de ces propriétés à la fois, ce résultat est appelé *théorème d'impossibilité d'Arrow*.

#### **Théorème 5.1 (Théorème d'impossibilité d'Arrow).**

Soit  $\Omega$  un ensemble d'agents, chacun muni d'une relation de préférence  $\succsim_i$ . Soit  $\mathcal{X}$ , un ensemble d'alternatives sur lequel porte la décision collective. Si  $\text{card}(\mathcal{X}) > 2$  alors il n'existe pas de relation de préférence sociale (notée  $\succ_*$ ) satisfaisant les propriétés suivantes :

- l'*universalité* : la relation de préférence sociale doit pouvoir être définie quel que soit l'ensemble des alternatives et quelles que soient les classements des agents sur ces alternatives ;
- la *non-dictature* : la relation de préférence sociale ne doit pas dépendre uniquement d'un seul agent qui la dicterait. Plus formellement,

$$\nexists i \in \Omega, \forall (x, y) \in \mathcal{X}^2, (x \succ_i y \Rightarrow x \succ_* y);$$

- l'*unanimité ou Pareto-efficacité* : si l'intégralité des agents préfèrent une alternative  $a$  à une alternative  $b$ , la relation de préférence sociale doit suivre cette préférence (voir *préférences de Pareto*, définition 4.2 page 47) ;
- la *transitivité* : la relation de préférence sociale doit être transitive ;
- l'*indépendance vis-à-vis des alternatives non pertinentes* : la préférence sociale d'une alternative à une autre ne doit pas dépendre de la présence ou non d'autres alternatives. On peut voir l'hypothèse d'indépendance comme la généralisation au niveau social de la propriété de Chernoff (cf. propriété 1.1 page 13).

Le tableau 5.3 synthétise les propriétés du *scrutin majoritaire*, de la *méthode Condorcet* et de la *méthode Borda*<sup>2</sup>.

Parmi les 5 propriétés énoncées par Arrow, ces trois modes de prises de décision respectent l'*unanimité* et la *non-dictature*. Comme expliqué précédemment, si elle garantit certaines propriétés

1. encore appelée *fonction de choix social* dans la littérature.

2. voir aussi [https://en.wikipedia.org/wiki/Voting\\_system](https://en.wikipedia.org/wiki/Voting_system) (page consultée le 4 octobre 2014).

dont l'unicité du résultat, la *méthode Condorcet* n'est pas suffisante pour prendre une décision dans certains cas, contrairement aux deux autres qui sont universelles. Concernant la *méthode Borda*, remarquons qu'elle garantit la transitivité des préférences et permet donc d'établir un classement des alternatives, ce qui n'est pas le cas des deux autres systèmes de vote. Enfin, nous constatons qu'aucun de ces trois systèmes ne respecte l'hypothèse d'indépendance vis-à-vis des alternatives non pertinentes.

Systemes de vote → Propriétés ↓	Scrutin majoritaire	Méthode Condorcet	Méthode Borda
universalité	✓	✗	✓
unanimité	✓	✓	✓
transitivité	✗	✗	✓
indépendance	✗	✗	✗
non-Dictature	✓	✓	✓
gagnant de Condorcet	✗	✓	✗
unicité	✗	✓	✗

TABLEAU 5.3 : Propriétés des systèmes de vote

### 5.3 Négociation

Le champ de la négociation est à la croisée de l'économie et l'informatique. On peut introduire la notion par la définition générale proposée par Bussmann et Muller qui décrivent la négociation comme un processus de communication auquel prend part un groupe d'agents dans le but d'atteindre un accord mutuellement accepté sur un sujet donné [Bussmann & Muller 1992].

Comme dit en introduction de ce chapitre, nous nous situons du côté de la négociation compétitive. En effet, nous nous intéressons plus particulièrement aux situations où chaque agent est muni de ses propres préférences.

Dans [Jennings *et al.* 2001], les auteurs proposent de voir la négociation comme un processus de recherche distribuée dans un espace d'accords potentiels où les acteurs ont au moins deux capacités : faire des propositions et y répondre. Ils distinguent trois axes de réflexion pour la négociation :

- l'*objet de la négociation*, autrement dit l'ensemble des alternatives et en particulier sa structuration ;
- les *protocoles* qui régissent l'interaction entre les parties prenantes de la négociation et déterminent l'état du processus ;
- les *modèles décisionnels des agents*, il s'agit des stratégies mises en œuvre par les agents pour atteindre le meilleur accord possible tout en suivant le protocole.

Nous choisissons ici d'illustrer chacun de ces points par l'exemple du protocole de concession monotone [Rosenschein & Zlotkin 1994].

### 5.3.1 Objet de négociation

Selon le contexte, un groupe d'agents peut négocier sur différents objets tels que :

- un ensemble de candidats, de plans d'actions ou encore de buts ;
- une allocation de ressources ou de tâches, il s'agit alors d'attribuer chaque élément à un agent ;
- un appariement, il s'agit alors de constituer des groupes d'agents en fonction de leurs préférences.

De manière générale, l'objet de la négociation est défini par un ensemble ou espace d'alternatives, non-vide, qui décrit l'ensemble des résultats possibles de la décision collective. Cet ensemble d'alternatives peut être :

- *mono-attribut*, c'est-à-dire qu'il est composé d'alternatives élémentaires ;
- *multi-attributs* et porter sur plusieurs dimensions simultanément (par exemple un lieu et un horaire de rencontre). Le problème de décision devient alors multi-critères (voir section 1.5.1 page 20).

Quand l'espace des accords potentiels est défini sur  $n \geq 2$  dimensions, on parle de *négociation multi-attributs* et dans le cas contraire de *négociation mono-attribut*.

De plus, l'ensemble d'alternatives peut être *dynamique* et évoluer au cours de la négociation. Par exemple, un vendeur peut faire une offre sous forme de lot de manière à trouver un terrain d'entente. On est alors dans un cadre de *négociation intégrative*, qui consiste à créer un accord ce qui nécessite d'inventer de nouvelles alternatives au cours du processus à l'aide des historiques de précédentes négociations et d'un mécanisme d'apprentissage [Kersten 2001]. Quand l'ensemble d'alternatives est *fixé* et établi a priori, on parle de *négociation distributive* [Aknine et al. 2004]. Le but est, comme nous le verrons par la suite, d'atteindre un accord « efficace ».

#### Exemple 5.6 (Objet de négociation du protocole de concession monotone).

Dans le cas du protocole de concession monotone, les alternatives sont définies sur une seule dimension, c'est donc un protocole de négociation mono-attribut. L'ensemble des alternatives sur lesquelles négocient les agents est fixe, il n'évolue pas au cours du temps.

### 5.3.2 Protocoles

Dans la négociation compétitive, le protocole de négociation est la pièce maîtresse du processus, il définit les « règles de la confrontation » (*rules of encounter*). En d'autres termes, un protocole décrit les règles du jeu de négociation et contraint les stratégies des participants (cf. section 2.2.2, page 30).

Un protocole se caractérise par le nombre de participants qu'il autorise. Lorsque le protocole est défini pour deux agents, on le qualifie de *bilatéral* ou *bipartie*. Avec un nombre d'agents supérieur, on emploie le terme de protocole *multilatéral* ou *multi-parties*. Tandis qu'une partie des protocoles de négociation sont totalement distribuables, d'autres requièrent la présence d'un arbitre.

On peut distinguer des *protocoles simultanés* où les offres sont soumises de manière concurrentes

et des *protocoles dynamiques* où les agents prennent la parole chacun leur tour. De plus, on distingue les *protocoles symétriques* où tous les agents jouent le même rôle des *protocoles asymétriques* où les rôles des participants peuvent être différents (initiateur / répondant, commissaire priseur / enchérisseur).

En faisant le constat que la négociation peut être coûteuse en terme de temps, certains travaux comme [Kraus 2001] proposent de s'assurer que les agents ne vont pas passer trop de temps et donc d'énergie à négocier en introduisant une *date butoir* après laquelle, si la négociation n'a pas aboutie, alors le résultat est un désaccord.

### Exemple 5.7 (Description du protocole de concession monotone).

Le protocole de concession monotone est décrit par un automate à états fini (voir aussi section 2.3.2, page 33) dans la figure 5.1. À chaque tour de négociation  $t$ , chaque agent  $x \in \{i, j\}$  propose une offre ( $o_x^t$ ). Un accord est atteint (état n° 5) si l'offre d'un agent égale ou dépasse (en terme d'utilité) ce que l'autre demande :  $u_i(o_j^t) \geq u_i(o_i^t) \vee u_j(o_i^t) \geq u_j(o_j^t)$ . Si les deux offres égalent ou dépassent la demande, l'accord est sélectionné aléatoirement. En cas de désaccord, un autre tour de négociation se déroule où chaque agent a l'obligation d'offrir à l'autre agent au moins aussi bien (en terme d'utilité) qu'il ne l'a fait au tour précédent. Par exemple, l'agent  $i$  doit proposer  $o_i^t$  avec  $u_j(o_i^t) \geq u_j(o_i^{t-1})$ . Il peut soit rester ferme, c'est-à-dire proposer la même utilité à son interlocuteur, soit concéder, c'est-à-dire proposer une meilleure alternative (en terme d'utilité) pour l'interlocuteur. Si aucun des agents ne concède ( $u_j(o_i^t) = u_j(o_i^{t-1}) \wedge u_i(o_j^t) = u_i(o_j^{t-1})$ ), alors la négociation se termine par un conflit (état n° 6). Une concession consiste donc en un pas vers son interlocuteur. Le protocole de concession monotone est donc bilatéral, symétrique – aucun agent ne joue un rôle spécifique –, dynamique et sans date butoir. Enfin, il nécessite que chaque agent connaisse la fonction d'utilité de son interlocuteur pour déterminer les offres autorisées.

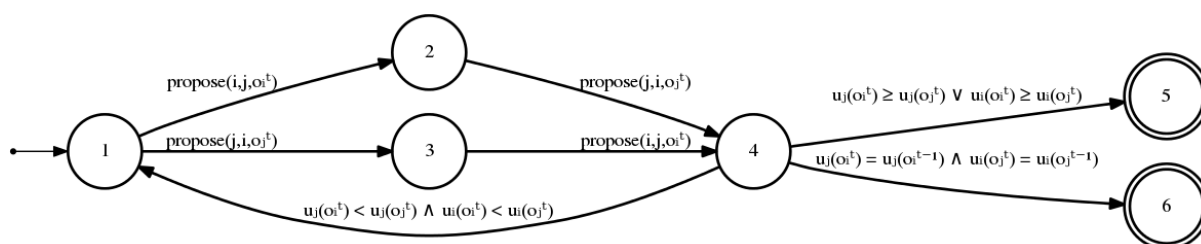


FIGURE 5.1 : Protocole de concession monotone entre deux agents  $a$  et  $b$  représenté sous la forme d'un automate fini déterministe.

### 5.3.3 Modèles décisionnels d'agents

Le modèle décisionnel d'un agent repose sur la modélisation de ses préférences ainsi que sa stratégie de négociation.

Concernant les modèles de préférence, nous en avons présenté différents types dans le chapitre 1. Les préférences des agents sont généralement modélisées par une fonction d'utilité ou par une relation de préférence.



**Exemple 5.8 (Stratégie de Zeuthen).**

La stratégie de Zeuthen [Rosenschein & Zlotkin 1994] s'interface avec le protocole de concession monotone décrit précédemment. Elle s'appuie sur les préférences de l'agent représentées par une fonction d'utilité. Cette stratégie se base sur le principe suivant : un agent qui a déjà concédé a moins à perdre en cas de désaccord (l'échec de la négociation) qu'un agent qui a fait moins de concessions. Afin d'évaluer la propension d'un agent à risquer le désaccord, le critère d'évaluation de risque de Zeuthen [Zeuthen 1930] est utilisé.

**Définition 5.2 (Facteur de risque du conflit).**

On considère  $\mathcal{X}$  un ensemble non vide d'alternatives et deux agents  $i$  et  $j$  munis des fonctions d'utilité  $u_i : \mathcal{X} \cup \{\text{disagreement}\} \rightarrow [0; 1]$  et  $u_j : \mathcal{X} \cup \{\text{disagreement}\} \rightarrow [0; 1]$  respectivement. Soient  $o_i^t, o_j^t$  les offres émises au tour  $t$  par  $i$  et  $j$  respectivement, le facteur de risque du conflit pour l'agent  $i$  au tour  $t$  (noté  $risk_i^t$ ) est défini tel que :

$$risk_i^t = \begin{cases} 1 & \text{si } u_i(o_i^t) = 0 \\ \frac{u_i(o_i^t) - u_i(o_j^t)}{u_i(o_i^t)} & \text{sinon} \end{cases}$$

Le facteur de risque pour un agent correspond au rapport entre sa perte maximale d'utilité s'il concède (c'est-à-dire s'il accepte l'offre de  $b$ ) et sa perte d'utilité en cas de désaccord (comme illustré dans la figure 5.2).

La stratégie de Zeuthen consiste à démarrer la négociation par proposer l'offre que l'on préfère. Ensuite, tant que la négociation n'est pas terminée, un agent  $i$  calcule à chaque tour son facteur de risque et celui de son opposant. Si son facteur de risque est inférieur ou égal à celui de son opposant, alors il effectue une concession minimale suffisante. Sinon, il réitère son offre précédente. Par concession minimale suffisante, on désigne l'offre qui fera basculer au prochain tour l'ordre entre les facteurs de risque (en supposant que l'opposant restera ferme) tout en transférant le moins d'utilité à l'opposant. Ainsi, une négociation entre deux agents qui adoptent cette stratégie consiste en une alternance de concessions minimales.

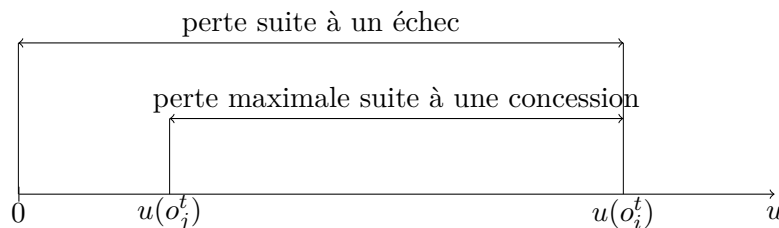


FIGURE 5.2 : Évaluation du facteur de risque pour l'agent  $i$  à l'instant  $t$

**5.3.4 Propriétés des protocoles de négociation**

Il existe une multitude de processus de négociation et on souhaite pouvoir les comparer.

Pour [Sandholm 1999], les critères pour évaluer le processus diffèrent selon le type de négociation. Dans le cas de la résolution distribuée coopérative, on évalue le processus à l'aune de la qualité sociale de la solution obtenue si les agents suivent un protocole et une stratégie imposée. Par contre, dans un système multi-agents, on considère que chaque agent dispose de sa propre stratégie, la problématique principale est alors d'établir des protocoles robustes garantissant des résultats sociaux dans l'hypothèse où les agents jouent la meilleure stratégie individuelle.

Dans [Russell & Norvig 1995], les auteurs caractérisent les mécanismes en fonction des stratégies compatibles. Lorsqu'une stratégie est *dominante*, c'est-à-dire qu'elle est gagnante contre toute autre, le protocole est dit *incitatif*. Si, de plus, cette stratégie nécessite de révéler ses préférences, elle est alors qualifiée de *vérace*.

Dans la littérature [Jennings *et al.* 2001, Rosenschein & Zlotkin 1994], on trouve de nombreux critères des protocoles, dont ceux qui suivent.

- La *garantie de terminaison* : c'est le cas si le protocole offre aux participants une garantie de se terminer tôt ou tard.
- La *garantie de succès* : c'est le cas si le protocole offre aux participants d'atteindre effectivement un accord. La garantie de succès implique donc la garantie de terminaison.
- La *rationalité individuelle* : pour que le protocole soit individuellement rationnel, il faut que pour tout agent, l'accord résultant du processus de négociation soit préférable à un désaccord.
- La *stabilité* : pour qu'un protocole soit stable, il faut qu'il incite les agents participants à se comporter selon une stratégie donnée. C'est typiquement le cas quand il existe une stratégie dominante. À défaut, on peut espérer atteindre une situation d'équilibre de Nash [Nash 1950], c'est-à-dire où aucun agent n'a intérêt à changer sa stratégie étant donnée celle des autres participants. Au contraire, si les agents peuvent se comporter de manière non désirée et ne pas aller dans le sens de l'objectif poursuivi par le protocole, ce dernier est dit manipulable.
- L'*efficacité communicationnelle* : la quantité de messages à échanger pour atteindre un accord peut constituer un critère d'efficacité à minimiser. C'est notamment le cas quand la communication engendre des coûts non négligeables ou encore quand une issue rapide est souhaitée.
- L'*efficacité computationnelle* : la puissance de calcul requise peut elle aussi constituer un critère à minimiser. Un protocole ne demandant pas une puissance de calcul considérable sera effectivement préféré, quitte dans certains cas à perdre sur la qualité sociale de l'accord obtenu.
- Le *bien-être social* : le critère est la propension du protocole à maximiser le bien-être au sein de la société d'agents. Comme indiqué dans le chapitre 4, il n'y a pas une interprétation unique de cette notion selon la forme des préférences et le principe d'agrégation adopté.
- La *Pareto-optimalité* de l'accord résultant de la négociation (cf. définition 4.2, page 47).

Les deux derniers critères portent sur le résultat tandis que les autres évaluent davantage le processus en lui-même.

Le tableau 5.4 synthétise les propriétés du protocole de concession monotone.

Protocole → Caractéristiques ↓	Concession monotone
Objet	mono-attribut
Nombre d'agents	2 (bilatéral)
Communication	dynamique
Rôles	symétrique
Distribuable	oui
Terminaison	oui
Succès	non
Rationalité individuelle	oui
Efficacité	Pareto-Nash

TABLEAU 5.4 : Propriétés du protocole de concession monotone

## 5.4 Synthèse

Dans ce chapitre, nous avons étudié et caractérisé les différents moyens offerts par la littérature pour la prise de décision collective (voir tableau 5.5). On peut les regrouper en deux grandes catégories : les systèmes de vote et les protocoles de négociation.

Dans la plupart des processus de vote comme de négociation, on fait l'hypothèse que les préférences des agents sont totales. Si les processus de vote garantissent généralement une décision dans l'anonymat, ils sont cependant centralisés et peu interactifs. Les protocoles de négociations impliquent davantage de communication et, pour une large part, demandent aux agents de révéler leurs préférences. De par ces aspects, le nombre de participants supporté par une négociation est couramment borné, contrairement au vote. Comme illustré par le protocole de concession monotone, la garantie d'un accord n'est pas systématique quand il s'agit de négociation. Tandis que la théorie du vote se focalise sur le résultat du processus, le champ de la négociation s'attache davantage au processus en lui-même.

Bien qu'on puisse distinguer système de vote et négociation, la limite entre les deux modèles n'est pas nette et ils ne sont pas à opposer. Ainsi, on peut voir le système de vote comme un protocole de négociation particulier. Néanmoins, l'inverse n'est pas vrai : il n'est pas toujours possible d'associer un système de vote à un protocole de négociation.

Processus → Caractéristiques ↓	Système de vote	Négociation
Décisif	✓	✓ / ✗
Anonyme	✓	✗
Interactif	✗	✓
Décentralisé	✗	✓
Scalabilité	✓	✗

TABLEAU 5.5 : Caractéristiques générales des systèmes de vote et de négociation

Enfin, le théorème d'Arrow nous démontre que quelque soit l'approche, il n'existe pas de processus permettant d'atteindre un résultat satisfaisant. Nous essaierons néanmoins dans le chapitre 9

de proposer un processus permettant de parvenir à une décision respectant certaines propriétés.

---

# Conclusions de l'état de l'art

---

## Sommaire

<b>6.1 Vers un agent dialogique proactif</b>	<b>67</b>
6.1.1 Modèle de préférence du décideur	67
6.1.2 Mode de dialogue homme-agent	68
6.1.3 Processus d'assistance à la décision	68
<b>6.2 Vers un processus de décision collective équitable</b>	<b>69</b>
6.2.1 Modèle de préférences des participants	69
6.2.2 Agrégation des préférences et évaluation sociale	70
6.2.3 Processus de décision collective	71

Dans les précédents chapitres, nous avons esquissé l'arrière-plan dans lequel notre contribution s'inscrit. Ce chapitre vient conclure l'état de l'art de cette thèse en présentant notre approche pour chacun des deux sujets à l'étude dans cette thèse : la prise de décision assistée et la prise de décision collective. La section 6.1 indique nos hypothèses et notre parti pris pour la conception d'un agent dialogique assistant. La section 6.2 décrit notre point de vue pour la mise en œuvre d'un processus de prise de décision collective.

## 6.1 Vers un agent dialogique proactif

Pour concevoir notre agent, nous avons à répondre à différentes problématiques :

- *Quelles hypothèses choisit-on de faire sur le modèle de préférence du décideur ?* (section 6.1.1)
- *Quelle forme choisir pour le dialogue entre notre agent et le décideur ?* (section 6.1.2)
- *Comment notre agent peut-il assister efficacement le décideur ?* (section 6.1.3)

### 6.1.1 Modèle de préférence du décideur

Comme indiqué dans le chapitre 1, de nombreux modèles de préférences du décideur aux caractéristiques diverses sont envisageables. Nous nous intéressons en particulier à un dialogue d'assistance dans un cadre de résolution de problèmes multi-attributs et considérons donc des préférences multi-critères. Nous ne faisons pas d'hypothèse restrictive quant au modèle de préférence à adopter, considérant que ce choix dépend en large partie du contexte. Ainsi, si le

but premier est d'identifier la ou les meilleures alternatives, l'approche ordinale convient. Par contre, si l'objectif est de détecter des alternatives satisfaisantes, sans nécessairement chercher à identifier les meilleures, l'approche cardinale présente l'avantage de permettre d'appliquer un seuil pour procéder à une sélection. La comparaison des paires d'alternatives n'est alors plus indispensable.

Pour choisir un modèle, il est pertinent de s'interroger sur la possibilité de dépendances inter-attributs dans les préférences du décideur. En effet, dans certains problèmes, la satisfaction du décideur est séparable attribut par attribut. Autrement dit, la satisfaction associée à la valeur d'un attribut ne dépend pas de celle des autres. Dans ces conditions, des fonctions d'agrégation linéaires (voir définition 1.17, page 23) ou des préférences lexicographiques (voir définition 1.14, page 22) peuvent être appropriées. Au contraire, s'il existe des dépendances préférentielles, des modèles tels que les CP-nets (voir définition 1.15) permettent de les capturer.

### 6.1.2 Mode de dialogue homme-agent

Idéalement, pour que le dialogue d'assistance entre notre agent logiciel et l'agent décideur humain soit le plus efficace possible, on voudrait que l'expressivité et l'intercompréhension entre les agents soit maximale. Cependant, 65 ans après le test de Turing [Turing 1950], le traitement automatique du langage naturel continue de poser des difficultés. Malgré ces difficultés, de nombreux agents conversationnels (voir section 3.4) proposent un dialogue en texte libre sans toutefois convaincre leurs interlocuteurs [Ben Mimoun & Poncin 2010]. Ainsi, plutôt que de faire nous aussi cette hypothèse et nous attaquer à ces problématiques ardues qui touchent au cœur de l'intelligence artificielle, nous prenons le parti pragmatique de limiter le périmètre du dialogue pour permettre de faciliter son interprétation.

Notre approche consiste à adopter un mode de dialogue dans lequel notre agent est proactif et dirige la conversation. Les énoncés exprimables par le décideur sont alors des réponses aux questions ou aux propositions formulées par notre agent. Bien que cette approche restreigne les possibilités d'expression du décideur, elle présente l'avantage d'assurer la compréhension des réponses qu'il a choisies. De plus, ce mode de dialogue nous semble particulièrement approprié à la problématique d'assistance dans la prise de décision dès lors que l'objet de celle-ci est identifié. En effet, on peut considérer que le décideur ne sait pas nécessairement formuler clairement ses désirs et que le rôle de l'assistant consiste aussi à faciliter ce processus.

### 6.1.3 Processus d'assistance à la décision

Dans le chapitre 3, nous avons présenté différentes formes de processus d'assistance à la décision. Comme exposé dans la section 3.6 (page 43), ces approches présentent leurs avantages et leurs limites. Nous cherchons ici à les mixer pour répondre à notre problématique.

Notre objectif est que l'agent soit capable de recommandations pertinentes et argumentées auprès du décideur. Pour mener à bien cette tâche d'assistance, notre agent pose des questions et énonce des propositions. Autrement dit, la conversation prend la forme d'une séquence de messages dans lesquels les agents s'expriment tour à tour. Dès lors, plusieurs manières de concevoir le choix des messages énoncés par notre agent sont envisageables.

Une approche naïve consiste à commencer par poser une séquence de questions fixée *a priori* puis, à partir des réponses du décideur, à recommander une ou plusieurs alternatives. Évidemment, un tel processus est linéaire et le nombre de questions peut être grand si l'on souhaite couvrir de grands ensembles d'alternatives. Ainsi, il nous semble pertinent de proposer une gestion du dialogue plus dynamique et adaptative. Dans cette perspective, une deuxième possibilité est d'utiliser un arbre de dialogue. Ce modèle consiste à brancher chaque réponse possible à une question suivante, permettant ainsi un dialogue plus dynamique. Là encore, le problème de la taille de l'arbre se pose et l'ajout d'alternatives peut requérir une modification de l'arbre. Par conséquent, nous proposons une solution plus générique qui consiste, au tour de l'agent, à calculer la question ou la proposition la plus adaptée en fonction des buts poursuivis. Pour ce faire, nous prenons le parti de doter notre agent d'une représentation du problème et d'outils de raisonnement logique du type des systèmes experts. Ces derniers présentent notamment l'avantage de justifier les recommandations. De plus, ils sont compatibles avec le mode de dialogue décrit en section 6.1.2 qui permet au dialogue d'être intuitif et convivial. Enfin, l'agent peut être personnifié comme c'est le cas pour de nombreux agents conversationnels en évitant toutefois une apparence anthropomorphique trop prometteuse.

## 6.2 Vers un processus de décision collective équitable

La mise en œuvre d'un processus de décision collective requiert de répondre à différentes questions :

- *Quel modèle de préférence considère-t-on pour nos agents ?* (section 6.2.1)
- *Comment agréger des préférences de manière à évaluer socialement une alternative ?* (section 6.2.2)
- *Quelles caractéristiques souhaitons-nous voir exhiber par notre processus de décision collective ?* (section 6.2.3)

### 6.2.1 Modèle de préférences des participants

Comme présenté dans le chapitre 1, on distingue généralement deux écoles pour représenter les préférences des agents : d'un côté, l'école cardinale qui utilise des fonctions d'utilité ; de l'autre, l'école ordinale qui leur préfère les relations binaires.

Dans une large part des travaux étudiés, en particulier dans ceux concernant le vote tels [de Borda 1781, Caritat 1785], les préférences des agents sont modélisées par des classements, c'est-à-dire des relations de préférences strictes totales. Pourtant, du point de vue d'un agent, deux alternatives peuvent être équivalentes, c'est-à-dire sembler aussi satisfaisante l'une que l'autre. On peut modéliser ce type de jugement en incluant une relation d'indifférence telle que présentée dans la définition 1.2 (page 14) ou avec une fonction d'utilité.

Si on s'en tient à ce seul critère, il est difficile alors de choisir une approche plutôt que l'autre. Cependant, comme présenté dans la section 1.6 en synthèse du chapitre 1, la fonction d'utilité présente aussi l'avantage de proposer une échelle d'intensité pour les préférences. Néanmoins, dans le contexte de la prise de décision collective, la modélisation d'utilités numériques personnelles ne nous semble pas pertinente. En effet, elle ouvre la voie à des comparaisons d'utilité

entre différents agents, encore dites interpersonnelles. Par exemple, en supposant  $u_i(x) = 4$  et  $u_j(x) = 8$ , une interprétation serait de considérer l'alternative  $x$  deux fois plus satisfaisante pour l'agent  $j$  qu'elle ne l'est pour l'agent  $i$ . Or, on peut s'interroger sur le sens de pareille comparaison entre deux évaluations subjectives. Pour cette raison, il nous semble plus prudent et adéquat de modéliser les préférences des agents par des relations binaires. De plus, contrairement aux fonctions d'utilité, celles-ci permettent de modéliser l'incomparabilité entre deux alternatives et donc une expressivité plus grande de la perception des alternatives par les agents. Nous modélisons donc les préférences de nos agents par des relations de préférences larges partielles (voir section 1.3, page 14).

### 6.2.2 Agrégation des préférences et évaluation sociale

En un premier temps, nous cherchons à identifier les alternatives visées par le processus de décision. Pour ce faire, nous souhaitons établir une relation de préférence sociale permettant de classer les alternatives et donc d'en déduire quelles sont les meilleures.

Dans le chapitre 4, nous avons étudié différentes fonctions d'agrégation des préférences proposées dans la littérature. Ayant pris le parti de ne pas faire d'hypothèse restrictive sur les préférences des agents, nous cherchons une fonction d'agrégation compatible avec des préférences larges partielles. Les préférences nécessaires et potentielles de Pareto proposent des agrégations de ces relations qui traduisent un critère d'efficacité sociale pertinent à satisfaire. Cependant, sauf à faire l'hypothèse d'agents ayant des préférences similaires, ce critère est généralement insuffisant pour décider sans recourir à l'arbitraire. Pour sortir de cet écueil, nous introduisons des valeurs numériques via une métrique de rang, ce qui nous permet d'employer les fonctions d'agrégations cardinales sans pour autant procéder à des comparaisons interpersonnelles d'utilités subjectives.

Il s'agit ensuite de choisir parmi les différents agrégateurs possibles. Il n'y a évidemment pas de critère meilleur de manière absolue, différentes doctrines philosophiques privilégieront plutôt l'un ou l'autre. Parmi les principales, on trouve la doctrine utilitariste de Mill [Mill 1863] qui préconise de maximiser le bien-être global en le considérant généralement comme la somme des utilités individuelles (cf. définition 4.8, page 50). Cependant, une telle définition peut promouvoir des alternatives maximisant le bien-être d'un agent et minimisant celui d'un autre. Souhaitant favoriser les alternatives plus équilibrées, le critère qui nous intéresse est donc celui de l'équité. Là encore, il n'y a pas consensus sur le sens à donner à la notion. Dans sa *Théorie de la Justice* [Rawls 1971], le philosophe John Rawls critique le principe utilitariste qui « voit les pertes de certains compensées par les gains des autres ». Rawls considère que dans une telle situation, la règle du *maximin*, qui consiste à baser l'évaluation du bien-être sur celui de l'agent le moins bien loti, serait adoptée. Si le contexte dans lequel est développé la théorie de Rawls est différent du nôtre, le principe d'évaluer socialement une alternative en nous basant sur le plus grand nombre d'efforts nous semble pertinent et plus équitable. Cependant, le critère du *maximin* n'assure pas la Pareto-optimalité, c'est donc le critère du *leximin* (cf. définition 4.10, page 51) que nous choisissons pour comparer socialement les alternatives.

Cette fonction d'agrégation des préférences remet en cause la propriété d'indépendance aux alternatives non pertinentes. Néanmoins, le théorème d'Arrow (voir théorème 5.1) affirme qu'il est impossible de satisfaire cette propriété en même temps que quatre autres qui nous semblent primordiales. Ainsi, comme d'autres travaux (cf. tableau 5.3), nous choisissons de ne pas respecter



ce principe.

### 6.2.3 Processus de décision collective

Comme présenté dans la section 5.4 (page 65), les systèmes de vote et les protocoles de négociation présentent chacun leurs avantages et leurs limites.

Nous souhaitons nous situer à l'intersection des deux approches, en évitant de faire des hypothèses trop restrictives. Ainsi, nous ne mettons pas de borne a priori aux nombres de participants et d'alternatives permis par notre processus. Nous voulons établir un processus distribuible qui incite dans la mesure du possible à un accord sur des alternatives respectant les critères décrits dans la section précédente. De plus, nous souhaitons que ce processus puisse, dans une version centralisée, garantir que les préférences des agents restent privées.

Comme nous cherchons à atteindre un accord qui soit équitable, il nous semble logique que notre processus le soit aussi. Nous optons donc pour un mécanisme simultané qui n'octroie pas de rôle particulier aux participants. Pour garantir la terminaison du processus en un nombre de tours finis, nous obligeons les participants à formuler au moins une offre inédite à chaque tour.

Dans la partie III, notre contribution (chapitre 9) prend la forme d'un protocole de négociation pour l'atteinte d'alternatives dont les rangs satisfont le critère d'optimalité – au sens *leximin* – sur les rangs. Ce protocole est accompagné de deux stratégies, l'une coopérative et l'autre égo-centrée. Dans le chapitre 10, nous évaluons l'efficacité de ces stratégies aux niveaux individuel et social. Puis, nous appliquons notre processus à la résolution distribuée du problème de recherche d'un point de rencontre dans un labyrinthe.



Deuxième partie

Un agent dialogique proactif pour  
assister la prise de décision



---

# Modèle d'agent dialogique proactif pour la recommandation experte

---

## Sommaire

---

<b>7.1 Introduction</b>	<b>75</b>
<b>7.2 Approche</b>	<b>76</b>
7.2.1 Problématique	76
7.2.2 Caractéristiques de l'agent dialogique proactif	77
<b>7.3 Modèle de dialogue</b>	<b>80</b>
7.3.1 Approche du dialogue	80
7.3.2 Jeu dialogique d'assistance à la décision	82
<b>7.4 Architecture d'agent</b>	<b>84</b>
7.4.1 Représentation des connaissances et raisonnement	84
7.4.2 Gestion des buts et du dialogue	86
<b>7.5 Conclusion</b>	<b>87</b>
7.5.1 Synthèse	87
7.5.2 Discussion	88
7.5.3 Perspectives	88

---

## 7.1 Introduction

Dans le chapitre 3, nous avons proposé un état de l'art des solutions existantes pour assister un utilisateur humain dans sa prise de décision. L'objet du présent chapitre est d'exposer notre contribution sur cette problématique. Elle prend la forme d'une approche et d'une architecture pour la conception d'agents dialogiques exhibant des capacités de proactivité et d'expertise dans la recommandation.

La section 7.2 présente les lignes directrices de cette contribution. Dans la section 7.3, nous décrivons notre approche du dialogue entre notre agent logiciel et le décideur. L'architecture générale de notre agent est décrite dans la section 7.4. La section 7.4.1 met le focus sur la représentation des connaissances et le raisonnement de notre agent. La section 7.4.2 est consacrée au comportement de notre agent, c'est-à-dire à la manière dont il gère ses buts et le dialogue. Enfin, une synthèse et des perspectives de notre contribution sont présentées en partie 7.5.

## 7.2 Approche

Dans cette section, nous décrivons notre approche pour concevoir un agent logiciel dédié à l'assistance d'utilisateurs dans leurs prises de décision. En guise d'illustration nous nous basons sur le scénario illustratif décrit dans l'exemple 7.1.

### Exemple 7.1 (Dialogue de recommandation experte de jeu de société).

*L'association étudiante « Dés À La Carte » organise fréquemment des activités jeux de société et propose à son public une centaine de jeux différents. Célia vient pour la première fois avec trois de ses amis, elle ne connaît pas l'ensemble des jeux disponibles mais souhaite cependant faire un choix éclairé. Yoann, fin connaisseur de l'ensemble des jeux détenus par l'association vient la conseiller, ils ont le dialogue suivant.*

1. **Yoann** : Salut, quel jeu cherches-tu ?
2. **Célia** : Je ne sais pas quoi choisir.
3. **Yoann** : À combien de joueurs vous voulez jouer ?
4. **Célia** : 4.
5. **Yoann** : Vous avez une contrainte de temps ?
6. **Célia** : Oui, 30 minutes maximum.
7. **Yoann** : Aimez-vous les jeux de stratégie ?
8. **Célia** : Je ne sais pas.
9. **Yoann** : Vous avez l'habitude de jouer à des jeux de société ?
10. **Célia** : Pas vraiment.
11. **Yoann** : Je vous suggère d'essayer le Blokus, c'est bien pour débuter et ça se joue en 20 minutes.
12. **Célia** : D'accord !

### 7.2.1 Problématique

Nous nous intéressons ici en particulier au cas d'une personne souhaitant prendre une décision mais faisant face à un déficit d'information. Plus précisément, elle peut mal connaître l'ensemble des options disponibles – de par leur nombre ou leur complexité – ou mal connaître le contexte en lui-même. Dans l'exemple 7.1, Célia est débutante et vient pour la première fois, elle ne connaît pas l'ensemble des jeux disponibles. Dans un contexte différent, on peut imaginer qu'Alice souhaite se procurer un appareil photo pour s'essayer à la photographie mais qu'elle ne sait pas du tout pour quel modèle opter dans la large gamme de ceux qui existent. Troisième cas de figure, Bob fait face à un choix d'orientation pour ses études, il cherche une spécialité de Master Informatique pour devenir un expert en intelligence artificielle mais connaît mal les formations proposées. Dans l'ensemble de ces situations, Alice, Bob comme Célia peuvent faire appel à un tiers pour les conseiller dans leur prise de décision. S'il est évident que les connaissances permettant de les aider ne sont pas les mêmes, les dialogues de conseil dans chacun de ces différents contextes comportent cependant des similarités. Ainsi, notre objectif ici est de concevoir un agent logiciel générique, qui, s'il est pourvu de l'expertise nécessaire dans le domaine visé et sur les alternatives existantes, puisse venir en aide à celui à qui se pose le problème de décision, à la manière de Yoann dans le dialogue 7.1.

Nos hypothèses pour concevoir un tel agent sont les suivantes :

- le problème de décision est multi-attributs à valeurs discrètes, c'est-à-dire que les alternatives sont des instances d'un produit cartésien (voir section 1.5.1, page 20) ;
- notre agent dispose des caractéristiques des alternatives ainsi que d'une expertise du domaine dans lequel se pose le problème de décision ;
- notre agent ne dispose, a priori, d'aucune connaissance sur le décideur.

Dans notre exemple 7.1, nous considérerons le concept **Jeu** tel que  $\text{Jeu} = \text{NbJoueursMin} \times \text{NbJoueursMax} \times \text{Temps} \times \text{Strategie} \times \text{Bruyant}$ <sup>1</sup>. Chacun des ensembles de ce produit cartésien est fini :  $\text{NbJoueursMin} = [1;10]$ ,  $\text{NbJoueursMax} = [1;20]$ ;  $\text{Temps} = \{10\text{min}; 20\text{min}; 30\text{min}; 45\text{min}; 1\text{h}; 1\text{h}30; 2\text{h}; 2\text{h}30; 3\text{h}\}$ ,  $\text{Strategie} = \{\text{faible}, \text{moyen}, \text{fort}\}$  et  $\text{Bruyant} = \{\text{oui}, \text{non}\}$ . Nous supposons que notre agent connaît les caractéristiques de chacun des jeux, par exemple  $\text{bokus} = (2, 4, 20\text{min}, \text{moyen}, \text{non})$ .

### 7.2.2 Caractéristiques de l'agent dialogique proactif

Dans notre approche, nous considérons l'agent dialogique comme un conseiller, expert d'un domaine, apte à guider l'utilisateur dans ses prises de décisions [Morge *et al.* 2011]. Contrairement aux approches classiques réactives où l'agent est attentiste et répond essentiellement aux questions posées par l'utilisateur, notre optique ici est de concevoir un agent au comportement proactif, qui va aux devants de l'utilisateur et agit en poursuivant le but de répondre à ses attentes. Ainsi, et comme nous l'avons exposé en partie dans [Delecroix *et al.* 2011], nous souhaitons que notre **Agent Dialogue Proactif** (*ADP* dans la suite du chapitre) exhibe certaines capacités, en particulier : adopter des buts, prendre des initiatives, acquérir de l'information, appliquer son expertise, s'adapter et argumenter. Ces capacités sont décrites plus en détails ci-après.

#### Adopter des buts

Assister l'interlocuteur dans sa prise de décision est une tâche cognitive. Pour la réaliser, il est pertinent que l'agent soit de type délibératif, c'est-à-dire capable d'adopter des buts et d'agir afin de les atteindre. Initialement, l'ADP adopte le but d'identifier des alternatives pertinentes. En conséquence, il poursuit les sous-butts consistant à caractériser de telles alternatives et donc à découvrir les préférences de son interlocuteur. Nous souhaitons que notre agent soit capable de poursuivre plusieurs buts en concurrence.

#### Exemple 7.2 (Buts du conseiller dans notre exemple).

*Pour agir tel que Yoann dans notre exemple, on considère que l'ADP adopte le but initial proposer(jeu) suite à la réponse de l'utilisateur (énoncé 2). Pour le résoudre, l'ADP doit identifier des jeux pertinents en fonction de leurs caractéristiques (par exemple, bokus = (2, 4, 20min, moyen, non)). À cette fin, l'ADP doit identifier les préférences de l'agent sur tout ou partie des attributs du jeu. En effet, dans notre exemple, on*

1. On pourrait ajouter de nombreuses autres caractéristiques, nous nous contentons de ces 4 là pour garder un exemple simple.

*peut considérer que Yoann poursuit successivement les sous-butts `identifier(NbJoueurs)` (énoncé 3), `identifier(Temps)` (énoncé 5), `identifier(Strategie)` (énoncé 7), `identifier(Niveau)` (énoncé 9). Son but d'identification du jeu ayant suffisamment progressé, il peut alors enfin effectuer des propositions (énoncé 11).*

### Prendre l'initiative

Afin de poursuivre activement ses buts, un ADP doit être capable de prendre l'initiative dans la conversation, diriger l'échange, imposer les sujets et le contexte dialogique. Autrement dit, plutôt que de rester passif et se tenir simplement à disposition de son interlocuteur, il doit engager et orienter le dialogue. Contrairement à l'approche classique des agents conversationnels (voir section 3.4, page 39), c'est donc l'agent logiciel qui pose des questions ou formule des suggestions et l'agent humain qui y répond.

#### **Exemple 7.3 (Prise d'initiative de l'animateur dans notre exemple).**

*Dans notre exemple, c'est Yoann qui dirige le dialogue. Il l'initie (énoncé 1) et pose des questions (énoncés 1, 3, 5, 7, 9) tandis que son interlocuteur répond (énoncés 2, 4, 6, 8, 10). Autrement dit, Yoann est ici proactif, ses actes de langage sont directifs (voir section 2.2.1, page 28). Il oriente ainsi le dialogue à chacun de ses tours de parole et mène la conversation. Le choix des énoncés produits est fait en fonction de ses buts et de leur avancement. Dans le cas présent, le désir de Célia est de trouver un jeu (énoncé 2). L'ADP lui pose des questions afin de restreindre l'espace des alternatives qui compte une centaine de possibilités. Il le questionne par exemple sur des critères comme le nombre de joueurs (énoncé 3), le temps de jeu (énoncé 5) ou encore son niveau (énoncé 9).*

### Acquérir de l'information

Comme indiqué précédemment, le but de l'ADP est de faire des suggestions pertinentes à l'utilisateur. Il doit pour cela collecter des données afin d'établir un modèle de l'utilisateur décrivant en particulier ses préférences. C'est parce qu'il est proactif et connaisseur que l'expert peut facilement acquérir les informations pertinentes dans le scénario dialogique présenté en 7.1. En effet, étant donné le désir exprimé par l'utilisateur, l'expert oriente le dialogue vers les critères pertinents pour choisir une alternative. De la même manière, nous souhaitons que notre ADP adopte le but d'acquérir ces mêmes informations et agisse en conséquence en posant les questions pertinentes. Il s'agit ensuite d'extraire des réponses du décideur l'information afin de mettre à jour son modèle du décideur et de pouvoir ensuite raisonner.

#### **Exemple 7.4 (Acquisition d'information dans notre exemple).**

*Dans notre exemple, le dialogue s'effectue en langage naturel et Yoann qui joue le rôle d'expert est capable de comprendre les énoncés produits par Célia. À chaque énoncé, il s'agit pour Yoann d'inférer les préférences de son interlocutrice à partir de ses réponses. Le cheminement de l'ADP est ainsi proche de celui d'un système expert par chaînage mixte (voir section 3.3, page 39) : un but d'identification conduit à chercher l'information manquante (par chaînage arrière) et à interroger l'utilisateur sur celle-ci. L'information fournie permet*



à son tour – par mécanisme de chaînage avant – la déduction de nouvelles informations. Ce cycle se répète jusqu'à satisfaction du but, ou constat d'échec.

Comme discuté dans le chapitre 2, l'interprétation du langage naturel présente encore à ce jour de nombreuses difficultés non dépassées. Par pragmatisme, nous ne concevons donc pas notre ADP comme étant dépendant des capacités à interpréter le langage naturel. Notre modèle doit donc pouvoir être suffisamment générique pour pouvoir être compatible avec des langages artificiels.

### Appliquer son expertise

L'ADP doit être capable de mettre à profit ses connaissances – que celles-ci soient acquises a priori ou au cours du dialogue – pour formuler des questions et suggestions pertinentes. Nous envisageons notre ADP comme un agent *cognitif* : capable de représenter et mémoriser des connaissances, de raisonner sur celles-ci et de définir des buts ainsi que des actions à effectuer pour les atteindre.

#### Exemple 7.5 (Application de l'expertise par Yoann dans notre exemple).

Dans notre exemple, l'expertise de Yoann se traduit par différents aspects. D'une part, il a une connaissance ontologique des caractéristiques du concept jeu pour effectuer un choix (ici, le nombre de joueurs, le temps, le degré de stratégie). D'autre part, il connaît les valeurs que prennent les instances existantes dans ces caractéristiques. Enfin, il pense que les jeux très stratégiques sont à éviter pour les débutants. Cette connaissance peut être représentée sous la forme d'une règle qu'on pourrait énoncer par exemple sous la forme suivante : si  $\text{niveau}(\text{debutant})$  alors  $\text{strategie}(\text{faible}) \vee \text{strategie}(\text{moyen})$ .

### S'adapter

L'ADP doit être capable de choisir ses actions en fonction de celles de son interlocuteur, c'est-à-dire de s'adapter, aussi bien dans les questions qu'il pose que dans les suggestions qu'il formule. Autrement dit, il ne doit pas suivre un script simple et linéaire. De plus, dans le dialogue, l'utilisateur n'est pas toujours à même de répondre aux questions posées par l'agent. C'est alors à l'ADP d'ajuster son comportement au profil de l'utilisateur.

#### Exemple 7.6 (Adaptabilité dans notre exemple).

Dans notre exemple, les réponses du décideur peuvent amener à résoudre un but ou un sous-but (énoncés 4, 6, 8 et 10) ou à en créer un nouveau (énoncés 2 et 8) provoquant une nouvelle demande d'information. Dans le cas où l'historique du dialogue aurait été différent, l'expert pour être efficace doit non seulement adapter ses suggestions mais il pourrait en plus adapter ses questions en fonction, d'une part, des instances disponibles et, d'autre part, des réponses du décideur. Par exemple, si Célia avait cherché un jeu en particulier et que celui-ci n'était pas disponible, il peut être pertinent de lui en proposer plusieurs ayant des caractéristiques similaires à la manière d'un système de recommandation classique.

## Argumenter

Argumenter consiste pour l'expert à justifier ses recommandations par des arguments, c'est-à-dire des informations qui défendent la pertinence de la recommandation auprès du décideur. Cette possibilité permet d'appuyer la recommandation et d'apporter au décideur de l'information pertinente pour sa prise de décision. Les arguments avancés par l'ADP sont des éléments qui interviennent dans son processus de raisonnement pour recommander l'alternative qu'ils supportent. Ainsi, là encore, l'ADP s'adapte en choisissant les arguments en fonction de l'historique du dialogue.

### Exemple 7.7 (Argumentation de Yoann dans notre exemple).

Dans notre exemple, la proposition du Blokus par Yoann est argumentée (énoncé 11). Il donne la caractéristique de temps et indique que le jeu convient pour des joueurs débutants : deux des critères collectés au fil du dialogue et qui ont conduit Yoann à proposer cette alternative.

## 7.3 Modèle de dialogue

Nous venons d'identifier les caractéristiques constituant les lignes directrices pour la conception d'un agent dialogique proactif capable de mener la conversation et ainsi d'acquérir l'information requise afin de proposer des alternatives pertinentes au décideur. Avant de présenter de manière modulaire le modèle de l'ADP dans les sections suivantes, nous exposons dans cette section notre approche du dialogue puis en définissons un modèle formel.

### 7.3.1 Approche du dialogue

Nous nous intéressons ici à la manière dont notre agent va dialoguer pour assister l'utilisateur dans sa prise de décision.

Nous proposons de distinguer deux modes de dialogue :

- Le *mode asymétrique* ou *dirigiste*, dans lequel l'un des agents, l'initiateur, dirige le dialogue tandis que le ou les autres lui répondent. On peut par exemple ranger dans cette catégorie l'interview, l'enquête de sondage ou encore l'enchère descendante, dans laquelle le commissaire priseur fait des offres successives jusqu'à ce que l'une d'elle soit acceptée.
- Le *mode symétrique*, dans lequel les agents dirigent successivement le dialogue, en prenant des initiatives et en réagissant à celles de leur(s) interlocuteur(s). Ce mode plus ouvert est celui qu'on retrouve typiquement pratiqué dans les dialogues entre des agents humains, de la conversation anodine au débat.

Pour diriger le dialogue, un participant peut employer des actes directifs (cf. section 2.2.1 page 28), typiquement des questions ou des propositions.

Appliquée à notre contexte, cette distinction permet d'envisager plusieurs approches du dialogue d'assistance à la prise de décision :

- dirigé par le décideur, celui-ci décrit son problème à l'assistant, lui demande des renseignements et des conseils, l'assistant lui répond ;
- dirigé par l'assistant, celui-ci demande au décideur ses attentes et lui fait des suggestions, le décideur lui répond ;
- symétrique, les deux agents peuvent poser des questions, changer de sujet, prendre des initiatives, etc.

Nous nous intéressons en particulier au cas du dialogue entre un agent humain et un agent logiciel. Pour communiquer, les deux agents doivent partager un langage commun (cf. chapitre 2). Ici, pour les raisons notamment présentées dans la section 2.2 (page 28), nous avons opté pour une communication à l'aide d'un langage artificiel et formel plutôt que du langage naturel.

Si l'approche symétrique est communément utilisée dans les rapports sociaux humains quotidiens, avec un agent logiciel comme c'est ici le cas, elle n'est pas sans poser des difficultés. En effet, comme discuté dans le chapitre 3, de nombreux agents conversationnels laissant une grande liberté à l'utilisateur atteignent vite leur limite en terme de gestion du dialogue et d'interprétation des énoncés. Pour cette raison, nous choisissons un mode de dialogue asymétrique, dirigé par l'agent logiciel assistant qui mène le dialogue tandis que l'agent humain décideur lui répond. Ainsi, nous pouvons restreindre le champ dialogique et assurer une meilleure compréhension des énoncés.

Puisqu'il mène la conversation, notre agent doit être capable d'émettre en particulier des actes directifs afin de *demande une information* ou d'effectuer une *proposition* (cf. figure 7.1). Ainsi, il peut restreindre le champ d'expression du décideur à un choix parmi un ensemble d'actes essentiellement assertifs et déclaratifs que notre agent est capable d'interpréter. Ce mode de dialogue peut-être encadré par un *protocole*.

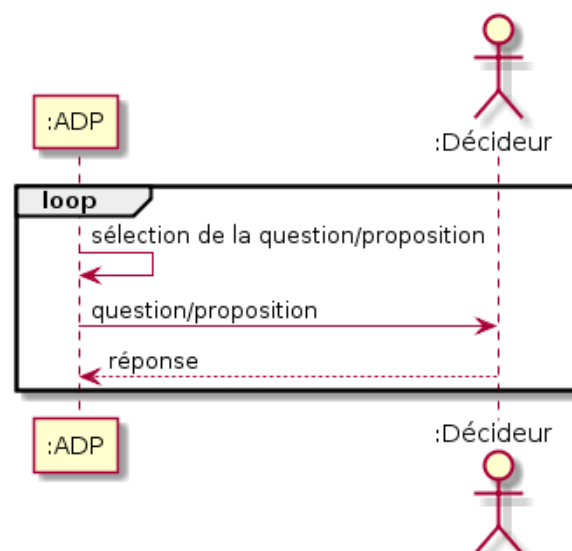


FIGURE 7.1 : Diagramme d'interaction entre l'ADP et le décideur

### 7.3.2 Jeu dialogique d'assistance à la décision

Nous définissons à présent un jeu dialogique d'assistance à la décision segmenté en tours de paroles. Un jeu de dialogue reflète la formalisation des interactions sociales entre les deux joueurs au cours du dialogue (voir section 2.3.3, page 35). Les règles du jeu définissent les énoncés ou coups que peuvent jouer chacun des deux participants. Pour définir notre jeu de dialogue d'assistance à la décision, nous nous basons sur la définition 2.1 (page 35).

#### Définition 7.1 (Jeu de dialogue d'assistance à la décision).

Soit  $ACL$  un langage de communication d'agents (cf. définition 2.2.2, page 29). Un **jeu de dialogue d'assistance** est un tuple  $JD = (P, \mathcal{M}, T, \text{protocol}, H)$  avec :

- $P = \{\text{initiator}, \text{participant}\}$ , l'ensemble des joueurs qui prennent part au jeu de dialogue : l'initiateur et le participant ;
- $\mathcal{M} \subseteq ACL$ , l'ensemble des actes de langage bien formés ;
- $T: H \rightarrow P$ , la fonction de distribution du tour de parole. Ici, les agents parlent chacun leur tour à commencer par l'initiateur. Ainsi, si la taille de l'historique  $h$  est nulle ou paire,  $T(h) = \text{initiator}$  sinon  $T(h) = \text{participant}$  ;
- $\text{protocol}: H \rightarrow 2^{\mathcal{M}}$ , la fonction correspondant au protocole qui détermine les coups légaux qui sont autorisés pour étendre l'historique ;
- $H$ , l'ensemble des historiques, c'est-à-dire des séquences de coups en conformité avec la fonction de tour de parole  $T$  et le protocole  $\text{protocol}$ .

Ici, dans notre jeu de dialogue d'assistance à la décision, le protocole général du dialogue ( $\text{protocol}$ ) est décrit par l'automate fini déterministe représenté en figure 7.2. Dans notre exemple, **Yoann** joue le rôle d'initiateur et **Célia** celui de participant. La conversation consiste en une succession de sous-dialogues de demande d'information (*query-inform*) et de sous-dialogues de délibération (*propose-accept/reject proposal*) (voir section 2.3.1, page 31). Elle prend fin lorsque le participant accepte une proposition ou lorsque l'initiateur abandonne. Chaque historique de dialogue  $h \in H$  ainsi produit est une séquence de coups bien formés respectant le protocole, lorsque le dialogue est clos, on a  $\text{protocol}(h) = \emptyset$ .

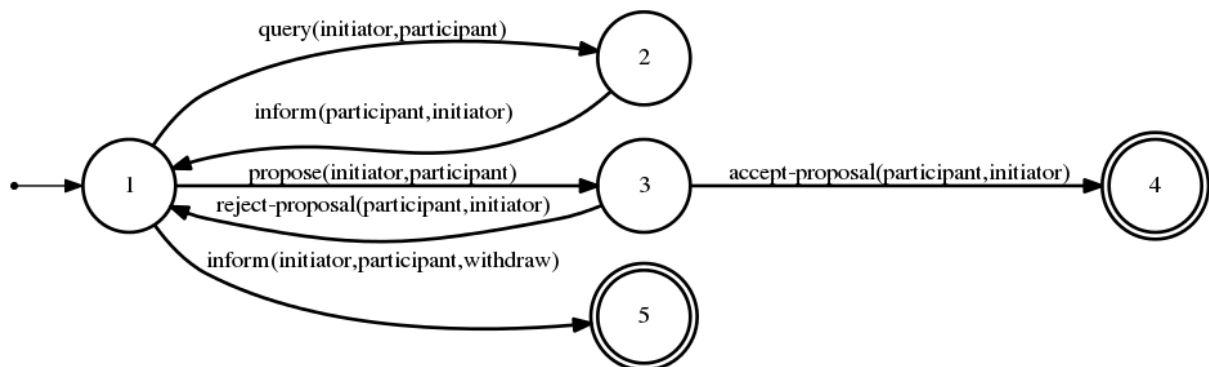


FIGURE 7.2 : Automate fini déterministe décrivant le protocole général d'un dialogue d'assistance à la décision

À chaque tour de parole, un participant envoie un message. Conformément aux termes classiquement employés dans la littérature (voir chapitre 2), tandis que l'émetteur d'un message le génère et l'envoie, son récepteur le reçoit et l'interprète. Nous basons notre définition d'un message sur le standard FIPA-ACL (voir section 2.2.2, page 29).

**Définition 7.2 (Message).**

Un message  $M_k$  est défini par  $M_k = (A_k, E_k, D_k, C_k, L_k, P_k, I_k, R_k)$  avec :

- $A_k = \text{performative}(M_k)$  l'acte de langage, c'est-à-dire une locution parmi celles possibles : *query*, *inform*, *propose*, *accept-proposal* et *reject-proposal* ;
- $E_k = \text{sender}(M_k)$ , l'agent expéditeur du message ;
- $D_k = \text{receiver}(M_k)$ , l'agent destinataire du message ;
- $C_k = \text{content}(M_k)$ , le contenu de l'acte de langage ;
- $L_k = \text{language}(M_k)$ , le langage dans lequel le contenu est exprimé ;
- $P_k = \text{proto}(M_k)$ , le protocole de dialogue utilisé ;
- $I_k = \text{reply-with}(M_k)$ , l'identifiant du message ;
- $R_k = \text{in-reply-to}(M_k)$ , l'identifiant du message auquel le message  $M_k$  répond ( $\theta$  dans le cas où le message ne répond à aucun message précédent).

On peut, à l'aide de cette définition, formaliser les énoncés du dialogue de notre exemple.

**Exemple 7.8 (Formalisation de notre scénario de dialogue d'assistance).**

Le dialogue de l'exemple 7.1 (page 76) peut être formalisé de la manière suivante :

- $M_1 = (\text{query}, \text{Yoann}, \text{Célia}, \text{jeu}(X), \mathcal{L}, \text{assist}, 1, \theta)$
- $M_2 = (\text{inform}, \text{Célia}, \text{Yoann}, \text{jeu}(\text{inconnu}), \mathcal{L}, \text{assist}, 2, 1)$
- $M_3 = (\text{query}, \text{Yoann}, \text{Célia}, \text{nbJoueurs}(X), \mathcal{L}, \text{assist}, 3, \theta)$
- $M_4 = (\text{inform}, \text{Célia}, \text{Yoann}, \text{nbJoueurs}(4), \mathcal{L}, \text{assist}, 4, 3)$
- $M_5 = (\text{query}, \text{Yoann}, \text{Célia}, \text{temps}(X), \mathcal{L}, \text{assist}, 5, \theta)$
- $M_6 = (\text{inform}, \text{Célia}, \text{Yoann}, \text{temps}(30\text{min}), \mathcal{L}, \text{assist}, 6, 5)$
- $M_7 = (\text{query}, \text{Yoann}, \text{Célia}, \text{strategie}(X), \mathcal{L}, \text{assist}, 7, \theta)$
- $M_8 = (\text{inform}, \text{Célia}, \text{Yoann}, \text{strategie}(\text{inconnu}), \mathcal{L}, \text{assist}, 8, 7)$
- $M_9 = (\text{query}, \text{Yoann}, \text{Célia}, \text{niveau}(X), \mathcal{L}, \text{assist}, 9, \theta)$
- $M_{10} = (\text{inform}, \text{Célia}, \text{Yoann}, \text{niveau}(\text{débutant}), \mathcal{L}, \text{assist}, 10, 9)$
- $M_{11} = (\text{propose}, \text{Yoann}, \text{Célia}, \text{jeu}(\text{blokus}) [\text{niveau}(\text{débutant}), \text{temps}(20\text{min})]^a, \mathcal{L}, \text{assist}, 11, \theta)$
- $M_{12} = (\text{accept}, \text{Célia}, \text{Yoann}, \text{jeu}(\text{blokus}), \mathcal{L}, \text{assist}, 12, 11)$

Le premier message ( $M_1$ ) est une question (*query*) expédiée par l'agent **Yoann** et destinée à l'agent **Célia**. Cette question a pour objet la valeur de  $X$  dans le prédicat  $\text{jeu}(X)$  exprimé dans le langage  $\mathcal{L}$ . Le message suit notre protocole d'assistance à la prise de décision (*assist*), a l'identificateur 1 et ne répond à aucun message précédent.

La connaissance est représentée ici à l'aide d'un langage logique du premier ordre où les prédicats représentent les caractéristiques des jeux ainsi que des joueurs. Par exemple,

- $\text{niveau}(X)$  est un prédicat représentant le niveau moyen des joueurs et  $\text{nbJoueurs}(X)$  leur nombre ;

- $temps(X)$  représente le temps maximal que les joueurs ont devant eux pour jouer ;
- $jeu(blokus)$  représente le choix du Blokus comme jeu.

a. on choisit ici de considérer que pour une proposition, les arguments sont listés entre crochets directement à la suite de l'alternative proposée.

Nous considérons que notre ADP a connaissance de l'ontologie du domaine sujet du dialogue. Une *ontologie* décrit un ensemble de concepts existants et de relations entre ces concepts. Dans un premier temps on considère les concepts comme définis par l'ensemble de leurs attributs. Un *attribut* est désigné par un nom unique et a un type. Un *type* est un ensemble de valeurs symboliques ou numériques, on associe à un attribut une valeur de son type.

Comme indiqué dans la définition 1.12 page 21, on appelle *instance* tout objet concret existant et étant de la classe d'un *concept*. Une *instance* associe une valeur à chacun des attributs du concept auquel elle appartient.

Tout ou partie de l'ontologie est utilisée dans le dialogue et doit donc être partagée entre l'ADP et son interlocuteur. Les messages envoyées par l'ADP peuvent ainsi être compris par l'interlocuteur. Et réciproquement, les réponses de ce dernier sont quant à elles contraintes par l'ADP.

## 7.4 Architecture d'agent

L'architecture générale de l'agent se divise en plusieurs couches : la couche de représentation des connaissances et de raisonnement de l'agent et la couche de gestion du dialogue et des buts de l'agent.

### 7.4.1 Représentation des connaissances et raisonnement

Cette section décrit comment sont représentées et organisées les connaissances de l'ADP ainsi que les mécanismes de raisonnement mis en œuvre par celui-ci.

On distingue trois types de connaissances de l'ADP :

- l'ensemble des alternatives, avec pour chacune d'entre elles les valeurs prises par leurs caractéristiques ;
- les *règles métiers* qui décrivent les connaissances de l'ADP sur le domaine ;
- les *faits* qui décrivent du point de vue de l'ADP les données du contexte du problème de décision donné.

L'ensemble des alternatives tout comme les règles métiers sont connues par l'ADP dès le début du dialogue avec le décideur et restent fixes au cours de celui-ci. Au contraire, l'ensemble des faits dépend du contexte et évolue au fil du dialogue. Ainsi, alors que les faits permettent à l'ADP de représenter un problème donné, les règles constituent son expertise, sa connaissance générale applicable à n'importe quel problème portant sur le même concept. Les connaissances sont ainsi regroupées en deux bases : une base de faits et une base de règles (voir section 3.3). Tandis que la base de faits est propre à une situation de dialogue et est mise à jour dynamiquement après chaque acquisition d'information, la base de règles est commune à l'ensemble des instances de dialogue. Les faits peuvent porter directement sur le *profil* du décideur (par exemple sa taille,

son âge, ses goûts, etc.) ou sur des éléments extérieurs (par exemple la météo, l'heure, etc.).

Le modèle de représentation des connaissances définit la forme sous laquelle sont mémorisées et traitées les connaissances par notre ADP. Différents modèles de représentation des faits et des règles sont envisageables. Comme nous visons à établir un modèle générique, nous ne formulons pas ici d'hypothèses particulières sur la représentation des connaissances. On considère ainsi que peuvent être implémentées des représentations sous forme de propositions ou de prédicats, incluant des aspects probabilistes ou non. Bien entendu, le choix du mode de représentation n'est pas anecdotique et détermine en partie les capacités cognitives de l'agent. En effet, les différents modèles permettent de représenter l'information sous différentes formes, plus ou moins complexes et délimitent le champ dans lequel l'agent peut raisonner. Enfin, utiliser un même langage pour la représentation interne des connaissances ainsi que pour le dialogue présente l'avantage de simplifier l'implémentation et le traitement des messages.

#### Exemple 7.9 (Règles métiers de Yoann).

Dans notre exemple, on peut choisir de modéliser la connaissance de **Yoann** en s'appuyant sur le langage décrit dans l'exemple 7.8. On peut alors considérer que la base de règles de **Yoann** inclut les règles suivantes (mais pas uniquement) :  $\mathcal{R} =$

- $niveau(débutant) \rightarrow strategie(faible) \vee strategie(moyen)$
- $niveau(expérimenté) \vee niveau(afficionados) \rightarrow strategie(moyen) \vee strategie(élevé)$
- $nbSallesDispo(0) \rightarrow bruyant(non)$

Les règles d'inférence sont chargées à l'initialisation du dialogue. Au contraire de la base de règles, la base de faits évolue au cours du dialogue.

#### Exemple 7.10 (Contenu de la base de faits de Yoann au cours du dialogue).

Dans notre exemple, étant donné le dialogue formalisé en exemple 7.8 et la base de règles décrite dans l'exemple 7.9, on peut modéliser la base de faits  $\mathcal{B}_i$  de **Yoann** au tour  $i$  comme suit :

- $\mathcal{B}_1 = \{nbSallesDispo(0), bruyant(non)\}$  ;
- $\mathcal{B}_5 = \mathcal{B}_1 \cup \{jeu(inconnu), nbJoueurs(4), nbJoueursMin \leq 4, nbJoueursMax \geq 4\}$  ;
- $\mathcal{B}_{11} = \mathcal{B}_5 \cup \{tempsMax(30min), Temps \leq 30min, niveau(debutant), strategie(faible) \vee strategie(moyen)\}$ .

Au début du dialogue, **Yoann** sait qu'aucune salle n'est disponible et infère donc comme critère que le jeu ne doit pas être bruyant. Après les deux premières réponses de **Célia**, **Yoann** sait de plus que le nombre de joueurs est de 4, il le traduit sous la forme de critères sur les nombres de joueurs minimum et maximum. Après les énoncés suivants, **Yoann** ajoute un critère de temps et de niveau de stratégie sur le jeu.

Pour raisonner sur ses connaissances, l'ADP utilise un mécanisme de chaînage avant (voir 3.3, page 39). Le *chaînage avant* est le processus consistant à appliquer les règles d'inférence afin de déduire de nouveaux faits. Il consiste à :

1. considérer tout nouveau fait ajouté à la base ;



2. évaluer si des conditions sont satisfaites par l'ajout de ce fait ;
3. si des conditions sont satisfaites, ajouter à la base de faits les nouveaux faits en conclusion.

En appliquant les règles d'inférence, le chaînage avant permet d'obtenir des faits à partir d'autres faits, autrement dit d'inférer. Ce mécanisme s'exécute tant que des règles peuvent s'appliquer.

## 7.4.2 Gestion des buts et du dialogue

Un agent proactif se définit par distinction avec un agent purement réactif. Il poursuit des buts et prend l'initiative d'agir pour les atteindre. Comme son nom l'indique, notre agent dialogique proactif résout donc ses buts au moyen du dialogue.

### Modélisation des buts

Les buts d'un agent assistant peuvent avoir de nombreux objets différents selon le contexte. On peut cependant distinguer deux types de buts spécifiques à la tâche de recommandation : les buts d'accord et les buts d'identification.

Le décideur souhaite choisir une alternative au sein d'un ensemble et notre ADP l'assiste dans cette tâche. Le but premier de l'ADP est donc d'obtenir un accord du décideur sur une alternative<sup>2</sup>.

Un *but d'accord* est paramétré par un concept. Il s'agit, étant donné ce concept de trouver une alternative – instance disponible de ce concept – que l'utilisateur accepte comme solution. Afin de répondre à ce but, l'ADP peut chercher à acquérir des informations concernant le décideur. En effet, s'il dispose d'informations sur ce que cherche le décideur, notre ADP pourra proposer des alternatives plus adéquates et donc plus pertinentes. Le but d'accord induit alors un but d'identification portant sur le profil du décideur.

Un *but d'identification* consiste à attribuer une valeur à un ou plusieurs éléments modélisant le problème. Dans notre contexte, l'assistant souhaite identifier le profil du décideur pour ensuite sélectionner et recommander des alternatives pertinentes. Selon le modèle de profil employé, ce but d'identification peut être décomposé en sous-but. Le but d'identification peut ainsi être paramétré par un littéral ou par un élément plus complexe, comme un concept. Enfin, un but d'identification ne porte pas nécessairement sur les caractéristiques du concept qui est l'objet du problème de décision. En effet, il peut aussi porter sur d'autres données connexes permettant d'inférer les informations utiles à la recommandation d'alternatives pertinentes.

#### Exemple 7.11 (Buts de Yoann au début du dialogue).

*Au début du dialogue, Yoann a pour but de trouver un jeu pour Célia. Autrement dit, l'ensemble des buts de Yoann est  $\{\text{accord}(\text{Jeu})\}$ .*

Pour atteindre ses buts, un agent poursuit une stratégie qui lui permet de choisir les coups de dialogue à jouer.

2. ou à défaut d'alternative satisfaisante d'abandonner et d'en informer le décideur.



## Stratégie

Dans l'état 1 du protocole de dialogue (cf. figure 7.2), la *stratégie* d'un ADP définit, étant donné un ensemble de buts, quel message envoyer à l'interlocuteur dans l'optique de progresser dans leur poursuite.

La stratégie s'interface avec le jeu de dialogue à travers les mécanismes d'énonciation des coups. À tout moment du dialogue, l'initiateur (*initiator*) est capable d'énoncer une question (*query*), une proposition (*propose*) ou d'abandonner (*inform(withdraw)*). C'est la stratégie de l'ADP qui remplit le rôle de choisir parmi l'ensemble des locutions et contenus légaux.

**Notation 7.1** (Stratégie). *Soit  $JD = (P, \mathcal{M}, T, \text{protocol}, H)$  un jeu de dialogue et  $M_p$  l'ensemble des coups bien formés légaux. On appelle stratégie d'un agent et on note  $s$  toute fonction qui sélectionne un coup de dialogue légal quand c'est au tour de l'agent de parler,  $s : H \times \mathcal{G} \rightarrow M_p$  avec  $H$  le domaine de définition des historiques et  $\mathcal{G}$  celui des buts.*

## Tactique

Afin de permettre des comportements proactifs variés et en particulier adaptatifs, notre agent est capable de gérer ses buts au fil du dialogue, d'en instancier des nouveaux ou encore de redéfinir leurs priorités. Nous appelons tactique ce niveau de raisonnement. Ainsi, après avoir appliqué sa stratégie de dialogue, joué un coup puis reçu le coup de réponse du décideur, l'ADP peut évaluer ses buts et sa stratégie et éventuellement les redéfinir.

### Exemple 7.12 (Tactique et stratégies de Yoann dans notre exemple).

*La tactique de Yoann consiste à instancier un but d'identification des caractéristiques recherchées par Célia sur le concept de Jeu. Pour ce faire, Yoann cherche à identifier les critères sur le nombre de joueurs, puis sur le temps et enfin sur le niveau de stratégie. Une fois l'ensemble des critères synthétisées, Yoann peut alors poursuivre son but d'accord en proposant une alternative qui satisfait leur conjonction. Ici, le **blockus** satisfait l'ensemble des critères, il est sélectionné par Yoann qui le propose alors en mettant en avant l'argument de profil **niveau(débutant)** et l'argument de caractéristique de l'alternative **Temps (Blockus, 20min)**.*

## 7.5 Conclusion

### 7.5.1 Synthèse

Dans ce chapitre, nous avons commencé par caractériser un agent dialogique proactif (ADP) à travers un ensemble de capacités : adopter des buts, prendre l'initiative, acquérir de l'information, appliquer son expertise, s'adapter et argumenter. Nous avons ensuite proposé un modèle de dialogue dirigé par l'assistant sous la forme d'un jeu régi par un protocole. Puis, nous avons dessiné une architecture servant de cadre à la conception d'un ADP. Celle-ci est générale et rend possible un large panel d'implémentations à différents niveaux comme la représentation des connaissances, le moteur de raisonnement ou encore la stratégie.

La représentation des connaissances et le raisonnement de l'ADP s'articulent autour de deux mécanismes liés. Un mécanisme de raisonnement dialogique, qui consiste pour l'ADP à établir, étant donné les règles et faits dont il dispose, quelle question est la plus pertinente à poser pour progresser dans un but d'identification. Un mécanisme de sélection, qui consiste à trouver quelles sont les alternatives pertinentes à partir des informations collectées dans le dialogue avec le décideur. La mise en œuvre de ces mécanismes relève de la stratégie de l'ADP qui est le processus général de production de coups par l'agent à son tour. Ainsi, des stratégies génériques peuvent être envisagées, c'est notamment le cas dans le chapitre 8 où nous proposons des stratégies de vente indépendamment du type de biens ou de services proposés. En outre, dans [Delecroix *et al.* 2012a], nous avons proposé une heuristique pour collecter des préférences de type lexicographiques (cf. définition 1.14, page 22) en minimisant la longueur du dialogue.

### 7.5.2 Discussion

En optant pour un jeu de dialogue dirigé par l'assistant, nous permettons à ce dernier de gérer le dialogue et donc de mettre en œuvre une stratégie pour atteindre son ou ses buts. Ce parti pris contextualise et limite le domaine des énoncés exprimables par le décideur. Ainsi, nous évitons l'écueil souvent pris d'une capacité d'expression totale pour le décideur humain mais d'une capacité de compréhension bornée voire faible de l'assistant logiciel. Enfin, en permettant à l'expert de diriger la recherche d'une solution et donc le dialogue, on peut penser aboutir à un processus relativement rapide et pertinent si la stratégie mise en œuvre est efficace.

### 7.5.3 Perspectives

Les perspectives de recherche de notre approche d'assistance dialogique proactive sont nombreuses et variées. Parmi celles-ci se pose la problématique délicate de l'évaluation. S'il nous semble que notre proposition est trop générale pour être évaluée expérimentalement, il est possible d'en proposer différentes instances concrètes sur un même problème donné et de les évaluer comparativement.

Au niveau du dialogue, le jeu peut être étendu de manière à permettre de poser des questions au décideur sur ses motifs de refus d'une alternative. Cela est possible avec la proposition présentée ici à condition que le langage partagée par les deux agents le permette. Cette question du langage est cruciale dans le dialogue entre un agent logiciel et un agent humain. On peut envisager des langages plus ou moins expressifs et complexes de description du contenu des actes. Cependant, au plus le nombre d'énoncés possibles est grand, au plus la fonction permettant leur traduction en langage naturel sera lourde à mettre en œuvre.

La mise en œuvre de l'ADP sur une application donnée pose aussi des questions de méthodologie et d'implémentation. Il s'agit notamment de trouver des moyens efficaces pour capturer une expertise ou un comportement et les intégrer ensuite au niveau logiciel. Pour ce faire, on peut envisager de proposer des outils. Par exemple, un outil permettant d'évaluer le taux de couverture des alternatives par les historiques de dialogue possibles : si une alternative n'est jamais proposée par un agent quels que soient les coups joués, on peut y voir soit un manque de pertinence de l'alternative, soit un déficit d'expertise de l'agent.

Enfin, concernant les capacités cognitives de l'ADP, les possibilités sont là encore nombreuses.

On peut par exemple chercher à l'interfacer avec un système de recommandation classique. Au-delà, une piste intéressante est celle d'un ADP apprenant au fil des dialogues, à la fois sur le plan des règles métiers mais aussi au niveau de sa stratégie.



---

# Application : Vendeur Virtuel Ubiquitaire

---

## Sommaire

---

<b>8.1</b>	<b>Introduction</b>	<b>91</b>
<b>8.2</b>	<b>Problématique applicative</b>	<b>92</b>
8.2.1	Contexte et objectifs	92
8.2.2	Analyse des mécanismes de vente	93
<b>8.3</b>	<b>Représentation des connaissances et raisonnement</b>	<b>93</b>
8.3.1	Produit	93
8.3.2	Client	94
8.3.3	Calcul du vecteur courant	96
<b>8.4</b>	<b>Gestions des buts, stratégie et tactique</b>	<b>97</b>
8.4.1	Identification du profil du client	98
8.4.2	Identification du produit	101
8.4.3	Comportement d'agent	105
<b>8.5</b>	<b>Implémentation</b>	<b>105</b>
<b>8.6</b>	<b>Conclusion</b>	<b>107</b>
8.6.1	Synthèse	107
8.6.2	Perspectives	107

---

## 8.1 Introduction

Ce chapitre présente une partie des travaux réalisés dans le cadre du projet **Vendeur Virtuel Ubiquitaire**, labellisé par le Pôle de compétitivité des industries du commerce (PICOM) de Lille. Ce projet s'est déroulé de 2009 à 2013 et a rassemblé différents acteurs de la recherche et de l'industrie parmi lesquels *Skema*, *Oxylane*, *Orange Labs*, *Nexstage*, le *LIFL*, *Inria*, l'*École des Mines de Douai*, *Becquet* et *Artificial Solutions*. Ma participation au projet a commencé durant mon stage de Master, en avril 2010, et a pris fin en mai 2012 avec la livraison des spécifications techniques. L'objectif du projet était de développer de nouveaux mécanismes de vente en ligne visant à augmenter le taux de transformation sur les sites de commerce électronique. L'approche adoptée consiste à rapprocher l'expérience d'achat en ligne de celle du magasin en

proposant un vendeur virtuel ubiquitaire (appelé *VVU* dans ce document) notamment à destination des internautes. Idéalement, ce programme doit posséder des qualités similaires à celles d'un conseiller-vendeur d'un magasin. Nous avons appréhendé cette problématique comme un cas d'application particulier de notre agent dialogique proactif présenté dans le chapitre 7.

Dans la section 8.2, nous exposons la problématique applicative qui nous intéresse dans ce chapitre. Nous décrivons ensuite les spécifications réalisées pour le *VVU* en suivant l'architecture d'agent proposée dans le chapitre précédent. Ainsi, la section 8.3 présente les modes de représentation de connaissances et de raisonnement du *VVU* et est à mettre en parallèle avec la section 7.4.1. Dans la continuité, la section 8.4 instancie la section 7.4.2 en décrivant comment le *VVU* gère ses buts et les stratégies et tactiques qu'il adopte pour les atteindre. La section 8.5 décrit l'implémentation du prototype du *VVU*. Enfin, la section 8.6 conclut ce chapitre en dessinant quelques perspectives.

## 8.2 Problématique applicative

Dans cette section, nous présentons plus en détails le contexte et les objectifs du projet *VVU* puis étudions les spécificités du commerce à travers une analyse des mécanismes de vente.

### 8.2.1 Contexte et objectifs

Le projet *VVU* a pour contexte particulier le domaine du commerce électronique, secteur économique en plein essor ces 15 dernières années. L'objectif est la mise en œuvre d'outils logiciels pour les sites de e-commerce dans le but d'améliorer le taux de transformation, c'est-à-dire le ratio du nombre de ventes sur le nombre de visites, ainsi que la satisfaction client.

Pour y parvenir, deux pistes ont principalement été envisagées :

- l'amélioration de l'expérience de navigation, en la rendant plus conviviale et plus proche de celle du magasin, avec un conseiller-vendeur virtuel qui dialogue avec le client ;
- l'amélioration de la pertinence des produits présentés et de leur description, en cherchant davantage d'adéquation avec les attentes du client.

Classiquement, la recherche d'un produit dans de tels sites se fait via un moteur de recherche, éventuellement accompagné d'une série de filtres. Cette approche suppose non seulement que le client connaisse précisément le produit recherché mais également l'application logicielle en elle-même. Ici, nous souhaitons éviter ces hypothèses et choisissons une approche alternative, en utilisant un agent conversationnel pour assister le client dans sa prise de décision d'achat. La mission de l'agent consiste alors à cibler au mieux les attentes de l'utilisateur pour proposer des produits adéquats. Autrement dit, nous cherchons à concevoir un agent qui joue un rôle similaire à celui d'un conseiller-vendeur dans un magasin. Pour ce faire, l'agent doit notamment disposer de la connaissance du catalogue de produits et d'une expertise métier pour lier des produits ou services à des attentes du client.

### 8.2.2 Analyse des mécanismes de vente

Afin d'exhiber les spécificités du cadre applicatif décrit précédemment, nous avons analysé les mécanismes de vente avec l'appui de spécialistes du domaine. Des éléments de cette analyse sont disponibles dans la section A.1 de l'annexe A en page 157. Un résumé en est proposé dans le tableau 8.1.

Données	Rôle	Type de règles
Profil utilisateur	adéquation produit	domaine
Profil acheteur	adéquation produit argumentation adaptation mécanisme	vente
Produit (endogènes)	adéquation utilisateur	domaine
Produit (exogènes)	argumentation client	vente
Enseigne	mécanisme général de vente de l'enseigne	vente
Contexte	argumentation opportune	vente

TABLEAU 8.1 : Rôle des données dans le mécanisme de vente

Un processus de vente peut être vu comme la mise en adéquation du profil du client (utilisateur et acheteur) et des données produits (endogènes et exogènes) dans un contexte de vente. Si le catalogue de produits, la politique de l'enseigne et le contexte sont globalement communes à un ensemble de processus de vente successifs, le profil client n'est quant à lui pas le même d'un processus à un autre. Les premiers font donc l'objet d'une identification en amont afin d'être utilisées par le VVU au cours de différents dialogues de vente. Au contraire, on considère que pour chaque dialogue, le profil du client fait l'objet d'une identification et que cette tâche est affectée au VVU.

Après l'étude des différents facteurs susceptibles d'entrer en jeu dans un mécanisme de vente du VVU, l'étape suivante est l'établissement des spécifications logicielles de l'application.

## 8.3 Représentation des connaissances et raisonnement

Les spécifications du VVU sont illustrées ici avec le cas d'utilisation correspondant à la vente d'appareils photos numériques (APN) et le dialogue proposé dans la figure 8.1. Dans cette section et la suivante, de nombreux choix *ad-hoc* de métriques et de critères ont été faits. Ces choix ont une portée illustrative et ne constituent pas le cœur de la contribution, d'autres sont tout à fait envisageables.

Dans notre problématique de commerce électronique, il s'agit comme dans tout problème de décision de représenter les alternatives d'une part – ici les produits – et le profil du décideur d'autre part – ici un client.

### 8.3.1 Produit

Le VVU doit être capable de représenter une **catégorie** de produits du catalogue d'une enseigne, dans le cas présent, des APN. Une catégorie de produits se définit comme un en-

1. **VVU** : Quel est votre budget ?
2. **Client**: Petit budget.
3. **VVU** : C'est pour un utilisateur débutant, avancé ou expert ?
4. **Client**: Débutant.
5. **VVU** : Quel sorte de modèle cherchez-vous ?
6. **Client**: Le plus efficace.
7. **VVU** : Pour prendre quel type de photos ?
8. **Client**: Des photos de famille.
9. **VVU** : Je suggère le canon digital ixus 1100 hs. La meilleure technologie pour ce prix.
10. **Client**: Non merci.
11. **VVU** : Je suggère le nikon coolpix s6200. Le compact au meilleur prix.
12. **Client**: D'accord.

FIGURE 8.1 : Conversation entre le vendeur virtuel ubiquitaire et un client internaute

semble de caractéristiques (e.g. le **type** d'appareil, le **zoom**) comme représenté dans le tableau 8.2. Chaque **caractéristique** est associée à un ensemble de **valeurs** (pour le **type**, **compact/hybride/bridge/reflex**; pour le **zoom**, **basique/medium/puissant/innovant**). Il est important de noter que les domaines de valeurs sont discrétisés. Par exemple, les valeurs possibles pour la caractéristique **nombrePixels** sont : **moinsDe8M**, **de8a13M**, **plusDe13M**.

Caractéristiques	Valeurs
<b>type</b>	<b>compact</b>
	<b>hybride</b>
	<b>bridge</b>
	<b>reflex</b>
<b>nombrePixels</b>	<b>moinsDe8M</b>
	<b>de8a13M</b>
	<b>plusDe13M</b>
...	...
	...

TABLEAU 8.2 : Représentation des caractéristiques et des valeurs correspondantes pour l'univers APN

Un *produit* est un objet qui associe à chaque caractéristique une valeur unique parmi celles du domaine de définition, comme représenté dans le tableau 8.3 et plus largement dans le tableau A.1 de l'annexe A en page 160.

### 8.3.2 Client

Un **critère** est un élément de comparaison pour évaluer les produits. Dans notre approche, un critère correspond à un couple caractéristique-valeur. Par exemple, le fait de souhaiter qu'un appareil soit hybride se traduit par une valuation positive du critère **type=hybride**.

Les désirs du client sont représentés par le VVU via un **vecteur** qui définit l'importance des critères. La mesure de l'importance d'un critère est appelé **poids**. Cette mesure sur une échelle de Likert représente l'adhésion ou la désapprobation d'un client vis-à-vis de la valeur de la



Critère			
Caractéristiques	Valeurs	DMC-LX5	Nikon coolpix s6200
type	compact	1	1
	hybride	0	0
	bridge	0	0
	reflex	0	0
nombrePixels	moinsDe8M	0	0
	de8a13M	1	0
	plusDe13M	0	1
...	...	...	...
	...	...	...

TABLEAU 8.3 : Représentation des produits

caractéristique. Ce poids est un entier dans l'intervalle  $[-5; 5]$  (cf figure 8.2). Un poids nul signifie que le client est indifférent vis à vis de ce critère. Un poids strictement positif signifie que ce critère est attrayant alors qu'un poids strictement négatif signifie que ce critère est rebutant.

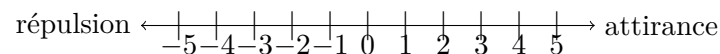


FIGURE 8.2 : Échelle de Likert pour le poids des critères

Afin de cerner les critères du client au fur et à mesure de l'interaction, le VVU utilise des **vecteurs de référence**. Ces vecteurs sont des stéréotypes définis *a priori* par l'expert marketing. Par exemple, un utilisateur « débutant » privilégiera un compact de moins de 8M pixels. Un acheteur « sécurité », privilégiera un compact ou un bridge plutôt qu'un hybride ou un reflex et aura une légère préférence pour une bonne résolution. Un vecteur de référence se représente, comme dans la table 8.4, par un vecteur de poids. Le caractère spécial \* indique que le critère n'est pas pertinent pour le vecteur de référence, c'est-à-dire que la réponse du client n'apporte pas d'information au niveau de ce critère. Cette sémantique est à distinguer d'un poids nul qui lui indique une neutralité ou indifférence du client sur le critère.

Caractéristiques	Valeurs	Sécurité	Débutant
type	compact	3	5
	hybride	1	*
	bridge	3	2
	reflex	1	-5
nombrePixels	moinsDe8M	0	3
	de8a13M	1	2
	plusDe13M	3	2
...	...	...	...
	...	...	...

TABLEAU 8.4 : Représentation des vecteurs de référence

Dans le cas spécifique présenté ici, la représentation du client comprend trois dimensions : la dimension « acheteur », la dimension « niveau d'utilisation » et la dimension « usage ». Pour

la dimension « acheteur », les vecteurs de référence choisis ici sont : **securite**, **orgueilleux**, **nouveaute**, **confort**, **argent** et **sympathie**. Cette typologie n'est pas spécifique au domaine des APN et peut être appliquée à d'autres types de produits ou services. Concernant le niveau d'utilisation, les vecteurs de référence possibles sont : **debutant**, **amateur** et **expert**. On retrouve ce type de catégorisation typiquement dans les domaines technologiques et on peut envisager un découpage plus ou moins fin, ici on a choisi de l'étendre sur 3 niveaux. Enfin, l'usage est quant à lui spécifique au domaine de la photographie et est décrit par les vecteurs de référence suivants : **familial**, **globeTrotter**, **vacancesEtvoyage**, **clubber**, **interieurEtportrait**, **fanDeSportetNature**, **reporter** et **polyvalent**. L'ensemble des vecteurs de référence ont été renseignés par des spécialistes et sont en annexe dans le tableau A.2, page 161.

Le **vecteur courant** est la représentation du profil du client par le VVU. Ce vecteur est dynamique, il évolue au fur et à mesure de l'avancement du processus. Ainsi, lors de l'initialisation, le VVU part d'un profil vide : tous les critères ont pour valeur le caractère spécial « ? » qui indique que le poids du critère pour le vecteur courant est encore inconnu. Le vecteur courant est complété et raffiné par les réponses du client, comme illustré dans le tableau 8.5.

Caractéristiques	Valeurs	Vecteur courant
<b>type</b>	<b>compact</b>	2
	<b>hybride</b>	?
	<b>bridge</b>	1,5
	<b>reflex</b>	-2
<b>nombrePixels</b>	<b>moinsDe8M</b>	0.5
	<b>de8a13M</b>	1.5
	<b>plusDe13M</b>	2
...	...	...
	...	...

TABLEAU 8.5 : Représentation d'un vecteur courant, la valeur associée à « hybride » est « inconnu »

Dans la section suivante, nous montrons comment le vecteur courant est calculé à partir des vecteurs de référence collectés au fil de la conversation.

### 8.3.3 Calcul du vecteur courant

À chaque réponse du client correspond un vecteur de référence qui est mémorisé. Le VVU calcule le vecteur courant à partir de cet historique. Le vecteur courant est une agrégation des vecteurs de référence qui correspondent à chacune des réponses du client (cf. algorithme 8.1).

---

#### Algorithme 8.1 : *calculVecteurCourant(vr)*

---

**Données** : vr : vecteurs de référence historisés

**Résultat** : vc : vecteur courant

- 1 **pour chaque**  $c \in \text{criteres}$  **faire**
  - 2   |  vc.c.poids := *agregationPoids*(vr,c);
  - 3 **fin**
-

**Algorithme 8.2** : *agregationPoids(vr, c)*


---

**Données** : vr : vecteurs de référence historisés, c : critere

```

1 sommePoids := 0;
2 nbPoidsExprimes := 0;
3 pour chaque v ∈ vr faire
4   | poids := v.getPoids(c);
5   | si poids ≠ * alors
6     | sommePoids += poids;
7     | nbPoidsExprimes++;
8   | si nbPoidsExprimes ≠ 0 alors retourner (sommePoids/nbPoidsExprimes);
9   | sinon retourner (?);
10 fin

```

---

Comme représenté dans l'algorithme 8.2, pour chaque critère, le poids dans le vecteur courant est la moyenne des poids exprimés dans les vecteurs de référence. Cet opérateur d'agrégation correspond à une attitude modérée. Une attitude sceptique consiste à prendre le minimum des poids et une attitude crédule à prendre le maximum. Si une réponse du client correspond à un vecteur pour lequel le critère est non-pertinent, alors ce vecteur n'est pas pris en compte pour ce critère.

Comme illustré dans le tableau 8.6, dans le cas particulier où aucun des vecteurs de l'historique n'est pertinent pour un critère, le poids de ce critère dans le vecteur courant est comme à l'initialisation inconnu (représenté par le caractère spécial « ? »).

Caractéristiques	Valeurs	Réponse 1	Réponse 2	Vecteur courant
type	compact	3	5	4
	hybride	1	*	1
	bridge	3	2	2.5
	reflex	1	-5	-2
nombrePixels	moinsDe8M	0	0	0
	de8a13M	1	3	2
	plusDe13M	2	3	2.5
capteur	classique	*	*	?
	innovant	*	*	?

TABLEAU 8.6 : Exemple de calcul du vecteur courant

Après avoir présenté comment les connaissances étaient représentés pour le VVU et le profil du client calculé à partir de celles-ci, nous nous intéressons à présent au calcul de ses coups de dialogue.

## 8.4 Gestions des buts, stratégie et tactique

Le VVU et le client interagissent à travers une conversation, c'est-à-dire une séquence cohérente de messages. Le VVU est proactif, il a l'initiative, il parle le premier (cf. section 7.3). À chaque

tour de parole, le VVU produit un énoncé (une question ou une proposition) en fonction de l'historique.

Cet échange de messages a pour but d'une part d'établir un profil du client (cf. section 8.4.1) et d'autre part d'identifier le(s) produit(s) à suggérer (cf. section 8.4.2).

### 8.4.1 Identification du profil du client

Le processus de modélisation du client vise à collecter de l'information dans le but d'établir le vecteur courant. Pour ce faire, le VVU pose des questions.

À chaque réponse possible à la question correspond un vecteur de référence. Ainsi les vecteurs collectés sont historisés. On peut noter que les questions et leurs réponses sont définies *a priori* par l'expert marketing. Par exemple, la question présente dans la figure A.1 en annexe (page 159) porte sur la classe d'acheteur. Pour chacune des questions, le client peut avouer son ignorance en répondant «je ne sais pas ». Dans notre cas d'utilisation, on compte parmi les questions disponibles :

- la question sur la classe de l'acheteur (cf. figure A.1 en annexe). En fonction de la réponse, le VVU considère que le client est soit **securite**, soit **orgueil**, etc. ;
- la question sur le niveau d'utilisation. En fonction de la réponse, le VVU considère que le client est soit **debutant**, soit **amateur**, soit **expert** ;
- la question sur l'usage. En fonction de la réponse, le VVU considère le client comme **familial**, **globeTrotter**, etc. ;
- les questions sur les caractéristiques. Par exemple, si le VVU demande le type d'appareil souhaité, le client peut répondre explicitement **compact**, **hybride**, **bridge**, ou **reflex**.

La stratégie de demande d'information est un algorithme qui sélectionne la prochaine question parmi la liste des questions mises à disposition par l'expert marketing.

Une stratégie de demande d'information du VVU consiste à prendre en compte les informations précédemment recueillies afin de poser la question susceptible de faire avancer au mieux le processus de modélisation du client. On propose ici deux stratégies : une première basée sur le gain informationnel au niveau du vecteur de profil et une seconde basée sur la circonscription des alternatives pertinentes.

### Calcul de question basé sur le gain informationnel

Chaque question apporte potentiellement de l'information. Le VVU choisit la question dont le gain informationnel potentiel est le plus important (cf. algorithme 8.3). Ce gain dépend du gain de la réponse future du client. Le gain informationnel de la question peut être estimé *a priori* en fonction des vecteurs de références associés aux réponses potentielles.

Comme représenté dans l'algorithme 8.4, nous considérons que le gain informationnel d'une question est le gain moyen de chacune des réponses possibles, i.e. des vecteurs de référence associés. Cet opérateur d'agrégation correspond à une attitude modérée. Une attitude optimiste correspond à prendre le maximum des gains et une attitude pessimiste à prendre le minimum.

Un vecteur de référence apporte un gain informationnel par rapport au vecteur courant si il

---

**Algorithme 8.3** : Calcul de la prochaine question

---

**Données** :  $vc$  : vecteur courant,  $qs$  : questions**Résultat** : question à poser

```

1 max := 0;
2 question := null;
3 pour chaque  $q \in questions$  faire
4   | lvr := q.getReponses();
5   | gain := agregationGains(vc,lvr);
6   | si  $gain > max$  alors
7     |   | max := gain;
8     |   | question := q;
9 fin
10 retourner (question);
```

---



---

**Algorithme 8.4** :  $agregationGains(vc, lvr)$ 

---

**Données** :  $vc$  : vecteur courant,  $lvr$  : vecteurs de référence**Résultat** : l'

```

1 sommeGain := 0;
2 nbVecteurs := 0;
3 pour chaque  $v \in lvr$  faire
4   | sommeGain += calculGain(v,vc);
5 fin
6 retourner (sommeGain/nbVecteurs);
```

---

permet de déterminer le poids d'un critère encore inconnu.

Nous considérons ici que le gain informationnel d'un vecteur de référence correspond au nombre de critères dans le vecteur courant qui sont évalués grâce à ce vecteur de référence (cf. algorithme 8.5). En d'autres termes, on incrémente le gain pour chacun des critères sur lesquels on n'avait jusqu'alors aucune information. Cette métrique pourrait être raffinée en prenant en considération les valeurs des poids dans le vecteur de référence. Intuitivement, un vecteur de référence qui permet d'avoir un avis tranché sur un critère est plus informatif qu'un vecteur de référence qui correspond à une indifférence sur ce critère.

Considérons la question caractéristique illustrée dans le tableau 8.7 et qui porte sur le capteur. Le gain informationnel de chacune des réponses est de 2 et donc le gain informationnel de la question est 2.

### Calcul de question basé sur la circonscription des alternatives pertinentes

Une seconde stratégie de demande d'information du VVU consiste à prendre en compte la base produit afin de poser la question susceptible de faire avancer au mieux le processus de sélection de produit. Chaque question permet de scinder la base produit. Le VVU choisit la question qui permet de sélectionner au mieux les produits et donc de minimiser le nombre de produits sélectionnables.

**Algorithme 8.5** : *calculGain(vr, vc)***Données** : vr :vecteur de référence, vc : vecteur courant**Résultat** : le gain informationnel apporté par le vecteur de référence *vr* sur le vecteur courant

```

1 gain := 0;
2 pour chaque c ∈ criteres faire
3   |   poidsRef := vr.getPoids(c);
4   |   poidsCourant := vc.getPoids(c);
5   |   si poidsRef ≠ * et poidsCourant = ? alors
6   |   |   gain+=1;
7 fin
8 retourner (gain);

```

Caractéristiques	Valeurs	Vecteur courant	Réponse 1	Réponse 2
type	compact	2	*	*
	hybride	1	*	*
	bridge	1.5	*	*
	reflex	-2	*	*
nombrePixels	moinsDe8M	0	*	*
	de8a13M	2	*	*
	plusDe13M	2.5	*	*
capteur	cmos	?	5	0
	ccd	?	0	5

TABLEAU 8.7 : Exemple de calcul de gain

L'objectif est de minimiser le nombre de produits potentiellement sélectionnables à l'issue d'une question (cf. algorithme 8.6). Ce nombre de produits dépend du nombre de produits potentiellement sélectionnables associé à la réponse future du client. Il peut être calculé *a priori* en fonction des vecteurs de références correspondants aux réponses potentielles.

Comme représenté dans l'algorithme 8.7, nous considérons que le nombre de produits potentiellement sélectionnables d'une question est le nombre de produits sélectionnables maximum parmi l'ensemble des réponses possibles, i.e. des vecteurs de référence associés. Cet opérateur d'agrégation correspond à une attitude prudente. Une attitude risquée consiste à considérer la réponse qui donne le nombre minimum de produits sélectionnables.

Nous considérons ici que le nombre de produits sélectionnables pour un vecteur courant correspond au nombre de produits dont le taux de satisfaction est supérieur à un seuil fixé *a priori* – e.g. 80 % – (cf. algorithme 8.8). Une alternative consiste à comptabiliser le nombre de critères satisfaits par le produit et qui sont importants pour le client.

Comme illustré dans l'algorithme 8.9, le calcul du taux de satisfaction dépend de la définition ad-hoc pour la satisfaction d'un produit vis-à-vis d'un vecteur courant (cf. algorithme 8.10).

---

**Algorithme 8.6** : Calcul de la prochaine question

---

**Données** : lvh : liste des vecteurs historisés, qs : questions**Résultat** : la question à poser

```

1 min := nbProduitsDansLaBase;
2 question := null;
3 pour chaque  $q \in \text{questions}$  faire
4   | nbProduits := nbProduitsSelectionnablesMax(lvh,q);
5   | si  $\text{nbProduits} < \text{min}$  alors
6     |   | min := nbProduits;
7     |   | question := q;
8 fin
9 retourner (question);
```

---



---

**Algorithme 8.7** :  $\text{nbProduitsSelectionnablesMax}(\text{lvh}, q)$ 

---

**Données** : lvh : liste des vecteurs historisés, q : question**Résultat** : nombre de produits sélectionnables au maximum

```

1 max := 0;
2 lvr := q.getReponses();
3 pour chaque  $vr \in \text{lvr}$  faire
4   | lvhTmp := concatenation(lvh,vr) ;
5   | vcTmp := calculVecteurCourant(lvhTmp);
6   | nbProdsTmp := nbProduitsSelectionnables(vcTmp, baseProd, 80) ;
7   | si  $\text{nbProdsTmp} > \text{max}$  alors
8     |   | max := nbProdsTmp;
9 fin
10 retourner (max)
```

---

### 8.4.2 Identification du produit

Un dialogue de délibération vise à atteindre un accord sur un produit. Pour ce faire, le VVU fait des propositions qui peuvent être acceptées ou refusées par le client.

#### Calcul de propositions

La stratégie de délibération est un algorithme qui sélectionne les produits à proposer en fonction du vecteur courant.

La stratégie de délibération du VVU repose sur la notion de satisfaction. La satisfaction pour un produit d'après le vecteur courant définit l'adéquation entre le produit et le vecteur courant. Une implémentation alternative serait par exemple de ne proposer que les produits dont le nombre de critères satisfaisants est supérieur à un seuil.

Comme représenté dans l'algorithme 8.10, la satisfaction pour un produit d'après un vecteur courant est la somme des poids du vecteur courant associés à chacun des critères satisfaits par le produit (e.g. plus de 3 sur l'échelle de Likert). On peut noter que le calcul de satisfaction est

---

**Algorithme 8.8** : *nbProduitsSelectionnables*(*vc*, *baseProd*, *seuil*)

---

**Données** : *vc* : vecteur courant, *baseProd* : base de produits, *seuil* : entier**Résultat** : nombre de produits dont le taux de satisfaction pour le vecteur courant dépasse le seuil

```

1 nbProduitsSelectionnables := 0;
2 pour chaque p ∈ baseProd faire
3   | si tauxSatisfaction(p,vc) > seuil alors
4   |   | nbProduitsSelectionnables++;
5 fin
6 retourner (nbProduitsSelectionnables);

```

---



---

**Algorithme 8.9** : *tauxSatisfaction*(*p*, *vc*)

---

**Données** : *p* : produit, *vc* : vecteur courant**Résultat** : taux de satisfaction du produit pour le vecteur courant

```

1 satisfactionMax := 0;
2 pour chaque cara ∈ caractéristiques faire
3   | poidsCaraMax := 0;
4   | pour chaque c ∈ criteres faire
5   |   | poids := vc.getPoids(c) ;
6   |   | si poids ≠ ? et poids > poidsCaraMax alors
7   |   |   | poidsCaraMax := poids;
8   |   | fin
9   |   | satisfactionMax += poidsCaraMax;
10 fin
11 tauxSatisfaction := satisfaction(p,vc) / satisfactionMax ;
12 retourner (tauxSatisfaction);

```

---

indifférent aux critères pour lesquels le vecteur courant n'est pas renseigné.

Finalement, le VVU propose les 5 propositions les plus satisfaisantes. Une alternative consiste à en proposer un autre nombre ou à ne proposer que les produits dont le taux de satisfaction est supérieur à un seuil.

Considérons les deux produits et le vecteur courant illustrés dans le tableau 8.8. La satisfaction pour le DMC-LX5 est de 4.5 et celle pour le Nikon est de 5. C'est donc le Nikon qui est privilégié.

### Argumentation

Chacune des propositions est argumentée. À cette intention, on distingue trois types d'arguments :

- un argument « récapitulatif » résume une réponse du client pour simuler le principe d'écoute active. Par exemple, l'argument associé au vecteur de référence *securite* est : « vous recherchez un appareil d'une grande qualité » ;
- un argument « critère » met en avant la valeur d'un attribut du produit. Par exemple,



**Algorithme 8.10** : Calcul de la satisfaction d'un produit pour un vecteur**Données** : p : produit, vc : vecteur courant**Résultat** : satisfaction

```

1 satisfaction := 0;
2 pour chaque c ∈ critères faire
3   | poids := pc.getPoids(c) ;
4   | si p.estSatisfait(c) et poids ≠ ? alors
5   |   | satisfaction += poids;
6 fin

```

Caractéristiques	Valeurs	Vecteur courant	DMC-LX5	Nikon coolpix s6200
type	compact	2.5	1	1
	hybride	1	0	0
	bridge	1.5	0	0
	reflex	-2	0	0
nombrePixels	moinsDe8M	0	0	0
	de8a13M	2	1	0
	plusDe13M	2.5	0	1
capteur	cmos	?	1	1
	ccd	?	0	0

TABLEAU 8.8 : Exemple de calcul de satisfaction des produits pour un vecteur courant

l'argument associé au critère `type=compact` est : « Petit et léger, il est le compagnon idéal de tous vos déplacements » ;

- un argument « référence/produit » spécifique à un couple vecteur de référence-produits. Par exemple, l'argument en faveur du produit DMC-LX5 pour le vecteur `securite` est : « le compact expert le plus fiable ».

Nous avons décidé de limiter notre argumentaire aux 5 meilleurs arguments « référence/produit ». La sélection des arguments « référence/produit » dépend du vecteur courant.

Nous souhaitons mettre en avant des arguments qui n'ont pas été évoqués explicitement lors du dialogue mais qui sont révélés par les interactions. À cette intention, on cherche à identifier le vecteur de référence le plus similaire au vecteur courant, on choisit de l'appeler « vecteur révélé ». Pour décider quel est ce vecteur, on recourt à l'algorithme 8.11 afin de calculer la dissimilarité entre un vecteur courant et un vecteur de référence.

Considérons les deux vecteurs de référence `securite` et `nouveaute` et le vecteur courant illustrés dans le tableau 8.9. La dissimilarité entre le vecteur courant et le vecteur `securite` est de 0,88 et celle entre le vecteur courant et le vecteur `nouveaute` est de 2,97. Le vecteur courant a une similarité plus forte avec le vecteur de référence `securite` qu'avec le vecteur de référence `nouveaute`.

---

**Algorithme 8.11** : Calcul de dissimilarité entre un vecteur courant et un vecteur de référence

---

**Données** : vr : vecteur de référence, vc : vecteur courant

**Résultat** : satisfaction

```

1 dissimilarite := 0;
2 nbCaracs := 0;
3 pour chaque carac ∈ caractéristiques faire
4   | nbValeurs := 0;
5   | valeurs := produit.getValeurs(carac);
6   | tmp := 0;
7   | pour chaque c ∈ valeurs faire
8     | poidsRef := vr.getPoids(c);
9     | poidsCourant := vc.getPoids(c);
10    | si poidsRef ≠ * alors
11      | nbValeurs++;
12      | si poidsCourant ≠ ? alors
13        | tmp += |poidsCourant - poidsRef| ;
14      | sinon
15        | tmp += (TAILLE_ECHELLE-1)/2 ;
16      | fin
17    | fin
18  | dissimilarite += tmp/nbValeurs;
19  | si nbCriteres ≠ 0 alors
20    | nbCaracs++;
21  | fin
22 fin
23 dissimilarite := dissimilarite/nbCaracs;
24 retourner dissimilarite

```

---

Caractéristiques	Valeurs	Vecteur courant	Sécurité	Nouveauté
type	compact	2.5	3	3
	hybride	1	1	5
	bridge	1.5	3	3
	reflex	-2	1	3
nombrePixels	moinsDe8M	0	0	-1
	de8a13M	2	1	0
	plusDe13M	2.5	3	3
capteur	cmos	?	*	*
	ccd	?	*	3

TABLEAU 8.9 : Exemple de calcul de dissimilarité

### 8.4.3 Comportement d'agent

Le comportement du VVU détermine quel énoncé produire à chaque tour. Ce comportement dépend du but principal de l'agent et de l'état associé. Ces états sont : l'état initial, l'état de demande d'information, l'état de délibération et l'état final (cf. figure 8.3). Dans les deux états centraux, le VVU utilise la stratégie adéquate dans le but soit, dans le premier cas, d'identifier le profil du client, soit, dans le second cas, que le client ajoute un produit à son panier.

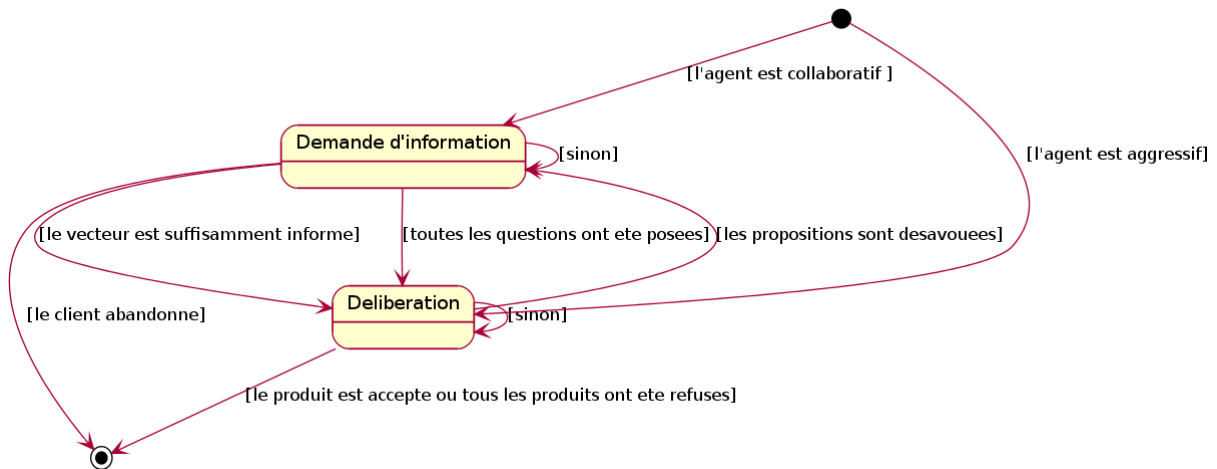


FIGURE 8.3 : Diagramme d'état-transition représentant le comportement interne du VVU

Le comportement actuel du VVU consiste à commencer par poser 5 questions (il est collaboratif) à l'issue desquelles, il effectue au plus 5 propositions.

## 8.5 Implémentation

Notre implémentation est composée : côté serveur, d'une plate-forme multi-agents sur laquelle interagissent le VVU et l'agent client. Ce dernier est contrôlé par l'humain via une interface de dialogue côté client. L'interface de dialogue prend la forme d'un formulaire web classique (cf. figure 8.4) avec un conteneur pour les questions et un autre pour les propositions. L'utilisateur

peut choisir entre différentes réponses prédéfinies pour chaque question posée par l'agent logiciel. Une sélection de produits sont ensuite recommandées au client et présentés sous la forme d'une fiche produit argumentée en fonction de l'historique du dialogue. D'un point de vue technique, l'interface utilisateur est codée avec les technologies AJAX. Nous avons défini un langage spécifique basé XML décrivant les questions et leurs réponses (cf. figure A.1, page 159).



FIGURE 8.4 : Interface web du prototype du VVU.

Le VVU est déployé côté serveur avec une plate-forme de prototype agent écrite en J2E qui permet l'interaction entre un agent client et le VVU. Pour chaque cas spécifique, les données à fournir par l'enseigne commerciale sont :

- la base de données produit contenant la description des alternatives potentiellement proposées ;
- les profils stéréotypes, c'est-à-dire la connaissance spécifique au domaine au niveau sémantique ;
- le comportement de l'agent, c'est-à-dire la stratégie de vente de l'enseigne ;
- les questions-réponses en langage naturel (cf. figure A.1) ;
- la liste des arguments.

Pour prouver la généralité de notre approche, nous avons implémenté le VVU sur les domaines de la téléphonie mobile et de la literie en plus de celui des APN.

## 8.6 Conclusion

### 8.6.1 Synthèse

Dans ce chapitre, nous avons détaillé les différents algorithmes mise en œuvre par le VVU pour le calcul : du vecteur courant, de la prochaine question, des propositions et des arguments. Ces algorithmes déterminent le comportement. Le choix de faire avancer le processus de modélisation du client ou le processus de modélisation du produit est configurable en fonction de la politique de vente de l'enseigne.

Dans un premier temps, nous avons analysé les mécanismes de vente à travers les données qu'ils manipulent. Quatre types de données ont ainsi été distingués : les données du client, du produit, de l'enseigne et du contexte. Le rôle de ces données dans le mécanisme de vente a été établi.

Dans un deuxième temps, nous avons proposé une modélisation des mécanismes de vente. Elle passe par différents processus d'identification : de l'acheteur, de l'utilisateur et du produit. Ces processus consistent à acquérir les données utiles au mécanisme en utilisant des connaissances métiers. Le pilotage des processus est ajustable au fil du dialogue et exprime le comportement du VVU.

Enfin, avec l'aide de spécialistes, nous avons implémenté un prototype du VVU sur les domaines de la photographie, de la literie et de la téléphonie mobile.

### 8.6.2 Perspectives

De nombreuses pistes d'extensions possibles ont été formulées dans la perspective de futurs travaux. Parmi ces pistes, certaines sont communes à celles présentées dans la section 7.5.3 (page 88) concernant notre modèle d'agent dialogique proactif. Plus spécifiquement au cadre applicatif présenté dans le courant chapitre, une perspective directe est l'implémentation de stratégies de vente différentes.

Dans notre approche, l'efficacité et la pertinence du VVU dépendent évidemment en premier lieu des données saisies par les experts marketing et les experts du domaine. Pour favoriser le bon déroulement de cette étape, un ensemble de ressources peuvent être envisagées. Elles peuvent notamment prendre la forme d'exemples et de conseils sur les bonnes pratiques à adopter dans les choix des questions, des vecteurs de référence et de la représentation des profils. De manière plus aboutie, on peut envisager la conception d'outils pour assister les experts dans la saisie et l'exploration des données du VVU. En particulier, de tels outils pourraient faciliter la tâche de discrétisation de la base de données des produits.

Enfin, nous avons choisi de ne pas traiter dans ce chapitre de multiples problématiques qui peuvent faire l'objet d'autres travaux. Parmi celles-ci, on peut évoquer l'utilisation de données sur le client disponibles en amont du dialogue, par exemple via des historiques de dialogue précédents ou les réseaux sociaux.



Troisième partie

Négociation multi-agents équitable





---

# Système délibératif multilatéral pour un accord équitable

---

## Sommaire

<b>9.1</b>	<b>Introduction</b>	<b>111</b>
<b>9.2</b>	<b>Hypothèses sur les préférences</b>	<b>112</b>
<b>9.3</b>	<b>Définition d'une alternative équitable</b>	<b>112</b>
9.3.1	Modélisation de l'effort	113
9.3.2	Critère pour l'agrégation des efforts	114
<b>9.4</b>	<b>Résolution centralisée de la recherche des alternatives équitables</b>	<b>117</b>
<b>9.5</b>	<b>Négociation multilatérale pour atteindre un accord équitable</b>	<b>118</b>
9.5.1	Protocole	119
9.5.2	Stratégies	121
<b>9.6</b>	<b>Conclusion</b>	<b>126</b>
9.6.1	Synthèse	126
9.6.2	Discussion	126
9.6.3	Perspectives	127

---

## 9.1 Introduction

En première partie de ce document, et en particulier dans le chapitre 5, nous avons mis en lumière les différentes hypothèses formulées dans la littérature traitant de la prise de décision collective ainsi que les résultats obtenus. L'objet du présent chapitre est d'exposer notre contribution sur cette problématique : un modèle de prise de décision collective pour atteindre un accord équitable.

Cette contribution est présentée en plusieurs étapes. D'abord, dans la section 9.2, nous mettons en exergue les partis pris dans notre approche. Ensuite, nous proposons dans la section 9.3 les critères retenus pour caractériser et définir les alternatives équitables. Puis, dans la section 9.4, nous présentons un algorithme permettant d'identifier l'ensemble des alternatives équitables. Dans la section 9.5, nous exposons un protocole de négociation multilatérale visant à favoriser des accords équitables. Enfin, nous discutons des résultats et perspectives de notre contribution dans la section 9.6.

## 9.2 Hypothèses sur les préférences

Dans la section 4.4 (page 53), nous avons discuté des différentes hypothèses pour modéliser les préférences des agents dans le cadre de la prise de décision collective. Nos objectifs ici étant de caractériser et d'atteindre les alternatives équitables, la prise en compte d'un degré d'intensité des préférences, subjectif aux agents, ne nous semble pas pertinente. En effet, comme défendu notamment par Arrow, Sen et Rawls dans leurs travaux, les comparaisons interpersonnelles d'utilité peuvent être à l'origine de biais dans l'agrégation. C'est pourquoi, dans l'optique d'éviter cet écueil, nous considérons des agents munis de **relations de préférence**.

Dans de nombreux cas, le décideur n'a pas de préférence stricte entre deux alternatives, soit parce qu'il les considère aussi satisfaisantes l'une que l'autre, soit parce qu'il est trop incertain pour procéder à un classement. Aussi, nous considérons que les préférences des agents peuvent être **larges** et **partielles** (cf. définitions section 1.3, page 14). En évitant ainsi des hypothèses restrictives, nous permettons une forte expressivité pour les préférences des agents et nous proposons un modèle générique.

Au long de ce chapitre, nous utiliserons en guise d'illustration la situation présentée dans l'exemple 9.1 où 3 agents ont à choisir ensemble parmi six alternatives sur lesquelles ils ont des préférences larges partielles.

### Exemple 9.1.

Nous considérons ici  $\succsim_1$ ,  $\succsim_2$  et  $\succsim_3$  les relations de préférences de 3 agents sur un ensemble d'alternatives  $\mathcal{X} = \{a, b, c, d, e, f\}$ . Les graphes de préférences associés sont représentés dans la figure 9.1, où un nœud représente une alternative et un arc de  $x$  vers  $y$  indique que  $x$  est au moins aussi bonne que  $y$  ( $x \succsim y$ ). On peut remarquer que les alternatives non-dominées pour chacun des agents sont  $c_{ND}(\mathcal{X}, \succsim_1) = \{a, b, c\}$ ,  $c_{ND}(\mathcal{X}, \succsim_2) = \{c\}$  et  $c_{ND}(\mathcal{X}, \succsim_3) = \{a, b\}$ . L'ensemble des alternatives potentiellement Pareto-optimales (cf. définition 4.7, page 49) est  $c_{ND}(\mathcal{X}, \succ^{PP}) = \{a, b, c, d, e\}$ .

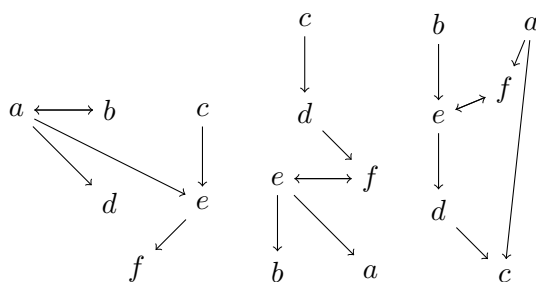


FIGURE 9.1 : Relations de préférence  $\succsim_1$  (à gauche),  $\succsim_2$  (au milieu) et  $\succsim_3$  (à droite)

## 9.3 Définition d'une alternative équitable

Dans le chapitre 4, nous avons étudié différentes manières d'agréger les préférences des agents. Ici, il s'agit, à partir de préférences ordinales, qui peuvent être larges et partielles, d'identifier

des alternatives équitables.

### 9.3.1 Modélisation de l'effort

Afin de définir la notion d'alternative équitable, nous choisissons de nous intéresser aux efforts concédés par les agents en fonction de l'alternative choisie à l'issue du processus.

#### Exemple 9.2 (Mesure de l'effort).

*Considérons par exemple un groupe de personnes qui souhaitent aller au restaurant ensemble, 3 possibilités s'offrent à elles : un estaminet, un restaurant italien ou une crêperie. Parmi les participants, Alice classe l'estaminet au premier rang, l'italien au deuxième et la crêperie en troisième et dernière position. Si la prise de décision collective mène à choisir la crêperie, on considère que l'effort fourni par Alice équivaut à une perte de deux rangs par rapport à son alternative préférée.*

Ici, nous décrivons les efforts en prenant comme unité les rangs perdus. De telles métriques ont déjà été envisagées dans la littérature. Dans [Goodman & Markowitz 1952], les auteurs considèrent qu'un agent a un nombre limité de niveaux de discriminations. Partant de cette hypothèse, ils proposent de réintroduire du numérique en mesurant la distance entre deux alternatives par le nombre de niveaux les séparant. Ils remettent ainsi en question l'hypothèse d'indépendance aux alternatives non pertinentes considérant qu'elles apportent de l'information. Par exemple, dans la figure 9.1, on peut considérer que pour l'agent 1, l'alternative  $a$  est 2 niveaux au-dessus de l'alternative  $f$  étant donné que  $a \succ e \succ f$ .

Cependant, dans [Goodman & Markowitz 1952], les relations de préférences considérées sont totales. Ici, nous souhaitons définir une métrique qui soit aussi valable sur des relations partielles. Cette problématique a notamment été étudiée dans le cadre des CP-nets dans [Li et al. 2010] où les auteurs se basent sur la notion de rang. Ici, nous généralisons cette notion à n'importe quelle relation de préférence. Ainsi, le rang d'une alternative s'obtient en retirant successivement les couches d'alternatives non-dominées.

#### Définition 9.1 (Rang).

*Soient  $\mathcal{X}$  l'ensemble des alternatives et  $\succsim_i$  la relation de préférence large de l'agent  $i$  sur  $\mathcal{X}$ . Le **rang** d'une alternative  $x$  pour  $\succsim_i$  dans  $\mathcal{X}$  est défini tel que :*

$$r(x, \mathcal{X}, \succsim_i) = \begin{cases} 1 & \text{si } x \in c_{ND}(\mathcal{X}, \succsim_i) \\ 1 + r(x, \mathcal{X} \setminus c_{ND}(\mathcal{X}, \succsim_i), \succsim_i) & \text{sinon} \end{cases}$$

La notion de rang d'une alternative est voisine de celle de profondeur en théorie des graphes. Ici, par convention, nous conférons un rang de valeur 1 à une alternative non-dominée. Le calcul du rang d'une alternative s'effectue récursivement en descendant en profondeur dans le graphe.

#### Exemple 9.3 (Calcul du rang d'une alternative).

*Considérons le graphe de préférence de l'agent 1, à gauche de la figure 9.1. Le rang de l'alternative  $f$  pour l'agent 1 est  $r(f, \{a, b, c, d, e, f\}, \succsim_1) = 1 + r(f, \{d, e, f\}, \succsim_1) = 1 + 1 +$*

$$\| \quad r(f, \{f\}, \succsim_1) = 3.$$

Le tableau 9.1 indique le rang des alternatives pour chacun des agents dont les préférences sont représentées dans la figure 9.1.

$x$	$r(x, \mathcal{X}, \succsim_1)$	$r(x, \mathcal{X}, \succsim_2)$	$r(x, \mathcal{X}, \succsim_3)$
$a$	1	4	1
$b$	1	4	1
$c$	1	1	4
$d$	2	2	3
$e$	2	3	2
$f$	3	3	2

TABLEAU 9.1 : Rangs des alternatives pour chaque agent.

### 9.3.2 Critère pour l'agrégation des efforts

Avec cette métrique de rang, nous définissons désormais une mesure du rang social des alternatives en utilisant les agrégateurs cardinaux proposés par la théorie du choix social (cf. section 4.2, page 45). Ici, l'objectif poursuivi est en particulier d'identifier des alternatives à la fois Pareto-optimales et équitables.

Considérons pour commencer le rang moyen, il correspond à la moyenne des rangs donnés par la société d'agents à une alternative.

#### Définition 9.2 (Rang moyen).

Soient  $\mathcal{X}$  un ensemble d'alternatives,  $\Omega$  un ensemble d'agents et  $x \in \mathcal{X}$  une alternative. Le **rang moyen**, noté  $\bar{r}(x, \mathcal{X}, \Omega)$  est défini tel que :

$$\bar{r}(x, \mathcal{X}, \Omega) = \frac{\sum_{i \in \Omega} r(x, \mathcal{X}, \succsim_i)}{\text{card}(\Omega)}$$

La fonction de rang moyen opère une moyenne des rangs des agents qui composent le groupe, elle est à mettre en parallèle avec celle de bien-être utilitaire.

#### Exemple 9.4 (Calcul du rang moyen d'une alternative).

Considérons les rangs individuels du tableau 9.1. Le rang moyen de l'alternative  $f$  est égal à la moyenne de ses rangs pour les trois agents,  $\bar{r}(f, \{a, b, c, d, e, f\}, \{1, 2, 3\}) = \frac{\sum_{i \in \{1, 2, 3\}} r(f, \{a, b, c, d, e, f\}, \succsim_i)}{\text{card}(\{1, 2, 3\})} = \frac{3+3+2}{3} = \frac{8}{3}$

Les rangs moyens des alternatives sont indiqués dans le tableau 9.2, d'après les rangs individuels du tableau 9.1 calculés à partir des préférences de la figure 9.1. On remarque que  $a$ ,  $b$  et  $c$  minimisent le rang moyen.

$x$	$\bar{r}(x, \mathcal{X}, \Omega)$
$a$	2
$b$	2
$c$	2
$d$	$\frac{7}{3}$
$e$	$\frac{7}{3}$
$f$	$\frac{8}{3}$

TABLEAU 9.2 : Rangs moyens des alternatives

L'agrégateur utilitaire de la moyenne traduit le critère d'efficacité mais pas celui de l'équité. Dans le tableau 9.2, les alternatives qui minimisent le rang moyen sont  $a$ ,  $b$  et  $c$ , alors qu'elles constituent dans le même temps les alternatives de plus mauvais rang pour un des trois agents.

Afin d'introduire de l'équité, on peut choisir de baser le rang social d'une alternative sur le rang le plus haut – autrement dit le pire – parmi les agents.

**Définition 9.3 (Rang maximal).**

Soient  $\mathcal{X}$  un ensemble d'alternatives,  $\Omega$  un ensemble d'agents et  $x \in \mathcal{X}$  une alternative.

$$r_{max}(x, \mathcal{X}, \Omega) = \max_{i \in \Omega} r(x, \mathcal{X}, \succsim_i)$$

Pour prendre en compte l'agent le plus mal loti, on emploie généralement le critère du *maximin*, qui consiste à maximiser la satisfaction minimale. Ici, comme nous raisonnons en terme d'efforts, il s'agit de minimiser le rang maximal (*minimax*).

**Exemple 9.5 (Calcul du rang maximal d'une alternative).**

Considérons les rangs individuels du tableau 9.1. Le rang maximal de l'alternative  $f$  est égal au maximum de ses rangs pour les trois agents,  $r_{max}(f, \{a, b, c, d, e, f\}, \{1, 2, 3\}) = \max(r(f, \{a, b, c, d, e, f\}, \succsim_1), r(f, \{a, b, c, d, e, f\}, \succsim_2), r(f, \{a, b, c, d, e, f\}, \succsim_3)) = \max(3, 3, 2) = 3$ .

Les rangs maximaux des alternatives sont indiqués dans le tableau 9.3, d'après les rangs individuels du tableau 9.1 calculés à partir des préférences de la figure 9.1.

$x$	$r_{max}(x, \mathcal{X}, \Omega)$
$a$	4
$b$	4
$c$	4
$d$	3
$e$	3
$f$	3

TABLEAU 9.3 : Rangs maximaux des alternatives

En visant à minimiser le nombre d'efforts de l'agent en ayant réalisé le plus, le critère égalitaire du *minimax* est plus équitable. Cependant, il a l'inconvénient de ne pas garantir la Pareto-

optimalité. En effet, dans notre exemple,  $d$ ,  $e$  et  $f$  minimisent le rang maximal. Cependant, parmi ces alternatives égalitaires,  $f$  n'est pas Pareto-optimale car Pareto-dominée par  $e$ .

Afin d'identifier des alternatives à la fois équitables et Pareto-optimales, nous raffinons le critère du *minimax* en généralisant le principe en cas d'*ex-æquo* au deuxième agent le plus mal loti et ainsi de suite (sur le même principe que le *leximin*, définition 4.10, page 51). C'est donc la règle du *leximax* que nous adoptons ici.

**Définition 9.4 (Préférences sociales *leximax*).**

Soient  $\mathcal{X}$  un ensemble d'alternatives,  $\Omega$  un ensemble de  $n$  agents et  $x, y \in \mathcal{X}$  deux alternatives. Notons  $\vec{r}_\Omega(x) = (r_1(x), \dots, r_n(x))$  le vecteur ordonné de manière décroissante des rangs de l'alternative  $x$  pour les agents dans  $\Omega$ . On dit que :

- l'alternative  $x$  est strictement *leximax*-préférée à  $y$  (on note  $x \succ^{lex,r} y$ ) si et seulement si :

$$\exists k \leq n, \forall i < k, r_i(x) = r_i(y) \text{ et } r_k(x) < r_k(y)$$

- l'alternative  $x$  est *leximax*-équivalente à  $y$  (on note  $x \sim^{lex,r} y$ ) si et seulement si :

$$\forall i \leq n, r_i(x) = r_i(y)$$

**Propriété 9.1.** La relation de préférences  $(\succ^{lex,r}, \mathcal{X})$  est un pré-ordre total.

Comme la relation  $(\succ^{lex,r}, \mathcal{X})$  est un pré-ordre total, elle permet de procéder à une comparaison entre n'importe quelle paire d'alternatives et donc d'établir un classement. Ainsi, on peut identifier les alternatives aux efforts répartis de manière *leximax-optimale* que nous qualifions d'*équitables*.

**Définition 9.5 (Alternative équitable).**

Soit  $\mathcal{X}$  un ensemble d'alternatives et  $\Omega$  un ensemble d'agents. Soit  $(\succ^{lex,r}, \mathcal{X})$  la relation de préférences sociales *leximax* pour l'ensemble d'agents  $\Omega$ . L'ensemble des **alternatives équitables** sur  $\mathcal{X}$  pour  $\Omega$  est l'ensemble  $c_{ND}(\mathcal{X}, \succ^{lex,r})$ .

L'ensemble des alternatives équitables est une restriction de l'ensemble des alternatives potentiellement Pareto-optimales (cf. définition 4.7, page 49).

**Propriété 9.2** (Pareto-optimalité potentielle des alternatives équitables).

$$c_{ND}(\mathcal{X}, \succ^{lex,r}) \subseteq c_{ND}(\mathcal{X}, \succ^{PP})$$

**Démonstration 9.1** (Propriété 9.2). Soit  $x$  une alternative équitable sur  $\mathcal{X}$  pour  $\Omega$ . Démontrons par l'absurde que  $x$  est potentiellement Pareto-optimale sur  $\mathcal{X}$  pour  $\Omega$ . Supposons que  $x$  n'est pas potentiellement Pareto-optimale, c'est-à-dire qu'il existe une alternative  $y \in \mathcal{X}$  qui Pareto-domine nécessairement  $x$ . De la définition 4.4 (page 48), on déduit (1) que  $\forall i \in \Omega, y \succ_i x$  et (2) qu'il existe un agent  $a \in \Omega$  tel que  $y \succ_a x$ . De (1) et de la définition 9.1, on déduit que  $\forall i \in \Omega, r(x, \mathcal{X}, \succ_i) \geq r(y, \mathcal{X}, \succ_i)$ . De (2) et de la définition 9.1, on déduit qu'il existe un agent  $a \in \Omega$  tel que  $r(x, \mathcal{X}, \succ_a) > r(y, \mathcal{X}, \succ_a)$ . De la définition 9.4, on déduit que  $y \succ^{lex,r} x$  et donc que  $x$  n'est pas une alternative équitable, ce qui vient contredire notre hypothèse de départ.

Toutes les alternatives équitables sont potentiellement Pareto-optimales mais l'inverse n'est pas nécessairement vrai. En considérant les préférences dans la figure 9.1, le tableau 9.4 présente, pour chaque alternative  $x$ , son vecteur de rangs  $\vec{r}_\Omega(x)$ . Les alternatives  $d$  et  $e$  sont des alternatives équitables et, *a fortiori*, potentiellement Pareto-optimales. L'alternative  $c$  est elle aussi Pareto-optimale mais n'est toutefois pas équitable.

$x$	$r(x, \mathcal{X}, \succsim_1)$	$r(x, \mathcal{X}, \succsim_2)$	$r(x, \mathcal{X}, \succsim_3)$	$\vec{r}_\Omega(x)$
$a$	1	4	1	(4,1,1)
$b$	1	4	1	(4,1,1)
$c$	1	1	4	(4,1,1)
$d$	2	2	3	(3,2,2)
$e$	2	3	2	(3,2,2)
$f$	3	3	2	(3,3,2)

TABLEAU 9.4 : Vecteur de rangs des alternatives.

Il existe toujours au moins une alternative équitable.

**Propriété 9.3** (Garantie d'existence d'une alternative équitable). *Soient  $\mathcal{X}$  un ensemble d'alternatives,  $\Omega$  un ensemble d'agents et  $(\succsim^{lex,r}, \mathcal{X})$  la relation de préférences sociales lexicmax pour l'ensemble d'agents  $\Omega$ , il existe toujours  $x \in \mathcal{X}$  tel que  $x \in c_{ND}(\mathcal{X}, \succsim^{lex,r})$ .*

**Démonstration 9.2** (Propriété 9.3). *La relation lexicmax est un ordre partiel sur un ensemble non vide et donc cet ensemble ordonné contient au moins un élément maximal.*

En résumé, il est raisonnable de vouloir atteindre une alternative équitable car son existence est garantie et ce critère est Pareto-inclusif. La notion de *leximax*-optimalité implique des hypothèses supplémentaires à la Pareto-optimalité. En particulier, la fonction de rang évalue le niveau relatif d'une alternative au vu des préférences d'un agent. La propriété d'indépendance aux alternatives non-pertinentes n'est alors plus respectée. En lui appliquant le critère du *leximax*, nous traduisons ainsi l'idée intuitive de répartition équilibrée des efforts.

## 9.4 Résolution centralisée de la recherche des alternatives équitables

Maintenant que nous avons défini quelles étaient les alternatives équitables, nous souhaitons pouvoir les atteindre. Dans cette section, nous proposons une résolution du problème de recherche des alternatives équitables de manière centralisée. Autrement dit, nous faisons l'hypothèse que l'ensemble des préférences des agents ont été collectées et nous proposons l'algorithme 9.1, qui calcule l'ensemble des alternatives équitables.

Le principe de l'algorithme 9.1 est de descendre de rang en rang dans les préférences des agents. À chaque nouveau rang  $k$ , pour chaque agent  $i$ , on sélectionne chaque alternative  $x$  qu'il positionne au rang  $k$  et on met à jour son vecteur de rangs  $\vec{r}_\Omega(x)$  (ligne 7). Les alternatives qui, les premières, ont leur vecteur de rang complet, sont égalitaires (ligne 10) : leur rang maximal est minimisé (cf. définition 9.3). Un sous-ensemble de celles-ci sont équitables, on les identifie grâce à un appel à l'algorithme 9.2.

---

**Algorithme 9.1** : Algorithme centralisé de recherche des alternatives équitables

---

**Données** :  $\mathcal{X}$  : l'ensemble des alternatives,  $(\succsim_i)_{i \in \Omega}$  : les relations de préférence

```

1 Candidats :=  $\emptyset$  ;
2  $k := 1$  ;
3 tant que Candidats =  $\emptyset$  faire
4   pour chaque  $i \in \Omega$  faire
5      $R_k := \{x \in \mathcal{X} \mid r(x, \mathcal{X}, \succsim_i) = k\}$  ;
6     pour chaque  $x \in R_k$  faire
7        $\vec{r}_\Omega(x).ajouterEnTete(k)$  ;
8     fin
9   fin
10  Candidats :=  $\{x \in \mathcal{X} \mid taille(\vec{r}_\Omega(x)) = card(\Omega)\}$ ;
11   $k := k + 1$  ;
12 fin
13 retourner  $calculLeximax(Candidats, \{\vec{r}_\Omega(x) \mid x \in Candidats\})$ 

```

---



---

**Algorithme 9.2** : *calculLeximax* : Calcul des alternatives dont le vecteur associé est *leximax-optimal*

---

**Données** : *Candidats* : un ensemble des alternatives,  $(\vec{u}_\Omega(x))_{x \in Candidats}$  : les vecteurs associés à ces alternatives

```

1 pour  $k := 1$  à  $card(\Omega)$  faire
2    $Candidats := \arg \min_{x \in Candidats} \{\vec{u}_\Omega(x).[k]\}$  ;
3   si  $card(Candidats) = 1$  alors retourner Candidats;
4 fin
5 retourner Candidats

```

---

L'algorithme 9.2 consiste à calculer les alternatives dont le vecteur associé est *leximax-optimal* (voir définition 9.5). Son appel dans l'algorithme 9.1 permet d'extraire les alternatives équitables des égalitaires. Pour ce faire, à chaque pas  $k$ , on conserve uniquement les alternatives dont le composant  $k$  du vecteur est minimal (ligne 2). Si à un pas de boucle, une seule alternative minimise ce composant, elle constitue l'unique *leximax-optimal* et l'algorithme la retourne (ligne 3). Dans le cas contraire, l'algorithme renvoie un ensemble comportant plusieurs alternatives dont les vecteurs associés sont égaux.

## 9.5 Négociation multilatérale pour atteindre un accord équitable

Une résolution centralisée suppose de disposer de l'intégralité des préférences des agents. On souhaite à présent établir un processus décentralisé dans lequel chaque agent participe et dont l'objectif est d'atteindre un accord équitable. À cette fin, nous proposons dans un premier temps un protocole de négociation multilatérale. Puis, dans un second temps, nous mettons en œuvre des stratégies conformes à ce protocole, selon les buts poursuivis par les participants.



### 9.5.1 Protocole

Pour élaborer un protocole permettant la résolution décentralisée de la recherche des alternatives équitables, nous cherchons à distribuer l'algorithme 9.1, en particulier la ligne 5. Ici, on souhaite qu'au tour  $k \geq 1$ , chaque agent  $i$  énonce l'ensemble de ses alternatives de rang  $k$ . Pour ce faire, nous envisageons le protocole sous la forme d'un échange d'offres régit par un ensemble de règles :

**Définition 9.6 (Protocole de négociation multilatérale).**

Soit  $\Omega$  un ensemble d'agents souhaitant procéder à un choix parmi un ensemble d'alternatives  $\mathcal{X}$ . Notre protocole de négociation est défini par les règles suivantes :

1. à un tour donné, l'ensemble des agents dans  $\Omega$  proposent simultanément un ensemble d'alternatives ;
2. la répétition par un agent d'une alternative déjà proposée par ce même agent dans un tour précédent est interdite ;
3. si un agent a épuisé toutes les alternatives dans  $\mathcal{X}$  lors des tours précédents, et uniquement dans ce cas, il peut proposer un ensemble vide ;
4. la négociation est close dès qu'une alternative au moins a été proposée par tous les agents ;
5. une fois le jeu clos, l'ensemble des accords est constitué des alternatives dont le vecteur des tours est *leximax-optimal*.

Par souci de clarté, nous choisissons de recourir à l'emploi d'un arbitre pour implémenter notre protocole de négociation. Cependant, celui-ci vise avant tout à simplifier la présentation du modèle et n'en est pas partie intégrante. En effet, on peut concevoir le protocole de négociation sous une forme totalement distribuée, qui ne requiert pas d'arbitre. Ce rôle est alors endossé par l'ensemble des agents.

Le protocole est décrit par un automate à états fini (voir aussi section 2.3.2, page 33) dans la figure 9.2<sup>1</sup>. À l'état initial, chaque participant envoie un acte *propose* à l'arbitre. À l'état suivant (numéroté 2), l'arbitre envoie un acte *accept* contenant les accords, deux cas de figures se présentent alors : soit aucun accord a été trouvé, le contenu est donc l'ensemble vide et on revient à l'état initial pour débiter un nouveau tour ; soit des accords ont été trouvés, l'acte *accept* transmis aux participants contient alors au moins une alternative et on atteint donc l'état final (numéroté 3) qui symbolise la clôture du protocole. Dans la version totalement décentralisée du protocole, chacun des agents revêt à la fois le rôle de participant et celui d'arbitre. Par conséquent, les participants sont alors informés à chaque tour des propositions effectuées par les autres agents.

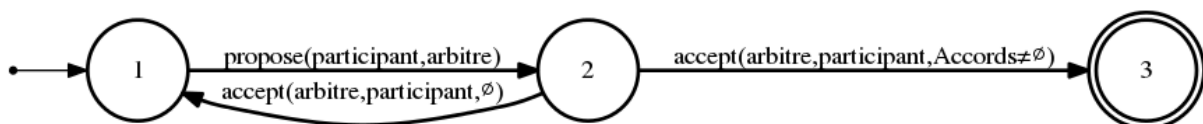


FIGURE 9.2 : Notre protocole de négociation sous la forme d'un automate à états fini.

1. Par simplification, nous avons utilisé des machines à état pour représenter le protocole quitte à omettre l'aspect temporel qui aurait pu être représenté par un réseau de Pétri.

Nous introduisons quelques notations.

**Notation 9.1** (Propositions d'un agent à un tour). *Soit  $i$  un agent participant à une négociation, on note  $p_i(t)$  l'ensemble des alternatives qu'il propose au tour  $t \geq 1$ .*

**Notation 9.2** (Ensemble des propositions précédemment formulées par un agent). *Soit  $i$  un agent participant à une négociation, l'ensemble des alternatives qu'il propose avant le tour  $t \geq 1$  est noté  $\mathcal{P}_i(t) = \bigcup_{k < t} p_i(k)$ .*

**Notation 9.3** (Vecteur de tours d'une alternative). *Soit  $\Omega$  un ensemble d'agents participant à une négociation, on note  $\vec{t}_\Omega(x)$  le vecteur, rangé dans l'ordre décroissant, des tours où  $x$  est proposé par les agents dans  $\Omega$ .*

L'algorithme 9.3 décrit la suite d'opérations à réaliser par l'arbitre à chaque tour. Après avoir pris connaissance des propositions effectuées par les agents, l'arbitre commence par mettre à jour pour chaque alternative  $x$  le vecteur de tours associé  $\vec{t}_\Omega(x)$  (lignes 2 à 8). Dès qu'une alternative au moins a été proposée par l'ensemble des participants (ligne 9), il existe un accord. L'arbitre recherche alors, avec l'algorithme 9.2, la ou les alternatives qui apparaissent le plus tôt – au sens *leximax* – dans la négociation, elles constituent **l'ensemble des accords** qu'il envoie alors aux participants (ligne 11 et 12). Si aucun accord n'a été trouvé à ce stade de la négociation, il retourne un ensemble vide aux participants (ligne 14) qui doivent alors soumettre de nouvelles propositions.

---

**Algorithme 9.3** : Mécanisme d'arbitrage

---

**Données** :  $\mathcal{X}$  : l'ensemble des alternatives,  $(\vec{t}_\Omega(x))_{x \in \mathcal{X}}$  : les vecteurs de tours  
**Entrées** :  $k$  : le tour courant,  $(p_i(k))_{i \in \Omega}$  : les dernières propositions

- 1 *Communes* :=  $\emptyset$ ;
- 2 **pour chaque**  $x \in \mathcal{X}$  **faire**
- 3     **pour chaque**  $i \in \Omega$  **faire**
- 4         **si**  $x \in p_i(k)$  **alors**
- 5              $\vec{t}_\Omega(x).ajouterEnTete(k)$  ;
- 6         **fin**
- 7     **fin**
- 8 **fin**
- 9 *Communes* :=  $\{x \in \mathcal{X} | taille(\vec{t}_\Omega(x)) = card(\Omega)\}$ ;
- 10 **si** *Communes*  $\neq \emptyset$  **alors**
- 11     *Accords* := *calculLeximax*(*Communes*,  $\{\vec{t}_\Omega(x) | x \in Communes\}$ );
- 12     *accept*( $\Omega$ , *Accords*);
- 13 **fin**
- 14 **sinon**
- 15     *accept*( $\Omega$ ,  $\emptyset$ );
- 16 **fin**

---

Une fois le protocole clos, un ensemble d'accords non-vide est systématiquement produit.

**Propriété 9.4** (Garantie d'arrêt et de succès du protocole de négociation). *Notre protocole de négociation s'arrête et garantit d'aboutir à un accord en au plus  $card(\mathcal{X})$  tours.*

**Démonstration 9.3** (Propriété 9.4 - Garantie d'arrêt et de succès du protocole de négociation). *D'après la règle 4 du protocole, la négociation s'arrête dès qu'une alternative a été proposée par l'ensemble des participants. Or, nous avons fait l'hypothèse que l'ensemble des alternatives en négociation est fini. De plus, la règle 2 du protocole interdit la répétition d'une alternative par un même participant. Ainsi, il existe nécessairement un tour de jeu où une alternative a été proposée par l'intégralité des agents et l'arrêt du processus est donc garanti. De plus, d'après la règle 5 du protocole et étant donné que le pré-ordre leximax est total (cf. propriété 9.1), il existe nécessairement au minimum un accord. Étant donné qu'un agent énonce au minimum une alternative par tour, la négociation s'achève après au plus  $\text{card}(\mathcal{X})$  tours.*

Le protocole produit toujours un ensemble non-vide d'accords. Cependant, un accord n'est pas nécessairement unique. Pour aboutir effectivement à une décision collective qui soit une alternative unique, on peut alors recourir à un tirage aléatoire parmi les alternatives restantes.

## 9.5.2 Stratégies

Un agent émet à chaque tour une proposition contenant une ou plusieurs alternatives. Le contenu des propositions qu'il émet est déterminé par sa stratégie qui s'interface avec le protocole via les conditions d'énonciation et d'interprétation des coups.

On peut envisager différentes stratégies conformes à notre protocole selon que les agents aient un but commun ou qu'ils soient en compétition. Nous présentons une stratégie pour chacun de ces cas. Nous appelons *Coopérative* la première et *Temporisatrice* la seconde.

Une exécution de notre protocole de négociation est donc paramétrée par le profil de stratégie de ses participants.

**Notation 9.4** (Profil de stratégies). *On note  $\mathcal{S} = (S_1, \dots, S_n)$  le profil de stratégies où l'agent  $i$  joue la stratégie  $S_i$ , avec pour tout  $i$ ,  $S_i$  égale soit à  $C$  (pour coopérative) soit à  $T$  (pour temporisatrice). Par souci de simplicité, on définit une notation pour chacun des profils de stratégies suivants :*

- $C_\Omega = (C_1, C_2, \dots, C_n)$ , le profil de stratégies où tous les agents appliquent la stratégie coopérative.
- $T_\Omega = (T_1, T_2, \dots, T_n)$ , le profil de stratégies où tous les agents appliquent la stratégie temporisatrice.
- $C_{\Omega-i} = (C_1, \dots, C_{i-1}, T_i, C_{i+1}, \dots, C_n)$ , le profil de stratégies où l'agent  $i$  applique la stratégie temporisatrice et tous les autres agents appliquent la stratégie coopérative.
- $T_{\Omega-i} = (T_1, \dots, T_{i-1}, C_i, T_{i+1}, \dots, T_n)$ , le profil de stratégies où l'agent  $i$  applique la stratégie coopérative et tous les autres agents appliquent la stratégie temporisatrice.

Nous introduisons à présent une notation permettant de désigner l'ensemble des accords produits par des agents adoptant un profil de stratégie donné.

**Notation 9.5** (Ensemble d'accords). *Soit  $\Omega$  un ensemble d'agents participant à une négociation sur un ensemble d'alternatives  $\mathcal{X}$  avec le profil de stratégie  $\mathcal{S}$ . L'ensemble d'accords issu de cette négociation est noté  $\sigma_\Omega(\mathcal{X}, \mathcal{S})$ .*

## Stratégie Coopérative

Dans le cas où les participants coopèrent pour trouver les alternatives équitables, la stratégie à suivre est simple. Il s'agit, pour l'agent  $i$ , d'envoyer au tour  $k$  les alternatives de rang  $k$ , ou, autrement dit, les non-dominées sur l'ensemble d'alternatives qu'il n'a pas déjà énoncées (cf. algorithme 9.4).

### Définition 9.7 (Stratégie Coopérative).

La stratégie coopérative, notée  $C$ , pour notre protocole de négociation multilatérale (cf. définition 9.6) est décrite par l'algorithme 9.4.

---

**Algorithme 9.4** : Stratégie Coopérative : envoi des alternatives non-dominées parmi les inédites

---

**Données** :  $\mathcal{X}$  : l'ensemble des alternatives

- 1  $Inedit := \mathcal{X} \setminus \mathcal{P}_i(t)$ ;
  - 2  $ND := c_{ND}(\succsim_i, Inedit)$ ;
  - 3  $propose(arbitre, ND)$ ;
- 

### Exemple 9.6 (Négociation entre trois agents coopératifs).

Considérons les préférences de la figure 9.1. Dans le tableau 9.5, nous représentons la trace d'une négociation où les trois agents adoptent une stratégie coopérative. Au tour  $t = 1$ , l'agent 1 commence par proposer les alternatives  $a, b$  et  $c$ , l'agent 2 propose l'alternative  $c$  et l'agent 3 propose les alternatives  $a$  et  $b$ . Comme aucune alternative n'a été proposée par l'ensemble des agents, un deuxième tour a lieu. Au troisième tour, les alternatives égalitaires  $d, e$  et  $f$  ont été proposées par l'ensemble des agents, la négociation prend donc fin (règle 5). L'ensemble des accords  $\sigma(\mathcal{S})$  est alors composé des alternatives équitables  $d$  et  $e$  dont les vecteurs de tours sont *leximax-optimaux*.

$t$	$C_1$	$C_2$	$C_3$
1	$\{a, b, c\}$	$\{c\}$	$\{a, b\}$
2	$\{d, e\}$	$\{d\}$	$\{e, f\}$
3	$\{f\}$	$\{e, f\}$	$\{d\}$
$\sigma_\Omega(\mathcal{X}, C_\Omega) = \{d, e\}$			

TABLEAU 9.5 : Trace de négociation où les trois agents appliquent la stratégie coopérative

Les accords atteints dans un jeu de négociation entre des agents coopératifs sont des alternatives équitables et réciproquement.

### Théorème 9.1 (Accords équitables quand tous les agents appliquent la stratégie Coopérative).

$\sigma(C_\Omega) = c_{ND}(\mathcal{X}, \succsim^{lex,r})$ .

Le jeu partage les efforts entre les agents et tous les accords équitables sont atteints.

**Lemme 9.1.** *Soit  $x \in \mathcal{X}$  une alternative et  $\theta$  le numéro du dernier tour. Si  $r(x, \mathcal{X}, \succ_i) \leq \theta$  alors  $x \in p_i(r(x, \mathcal{X}, \succ_i))$ .*

**Démonstration 9.4** (Lemme 9.1). *Preuve par récurrence.*

*Cas initial :* Au tour 1, un agent coopératif  $i$  joue l'ensemble de ses alternatives non-dominées (cf. algorithme 9.4), c'est-à-dire toutes les alternatives  $x \in \mathcal{X}$  avec  $r(x, \mathcal{X}, \succ_i) = 1$  (cf. définition 9.1).

*Hypothèse de récurrence :* Supposons qu'à l'issue du tour  $k$ , un agent coopératif ait déjà joué toutes les alternatives  $x \in \mathcal{X}$  telles que  $r(x, \mathcal{X}, \succ_i) \leq k$ .

Au tour  $k + 1 < \theta$ , un agent coopératif  $i$  joue toutes les alternatives non-dominées qui sont légales. Plus formellement, il joue  $p_i(k + 1) = c_{ND}(\mathcal{X} \setminus \mathcal{P}_i(k + 1), \succ_i)$ . Par l'hypothèse de récurrence,  $p_i(k + 1) = c_{ND}(\mathcal{X} \setminus \{x \in \mathcal{X} \mid r(x, \mathcal{X}, \succ_i) \leq k\}, \succ_i)$ . Par l'algorithme 9.4, nous concluons qu'un agent coopératif joue au tour  $k + 1$  ou auparavant toutes les alternatives  $x \in \mathcal{X}$  telles que  $r(x, \mathcal{X}, \succ_i) \leq k + 1$ .

**Démonstration 9.5** (Théorème 9.1). (a) *Démontrons par l'absurde que  $\sigma(\mathcal{X}, C_\Omega) \subseteq c_{ND}(\mathcal{X}, \succ_{lex,r})$ . Considérons un accord  $x \in \sigma(\mathcal{X}, C_\Omega)$ . Faisons l'hypothèse que  $x \notin c_{ND}(\mathcal{X}, \succ_{lex,r})$ . Alors, d'après la définition 4.10, il existe nécessairement  $y \in \mathcal{X}$  tel que  $y \succ_{lex,r} x$ . Or, par le lemme 9.1 et l'algorithme 9.3, on a  $x \notin \sigma(\mathcal{X}, C_\Omega)$  ce qui contredit notre hypothèse.*

(b) *Démontrons par l'absurde que  $c_{ND}(\mathcal{X}, \succ_{lex,r}) \subseteq \sigma(\mathcal{X}, C_\Omega)$ . Considérons  $x \in c_{ND}(\mathcal{X}, \succ_{lex,r})$ . Faisons l'hypothèse que  $x \notin \sigma(\mathcal{X}, C_\Omega)$ . Par la propriété 9.4, il existe nécessairement  $y \in \sigma(\mathcal{X}, C_\Omega)$ . Par le théorème 9.1 (a),  $y \in c_{ND}(\mathcal{X}, \succ_{lex,r})$ . D'après la définition 4.10  $\vec{r}_\Omega(x) = \vec{r}_\Omega(y)$ . Par le lemme 9.1,  $\vec{t}_\Omega(x) = \vec{r}_\Omega(y)$ . Par l'algorithme 9.3,  $x \in \sigma(\mathcal{X}, C_\Omega)$  ce qui contredit notre hypothèse.*

## Stratégie Temporatrice

Adoptons maintenant le point de vue d'un agent égoïste  $i$  qui ne se préoccupe pas de la qualité sociale de l'accord et a pour unique but que celui-ci porte sur une de ses alternatives préférées. Nous cherchons à concevoir une stratégie qui soit meilleure que la stratégie coopérative dans cette perspective.

L'algorithme 9.5 compose le premier filtre de la stratégie d'un agent égoïste. Comme la stratégie coopérative, cette stratégie doit être **légale** : les alternatives ne sont pas répétées dans ses propositions (cf. ligne 1). De plus, il ne serait pas rationnel de proposer une alternative alors qu'une meilleure n'a pas encore été dite, c'est pourquoi nous nous restreignons aux alternatives non-dominées parmi celles restantes (cf. ligne 2). En appliquant la stratégie coopérative, un agent peut énoncer plusieurs alternatives à un même tour. Or, s'il souhaite que ses alternatives préférées aient un vecteur de tours *leximax*-optimal, il a intérêt à n'en proposer qu'une seule par tour. Nous proposons donc de faire appel à une fonction nommée *choixUnique* visant à choisir une seule alternative (cf. ligne 3) puis la proposer (cf. ligne 4).

En se restreignant aux alternatives non-dominées parmi les restantes, un agent minimise à chaque étape le niveau de profondeur dans le graphe restreint à *Inedit*. Autrement dit, à chaque étape, l'agent ne s'autorise à jouer que des alternatives qui minimisent le rang parmi les restantes.

---

**Algorithme 9.5 :** Stratégie de proposition rationnelle d'un participant égoïste

---

**Données :**  $\mathcal{X}$  : l'ensemble des alternatives

- 1  $Inedit := \mathcal{X} \setminus \mathcal{P}_i(t)$ ; // les alternatives inédites
  - 2  $ND := c_{ND}(Inedit, \succsim_i)$ ; // parmi elles, celles qui sont non-dominées
  - 3  $E := choixUnique(ND)$ ; // l'agent en choisit une seule
  - 4  $propose(arbitre, E)$ ;
- 

L'algorithme 9.6 présente une implémentation assez simple de la fonction de *choixUnique*. Afin de choisir parmi les alternatives rationnelles, l'agent opte pour celle(s) dont le rang est minimal (cf. algorithme 9.6). Enfin, l'agent temporisateur renvoie une unique alternative, quitte à procéder à un tirage aléatoire dans le cas où plusieurs alternatives restantes ont un rang minimal.

---

**Algorithme 9.6 :** *choixUnique* : Minimiser le rang pour l'agent  $i$

---

**Données :**  $Rat$  : ensemble rationnel d'alternatives

**Résultat :** un ensemble d'alternatives

- 1  $Min := \arg \min_{x \in Rat} \{r(x, \mathcal{X}, \succsim_i)\}$  ;
  - 2 **retourner**  $tirageAlea(Min)$  ;
- 

**Définition 9.8 (Stratégie Temporisatrice).**

La stratégie *temporisatrice* (notée  $T$ ) résulte de la combinaison de :

1. la stratégie de proposition d'un participant (cf. algorithme 9.5) ;
2. le choix d'une unique alternative dont le rang est minimal (cf. algorithme 9.6).

Comme illustré dans l'exemple suivant, si au moins un des agents applique la stratégie temporisatrice, l'accord obtenu n'est plus forcément Pareto-optimal.

**Exemple 9.7 (Négociation avec deux agents coopératifs et un agent temporisateur).**

Considérons les préférences de la figure 9.1. Dans le tableau 9.6, nous représentons la trace d'une négociation où les deux premiers agents adoptent une stratégie coopérative tandis que le troisième adopte une stratégie temporisatrice. Les agents coopératifs jouent de la même manière que dans l'exemple 9.1. Le troisième agent quant à lui temporise et ne propose jamais plus d'une alternative par tour. Ainsi, au tour 1, il propose soit l'alternative  $a$  soit l'alternative  $b$ . Au tour suivant, il propose celle des deux qu'il n'a pas encore proposée. Au troisième et dernier tour, l'alternative qu'il propose – c'est-à-dire  $e$  ou  $f$  – ayant été aussi proposée par les deux autres participants, le jeu s'arrête. C'est alors cette dernière alternative proposée par l'agent 3 qui constitue l'unique accord issu de la négociation : soit  $e$ , soit  $f$ . Remarquons que l'alternative  $d$ , moins bonne que  $e$  et  $f$  pour l'agent 3, n'est pas dans l'ensemble d'accord alors qu'elle y était dans l'exemple 9.6. En temporisant, l'agent 3 a donc obtenu un ensemble d'accord plus favorable. Enfin, rappelons que l'alternative  $f$ , susceptible d'être un accord ici, n'est pas potentiellement Pareto-optimale.

Dans l'exemple 9.7, l'agent 3 a gagné à employer la stratégie temporisatrice plutôt que la coopérative. Ce résultat n'est cependant pas généralisable. En particulier, dans le cas où les préférences

$t$	$C_1$	$C_2$	$T_3$
1	$\{a, b, c\}$	$\{c\}$	$\{a\}$ ou $\{b\}$
2	$\{d, e\}$	$\{d\}$	$\{b\}$ ou $\{a\}$
3	$\{f\}$	$\{e, f\}$	$\{e\}$ ou $\{f\}$
$\sigma_\Omega(\mathcal{X}, C_{\Omega-3}) = \{e\}$ ou $\sigma_\Omega(\mathcal{X}, C_{\Omega-3}) = \{f\}$			

TABLEAU 9.6 : Trace de négociation où les deux premiers agents appliquent la stratégie coopérative et le troisième la stratégie temporisatrice

d'un agent sont strictes et totales, les stratégies coopératives et temporisatrices décrites ici sont strictement équivalentes. En effet, avoir des alternatives *ex-aequo* ou incomparables est une condition nécessaire pour pouvoir temporiser.

**Théorème 9.2 (Équivalence des stratégies coopérative et individuellement rationnelles avec des préférences strictes totales).**

*Soit  $i$  un agent muni d'une relation  $\succ_i$  de préférence stricte et totale et participant à notre protocole de négociation multilatérale. Les propositions émises en appliquant la stratégie coopérative décrite dans l'algorithme 9.4, sont les mêmes qu'en appliquant l'algorithme 9.5, quelque soit l'implémentation de la fonction *choixUnique*.*

**Démonstration 9.6** (Théorème 9.2). *Dans l'algorithme 9.5, à la ligne 3,  $\text{card}(ND) = 1$  car il ne peut y avoir qu'une seule alternative non-dominée avec des préférences strictes totales et donc l'ensemble d'alternatives proposé  $E$  (lignes 3 et 4) est égal à  $ND$  quelque soit l'implémentation de *choixUnique*. Or, c'est aussi le cas dans l'algorithme 9.4.*

Le corollaire du théorème 9.2 est qu'un groupe d'agents munis de préférences strictes totales et appliquant une stratégie basée sur l'algorithme 9.5 obtiennent l'ensemble des accords équitables.

Il est cependant possible d'établir des résultats généraux démontrant l'intérêt de la stratégie temporisatrice individuellement. Si nous fixons les stratégies des autres participants ainsi que la fonction de tirage aléatoire (nommée *tirageAlea* dans l'algorithme 9.6), nous pouvons comparer les accords obtenus selon la stratégie employée par un agent, toutes choses étant égales par ailleurs.

Lorsque tous les autres participants coopèrent, un agent ne peut jamais obtenir d'accord plus avantageux en étant coopératif plutôt que temporisateur.

**Théorème 9.3 (Non-dominance de la stratégie coopérative sur la stratégie temporisatrice).**

*Si la fonction *tirageAlea* d'un joueur temporisateur  $i$  est fixée arbitrairement alors :*

$$\forall x \in \sigma_\Omega(\mathcal{X}, C_\Omega) \setminus \sigma_\Omega(\mathcal{X}, C_{\Omega-i}), \nexists y \in \sigma_\Omega(\mathcal{X}, C_{\Omega-i}), (x \succ_i y)$$

**Démonstration 9.7** (Théorème 9.3). *Soit  $x \in \sigma_\Omega(\mathcal{X}, C_\Omega) \setminus \sigma_\Omega(\mathcal{X}, C_{\Omega-i})$ . Démontrons par l'absurde que  $\forall y \in \sigma_\Omega(\mathcal{X}, C_{\Omega-i}), \neg(x \succ_i y)$ . Considérons  $y \in \sigma_\Omega(\mathcal{X}, C_{\Omega-i})$  et supposons que  $x \succ_i y$ . On sait que : i) avec le profil de stratégies  $\sigma_\Omega(\mathcal{X}, C_\Omega)$ , tous les agents ont proposé  $x$  ( $\exists k > 0, x \in \mathcal{P}_i(k) \wedge x \in \bigcap_{j \neq i} \mathcal{P}_j(k)$ ) car c'est un accord; ii) avec le profil de stratégies  $C_{\Omega-i}$ ,*



tous les agents ont proposé  $y$  ( $\exists k' > 0, y \in \mathcal{P}_i(k') \wedge y \in \bigcap_{j \neq i} \mathcal{P}_j(k')$ ) car c'est un accord.

Chaque participant coopératif  $j$  fait les mêmes propositions dans les deux configurations ( $\forall k > 0, \mathcal{P}_j(k) = \mathcal{P}_j(k')$ ). Puisque  $x \notin \sigma_\Omega(\mathcal{X}, C_{\Omega-i})$ ,  $i$  propose  $y$  avant  $x$  ( $\exists k' > 0, y \in \mathcal{P}_i(k') \wedge x \notin \mathcal{P}_i(k')$ ). Cette dernière assertion contredit notre hypothèse ( $x \succ_i y$ ).

Si les agents sont coopératifs et adoptent le but commun d'atteindre les alternatives équitables, ils sélectionnent à chaque tour l'ensemble des alternatives non dominées et favorisent ainsi une issue rapide. A contrario, si un agent a pour seul but de privilégier ses alternatives préférées, il a intérêt à temporiser en choisissant une unique alternative. En plus de chercher à minimiser le rang des accords, il repousse ainsi l'obtention d'un accord. Cependant, il n'y a alors plus de garantie d'atteindre un Pareto-optimal.

## 9.6 Conclusion

### 9.6.1 Synthèse

Dans ce chapitre, nous avons défini une alternative équitable comme répartissant les efforts de manière *leximax*-optimale. Nous avons proposé un algorithme centralisé pour identifier l'ensemble de ces alternatives. Puis, nous avons décentralisé la recherche de ces alternatives sous la forme d'un protocole de négociation qui garantit d'arriver effectivement à un accord. Nous avons ensuite mis en œuvre deux stratégies s'y conformant. Premièrement, la stratégie coopérative qui permet à des agents poursuivant le but commun d'atteindre un accord équitable d'y parvenir. Deuxièmement, la stratégie temporisatrice, égo-centrée, au service des préférences de l'agent qui l'emploie, qui consiste à proposer une seule alternative par tour. Après avoir remarqué que les deux stratégies sont équivalentes pour un agent ayant des préférences strictes totales, nous avons établi qu'avec des relations larges et/ou partielles, lorsqu'au moins un agent temporise, il se peut que l'accord obtenu ne soit ni équitable, ni même Pareto-optimal.

Dans un cadre purement compétitif, un agent a intérêt à employer la stratégie temporisatrice quelque soit la stratégie choisie par son adversaire (coopérative ou temporisatrice). Cependant, la configuration de jeu où tous les agents coopèrent est la seule qui garantisse d'arriver à des accords qui soient équitables, c'est-à-dire qui évite que le rang de l'accord soit particulièrement élevé – et donc mauvais – pour de nombreux agents. Ce résultat est à rapprocher de la situation bien connue du dilemme des prisonniers en théorie des jeux où intérêt individuel et intérêt collectif s'opposent.

Dans le chapitre suivant, nous évaluons expérimentalement le mécanisme proposé ici et cherchons notamment à mettre en balance les conditions dans lesquelles temporiser s'avère une stratégie gagnante et l'influence d'agents temporisateurs sur le caractère équitable de l'ensemble d'accord produit.

### 9.6.2 Discussion

La métrique du rang a fait l'objet de débats dans la littérature et a notamment été discutée par Arrow et Sen (cf. Chap. 7 de [Sen 1977]). Le contre-argument qu'ils présentent consiste à considérer un individu qui porte des jugements binaires sur les alternatives (deux niveaux :



magnifique ou horrible) face à un second individu exprimant beaucoup plus de nuances (cinq niveaux : excellent, bien, moyen, mauvais, affreux). Peut-on alors considérer équivalentes la différence de degré entre “magnifique” et “horrible” pour le premier agent à celle entre “excellent” et “bien” pour le second ? Ici, nous choisissons de ne pas poser la question en ces termes. Nous considérons que ces différences de degré sont par nature subjectives et ne cherchons pas à les mesurer et les comparer. Cependant, le nombre de niveaux intermédiaires constitués par les alternatives peut servir de critère, arbitraire certes – en existe-t-il un qui ne le serait pas ? – mais pas partial pour autant.

Concernant le théorème d'impossibilité d'Arrow 5.2.4, notre processus de décision ne respecte pas la propriété de l'indépendance aux alternatives non pertinentes. En effet, ici, l'ajout d'une alternative est susceptible de modifier les préférences sociales *leximax* entre deux autres alternatives. Ainsi, le résultat d'une prise de décision collective effectuée à l'aide de notre protocole dépend évidemment de l'ensemble d'alternatives sur lequel porte le problème de décision.

Selon l'implémentation qu'on en fait, notre processus de prise de décision collective peut être vu sous soit l'angle des systèmes de votes, soit sous celui de la négociation. En effet, ici nous ne faisons pas d'hypothèse particulière sur la communication entre les agents, celle-ci peut soit être directe, soit passer par l'intermédiaire d'un arbitre.

### 9.6.3 Perspectives

La stratégie temporisatrice présentée ici est assez simple, une première perspective est de chercher à l'améliorer afin de la rendre plus efficace. Si on se situe dans un cadre totalement distribué, sans arbitre, chaque agent peut connaître à l'issue d'un tour les propositions énoncées par ses interlocuteurs. On peut alors envisager une stratégie plus adaptative, consistant par exemple à ne choisir que les alternatives les moins proposées dans la fonction *choixUnique* (algorithme 9.6).

Une perspective plus générale consiste à proposer d'autres protocoles de négociation voisins de celui-ci. En effet, la démarche suivie pour concevoir notre protocole peut être étendue à d'autres critères que le *leximax* sur les rangs. Ainsi, il est possible de modifier les règles 4 et 5 du protocole (cf. définition 9.6) pour chercher par exemple à atteindre une alternative maximisant le rang moyen ou le produit de Nash des rangs. Ce dernier agrégateur nous semble particulièrement pertinent comme approche intermédiaire pour avoir à la fois un rang moyen raisonnable et limiter le nombre d'agents pour lesquels le rang de l'accord est très bas.

On peut aussi établir une autre métrique que le rang pour l'agrégation des préférences et la définition mathématique d'un objectif. Inversement à notre notion de rang qui consiste en un coût fonction de la distance aux meilleures alternatives, dans [Aydoğan *et al.* 2013], les auteurs proposent de définir une utilité numérique par la distance aux pires alternatives. Une autre piste est de compter le nombre d'alternatives préférables à une alternative pour un agent. De manière générale, l'approche consiste à attribuer une valeur numérique aux alternatives *a posteriori*, à partir des préférences ordinales, afin de procéder ensuite à une agrégation.



---

# Expérimentation du protocole de négociation pour un accord équitable

---

## Sommaire

---

<b>10.1 Introduction</b> . . . . .	<b>129</b>
<b>10.2 Protocole expérimental</b> . . . . .	<b>129</b>
10.2.1 Génération des préférences . . . . .	130
10.2.2 Méthode de comparaison des ensembles d'accord . . . . .	131
<b>10.3 Résultats</b> . . . . .	<b>132</b>
10.3.1 Gain à temporiser . . . . .	132
10.3.2 Équité des accords . . . . .	134
10.3.3 Recours à l'arbitraire . . . . .	134
<b>10.4 Conclusion</b> . . . . .	<b>136</b>
10.4.1 Synthèse . . . . .	136
10.4.2 Perspectives . . . . .	136

---

## 10.1 Introduction

Dans le chapitre précédent, nous avons proposé un protocole de négociation pour des alternatives équitables ainsi que deux stratégies, l'une coopérative et l'autre temporisatrice. Ici, nous souhaitons évaluer empiriquement notre protocole. D'une part, nous étudions les conditions dans lesquelles un agent égoïste à effectivement intérêt à appliquer la stratégie temporisatrice plutôt que la coopérative. D'autre part, nous étudions l'influence sur la qualité du résultat de l'emploi de la stratégie temporisatrice par tout ou partie des agents. Pour ce faire, nous procédons à des expérimentations. La méthode employée est décrite dans la section 10.2. Dans la section 10.3, nous présentons les résultats produits et en en faisons l'analyse. Enfin, la section 10.4 conclut ce chapitre par une synthèse et des perspectives pour évaluer notre protocole sur d'autres critères.

## 10.2 Protocole expérimental

Nous souhaitons ici observer :

- les conditions dans lesquelles un agent gagne en pratique à temporiser plutôt qu'à être coopératif ;
- l'influence de la présence d'agents temporisateurs sur l'équité d'un accord obtenu à l'issue de notre protocole ;
- si, à l'issue du processus de négociation, il est généralement nécessaire de recourir à l'arbitraire pour atteindre une unique décision, c'est-à-dire de procéder à un tirage aléatoire dans un ensemble d'accords en contenant plusieurs.

Dans ce but, nous commençons par définir un protocole expérimental. Tout d'abord, nous proposons une méthode pour générer de manière pseudo-aléatoire des relations de préférences sur des ensembles d'alternatives puis nous proposons une relation de préférence sur les ensembles d'alternatives afin de comparer les ensembles d'accord.

### 10.2.1 Génération des préférences

Nous souhaitons pouvoir réaliser un grand nombre d'expériences avec des agents ayant des préférences différentes, y compris dans la topologie de leur graphe.

La génération d'une relation de préférences est paramétrée par :

- $card(\mathcal{X})$ , son nombre d'alternatives ;
- $txEqMin$ , sa proportion minimale de couples d'alternatives différentes équivalentes (avec  $0 \leq txEqMin \leq 1$ ) ;
- $txIncMax$ , sa proportion maximale de couples d'alternatives différentes incomparables (avec  $0 \leq txIncMax \leq 1$ ).

À partir de ces paramètres, nous générons un graphe de préférences (cf. définition 1.6, page 16). Cette génération consiste à enrichir un graphe initialement sans arc en ajoutant :

1. des relations d'équivalence, jusqu'à atteindre un taux de couples équivalents supérieur ou égal à  $txEqMin$  (cf. algorithme 10.1) ;
2. des relations de préférence, jusqu'à atteindre un taux de couples incomparables inférieur ou égal à  $txIncMax$  (cf. algorithme 10.2).

---

#### Algorithme 10.1 : Ajout de relations d'équivalences

---

**Données :**  $(\mathcal{X}, A)$  : graphe de préférence avec  $\mathcal{X}$  les nœuds et  $A$  les arcs,  $txEqMin$  : le taux d'équivalence minimum souhaité

```

1  $nbCouples := card(\mathcal{X}) \times (card(\mathcal{X}) - 1)$  ;
2  $txEq := card(\{(x, y) \in \mathcal{X}^2 | (x \sim y)\}) / nbCouples$  ;
3 tant que  $txEq < txEqMin$  faire
4    $Inc := \{(x, y) \in \mathcal{X}^2 | (x \parallel y)\}$  ;
5    $(a, b) := random(Inc)$  ;
6    $A := A \cup \{(a, b)\} \cup \{(b, a)\}$  ;
7    $txEq := card(\{(x, y) \in \mathcal{X}^2 | (x \sim y)\}) / nbCouples$  ;
8 fin
```

---

Nous insistons sur le fait que les taux paramétrant la génération des relations pseudo-aléatoires sont des seuils.

**Algorithme 10.2** : Ajout de relations de dominance

---

**Données** :  $(\mathcal{X}, A)$  : graphe de préférence,  $txIncMax$

- 1  $nbCouples := card(\mathcal{X}) \times (card(\mathcal{X}) - 1)$  ;
- 2  $txInc := card(\{(x, y) \in \mathcal{X}^2 | (x \parallel y)\}) / nbCouples$  ;
- 3 **tant que**  $txInc > txIncMax$  **faire**
- 4      $Inc := \{(x, y) \in \mathcal{X}^2 | (x \parallel y)\}$  ;
- 5      $(a, b) := random(Inc)$  ;
- 6      $A := A \cup \{(a, b)\}$  ;
- 7      $txInc := card(\{(x, y) \in \mathcal{X}^2 | (x \parallel y)\}) / nbCouples$  ;
- 8 **fin**

---

Dans chacune de nos expériences, nous considérons 100 agents dont les préférences ont été générées en suivant la démarche présentée en début de cette section. Ici, une expérience est un ensemble de 4 exécutions du processus de négociation où les profils de stratégie des agents sont  $C_\Omega$ ,  $C_{\Omega-i}$ ,  $T_\Omega$  et  $T_{\Omega-i}$  (voir notations 9.4, page 121).

Afin de comparer les ensembles d'accords, nous fixons la fonction de choix pseudo-aléatoire des agents appliquant la stratégie temporisatrice en utilisant l'ordre lexicographique. Les paramètres pour la génération des relations de préférence sont :

- $card(\mathcal{X}) = 25$  ;
- $txEqMin \in \{0, 0.1, 0.2\}$  et
- $txIncMax \in [0, 0.6]$ .

Pour chaque jeu de paramètres, nous procédons à 1000 générations des préférences des agents faisant chacune l'objet d'une expérience.

### 10.2.2 Méthode de comparaison des ensembles d'accord

Afin de comparer les accords issus de différentes configurations du processus de négociation, nous souhaitons pouvoir généraliser la relation de préférence au niveau d'ensembles d'alternatives en nous inspirant de [Barberà *et al.* 2004].

**Définition 10.1 (Relation de préférence sur un ensemble).**

Soit  $\mathcal{X}$  un ensemble fini non-vide d'alternatives et  $\succsim_i$  la relation de préférences de l'agent  $i$  sur ces alternatives. Soient  $A \subseteq \mathcal{X}$  et  $B \subseteq \mathcal{X}$  deux sous-ensembles d'alternatives de  $\mathcal{X}$ . On dit que  $A$  est **strictement préféré** à  $B$  pour  $i$  (on note  $A \succ_i B$ ) si et seulement si :

$$\left\{ \begin{array}{l} \forall b \in B, \forall a \in A, \neg(b \succ a) \text{ et} \\ \exists a \in A, \exists b \in B, [(a \succ b) \wedge (\forall b' \in B, b' \neq b, \neg(b' \succ a))] \end{array} \right.$$

Ainsi, un premier ensemble d'alternatives est préféré à un second si :

1. aucune alternative du second ensemble n'est meilleure qu'une alternative du premier ensemble ;
2. il y a au moins une alternative du premier ensemble qui est meilleure qu'au moins une

alternative du second et pour laquelle il n'y a aucune alternative du second ensemble qui soit meilleure.

## 10.3 Résultats

Après avoir implémenté en langage *Java* le protocole de négociation (définition 9.6, page 119), les deux stratégies (section 9.5.2, page 121), les algorithmes de génération aléatoire de préférences (algorithmes 10.1 et 10.2) ainsi que différentes métriques, nous avons effectué des expérimentations dont les résultats sont présentés dans cette section.

Nous étudions d'une part, si un agent  $i$  peut gagner à choisir la stratégie  $T$  plutôt que la stratégie  $C$  et d'autre part, la proportion des accords équitables dans le cas où tout ou partie des agents temporisent.

Nos algorithmes de génération aléatoire de préférences ne permettent pas d'atteindre des taux d'incomparabilité ou d'équivalence exacts mais uniquement bornés. Ainsi, dans les résultats qui suivent, ce sont les taux effectifs d'incomparabilité effectifs moyens qui sont présentés en abscisse des graphiques qui suivent.

### 10.3.1 Gain à temporiser

On souhaite étudier le gain pour un agent à temporiser plutôt qu'à être coopératif. Pour ce faire, nous distinguons deux situations opposées : la première où tous les autres agents coopèrent et la seconde où tous les autres agents temporisent.

#### Si les autres agents sont coopératifs

La figure 10.1 présente les résultats obtenus quand l'agent  $i$  applique la stratégie  $T$  tandis que l'ensemble des 99 autres agents appliquent la stratégie  $C$ .

Quand les autres agents sont coopératifs, la stratégie  $T$  est meilleure que la stratégie  $C$  (on a  $\sigma_{\Omega}(\mathcal{X}, C_{\Omega-i}) \succ_i \sigma_{\Omega}(\mathcal{X}, C_{\Omega})$ ) pour l'agent  $i$  dans au plus 65 % des exécutions. La dominance de  $T$  sur  $C$  atteint son plus haut avec un taux d'équivalence minimum de 0.2 et un taux d'incomparabilité proche de 0.1. Dans ces conditions, tandis que les autres agents coopèrent et proposent plusieurs alternatives à chaque tour, l'agent  $i$  peut s'assurer, en temporisant, de ne pas proposer des alternatives au rang trop bas. Dès que l'indécision des agents est trop forte, la dominance de la stratégie  $T$  devient moins fréquente. Ceci s'explique par le fait que la comparaison d'ensemble d'alternatives (définition 10.1) nécessite que les alternatives soient elles-mêmes comparables entre elles.

#### Si les autres agents temporisent

La figure 10.2 présente les résultats obtenus quand l'agent  $i$  applique la stratégie  $C$  tandis que l'ensemble des 99 autres agents appliquent la stratégie  $T$ .

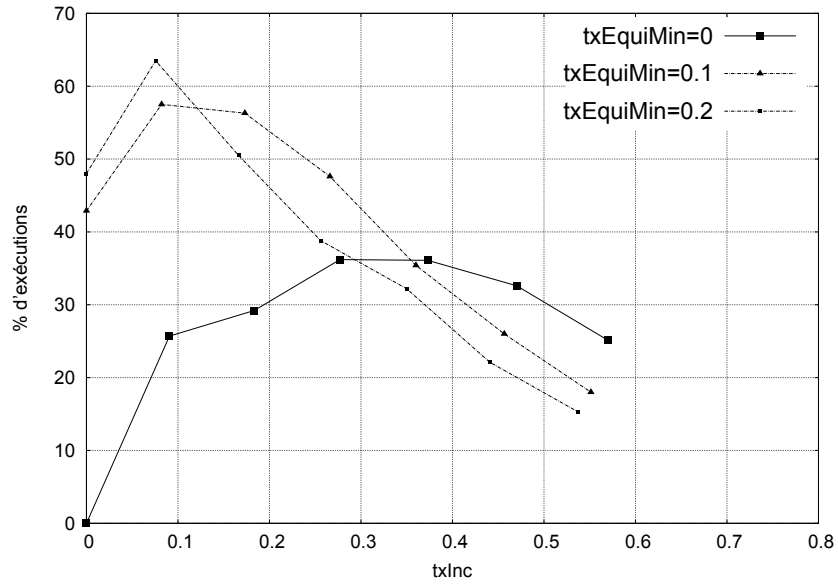


FIGURE 10.1 : Pourcentage des exécutions où  $(\sigma_{\Omega}(\mathcal{X}, C_{\Omega-i}) \succ_i \sigma_{\Omega}(\mathcal{X}, C_{\Omega}))$ .

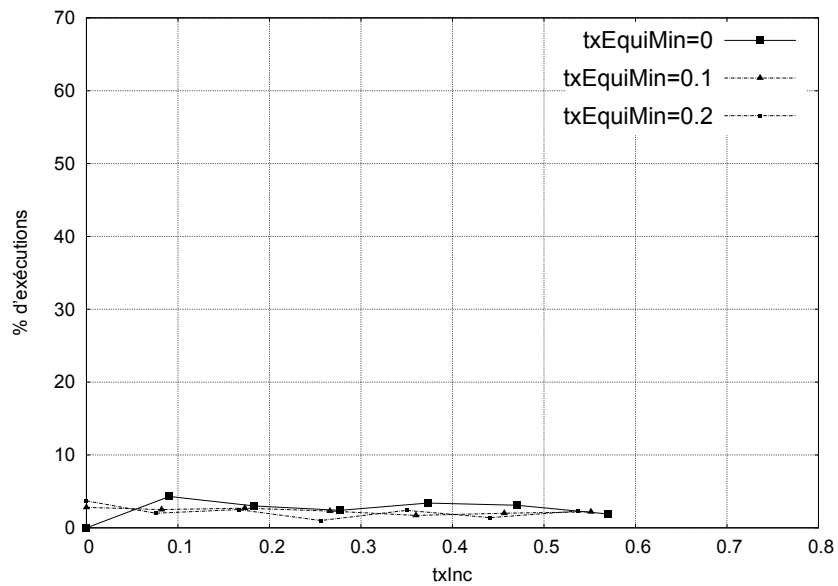


FIGURE 10.2 : Pourcentage des exécutions où  $(\sigma_{\Omega}(\mathcal{X}, T_{\Omega}) \succ_i \sigma_{\Omega}(\mathcal{X}, T_{\Omega-i}))$ .

Dans ces conditions, la stratégie  $T$  domine la stratégie  $C$  ( $\sigma_{\Omega}(\mathcal{X}, T_{\Omega}) \succ_i \sigma_{\Omega}(\mathcal{X}, T_{\Omega-i})$ ) pour l'agent  $i$  dans moins de 5 % des exécutions. Notre interprétation de ce résultat est que l'influence de l'agent sur la décision est ici réduite, chacun des autres participants ne jouant plus qu'une alternative par tour.

## Analyse

Lorsque l'ensemble des autres agents coopèrent et avec des taux d'équivalence et d'incomparabilité adéquats, un agent qui temporise peut raisonnablement espérer obtenir un meilleur accord pour lui. Cependant, dès qu'une proportion importante d'agents temporisent, l'influence du choix de stratégie de l'agent  $i$  devient limitée. En résumé, l'avantage d'employer la stratégie  $T$  plutôt que  $C$  est loin d'être systématique et peut même s'avérer nul.

Enfin, le théorème 9.2 est ici observable expérimentalement : avec des préférences strictes totales ( $txEquiMin = 0$  et  $txInc = 0$ ), les agents n'ont rien à gagner à appliquer la stratégie temporisatrice plutôt que coopérative étant donné que dans ce cas particulier, elles produisent toutes deux la même séquence de propositions.

### 10.3.2 Équité des accords

Nous évaluons ici l'équité des accords dans le cas où au moins un agent temporise.

La figure 10.3 présente le nombre d'exécutions où les accords sont équitables alors qu'un unique agent temporise ( $\sigma_{\Omega}(\mathcal{X}, C_{\Omega-i})$ ). Dans le pire des cas, les accords sont équitables dans seulement 20 % des exécutions. Il existe donc un nombre important de cas où on n'aboutit pas à un accord équitable dès qu'un agent temporise.

La figure 10.4 présente le nombre d'exécutions où les accords sont équitables alors que tous les agents temporisent ( $\sigma_{\Omega}(\mathcal{X}, T_{\Omega})$ ).

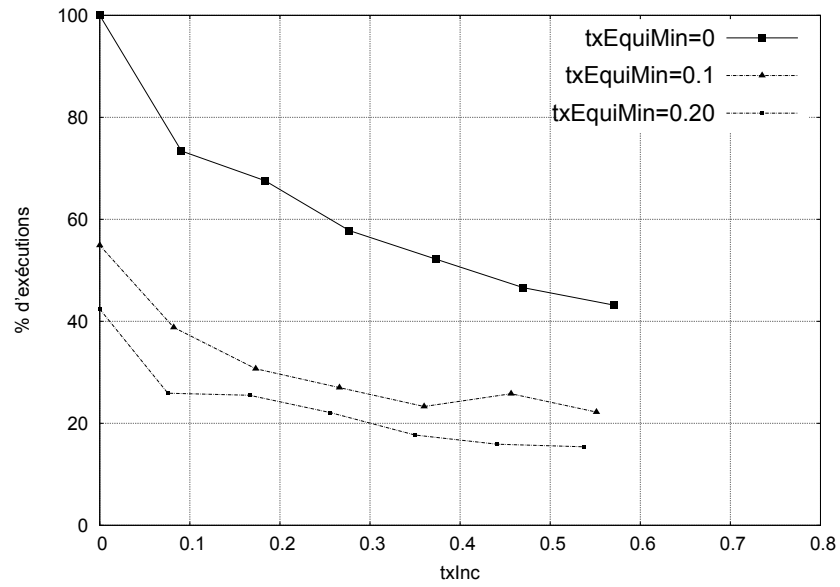
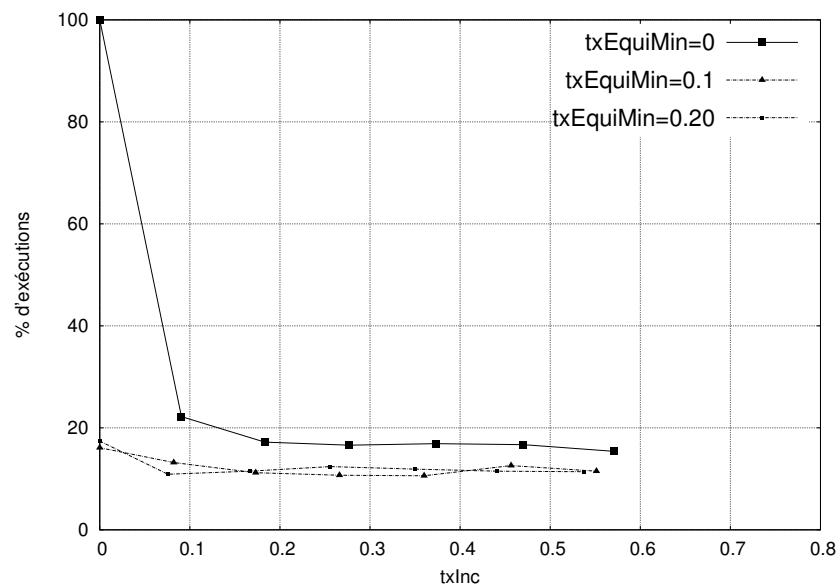
Tout d'abord, remarquons que l'on observe bien expérimentalement qu'avec des préférences strictes totales ( $txEquiMin = 0$  et  $txInc = 0$ ), si les agents qui appliquent la stratégie temporisatrice, les accords produits sont équitables. Ceci est une conséquence du théorème 9.2.

Intéressons nous à présent à la situation générale dans laquelle les préférences des agents leurs permettent de temporiser. On peut observer que l'accord alors produit est équitable dans seulement 15 à 20 % des exécutions.

### 10.3.3 Recours à l'arbitraire

Dans la figure 10.5, nous comparons pour chaque expérience, la cardinalité moyenne (sur les exécutions) des alternatives non-dominées, c'est-à-dire le résultat du processus de décision mono-agent et la cardinalité de l'ensemble des accords équitables issus d'une négociation coopérative entre des agents ayant des préférences de structure similaire (avec des taux voisins d'incomparabilité et d'équivalence). À l'échelle individuelle, plus il y a d'alternatives dont les coûts sont ex æquo - incomparables ou équivalentes, plus les cardinalités sont élevées et donc plus le recours à l'arbitraire est fréquent. Comme escompté, au niveau collectif, avec 100 agents et 25 alternatives,



FIGURE 10.3 : Pourcentage des exécutions où  $\sigma_{\Omega}(\mathcal{X}, C_{\Omega-i})$  sont des accords équitablesFIGURE 10.4 : Pourcentage des exécutions où  $\sigma_{\Omega}(\mathcal{X}, T_{\Omega})$  sont des accords équitables

l'ensemble d'accord contient dans la grande majorité des cas une unique alternative. Le recours à un tirage aléatoire arbitraire pour opérer un choix dans l'ensemble d'accord n'est alors pas nécessaire.

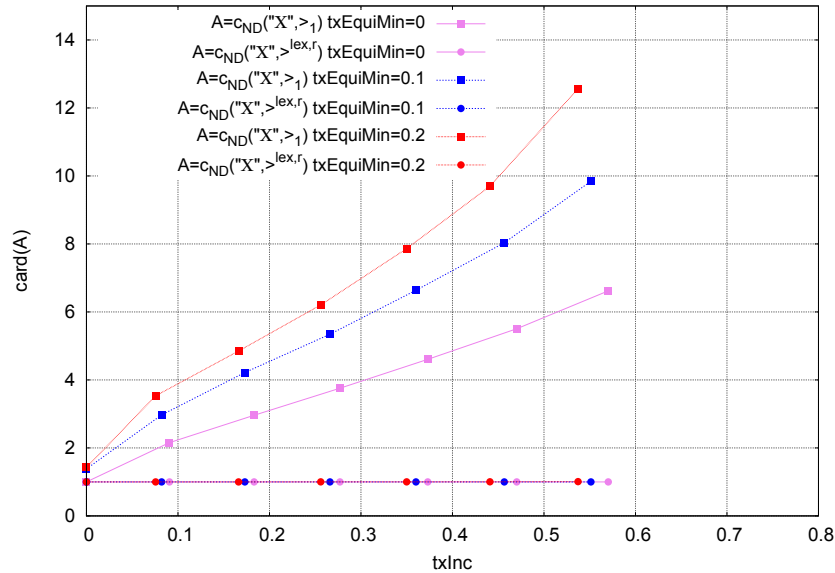


FIGURE 10.5 : Cardinalités de l'ensemble des non-dominés pour l'agent 1 et de l'ensemble d'accord quand 100 agents coopèrent ( $\sigma_{\Omega}(\mathcal{X}, C_{\Omega})$ ) en fonction des taux d'équivalence et d'incomparabilité.

## 10.4 Conclusion

### 10.4.1 Synthèse

Dans ce chapitre, nous avons mené diverses expérimentations sur notre protocole de négociation et nos stratégies en faisant varier la largeur et la complétude des préférences des agents.

Les résultats montrent que temporiser est une stratégie dominante dans des conditions assez restreintes. Dans le même temps, dans un nombre de cas non négligeable, on observe que l'adoption de la stratégie temporisatrice par une faible proportion des agents peut être suffisante pour perdre le caractère équitable de l'accord trouvé. Enfin, nous observons que si le recours à l'arbitraire est nécessaire à un agent hésitant pour décider, notre protocole mène généralement à un accord unique avec un certain nombre d'agents et d'alternatives.

### 10.4.2 Perspectives

Comme nous l'avons remarqué précédemment, lorsqu'un agent temporise, la décision produite par notre protocole n'est plus nécessairement équitable. Après avoir estimé la proportion des cas dans lesquels on perdait cette caractéristique, nous envisageons de mesurer la perte de manière quantitative. Autrement dit, nous souhaitons étudier la distance entre les vecteurs de rangs des accords obtenus et celui des accords équitables.

En outre, pour évaluer les caractéristiques sociales des accords, on peut envisager d'autres métriques que celles utilisées ici telles que le rang moyen, le rang maximal et la Pareto-optimalité.

Dans la section 10.3.3, nous avons observé qu'à partir d'un certain nombre d'agents et d'alternatives, il n'est jamais ou presque nécessaire de recourir à l'arbitraire pour décider à l'issue de notre protocole de négociation. Nous envisageons de réaliser à nouveau ces expérimentations en faisant varier le nombre d'agents et d'alternatives afin d'étudier pour quelles valeurs des paramètres du problème cette observation est généralisable.

Enfin, une autre perspective est d'évaluer les coûts de communication en terme de nombre de tours de paroles et de nombre d'alternatives proposées.



---

# Application : Coopération pour la recherche d'un point de rencontre

---

## Sommaire

---

<b>11.1 Introduction</b>	<b>139</b>
<b>11.2 Problème</b>	<b>139</b>
<b>11.3 Négociation pour la recherche d'un point de rencontre</b>	<b>141</b>
11.3.1 Modèle de préférences	141
11.3.2 Recherche collective du point de rencontre optimal	142
<b>11.4 Démonstrateur logiciel</b>	<b>144</b>
<b>11.5 Évaluation expérimentale</b>	<b>145</b>
11.5.1 Efficacité de la négociation	147
11.5.2 Coût communicationnel	147
11.5.3 Efficacité des tactiques	149
<b>11.6 Conclusion</b>	<b>149</b>
11.6.1 Synthèse	149
11.6.2 Perspectives	149

---

## 11.1 Introduction

Dans ce chapitre, nous appliquons le protocole de négociation présenté dans le chapitre 9 au problème de recherche d'un point de rencontre dans un labyrinthe. La section 11.2 décrit le problème et la section 11.3 comment notre processus peut permettre de le résoudre de manière distribuée. La section 11.4 décrit le démonstrateur logiciel réalisé pour ce problème. Dans la section 11.5, nous présentons quelques expérimentations et commentons les résultats obtenus. Enfin, la section 11.6 conclut ce chapitre en dressant quelques perspectives.

## 11.2 Problème

Dans le problème auquel nous nous intéressons, nous considérons que  $n$  agents sont parachutés dans un labyrinthe qui leur est inconnu. Leur objectif commun est de se rejoindre au plus vite.

Cependant, comme ils ne connaissent pas le labyrinthe dans son intégralité, ils ne peuvent pas calculer le point de rencontre optimal. Dans cette instanciation particulière du problème de *rendezvous* décrit dans [Alpern & Gal 2003], il est donc nécessaire pour les agents d'explorer le labyrinthe tout en cherchant à atteindre le plus rapidement possible un point de rencontre.

Afin d'illustrer ce problème, on considère le labyrinthe carré  $3 \times 3$  présenté dans la figure 11.1. Au premier tour de la résolution, Alice est sur la cellule  $c_1$  tandis que Bob est sur la cellule  $c_9$ . En considérant que les agents ont un halo de perception d'une portée d'une cellule, Alice ne voit aucun mur tandis que Bob voit uniquement le mur qui sépare les cellules  $c_8$  et  $c_9$ .

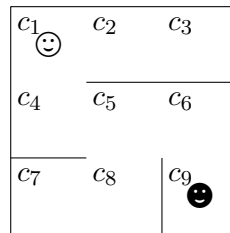


FIGURE 11.1 : Un labyrinthe de taille  $3 \times 3$  dans lequel Alice est située dans la cellule  $c_1$  et Bob dans la cellule  $c_9$ .

Un labyrinthe est défini par l'ensemble de ses cellules, leurs adjacences et les murs qui les séparent.

#### Définition 11.1 (Labyrinthe).

Un labyrinthe  $\mathcal{L}$  est un triplet  $\langle \mathcal{X}, Adj, M \rangle$  défini tel que :

- $\mathcal{X}$  est l'ensemble des cellules du labyrinthe ;
- $Adj \subseteq \mathcal{X} \times \mathcal{X}$  est l'ensemble des paires de cellules adjacentes ;
- $M \subseteq Adj$  est l'ensemble des murs du labyrinthe.

Un labyrinthe est donc un graphe non-orienté particulier. Nous considérons ici des labyrinthes qui ne sont pas nécessairement parfaits, c'est-à-dire qu'il peut exister plusieurs chemins pour aller d'une cellule à une autre.

#### Exemple 11.1 (Un labyrinthe et sa définition formelle).

Le labyrinthe représenté dans la figure 11.1 est défini par le triplet  $\langle \mathcal{X}, Adj, M \rangle$  où :

- $\mathcal{X} = \{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9\}$  ;
- $Adj = \{\{c_1, c_2\}, \{c_1, c_4\}, \{c_2, c_3\}, \{c_2, c_5\}, \{c_3, c_6\}, \{c_4, c_5\}, \{c_4, c_7\}, \{c_5, c_6\}, \{c_5, c_8\}, \{c_6, c_9\}, \{c_7, c_8\}, \{c_8, c_9\}\}$  ;
- $M = \{\{c_2, c_5\}, \{c_3, c_6\}, \{c_4, c_7\}, \{c_8, c_9\}\}$ .

Dans ce problème, tout agent  $i$  est défini par ses capacités de perception, d'action et sa mémoire.

Sa représentation interne du labyrinthe est notée  $\mathcal{L}_i$  avec  $\mathcal{L}_i = \langle \mathcal{X}, Adj, M_i \rangle$ . Il connaît la topologie de ses cellules ( $\mathcal{X}$  et  $Adj$ ) mais pas celle des murs, initialement, on a donc  $M_i = \emptyset$ . De plus, on considère qu'il sait à tout moment sur quelle cellule il est positionnée (notée  $x_i \in \mathcal{X}$ ).

Un agent perçoit son environnement : il détecte les murs présents dans son halo de perception d'une portée d'une cellule, formellement  $\{\{x_i, y\} \in M\}$ , et il met sa représentation interne des murs ( $M_i$ ) à jour.

À chaque pas de la simulation, il peut agir en se déplaçant sur une cellule adjacente si aucun mur ne l'en empêche, c'est-à-dire qu'il est capable d'atteindre toute cellule  $y \in \mathcal{X}$  telle que  $\exists\{x_i, y\} \in Adj$  et  $\nexists\{x_i, y\} \in M$  ;

Le temps mis par les agents pour se rejoindre est mesuré en nombre de pas de simulation. On considère que le problème est résolu une fois que tous les agents sont sur une même cellule. L'objectif est donc de minimiser le nombre de pas de simulation mis par l'agent atteignant en dernier le point de rencontre effectif.

## 11.3 Négociation pour la recherche d'un point de rencontre

Après avoir défini formellement dans la section précédente le problème que nous cherchons à résoudre, nous proposons dans cette section notre solution. Elle consiste à appliquer notre protocole de négociation présenté dans le chapitre 9 (définition 9.6, page 119). Pour ce faire, nous commençons par exhiber le modèle de préférences des agents dans le cadre du problème, puis, nous adoptons la stratégie coopérative afin de le résoudre.

### 11.3.1 Modèle de préférences

Dans ce problème de décision, les cellules du labyrinthe constituent l'ensemble des alternatives. Pour établir leurs préférences sur celles-ci, les agents calculent le nombre de déplacements nécessaires pour atteindre une cellule du labyrinthe. Dans notre approche, chaque agent raisonne en faisant l'hypothèse que les seuls murs du labyrinthe sont ceux qu'il a perçus. Autrement dit, par défaut, un agent considère qu'il n'y a pas de mur entre deux cellules. Un agent révisé cette croyance uniquement s'il arrive à proximité d'un mur et qu'il le perçoit. Le calcul de distance est réalisé par l'application d'un algorithme  $A^*$  reposant sur la distance de Manhattan [Russell & Norvig 1995]. Ainsi, comme l'illustre l'exemple 11.2, toute cellule est à une distance calculable.

#### Exemple 11.2 (Calcul de distance des cellules par les agents).

La figure 11.2 présente le résultat du calcul de distance aux cellules du point de vue d'Alice à gauche et de Bob à droite. Alice n'ayant connaissance à ce stade d'aucun mur, elle estime la cellule  $c_7$  à une distance de 2 cellules de la sienne. Bob quant à lui considère cette même cellule à une distance de 4 cellules de la sienne.

Chaque agent est en capacité de comparer l'ensemble des alternatives via leur distance. Pour la définir, on s'inspire de la théorie des graphes en adaptant la définition de manière à prendre en compte les murs perçus dans le labyrinthe.

#### Définition 11.2 (Fonction de distance).

Soit  $\mathcal{L}_i = \langle \mathcal{X}, Adj, M_i \rangle$ , la représentation interne du labyrinthe par l'agent  $i$  positionné sur la cellule  $x_i \in \mathcal{X}$ . La fonction de distance  $d_i : \mathcal{X} \rightarrow \mathbb{N}$  est définie telle que  $\forall x \in \mathcal{X}$  :

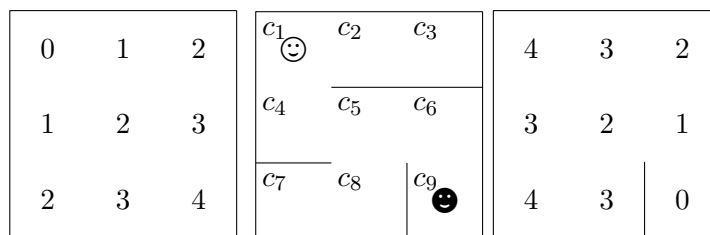


FIGURE 11.2 : Premier pas de simulation : le labyrinthe  $\mathcal{L}$  au milieu et ses représentations par les agents  $\mathcal{L}_{\odot}$  à gauche et  $\mathcal{L}_{\ominus}$  à droite.

$d_i(x) = \text{card}(\text{chemin}(x_i, x, \text{Adj}, M_i))$

où *chemin* est une fonction donnant le plus court chemin pour rejoindre la cellule  $x$  à partir de la cellule  $x_i$  étant donnés des ensembles d'adjacences de cellules et de murs.

Ici, nous considérons la fonction de distance comme une fonction de coût qui décrit les préférences de manières inverses à une fonction d'utilité. Autrement dit,  $d_i(x) \geq d_i(y) \Leftrightarrow y \succsim_i x$ . Dans le cas où deux cellules sont à une distance égale, l'agent est indifférent au choix de l'une plutôt que de l'autre.

### 11.3.2 Recherche collective du point de rencontre optimal

Pour que l'ensemble des agents se rejoignent en un nombre minimum de pas de simulation, il faut minimiser le nombre de déplacements mis au maximum par l'agent arrivant au point de rencontre le dernier. Or, notre protocole retient l'alternative dont le vecteur ordonné des tours est *leximin*-optimal. Il suffit donc que chaque agent  $i$  propose à chaque tour  $t \geq 1$ , l'ensemble des alternatives  $x$  à une distance  $d_i(x) = t - 1$  comme illustré dans l'exemple 11.3.

**Exemple 11.3 (Recherche d'un point de rencontre par Alice (☺) et Bob (☹)).**  
 Considérons les distances décrites dans l'exemple 11.2. Dans le tableau 11.1, nous représentons la trace d'une négociation où Alice et Bob coopèrent pour rejoindre un point de rencontre. Au tour  $t = 1$ , chacun des deux agents propose la cellule sur laquelle il est positionné et qui a donc une distance de 0 :  $c_1$  pour ☺ et  $c_9$  pour ☹. Au tour n° 2, les deux agents proposent leurs alternatives à une distance de 1. Comme aucune alternative n'a été proposée par les deux agents, un troisième tour a lieu où les cellules  $c_3$  et  $c_5$  sont proposées par les deux agents. La négociation prend donc fin et ces deux alternatives sont des accords.

Dans le cadre de la recherche d'un point de rencontre dans un labyrinthe, la stratégie coopérative consistant à proposer l'ensemble des alternatives de rang le plus bas parmi les inédites revient à proposer l'ensemble des alternatives de distance minimale parmi les inédites. En effet, le rang d'une alternative pour un agent est égal à la distance augmentée de 1 de la cellule correspondante. Par conséquent, dans le contexte de notre problème, la stratégie coopérative (définition 9.7, page 122) est équivalente à l'algorithme 11.1.



$t$	☺	☹
1	$\{c_1\}$	$\{c_9\}$
2	$\{c_2, c_4\}$	$\{c_6\}$
3	$\{c_3, c_5, c_7\}$	$\{c_3, c_5\}$
$\sigma_\Omega(\mathcal{X}, C_\Omega) = \{c_3, c_5\}$		

TABLEAU 11.1 : Trace de négociation où Alice et Bob coopèrent pour se rejoindre au plus tôt

---

**Algorithme 11.1** : Stratégie Coopérative : envoi des cellules de distance minimale parmi les restantes

---

**Données** :  $\mathcal{X}$  : l'ensemble des cellules,  $t$  : le tour courant

1 *proposer*( $\{x \in \mathcal{X} \mid d_i(x) = t - 1\}$ );

---

L'ensemble des agents trouvent un ensemble d'accord en appliquant cette stratégie. En cas d'égalité, ils procèdent à un tirage à aléatoire pour choisir une unique cellule. Une fois la cellule fixée, les agents empruntent le chemin le plus court – selon leur connaissance du labyrinthe – pour la rejoindre. Après chaque déplacement d'un agent  $i$ , les distances des cellules sont recalculées. Cependant, celles-ci restent statiques au cours d'une même itération du processus de négociation.

**Exemple 11.4 (Réévaluation des distances au deuxième pas de simulation).**

*Considérons qu'à l'issue de la négociation présentée dans l'exemple 11.3, la cellule  $c_3$  ait été tirée aléatoirement. L'agent ☺ se déplace donc sur la cellule  $c_2$  tandis que l'agent ☹ rejoint la cellule  $c_6$ . Après leur déplacement, les agents perçoivent leur environnement et mettent à jour la représentation qu'ils en ont en ajoutant les nouveaux murs perçus le cas échéant :  $M_{☺} = \{\{c_2, c_5\}\}$  et  $M_{☹} = \{\{c_8, c_9\}, \{c_3, c_6\}\}$ . Les distances sont alors réévaluées comme illustré dans la figure 11.3. Les agents entament une nouvelle négociation, qui aboutit à un accord sur les cellules  $c_2$  et  $c_6$  qui ont toutes deux un vecteur de distance leximax-optimal.*

1	0	1	$c_1$	$c_2$ ☺	$c_3$	3	2	3
2	3	2	$c_4$	$c_5$	$c_6$ ☹	2	1	0
3	4	3	$c_7$	$c_8$	$c_9$	3	2	1

FIGURE 11.3 : Deuxième pas de simulation : le labyrinthe  $\mathcal{L}$  au milieu et ses représentations par les agents  $\mathcal{L}_{☺}$  à gauche et  $\mathcal{L}_{☹}$  à droite.

Comme l'illustre l'exemple 11.3, en ne connaissant pas l'emplacement de tous les murs, les agents sont susceptibles de sous-évaluer les distances et d'emprunter de mauvais chemins pour rejoindre une cellule.

**Exemple 11.5 (Fin de la simulation et rencontre des agents).**

*Supposons ici qu'après tirage aléatoire, la cellule  $c_2$  ait été tirée aléatoirement comme nou-*

veau point de rencontre. L'agent ☹ est déjà au point de rencontre, il n'a pas à se déplacer ; l'agent ☺ se déplace en cellule  $c_5$  et perçoit un mur entre  $c_5$  et  $c_2$  (figure ). Après négociation et deux pas de simulation supplémentaires, les agents se rejoindront soit en  $c_1$ , soit en  $c_4$ .

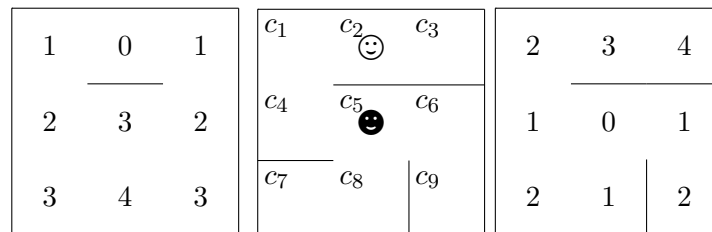


FIGURE 11.4 : Troisième pas de simulation : le labyrinthe  $\mathcal{L}$  au milieu et ses représentations par les agents  $\mathcal{L}_{☹}$  à gauche et  $\mathcal{L}_{☺}$  à droite.

En cherchant à rejoindre le dernier point de rencontre fixé, les agents découvrent potentiellement de nouveaux murs et réévalue les distances. Une nouvelle itération de la négociation vient alors corriger le point de rencontre initialement choisi. Ainsi, les agents finissent nécessairement par se rejoindre.

## 11.4 Démonstrateur logiciel

Le démonstrateur <sup>1</sup> a été réalisé dans le langage *Java* par Thomas Nachtergaele lors d'un projet de Master Informatique à l'Université Lille 1 en réutilisant l'implémentation que j'ai réalisée de notre protocole de négociation. La figure 11.5 présente une vue de son interface. Le logiciel comprend un modèle d'environnement, un modèle d'agent ainsi que l'implémentation de notre protocole de négociation.

Le modèle d'environnement est implémenté sous la forme d'un graphe non-orienté qui permet la génération aléatoire de labyrinthe de différentes topologies :

- des labyrinthes avec de nombreux petits cul-de-sac et la solution est généralement plutôt directe, générés par une version pseudo-aléatoire de l'algorithme de Prim ;
- des labyrinthes ayant un petit nombre de longs cul-de-sac, générés par un algorithme de recherche en profondeur d'abord, où la solution est généralement très longue et sinueuse.

Un agent est capable de :

- percevoir l'environnement ;
- de se déplacer ;
- de calculer le plus court chemin entre sa position et un point de rencontre ;
- de générer ses préférences à partir de ces calculs ;
- de communiquer avec les autres agents.

L'interface de l'application repose sur la bibliothèque graphique *Swing*. L'exécution d'une simulation (cf. figures 11.5 et 11.6) permet de visualiser :

- la vue globale de l'environnement qui présente la trace des agents ;

1. téléchargeable à l'URL suivante : <http://www.lifl.fr/~morge/software/#NegoMaze>

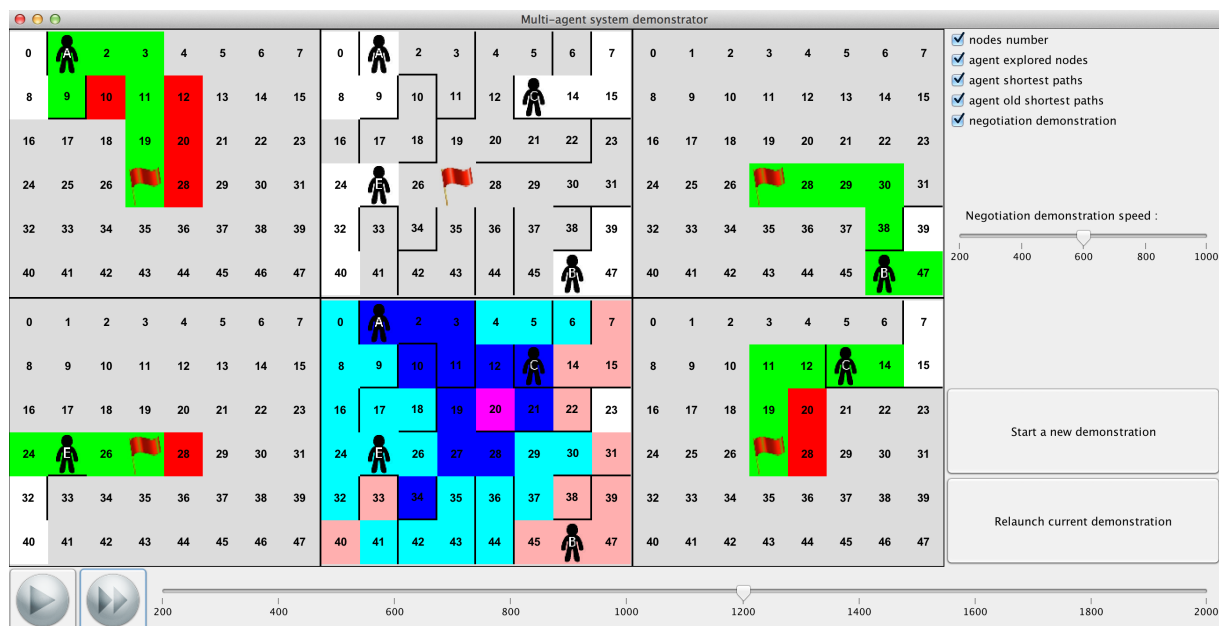


FIGURE 11.5 : Capture d'écran du démonstrateur. Parmi les six panneaux, celui au centre en haut permet d'observer la progression des agents, celui en-dessous révèle les offres échangées, les quatre autres représentent la perception locale de chaque agent.

- la représentation interne de chacun des agents, notamment les calculs de chemin ;
- les différentes étapes du processus de négociation et l'accord atteint.

Comme présenté dans la figure 11.7, le démonstrateur permet de configurer pour une simulation :

- le nombre d'agents à placer dans le labyrinthe initialement ;
- le halo de perception des agents ;
- l'algorithme de génération du labyrinthe ;
- la taille du labyrinthe ;
- la fréquence de négociation. Celle-ci intervient tous les  $n$  pas de simulation ou si au moins l'un des agents doit faire un détour pour atteindre le point de rencontre parce qu'il rencontre un mur.

De par sa généralité, l'outil développé constitue un véritable tube à essai qui permet d'évaluer notre protocole de négociation sur des problèmes de grandes tailles.

## 11.5 Évaluation expérimentale

Nos expérimentations visent à comparer l'efficacité de la résolution distribuée avec la négociation ainsi que son coût communicationnel. Elles ont été effectuées pour un nombre d'agents égal à 2, le modèle n'ayant pas encore été étendu à  $n$  agents au moment de leur réalisation.

Pour chaque expérience, nous générons aléatoirement un nouveau labyrinthe de taille  $n \times n$  et y plaçons aléatoirement les agents. Nous essayons ensuite de résoudre le problème correspondant en :

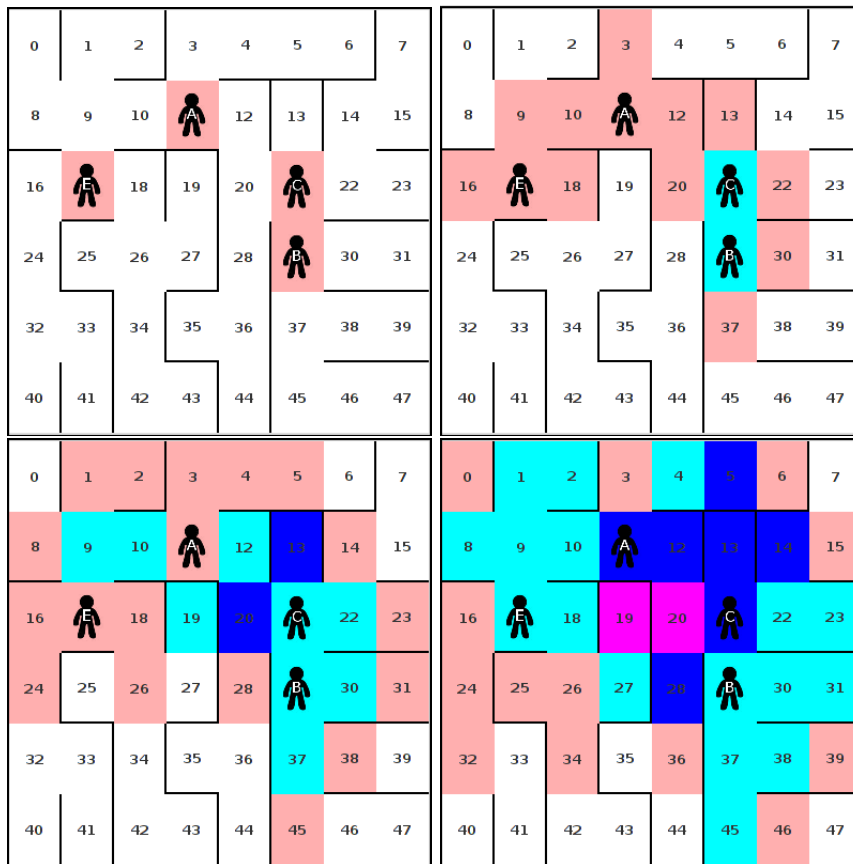


FIGURE 11.6 : De gauche à droite et de haut en bas, tours n° 1 à 4 de la négociation d'un point de rencontre entre les agents *A*, *B*, *C* et *E*. En rose, les cellules proposées par un seul agent, en cyan par deux, en bleu foncé par trois et en violet par quatre. Ces dernières constituent l'ensemble d'accords.

Demonstrator configuration	
Number of agents ( 2 to 4):	4
Agent vision range :	1
Maze height ( 1 to 20) :	8
Maze width ( 1 to 20) :	6
Additional breaking wall percentage :	0
Maze generation algorithm :	prim
Frequency of meeting point computation :	each step
Meeting point fixing strategy :	negotiation
Launch demonstrator	

FIGURE 11.7 : Panneau de configuration du démonstrateur

1. sélectionnant pseudo-aléatoirement un point de rencontre dans le labyrinthe ;
2. utilisant notre protocole de négociation à chaque pas de simulation ;
3. utilisant notre protocole de négociation tous les 2 pas de simulation ;
4. utilisant notre protocole de négociation tous les 4 pas de simulation ;
5. utilisant notre protocole de négociation à chaque pas de simulation où au moins un agent doit recalculer son chemin le plus court après avoir découvert un nouveau mur.

La première tactique est naïve et sert de point de comparaison. Elle ne nécessite aucune communication et consiste simplement à choisir aléatoirement une cellule où se rencontrer dans le labyrinthe. Les quatre tactiques suivantes mettent en œuvre notre protocole de négociation en l'itérant à des fréquences différentes.

Nous considérons  $n \in [2; 16]$  et pour chaque  $n$ , nous générons 100 expériences.

### 11.5.1 Efficacité de la négociation

Nous souhaitons étudier le nombre maximum de déplacements effectués par un agent ainsi que le nombre total des déplacements effectués par les deux agents. La figure 11.8 présente les résultats obtenus avec un labyrinthe généré par notre version de l'algorithme de Prim, la figure 11.9 décrit quant à elle les résultats quand l'algorithme de recherche en profondeur d'abord est utilisé.

Négocier plutôt que choisir aléatoirement un point de rencontre est plus efficace et réduit le nombre de pas de simulation nécessaires à la rencontre des agents. De manière plus surprenante, la satisfaction globale des agents est améliorée par la négociation. En d'autres termes, la négociation aide à trouver le chemin le plus court.

### 11.5.2 Coût communicationnel

Dans les figures 11.10, 11.11 et 11.12, nous comparons les coûts de communication de nos différentes stratégies en terme de nombre de négociations, nombre total de tours de négociations et nombre d'alternatives proposées. Les résultats sont présentés pour des labyrinthes générés avec

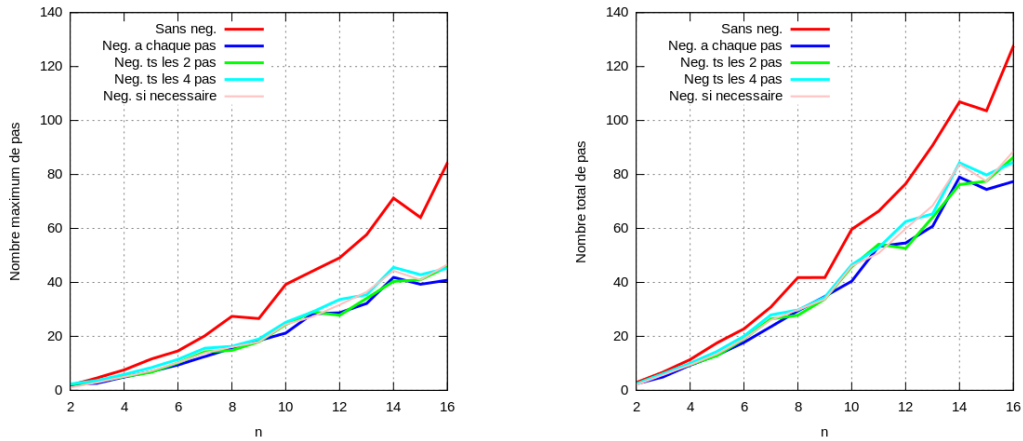


FIGURE 11.8 : Nombre maximum de déplacements effectués par un agent (à gauche) et nombre total de déplacements effectués par les deux agents (à droite) avec un labyrinthe généré par une version pseudo-aléatoire de l'algorithme de Prim.

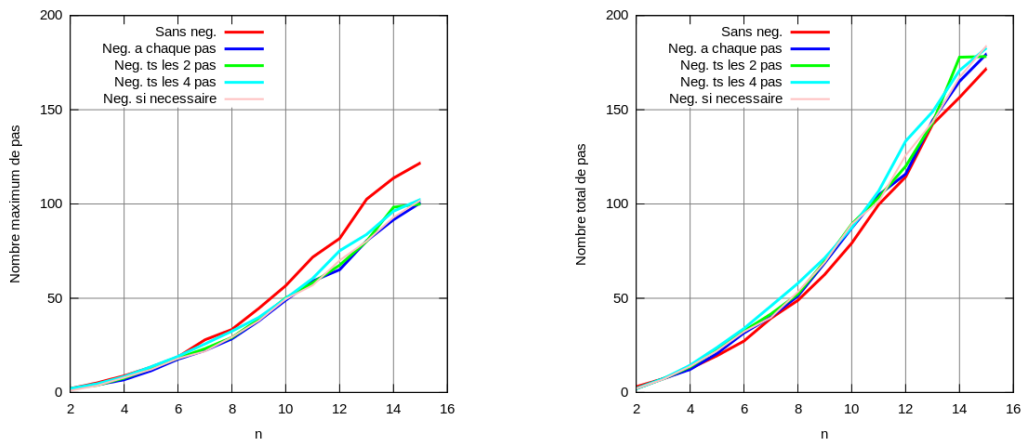


FIGURE 11.9 : Nombre maximum de déplacements effectués par un agent (à gauche) et nombre total de déplacements effectués par les deux agents (à droite) avec un labyrinthe généré par un algorithme de recherche en profondeur d'abord.

l'algorithme de Prim mais sont très similaires en utilisant l'algorithme de profondeur d'abord. Négocier à chaque pas de simulation diminue le coût de communication avec des résultats similaires pour le nombre de déplacement. Il semble que les coûts de communication des tactiques (3) et (5) sont très similaires.

### 11.5.3 Efficacité des tactiques

Nous générons à présent un labyrinthe de taille  $8 \times 8$  avec la version pseudo-aléatoire de l'algorithme de Prim et y positionnons les agents aléatoirement. Nous détruisons partiellement les murs jusqu'à ce que la densité (le nombre de murs divisé par le nombre initial de murs) soit égale à  $d$  %. Ensuite, nous essayons de résoudre le problème correspondant avec l'aide de nos 5 tactiques. Nous considérons  $d \in [0; 100]$ . Pour chaque  $d$ , nous générons 100 expériences. Dans les figures 11.13 et 11.14, nous comparons la tactique de négociation à celle du choix aléatoire d'un point de rencontre. Nous observons que le nombre de pas de simulation nécessaires à la rencontre des agents est inférieur quand ils négocient plutôt qu'en définissant un point aléatoire.

De plus, l'efficacité des différentes tactiques est difficile à déterminer. Cependant, ces stratégies ont des coûts de communication différents. Par exemple, le coût de la tactique n° 5 est proche de celui de la n° 4 quand la densité est basse et proche de celui de la tactique n° 3 quand la densité est haute.

En résumé, quelque soit la densité du labyrinthe, la tactique n° 5 est performante à la fois sur le critère du nombre de tours requis nécessite pour que les agents se rencontrent ainsi que sur le critère du coût de communication.

## 11.6 Conclusion

### 11.6.1 Synthèse

Dans ce chapitre, nous avons appliqué notre protocole de négociation au problème de recherche d'un point de rencontre dans un labyrinthe. Nous avons ensuite présenté un logiciel permettant de résoudre ce type de problème. Ce démonstrateur est un outil pédagogique permettant de visualiser l'application de notre protocole de négociation dans une optique de résolution distribuée de problème. À partir de celui-ci, nous avons procédé à des expérimentations où deux agents cherchent à se rejoindre dans des labyrinthes de différentes tailles et typologies, en employant diverses tactiques de communication. Nos expériences montrent que la négociation répétée au fur et à mesure de l'exploration du labyrinthe permet un bon compromis entre quantité de déplacements et quantité de communications.

### 11.6.2 Perspectives

Dans le cadre de ce problème, l'itération répétée de notre protocole permet à un ensemble d'agents de se rejoindre rapidement dans un labyrinthe. Cependant, dans l'exemple 11.4, on constate que les agents auraient pu opter pour un meilleur choix. Pour pallier à cette situation,

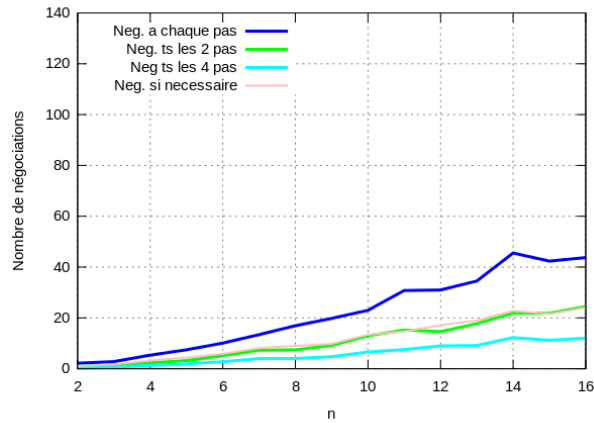


FIGURE 11.10 : Nombre de négociation en fonction de la taille du labyrinthe

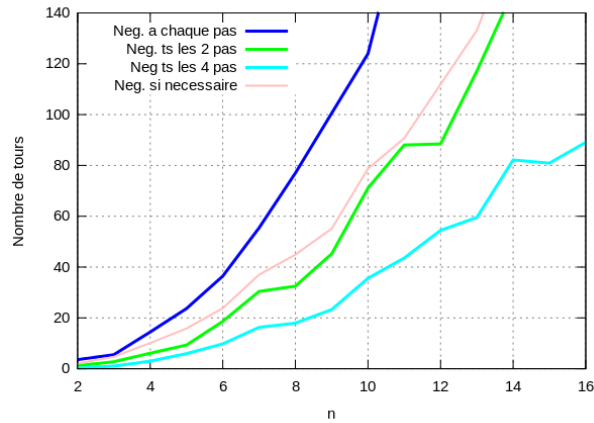


FIGURE 11.11 : Nombre total de tours de négociation en fonction de la taille du labyrinthe

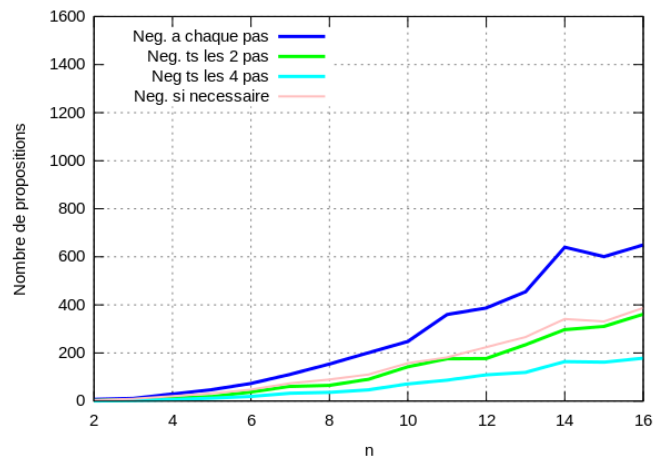


FIGURE 11.12 : Nombre d'alternatives proposées en fonction de la taille du labyrinthe



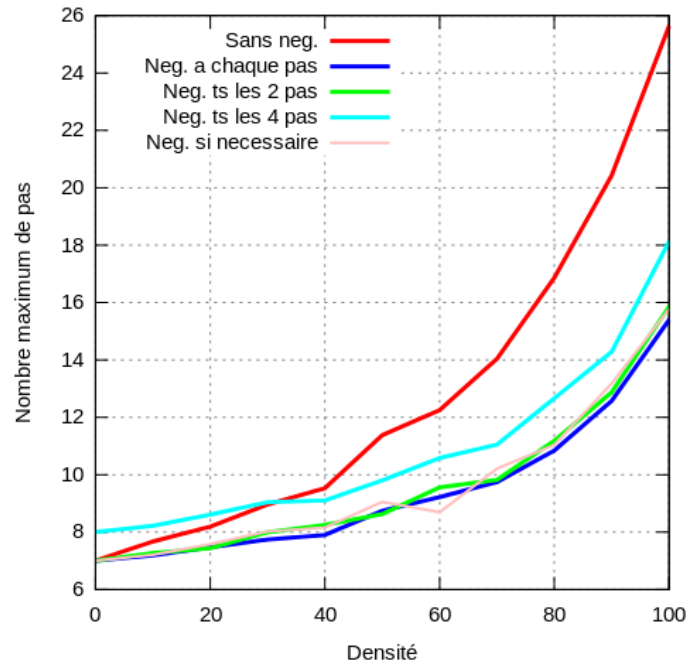


FIGURE 11.13 : Nombre maximum de pas de simulations nécessaires à la rencontre en fonction de la densité du labyrinthe.

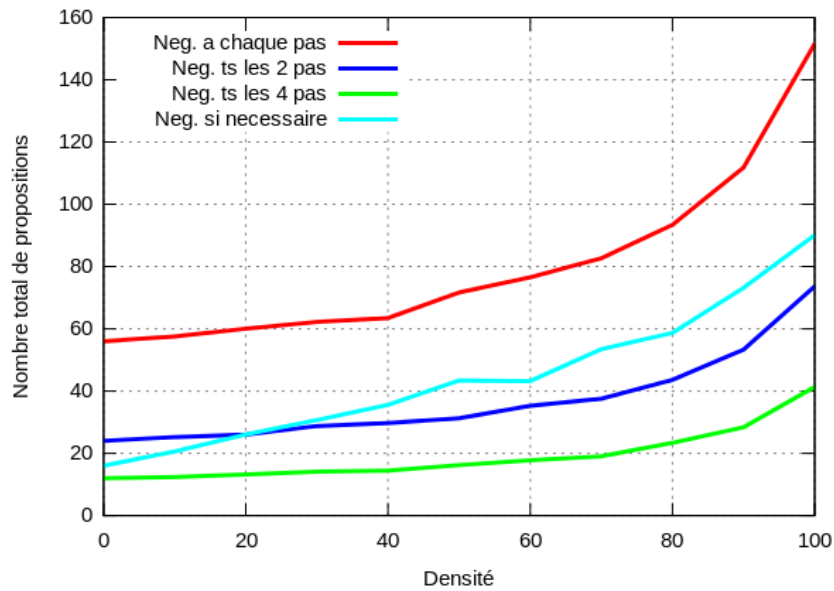


FIGURE 11.14 : Nombre maximum de propositions en fonction de la densité du labyrinthe.

on peut adapter la stratégie des agents au problème en inférant la présence de murs à partir des cellules proposées par les autres participants.

Bien que le démonstrateur se limite à l'affichage de labyrinthes, la modélisation de l'environnement est suffisamment générique pour envisager de s'attaquer à d'autres types de graphe (par exemple les *small worlds*). De plus, nous envisageons de généraliser le problème en pondérant le graphe du labyrinthe. Plutôt que chaque paire de cellules adjacentes soient, de manière binaire, séparées ou non par un mur, une distance ou un coût serait associée sous forme numérique.

Enfin, nos expérimentations ayant été menées avec deux agents, nous envisageons de les élargir à  $n$  agents afin d'étudier si les résultats sont sensiblement différents ou non.

---

# Conclusion

---

Ce chapitre conclut cette thèse. Pour chacun des deux axes de la thèse, les contributions sont synthétisées, leurs limites dessinées et des perspectives esquissées.

## Recommandation experte proactive

Sur ce thème, notre objectif de recherche était de concevoir un agent capable d'avoir un dialogue cohérent avec le décideur et de lui soumettre des recommandations pertinentes.

### Contributions

Le cœur de notre contribution prend la forme d'un modèle d'Agent Dialogique Proactif (ADP) pour des problèmes de décision multi-attributs. Sa principale caractéristique est de diriger le dialogue avec l'utilisateur humain afin de lui faire part de recommandations fondées sur son expertise et les informations acquises au cours du dialogue. Autrement dit, notre agent pose des questions et émet des propositions argumentées, tandis que l'agent humain y répond. En suivant ce schéma de dialogue asymétrique, nous délimitons le champ de la conversation et facilitons l'interprétation des réponses de l'interlocuteur humain. L'architecture de l'agent est découpée en deux modules. Le premier module est dédié à la représentation et à l'acquisition de connaissances : il permet l'implémentation de l'expertise métier de l'agent, c'est-à-dire d'une ontologie du domaine, de la description de l'ensemble des alternatives et de règles expertes. Le second module est dédié à la gestion du dialogue et comprend deux mécanismes. Le premier est un mécanisme de raisonnement dialogique, qui consiste pour l'ADP à établir, étant donné les règles et faits dont il dispose, quelle question est la plus pertinente à poser pour progresser dans sa connaissance des attentes du décideur. Le second est un mécanisme de sélection, qui consiste à trouver quelles sont les alternatives et arguments pertinents à partir des informations collectées dans le dialogue avec le décideur.

Notre proposition a été appliquée dans le cadre du projet PICOM Vendeur Virtuel Ubiquitaire, visant à améliorer l'expérience d'achat en ligne. Cette implémentation sur trois cas concrets de commerce électronique valide notre modèle. Les connaissances de l'agent sont représentées sous la forme de vecteurs d'utilité et son raisonnement dialogique est développé sous la forme de stratégies d'identification et de proposition indépendantes du type de produits vendus.

### Perspectives

Notre contribution présente des limites que nous essayons d'identifier ici.

Pour qu'un agent dialogique proactif soit opérationnel, il est nécessaire qu'il ait accès à de nombreuses connaissances formalisées : l'ontologie du domaine, la description des alternatives et un ensemble de règles d'inférence. La saisie de ces connaissances est une étape cruciale pour le bon fonctionnement et l'efficacité de l'agent. Or, celle-ci n'est pas facile, en particulier pour un non-initié. Nous considérons que la solution pour pallier à cette difficulté peut prendre la forme d'un environnement de développement intégré dédié. Concernant les mécanismes de raisonnement de l'agent, ils peuvent, pour partie au moins, être communs à différents contextes et domaines applicatifs. Une des perspectives est alors d'implémenter des heuristiques sous la forme d'une librairie documentée, celles-ci pourraient faciliter le développement de l'agent mais aussi servir à titre d'exemples.

L'approche du dialogue dirigé par l'assistant pour laquelle nous avons opté ne permet pas au décideur humain de s'exprimer aussi librement qu'il pourrait le souhaiter. En effet, à la frustration de capacités insuffisantes de l'agent pour traiter convenablement des énoncés libres en langage naturel et garantir une cohérence au dialogue, nous avons préféré celle d'une interaction certes plus cloisonnée mais mieux maîtrisée. Cependant, on peut envisager laisser davantage de latitude au décideur en lui permettant de prendre l'initiative d'exprimer certaines préférences ou données du problème dans le champ dialogique maîtrisé par l'agent. Cette problématique est étroitement liée à celle de l'interface de dialogue que nous avons choisi de ne pas traiter en profondeur dans cette thèse.

Sur le plan applicatif, on peut identifier certaines limites spécifiques à notre Vendeur Virtuel Ubiquitaire (VVU). En particulier, le choix d'un modèle de préférences sous la forme d'une variante d'une agrégation additive ne permet pas l'expression de dépendances préférentielles. En effet, dans certains domaines, des synergies ou au contraire des antagonismes existent entre différents critères. Pour les exprimer et les prendre en compte dans le raisonnement, on peut envisager d'autres modèles de préférence. Par ailleurs, le cadre commercial pose la question du but de l'agent assistant. En effet, souvent, le conseiller est aussi vendeur et ses buts peuvent alors entrer en contradiction. Dans notre implémentation, le VVU est situé sur le serveur et accompagne le client sur le site de l'enseigne. Le client peut alors s'interroger sur la bienveillance de l'agent et se demander si ses recommandations sont réellement pertinentes ou si, au contraire, elles ont pour objectif premier de maximiser les marges de l'enseigne. Pour éviter ce problème de confiance, on peut imaginer une implémentation du VVU comme un logiciel libre s'exécutant côté client (par exemple sous la forme d'un module du navigateur) en consultant la description des alternatives sur le serveur de l'enseigne. Cette solution présenterait de plus l'avantage de permettre la confidentialité des données pour le client.

Au-delà du commerce électronique, l'ADP peut s'appliquer à bien des domaines. En particulier, on peut imaginer des applications sur des problématiques d'orientation ou de recherche d'emploi, qu'on peut considérer comme étant des problèmes de décision multi-attributs. De manière générale, l'ADP peut constituer une alternative à des interfaces qui proposent de choisir dans un large éventail de possibilités que l'utilisateur méconnaît.

Enfin, on peut notamment élargir le champ d'applications de notre modèle d'agent dialogique proactif aux jeux vidéo. En particulier, cette approche peut permettre d'implémenter des personnages non joueurs aux capacités dialogiques renforcées, que ce soit pour des jeux vidéo de rôle classique mais aussi pour des jeux sérieux, par exemple dans l'objectif de former aux métiers du conseil et de la vente [Mathieu *et al.* 2012].

## Prise de décision multi-agents équitable

Notre objectif de recherche était double : caractériser des alternatives socialement équitables à partir de préférences pouvant être incomplètes puis, élaborer un processus de décision collective permettant d'atteindre ces alternatives.

### Contributions

Dans cette thèse, nous avons défini le critère de la *leximax*-optimalité sur les rangs des alternatives sous l'appellation d'alternative équitable. Ce critère vise à l'équilibre des efforts effectués par les différents agents pour atteindre un accord. Il présente de plus des caractéristiques intéressantes : la Pareto-inclusivité, la garantie d'existence et la compatibilité avec des préférences partielles.

Une fois les alternatives équitables définies, nous avons proposé un processus pour les atteindre. Celui-ci prend la forme d'un protocole de négociation multilatéral qui consiste en un jeu simultané répété. Notre protocole garantit l'obtention d'un accord. Nous avons implémenté deux stratégies conformes à ce protocole : l'une, coopérative, permet à des agents poursuivant le but commun d'atteindre un accord équitable d'y parvenir ; l'autre temporisatrice, égo-centrée, au service des préférences de l'agent qui l'emploie, consiste à proposer une seule alternative par tour. Si ces stratégies sont équivalentes quand les préférences des agents sont strictes et totales, ça n'est pas le cas avec des relations larges et/ou partielles. En effet, sous cette hypothèse, lorsqu'au moins un agent temporise, la Pareto-optimalité de l'accord n'est plus garantie. Autrement dit, temporiser permet potentiellement d'améliorer son résultat sur le critère individuel en l'affaiblissant sur le critère social.

Après avoir implémenté notre processus, nous avons étudié expérimentalement le gain individuel à jouer temporisateur plutôt que conciliant. De nos expériences, il ressort que temporiser est une stratégie dominante dans des conditions assez restreintes. De plus, on constate que dans de nombreux cas l'emploi de la stratégie temporisatrice, même par une faible proportion d'agents, suffit à diminuer la qualité sociale de l'accord obtenu.

Enfin, nous avons illustré notre processus par une application au problème de recherche d'un point de rencontre dans un labyrinthe. Dans ce problème de résolution distribuée, des agents cherchent à se rejoindre dans des labyrinthes de tailles et de typologies différentes. La négociation répétée au fur et à mesure de l'exploration permet de réduire le nombre de pas de simulation pour atteindre un point de rendez-vous. Différentes tactiques de communication ont été testées expérimentalement, la plus adaptative, celle qui consiste à renégocier dès qu'un mur a été découvert, présente un bon compromis entre coût de communication et nombre de déplacements.

### Perspectives

Bien qu'ayant d'ores et déjà réalisé des évaluations expérimentales de notre processus de décision, nous envisageons de les approfondir davantage. En particulier, on peut envisager d'évaluer les coûts de communication engendrés par notre protocole en terme de nombre d'alternatives proposées et de nombre de tours de paroles requis. D'autre part, nous avons évalué le caractère

équitable ou non des accords lorsqu'une partie des agents temporisent. Nous pouvons raffiner cette évaluation en introduisant une métrique de distance à un accord équitable.

La règle de décision collective que nous avons adoptée dans cette thèse est la *leximax*-optimalité sur les rangs des alternatives. Nous pourrions envisager d'autres règles pour capturer l'équité comme celle du produit de Nash. Ce dernier agrégateur nous semble particulièrement pertinent comme approche intermédiaire pour avoir un rang moyen plus faible tout en limitant le nombre d'agents pour lesquels le rang de l'accord est élevé. À cette intention, le protocole doit être adapté, en particulier la règle qui définit l'ensemble d'accords.

D'une manière générale, on peut envisager d'adapter notre processus de décision collective à n'importe quelle règle de choix social (utilitaire, égalitaire, etc.). Dans cette optique, la perspective est de développer une plate-forme de processus de décision collective. Cette idée est à rapprocher de [Verrons 2004] qui propose un modèle général de négociation de contrats entre agents. Dans notre plate-forme, la règle de décision collective serait paramétrable ainsi que le modèle de préférences des agents. Les méthodes de prise de décision collective décrites dans la littérature présentent chacune des caractéristiques différentes et aucune d'entre elles n'est, de manière absolue, meilleure à toute autre. C'est pourquoi, plutôt que de les opposer les unes aux autres et d'entrer sur un terrain moral, nous envisageons d'en proposer un large éventail.

Les décisions collectives sont courantes en société. Pour preuve, les sondages en ligne ont un succès grandissant<sup>2</sup>. Les usages imaginables pour ce genre d'outils sont nombreux, à titre illustratif pour le lecteur, nous choisissons notamment d'évoquer :

- la prise de décision d'un groupe d'amis souhaitant choisir un restaurant ou un film ;
- les élections et prises de décision au sein d'associations, d'entreprises, etc. ;
- la prise de décision communautaire en ligne sur des projets collectifs participatifs tels Wikipédia ou des réseaux sociaux ;
- la planification de réunions.

La méthode que nous proposons dans notre contribution nous semble particulièrement pertinente pour ces cas d'application. En effet, si les agents ont à la fois leurs propres préférences mais aussi une bienveillance envers les autres agents, on peut considérer que leur satisfaction dépend aussi de celle des autres participants. On est alors dans un cas hybride que certains qualifient de co-opétition. Utiliser le protocole que nous avons présenté accompagné d'une stratégie coopérative semble alors rationnel et particulièrement pertinent.

---

2. <https://framadate.org/>

---

# Éléments d'expertise dans le domaine de la vente

---

## A.1 Analyse des mécanismes de vente

Un mécanisme de vente est un procédé ayant pour but d'aboutir à la vente d'un produit. Un tel procédé est composé d'un ensemble d'opérations parmi lesquelles :

- l'acquisition d'informations, dans le but de formuler des propositions pertinentes, c'est-à-dire en adéquation avec les données disponibles du scénario ;
- la formulation de propositions argumentées dans la perspective de conclure la vente.

À ces deux opérations viennent s'ajouter des processus internes au vendeur. Il s'agit d'une part de déterminer les informations utiles à la vente, et d'autre part de déduire à partir de ces informations quelles sont les alternatives et arguments pertinents. Cette problématique se situe donc dans un cadre d'application directe des hypothèses de l'ADP. On note en particulier qu'on souhaite que le VVU présente des arguments pertinents pour soutenir ses propositions.

Un mécanisme de vente s'appuie sur un ensemble de données. La nature, le traitement et l'usage de ces données sont les éléments sur lesquels repose le mécanisme. Ici, on considère la vente par une enseigne d'un produit à un client dans un contexte. Les données correspondantes à chacun de ces quatre paramètres peuvent servir à différents égards dans le mécanisme de vente. Il s'agit ici d'en faire le détail et de déterminer leurs rôles respectifs dans le mécanisme. L'objectif est notamment d'identifier les données pertinentes dans la mise en œuvre d'un mécanisme de vente en ligne, c'est-à-dire utiles, accessibles et dont l'utilisation est acceptable par le client et l'enseigne.

Les **données du client** caractérisent d'une part l'acheteur du produit et d'autre part l'utilisateur de ce produit. Cette distinction a plusieurs intérêts : en plus de séparer explicitement les deux aspects qui ont des implications différentes au niveau de la vente, elle permet de traiter le cas où l'acheteur ne recherche pas un produit pour sa propre utilisation. Ainsi :

- les **données utilisateur** permettent d'identifier un produit qui corresponde à son ou ses utilisateurs et de servir d'arguments pour soutenir une offre. Typiquement, on retrouve dans cet ensemble les caractéristiques objectives de l'utilisateur (sexe, taille, âge, lieu d'habitation, etc.), les usages qu'il souhaite faire du produit (compétition, loisir, etc.), ses préférences (couleur, motif, marque, etc.). Ces données permettent d'établir une adéquation entre utilisateur(s) et produit via des règles métiers du domaine. Cette adéquation peut être physique, entre les caractéristiques objectives du client et du produit (par exemple la

taille du produit en fonction de celle de son utilisateur). De façon similaire, il est possible à partir des usages souhaités par l'utilisateur de déduire les fonctionnalités que doit présenter le produit (par exemple le mode rafale pour des photos d'événements sportifs). Enfin, les préférences de l'utilisateur peuvent permettre de déterminer les options du produit (par exemple la couleur de l'appareil photo). Ces données varient d'un processus de vente à l'autre et sont exploitées au cours de ce processus. Elles peuvent servir de base à des arguments et permettre de pousser des produits. Les données d'un utilisateur peuvent par exemple être les suivantes : *un client cherche un appareil photo numérique facile à utiliser pour l'offrir à sa grand-mère qui veut faire des photos de famille.*

- les **données acheteur** permettent de définir une stratégie de vente adéquate à celui-ci et à ses propres critères. Ce sont notamment :
  - les dispositions de l'acheteur (temps, délai de livraison, etc.) ;
  - les critères de choix de l'acheteur (prix, marque, avis, etc.) et les priorités données à ces critères ;
  - le niveau d'expertise de l'acheteur.

Par exemple, un acheteur expert en photo, habitué du site, qui dispose d'un budget de 150 euros et souhaite recevoir son produit dans les 3 jours. Les informations sur l'acheteur permettent de déterminer les produits correspondant aux souhaits de l'acheteur via des **règles métiers de la vente** mais aussi et surtout d'adapter le mécanisme et les arguments de vente (par exemple, à l'expertise de l'acheteur). Les dispositions et critères de l'acheteur permettent de définir les produits acceptables par l'acheteur et les arguments qui seront déterminants pour le convaincre.

Les **données du produit** sont de deux types : endogènes et exogènes. Les données endogènes constituent une fiche du produit et le définissent objectivement. Ce sont en particulier les informations techniques (type, zoom, résolution, etc.) et financières (prix, coût de livraison, etc.). Par exemple, l'appareil photo *LX5* est de type compact, est doté d'un zoom moyen et permet de prendre des photos entre 8 et 12 Mégapixels. Les données exogènes le mettent en perspective au moyen d'éléments extérieurs. Il s'agit notamment de l'image (marque, réputation, positionnement, etc.), de données statistiques (corrélations, labels, proximité produit/panier) et de données communautaires (avis, notations, réseaux sociaux). Par exemple : Le *LX5* a une moyenne de 4/5 sur 18 avis, c'est un produit de la marque *Panasonic*. Les données statistiques et communautaires peuvent constituer un moyen de déterminer un produit susceptible de convenir au client. De la même manière, la marque peut être un facteur déterminant pour le client dans l'achat d'un produit.

Les **données de l'enseigne** vont permettre de définir les buts du vendeur, elles regroupent :

- les objectifs de vente (produits à privilégier, règles selon les canaux) ;
- les canaux de vente (internet, boutique, téléphone, revendeurs, etc.) ;
- les données logistiques (disponibilité, délais de livraison) .

Par exemple, l'entreprise *Becquet* base son activité essentiellement sur la vente en ligne et par correspondance tandis que *Décathlon* dispose d'un grand nombre de magasins. Les informations relatives à l'enseigne permettent de déterminer la politique de vente. Elles sont communes à l'ensemble des processus de vente et interviennent dans la conception du mécanisme de vente de l'enseigne en lui-même.



Les **données du contexte** de la vente décrivent à la fois la situation dans laquelle se fait la vente et l'ensemble des données extérieures susceptibles de l'influencer. Ce sont notamment :

- les données de la concurrence (comparateur en ligne);
- le type de vente (unitaire ou liée);
- le moment de la vente (soldes, fêtes, ...);
- le canal de la vente (magasin/Internet/téléphone).

Le contexte peut être exploité dans le mécanisme de vente, notamment pour jouer sur un effet d'aubaine (par exemple lors de période de soldes).

## A.2 Données du VVU pour le domaine des APN

```
<query>
  <question>
    <nlQuestion>Vous preferez en priorite un modele ?</nlQuestion>
    <tag>Acheteur</tag>
    <attributName>profile</attributName>
  </question>
  <answers>
    <answer>
      <nlAnswer>tres solide , robuste</nlAnswer>
      <attributValue>securite</attributValue>
    </answer>
    <answer>
      <nlAnswer>le plus performant</nlAnswer>
      <attributValue>orgueil</attributValue>
    </answer>
    <answer>
      <nlAnswer>a la pointe de l innovation</nlAnswer>
      <attributValue>nouveaute</attributValue>
    </answer>
    <answer>
      <nlAnswer>le plus simple et pratique, fonctionnel</nlAnswer>
      <attributValue>confort</attributValue>
    </answer>
    <answer>
      <nlAnswer>abordable , meilleur rapport qualite/prix</nlAnswer>
      <attributValue>argent</attributValue>
    </answer>
    <answer>
      <nlAnswer>convivial , beau</nlAnswer>
      <attributValue>sympathie</attributValue>
    </answer>
    <answer>
      <nlAnswer>je ne sais pas</nlAnswer>
      <attributValue>>null</attributValue>
    </answer>
  </answers>
</query>
```

FIGURE A.1 : Question et réponses possibles pour la classe « acheteur »







# Table des figures

---

1	Organisation du manuscrit . . . . .	5
1.1	Graphe de préférence correspondant à la relation $\succsim$ d'un agent . . . . .	17
2.1	Vue analytique des dialogues . . . . .	33
2.2	Diagramme d'interaction du protocole de proposition FIPA . . . . .	34
2.3	Automate à états fini du protocole de proposition FIPA . . . . .	35
4.1	Deux relations de préférence : $\succsim_1$ (à gauche) et $\succsim_2$ (à droite). . . . .	46
5.1	Protocole de concession monotone entre deux 2 agents $a$ et $b$ représenté sous la forme d'un automate fini déterministe. . . . .	62
5.2	Évaluation du facteur de risque pour l'agent $i$ à l'instant $t$ . . . . .	63
7.1	Diagramme d'interaction entre l'ADP et le décideur . . . . .	81
7.2	Automate fini déterministe décrivant le protocole général d'un dialogue d'assistance à la décision . . . . .	82
8.1	Conversation entre le vendeur virtuel ubiquitaire et un client internaute . . . . .	94
8.2	Échelle de Likert pour le poids des critères . . . . .	95
8.3	Diagramme d'état-transition représentant le comportement interne du VVU . . . . .	105
8.4	Interface web du prototype du VVU. . . . .	106
9.1	Relations de préférence $\succsim_1$ (à gauche), $\succsim_2$ (au milieu) et $\succsim_3$ (à droite) . . . . .	112
9.2	Notre protocole de négociation sous la forme d'un automate à états fini. . . . .	119
10.1	Pourcentage des exécutions où $(\sigma_\Omega(\mathcal{X}, C_{\Omega-i}) \succ_i \sigma_\Omega(\mathcal{X}, C_\Omega))$ . . . . .	133
10.2	Pourcentage des exécutions où $(\sigma_\Omega(\mathcal{X}, T_\Omega) \succ_i \sigma_\Omega(\mathcal{X}, T_{\Omega-i}))$ . . . . .	133
10.3	Pourcentage des exécutions où $\sigma_\Omega(\mathcal{X}, C_{\Omega-i})$ sont des accords équitables . . . . .	135
10.4	Pourcentage des exécutions où $\sigma_\Omega(\mathcal{X}, T_\Omega)$ sont des accords équitables . . . . .	135
10.5	Cardinalités de l'ensemble des non-dominés pour l'agent 1 et de l'ensemble d'accord quand 100 agents coopèrent $(\sigma_\Omega(\mathcal{X}, C_\Omega))$ en fonction des taux d'équivalence et d'incomparabilité. . . . .	136

11.1	Un labyrinthe de taille $3 \times 3$ dans lequel Alice est située dans la cellule $c_1$ et Bob dans la cellule $c_9$ . . . . .	140
11.2	Premier pas de simulation : le labyrinthe $\mathcal{L}$ au milieu et ses représentations par les agents $\mathcal{L}_{\ominus}$ à gauche et $\mathcal{L}_{\oplus}$ à droite. . . . .	142
11.3	Deuxième pas de simulation : le labyrinthe $\mathcal{L}$ au milieu et ses représentations par les agents $\mathcal{L}_{\ominus}$ à gauche et $\mathcal{L}_{\oplus}$ à droite. . . . .	143
11.4	Troisième pas de simulation : le labyrinthe $\mathcal{L}$ au milieu et ses représentations par les agents $\mathcal{L}_{\ominus}$ à gauche et $\mathcal{L}_{\oplus}$ à droite. . . . .	144
11.5	Capture d'écran du démonstrateur. Parmi les six panneaux, celui au centre en haut permet d'observer la progression des agents, celui en-dessous révèle les offres échangées, les quatre autres représentent la perception locale de chaque agent. . .	145
11.6	De gauche à droite et de haut en bas, tours n° 1 à 4 de la négociation d'un point de rencontre entre les agents $A$ , $B$ , $C$ et $E$ . En rose, les cellules proposées par un seul agent, en cyan par deux, en bleu foncé par trois et en violet par quatre. Ces dernières constituent l'ensemble d'accords. . . . .	146
11.7	Panneau de configuration du démonstrateur . . . . .	147
11.8	Nombre maximum de déplacements effectués par un agent (à gauche) et nombre total de déplacements effectué par les deux agents (à droite) avec un labyrinthe généré par une version pseudo-aléatoire de l'algorithme de Prim. . . . .	148
11.9	Nombre maximum de déplacements effectués par un agent (à gauche) et nombre total de déplacements effectué par les deux agents (à droite) avec un labyrinthe généré par un algorithme de recherche en profondeur d'abord. . . . .	148
11.10	Nombre de négociation en fonction de la taille du labyrinthe . . . . .	150
11.11	Nombre total de tours de négociation en fonction de la taille du labyrinthe . . .	150
11.12	Nombre d'alternatives proposées en fonction de la taille du labyrinthe . . . . .	150
11.13	Nombre maximum de pas de simulations nécessaires à la rencontre en fonction de la densité du labyrinthe. . . . .	151
11.14	Nombre maximum de propositions en fonction de la densité du labyrinthe. . . . .	151
A.1	Question et réponses possibles pour la classe « acheteur » . . . . .	159

# Liste des tableaux

---

1.1	Propriétés des relations de comparaison . . . . .	17
1.2	Fonction d'utilité $u$ définie sur les alternatives $a, b, c, d$ . . . . .	19
1.3	Sémantiques possibles pour les différents modèles de préférence . . . . .	25
1.4	Propriétés des fonctions de choix selon leur modèle de préférence . . . . .	25
2.1	Illustration des différents types d'actes de langage . . . . .	29
2.2	Paramètres des messages FIPA . . . . .	30
2.3	Classification des actes de communication de FIPA-ACL . . . . .	31
2.4	Description des types de dialogues d'après [Walton 2010] . . . . .	32
3.1	Grille d'analyse des objectifs poursuivis dans les différentes approches . . . . .	43
3.2	Grille d'analyse des différentes approches logicielles pour assister la décision . . . . .	43
4.1	Fonctions d'utilité des agents 1 et 2 . . . . .	50
4.2	Coût social des alternatives selon les fonctions d'agrégation avec en gras les optimaux. . . . .	53
4.3	Agrégateurs d'utilité et Pareto-optimalité. . . . .	54
5.1	Préférences d'une population de 7 agents sur l'ensemble d'alternatives $\{a, b, c, d\}$ . . . . .	57
5.2	Préférences d'une population de 3 agents sur l'ensemble d'alternatives $\{a, b, c, d\}$ . . . . .	58
5.3	Propriétés des systèmes de vote . . . . .	60
5.4	Propriétés du protocole de concession monotone . . . . .	65
5.5	Caractéristiques générales des systèmes de vote et de négociation . . . . .	65
8.1	Rôle des données dans le mécanisme de vente . . . . .	93
8.2	Représentation des caractéristiques et des valeurs correspondantes pour l'univers APN . . . . .	94
8.3	Représentation des produits . . . . .	95
8.4	Représentation des vecteurs de référence . . . . .	95
8.5	Représentation d'un vecteur courant, la valeur associée à « hybride » est « inconnu » . . . . .	96
8.6	Exemple de calcul du vecteur courant . . . . .	97

---

8.7	Exemple de calcul de gain . . . . .	100
8.8	Exemple de calcul de satisfaction des produits pour un vecteur courant . . . . .	103
8.9	Exemple de calcul de dissimilarité . . . . .	105
9.1	Rangs des alternatives pour chaque agent. . . . .	114
9.2	Rangs moyens des alternatives . . . . .	115
9.3	Rangs maximaux des alternatives . . . . .	115
9.4	Vecteur de rangs des alternatives. . . . .	117
9.5	Trace de négociation où les trois agents appliquent la stratégie coopérative . . . . .	122
9.6	Trace de négociation où les deux premiers agents appliquent la stratégie coopérative et le troisième la stratégie temporisatrice . . . . .	125
11.1	Trace de négociation où Alice et Bob coopèrent pour se rejoindre au plus tôt . . . . .	143
A.1	Les appareils photo numériques et leurs caractéristiques. Par exemple, le LX5 est un compact avec une résolution entre 8 et 12 MP et un zoom moyen. . . . .	160
A.2	Les profils stéréotypes correspondent à des clients typiques. Par exemple, un client prudent (safety) préfère un compact ou un bridge plutôt qu'un hybride ou un SLR et la résolution n'est pas pertinente pour lui. . . . .	161



# Index

---

- accord, 120, 121
- acte de communication, 29
- acte de langage, 28
- adaptabilité, 79
- ADP, 77
- agent, 2
- agent conversationnel, 39
- agrégation des préférences, 45
- aide à la décision, 37
- alternative, 1, 11
- architecture d'agent, 84
- argumentation, 102
- argumenter, 80
- Arrow, 59
- attribut, 20
- automate, 34
  
- bienveillant, 55
- Borda, 58
- but, 77, 86
  
- choix maximal, 19
- choix social, 55
- communication, 27
- compétition, 1
- comportement, 105
- concept, 20
- Condorcet, 57
- consensus, 46
- coopération, 1
- coopérer, 122
- CP-net, 22
- critère, 94
  
- décision, 11
- décision multi-attribut, 20
- décision multi-critères, 19
- délibération, 32
- demande d'information, 32
  
- diagramme d'interaction, 34
- dialogue, 27, 31, 80
  
- e-commerce, 92
- effort, 113
- égoïste, 55
- émetteur, 27
- enquête, 32
- équité, 116
- expertise, 79
  
- faits, 84
- fonction d'agrégation linéaire, 23
- fonction d'utilité, 18
- fonction de choix, 12
  
- gain informationnel, 98
- graphe de préférences, 16
  
- incomparabilité, 16
- indépendance, 59
- indifférence, 14
- initiative, 78
- instance, 21
- interaction, 27
  
- langage, 28, 29
- leximax, 116
- leximin, 51
  
- majorité, 56
- modélisation des préférences, 11
  
- négociation, 32, 55, 60, 119
- négociation compétitive, 55
- négociation coopérative, 56
- ensemble des non dominés, 18
- non-dictature, 59
  
- objet de négociation, 61

optimalité, 17

Pareto, 47

Pareto-optimalité, 48

persuasion, 32

préférence large, 15

préférence stricte, 14

préférences cardinales, 18

préférences de Pareto nécessaires, 48

préférences de Pareto potentielles, 49

préférences lexicographiques, 22

préférences ordinales, 14

préférences unanimes, 21

pré-ordre, 15

prise de décision assistée, 1

prise de décision collective, 1, 55

profil de stratégie, 121

proposition, 120

propriété  $\alpha$ , 13

propriété  $\beta$ , 13, 16

protocole, 61, 119

rang, 113

rationalité, 13

récepteur, 27

règles métiers, 84

relation d'importance, 22

relation partielle, 16

relation totale, 15

résolution distribuée, 55

sémantique, 30

SMA, 2

stratégie, 87, 97, 121

syntaxe, 29

système de recommandation, 42

système expert, 39

tactique, 87

temporiser, 123

transitivité, 59

unanimité, 59

universalité, 59

vecteur de référence, 95

vote, 56

VVU, 91

# Bibliographie

---

- [Aknine *et al.* 2004] Samir Aknine, Suzanne Pinson et Melvin F. Shakun. *An Extended Multi-Agent Negotiation Protocol*. Autonomous Agents and Multi-Agent Systems, vol. 8, no. 1, pages 5–45, 2004. (Cité en page 61.)
- [Alpern & Gal 2003] Steve Alpern et Shmuel Gal. The theory of search games and rendezvous, volume 55. Springer, 2003. (Cité en page 140.)
- [Armstrong 1950] Wallace Edwin Armstrong. *A note on the theory of consumer's behaviour*. Oxford Economic Papers, vol. 2, no. 1, pages 119–122, 1950. (Cité en page 15.)
- [Arrow 1951] Kenneth Joseph Arrow. Social choice and individual values. Yale University Press, 1951. (Cité en page 59.)
- [Austin 1962] John Langshaw Austin. How to do things with words. Harvard Press University, 1962. (Cité en page 28.)
- [Aydoğan *et al.* 2013] Reyhan Aydoğan, Tim Baarslag, Koen V Hindriks, Catholijn M Jonker et Pinar Yolum. *Heuristic-based Approaches for CP-Nets in Negotiation*. In Complex Automated Negotiations : Theories, Models, and Software Competitions, pages 113–123. Springer, 2013. (Cité en page 127.)
- [Barberà *et al.* 2004] S. Barberà, W. Bossert et P. K. Pattanaik. Handbook of utility theory, chapitre Ranking sets of objects. Springer, 2004. (Cité en page 131.)
- [Barthès 2007] Jean-Paul Barthès. *Traitement de dialogues en langage libre par un agent assistant et son staff*. In Valérie Camps et Philippe Mathieu, editeurs, Actes des JFSMA'07 - Systèmes Multi-Agents - Modèles de comportements, pages 191–200. Cépaduès, 2007. (Cité en page 41.)
- [Bauer *et al.* 2001] Bernhard Bauer, Jörg P Müller et James Odell. *Agent UML : A formalism for specifying multiagent software systems*. International Journal of Software Engineering and Knowledge Engineering, vol. 11, no. 03, pages 207–230, 2001. (Cité en page 34.)
- [Ben Mimoun & Poncin 2010] M.S. Ben Mimoun et I. Poncin. *Agents virtuels vendeurs : que veulent les consommateurs ?* In WACA'10 : 4ème Workshop sur les Agents Conversationnels Animés, pages 99–106, 2010. (Cité en pages 44 et 68.)
- [Bentham 1789] Jeremy Bentham. An introduction to the principles of morals and legislation. T. Payne and Son, 1789. (Cité en page 50.)
- [Bolt 1980] Richard A Bolt. “put-that-there” : Voice and gesture at the graphics interface, volume 14. ACM, 1980. (Cité en page 40.)

- [Boutilier *et al.* 1999] Craig Boutilier, Ronen I Brafman, Holger H Hoos et David Poole. *Reasoning with conditional ceteris paribus preference statements*. In Proceedings of the Fifteenth conference on Uncertainty in Artificial Intelligence (UAI), pages 71–80. Morgan Kaufmann Publishers Inc., 1999. (Cit  en page 22.)
- [Boutilier *et al.* 2001] Craig Boutilier, Fahiem Bacchus et Ronen I Brafman. *UCP-networks : A directed graphical representation of conditional utilities*. In Proceedings of the Seventeenth conference on Uncertainty in Artificial Intelligence (UAI), pages 56–64. Morgan Kaufmann Publishers Inc., 2001. (Cit  en page 24.)
- [Bouveret *et al.* 2010] Sylvain Bouveret, Ulle Endriss et J r me Lang. *Fair Division under Ordinal Preferences : Computing Envy-Free Allocations of Indivisible Goods*. In Proceedings of the 17th European Conference on Artificial Intelligence (ECAI), pages 387–392, 2010. (Cit  en page 48.)
- [Bouyssou *et al.* 2009] Denis Bouyssou, Thierry Marchant et Patrice Perny. *Social choice theory and multicriteria decision aiding*. Decision-making Process : Concepts and Methods, pages 779–810, 2009. (Cit  en page 21.)
- [Braffort *et al.* 2010] Annelies Braffort, Jean-Paul Sansonnet, Jean-Claude Martin et Cyril Verrecchia. *DIVA, une architecture pour le support des agents gestuels interactifs sur internet*. TSI, vol. Vol. 29 (7), pages 777–806, 2010. (Cit  en page 41.)
- [Brams & Fishburn 1978] Steven J Brams et Peter C Fishburn. *Approval voting*. The American Political Science Review, pages 831–847, 1978. (Cit  en page 59.)
- [Bussmann & Muller 1992] Stefan Bussmann et Jorg Muller. *A negotiation framework for cooperating agents*. Proceedings of CKBS-SIG, pages 1–17, 1992. (Cit  en page 60.)
- [Caritat 1785] Marie Jean Antoine Nicolas (marquis de Condorcet) Caritat. *Essai sur l’application de l’analyse   la probabilit  des d cisions rendues   la pluralit  des voix*. Imprimerie Royale, Paris, 1785. (Cit  en pages 56, 58 et 69.)
- [Coombs 1964] Clyde H Coombs. *A Theory of Data*. 1964. (Cit  en page 59.)
- [Davis & Smith 1983] Randall Davis et Reid G Smith. *Negotiation as a metaphor for distributed problem solving*. Artificial intelligence, vol. 20, no. 1, pages 63–109, 1983. (Cit  en page 56.)
- [de Borda 1781] Jean Charles de Borda. *M moire sur les  lections au scrutin*. Histoire de l’Acad mie Royale des Sciences, 1781. (Cit  en pages 56, 58 et 69.)
- [Delecroix *et al.* 2011] Fabien Delecroix, Maxime Morge et Jean-Christophe Routier. *Vers un agent dialogique proactif*. In Actes des 19 me Journ es Francophones sur les Syst mes Multi-Agents (JFSMA), pages 203–212. Cepudes, 2011. (Cit  en pages 3 et 77.)
- [Delecroix *et al.* 2012a] Fabien Delecroix, Maxime Morge et Jean-Christophe Routier. *An algorithm for active learning of lexicographic preferences*. In Proceedings of Multiple Criteria Decision Aiding to Preference Learning (DA2PL), page 115–122, 2012. (Cit  en page 88.)

- [Delecroix *et al.* 2012b] Fabien Delecroix, Maxime Morge et Jean-Christophe Routier. *A Virtual Selling Agent which is Proactive and Adaptive*. In Proc. of of the 10th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 12), Advances in Intelligent and Soft-Computing, pages 57–66, Salamanca, April 2012. Springer. (Cité en page 3.)
- [Delecroix *et al.* 2013a] F. Delecroix, M. Morge et J.-C. Routier. *A Virtual Selling Agent which is Persuasive and Adaptive*. In Sascha Ossowski, editeur, Agreement Technologies, volume 8 of *Law, Governance and Technology Series*, pages 625–645. Springer Netherlands, 2013. (Cité en page 4.)
- [Delecroix *et al.* 2013b] Fabien Delecroix, Maxime Morge et Jean-Christophe Routier. *Négociation bilatérale pour la recherche d'un compromis*. In Salima Hassas et Maxime Morge, editeurs, Actes des 21èmes Journées Francophones sur les Systèmes Multi-Agents (JF-SMA2013), pages 19–28, Lille, 2013. Cépaduès. (Cité en page 4.)
- [Delecroix *et al.* 2014a] Fabien Delecroix, Maxime Morge et Jean-Christophe Routier. *Bilateral negotiation of a meeting point in a maze*. In Yves Demazeau, Franco Zambonelli, Juan M. Corchado et Javier Bajo, editeurs, Proc. of 12th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS), Salamanca, Spain., volume 8473 of *LNAI*, pages 86–97. Springer, 2014. (Cité en page 4.)
- [Delecroix *et al.* 2014b] Fabien Delecroix, Maxime Morge et Jean-Christophe Routier. *Réduire l'arbitraire par la négociation quitte à concéder*. *Revue d'Intelligence Artificielle*, vol. 28, no. 4, pages 433–462, 2014. (Cité en page 4.)
- [Devillers *et al.* 2005] Laurence Devillers, Laurence Vidrascu et Lori Lamel. *Challenges in real-life emotion annotation and machine learning based detection*. *Neural Networks*, vol. 18, no. 4, pages 407 – 422, 2005. (Cité en page 40.)
- [Dubuisson Duplessis 2014] Guillaume Dubuisson Duplessis. *Modèle de comportement communicatif conventionnel pour un agent en interaction avec des humains : Approche par jeux de dialogue*. Theses, INSA de Rouen, Mai 2014. (Cité en page 41.)
- [Dubus 2010] Jean-Philippe Dubus. *Prise de décision multiattribut avec le modèle GAI*. Thèse de doctorat, Université Pierre et Marie Curie (UPMC), 2010. (Cité en pages 14 et 19.)
- [Ferber & Perrot 1995] Jacques Ferber et Jean-François Perrot. *Les systèmes multi-agents : vers une intelligence collective*. InterEditions, 1995. (Cité en page 35.)
- [Figueira *et al.* 2005] José Figueira, Salvatore Greco et Matthias Ehrgott. *Multiple criteria decision analysis : state of the art surveys*, volume 78. Springer, 2005. (Cité en page 24.)
- [Finin *et al.* 1994] Tim Finin, Richard Fritzson, Don McKay et Robin McEntire. *KQML as an agent communication language*. In Proceedings of the Third International Conference on Information and Knowledge Management, pages 456–463. ACM, 1994. (Cité en page 29.)
- [FIPA 2002a] ACL FIPA. *FIPA acl message structure specification*. Foundation for Intelligent Physical Agents, <http://www.fipa.org/specs/fipa00061/>, 2002. (Cité en page 29.)

- [FIPA 2002b] TCC FIPA. *FIPA Communicative Act Library Specification*. Foundation for Intelligent Physical Agents <http://www.fipa.org/specs/fipa00037/>, 2002. (Cité en pages 29, 30 et 34.)
- [Fishburn 1970] Peter C Fishburn. *Utility theory for decision making*. Rapport technique, DTIC Document, 1970. (Cité en pages 18, 19, 23 et 24.)
- [Gilardone 2007] Muriel Gilardone. *Contexte, sens et portée de l'approche par les capacités de Amartya Kumar Sen : vers une économie normative post-welfariste*. Thèse de doctorat, Université Lumière Lyon-II, 2007. (Cité en page 54.)
- [Gonzales *et al.* 2007] Christophe Gonzales, Patrice Perny et Sergio Queiroz. *Réseaux GAI pour la prise de décision*. Revue d'Intelligence Artificielle (RIA), vol. 21, no. 4, pages 555–587, 2007. (Cité en page 24.)
- [Goodman & Markowitz 1952] Leo A Goodman et Harry Markowitz. *Social welfare functions based on individual rankings*. American Journal of Sociology, pages 257–262, 1952. (Cité en page 113.)
- [Halldén 1957] Sören Halldén. On the logic of "better". Cwk Gleerup, 1957. (Cité en page 14.)
- [Hansson & Grüne-Yanoff 2012] Sven Ove Hansson et Till Grüne-Yanoff. *Preferences*. In Edward N. Zalta, éditeur, The Stanford Encyclopedia of Philosophy. 2012. (Cité en pages 12 et 13.)
- [Hare 1873] Thomas Hare. The election of representatives, parliamentary and municipal : A treatise. Longmans, Green, Reader, and Dyer, 1873. (Cité en page 59.)
- [Harsanyi & Selten 1972] John C. Harsanyi et Reinhard Selten. *A generalized Nash solution for two-person bargaining games with incomplete information*. Management Science, vol. 18, no. 5-part-2, pages 80–106, 1972. (Cité en page 52.)
- [Huget & Koning 2001] Marc-Philippe Huget et Jean-Luc Koning. *Une ingénierie des protocoles d'interaction pour les systèmes multi-agents*. Thèse de doctorat, ANRT, 2001. (Cité en pages 33 et 35.)
- [Jennings *et al.* 2001] N. R. Jennings, P. Faratin, A. R. Lomuscio, S. Parsons, C. Sierra et M. Wooldridge. *Automated Negotiation : Prospects Methods and Challenges*. Group Decision and Negotiation, vol. 10, no. 2, pages 199–215, 2001. (Cité en pages 60 et 64.)
- [Keeney & Raiffa 1976] Ralph L. Keeney et Howard Raiffa. *Decisions with multiple objectives : preferences and value trade-offs*. New York : Wiley, 1976. (Cité en page 23.)
- [Kersten 2001] Gregory E. Kersten. *Modeling Distributive and Integrative Negotiations. Review and Revised Characterization*. Group Decision and Negotiation, vol. 10, no. 6, pages 493–514, 2001. (Cité en page 61.)
- [Kraus 2001] Sarit Kraus. *Strategic negotiation in multiagent environments*. MIT Press, 2001. (Cité en page 62.)

- [Kubera *et al.* 2011] Yoann Kubera, Philippe Mathieu et Sébastien Picault. *IODA : An interaction-oriented approach for Multi-Agent Based Simulations*. Journal of Autonomous Agents and Multi-Agent Systems (JAAMAS), vol. 23, no. 3, pages 303–343, November 2011. (Cité en page 27.)
- [Lester *et al.* 1997] James C. Lester, Sharolyn A. Converse, Susan E. Kahler, S. Todd Barlow, Brian A. Stone et Ravinder S. Bhogal. *The persona effect : affective impact of animated pedagogical agents*. In Proceedings of the SIGCHI conference on Human factors in computing systems, pages 359–366. ACM, 1997. (Cité en page 40.)
- [Li *et al.* 2010] Minyi Li, Quoc Bao Vo et Ryszard Kowalczyk. *An Efficient Procedure for Collective Decision-making with CP-nets*. In Proceedings of the 2010 conference on ECAI 2010 : 19th European Conference on Artificial Intelligence, pages 375–380. IOS Press, 2010. (Cité en page 113.)
- [Luce 1956] R Duncan Luce. *Semiorders and a theory of utility discrimination*. Econometrica, Journal of the Econometric Society, pages 178–191, 1956. (Cité en page 15.)
- [Maes *et al.* 1994] Pattie Maes *et al.* *Agents that reduce work and information overload*. Communications of the ACM, vol. 37, no. 7, pages 30–40, 1994. (Cité en page 39.)
- [Mathieu *et al.* 2012] Philippe Mathieu, David Panzoli et Sébastien Picault. *Virtual Customers in an Agent World*. In Proceedings of the 10th International conference on Practical Applications of Agents and Multi-Agents Systems (PAAMS'2012), Practical Advances in Intelligent and soft computing, pages 147–152. Springer, 2012. (Cité en page 154.)
- [McBurney *et al.* 2007] Peter McBurney, David Hitchcock et Simon Parsons. *The eightfold way of deliberation dialogue*. International Journal of Intelligent Systems, vol. 22, no. 1, pages 95–132, 2007. (Cité en page 32.)
- [Mill 1863] John S. Mill. *Utilitarianism*. Oxford University Press, 1863. (Cité en pages 50 et 70.)
- [Montaner *et al.* 2003] Miquel Montaner, Beatriz López et Josep Lluís De La Rosa. *A taxonomy of recommender agents on the Internet*. Artificial Intelligence Review (AIR), vol. 19, no. 4, pages 285–330, 2003. (Cité en page 43.)
- [Morge *et al.* 2011] Maxime Morge, Sameh Abdel-Naby et Bruno Beaufils. *Towards a dialectical approach for conversational agents in selling situations*. In Proc of ArgMAS 2010, numéro 6614 de LNAI, pages 141–158. Springer, 2011. (Cité en page 77.)
- [Morge 2005] Maxime Morge. *Syst'eme dialectique multi-agents pour l'aide à la concertation*. Thèse de doctorat, Ecole Nationale Supérieure des Mines de Saint-Etienne, 2005. (Cité en page 35.)
- [Mori *et al.* 2012] Masahiro Mori, Karl F MacDorman et Norri Kageki. *The uncanny valley [from the field]*. Robotics & Automation Magazine, IEEE, vol. 19, no. 2, pages 98–100, 2012. (Cité en page 44.)
- [Moulin 2002] Hervé J. Moulin. *Fair division and collective welfare*. The MIT Press, Cambridge, Massachusetts, USA, 2002. (Cité en page 50.)



- [Nash 1950] John F. Nash. *Equilibrium points in n-person games*. Proceedings of the national academy of sciences, vol. 36, no. 1, pages 48–49, 1950. (Cité en page 64.)
- [Ouerdane 2009] Wassila Ouerdane. *Multiple Criteria Decision Aiding : a Dialectical Perspective*. Thèse de doctorat, University Paris-Dauphine, Paris, France, 2009. (Cité en page 38.)
- [Pareto 1897] Vilfredo Pareto. *Cours d'économie politique*. Rouge, Lausanne, 1897. (Cité en page 47.)
- [Pasquier 2005] Philippe Pasquier. *Aspects cognitifs des dialogues entre agents artificiels : l'approche par la cohérence cognitive*. Thèse de doctorat, Université Laval, Juin 2005. (Cité en page 31.)
- [Petri 1979] Carl Adam Petri. *Introduction to General Net Theory*. In Proceedings of the Advanced Course on General Net Theory of Processes and Systems : Net Theory and Applications, pages 1–19. Springer-Verlag, 1979. (Cité en page 35.)
- [Poggi et al. 2005] I. Poggi, C. Pelachaud, F. Rosis, V. Carofiglio et B. Carolis. *Greta. A Believable Embodied Conversational Agent*. In Oliviero Stock et Massimo Zancanaro, éditeurs, *Multimodal Intelligent Information Presentation*, volume 27 of *Text, Speech and Language Technology*, pages 3–25. Springer Netherlands, 2005. (Cité en pages 41 et 44.)
- [Ramakrishnan et al. 2001] Naren Ramakrishnan, Benjamin J. Keller, Batul J. Mirza, Ananth Y. Grama et George Karypis. *Privacy risks in recommender systems*. IEEE Internet Computing, vol. 5, no. 6, pages 54–62, 2001. (Cité en page 43.)
- [Rawls 1971] John Rawls. *A theory of justice*. Harvard University Press, 1971. (Cité en pages 51 et 70.)
- [Resnick & Varian 1997] Paul Resnick et Hal R Varian. *Recommender systems*. Communications of the ACM, vol. 40, no. 3, pages 56–58, 1997. (Cité en page 42.)
- [Ricci et al. 2011] Francesco Ricci, Lior Rokach, Bracha Shapira et Paul B Kantor. *Recommender Systems Handbook*. Recommender Systems Handbook, vol. 1, 2011. (Cité en page 42.)
- [Rivière & Pesty 2010] J. Rivière et S. Pesty. *Actes de langage et émotions : vers un langage de conversation multimodal*. In JFSMA'10 : Journées Francophones des Systèmes Multi-Agents, pages 107 – 116, 2010. (Cité en page 41.)
- [Rosenschein & Zlotkin 1994] J. S. Rosenschein et G. Zlotkin. *Rules of Encounter - Designing Conventions for Automated Negotiation among Computers*. MIT Press, 1994. (Cité en pages 56, 60, 63 et 64.)
- [Roy & Bouyssou 1993] Bernard Roy et Denis Bouyssou. *Aide multicritère à la décision : Méthodes et cas*. Economica, Paris, 1993. (Cité en page 38.)
- [Roy 1985] Bernard Roy. *Méthodologie multicritère d'aide à la décision*. Economica, 1985. (Cité en pages 37 et 38.)



- [Russell & Norvig 1995] Stuart Russell et Peter Norvig. *Artificial Intelligence : A modern approach*. Artificial Intelligence. Prentice-Hall, Englewood Cliffs, vol. 25, 1995. (Cité en pages 64 et 141.)
- [Russell 1997] Stuart J Russell. *Rationality and intelligence*. Artificial Intelligence, vol. 94, no. 1, pages 57–77, 1997. (Cité en page 13.)
- [Sandholm 1999] Tuomas W Sandholm. *Distributed rational decision making*. Multiagent systems : a modern approach to distributed artificial intelligence, pages 201–258, 1999. (Cité en page 64.)
- [Sansonnnet *et al.* 2013] Jean-Paul Sansonnnet, François Bouchet et Nicolas Sabouret. *Moteurs de personnalité pour agents dialogiques : étude d'un modèle pour les traits interpersonnels*. In JFSMA, pages 105–114, 2013. (Cité en page 41.)
- [Searle 1969] John R. Searle. *Speech acts : An essay in the philosophy of language*. Cambridge University Press, Cambridge, 1969. (Cité en page 28.)
- [Sen 1970] Amartya K Sen. *Collective choice and social welfare*. North-Holland Publishing Co., 1970. (Cité en page 51.)
- [Sen 1976] Amartya Sen. *Welfare inequalities and Rawlsian axiomatics*. Theory and Decision, vol. 7, no. 4, pages 243–262, 1976. (Cité en page 51.)
- [Sen 1977] Amartya Sen. *Social Choice Theory : A Re-Examination*. Econometrica, vol. 45, 1977. (Cité en pages 13, 45, 47 et 126.)
- [Shannon 1948] C. E. Shannon. *A mathematical theory of communication*. Bell system technical journal, vol. 27, 1948. (Cité en page 28.)
- [Sheng *et al.* 2008] Hong Sheng, Fiona Fui-Hoon Nah et Keng Siau. *An experimental study on ubiquitous commerce adoption : Impact of personalization and privacy concerns*. Journal of the Association for Information Systems, vol. 9, no. 6, page 15, 2008. (Cité en page 44.)
- [Shortliffe *et al.* 1975] Edward H Shortliffe, Randall Davis, Stanton G Axline, Bruce G Buchanan, C Cordell Green et Stanley N Cohen. *Computer-based consultations in clinical therapeutics : explanation and rule acquisition capabilities of the MYCIN system*. Computers and biomedical research, vol. 8, no. 4, pages 303–320, 1975. (Cité en page 39.)
- [Tsoukiàs 2008] Alexis Tsoukiàs. *From decision theory to decision aiding methodology*. European Journal of Operational Research, vol. 187, no. 1, pages 138–161, 2008. (Cité en page 38.)
- [Turing 1950] Alan Turing. *Computing machinery and intelligence*. Mind, vol. 59, pages 433–460, 1950. (Cité en pages 28 et 68.)
- [Tversky 1969] A. Tversky. *Intransitivity of preferences*. Psychological Review, vol. 76, pages 31–48, 1969. (Cité en page 15.)
- [Verrons 2004] Marie-Hélène Verrons. *GeNCA : un modèle général de négociation de contrats entre agents*. PhD thesis, Université Lille I, 2004. (Cité en page 156.)

- [Von Neumann & Morgenstern 1944] John Von Neumann et Oskar Morgenstern. *Theory of games and economic behavior*. Princeton University Press, 1944. (Cité en page 18.)
- [Von Wright 1963] Georg Henrik Von Wright. *The logic of preference : an essay*. University Press, 1963. (Cité en page 14.)
- [Walras 1874] Léon Walras. *Eléments d'économie pure*. Guillaumin et Cie, page 34, 1874. (Cité en page 19.)
- [Walton & Krabbe 1995] D. Walton et E. Krabbe. *Commitment in dialogue*. SUNY Press, 1995. (Cité en page 31.)
- [Walton 2010] Douglas Walton. *Types of Dialogue and Burdens of Proof*. In *Proceedings of the 2010 Conference on Computational Models of Argument : Proceedings of COMMA 2010*, pages 13–24, Amsterdam, The Netherlands, The Netherlands, 2010. IOS Press. (Cité en pages 32 et 165.)
- [Weizenbaum 1966] J. Weizenbaum. *ELIZA - A computer program for the study of natural language communication between man and machine*. *Communications of the ACM*, vol. Vol. 9, pages 36–45, 1966. (Cité en page 39.)
- [Wooldridge 2009] Michael Wooldridge. *An introduction to multiagent systems*. John Wiley & Sons, 2009. (Cité en page 56.)
- [Yager 1988] R.R. Yager. *On ordered weighted averaging aggregation operators in multicriteria decisionmaking*. *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 18, no. 1, pages 183–190, Jan 1988. (Cité en page 24.)
- [Zeuthen 1930] Frederik Zeuthen. *Problems of monopoly and economic welfare*. G. Routledge and Sons, London, 1930. (Cité en page 63.)

## **Dialoguer pour décider : recommandation experte proactive et prise de décision multi-agents équitable**

**Résumé :** Si la prise de décision peut être purement individuelle, elle peut aussi impliquer plusieurs acteurs et revêtir des aspects sociaux. Dans cette thèse, je considère deux types de processus décisionnels sociaux : la prise de décision assistée et la prise de décision collective. Dans le cas de la prise de décision assistée, deux acteurs ont des rôles distincts : décideur et assistant. Ici, le décideur est un agent humain et l'assistant un agent logiciel. Dans de nombreuses applications, les capacités dialogiques de l'assistant sont décevantes et le dialogue manque de cohérence. Pour y remédier, nous avons conçu un agent dialogique proactif visant la crédibilité conversationnelle et la pertinence des recommandations : l'agent dirige la conversation en posant les questions adéquates pour acquérir les préférences du décideur afin de lui recommander les alternatives les plus pertinentes. Notre proposition trouve une de ses applications dans le champ du e-commerce. La deuxième contribution concerne la prise de décision collective. L'objectif visé est d'établir un processus permettant d'aboutir à un accord compatible avec des préférences incomplètes, qui engage l'ensemble des participants et présente un caractère équitable. Pour ce faire, je définis les accords équitables en appliquant le critère du leximax sur le rang des alternatives. Je propose ensuite un protocole de négociation permettant d'atteindre de tels accords et l'évalue au regard de la stratégie employée par les agents participant. Enfin, le protocole est appliqué à la recherche d'un point de rencontre dans un labyrinthe.

**Mots clés :** **Décision, Choix Social, Aide à la Décision, Système Multi-Agents, Dialogue, Proactivité, Négociation**

## **From dialogue to decision : proactive expert recommendation and fair agent-based negotiation**

**Abstract :** If decision making can be a pure individual process, it can involve several actors and present social aspects. In this thesis, I consider two types of social decision process : supported decision making and collective decision making. Concerning supported decision making, two actors have distinct roles : the decision maker and the assistant. Here, the decision maker is a human agent and the assistant a software one. In many applications, the dialogical abilities of the assistant are deceptive and the dialogue lacks of consistency. To tackle this problem, we design a proactive dialogical agent aiming for the credibility in conversation and the relevance of recommendations : our agent leads the conversation in asking relevant questions to collect the preferences of the decision maker and use them in recommending the alternatives that fit the most. We apply our approach on the e-commerce field. The second contribution concerns collective decision. The objective is to define a process that lead to a fair agreement, even if participants have incomplete preferences. For this purpose, I define the fair agreements by applying the leximax criterion on the rank of alternatives. Then, I propose a negotiation protocol to reach such agreements and the strategy is taken into account to evaluate it. Finally, the protocol is applied to the search of a meeting point in a maze.

**Keywords :** **Decision, Social Choice, Decision Aiding, Multi-Agent System, Dialogue, Proactivity, Negotiation**