



HAL
open science

Contributions à la négociation multi-agents

Maxime Morge

► **To cite this version:**

Maxime Morge. Contributions à la négociation multi-agents. Système multi-agents [cs.MA]. Université de Lille, 2019. tel-02447724

HAL Id: tel-02447724

<https://theses.hal.science/tel-02447724v1>

Submitted on 21 Jan 2020

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

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



Contributions à la négociation multi-agents

Mémoire d'habilitation à diriger des recherches
de l'Université de Lille
Spécialité : Informatique
présenté par :

Maxime Morge
Centre de Recherche
en Informatique, Signal et Automatique de Lille

Soutenue le 13 septembre 2019
Composition du jury :

Salima HASSAS	Pr., Université Claude Bernard Lyon 1	Rapporteur
Nicolas MAUDET	Pr., Université Paris Sorbonne	Rapporteur
Nicolas SABOURET	Pr., Université Paris Sud	Rapporteur
Clarisse DHAENENS	Pr., Université de Lille	Examinatrice
Amal EL FALLAH SEGHRUCHNI	Pr., Université Paris Sorbonne	Examinatrice
Philippe MATHIEU	Pr., Université de Lille	Garant

À Karine, Margaux, Matteo et Hugo.

Table des matières

Remerciements	ix
1 Introduction	1
1.1 Contexte scientifique	1
1.2 Problématique	2
1.3 Contributions	3
1.3.1 Négociation bilatérale	5
1.3.2 Négociation multi-latérale	6
1.3.3 Négociation multi-bilatérale	6
1.3.4 Négociations bilatérales concurrentes	7
1.3.5 Négociations un-à-plusieurs concurrentes	7
I Synthèse des travaux	9
2 État de l'art	11
2.1 Objectifs	12
2.2 Conception	12
2.2.1 Objet de négociation	12
2.2.2 Parties prenantes	13
2.2.3 Protocole d'interaction	14
2.2.4 Stratégie	14
2.3 Accord	16
2.3.1 Optimalité de Pareto	16
2.3.2 Bien-être social	17
2.4 Processus	17
2.5 Exemple	18
2.6 Synthèse	18
2.7 Conclusion	19
3 Négociation bilatérale	23
3.1 Introduction	23
3.2 Scénario d'approvisionnement	25
3.3 Architecture d'agent	27
3.3.1 Prise de décision individuelle	29
3.3.2 Prise de décision sociale	32

3.3.3	Interaction sociale	34
3.4	Scénario nominal	36
3.5	Conclusion	37
4	Négociation multi-latérale	39
4.1	Introduction	39
4.2	Modélisation des préférences	40
4.3	Accord équitable	41
4.4	Négociation	42
4.5	Évaluation de la négociation	44
4.6	Scénario	44
4.7	Résolution	45
4.8	Implémentation	46
4.9	Évaluation de la résolution	47
4.10	Conclusion	48
5	Négociation multi-bilatérale	49
5.1	Introduction	50
5.2	Problème du mariage stable	51
5.3	Évaluation sociale des appariements	52
5.3.1	Optimalité sociale	52
5.3.2	Bien-être social	52
5.4	Swing	53
5.5	Swing++	55
5.6	Travaux connexes	56
5.7	Évaluation	58
5.7.1	Évaluation de Swing	58
5.7.2	Évaluation de Swing++	58
5.7.3	Synthèse	59
5.8	Problème des hôpitaux/résidents	59
5.9	Adaptation de Swing	60
5.10	Application pratique	61
5.11	Expérimentations	63
5.12	Conclusion	64
II	Travaux en cours et perspectives	67
6	Négociations bilatérales concurrentes	69
6.1	Introduction	69
6.2	Affectation d'individus à des activités	70
6.3	Procédure centralisée d'appariement	73
6.4	Comportement d'agent	75
6.5	Évaluation empirique	76
6.6	Application pratique	78
6.7	Conclusion	78

7 Enchères multi-parties concurrentes	81
7.1 Introduction	81
7.2 MapReduce	82
7.3 Proposition	83
7.3.1 Processus de négociation	84
7.3.2 Multi-enchères	84
7.4 Expérimentations	85
7.5 Conclusion	86
8 Projet de recherche	89
8.1 Synthèse	89
8.1.1 Verrous scientifiques	89
8.1.2 Verrous technologiques	90
8.2 Perspectives	91
8.2.1 Contexte scientifique	91
8.2.2 Périmètre scientifique	92
8.2.3 Verrous scientifiques	94
8.2.4 Verrous technologiques	96
8.2.5 Résumé analytique	96
Publications choisies	97
Bibliographie	102
Index	110
III Pièces annexes	113
Curriculum vitæ synthétique	115
Curriculum vitæ	117
Résumé	125
Abstract	125

TABLE DES MATIÈRES

Remerciements

Je tiens tout d’abord à exprimer ma gratitude aux membres du jury. J’adresse mes plus vifs remerciements à Salima HASSAS, Nicolas MAUDET et Nicolas SABOURET qui m’ont fait l’honneur de rapporter ce mémoire, en dépit d’un emploi du temps chargé. Je suis également très reconnaissant à Clarisse DHAENENS et Amal EL FALLAH SEGHROUCHNI pour l’intérêt qu’elles ont manifesté à l’égard de ce travail en acceptant de participer au jury. Je tiens à remercier tout particulièrement Philippe MATHIEU envers lequel j’ai une dette considérable pour m’avoir accueilli au sein de son équipe, d’abord en 2006 en tant qu’attaché temporaire d’étude et de recherche, puis en 2009 en tant que maître de conférence. Je lui suis infiniment redevable de m’avoir fait partager sa passion de la recherche et son enthousiasme pour l’enseignement ainsi que pour son amitié indéfectible.

Le travail réflexif sous-jacent à la rédaction de ce mémoire m’a permis de mesurer à quel point la recherche est un processus collectif. Je voudrais donc ici exprimer ma gratitude à tous ceux qui y ont contribué directement ou indirectement, et d’avance solliciter la clémence de ceux, rares j’espère, qu’un injuste oubli aurait lésés.

Je suis reconnaissant à Olivier BOISSIER d’avoir suscité ma vocation pour le métier d’enseignant-chercheur et de m’avoir fait découvrir la communauté « système multi-agents », où je me suis épanoui au côté notamment de Gauthier PICARD et de Laurent VERCOUTER. Je tiens à exprimer ma profonde gratitude à Jean-Christophe ROUTIER pour son professionnalisme ainsi que pour la qualité et la richesse de nos échanges hebdomadaires depuis plus de dix ans. Je tiens également à remercier les collègues et amis que j’ai rencontrés lors mon séjour à l’*Università di Pisa* : Stefano BROMURI, Phan Minh DUNG, Paolo MANCARELLA, Jarred MCGINNIS, Kostas STATHIS et Francesca TONI.

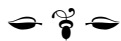
Mon intégration dans le Nord tient largement aux qualités humaines des membres, passés ou présents, de l’équipe SMAC, notamment Sameh ABDEL-NABY, Bruno BEAUFILS, Anne-Cécile CARON, Jean-Paul DELAHAYE, Patricia EVERAERE, Antoine NONGAILLARD, David PANZOLI, Sébastien PICAULT, Lisa ROUGETET et Yann SECQ. Mon attachement à cette région n’est pas étranger à la cohésion de l’équipe pédagogique du master SIAD : Gérard CARIN, Virginie DELSART, Nicolas VANEECLOO et Nadarajen VEERAPEN.

Une bonne part des travaux présentés dans ce mémoire n’auraient pas vu le jour sans l’enthousiasme d’étudiants et de doctorants motivés et dynamiques que j’ai eu la chance d’encadrer, notamment Quentin BAERT, Fabien DELECROIX, Eric PIETTE et Thomas NACHTERGAELE.

Je tiens aussi à saluer mes collègues, en particulier les personnels techniques et administratifs que cela soit à la Faculté des Sciences Économiques et Sociales ou au laboratoire CRISTAL sans lesquels aucune activité d’enseignement et de recherche n’est possible.

Merci enfin à ma famille de cœur pour leur soutien moral dans cette aventure humaine : Danielle BARRET, Giuliana BIAGIOLI, Eugene CLEUR, Joseph, Marie-Paule et Sophie MARTIN ainsi que Magali MOREL.

REMERCIEMENTS



CHAPITRE 1

Introduction

Sommaire

1.1	Contexte scientifique	1
1.2	Problématique	2
1.3	Contributions	3
1.3.1	Négociation bilatérale	5
1.3.2	Négociation multi-latérale	6
1.3.3	Négociation multi-bilatérale	6
1.3.4	Négociations bilatérales concurrentes	7
1.3.5	Négociations un-à-plusieurs concurrentes	7

Dans ce chapitre, je présente la problématique générale de mes travaux. Je situe en premier lieu mon travail de recherche dans le domaine des Systèmes Multi-Agents (SMA). Je pose ensuite la problématique qui est celle de la négociation. Finalement, j'expose chacune des contributions détaillées dans ce manuscrit.

1.1 Contexte scientifique

Mes travaux s'intègrent aux recherches menées dans l'équipe SMAC du laboratoire CRISTAL en Intelligence Artificielle Distribuée. Ils se situent dans le cadre des **Systèmes Multi-Agents** (FERBER, 1995; WEISS, 2013; WOOLDRIDGE, 2002).

Un SMA est composé de multiples entités autonomes en interaction, appelées agents. Un agent est une entité de calcul comme un programme informatique ou un robot qui exhibe un comportement. Un agent est situé dans un environnement, physique ou informatique. Il est capable de percevoir son environnement et d'agir sur celui-ci. Je considère ici des **agents cognitifs**, appelés également agents délibératifs. Ceux-ci sont munis d'une mémoire et de mécanismes de raisonnement manipulant une représentation explicite de leur but ou de leur environnement, qu'elle soit numérique (comme une fonction d'utilité) ou symbolique (comme c'est le cas avec un langage logique). Les agents sont autonomes, c'est-à-dire qu'ils agissent de manière à atteindre les objectifs qui leur ont été fixés lors de la conception.

En interagissant, les agents ne se contentent pas simplement d'échanger des données mais ils sont engagés dans des scénarios, qu'ils soient en compétition ou qu'ils coopèrent. Les influences mutuelles qu'ils exercent les uns sur les autres peuvent prendre de multiples formes : marqueurs chimiques, forces d'attraction ou de répulsion, etc. Je ne considère ici que les **modèles d'interaction**

directe. Ainsi, les interactions prennent la forme d'échange de messages, chacun ayant un expéditeur et un, voire plusieurs, destinataires. Pour cela, les agents utilisent un langage de communication muni d'une sémantique afin d'énoncer et d'interpréter ces messages. Ces communications ont pour but d'atteindre un accord, de prendre collectivement des décisions en négociant, en votant, en argumentant, en participant à des enchères ou en formant des coalitions.

Finalement, le SMA peut être considéré d'un point de vue macroscopique comme une entité collective dont l'action est le résultat de structures sociales explicites ou de schémas stables émergeant de l'activité conjointe des agents. On parle alors d'**organisation**. Lorsqu'elle est explicite, celle-ci est une spécification structurelle (i.e. les rôles joués par les agents, les groupes formés et leurs relations), fonctionnelle (i.e. les missions assignées aux agents) et déontique (leur droit et leur devoir). Dans ce manuscrit, je n'aborderai pas l'aspect organisationnel des SMA.

L'intelligence artificielle distribuée (RANDALL, 1980) est un domaine fertile de l'Informatique, de l'intelligence artificielle en particulier, qui s'est inspirée des Mathématiques (e.g. la logique), de l'Économie (e.g. la théorie des jeux) et de la Sociologie (e.g. la théorie de l'activité). Influencé par les langages d'acteur (HEWITT, 1977) et par les architectures de type tableau noir (HAYES-ROTH, 1985), ce domaine a beaucoup évolué, en particulier dans les années 1990 (SYCARA, 1998), mais l'objet d'étude est resté le même. Un système multi-agents est caractérisé par le fait que :

1. chaque agent a des informations incomplètes et des capacités limitées ;
2. il n'y a pas de contrôle global du système ;
3. les données sont décentralisées ;
4. les calculs sont réalisés de manière asynchrone ;

En résumé, les SMA représentent un paradigme de premier ordre pour l'analyse, la conception et l'implémentation de systèmes composés d'entités autonomes en interaction.

1.2 Problématique

Dans le domaine des SMA, la problématique de la gestion des conflits a deux finalités majeures : d'une part la conception de systèmes distribués ; d'autre part l'aide à la résolution des conflits entre humains. Quel que soit l'objectif poursuivi, les conflits constituent une caractéristique intrinsèque des SMA. Dans un SMA, deux causes principales sont à l'origine de ces conflits :

1. les agents ont des perceptions partielles de l'environnement, éventuellement contradictoires. Par exemple, un premier agent croit que « Tweety ne vole pas » alors qu'un second agent croit que « Tweety est un oiseau ». Ces deux croyances semblent contradictoires mais elles s'avèrent compatibles quand les agents apprennent que « Tweety est un pingouin » ;
2. les agents ont des objectifs individuels qui ne peuvent pas être pleinement satisfaits ni simultanément ni individuellement. Par exemple, un agent veut vendre un service de qualité médiocre au prix fort et un autre agent veut acheter ce même service avec une qualité élevée et à un prix bas. Pour atteindre leur objectif commun, i.e. conclure une affaire, les capacités individuelles des agents sont insuffisantes. Ils sont inter-dépendants.

Alors que le premier point concerne des **conflits épistémiques**, i.e. sur des croyances, le second point est relatif aux **conflits pratiques**, i.e. sur les actions.

La thèse de doctorat que j'ai soutenue aux Mines de Saint-Étienne s'attelait à la résolution des conflits épistémiques. J'y ai présenté un système multi-agents dialectique qui résout ce type de conflits (MORGE, 2005). J'ai proposé un modèle de raisonnement, des stratégies et un protocole qui permettent de prendre une décision en ce qui concerne la véracité d'un fait en prenant en compte la réputation des agents (MORGE, 2006). À mon arrivée à l'Université de Lille 1, ce système a été appliqué à la réconciliation d'ontologies hétérogènes (MORGE & ROUTIER, 2007).

Les travaux que j'ai menés depuis s'intéressent à la résolution des conflits pratiques. C'est une synthèse de ces travaux que je propose ici.

Même en compétition, des agents ayant des intérêts conflictuels, doivent collaborer pour atteindre leurs objectifs. À cette intention, ils peuvent négocier. Cette forme d'interaction permet d'aboutir à un accord mutuel. Ce dernier porte typiquement sur une allocation de ressources ou de tâches, un appariement entre agents ou une décision collective. Les intérêts des agents sont conflictuels dans la mesure où ils ne peuvent pas être pleinement satisfaits simultanément. Selon cette perspective, la négociation peut être vue comme la recherche distribuée dans un espace d'accords potentiels (JENNINGS et al., 2001).

Par négociation, on désigne le processus grâce auquel plusieurs parties prenantes aboutissent à une décision collective en verbalisant leurs demandes et convergent vers un accord par succession de concessions ou recherchent de nouvelles alternatives (PRUITT, 1981). De nombreux cadres de négociation proposés dans la littérature sont passés en revue dans le chapitre 2. C'est à la lumière de leurs caractéristiques, qu'ils portent sur l'environnement de négociation, sur l'accord ou sur le processus, que peuvent être comparés les différents travaux sur la négociation multi-agents.

1.3 Contributions

Ce manuscrit contribue à l'étude des modèles de négociation multi-agents. La question à laquelle répondent mes travaux est :

« Quels mécanismes opératoires mettre en œuvre pour la négociation ? »

Cette question est abordée non pas d'un point de vue Mathématique, c'est-à-dire en se focalisant sur l'existence de stratégies qui exhibent un certain nombre de propriétés, mais d'un point de vue Informatique. Dans une perspective algorithmique, je construis des modèles de raisonnement, des protocoles et des stratégies. Les cadres de négociation proposés sont systématiquement implémentés, évalués de manière expérimentale et appliqués soit via des scénarios jouets ou pour des applications pratiques s'appuyant sur des données réelles ou des problèmes industriels.

Le mécanisme principal que j'ai exploré lors de ma thèse de doctorat est l'**argumentation**. Dans ces précédents travaux, j'ai envisagé la dialectique, i.e. le processus de confrontation d'arguments et de contre-arguments pour la résolution des conflits épistémiques. L'argumentation permet notamment d'enrichir la base de connaissance de chacun des agents. L'ultime décision concernant la véracité d'un fait peut être alors établie par un agent tiers au regard des connaissances échangées et éventuellement en prenant en considération les niveaux d'expertise de chacun des agents si nécessaire pour trancher. C'est, par exemple, le mécanisme privilégié qu'un adulte utilise avec son enfant pour que ce dernier apprenne. L'asymétrie des rôles permet de trancher dans un ultime recours. Ce seul mécanisme s'avère inopérant quand le conflit est pratique et que les rôles des participants sont symétriques. L'échange d'information ne concerne alors pas simplement des connaissances, mais également des décisions, des buts et des préférences. À partir du moment où l'hypothèse des comparaisons interpersonnelles est exclue, il s'avère impossible pour un agent tiers de comparer les buts des agents et de les prioriser.

Conflit	épistémique	pratique
Situation	asymétrique	symétrique
Mécanisme	argumentation	concession

TABLE 1.1 – Mécanismes de résolution de conflit

Comme l'illustre le tableau 1.1, je défends ici la thèse selon laquelle le mécanisme opératoire nécessaire pour résoudre un conflit pratique dans une situation symétrique est la **concession**. Attendu

que les agents ont des objectifs individuels qui ne peuvent pas être pleinement satisfaits simultanément, chacun d'eux doit, au fur et à mesure de la négociation, abandonner une partie de ses objectifs pour satisfaire une partie des objectifs de ses interlocuteurs. C'est, par exemple, le mécanisme privilégié qu'un couple utilise pour prendre des décisions communes malgré la symétrie des rôles et l'antagonisme de leurs préférences. Ce mécanisme a été étudié, notamment à travers le protocole de concession monotone (ROSENSCHEIN & ZLOTKIN, 1994). Toutefois, ce protocole nécessite que chaque agent connaisse la fonction d'utilité de son interlocuteur. Dans une perspective SMA (cf section 1.1), les travaux que je présente ici font l'hypothèse que les données sont décentralisées et donc que les agents ne connaissent pas a priori les préférences des autres. La difficulté commune abordée par ces travaux réside dans le fait que l'information dont disposent les agents est imparfaite. Ainsi, selon la stratégie de concession, un agent débute la négociation en proposant l'alternative qui le satisfait le mieux. Si cette dernière est refusée, l'agent réduit pas-à-pas ses prétentions jusqu'à ce qu'une proposition soit retenue ou que sa valeur de réserve soit atteinte, i.e. le seuil au-dessus duquel l'agent est prêt à négocier. Le verrou scientifique auquel la mise en œuvre d'une stratégie de concession se confronte consiste à garantir la qualité de l'accord obtenu. Je souhaite non seulement **garantir le succès** du processus de négociation mais également m'assurer de son optimalité sociale. Ce dernier aspect se concrétise via la **Pareto-optimalité**, voire en réintroduisant et en maximisant le **bien-être utilitaire**. Nous verrons que, dans certains cadres de négociation présentés, ces critères s'avèrent insuffisants. En effet, je souhaite m'assurer que les concessions, c'est-à-dire les efforts réalisés par les agents pour atteindre un accord sont équitablement répartis. À cette intention, j'introduirai la notion de **bien-être équitable** (cf chapitre 5) et d'**accord équitable** (cf chapitre 4). On peut finalement remarquer que les deux mécanismes évoqués pour la résolution des conflits, l'argumentation et la concession, ne sont pas mutuellement exclusifs. Comme nous le verrons, l'argumentation permet de guider les concessions afin d'atteindre rapidement un accord.

L'ensemble des travaux passés et en cours que je présente dans ce manuscrit sont résumés dans la figure 1.1 à travers une grille d'analyse qui distingue la complexité du modèle de préférence (sur l'axe des abscisses) et l'arité du processus (sur l'axe des ordonnées). L'espace hachuré correspond au domaine de recherche que j'ai exploré. La flèche épaisse pointe la direction de recherche prise.

La partie I présente les travaux passés. Leur point commun réside dans le modèle de préférence adopté par les agents. Contrairement à la majorité des travaux existants, je ne représente pas les penchants d'un agent à l'aide d'une fonction d'utilité mais à l'aide d'un **modèle de préférence ordinal et qualitatif**. Cette hypothèse permet d'éviter les comparaisons interpersonnelles d'utilité qui peuvent être à l'origine de biais dans l'agrégation des préférences. En évitant ainsi des hypothèses restrictives, la modélisation des préférences est expressive et générique. Toutefois, ce modèle de préférence ne nous permet pas d'adopter la fonction de concession proposée dans (FARATIN, SIERRA & JENNINGS, 1998). Les modèles de préférence que j'utilise dans la partie I prennent successivement la forme d'un langage logique pour la représentation incomplète d'un problème de décision avec des buts priorités, des décisions et des croyances présumées, puis d'une relation de préférence complète (i.e. un ordre strict) et enfin d'une relation de préférence incomplète (i.e. un pré-ordre). L'expressivité de ces modèles et donc leur complexité est décroissante pour se focaliser sur l'arité du processus de négociation. J'explore tour à tour, le problème de la **négociation bilatérale**, de la **négociation multi-latérale** puis de la **négociation multi-bilatérale**. La complexité du cadre de négociation croit avec l'arité du processus.

La partie II présente les travaux en cours. Leur point commun réside dans le modèle de préférence adopté par les agents. Leurs penchants sont exprimés à l'aide d'une fonction d'utilité. Ce **modèle de préférence cardinal et quantitatif** dont l'expressivité est limitée permet d'explorer la complexité de la négociation liée à son arité. En particulier, les travaux en cours envisagent un processus de décision collective qui consiste en de **multiples négociations concurrentes**, qu'elles soient bilatérales ou qu'elles consistent en des enchères.

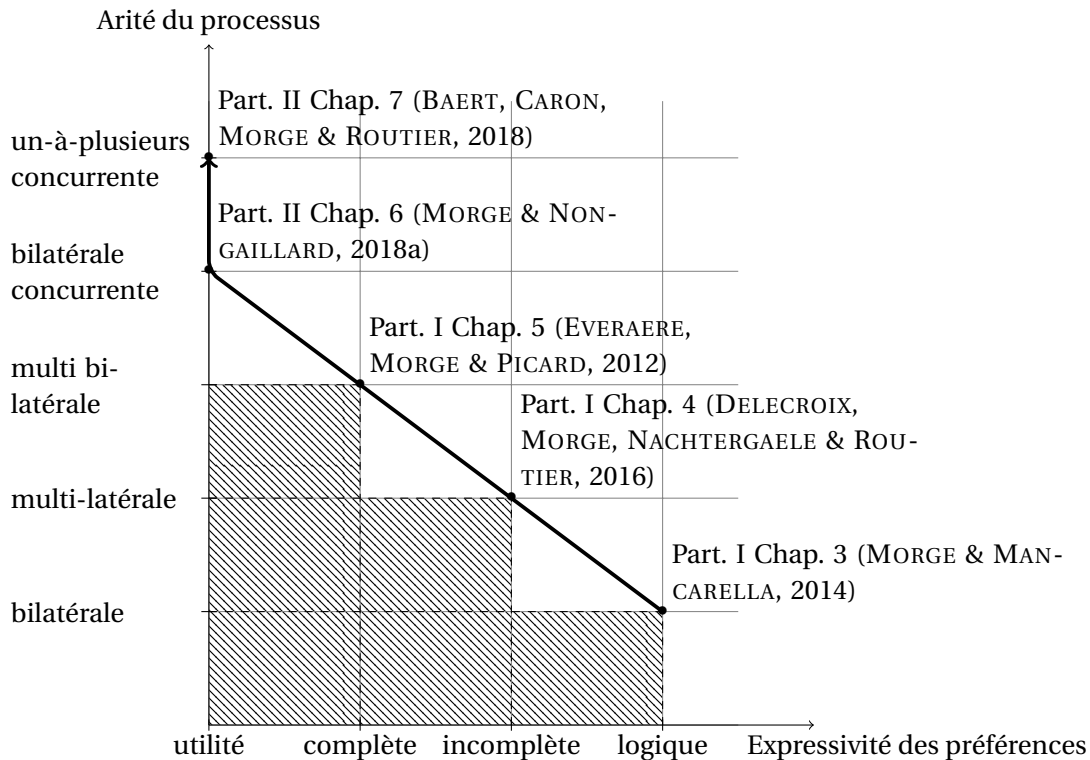


FIGURE 1.1 – Grille d'analyse des contributions passées et en cours

Les contributions représentées dans la figure 1.1 sont synthétisées dans chacune des sections suivantes.

1.3.1 Négociation bilatérale

Dans le chapitre 3, je considère la négociation bilatérale pour l'établissement d'un contrat sur un bien ou un service. L'objet de négociation est multi-attributs.

Je propose un modèle et une architecture d'agent pour la sélection automatique de services et de partenaires. Trois modules internes sont dédiés à la prise de décision, à la négociation, et à la communication, respectivement. Si le premier peut être considéré comme le module de raisonnement de l'agent, le second est un module de raisonnement social et le troisième un moteur des règles de dialogues encodées sous la forme de protocoles d'interaction. La terminaison et le succès des négociations sont garantis. De plus, l'accord atteint est systématiquement Pareto-optimal. Toutefois, il n'est pas dans l'intérêt individuel d'un agent de coopérer. Afin de tester notre approche, l'architecture a été instanciée dans le projet ArguGRID via des scénarios d'approvisionnement électronique.

Les modules de raisonnement utilisent un mécanisme argumentatif pour prendre des décisions en concédant. Ce dernier s'appuie sur un langage logique permettant de représenter des connaissances, des buts et des décisions. Des préférences sont attachées aux buts. J'adopte ici un modèle de préférences ordinales et qualitatives. De plus, le mécanisme de prise de décision modélise l'intuition selon laquelle les buts les plus prioritaires sont privilégiés alors que les buts les moins prioritaires peuvent être abandonnés. L'implémentation, intitulée MARGO, est un outil de prise de décision multi-attributs comme cela est requis pour la négociation d'un bien et d'un service.

1.3.2 Négociation multi-latérale

Dans le chapitre 4, la négociation porte sur une décision collective. Je propose un jeu de négociation multi-latérale où les agents échangent des offres mono-attributs.

Je considère un modèle de préférences ordinales et qualitatives. J'utilise des relations de préférences incomplètes, c.-à.d. des relations de pré-ordre pour modéliser les préférences individuelles des agents. De cette manière, les agents peuvent être indécis car ils ne sont pas toujours en mesure de comparer les alternatives en raison du manque d'information, d'une incertitude, ou à cause de préférences contradictoires.

D'une part, je raffine la règle de Pareto en considérant le leximax-optimal afin d'évaluer l'équité des accords. D'autre part, je propose un jeu de négociation multi-latérale, i.e. un protocole, un critère d'acceptabilité et des stratégies. Afin d'atteindre un accord équitable, le protocole ne priorise aucun des agents, il est symétrique. Je présente deux stratégies de concession qui consistent à proposer une ou plusieurs alternatives préférées et à réduire pas-à-pas ses prétentions si ces propositions ne sont pas retenues. Alors que la stratégie coopérative réalise autant de concessions que possible, la stratégie temporisatrice retarde un accord en ne soumettant les alternatives qu'une-à-une. La terminaison et le succès des négociations sont garantis. Je montre que l'accord obtenu entre des agents coopératifs est leximax-optimal et donc Pareto-optimal et que la stratégie temporisatrice sert l'intérêt individuel de l'agent. Toutefois, les expérimentations montrent que l'avantage qu'un agent peut tirer de la stratégie temporisatrice n'est pas systématique. En particulier, cette stratégie est relativement inefficace si tous les opposants temporisent.

Dans le chapitre 4, je propose une application pratique à ce jeu de négociation : la recherche d'un point de rendez-vous dans un labyrinthe. Dans ce scénario concret de résolution de problème distribué, deux ou plusieurs agents sont parachutés dans un labyrinthe qui leur est inconnu. Leur but est alors de se retrouver le plus rapidement possible. Ils communiquent donc pour fixer un point de rendez-vous. Les agents négocient en échangeant des offres, c'est-à-dire, des cellules potentielles où se rencontrer. La convergence de nos simulations est garantie. Les expérimentations montrent que la négociation ne réduit pas l'effort total des agents mais permet de le partager équitablement entre eux et donc de résoudre rapidement le problème distribué. J'étudie également l'effet du partage d'informations lors de ces résolutions.

1.3.3 Négociation multi-bilatérale

Dans le chapitre 5, la négociation porte sur un appariement entre agents. Je m'intéresse au problème bien connu des mariages stables – en anglais, *Stable Marriage Problem* (SM). L'enjeu est simple. En considérant deux communautés dont chaque membre a des préférences vis-à-vis des individus de l'autre communauté, l'objectif est de former des couples en prenant en compte leurs préférences.

L'algorithme séminal de GALE et SHAPLEY (1962) - dénoté GS - est une preuve constructive de l'existence d'une solution admissible. L'agentification de cet algorithme proposé par BRITO et MESEGUER (2005) consiste à considérer l'appariement obtenu comme le résultat émergent de multiples négociations bilatérales. Ainsi, on distingue deux types d'agent : une communauté de proposants et une communauté de disposants. Les négociations entre ces agents aboutissent à une solution stable mais inéquitable : la communauté des proposants est favorisée. Je propose ici la méthode Swing (EVE-RAERE, MORGE & PICARD, 2012) où les agents jouent alternativement ces deux rôles. Cette approche permet l'émergence d'un appariement stable qui ne peut être atteint avec l'algorithme GS. Malheureusement, Swing, peut ne pas se terminer pour certaines instances du problème. En effet, cette méthode peut mener à des situations de dilemme. Swing a été étendu pour : i) détecter les éventuels dilemmes; ii) et les résoudre. Comme Swing, Swing++ prône l'équité.

Dans ce travail, je fais l'hypothèse que les relations de préférence sont des ordres stricts. Les concessions des agents s'appuient sur ce modèle de préférence. Chacun s'adresse à l'agent qu'il pré-

fière. Si ce dernier refuse, l'agent concède en s'adressant au partenaire suivant dans sa liste de préférence. Pendant ce temps, les partenaires potentiels jouent le rôle de disposant en acceptant/refusant les propositions au regard de leur niveau de concession. Ainsi, le processus de négociation est constitué de multiples négociations bilatérales : les deux communautés se proposent alternativement. Chaque proposant envoie une demande à l'ensemble des individus acceptables. Swing++ est la seule méthode permettant d'obtenir systématiquement un appariement équitable (au sens du bien-être équitable) et optimal (au sens du bien-être utilitaire). Malheureusement, cette méthode n'est pas, comme GS ou Swing, distribuable.

Dans le chapitre 5, je propose une application pratique pour la méthode Swing : l'affectation de stages pour les Professeurs des écoles en formation dans l'Académie de Lille. Ainsi, les listes de préférences des agents prennent en compte les vœux des stagiaires, le diplôme des encadrants, la distance entre eux et les priorités des stagiaires (nombre d'enfants, voiture à disposition, etc.). Les expérimentations confirment que Swing est plus équitable. Ainsi, la méthode Swing permet de décroître le nombre d'encadrants et donc le coût de gestion.

1.3.4 Négociations bilatérales concurrentes

Dans le chapitre 6, je considère de multiples négociations bilatérales concurrentes pour la formation de coalition. L'objet de négociation est mono-attribut.

J'aborde dans chapitre 6 le problème générique où des individus doivent être affectés à des activités en privilégiant les activités qui leur plaisent avec les partenaires qu'ils apprécient. Les préférences des agents sont capturées par des fonctions d'utilité. Bien que souhaitable, la stabilité d'un appariement n'est pas garantie. À l'inverse, l'optimalité au sens de Pareto semble ne pas suffisamment discriminante. C'est la raison pour laquelle nous évaluons la qualité d'un appariement du point de vue de la société d'individus à l'aide des concepts issus de la théorie du choix social. Même si une étude de la complexité dans le pire des cas est nécessaire, les expérimentations, notamment celles concernant des données réelles, montrent que les bien-être sociaux sont pour ce problème des fonctions qui possèdent de nombreux optima locaux. C'est pourquoi je propose deux heuristiques pour chacune des notions de bien-être social. Elles retournent systématiquement des appariements valides, selon une règle sociale (utilitaire ou égalitaire) pour la première et socialement cohésifs pour la seconde.

La modélisation multi-agents nous permet de distribuer l'exécution de ces algorithmes décrits sous la forme de comportements d'agent qui consiste en de multiples négociations bilatérales concurrentes. Les procédures proposées s'appuient sur l'aspect organisationnel de la modélisation multi-agents en introduisant explicitement un niveau intermédiaire entre l'individu et la société, i.e. les coalitions d'individus qui partagent un intérêt commun pour une activité. Conformément au principe de subsidiarité, la reconfiguration d'une coalition est soumise à l'arbitrage d'un agent représentant les objectifs et contraintes d'un groupe d'individus.

Comme l'illustre ce chapitre, les méthodes de résolution multi-agents peuvent être plus efficaces que les méthodes générales pour l'optimisation de problème dont la formulation est inadaptée. De plus, au-delà d'une simple distribution qui comme ici accélère son exécution, la modélisation des comportements d'agents permet d'introduire des stratégies variées pour mettre en œuvre les contraintes et les objectifs des différentes coalitions d'agents.

1.3.5 Négociations un-à-plusieurs concurrentes

Dans le chapitre 7, je considère de multiples négociations par enchères concurrentes pour l'allocation de tâches. L'objet de négociation est mono-attribut.

Dans ce chapitre, je mets en œuvre une distribution du modèle de programmation MapReduce où l'allocation des tâches est le résultat de négociations multi-agents au cours de la phase de *reduce*. Cette réallocation dynamique des tâches s'appuie sur des décisions locales des reducers incarnés par

des agents qui s'appuient sur des fonctions d'utilités. Le rééquilibrage de charge est réalisé sans orchestration ni prétraitement des données et sans paramètre dépendant des données ou de l'environnement informatique. Pour ce faire, ces agents négocient des tâches en fonction de leurs contributions pour diminuer celle des reduceurs les moins performants, c'est-à-dire ceux qui retardent la phase de reduce. Je prouve que le processus de négociation améliore l'équité de l'allocation des tâches. Les expériences confirment que ma proposition diminue le temps d'exécution de la phase de reduce. De plus, afin d'améliorer la réactivité, je propose un processus d'enchères multiples qui permet à un agent d'enchérir dans plusieurs négociations simultanées. Il améliore la réactivité du SMA.

N.B. : Dans cette synthèse j'ai utilisé le « Je » pour souligner le caractère original de l'approche pour aborder et présenter les travaux passés et en cours. J'utiliserai le « Nous » de modestie dans la suite de ce manuscrit afin de souligner le caractère collectif de ces recherches et pour rendre hommage à l'ensemble des contributeurs.



Première partie

Synthèse des travaux

CHAPITRE 2

État de l'art

Sommaire

2.1 Objectifs	12
2.2 Conception	12
2.2.1 Objet de négociation	12
2.2.2 Parties prenantes	13
2.2.3 Protocole d'interaction	14
2.2.4 Stratégie	14
2.3 Accord	16
2.3.1 Optimalité de Pareto	16
2.3.2 Bien-être social	17
2.4 Processus	17
2.5 Exemple	18
2.6 Synthèse	18
2.7 Conclusion	19

La revue de la littérature présentée ici a été publiée dans l'article suivant :

1. MORGE, M. (2018a). Négociation bilatérale par concession : un état de l'art. In G. PICARD, C. LANG & N. MARILEAU (Éd.), *Actes des 25ème Journées Francophones sur les Systèmes Multi-Agents (JFSMA)*. (p. 75-84). Acceptance rate= 22%. Cépaduès éditions

L'objectif de ce chapitre est d'établir un état de l'art de la négociation multi-agents en proposant une grille d'analyse qui s'inspire de (JENNINGS et al., 2001; LOMUSCIO, WOOLDRIDGE & JENNINGS, 2003; MÜLLER, 1996; PARSONS, RODRIGUEZ-AGUILAR & KLEIN, 2011; WURMAN, WELLMAN & WALSH, 1998). Cette grille permet de classer ces travaux d'une part en fonction des différents ingrédients utilisés pour la conception du système de négociation (parties prenantes, protocole, stratégies) et d'autre part en fonction des propriétés exhibées par ces systèmes (efficacité, stabilité, distribution, etc).

Ce chapitre est organisé de la manière suivante. Nous commencerons par établir la définition de la négociation que nous adopterons puis ses objectifs (cf section 2.1). Dans la section 2.2, nous présenterons les ingrédients essentiels de cette négociation et nous ferons un état des lieux des différentes configurations possibles. Ensuite nous verrons comment le résultat de la négociation peut être évalué en présentant les propriétés et les métriques utilisées (cf section 2.3). Dans la section 2.4, nous présenterons comment le processus peut également être évalué. Finalement, nous illustrerons les notions présentées à travers un exemple jouet (cf section 2.5) et nous ferons une synthèse de la littérature à l'aide de la grille d'analyse décrite dans ce chapitre (cf section 2.6) et nous en tirerons quelques conclusions puis nous dresserons quelques perspectives (cf section 2.7).

2.1 Objectifs

La majorité des modèles informatiques de négociation n'explorent qu'un nombre restreint de dimensions et d'étapes envisagées par les sciences humaines et sociales (GUTTMAN, MOUKAS & MAES, 1998; LOPES, WOOLDRIDGE & NOVAIS, 2008). Les modèles informatiques se limitent à la recherche d'un accord, i.e. au « processus grâce auquel plusieurs parties prenantes aboutissent à une décision collective en verbalisant leurs demandes et convergeant vers un accord par succession de concessions ou recherche de nouvelles alternatives » (PRUITT, 1981). Ainsi, la pré-négociation (détection des conflits, définition de l'agenda et de l'objet, etc.) et l'après-négociation (évaluation, sanctions, etc.) sont écartées (LOPES et al., 2008).

La négociation intervient dans des situations où les agents sont en compétition pour satisfaire leur but mais où ils doivent également collaborer (SCHELLING, 1960). Ainsi, la situation initiale est un conflit d'intérêt mêlé à un besoin de coopération. Le but est de réaliser un échange ou de conclure une affaire, i.e. trouver un compromis attractif pour les participants. Chacun d'eux souhaite maximiser son gain via un échange de concessions.

L'émergence de la problématique de gestion des conflits dans la littérature en Informatique, en particulier en Intelligence Artificielle, a deux raisons majeures (AYDOGAN, SANCHEZ, JULIAN, BROEKENS & JONKER, 2014) : d'une part pour la conception de systèmes distribués; d'autre part pour la résolution des conflits entre humains.

La négociation est un processus coûteux en temps et en énergie, ce qui peut avoir un impact négatif sur son résultat. À cette intention, les **systèmes d'aide à la négociation** souhaitent réduire l'effort cognitif des parties prenantes dans l'exploration des possibles. Ce sont des logiciels qui implémentent des modèles et des processus pour soutenir la négociation, comme le système Inspire (G. E. KERSTEN & NORONHA, 1999). Dans cette perspective, G. KERSTEN et LAI (2007) distingue les agents assistants qui médiatisent la négociation humaine de manière passive des agents logiciels auxquels les humains délèguent (au moins partiellement) une partie du processus. Parmi ces systèmes de **négociation automatique**, les enchères électroniques ont largement été étudiées (PARSONS et al., 2011).

La négociation a également été utilisée pour la résolution distribuée de problème (KRAUS, 1997). Cette forme d'interaction permet de coordonner des entités faiblement couplées ayant un but commun dans la recherche d'une solution (DAVIS & SMITH, 1980), comme dans (DURFEE & LESSER, 1989).

En résumé, les agents qui négocient pour la résolution distribuée de problème sont coopératifs : ils ont un but commun qui prime sur leur intérêt individuel. À l'inverse, dans les systèmes de négociations automatiques, les agents favorisent leur intérêt individuel, ils sont dits « égoïstes ».

2.2 Conception

Afin d'analyser, de concevoir et d'implémenter un système de négociation, nous identifions les ingrédients essentiels de la négociation : i) l'objet de la négociation, c.-à-d. l'ensemble des résultats potentiels; ii) les parties prenantes, c.-à-d. les agents ayant leurs propres objectifs/préférences éventuellement conflictuels; iii) le protocole d'interaction, c.-à-d. les règles qui régissent les échanges verbaux; iv) les stratégies individuelles, c.-à-d. le comportement des agents.

2.2.1 Objet de négociation

L'objet de négociation est le sujet sur lequel porte les échanges. On peut distinguer :

- la négociation d'un appariement entre agents pour associer les agents les uns avec les autres (GALE & SHAPLEY, 1962);
- la négociation de tâches/ressources qui vise à établir une (ré-)allocation au sein du groupe d'agents dont les coûts/bénéfices sont évalués individuellement (ENDRISS, MAUDET, SADRI &

FRANCESCA, 2006; NONGAILLARD & MATHIEU, 2011);

- la prise de décision collective où les agents essayent de sélectionner conjointement un candidat, un plan d'action ou un ensemble de buts mutuellement acceptables (DELECROIX, MORGE, NACHTERGAELE & ROUTIER, 2016).

L'espace des résultats possibles, appelés également accords, est une variable ou un ensemble de variables discrètes ou continues. L'ensemble des accords potentiels est l'ensemble non-vide des alternatives réelles. Toutes les valeurs possibles de l'espace des accords ne sont pas nécessairement des solutions existantes. Pour au moins deux attributs, on parle de **négociation multi-attributs** et de **négociation mono-attribut** dans l'autre cas. L'objet de négociation peut s'avérer plus complexe et nécessiter un langage logique pour représenter les accords potentiels (PARSONS, SIERRA & JENNINGS, 1998).

Dans la suite de ce chapitre, nous nous focalisons sur la **négociation distributive**, dont le but est d'atteindre un accord optimal dans un ensemble d'accords qui est fixe, plutôt que sur la **négociation intégrative** qui consiste à créer un accord quitte à inventer de nouvelles alternatives au cours du processus (G. E. KERSTEN, 2001).

2.2.2 Parties prenantes

Les parties prenantes (en anglais, *stakeholder*) peuvent être deux ou plus. On parle respectivement de **négociation bilatérale** ou de **négociation multi-parties**. De par la nature compétitive de la négociation, les agents évaluent de manière individuelle et éventuellement contradictoire les accords potentiels¹.

Préférences cardinales

Une première approche consiste à capturer les penchants à l'aide d'une fonction d'utilité voire d'une valeur de réserve, i.e. le seuil au-dessus duquel l'agent est prêt à négocier. L'introduction d'une date butoir après laquelle le résultat est un désaccord permet de réduire la durée de la négociation (BAARSLAG et al., 2013; FARATIN et al., 1998; FATIMA, WOOLDRIDGE & JENNINGS, 2006; GATTI, DI GIUNTA & MARINO, 2008; KRAUS, 2001; KRAUS, WILKENFELD & ZLOTKIN, 1995; SANDHOLM & VULKAN, 1999). Afin de modéliser le fait que les agents sont moins satisfaits quand le processus est long, la plupart des travaux introduisent un taux de réduction.

Quand l'objet de négociation est multi-attributs, les fonctions d'utilité le sont également. KEE-NEY et RAIFFA (1976) introduit la fonction d'utilité additive, i.e. une somme pondérée d'utilités pour chaque critère. Cela suppose que les préférences sur les attributs sont indépendantes. Dans le cas contraire, la fonction k-additive capture la synergie entre attributs en évaluant l'utilité de chaque alternative comme la somme des utilités de chaque caractéristique plus la valeur de dépendance préférentielle entre plusieurs caractéristiques (GRABISCH, 1997). D'une manière plus générale, l'approche hypercube pondérée de ITO, KLEIN et HATTORI (2008) permet de représenter n'importe quelle fonction d'utilité multi-attributs définie sur des domaines discrets.

Préférences ordinales

Pour représenter les penchants d'un agent, une seconde approche consiste à employer une relation de préférence qui permet de comparer les alternatives les unes aux autres ainsi que le désaccord. Une relation de préférence est binaire et transitive (ARROW, 1958). Ainsi, un agent rationnel cherche à sélectionner une alternative optimale ou non-dominée si la relation de préférence est partielle.

Dans un graphe de préférence, les sommets sont des alternatives et les arcs matérialisent les préférences entre elles. Quand l'objet de négociation est multi-attributs, la représentation de ce graphe

1. ÖZTÜRK, TSOUKIÀS et VINCKE (2012) proposent un état de l'art sur la modélisation des préférences.

devient coûteuse. Par exemple, 10 attributs et 10 valeurs pour chaque attribut nécessitent 10 milliards de nœuds (10^{10}). Toutefois, l'hypothèse de l'indépendance des préférences sur les critères est souvent rejetée au profit de celle moins forte du *Ceteris Paribus*². Ainsi, les CP-nets constituent un modèle graphique pour la représentation compacte des préférences (BOUTILIER, BRAFMAN, DOMSH-LAK, HOOS & POOLE, 2004). Plus spécifiquement, les graphes d'utilité (ROBU, SOMEFUN & LA POUTRÉ, 2005) et les langages de préférences d'enchères (BOUTILIER & HOOS, 2001) sont des représentations compactes pour la négociation de ressources.

En résumé, les fonctions d'utilité constituent un modèle cardinal et quantitatif des préférences, alors que la relation de préférence est un modèle ordinal et qualitatif. Les préférences cardinales subsument les préférences ordinales.

2.2.3 Protocole d'interaction

Un protocole d'interaction est un ensemble de règles partagées par les agents auxquelles les interactions doivent se conformer³. Pour la négociation, ce sont les « règles de la confrontation » (*rules of encounter*).

Le **protocole d'offres alternées** (M. OSBORNE & RUBINSTEIN, 1990; RUBINSTEIN, 1982; STAHL, 1972), vise à allouer une ressource divisible. À chaque tour de négociation, un agent propose une alternative et son interlocuteur peut soit accepter, soit refuser cette offre. Dans le premier cas, la négociation se termine avec succès sur un accord. Sinon, la proposition n'est plus valable et les rôles sont inversés au tour suivant : le second agent propose et le premier dispose. La négociation se poursuit jusqu'à ce qu'un accord soit atteint. On peut remarquer que ce protocole est asymétrique : l'un des agents initie le processus. Ce protocole a été généralisé pour la négociation multi-parties (KRAUS, 2001).

Le **protocole de concession monotone** (ROSENSCHEIN & ZLOTKIN, 1994) est une version simultanée du précédent qui permet de ré-allouer un ensemble de tâches. À chaque tour, chaque agent propose une allocation de tâches. Un accord est atteint si l'une des offres correspond ou dépasse (en terme d'utilité) ce que l'autre demande. Si les deux offres correspondent ou dépassent la demande, l'accord peut être sélectionné de manière pseudo-aléatoire. En cas de désaccord, un autre tour se déroule où chaque agent n'est pas autorisé à offrir à l'autre agent moins (en terme d'utilité) qu'il ne l'a fait au tour précédent. Il peut soit rester ferme, c.-à-d. proposer la même utilité à son interlocuteur, soit concéder, c.-à-d. proposer une meilleure alternative (en terme d'utilité) pour l'interlocuteur. Si aucun des agents ne concède, alors la négociation se termine et l'allocation de tâche reste inchangée. Une concession consiste donc en un pas vers son interlocuteur. Contrairement au protocole précédent, celui-ci nécessite que chaque agent connaisse la fonction d'utilité de son interlocuteur pour déterminer les coups autorisés. De plus, il est symétrique : aucun agent ne joue le rôle de l'initiateur.

On distingue des protocoles simultanés où les offres sont soumises de manière concurrentes des protocoles dynamiques, où les agents prennent la parole chacun à 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.

2.2.4 Stratégie

La stratégie interface les préférences de l'agent et le protocole afin de générer l'offre initiale, des contre-offres et évaluer celles reçues. La stratégie définit le comportement de l'agent à chaque point de choix en déterminant les messages émis en conformité au protocole. Elle est spécifique à l'environnement de négociation (objet, préférences, protocole et connaissances *a priori*).

2. Toutes choses étant égales par ailleurs.

3. Dans la littérature, le terme de protocole est parfois employé comme synonyme de mécanisme ou de processus. Nous distinguons ici l'aspect normatif (le protocole) et l'aspect individuel (les stratégies) du processus.

Pour étudier les stratégies de négociation, on distingue trois approches. La **théorie des jeux** propose des modèles théoriques abstraits dont les propriétés sur les accords (cf section 2.3) et sur les processus (cf section 2.4) sont prouvées formellement. À l'inverse, l'**approche heuristique** propose des modèles qui s'appuient sur des hypothèses réalistes (e.g. rationalité limitée), qui sont évalués empiriquement, et mises en œuvre dans des applications pratiques. Finalement, l'**approche argumentative**, où les agents échangent des offres et des arguments, est hors de la portée de ce document.

Théorie de jeux

La stratégie de Zeuthen (ROSENSCHEIN & ZLOTKIN, 1994) permet d'illustrer cette approche. Selon cette stratégie, qui interface une fonction d'utilité avec le protocole de concession monotone, un agent qui a déjà concédé a moins à perdre en cas de désaccord qu'un agent qui a fait moins de concessions. Le facteur de risque de ZEUTHEN (1930) évalue la propension d'un agent à risquer le désaccord. Il correspond au rapport entre sa perte maximale d'utilité s'il concède et sa perte d'utilité en cas de désaccord. Ainsi, la négociation débute par l'offre préférée. Ensuite, tant que la négociation n'est pas terminée, un agent 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 concession minimale. On peut remarquer que cette stratégie suppose que l'agent dispose d'une information parfaite, c.-à-d. qu'il connaisse les préférences de son interlocuteur. C'est la raison pour laquelle, nous adopterons dans la suite de ce manuscrit la définition suivante :

Définition 1 (Stratégie de concession) *Selon la stratégie de concession, un agent débute la négociation en proposant l'alternative qui le satisfait le mieux. Si cette dernière est refusée, l'agent réduit pas-à-pas ses prétentions jusqu'à ce qu'une proposition soit retenue ou que sa valeur de réserve soit atteinte, i.e. le seuil au-dessus duquel l'agent est prêt à négocier.*

Un grand nombre de stratégies ont été proposées pour la négociation bilatérale mono-attribut. Bien qu'elles s'appuient toutes sur le principe de concession selon lequel l'agent doit réduire ses prétentions au fur et à mesure des échanges, toutes ces variantes correspondent à des environnements de négociation différents :

- FATIMA et al. (2006) propose une stratégie pour le protocole d'offres alternées où chaque agent a une information incomplète sur l'opposant ;
- GATTI et al. (2008) suppose que chaque agent a un taux de réduction, une valeur de réserve et une date butoir qui lui est propre dans une situation asymétrique : l'un des agents ne connaît pas la date butoir de son opposant ;
- RUBINSTEIN (1982) considère un processus sans date butoir avec information parfaite puis avec information imparfaite dans (RUBINSTEIN, 1985) ;
- SANDHOLM et VULKAN (1999) s'appuie sur une version continue du protocole, i.e. chaque agent peut soumettre une offre à tout moment et où chaque agent a sa propre date butoir.

Le résultat d'une négociation dépend non seulement de la stratégie et des préférences des agents mais également du protocole et de l'information dont disposent les agents (HARSANYI, 1977).

Approche heuristique

Dans l'approche heuristique, les fonctions de décision introduites par FARATIN et al. (1998) sont fondatrices. C'est une version continue du protocole d'offres alternées avec une date butoir qui est

considérée. Alors que les offres reçues sont évaluées par des fonctions d'utilité multi-attributs, les offres sont générées à partir d'une combinaison linéaire de fonctions simples, appelées tactiques. Une tactique calcule la valeur d'un attribut (le prix, la quantité, la qualité, etc.) en fonction d'un critère. Ainsi, trois familles de tactiques sont suggérées :

- les tactiques temporelles qui déterminent quand un agent doit concéder;
- les tactiques qui contraignent les offres en fonction des ressources disponibles;
- les tactiques imitatives qui déterminent le comportement de l'agent en fonction de celui de son opposant.

Selon la fonction de concession d'une tactique temporelle, un agent débute la négociation en proposant une valeur maximale (ou minimale si l'utilité décroît avec la valeur de l'attribut) puis concède jusqu'à atteindre à la date butoir la valeur de réserve. Cette fonction est paramétrée. Une fonction polynomiale concède initialement plus vite qu'une fonction exponentielle. Ainsi, on distingue deux tactiques temporelles :

- la tactique *Boulware*⁴ qui consiste à rester sur ses positions jusqu'à la date butoir pour y proposer sa valeur de réserve;
- la tactique *Conceder* qui consiste à concéder rapidement.

Selon la fonction de concession réciproque, l'agent reproduit proportionnellement le comportement de son opposant. Cette fonction peut être raffinée pour être pseudo-aléatoire ou lissée en prenant en compte le comportement de l'opposant dans une fenêtre temporelle.

S'inspirant de ces travaux, la compétition annuelle ANAC, réalisée avec la plate-forme GENIUS (*General Environment for Negotiation with Intelligent multi-purpose Usage Simulation*) proposée par LIN et al. (2012), confronte depuis 2011 des agents dans des négociations bilatérales multi-attributs au travers d'un environnement aussi réaliste que possible. Contrairement à la grande majorité des stratégies de négociation, la stratégie de négociation proposée par AYDOGAN, BAARSLAG, HINDRIKS, JONKER et YOLUM (2013) est basée sur un modèle de préférences qualitatif et ordinal : un CP-net (BOUTILIER et al., 2004).

Une stratégie est constituée de trois composants :

1. une règle d'acceptabilité qui décide si une offre est admissible;
2. une stratégie d'offre qui détermine la concession en fonction des préférences, de l'historique et éventuellement d'une valeur de réserve, du temps et du modèle d'opposant (si disponible);
3. éventuellement une stratégie d'apprentissage ou d'adaptation qui s'appuie sur un modèle d'opposant.

2.3 Accord

Afin d'évaluer la qualité d'un accord, nous considérons le critère de Pareto puis la théorie du choix social.

2.3.1 Optimalité de Pareto

Le critère de Pareto compare les alternatives du point de vue d'un groupe d'agents en conflits. Une alternative x domine au sens de Pareto une alternative y pour un groupe d'agents si x est au moins aussi bien que y pour tous les agents et qu'au moins un agent préfère strictement x à y . Une alternative est considérée comme optimale au sens de Pareto si elle n'est pas dominée au sens de Pareto. En d'autres termes, une alternative n'est pas optimale s'il en existe une meilleure pour un agent

4. Le nom de cette stratégie semble provenir d'une incompréhension de la stratégie « à prendre ou à laisser » promue par (BOULWARE, 1969).

sans qu'elle ne soit pire pour aucun des autres. On peut remarquer que la relation de dominance est une relation de préordre (réflexive et transitive), éventuellement partielle. Selon le lemme de Zorn, l'existence d'une alternative optimale au sens de Pareto est garantie.

Le caractère compétitif d'une négociation peut être évalué a priori en fonction du profil (des préférences) des agents. La distance entre alternatives est définie par une métrique, i.e. une fonction définie positive qui est symétrique et sous-additive. Si l'alternative optimale au sens de Pareto dans l'espace des accords est unique, le taux d'opposition est la distance minimale entre les alternatives de l'ensemble des accords et celle-ci (n'étant pas nécessairement une alternative possible). Sinon, le taux d'opposition est mesuré par la distance moyenne vis-à-vis de la frontière de Pareto, i.e. l'ensemble des alternatives optimales dans l'espace des accords. G. E. KERSTEN, NORONHA et TEICH (2000) définissent le taux d'opposition par le gradient des utilités.

Le critère de Pareto suppose un minimum d'hypothèse sur les préférences individuelles mais débouche sur un classement partiel des alternatives. La posture Parétienne, selon laquelle la satisfaction ne peut se comparer d'un individu à un autre, est largement adoptée (ARROW, 1951).

2.3.2 Bien-être social

Contrairement à l'approche Parétienne, la théorie du choix social réhabilite l'analyse des comparaisons interpersonnelles (SEN, 1970). Cette théorie réconcilie l'utilitarisme de Jeremy Bentham et la justice distributive prônée par John Rawls (MOULIN, 2002).

Les fonctions de choix social agrègent les utilités des individus. Alors que maximiser le bien-être utilitaire (la somme des utilités) vise à satisfaire la société dans son ensemble, maximiser le bien-être égalitaire (l'utilité minimale) consiste à réduire les inégalités dans la population, i.e. améliorer la satisfaction de l'agent le moins bien doté. Maximiser le bien-être de Nash (le produit des utilités) est un compromis entre ces deux règles de décision collective. Une alternative qui maximise le bien-être utilitaire ou le bien-être de Nash est optimale au sens de Pareto.

2.4 Processus

Nous proposons ici des critères pour évaluer un processus de négociation qui comprend un environnement (objet, préférences, protocole, contraintes temporelles et connaissances *a priori*) et des stratégies.

Rationalité individuelle. Les agents doivent être incités à participer à la négociation. Une négociation est individuellement rationnelle pour un agent s'il préfère un accord à un désaccord. Une négociation individuellement rationnelle doit l'être pour tous les agents.

Stabilité. Un protocole est stable s'il incite les agents à se comporter d'une manière particulière. Par exemple, l'équilibre de Nash est une situation dans laquelle chaque agent connaît les stratégies des autres agents et aucun agent n'a intérêt à modifier sa propre stratégie. En d'autres termes, aucun joueur ne peut faire mieux en changeant unilatéralement sa stratégie. L'existence d'un équilibre de Nash n'est pas garantie et il n'est pas nécessairement optimal au sens de Pareto. La notion d'équilibre de Nash a été étendue aux jeux dynamiques qui modélisent plus fidèlement les négociations qui s'appuient sur le protocole d'offres alternées. On parle alors d'équilibre parfait en sous-jeu et d'équilibre séquentiel (M. J. OSBORNE & RUBINSTEIN, 1994).

Distribution. Un processus de négociation est distribué s'il ne nécessite pas un point central qui collecte les offres (on parle alors de médiateur) voire qui calcule l'accord (on parle alors d'arbitre). Cette situation est souhaitable pour éviter les éventuelles pannes du point central et les goulots d'étranglement.

Complexité algorithmique. Un processus de négociation est dit « simple » si le calcul de la stratégie optimale est traitable.

Complexité communicationnelle. Un processus de négociation est dit « bref » si le nombre d'offres échangées pour atteindre un accord est restreint. De plus, l'appel à la cantonade (*broadcasting*) est coûteux.

Garantie de succès. Un processus de négociation est garanti de succès si qu'elles que soient les préférences des agents, un accord est atteint.

2.5 Exemple

Afin d'illustrer les notions précédentes, nous présentons ici un exemple jouet. Afin de déterminer une allocation au sein d'un groupe d'agents, nous considérons deux ressources r_1 et r_2 ainsi que deux agents a_1 et a_2 . Nous sommes dans un cas typique de négociation distributive. L'objet de négociation est un partitionnement de l'ensemble des ressources entre les deux agents. La négociation est mono-attribut.

Les préférences des agents sont capturées par les fonctions d'utilité u_1 et u_2 des agents a_1 et a_2 , respectivement (cf tableau 2.1). Chacune d'elle évalue la satisfaction individuelle de l'agent vis-à-vis du lot qui lui est alloué. Nous supposons ici que ces fonctions sont :

- sans externalité, la satisfaction d'un agent dépend uniquement de son propre lot;
- additives, i.e. aucune synergie entre ressources n'opère.

Si aucune ressource n'est initialement allouée, le processus doit, pour être individuellement rationnelle, garantir l'allocation d'au moins une ressource par agent.

lot	\emptyset	$\{r_1\}$	$\{r_2\}$	$\{r_1, r_2\}$
$u_1(lot)$	0	10	8	18
$u_2(lot)$	0	5	6	11

TABLE 2.1 – Exemple de fonctions d'utilité additives sans externalité

Nous considérons un protocole de négociation bilatérale qui est dynamique, asymétrique, sans date butoir et sans taux de réduction. L'agent a_1 joue arbitrairement le rôle d'initiateur. Si les deux agents adoptent la stratégie de concession (cf définition 1), la négociation se déroule comme suit :

1. l'agent a_1 initie la négociation avec l'alternative qui le satisfait le mieux : il propose de s'approprier la totalité des ressources;
2. Cette proposition est inadmissible pour l'agent a_2 qui a son tour propose de s'approprier la totalité des ressources;
3. Cette proposition est inadmissible pour l'agent a_1 qui réduit ses prétentions en s'appropriant la ressource qu'il préfère r_1 et en concédant r_2 à son opposant;
4. La précédente proposition est admissible et c'est la concession minimale que puisse faire a_2 comme il préfère r_2 à r_1 . L'agent a_2 accepte cette allocation.

À l'issue de cette négociation, un accord est atteint. Celui-ci est un optimal au sens de Pareto qui maximise le bien-être utilitaire mais pas le bien-être égalitaire.

2.6 Synthèse

Comme nous l'avons vu, les occasions de complexifier la négociation sont multiples (multi-parties, multi-attributs, argumentation, hypothèses sur les préférences des agents). Nous nous sommes focalisés ici sur la négociation bipartie mono-attribut à base de concessions, où les échanges se réduisent en une séquence d'offres et de contre-offres.

Les tableaux 2.2 et 2.3 synthétisent l'ensemble des travaux évoqués ici selon notre grille d'analyse. Sans être exhaustifs, ces tableaux présentent des travaux emblématiques du domaine. Par souci de clarté, nous avons restreint notre revue de la littérature aux négociations bilatérales qui se basent soit sur des protocoles asymétriques (comme le protocole d'offres alternées) soit sur des protocoles symétriques (comme le protocole de concession monotone) éventuellement dans leur version continue. Pour les mêmes raisons, nous nous sommes focalisés sur la négociation mono-attribut, même si (AYDOGAN et al., 2013; BAARSLAG et al., 2013; FARATIN et al., 1998; FATIMA et al., 2006) concernent également les négociations multi-attributs. Les tableaux 2.2 et 2.3 soulignent que la plupart des travaux s'appuient sur des fonctions d'utilité (noté u), qu'elles prennent en compte (ou non) un taux de réduction (noté δ) et/ou une valeur de réserve (notée θ) à l'exception de (AYDOGAN et al., 2013) qui s'appuie sur une relation de préférence (noté \succsim).

ROSENSCHEIN et ZLOTKIN (1994) montrent que si les deux agents adoptent la stratégie de Zeuthen, alors l'accord maximise le produit des utilités et est donc Pareto-optimal mais que cette situation n'est pas un équilibre de Nash⁵. Étant donné l'environnement de négociation défini dans (RUBINSTEIN, 1982), un équilibre parfait en sous-jeu est atteignable au premier tour. Dans son extension à information imparfaite (RUBINSTEIN, 1985), l'environnement possède un équilibre séquentiel. Étant donné que FATIMA et al. (2006) étudient plusieurs environnements de négociation (mono/multi-attribut(s) à information parfaite/imparfaite) et plusieurs stratégies, nous ne pouvons résumer ici l'ensemble des résultats théoriques obtenus en terme de stabilité, d'efficacité et de complexité. Dans l'environnement de négociation envisagé par GATTI et al. (2008), deux stratégies mixtes mènent à deux situations d'équilibre distincts. La complexité algorithmique de la stratégie optimale est $\mathcal{O}(m.\gamma)$ où m est le nombre d'attributs et γ est la date butoir. D'une manière générale, dans l'environnement de négociation de SANDHOLM et VULKAN (1999), le seul équilibre séquentiel est atteint quand les agents attendent la première date butoir. Ainsi, la puissance de l'agent ne réside pas dans son calcul mais au contraire dans sa capacité à attendre. Dans un environnement de négociation muni d'un taux de réduction, la solution n'est Pareto-optimale que si l'accord est immédiat, i.e. obtenu au premier tour, car sinon la ressource et donc la satisfaction des agents décroissent. En théorie des jeux, on distingue les travaux mathématiques qui ne s'intéressent pas aux problèmes de complexité et les travaux informatiques qui intègrent les problèmes liés au calcul. L'approche heuristique adoptée dans (AYDOGAN et al., 2013; BAARSLAG et al., 2013; FARATIN et al., 1998) ne permet pas d'aboutir à des résultats formels (noté ND pour non déterminé), mais ces propriétés sont évaluées empiriquement. Finalement, la rationalité individuelle et la distribution sont des propriétés requises pour la négociation.

2.7 Conclusion

Nous avons proposé dans ce chapitre une revue de l'état de l'art sur la négociation multi-agents en nous focalisant sur la variante la plus élémentaire : la négociation distributive, bilatérale, mono-attribut à base de concessions. Dans ce contexte, nous avons distingué les différents travaux existants à partir des différents environnements de négociation, i.e. :

1. l'objet de négociation ici restreint à un attribut discret ou continu ;
2. les préférences des agents qui sont modélisables par une relation de préférences ou par une fonction d'utilité, éventuellement associée à une valeur de réserve voire un facteur de réduction ;
3. le protocole de négociation qui peut être symétrique ou asymétrique, discret ou continu, voire associé à une date butoir ;
4. la connaissance *a priori* des agents vis-à-vis de leurs opposants qui peut être parfaite ou imparfaite.

5. Nous ne considérons pas ici la stratégie étendue de Zeuthen.

	(RUBINSTEIN, 1982)	(RUBINSTEIN, 1985)	(ROSENSCHEIN & ZLOTKIN, 1994)	(SANDHOLM & VUKAN, 1999)	(FATMA, WOOLDRIDGE & JENNINGS, 2006)	(GATTI, DI GIUNTA & MARINO, 2008)
Agents	2	2	2	2	2	2
Objet	mono	mono	mono	mono	multi	mono
Préférences	u δ	u δ	u δ	u δ	u $\delta - \theta$	u δ
Protocole	asym. non	asym. non	sym. non	asym. continu oui	asym. oui	asym. oui
date butoir	non	non	non	oui	oui	oui
Connaissance	impartiale	partiale	partiale	impartiale	partiale/impartiale	impartiale
Efficacité	non-Pareto	Pareto	Pareto - Nash	non Pareto
Équilibre	séquentiel	Partait	partait	séquentiel	...	séquentiel
Complexité algorithmique	ND	ND	ND	$\emptyset(1)$...	$\emptyset(m \cdot \gamma)$
Complexité comm.	ND	ND	ND	$\emptyset(1)$...	$\emptyset(\gamma)$
Rationalité individuelle	oui	oui	oui	oui	oui	oui
Distribution	oui	oui	oui	oui	oui	oui

TABLE 2.2 – Théorie des jeux

	(FARATIN, SIERRA & JENNINGS, 1998)	(BAARSLAG et al., 2013)	(AYDOGAN, BAARSLAG, HINDRIKS, JONKER & YOLUM, 2013)
Agents	n	2	2
Objet	multi	multi	multi
Préférences	u δ	u $\delta - \theta$	λ $\delta - \theta$
Protocole	asym. continu	asym. continu	asym. continu
date butoir	oui	oui	non
Connaissance	imparfaite	imparfaite	imparfaite
Efficacité	ND	ND	ND
Équilibre	ND	ND	ND
Complexité algorithmique	ND	ND	ND
Complexité comm.	ND	ND	ND
Rationalité individuelle	oui	oui	oui
Distribution	oui	oui	oui

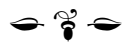
TABLE 2.3 – Approche heuristique

Les stratégies de négociation sont adaptées à un environnement de négociation particulier. Outre la rationalité individuelle et la distribution qui sont des propriétés requises pour un processus de négociation, ils se distinguent de par leurs propriétés :

1. l'efficacité qui détermine si l'accord atteint est optimal au sens de Pareto voire s'il maximise un bien-être social;
2. la stabilité qui détermine si les agents sont incités à se comporter d'une manière particulière;
3. la complexité algorithmique et communicationnelle, i.e. la simplicité du calcul de la stratégie et le nombre d'échanges nécessaires à un accord.

Deux approches peuvent être adoptées pour la conception des stratégies d'agents. En théorie des jeux, l'environnement de négociation est restreint et les propriétés peuvent être prouvées formellement. Selon l'approche heuristique, l'environnement de négociation est réaliste et les propriétés sont empiriquement évaluées.

Enfin, nous avons observé que le modèle des préférences qui est le plus largement utilisé est quantitatif et cardinal. Toutefois, la correction et la pertinence de la sémantique de la fonction d'utilité restent controversées en théorie de la décision et en théorie du choix social. D'abord, la représentation numérique de l'utilité mène à des opérations d'agrégation qui peuvent être mal interprétées. De plus, l'utilité suppose l'évaluation intrinsèque des alternatives plutôt que leur évaluation comparative (DELECROIX et al., 2016). Finalement, si un millier d'alternatives ont la même valeur d'utilité, ces alternatives sont représentées par la même satisfaction : la multiplicité des offres n'est pas capturée. Ce choix doit être questionné pour : i) l'élicitation des préférences individuelles dans les systèmes de négociation automatique; ii) la conception des systèmes de négociation automatique à base d'argumentation qui s'appuient sur des modèles qualitatifs.



CHAPITRE 3

Négociation bilatérale

Sommaire

3.1 Introduction	23
3.2 Scénario d’approvisionnement	25
3.3 Architecture d’agent	27
3.3.1 Prise de décision individuelle	29
3.3.2 Prise de décision sociale	32
3.3.3 Interaction sociale	34
3.4 Scénario nominal	36
3.5 Conclusion	37

Les travaux présentés dans ce chapitre ont été réalisés entre 2006 et 2009 avec Paolo MANCARELLA à l’Università di Pisa. Dans le cadre du projet européen FP6 “*ARGUmentation as a foundation for the semantic Grid*” (ArguGRID), nous avons collaboré avec Stefano BROMURI, Jarred MCGINNIS et Kostas STATHIS (Royal Holloway University of London). Ces travaux ont été supervisés par Francesca TONI (Imperial College). Ils ont notamment fait l’objet des publications suivantes :

- MORGE, M. & MANCARELLA, P. (2014). Arguing over Goals for Negotiation : Adopting an Assumption-Based Argumentation Decision Support System. *Group Decision and Negotiation*, 23(5), 979-1012. h-index=50, Q2 (Scimago)
- MORGE, M., MCGINNIS, J., BROMURI, S., MANCARELLA, P., STATHIS, K. & TONI, F. (2013). Argumentative Agents for Service-Oriented Computing. In M. GANZHA & L. C. JAIN (Éd.), *Multiagent Systems and Applications* (T. 45, p. 217-255). Intelligent Systems Reference Library. Springer Berlin Heidelberg
- MATT, P.-A., MORGE, M. & TONI, F. (2010). Combining statistics and arguments to compute trust. In *Proceedings of the ninth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS’2010)* (p. 209-216). A* (CORE). IFAAMAS

Dans ce projet, notre contribution réside dans le modèle d’argumentation pour la prise de décision et dans l’architecture d’agent sur laquelle ce chapitre se focalise.

3.1 Introduction

Dans ce chapitre, nous nous intéressons à la prise de décision collective. En particulier, nous considérons la négociation bilatérale pour l’établissement d’un contrat sur un bien ou un service. Afin que les participants puissent « faire affaire », ils marchandent dans leur propre intérêt afin de

maximiser leur satisfaction en insistant sur certains aspects tout en concédant sur d'autres. Nous considérons donc que l'objet de négociation est multi-attributs.

L'argumentation permet aux agents de négocier l'approvisionnement de services atomiques et la résolution des conflits au cours du processus de négociation. La plus-value de l'argumentation réside dans la justification des positions (RAHWAN, 2005) et constitue un modèle de délibération dialectique pour l'interaction et la prise de décision afin d'évaluer la validité des informations obtenues et de résoudre les différents d'opinion. C'est un ingrédient essentiel pour la prise de décision (AMGOUD & PRADE, 2006; BENCH-CAPON & PRAKKEN, 2006; KAKAS & MORAITIS, 2003; OUERDANE, MAUDET & TSOUKIAS, 2007), la communication entre agents (MCBURMEY & PARSONS, 2002) et la négociation (AMGOUD, DIMOPOULOS & MORAITIS, 2007; KAKAS & MORAITIS, 2006; RAHWAN et al., 2004).

Le projet ArguGRID a développé une grille de calcul orientée service peuplée d'agents logiciels agissant au nom de fournisseurs et de demandeurs de service. La plate-forme ArguGRID (cf figure 3.1) est constituée de quatre composants.

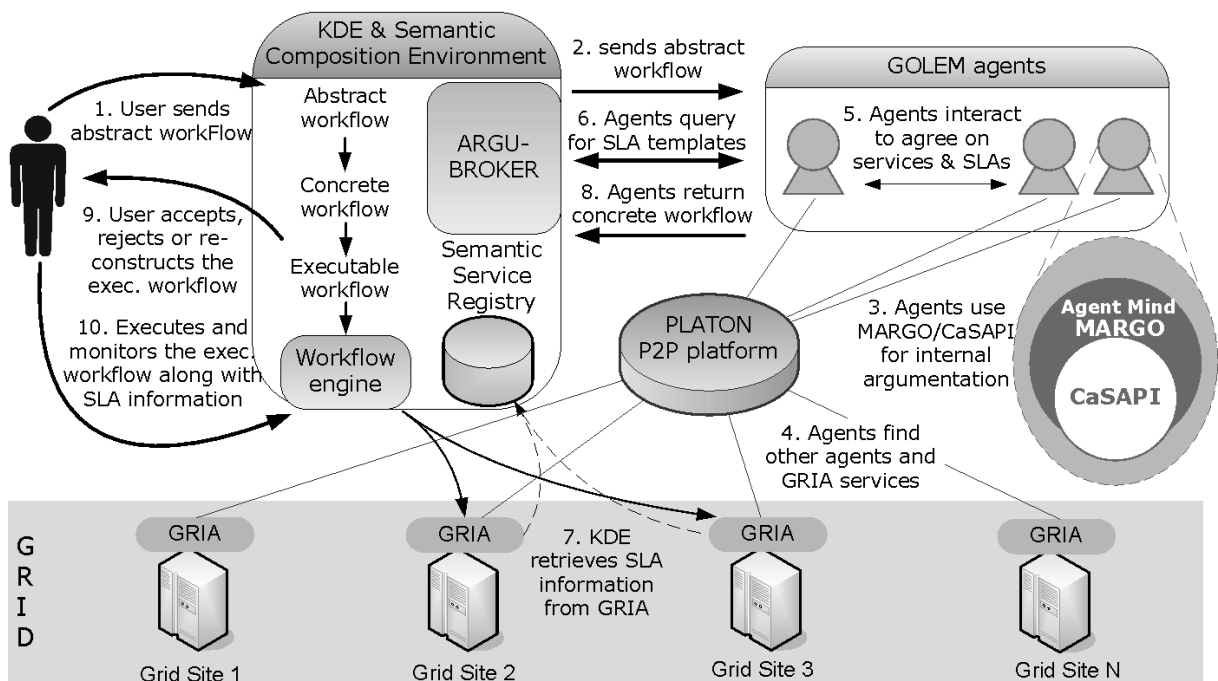


FIGURE 3.1 – La plate-forme ArguGRID

KDE est un logiciel commercial développé par InforSense Ltd. Ce système permet de construire des applications à partir de services Web ou de services Grid (GHANEM, AZAM, BONIFACE & FERRIS, 2006). Pour les besoins d'ArguGRID, le système KDE a été étendu pour permettre la description sémantique des services. De cette manière, les besoins de l'utilisateur sont mis en correspondance avec des services concrets exécutables sur la grille de calcul.

GRIA est le *middleware* GRID que le projet ArguGRID a choisi pour déployer les scénarios. GRIA est une infrastructure orientée service élaborée pour les collaborations B2B au travers de l'approvisionnement de services entre différentes organisations de manière sécurisée, interopérable et flexible. **PLATON** (*Peer-to-Peer Load Adjusting Tree Overlay Networks*) est une plate-forme de pair-à-pair pour permettre des requêtes multi-attributs et variées (LYMBEROPOULOS, BROMURI, STATHIS, KAFETZOGLOU & GRAMMATIKO, 2007) qui a été développée pour permettre les mécanismes de découverte de services et la répartition de charges. Ce dernier mécanisme est nécessaire pour garantir une complexité logarithmique de la recherche dans un arbre distribué multi-attribut.

GOLEM (*Generalized OntoLogical Environments for Multi-agent systems*) est un middleware d'environnement multi-agents (BROMURI & STATHIS, 2007) qui a été développé pour le projet ArguGRID. Dans ce chapitre, nous présentons comment GOLEM est utilisé pour accueillir les agents MARGO.

Étant donné le contexte d'ArguGRID, ce chapitre se focalise sur l'architecture d'agent pour la sélection de services et de partenaires dans un scénario d'approvisionnement électronique. Le scénario d'approvisionnement électronique illustre la manière dont les agents sélectionnent et échangent des services pour leur intégration dans un environnement ouvert et distribué. Dans ce contexte, un utilisateur humain demandant un service doit spécifier un workflow abstrait décrivant ses besoins, ses préférences et ses contraintes. La sélection des services, mais également la sélection des partenaires, est une tâche déléguée à des agents autonomes.

L'esprit de l'agent est composé de trois modules : (i) le module de prise de décision individuelle permettant la traduction des buts, des préférences et des contraintes relatives à la requête de l'utilisateur sous la forme d'une représentation abstraite des besoins de l'utilisateur ; (ii) le module de la prise de décision sociale permettant de traduire ces représentations abstraites en une représentation concrète grâce à la négociation de contrats ; (iii) le module d'interaction sociale gérant les communications selon les règles sociales d'interaction.

On peut noter que les services disponibles sur la plate-forme ArguGRID sont encapsulés dans des services Web et accessibles aux agents déployés. Dans ce chapitre, nous ne nous focaliserons pas sur la mise en œuvre de ces services, mais plutôt sur la manière dont les services sont sélectionnés, au moyen de l'argumentation. Ainsi, nous ne considérons pas les mécanismes de raisonnement à partir de données du Web sémantique ou les ontologies pour décrire ces services Web. Nous nous appuyons sur une ontologie fixe partagée par tous les agents. Nous nous focaliserons sur l'argumentation, les protocoles de communications et le processus de raisonnement.

La suite du chapitre est organisée comme suit. La section 3.2 décrit notre scénario d'approvisionnement. La section 3.3 propose l'architecture de notre modèle cognitif. La section 3.4 illustre le processus de délibération des agents et les capacités communicationnelles à travers un scénario nominal. La section 3.5 résume notre proposition.

3.2 Scénario d'approvisionnement

Afin d'illustrer notre modèle et notre architecture, nous considérons un scénario d'approvisionnement où un acheteur souhaite acquérir un service/produit combiné auprès de fournisseurs de type A et de type B. Les fournisseurs combinent leurs compétences afin de proposer des solutions à l'acheteur. Chaque agent représente un utilisateur, i.e. un demandeur de service ou un fournisseur de service. Chaque agent « fournisseur » est responsable de différents services Web. Cette opération est réalisée à travers un processus en 6 étapes où :

1. un demandeur recherche des fournisseurs potentiels ;
2. il collecte des informations afin d'évaluer ces fournisseurs ;
3. il crée une *shortlist* de fournisseurs potentiels ;
4. le demandeur commande aux fournisseurs de cette liste un devis pour les services ;
5. il choisit un fournisseur ;
6. le demandeur et le fournisseur négocient les termes et les conditions du contrat comme le prix ou la qualité de service.

Si la dernière étape échoue, alors l'étape 5 est réitérée. Quand aucun service convenable n'est trouvé, l'agent retourne vers l'utilisateur pour lui demander de relâcher ses contraintes.

L'échange d'offres et d'acceptations mène à des contrats, i.e. des liens légaux entre les fournisseurs et les demandeurs de services qui sont des engagements (e.g. des obligations) d'un agent vis-à-vis d'un autre agent à propos d'un service. Dans ce chapitre, nous supposons que les contrats sont de

simples transactions entre un fournisseur et un demandeur caractérisées par les propriétés des services fournis. Nous ignorons les problèmes complexes relatifs à la gestion des engagements, aux sanctions et aux pénalités, etc. Notre définition de contrat est donnée dans la section 3.3.3. Les contrats créent des organisations virtuelles constituées de l'acheteur, de l'agent de type A sélectionné et de l'agent de type B sélectionné.

Au sein du scénario d'approvisionnement électronique, nous considérons le cas particulier d'un acheteur qui souhaite acquérir un système de télé-commande (S) qui est composé d'un système informatique (S_a) et d'une connexion Internet (S_b). Les agents de type A qui fournissent S_a sont A1 et Alice. Alors que Alice est responsable des instances concrètes de services $S_a(a_1)$ et $S_a(a_2)$, A1 peut fournir $S_a(a_3)$ et $S_a(a_4)$. Les agents de type B fournissant S_b sont Bob et Barbara. Bob représente les services concrets $S_b(c)$, $S_b(d)$, $S_b(e)$ et $S_b(f)$. D'autre part, Barbara représente les services $S_b(g)$ et $S_b(h)$. Les agents de type A sont responsables de la sélection des agents de type B et ils fournissent le service combiné S à l'acheteur (cf figure 3.2). Une organisation virtuelle possible est constituée de Bob, de A1 et de l'acheteur à l'aide de deux contrats : le premier entre l'acheteur et A1 concerne le service combiné S , le second entre A1 et Bob concerne le service S_b . Ces contrats peuvent être négociés comme suit. Tout d'abord, l'acheteur joue le rôle de demandeur et tous les agents de type A jouent les rôles de fournisseurs potentiels dans un processus d'approvisionnement du service S . Ensuite, les agents de type A jouent le rôle de demandeur et les agents de type B jouent les rôles de fournisseurs potentiels dans le processus d'approvisionnement du service S_b .

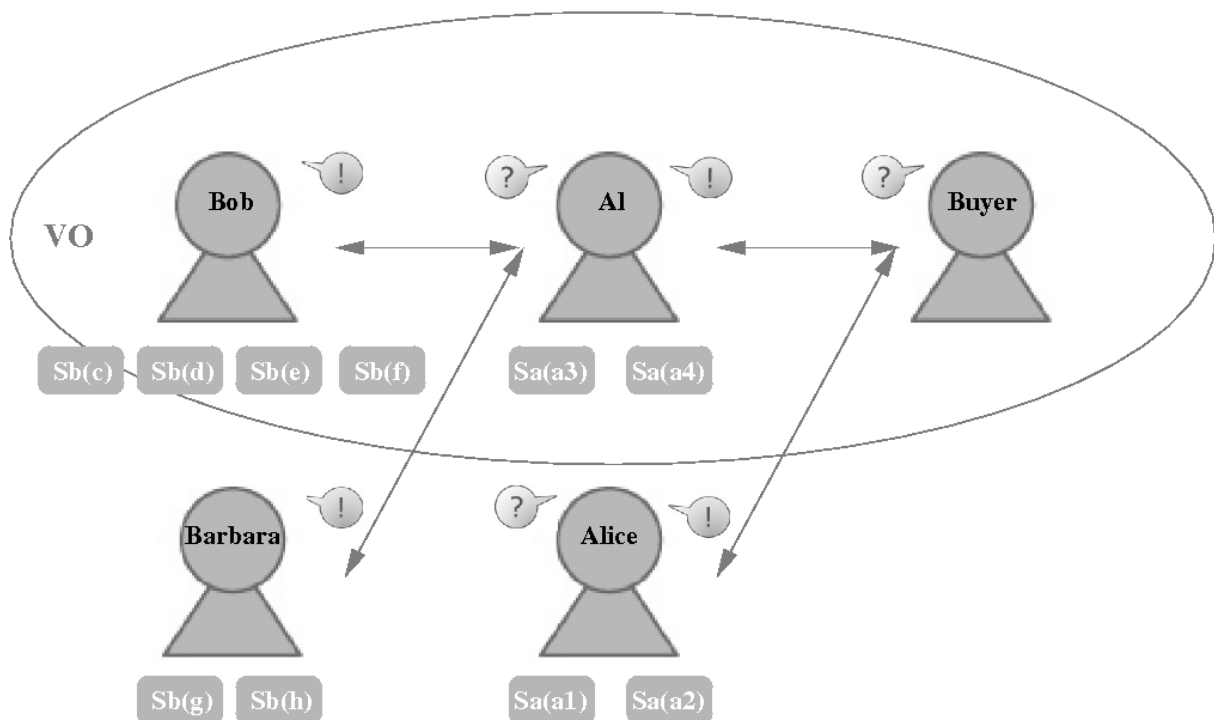


FIGURE 3.2 – Les cas d'usage de notre scénario

Dans la suite du chapitre, nous nous focaliserons sur le processus d'approvisionnement impliquant A1 en tant que demandeur et Bob en tant que fournisseur. Le but de A1 consiste à trouver un service S_b fourni par un agent de type B. Selon ses préférences et contraintes :

- le coût du service doit être faible (e.g. dans un budget de 10 euros par mois);

- la qualité du service doit être élevée (e.g. avec une assistance 24 heures sur 24).

En prenant en compte son but et ses préférences voire ses contraintes, A1 doit résoudre un problème de prise de décision qui porte sur un service et un fournisseur pour celui-ci. D'autre part, le but du fournisseur Bob consiste à prendre une décision sur le service à fournir. Selon les préférences et contraintes de Bob :

- le coût du service doit être élevé (e.g. dans un budget de 40 euros par mois), et
- la qualité du service doit être faible (e.g. avec une assistance de deux heures par jour).

Une négociation est une interaction en plusieurs étapes entre A1 et Bob au travers de laquelle le but commun consiste à « faire affaire » tout en résolvant les conflits en ce qui concerne le coût et la qualité. La plus-value de l'approche argumentative que nous allons nous approprier réside dans l'utilisation d'informations supplémentaires au cours de l'interaction. À cette intention, nous supposons que les agents partagent une ontologie commune pour décrire les services (en particulier les intervalles de prix et de garantie pour le contrat). De plus, l'argumentation permet aux agents d'identifier et de résoudre les conflits potentiels. Les deux processus de décision des participants sont dynamiques de par les informations à propos des autres agents, des services demandés/fournis et leurs caractéristiques obtenues de façon incrémentale au cours du processus d'approvisionnement. Comme nous le verrons par la suite, le résultat du processus est un contrat pour que Bob fournisse le service S_b à A1, avec un coût faible (e.g. 10 euros par mois) et une qualité faible (e.g. une assistance de deux heures par jour).

Dans notre architecture d'agent, le processus de décision de chaque agent et le processus de négociation sont réalisés à travers l'argumentation. Dans le cas d'usage, A1, en tant que demandeur, utilise l'argumentation pour collecter l'information sur les services et fournisseurs. Par exemple, A1 peut demander à Alice son opinion à propos de Bob et lui demander de se justifier (avec un argument). A1 (respectivement Bob), en tant que demandeur (respectivement fournisseur), utilise l'argumentation pour décider quel service demander (respectivement fournir) en prenant en compte des préférences et contraintes conflictuelles et éventuellement des informations récoltées inconsistantes. De plus, avec l'argumentation, les participants fournissent une explication interactive et intelligible de leur choix. Par exemple, A1 peut soutenir qu'un service lui convient car son coût est faible. Cet argument incite Bob à concéder sur le prix pour atteindre rapidement un accord. En conséquence, dans notre approche, les agents argumentent pour influencer les autres.

3.3 Architecture d'agent

Dans cette section, nous décrivons l'architecture de notre agent en terme de corps et d'esprit en nous focalisant sur le composant correspondant à l'esprit.

Notre architecture d'agent, représentée dans la figure 3.3, est une adaptation de l'architecture de BRACCIALI et al. (2006) étendue dans GOLEM (BROMURI & STATHIS, 2007). Le « corps » appréhende les stimuli en utilisant le module de communication (CM), i.e. les événements générés par l'interface graphique (GUI) et les messages provenant des autres agents comme ceux provenant du registre utilisé pour la découverte de partenaires. Les messages sont acquis et stockés dans la queue des messages entrants (*Incoming Message Queue*, IMQ) jusqu'à leur traitement. De manière similaire, les messages envoyés par l'agent sont stockés dans la queue des messages sortants (*Outcoming Message Queue*, OMQ). Le corps et l'esprit fonctionnent comme des co-routines, permettant ainsi au processus de perception/action et au processus de raisonnement de fonctionner de manière concurrente.

L'esprit cognitif de l'agent est divisé en trois modules : le module de prise de décision individuelle (*Individual Decision Making Module*, IDMM), le module de prise de décision sociale (*Social Decision Making Module*, SDMM) et le module d'interaction sociale (*Social Interaction Module*, SIM).

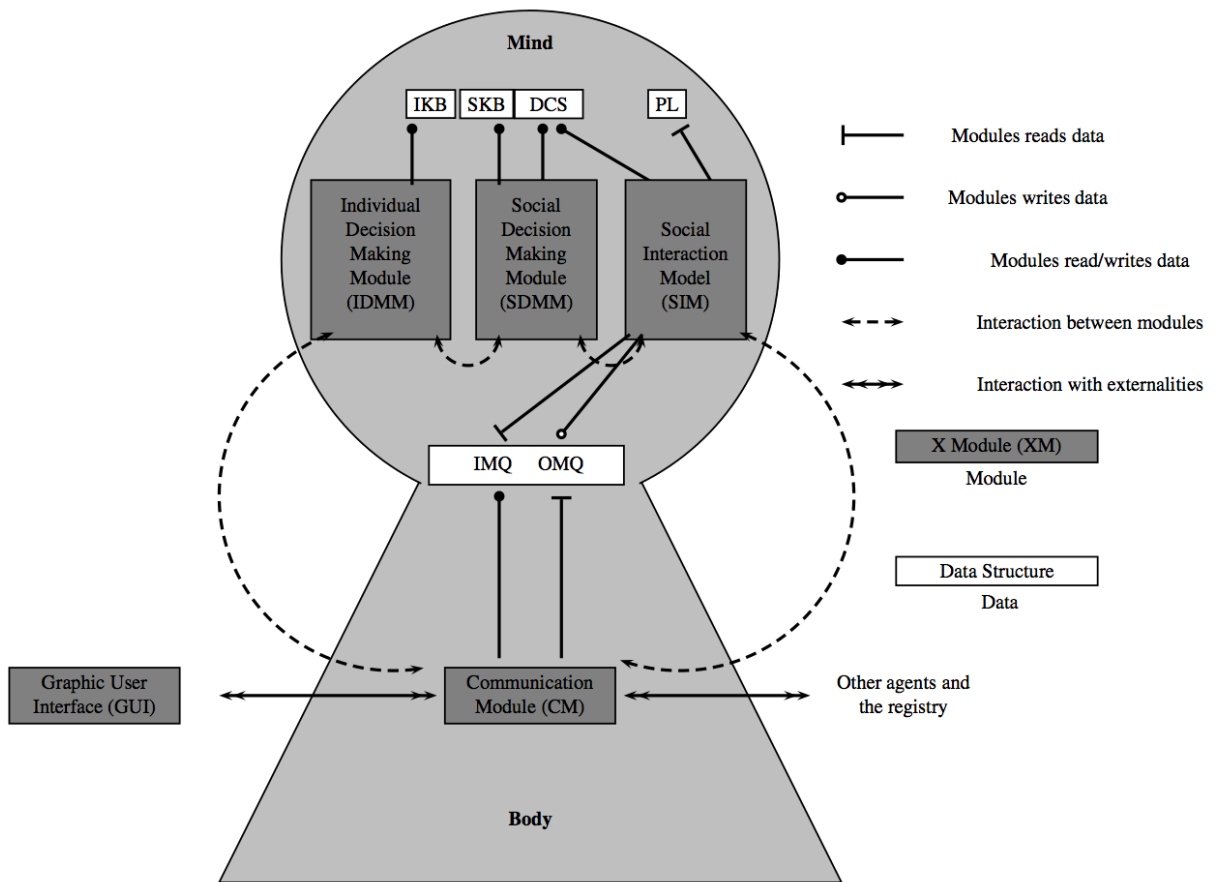


FIGURE 3.3 – L'architecture modulaire de notre agent.

3.3.1 Prise de décision individuelle

Le module de prise de décision individuelle (*Individual Decision Making Module*, IDMM) réalise le raisonnement à propos des services fournis ou demandés. L'IDMM s'appuie sur les structures de données concrètes dans la base de connaissance individuelle (*Individual Knowledge Base*, IKB). Les décisions sont prises selon les besoins ou les compétences de l'utilisateur à propos des services, des types de services possibles et des préférences/contraintes de l'utilisateur.

Les utilisateurs indiquent aux agents leurs exigences à travers l'interface graphique en dessinant des diagrammes d'influence (CLEMEN, 1996) où ils peuvent afficher la structure du problème de décision à propos des services fournis ou demandés. Les éléments des problèmes de décisions sont : des décisions à prendre, des croyances incertaines et des valeurs prônées. Ils sont représentés dans des diagrammes d'influence comme des nœuds de différentes formes. Les nœuds sont les suivants : les nœuds de décision (représentés par des rectangles), les nœuds de chance (représentés par des ovals) et les nœuds de valeur (représentés par des rectangles avec des coins arrondis). Les nœuds sont connectés dans un graphe par des arcs, chacun reliant un prédécesseur avec un successeur. Pour chaque nœud, les prédécesseurs sont indépendants et affectent le successeur. Les diagrammes d'influence sont construits sans cycle. Afin de modéliser une décision multi-critère, nous avons ajouté un autre type de nœud qui agrège les résultats des nœuds prédécesseurs. Un nœud de valeur abstraite est un nœud de valeur spéciale représenté par un rectangle à bord arrondi avec un bord double. Un nœud de valeur concrète est spécifié pour chaque combinaison de décisions et de croyances qui alimentent ce nœud. Par contraste, une valeur abstraite est spécifiée pour chaque combinaison de valeurs qui l'alimentent. Dans de nombreux cas, une décision est le résultat d'un compromis entre des attributs. Dans un tel cas, il est possible de représenter explicitement ces attributs multiples dans une hiérarchie de valeurs où une valeur abstraite supérieure agrège les valeurs concrètes inférieures. De plus, l'interface graphique permet de spécifier les préférences et contraintes de l'utilisateur sur les valeurs et les croyances. Par exemple, la figure 3.4 présente le diagramme d'influence pour l'évaluation d'un service par un agent de type A, e.g. A1. Le but principal supérieur (provision) est divisé en deux sous-buts abstraits indépendants concernant le coût (*cost*) et la qualité de données (*qos*). Ces sous-buts sont réduits à des sous-buts concrets. Par exemple, la qualité de service dépend de la qualité du service S_b (qos_b). Alors que les buts abstraits reflètent les besoins de l'utilisateur, les buts concrets fournissent des critères pour évaluer les différentes alternatives.

Le but principal, i.e. la composition effective du service S , est réalisé par des décisions, e.g. quels services concrets S_a et S_b choisir (en instanciant de manière appropriée les variables x et y ¹ dans la figure 3.4). On note que A1 est un candidat pour fournir le service S_a dans notre cas d'usage, mais il peut fournir plusieurs instances de ce service. En d'autres termes, x peut être instancié de différentes manières. À l'inverse, A1 n'est pas un fournisseur potentiel pour le service S_b et il a besoin d'en choisir un. Ces décisions dépendent de la connaissance de l'agent, c'est-à-dire les informations concernant les services concrets fournis par A1 lui-même et les agents de type B (e.g. *price*, *warranty*).

L'utilisateur indique également ses préférences et contraintes. Par exemple, l'utilisateur indique que le coût est plus important que la qualité de service. Les priorités sont attachées aux buts, aux décisions et aux connaissances dans le diagramme d'influence pour représenter les préférences entre les buts, les utilités espérées des décisions et l'incertitude des connaissances.

La structure du problème de décision relatif à l'évaluation des services et les priorités associées sont stockées dans la base de connaissances de l'agent, en particulier dans la base de connaissances individuelle (IKB). Le raisonnement est effectué par le module IDMM. Au sein du projet ArguGRID, le module IDMM a été réalisé à l'aide du cadre d'argumentation pour la prise de décision. Par exemple, la base de connaissances, qui correspond au diagramme d'influence de la figure 3.6, est représentée dans la figure 3.5. Les règles de but sont représentées en haut alors que les règles de décision sont en

1. Dans ce chapitre, nous adopterons la convention suivante : les variables sont en italique et les constantes sont en police de caractère courrier.

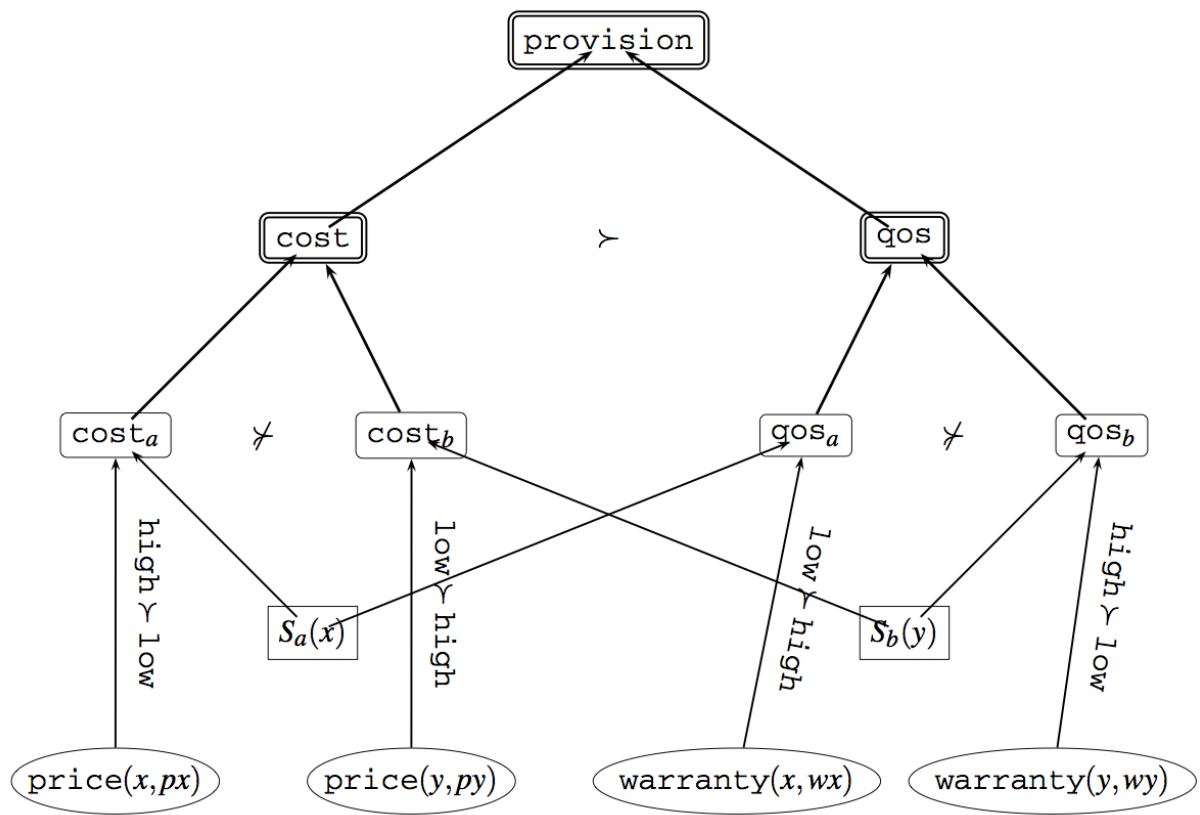


FIGURE 3.4 – Le diagramme d'influence pour structurer la décision.

bas. Dans cet exemple, il n'y a pas de règle épistémique. C'est la raison pour laquelle l'inclusion de présomptions sur les informations manquantes est cruciale pour réaliser le raisonnement individuel des agents demandeurs. Une règle au-dessus d'une autre a la priorité. Afin de simplifier la présentation, la théorie est subdivisée en couches disjointes, i.e. différents niveaux. Les règles *ex æquo* sont regroupées au sein d'un même niveau. Les règles incomparables sont arbitrairement assignées à un niveau. Les règles de but expriment le fait qu'atteindre les buts *cost* et *qos* sont dans l'idéal nécessaire pour atteindre le but *provision*, mais ceci peut être relâché : atteindre le but *cost* est suffisant pour atteindre *provision*. Contrairement aux autres règles du tableau 3.5, r_{01} n'est pas dans IKB ce qui reflète non pas l'exigence de l'utilisateur représenté par A1 mais r_{01} est le résultat de l'interaction précédente entre A1 et l'acheteur stocké dans le tableau d'engagement dialogique (*Dialogical Commitment Store*, DCS). r_{01} reflète la préférence du client représenté par l'agent acheteur.

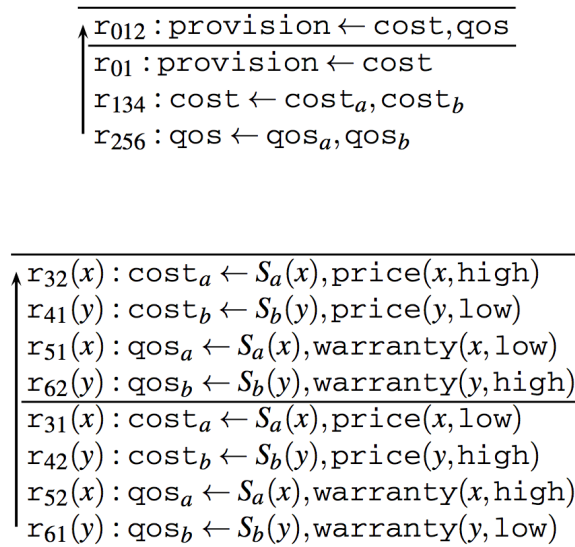


FIGURE 3.5 – Les règles de but (en haut) et les règles de décision (en bas) représentant les exigences des utilisateurs

Ces structures de données concrètes (règles et priorités) fournissent la colonne vertébrale des arguments. Par exemple, A1 peut construire un argument admissible concluant que le but relatif au coût du service S_b est atteint en sélectionnant $S_b(x)$ si on suppose que le prix du service $S_b(x)$ est faible.

Le module IDMM interagit avec les autres composants de l'architecture comme suit. Il interagit avec la GUI, à travers le module CM, qui utilise l'environnement GOLEM comme un médiateur pour interagir avec les autres entités du système et avec l'utilisateur. L'IDMM est informé et répond quand un service est (ou doit être) instancié. L'IDMM interagit avec le SDMM en demandant ou en fournissant l'instanciation du service abstrait ou partiellement instancié. De cette manière, l'IDMM traduit les buts et préférences de l'utilisateur en une représentation abstraite de services atomiques (ou de services composés, si nécessaire). Par exemple, A1 peut construire un argument admissible concluant que le but relatif à l'approvisionnement du service S est atteint si on suppose que le prix du service $S_b(y)$ est faible. Ceci est à nouveau traduit dans une représentation concrète (le choix de y satisfait la contrainte) par le module SDMM.

3.3.2 Prise de décision sociale

Le module de décision sociale (*Social Decision Making Module*, SDMM) raisonne sur les instances concrètes de services qui peuvent être demandées/fournies. Les décisions sont prises selon les exigences ou les compétences des utilisateurs, la connaissance des partenaires potentiels et des alternatives de services et les préférences les concernant. À cette intention, le module SDMM est soutenu par les structures de données concrètes dans la base de connaissance sociale (*Social Knowledge Base*, SKB) et le tableau d'engagement dialogique (*Dialogical Commitment Store*, DCS).

Le tableau d'engagement dialogique est une structure de données interne qui contient des propositions et des suggestions d'actions impliquant l'agent, celui-ci pouvant être le débiteur ou le créancier. Cette structure de données est partagée par les modules SIM et SDMM. Concrètement, le tableau d'engagement dialogique peut contenir des représentations concrètes de services atomiques ou composés fournis par l'agent. De plus, la SKB contient les préférences sur les services et les partenaires. La sélection (resp. la suggestion) de services concrets est réalisée selon les exigences (resp. les compétences) de l'utilisateur à propos des alternatives pour les services concrets, l'information à propos des partenaires et les préférences relatives.

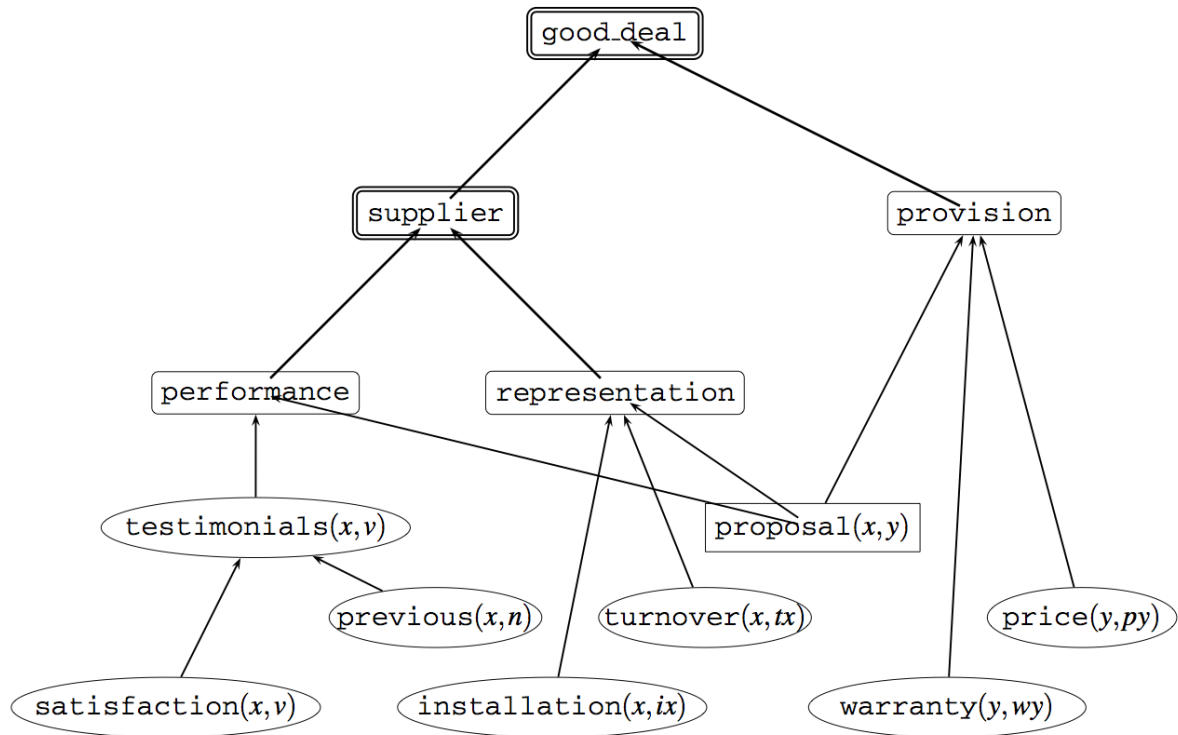


FIGURE 3.6 – Diagramme d'influence pour structurer la négociation.

La figure 3.6 représente le problème de négociation de S_b du point de vue d'un agent de type A, e.g. A1. L'évaluation du contrat (`good_deal`) dépend de l'approvisionnement du service (`provision`) comme considéré par le module IDMM et il dépend également du fournisseur (`supplier`). L'évaluation du partenaire dépend de sa représentation (`representation`) et de sa performance (`performance`). La performance du fournisseur est influencée par la connaissance à propos du témoignage des clients (`testimonials(x, v)`) dépendant du nombre de collaborations précédentes avec ce fournisseur (`previous(x, n)`) et la satisfaction moyenne à l'issue de ces collaborations (`satisfaction(x, v)`). Par

souci de simplicité, les préférences ne sont pas représentées dans la figure 3.6. Dans le projet ArguGRID, le module SDMM, comme le module IDMM, est construit sur le cadre d'argumentation pour la prise de décision. Les positions et les priorités sont stockées dans la base de connaissance sociale de l'agent (SKB) et dans le tableau des engagements dialogiques (DCS). Le raisonnement est réalisé par le module SDMM. De manière alternative, l'utilisateur peut directement alimenter le SKB et le DCS. Les positions correspondantes au diagramme d'influence de la figure 3.6 sont représentées dans la figure 3.7 sous la forme de règles de but, de règles épistémiques et de règles de décision. Par exemple, l'agent dont le diagramme d'influence est donné dans la figure 3.4 exige que les fournisseurs aient un chiffre d'affaire de plus de deux millions d'euros (turnover) et au moins cinquante installations (installation). À cette intention, les règles $r_{31}(x, y)$ et $r_{32}(x, y)$ sont incluses dans la SKB.

$r_{012} : \text{good_deal} \leftarrow \text{supplier, provision}$												
$r_{01} : \text{good_deal} \leftarrow \text{supplier}$												
$r_{134} : \text{supplier} \leftarrow \text{performance, representation}$												
$r_{256} : \text{provision} \leftarrow \text{cost}_b, \text{qos}_b$												
$r_{25} : \text{provision} \leftarrow \text{cost}_b$												
$r_{26} : \text{provision} \leftarrow \text{qos}_b$												
$r_{02} : \text{good_deal} \leftarrow \text{provision}$												
<table style="width: 100%; border-collapse: collapse;"> <tr><td style="padding: 2px;">$f_1 : \text{testimonials}(\text{Bob}, \text{high}) \leftarrow$</td></tr> <tr><td style="padding: 2px;">$f_2 : \text{turnover}(\text{Bob}, 5) \leftarrow$</td></tr> <tr><td style="padding: 2px;">$f_3 : \text{installation}(\text{Bob}, 100) \leftarrow$</td></tr> <tr><td style="padding: 2px;">$f_4 : \text{price}(d, \text{high}) \leftarrow$</td></tr> <tr><td style="padding: 2px;">$f_5 : \text{warranty}(d, \text{low}) \leftarrow$</td></tr> <tr><td style="padding: 2px;">$f_6 : \text{price}(c, \text{low}) \leftarrow$</td></tr> <tr><td style="padding: 2px;">$f_7 : \text{warranty}(c, \text{high}) \leftarrow$</td></tr> <tr><td style="padding: 2px;">$f_8 : \text{price}(e, \text{low}) \leftarrow$</td></tr> <tr><td style="padding: 2px;">$f_9 : \text{warranty}(e, \text{low}) \leftarrow$</td></tr> <tr><td style="padding: 2px;">$f_{10} : \text{price}(f, \text{high}) \leftarrow$</td></tr> <tr><td style="padding: 2px;">$f_{11} : \text{warranty}(f, \text{high}) \leftarrow$</td></tr> </table>		$f_1 : \text{testimonials}(\text{Bob}, \text{high}) \leftarrow$	$f_2 : \text{turnover}(\text{Bob}, 5) \leftarrow$	$f_3 : \text{installation}(\text{Bob}, 100) \leftarrow$	$f_4 : \text{price}(d, \text{high}) \leftarrow$	$f_5 : \text{warranty}(d, \text{low}) \leftarrow$	$f_6 : \text{price}(c, \text{low}) \leftarrow$	$f_7 : \text{warranty}(c, \text{high}) \leftarrow$	$f_8 : \text{price}(e, \text{low}) \leftarrow$	$f_9 : \text{warranty}(e, \text{low}) \leftarrow$	$f_{10} : \text{price}(f, \text{high}) \leftarrow$	$f_{11} : \text{warranty}(f, \text{high}) \leftarrow$
$f_1 : \text{testimonials}(\text{Bob}, \text{high}) \leftarrow$												
$f_2 : \text{turnover}(\text{Bob}, 5) \leftarrow$												
$f_3 : \text{installation}(\text{Bob}, 100) \leftarrow$												
$f_4 : \text{price}(d, \text{high}) \leftarrow$												
$f_5 : \text{warranty}(d, \text{low}) \leftarrow$												
$f_6 : \text{price}(c, \text{low}) \leftarrow$												
$f_7 : \text{warranty}(c, \text{high}) \leftarrow$												
$f_8 : \text{price}(e, \text{low}) \leftarrow$												
$f_9 : \text{warranty}(e, \text{low}) \leftarrow$												
$f_{10} : \text{price}(f, \text{high}) \leftarrow$												
$f_{11} : \text{warranty}(f, \text{high}) \leftarrow$												
<table style="width: 100%; border-collapse: collapse;"> <tr><td style="padding: 2px;">$r_{21}(x, y) : \text{performance} \leftarrow \text{proposal}(x, y), \text{testimonials}(x, \text{high})$</td></tr> <tr><td style="padding: 2px;">$r_{31}(x, y) : \text{representation} \leftarrow \text{proposal}(x, y), \text{turnover}(x, tx), tx > 2\text{M euros}$</td></tr> <tr><td style="padding: 2px;">$r_{32}(x, y) : \text{representation} \leftarrow \text{proposal}(x, y), \text{installation}(x, ix), ix > 50$</td></tr> <tr><td style="padding: 2px;">$r_{51}(x) : \text{cost}_b \leftarrow \text{proposal}(x, y), \text{price}(y, \text{low})$</td></tr> <tr><td style="padding: 2px;">$r_{61}(x) : \text{qos}_b \leftarrow \text{proposal}(x, y), \text{warranty}(y, \text{high})$</td></tr> </table>		$r_{21}(x, y) : \text{performance} \leftarrow \text{proposal}(x, y), \text{testimonials}(x, \text{high})$	$r_{31}(x, y) : \text{representation} \leftarrow \text{proposal}(x, y), \text{turnover}(x, tx), tx > 2\text{M euros}$	$r_{32}(x, y) : \text{representation} \leftarrow \text{proposal}(x, y), \text{installation}(x, ix), ix > 50$	$r_{51}(x) : \text{cost}_b \leftarrow \text{proposal}(x, y), \text{price}(y, \text{low})$	$r_{61}(x) : \text{qos}_b \leftarrow \text{proposal}(x, y), \text{warranty}(y, \text{high})$						
$r_{21}(x, y) : \text{performance} \leftarrow \text{proposal}(x, y), \text{testimonials}(x, \text{high})$												
$r_{31}(x, y) : \text{representation} \leftarrow \text{proposal}(x, y), \text{turnover}(x, tx), tx > 2\text{M euros}$												
$r_{32}(x, y) : \text{representation} \leftarrow \text{proposal}(x, y), \text{installation}(x, ix), ix > 50$												
$r_{51}(x) : \text{cost}_b \leftarrow \text{proposal}(x, y), \text{price}(y, \text{low})$												
$r_{61}(x) : \text{qos}_b \leftarrow \text{proposal}(x, y), \text{warranty}(y, \text{high})$												

FIGURE 3.7 – Les règles de but (en haut), les règles épistémiques (au centre) et les règles de décision (en bas) correspondantes aux positions sociales.

Le tableau dialogique de A1, qui inclut les positions publiques des agents dont A1 est conscient, inclut les suggestions impliquant A1 : soit A1 est le créancier de la suggestion, par exemple $\text{commit}(\text{Bob}, [\text{good_deal}, S_b(e), \emptyset])$ est ajouté au tableau quand Bob le suggère (voir la section suivante); ou A1 est le créancier par exemple $\text{commit}(A1, [\text{good_deal}, S_b(e), \emptyset])$ est ajouté au tableau quand A1 l'accepte (voir la section suivante).

Le module SDMM interagit avec l'IDMM en échangeant les services abstraits qui sont (ou doivent être) instanciés et en communiquant quand un service concret est (ou doit être) instancié. Le module SDMM interagit avec le module SIM (voir la section suivante) : i) en notifiant quand l'agent doit jouer un certain rôle dans un protocole en particulier; ii) en étant informé par le SIM quand des offres, des faits ou des arguments doivent être évalués ou construits; iii) en informant le SIM quand des offres, des faits ou des arguments sont évalués ou construits.

De cette manière, le SDMM raisonne et prend des décisions à propos des propositions et des

arguments qui sont échangés au cours du dialogue. Par exemple, A1 peut construire un argument admissible concluant que le but relatif au coût du service S_b est atteint car le prix du service $S_b(c)$ est faible. Cet argument est utile pour A1 afin de justifier ce choix, $S_b(c)$, auprès de Bob.

3.3.3 Interaction sociale

Le module d'interaction sociale (*Social Interaction Module*, SIM) gère les communications et interactions en conformité avec les protocoles. Ces protocoles sont des structures de données concrètes qui sont stockées dans la bibliothèque de protocoles (*Protocol Library*, PL).

Dans le projet ArguGRID project, nous utilisons LCC (ROBERTSON, 2004) (*Lightweight Coordination Calculus*) qui est un langage déclaratif de programmation logique à la Prolog pour le calcul des communications. LCC permet de gérer des interactions sociales via des protocoles. Ainsi, un protocole est un ensemble de clauses décrivant les actions communicatives que peut réaliser l'agent en fonction de son rôle. Une action consiste à émettre ou à réceptionner des messages, voire à adopter un rôle. Les actions peuvent être réalisées séquentiellement, simultanément ou faire l'objet d'une décision. Chaque action est réalisée si les pré-conditions sont valides et elle peut déclencher des post-conditions.

Les décisions nécessitent de mener des interactions mises en œuvre par le module SDMM. En pratique, le module SDMM, une fois réalisé le raisonnement à propos des exigences sur les services et les partenaires, utilise un mécanisme de *boot strap* qui initialise le protocole nécessaire, le rôle joué par l'agent dans ce protocole et le rôle joué par les autres participants. À partir de là, le moteur de protocole du module SIM détermine les messages devant être envoyés étant donné les rôles. S'il n'existe pas d'ambiguïté sur le message devant être envoyé, le module SIM envoie automatiquement le message dans la file des messages sortants. Quand une décision doit être prise soit en ce qui concerne le choix d'une locution (e.g. accepter ou rejeter une offre) soit pour instancier le contenu d'une locution (e.g. établir une proposition), le module SIM utilise un mécanisme de pré-condition pour déclencher le module SDMM afin d'obtenir une solution. En fonction de la satisfaction des pré-conditions, le module SIM envoie la locution dans la file des messages sortants. S'il est nécessaire de mettre à jour le tableau d'engagement dialogique de l'agent, ceci est fait via le mécanisme de post-condition. Concrètement, MARGO s'interface avec LCC à travers les conditions d'énonciation des messages.

Les agents transmettent des messages pour échanger des buts, des décisions et des connaissances. La syntaxe des messages se conforme à un langage de communication d'agent. Dans nos scénarios, la locution est l'une des suivantes : *question*, *assert*, *accept*, *why*, *withdraw* (voir tableau 3.1 pour des exemples). Le contenu est un triplet qui consiste en : un but G_k , une décision D_k , et une connaissance K_k . Nous noterons θ le fait qu'aucun but n'est donné et \emptyset pour noter qu'aucune connaissance n'est fournie.

Dans notre scénario, nos agents utilisent deux protocoles pour deux types de dialogue : la demande d'information et la négociation. La figure 3.8 représente le protocole de négociation du point de vue du demandeur de service à l'aide d'un automate à états finis. Le module SDMM s'interface avec le protocole du SIM via les mécanismes d'énonciation des messages. Par exemple, lors d'un dialogue, un demandeur peut émettre *accept*, *assert* et *why*. Le choix de cette locution dépend du SDMM qui doit satisfaire les pré-conditions de l'une de ces locutions.

Le haut du tableau 3.1 montre les actes de langage échangés entre A1 et Bob jouant un dialogue de demande d'information. Selon le protocole de ce type de dialogue, le premier coup dialogique consiste pour A1 à poser une question à Bob, M_0 . Cette locution recherche l'intervalle de prix pour les services S_b disponibles. Bob informe A1 avec plusieurs locutions concernant les différents services disponibles et leur prix (M_1, \dots, M_4). Une demande d'information similaire a lieu entre Bob et A1 pour informer ce dernier des garanties concernant les différents services S_b disponibles. Selon cette connaissance, l'agent A1 de type A est capable de considérer lesquels de ces services pourraient satisfaire ses buts.

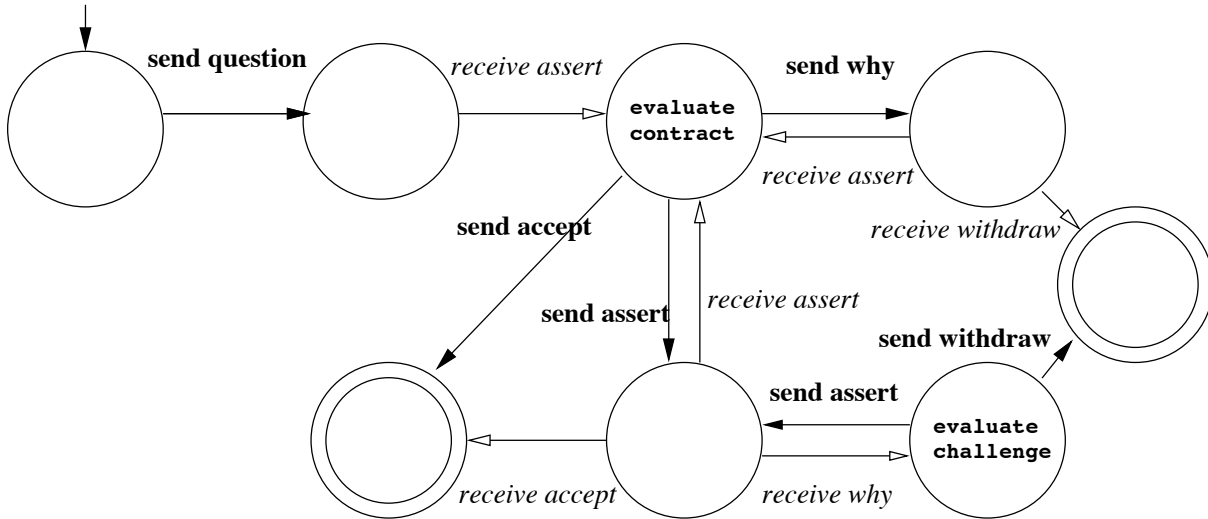


FIGURE 3.8 – Le protocole de négociation pour le demandeur

Le centre du tableau 3.1 représente les actes de langage échangés entre A1 et Alice jouant un dialogue de demande d'information. Le premier coup dialogique consiste pour A1 à poser une question à Alice, M_0 . Cette locution demande un témoignage de Alice à propos de Bob. Ce témoignage est high (cf M_1) et soutenu par le nombre d'expériences précédentes et leurs valeurs.

Le bas du tableau 3.1 représente les actes de langage échangés entre A1 et Bob jouant un dialogue de négociation. Ils tentent d'atteindre un accord sur le contrat pour l'approvisionnement d'un service S_b pour atteindre le but commun `good_deal`. Un contrat est un tuple $\langle cid, debtor, creditor, service, terms \rangle$ où `cid` est l'identifiant du contrat, `debtor` est l'agent fournissant le service et `creditor` est l'agent demandant le service. Un contrat concerne l'approvisionnement d'un service si la liste des termes et des conditions (denoted `terms`) sont satisfaits. Avec le message M_1 , Bob informe A1 qu'il trouve que les termes et les conditions du contrat pour l'approvisionnement du service $S_b(d)$ sont justifiés par le but commun (`good_deal`). Toutefois, A1 ne trouve pas que $S_b(d)$ est justifié et il propose $S_b(c)$. Comme aucune de ces propositions n'est mutuellement acceptable, elles ne sont plus considérées dans la suite de la négociation. Bob essaye de déterminer les raisons du choix de A1 (cf M_3) qui est le coût (plutôt que la qualité de service). Étant donné la réponse de A1 dans M_4 , Bob adopte le but fourni par A1. En conséquence, il trouve, parmi les solutions restantes ($S_b(e)$ et $S_b(f)$), celle préférée par A1 ($S_b(e)$) et la suggère (M_5). Finalement, A1 communique cet accord à l'aide de `accept` (M_6) qui termine le dialogue.

Afin de simplifier l'évaluation du processus et de l'accord atteint, nous avons considéré dans (MORGE & MANCARELLA, 2014) une version simplifiée du protocole sans échange d'argument. Alors que l'un des agents débute en réalisant une première offre, son interlocuteur répond avec une contre-offre. Dans la suite du processus, chaque agent doit adopter l'une des attitudes suivantes : i) soit il campe sur ses positions, i.e. il répète la proposition précédente ; ii) soit il concède, i.e. il réduit ses prétentions en abandonnant sa précédente offre en en effectuant une nouvelle. Afin d'articuler ces deux attitudes, mener une négociation consiste à adopter une stratégie de concession minimale qui s'appuie sur le principe de réciprocité. Si l'interlocuteur campe sur ses positions, l'agent en fait de même. Dès que l'interlocuteur fait une concession, l'agent en fait de même. On peut noter que la troisième étape de la négociation a un statut particulier dans la mesure où l'agent qui a la parole doit concéder. Si l'agent ne peut pas concéder (e.g. il n'y a pas d'autre service qui satisfait ces contraintes), alors l'agent campe sur ses positions. Si une offre acceptable a été émise, alors l'agent l'accepte. Quand un agent ne peut plus concéder, il met un terme à la négociation.

Comme notre langage est fini et que le nombre de décisions possibles l'est également, la négocia-

M_k	S_k	H_k	A_k	R_k
M_0	Al	Bob	question($\theta, S_b(x), [\text{price}_b(x, px)]$)	θ
M_1	Bob	Al	assert($\theta, S_b(c), [\text{price}_b(c, pc), \text{low} \leq pc \leq \text{medium}]$)	M_0
M_2	Bob	Al	assert($\theta, S_b(e), [\text{price}_b(e, pe), \text{low} \leq pe \leq \text{medium}]$)	M_0
M_3	Bob	Al	assert($\theta, S_b(d), [\text{price}_b(d, pd), \text{medium} \leq pd \leq \text{high}]$)	M_0
M_4	Bob	Al	assert($\theta, S_b(f), [\text{price}_b(f, pf), \text{medium} \leq pf \leq \text{high}]$)	M_0

M_k	S_k	H_k	A_k	R_k
M_0	Al	Alice	question(performance, proposal(Bob, y), [testimonials(Bob, z)])	θ
M_1	Alice	Al	assert(performance, proposal(Bob, y), [testimonials(Bob, high)])	M_0
M_2	Al	Alice	why(θ , proposal(Bob, y), [testimonials(Bob, high)])	M_1
M_3	Alice	Al	assert(θ , proposal(Bob, y), [previous(Bob, l0), satisfaction(Bob, high)])	M_2

M_k	S_k	H_k	A_k	R_k
M_0	Al	Bob	question(good_deal, $\langle cid, \text{Bob}, \text{Al}, S_b(x), [\text{price}(x, px), \text{warranty}(x, wx)] \rangle, \emptyset$)	θ
M_1	Bob	Al	assert(good_deal, $\langle c1, \text{Bob}, \text{Al}, S_b(d), [\text{price}(d, \text{high}), \text{warranty}(d, \text{low})] \rangle, \emptyset$)	M_0
M_2	Al	Bob	assert(good_deal, $\langle c2, \text{Bob}, \text{Al}, S_b(c), [\text{price}(c, \text{low}), \text{warranty}(c, \text{high})] \rangle, \emptyset$)	M_1
M_3	Bob	Al	why(good_deal, $\langle c2, \text{Bob}, \text{Al}, S_b(c), [\text{price}(c, \text{low}), \text{warranty}(c, \text{high})] \rangle, \emptyset$)	M_2
M_4	Al	Bob	assert(cost _{Al} , $\langle c2, \text{Bob}, \text{Al}, S_b(c), [\text{price}(c, \text{low}), \text{warranty}(c, \text{high})] \rangle, \emptyset$)	M_3
M_5	Bob	Al	assert(good_deal, $\langle c3, \text{Bob}, \text{Al}, S_b(e), [\text{price}(e, \text{low}), \text{warranty}(e, \text{low})] \rangle, \emptyset$)	M_1
M_6	Al	Bob	accept(good_deal, $\langle c3, \text{Bob}, \text{Al}, S_b(e), [\text{price}(e, \text{low}), \text{warranty}(e, \text{low})] \rangle, \emptyset$)	M_5

TABLE 3.1 – Les dialogues de demande d’information (en haut et au centre) et le dialogue de négociation (en bas).

tion est finie. Comme notre langage est fini et de par la stratégie de concession minimale, il n’est pas difficile de constater que les négociations sont couronnées de succès si un accord potentiel existe. Comme un agent concède nécessairement au cours du processus même si son interlocuteur campe sur ses positions, la négociation entre deux agents adoptant la stratégie de concession minimale parcourt la totalité de l’espace des accords potentiels. En d’autres termes, cette stratégie permet d’atteindre systématiquement un accord. Toutefois, on peut remarquer que cette stratégie ne permet pas d’atteindre un équilibre de Nash en stratégie pure. Il n’est pas dans l’intérêt individuel d’un agent d’adopter cette stratégie.

L’accord obtenu par la négociation est Pareto-optimal s’il n’est pas possible d’améliorer strictement la satisfaction d’un agent sans réduire celle de son interlocuteur. C’est le cas dans notre processus de négociation. L’accord est Pareto-optimal car les concessions sont minimales.

3.4 Scénario nominal

Nous considérons ici un scénario nominal qui illustre la délibération des agents et la communication dans le cas d’usage de la section 3.2. Ce scénario nominal implique Al, un agent de type A, et Bob, un agent de type B, négociant l’approvisionnement d’un service S_b .

Les termes et conditions considérés pour l’évaluation du contrat à propos de S_b au cours de la négociation (cf bas de la table 3.1) sont représentés sur les axes des abscisses et des ordonnées de la figure 3.9. Les aires d’acceptabilité des deux participants sont représentées en gris et dépendent du prix (axe des ordonnées) et de la garantie (axe des abscisses). Quatre points correspondent à des combinaisons de valeurs : $S_b(c)$ où la garantie est élevée et le prix est faible, $S_b(d)$ où la garantie est faible et le prix est élevé, $S_b(e)$ où la garantie et le prix sont faibles et $S_b(f)$ où ils sont élevés. Après le message M_2 (cf gauche de la figure 3.1), Bob trouve seulement $S_b(d)$ justifié et Al trouve seulement $S_b(c)$ justifié. Après le message M_3 (cf centre de la figure 3.9), les aires d’acceptabilité des deux agents se sont déplacées car ni $S_b(c)$ ni $S_b(d)$ n’ont été mutuellement acceptés. Les deux agents font des concessions comme le permet notre cadre d’argumentation pour la prise de décision. Après le

message M_6 (cf droite de la figure 3.9), les deux agents ont identifié $S_b(e)$ comme étant une solution commune. On peut noter que l'influence de A1 sur Bob permet d'éviter d'explorer l'alternative $S_b(f)$ qui n'est pas justifiée du point de vue de A1. L'influence de A1 sur Bob est soutenue par les informations supplémentaires portées par l'argument.

Nous avons illustré ici la principale plus-value de la négociation à base d'argumentation qui est, comme signalé dans (RAHWAN et al., 2004), de permettre aux agents de s'influencer mutuellement. De plus, nos agent font des concessions lorsque c'est nécessaire comme suggéré par AMGOUD et al., 2007.

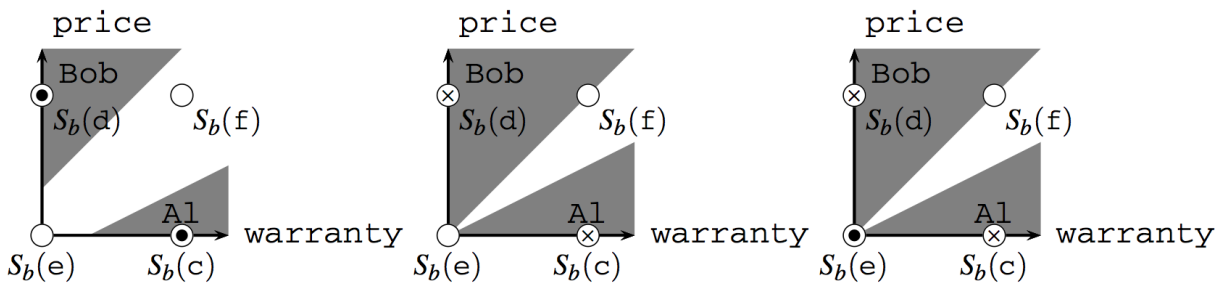


FIGURE 3.9 – Les aires d'acceptabilité des participants après les messages M_2 (gauche), M_3 (centre) et M_6 (droite).

3.5 Conclusion

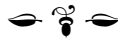
Dans ce chapitre, nous avons décrit un modèle et une architecture d'agent utilisant l'argumentation pour la sélection automatique de services et de partenaires. La conception modulaire du processus délibératif et communicationnel apporte l'avantage bien connu en ingénierie d'une approche modulaire et permet de créer un modèle d'agent unique. Les trois modules internes sont respectivement dédiés à la prise de décision (pour l'IDMM), à la négociation (pour le SDMM), et à la communication (pour le SIM). Ils utilisent un cadre d'argumentation. Si l'IDMM peut être considéré comme le module de raisonnement de l'agent pour atteindre ses buts individuels, le SDMM peut être considéré comme le module de raisonnement social, conscient des concessions nécessaires à la résolution collaborative des buts, et le SIM peut être considéré comme la conscience sociale filtrant ces motivations en suivant les normes sociales de la société d'agents. Les normes sociales et les règles des dialogues sont encodées comme des protocoles d'interaction. Afin de tester notre approche, l'architecture a été instanciée dans le projet ArguGRID, au moyen du moteur argumentatif pour la prise de décision MARGO (pour les deux premiers modules) et avec l'outil LCC pour faire respecter les protocoles (pour le dernier module).

Nous nous sommes intéressés à la découverte automatique de services par les agents et nous montrons comment la sélection de ces services peut être réalisée concrètement à l'aide d'une architecture d'agent argumentatif qui interprète les spécifications abstraites des buts de l'utilisateur. La plus-value de l'argumentation réside dans les informations supplémentaires qui soutiennent le processus de sélection d'un service, ie. des positions qui justifient pourquoi un service doit être sélectionné plutôt qu'un autre.

Nous avons proposé une architecture générale qui distingue le raisonnement interne, le raisonnement social et la communication. Cette architecture a été instanciée à l'aide des techniques argumentative et logique. MARGO (MORGE & MANCARELLA, 2006), i.e. l'implémentation de notre cadre d'argumentation, fournit une représentation des connaissances, buts et décisions sans capacité de

planification comme requis par notre application. MARGO utilise l'implémentation de DUNG, MANCARELLA et TONI (2007) dans le système CaSAPI (GARTNER & TONI, 2007). Nous utilisons LCC pour représenter et déclencher les protocoles afin de permettre aux normes sociales utilisées par les agents d'être vérifiables, communicables, inspectables et potentiellement modifiables.

Comme mentionné précédemment, les contrats créent des organisations virtuelles constituées de débiteurs et de créditeurs. Dans ce chapitre, nous nous sommes focalisés sur le modèle et l'architecture d'agents et nous avons passé sous silence le problème des organisations virtuelles. Un modèle formel d'organisations virtuelles utilisant la négociation multi-agents pour sa configuration est décrite dans (MCGINNIS, STATHIS & TONI, 2011). La capacité de prise de décision de nos agents est utile au cours des opérations réalisées par les organisations virtuelles.



CHAPITRE 4

Négociation multi-latérale

Sommaire

4.1	Introduction	39
4.2	Modélisation des préférences	40
4.3	Accord équitable	41
4.4	Négociation	42
4.5	Évaluation de la négociation	44
4.6	Scénario	44
4.7	Résolution	45
4.8	Implémentation	46
4.9	Évaluation de la résolution	47
4.10	Conclusion	48

Les travaux présentés dans ce chapitre ont été réalisés entre 2011 et 2015 avec Jean-Christophe ROUTIER (Université de Lille) avec lequel j'ai co-encadré le stage de recherche et la thèse de Fabien DELECROIX ainsi que le projet de recherche de Thomas NACHTERGAELE à l'Université de Lille. Ces travaux ont notamment été publiés dans l'article suivant :

- DELECROIX, F., MORGE, M., NACHTERGAELE, T. & ROUTIER, J.-C. (2016). Multi-party negotiation with preferences rather than utilities. *Multiagent and Grid Systems – An International Journal*, 12(2), 139-165. h-index=16, Q2 (Scimago)

De plus, ces travaux ont été couronnés par le troisième prix de la meilleure démonstration décerné par IBM à la conférence PAAMS en 2014 (DELECROIX, MORGE & ROUTIER, 2014).

4.1 Introduction

Nous nous intéressons ici à la prise de décision collective. Dans ce contexte, un ensemble d'agents sélectionne conjointement un candidat, un plan d'action ou un ensemble de buts mutuellement acceptables qu'il considère plus ou moins préférables.

Comme nous l'avons vu dans le chapitre 2, les occasions de complexifier un système de négociation sont multiples. Nous appliquons ici le principe de parcimonie en étudiant la négociation multi-agents mono-attribut à base de propositions. Contrairement à la plupart des travaux évoqués dans le chapitre 2, nous ne considérons pas que les agents sont munis d'une fonction d'utilité. À l'inverse, nous considérons un modèle de préférences ordinales et qualitatives. Ainsi, les agents sont munis de relations de préférence. Cette hypothèse nous permet d'éviter les comparaisons interpersonnelles

d'utilité qui peuvent être à l'origine de biais dans l'agrégation des préférences. Plus précisément, nous utilisons des relations de préférences incomplètes, c.-à.d. des relations de pré-ordre pour modéliser les préférences individuelles des agents. De cette manière, nous considérons que :

- les agents ne sont pas toujours en mesure de comparer deux alternatives. Comme les relations de préférences sont incomplètes, un agent peut être indifférent vis-à-vis de ces alternatives ou il peut les trouver incomparables par manque d'information ou à cause de préférences conflictuelles;
- les alternatives ne peuvent pas être remplacées par des valeurs numériques qui servent à la négociation car des préférences incomplètes n'admettent pas de représentation sous la forme de fonctions d'utilité classiques.

En évitant ainsi des hypothèses restrictives sur la modélisation des préférences, nous permettons une forte expressivité pour les préférences des agents et nous proposons donc un modèle générique.

Dans ce chapitre, nous raffinons la règle de Pareto en considérant le leximax-optimal afin d'évaluer l'équité des accords. De plus, nous proposons un jeu de négociation multi-latérale, i.e. un protocole, un critère d'acceptabilité et deux stratégies : l'une coopérative et l'autre temporisatrice. Comme les alternatives peuvent être indifférentes ou incomparables, notre protocole permet à chaque agent de soumettre plus d'une alternative par tour. De cette manière, nous garantissons que, si les agents sont coopératifs, la négociation aboutit à un accord équitable. Nous verrons que si un agent temporise, son comportement est dans son intérêt individuel mais en conflit avec l'intérêt collectif. Toutefois, nos expérimentations montrent que l'avantage qu'un agent peut tirer de la stratégie temporisatrice n'est pas systématique. En particulier, cette stratégie est relativement inefficace si tous les opposants temporisent.

Nous abordons ici un problème de résolution distribuée : la recherche d'un point de rendez-vous dans un labyrinthe. Deux ou plusieurs agents sont parachutés dans un labyrinthe qui leur est inconnu. Leur but est alors de se retrouver le plus rapidement possible. Ils communiquent donc pour fixer un point de rendez-vous. Les agents négocient en échangeant des offres, c'est-à-dire, des cellules potentielles où se rencontrer. Nous montrons ici que nos simulations convergent toujours vers un point de rendez-vous. Nos expérimentations montrent que la négociation ne réduit pas l'effort total déployé par l'ensemble des agents, par contre, elle permet de partager cet effort équitablement entre les agents et donc de résoudre rapidement le problème distribué. Nous étudions également l'effet du partage d'informations lors de ces résolutions.

Le chapitre est organisé de la manière suivante. Dans la section 4.2, nous présentons notre modélisation des préférences. Ensuite, nous définissons dans la section 4.3 ce qu'est un accord équitable. Dans la section 4.4, nous exposons notre processus de négociation : le protocole, le critère d'acceptabilité et les stratégies. Dans la section 4.5, nous décrivons nos expérimentations concernant la négociation. La section 4.6 explicite les détails de notre scénario applicatif. Nous adaptons notre jeu de négociation à ce scénario dans la section 4.7. Nous présentons brièvement notre implémentation (cf section 4.8). Puis, nous présentons nos expérimentations concernant la résolution distribuée de problème dans la section 4.9. Finalement, nous tirons quelques conclusions puis nous dressons quelques perspectives (cf section 4.10).

4.2 Modélisation des préférences

Dans de nombreuses situations, un agent 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. Une relation de préférence large, qui est réflexive et transitive (i.e. un pré-ordre), permet d'exprimer l'équivalence qui indique l'indifférence de l'agent entre deux alternatives. Elle signifie que l'agent estime que, au sens des préférences, il n'existe pas de différence réelle entre x et y . Une relation de

préférence large est dite totale si pour toute paire d'éléments, elle établit au moins une association entre eux. Au contraire, nous considérons ici que la relation est dite partielle. Ainsi, certaines alternatives ne sont pas comparables. On peut interpréter l'incomparabilité comme le refus du décideur de s'engager sur un jugement incertain. On remarque que contrairement à la relation d'équivalence, la relation d'incomparabilité n'est pas transitive.

Comme la relation de préférence est partielle, il n'existe pas nécessairement d'alternative optimale. Pour sélectionner les meilleures alternatives, on définit la notion de non-dominance pour distinguer, dans un ensemble d'alternatives, celles auxquelles un agent ne préfère aucune autre. On peut remarquer qu'il existe toujours au moins une alternative non dominée. De plus, toute alternative optimale est aussi non-dominée alors que l'inverse n'est pas nécessairement vrai.

Exemple 1 Nous considérons ici \succsim_1 , \succsim_2 et \succsim_3 les relations de préférences de 3 agents sur un ensemble d'alternatives $\{a, b, c, d, e, f\}$. On peut représenter une relation de préférence par un graphe orienté où les sommets sont les alternatives (cf figure 4.1). Le fait que x est au moins aussi bonne que y est représenté par un chemin depuis x vers y . On peut remarquer que les alternatives non-dominées pour chacun des agents sont $\{a, b, c\}$, $\{c\}$ et $\{a, b\}$, respectivement. L'ensemble des alternatives potentiellement Pareto-optimales est $\{a, b, c, d, e\}$.

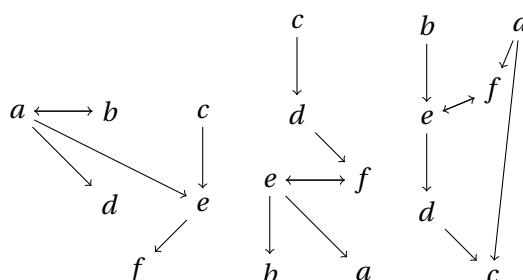


FIGURE 4.1 – Exemples de relations de préférence \succsim_1 (à gauche), \succsim_2 (au milieu) et \succsim_3 (à droite).

4.3 Accord équitable

Nous avons étudié différentes métriques pour évaluer la qualité d'un accord dans la section 2.3. Notre objectif est ici d'identifier les alternatives qui sont équitables à partir de préférences incomplètes (larges et partielles).

Comme dans (GOODMAN & MARKOWITZ, 1952), nous considérons ici qu'un agent dispose de plusieurs niveaux de discriminations. Ainsi, nous pouvons réintroduire une valeur numérique en mesurant la distance entre deux alternatives, i.e. les efforts concédés par l'agent, via le nombre de niveaux qui les séparent. Le rang d'une alternative correspond à sa profondeur dans le graphe des préférences en partant des alternatives non-dominées. Une alternative est d'autant plus satisfaisante, qu'elle a un rang minimal. Par convention, le rang d'une alternative non-dominée est 1. Le calcul du rang des autres alternatives s'effectue récursivement. On peut noter que, contrairement à GOODMAN et MARKOWITZ (1952), les relations de préférences considérées sont partielles. Toutefois, comme dans (GOODMAN & MARKOWITZ, 1952), notre métrique rompt avec l'hypothèse d'indépendance aux alternatives non pertinentes. Nous considérons ici que toutes les alternatives sont pertinentes. En d'autres termes, nous supposons ici que le problème de décision est bien posé.

Afin d'évaluer la qualité des alternatives du point de vue de la société d'agents, nous définissons le rang social d'une alternative en utilisant les fonctions d'agrégation de la Théorie du Choix Social.

Une alternative est d'autant plus satisfaisante que son rang social est minimal. Alors que l'agrégateur utilitaire mesure la moyenne des rangs, l'agrégateur égalitaire considère l'agent le plus mal loti, i.e. celui dont le rang est maximal.

En visant à minimiser le nombre d'efforts de l'agent en ayant réalisé le plus, le rang égalitaire est plus équitable. Cependant, comme nous l'avons vu dans notre exemple, il a l'inconvénient de ne pas garantir la Pareto-optimalité.

Afin d'identifier des alternatives à la fois équitables et Pareto-optimales, nous raffinons la règle 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*). C'est donc la règle du *leximax* que nous adoptons ici. On introduit alors un pré-ordre total et donc cette relation permet la comparaison entre n'importe quelle paire d'alternatives afin d'établir un classement. Ainsi, on peut identifier les alternatives aux efforts répartis de manière *leximax*-optimale que nous qualifions d'équitables.

L'ensemble des alternatives équitables est une restriction de l'ensemble des alternatives potentiellement Pareto-optimales. Il existe toujours au moins une alternative équitable. Toutes les alternatives équitables sont potentiellement Pareto-optimales mais l'inverse n'est pas nécessairement vrai.

En résumé, il est raisonnable de vouloir atteindre un accord équitable car son existence est garantie. Intuitivement, un tel accord a pour but de répartir de manière équilibrée les efforts des agents. De plus, cette règle est Pareto-inclusive. Toutefois, elle nécessite d'abandonner l'hypothèse d'indépendance aux alternatives non-pertinentes.

Exemple 2 Le tableau 4.1 indique le rang des alternatives pour chacun des agents dont les préférences sont représentées dans la figure 4.1 ainsi que le rang social des alternatives. Dans notre exemple, les

x	$r(x, \succ_1)$	$r(x, \succ_2)$	$r(x, \succ_3)$	$r_u(x)$	$r_e(x)$
a	1	4	1	2	4
b	1	4	1	2	4
c	1	1	4	2	4
d	2	2	3	$7/3$	3
e	2	3	2	$7/3$	3
f	3	3	2	$8/3$	3

TABLE 4.1 – Exemple de rangs des alternatives pour chaque agent.

alternatives qui minimisent le rang utilitaire sont a , b et c , alors qu'elles constituent dans le même temps les alternatives de plus mauvais rang pour un des trois agents. d , e et f minimisent le rang égalitaire. Cependant, parmi ces alternatives égalitaires, f n'est pas Pareto-optimal car f est Pareto-dominée par e .

4.4 Négociation

Afin d'établir un processus décentralisé atteignant un accord équitable, nous proposons un jeu de négociation multilatérale.

Notre protocole détermine les règles de la confrontation (ROSENSCHEIN & ZLOTKIN, 1994). Afin de ne pas prioriser l'initiateur de la négociation nous proposons ici un protocole symétrique. De plus, celui-ci permet à un agent de soumettre plusieurs offres à chaque tour. Ainsi, l'agent n'a pas à trancher entre certaines alternatives pour déterminer celle qui est prioritaire.

Un agent considère acceptable les alternatives qui ont été proposées par tous les agents. De plus, il privilégie les alternatives qui apparaissent au plus tôt dans l'historique. Pour que notre algorithme

implémente ce critère d'acceptabilité, un agent calcule quand au moins une alternative a été proposée par tous les agents. L'agent calcul alors l'ensemble leximax-optimal correspondant et les accepte. Sinon, l'agent fait une proposition en fonction de sa stratégie. 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.

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.

Dans le cas où un agent vise à trouver les alternatives équitables, la **stratégie coopérative** à suivre est simple. Il s'agit, pour un agent, 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.

Considérons un agent égoïste 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. La **stratégie temporisatrice** se veut meilleure que la stratégie coopérative dans cette perspective. Comme la stratégie coopérative, cette stratégie doit être légale : les alternatives ne sont pas répétées dans ses propositions. 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. 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.

En se restreignant aux alternatives non-dominées parmi les restantes, un agent minimise à chaque étape le niveau de profondeur dans le graphe de préférence. Autrement dit, à chaque étape, l'agent ne s'autorise à jouer que des alternatives qui minimisent le rang parmi les restantes. Afin de choisir parmi les alternatives rationnelles, l'agent opte pour celle(s) dont le rang est minimal. 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.

En résumé, la stratégie temporisatrice résulte de la combinaison de la stratégie de proposition et du choix d'une unique alternative dont le rang est minimal.

Tout d'abord, on peut remarquer que la négociation se conclut systématiquement sur un accord. Comme les agents peuvent adopter différentes stratégies, la négociation est paramétrée par le profil de stratégies des agents. Les accords atteints dans un jeu de négociation entre des agents coopératifs sont des alternatives équitables et réciproquement. Si au moins un des agents adopte la stratégie temporisatrice, l'accord obtenu n'est plus forcément Pareto-optimal.

Exemple 3 *Considérons les préférences de la figure 4.1. À gauche du tableau 4.2, nous représentons la trace d'une négociation où les trois agents adoptent une stratégie coopérative. Au tour $k = 1$, 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. L'ensemble des accords est alors composé des alternatives équitables d et e dont les vecteurs de tours sont *leximax*-optimaux.*

À droite du tableau 4.2, 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 précédemment. Le troisième agent quant à lui tempore 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

k	C_1	C_2	C_3	k	C_1	C_2	T_3
1	{a, b, c}	{c}	{a, b}	1	{a, b, c}	{c}	{a} ou {b}
2	{d, e}	{d}	{e, f}	2	{d, e}	{d}	{b} ou {a}
3	{f}	{e, f}	{d}	3	{f}	{e, f}	{e} ou {f}
{d, e}				{e} ou {f}			

TABLE 4.2 – Exemple de trace de négociation où les trois agents adoptent la stratégie coopérative (à gauche) et où les deux premiers agents adoptent la stratégie coopérative et le troisième la stratégie temporisatrice (à droite)

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 la négociation précédente. 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.

Il est possible de démontrer l'intérêt de la stratégie temporisatrice individuellement. Si nous fixons les stratégies des autres participants ainsi que la fonction de tirage aléatoire, 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.

4.5 Évaluation de la négociation

Nos expérimentations ont pour but d'établir :

- 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.

Après avoir implémenté avec le langage *Java* le protocole de négociation, le critère d'acceptabilité et les deux stratégies, nous avons effectué des expérimentations.

En résumé, nos résultats montrent que temporiser est une stratégie dominante dans des conditions assez restreintes. 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 devient limitée. 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 avons observé 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.

4.6 Scénario

Nous considérons n agents vus comme des parachutistes atterrissant dans un labyrinthe. Cet environnement est représenté par un graphe où chaque sommet est une cellule et il existe une arête

entre deux sommets si les cellules correspondantes sont adjacentes et non séparées par un mur. Dans notre scénario, chaque agent a une perception locale de l'environnement, c'est-à-dire qu'il découvre les murs au cours de son exploration. Initialement, un agent connaît la topologie de l'environnement mais il ne détient aucune connaissance sur les murs, i.e. sur l'absence d'une arête entre deux sommets adjacents. Un agent voit les murs dans son champ de vision.

En se basant sur sa perception locale de l'environnement, un agent est capable d'estimer la distance entre lui et n'importe quelle cellule atteignable. Cette distance est la longueur du plus court chemin entre la position courante de l'agent et la cellule. À chaque pas de simulation, l'agent peut se déplacer dans une cellule adjacente s'il n'existe pas de murs empêchant ce mouvement.

Les agents souhaitent se rencontrer le plus rapidement possible. Dans le but de déterminer le point de rendez-vous optimal, nous tendons à minimiser le nombre maximal de mouvements faits par un agent, et, en cas d'égalité, nous voulons décroître le nombre de mouvements réalisés par le second agent faisant le plus de pas et ainsi de suite. Par conséquent, nous appliquons la règle du leximax.

Il est intéressant de remarquer que la règle du leximax est Pareto-inclusive et capture l'idée d'équité puisque les efforts réalisés par les agents sont équilibrés. Nous appelons optimum local, une cellule leximax-optimale selon les estimations des distances faites par les agents. En fait, le calcul du point de rendez-vous optimal requiert la connaissance du labyrinthe entier c'est-à-dire de ses murs. Sous ces hypothèses, les agents sont capables de calculer le plus court chemin entre eux et le point de rendez-vous optimal.

Comme le labyrinthe peut être imparfait (un chemin entre deux cellules n'est pas nécessairement unique), les agents doivent se fixer un point de rendez-vous. Sinon, ils pourraient se tourner autour sans jamais se rencontrer. Contrairement au problème classique de rendez-vous (ALPERN & GAL, 2003), les agents sont autorisés à communiquer. Chaque agent peut informer les autres participants de l'existence de murs (ou ils peuvent proposer/accepter des points de rendez-vous potentiels, appelés des alternatives. À chaque pas de simulation, la préférence d'un agent entre deux cellules dépend des distances entre lui et celles-ci à cette étape. Évidemment, la distance évaluée et donc le résultat de la négociation dépendent de la connaissance partielle des agents, i.e. leur perception de l'environnement.

Exemple 4 *Pour illustrer ce problème, nous considérons un labyrinthe de taille 3×3 (cf figure 4.2). Au premier pas de simulation, Alice est en c_1 tandis que Bob est en c_9 . Nous considérons que le champ de vision est limité à la cellule courante, par conséquent, Alice ne voit aucun mur alors que Bob voit un mur entre c_8 et c_9 . Puisque Alice ne voit aucun mur, elle estime que la distance entre elle et la cellule c_7 est 2. De son côté, Bob estime que la distance entre lui et cette cellule est 4.*

4.7 Résolution

Le protocole de négociation, le critère d'acceptation et la stratégie d'offres utilisés ici sont présentés dans la section 4.4.

Toutefois, comme nous considérons des résolutions de problèmes distribués, nous supposons que les agents sont collaboratifs et donc conciliants dans le but de trouver un accord aussi vite que possible. Un agent commence par proposer la meilleure offre pour lui puis concède. Dire qu'un agent concède signifie qu'il propose une nouvelle offre légale telle qu'il n'existe pas d'autres alternatives préférées et légales ne figurant pas dans son historique. Le point de rendez-vous est sélectionné aléatoirement. Si ce dernier appartient à l'ensemble des accords de la précédente négociation alors le point de rendez-vous est le même.

Tous les accords atteints par un jeu de négociation sont des optima locaux. On peut noter qu'une simulation se termine toujours avec succès.

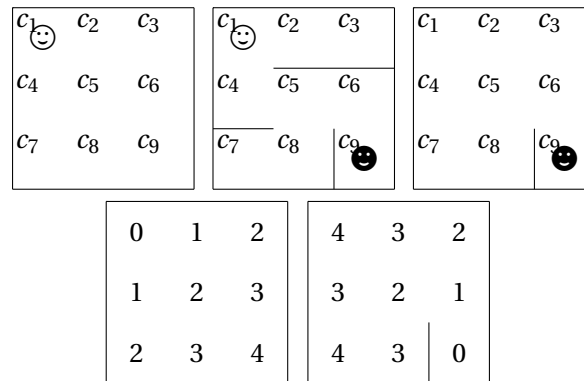


FIGURE 4.2 – Premier pas de simulation : G (en haut au centre) et sa représentation interne pour Alice (en haut à gauche) et Bob (en haut à droite) et l’estimation par les agents des fonctions de distances pour Alice (en bas à gauche) et pour Bob (en bas à droite).

4.8 Implémentation

Le démonstrateur (DELECROIX, MORGE, NACHTERGAELE & ROUTIER, 2014) a été réalisé en Java. Cette implémentation comprend :

- un modèle d’environnement sous la forme d’un graphe non-orienté qui permet de représenter et générer des labyrinthes avec des topologies variables (couloirs longs/courts, nombre de murs) ;
- un modèle d’agent capable de
 - percevoir l’environnement,
 - calculer le plus court chemin entre sa position et un point de rencontre,
 - générer ses préférences à partir de ces calculs,
 - se déplacer ;
- le processus de négociation décrit dans la section 4.4 ;
- une interface graphique qui repose sur l’API Swing pour afficher des labyrinthes, visualiser le processus de négociation et paramétrer une simulation.

Le démonstrateur permet de configurer dans une simulation :

- le nombre d’agents ;
- le halo de perception ;
- 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.

L’exécution d’une simulation (cf figure 4.3) permet de visualiser :

- la vue globale de l’environnement qui présente la trace des agents ;
- la représentation interne de chacun des agents, notamment les calculs de chemin ;
- les différentes étapes du processus de négociation et l’accord atteint.

D’autre part, nous avons élaboré un « tube à essai » paramétrable permettant de lancer les expérimentations désirées.

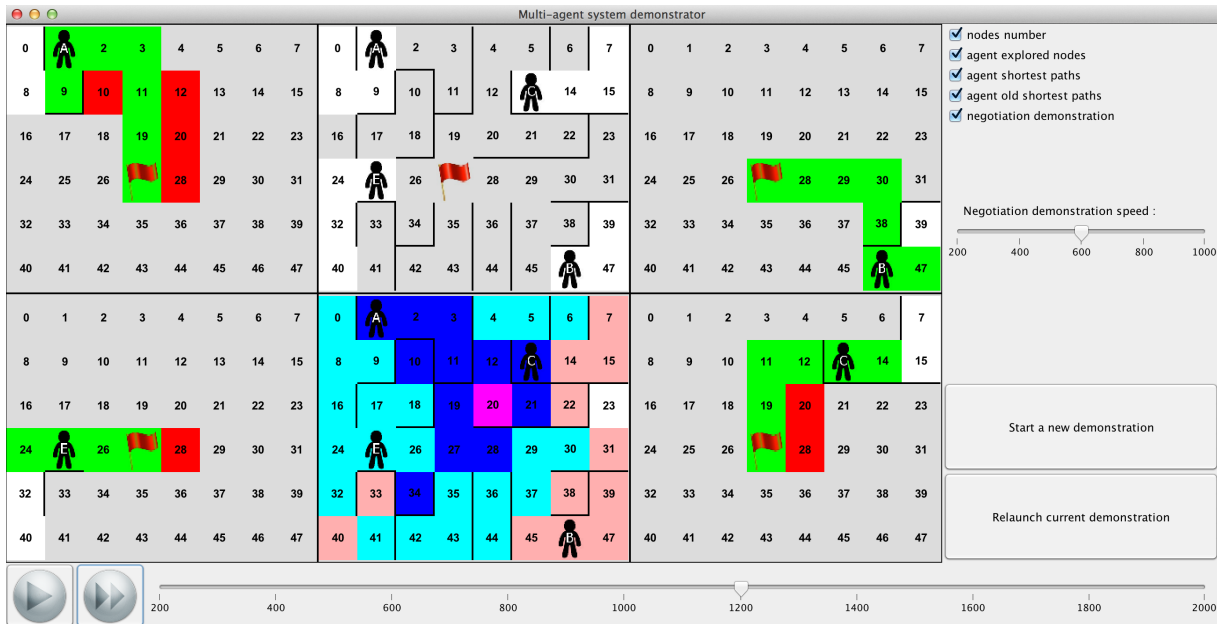


FIGURE 4.3 – Capture d’écran du démonstrateur. Parmi les six panneaux, celui au centre en haut permet d’observer la progression des agents, celui en-dessous révèle les offres échangées, les quatre autres représentent la perception locale de chaque agent.

4.9 Évaluation de la résolution

Nos expérimentations ont pour but de : (i) comparer la résolution du problème distribué par la négociation avec sa résolution par d’autres heuristiques ; (ii) étudier l’impact de la communication sur cette résolution.

La génération pseudo-aléatoire de labyrinthes a été réalisée par :

- une version pseudo-aléatoire de l’algorithme de Prim. Cet algorithme produit des labyrinthes simples (i.e. avec des courts chemins sans issue et donc une solution souvent directe) ;
- l’algorithme *recursive backtracker*. Cet algorithme produit des labyrinthes difficiles (i.e. avec des chemins sans issue moins nombreux mais plus longs et donc une solution souvent longue à déterminer et complexe).

Nos expérimentations montrent que la négociation ne constitue pas une meilleure heuristique que fixer le point de rencontre au centre des agents en ce qui concerne la réduction de l’effort total déployé par les agents. Autrement dit, la satisfaction globale des agents n’est pas améliorée par la négociation. Cependant, dans les labyrinthes simples, les heuristiques adaptatives, en particulier la négociation, requièrent moins de pas de simulation. Le bénéfice croît avec la taille du labyrinthe considéré. La valeur ajoutée de la négociation, i.e. une résolution rapide, est perdue dans les labyrinthes complexes. De plus, la négociation est équitable puisqu’elle partage plus les efforts entre les agents que les autres heuristiques.

De plus, il est clair que la phase de partage de l’information améliore la rapidité de la résolution. Dans les labyrinthes difficiles, quand les agents ne partagent pas d’informations, les heuristiques adaptatives ont de moins bonnes performances. En fait, les erreurs, causées par les voies sans issue, coûtent cher. À cause du recalcul du point de rendez-vous, les erreurs se cumulent. De plus, les agents peuvent répéter les erreurs faites par les autres puisqu’ils ne partagent pas l’information. Quand un problème est difficile, une heuristique adaptative n’est pas efficace sans informations additionnelles. Cependant, ces stratégies requièrent moins de pas de simulation quand les agents partagent leur

perception. Plus le problème est difficile, plus le partage de l'information est important. De surcroît, le nombre de pas de simulation diminuent dans les labyrinthes difficiles. Plus nous avons d'agents, plus les informations sur l'environnement sont nombreuses et partagées. La communication permet d'éviter de répéter les erreurs coûteuses évoquées précédemment. Il n'est pas surprenant que davantage d'informations aident à la résolution du problème. Plus le problème est difficile, plus cette aide est bénéfique.

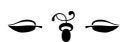
Finalement, nous observons que même si la communication des murs induit plus de messages, le nombre de murs communiqués est insignifiant par rapport au nombre de propositions. De plus, le coût de la communication de l'heuristique de négociation pure est plus important que celle de la négociation avec partage d'information. En effet, cette dernière est plus efficace et donc requiert moins de pas de simulation.

4.10 Conclusion

Dans ce chapitre, nous avons proposé un jeu de négociation multi-latérale où les agents échangent des offres mono-attribut. Contrairement à la plupart des travaux (cf chapitre 2), nous avons modéliser la rationalité d'un agent à l'aide de préférences incomplètes (larges et partielles) plutôt qu'avec une fonction d'utilité. De cette manière, les agents peuvent être indécis car ils ne sont pas toujours en mesure de comparer les alternatives en raison du manque d'information, d'une incertitude, ou à cause de préférences contradictoires. Afin d'atteindre un accord équitable, notre protocole ne priorise aucun des agents, il est symétrique. Comme un agent peut être indécis, il est autorisé à soumettre plus d'une offre à chaque tour. Nous avons proposé deux stratégies de concession qui consistent à proposer une ou plusieurs alternatives préférées et à réduire pas-à-pas ses prétentions si ces propositions ne sont pas retenues. Alors que la stratégie coopérative réalise autant de concessions que possible, la stratégie temporisatrice retarde un accord en ne soumettant les alternatives qu'une-à-une. Nous avons démontré que l'accord obtenu entre des agents coopératifs est leximax-optimal et donc Pareto-optimal et que la stratégie temporisatrice sert l'intérêt individuel de l'agent. Nos expérimentations 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.

Nous avons appliqué nos travaux dans le cadre de la résolution de problèmes distribués. En particulier, nous avons considéré le cas où deux agents ou plus sont dans un labyrinthe. Ils souhaitent se rencontrer le plus vite possible en négociant un point de rendez-vous dans le but de minimiser et de partager l'effort pour l'atteindre. Nous avons prouvé que nos simulations atteignent un rendez-vous. Nos expérimentations montrent que la négociation ne réduit pas l'effort total des agents mais permet de le partager équitablement entre eux. De plus, nous avons proposé une tactique de coordination adaptative consistant à négocier à chaque fois que la longueur du chemin menant au point de rendez-vous augmente pour l'un des agents (due à la découverte d'un mur). Ce problème distribué permet de s'attaquer à des négociations dans de nombreux domaines avec un grand nombre de participants. Notre algorithme est donc déployable à grande échelle.

Même si nous considérons ici uniquement des labyrinthes, notre négociation permet d'explorer tous les types de graphe, quelle que soit leur topologie. De plus, notre problème distribué consiste en un « tube à essai » pour évaluer et comparer les protocoles de négociation et les stratégies.



CHAPITRE 5

Négociation multi-bilatérale

Sommaire

5.1	Introduction	50
5.2	Problème du mariage stable	51
5.3	Évaluation sociale des appariements	52
5.3.1	Optimalité sociale	52
5.3.2	Bien-être social	52
5.4	Swing	53
5.5	Swing++	55
5.6	Travaux connexes	56
5.7	Évaluation	58
5.7.1	Évaluation de Swing	58
5.7.2	Évaluation de Swing++	58
5.7.3	Synthèse	59
5.8	Problème des hôpitaux/résidents	59
5.9	Adaptation de Swing	60
5.10	Application pratique	61
5.11	Expérimentations	63
5.12	Conclusion	64

Les travaux présentés dans ce chapitre ont été réalisés entre 2011 et 2014 avec Patricia EVERAERE (Université de Lille) et Gauthier PICARD (Mines de Saint-Étienne) avec lequel j'ai co-encadré les stages de recherche de Santiago VILAREAL aux Mines de Saint-Étienne et d'Eric PIETTE à l'Université de Lille. Ces travaux ont notamment été publiés dans les articles suivants :

1. EVERAERE, P., MORGE, M. & PICARD, G. (2012). Casanova : un comportement d'agent respectant la privacité pour des mariages stables et équitables. *Revue d'Intelligence Artificielle*, 5, 471-494
2. EVERAERE, P., PICARD, G. & MORGE, M. (2013). Minimal Concession Strategy for Reaching Fair, Optimal and Stable Marriages (Extended Abstract). In *Proc. of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)* (p. 1319-1320). A* (CORE). IFAAMAS
3. PIETTE, E. & MORGE, M. (2014). Practical Application of Matchmaking Problem : Trainee Allocation for Teachers. In Y. DEMAIZEAU, F. ZAMBONELLI, J. M. CORCHADO & J. BAJO (Éd.), *Proc. of 12th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS), Salamanca, Spain.* (T. 8473, p. 195-206). LNAI. C (CORE). Springer

De plus, ces travaux ont été couronnés par le prix du meilleur papier des 19^{ème} Journées Franco-phones sur les Systèmes Multi-Agents (JFSMA'11) à Valenciennes décerné à l'article (MORGE, EVERAERE & PICARD, 2011).

5.1 Introduction

Nous nous intéressons ici à un problème bien connu en Informatique (KNUTH, 1971) et en Économie (ROTH & OLIVEIRA SOTOMAYOR, 1990), celui des mariages stables – en anglais, *Stable Marriage Problem* (SM). L'enjeu de ce problème est simple. En considérant deux communautés dont chaque membre a des préférences vis-à-vis des individus de l'autre communauté, l'objectif est de former des couples en prenant en compte leurs préférences. Ce problème abstrait a de nombreuses applications. Par exemple, pour affecter des projets à des étudiants, chaque étudiant classe par ordre de préférence les projets et réciproquement le responsable de chaque projet classe lui aussi les candidats potentiels. Nous considérons dans ce chapitre l'application réelle qui consiste à affecter des enseignants-stagiaires dans les écoles maternelles et élémentaires de l'Académie de Lille.

L'algorithme séminal de GALE et SHAPLEY (1962) - dénoté GS - est une preuve constructive de l'existence d'une solution admissible pour n'importe quelle instance de SM. L'agentification de cet algorithme proposé par BRITO et MESEGUER (2005) consiste à considérer l'appariement obtenu comme le résultat émergent de multiples négociations bilatérales. Ainsi, on distingue deux types d'agent : une communauté de proposants et une communauté de disposants. Les négociations entre ces agents aboutissent à une solution stable mais inéquitable : la communauté des proposants est favorisée. Dans ce chapitre, nous proposons la méthode Swing où les agents jouent alternativement ces deux rôles dans de multiples négociations bilatérales. Notre approche permet l'émergence d'un appariement stable qui ne peut être atteint avec l'algorithme GS. Ces appariements sont plus équitables car notre méthode ne favorise pas une communauté au détriment d'une autre. À notre connaissance, Swing est la seule méthode distribuible permettant d'atteindre une solution stable et équitable. Malheureusement, Swing, peut ne pas se terminer pour certaines instances du problème. En effet, cette méthode peut mener à des situations de dilemme. Dans ce chapitre, nous étendons Swing pour : i) détecter les éventuels dilemmes; ii) et les résoudre. Comme Swing, notre méthode Swing++ prône l'équité.

On peut remarquer que pour des raisons de scalabilité aucune méthode d'optimisation ne peut être utilisée pour l'affectation des enseignants-stagiaires dans les écoles de l'Académie de Lille. Nous avons constaté que ce problème réel peut être réduit au problème bien connu des hôpitaux/résidents (HR) formalisé par GALE et SHAPLEY (1962). Toutefois, la difficulté majeure réside dans la génération des listes de préférences en fonction des contraintes réelles. De plus, en appliquant l'algorithme GS existants, nous privilégions une communauté (e.g. la distance moyenne) ou un autre (e.g. la priorité entre stagiaires). C'est la raison pour laquelle, nous utilisons ici la méthode Swing qui est plus équitable. Dans ce chapitre, nous adaptons la méthode Swing au problème HR et nous l'appliquons au problème réel envisagé afin de trouver un équilibre entre les différents objectifs. La méthode Swing permet alors de relaxer certaines contraintes et ainsi elle permet de décroître les coûts de gestion.

Nous commençons ce chapitre par une présentation du problème SM puis nous présentons l'algorithme GS dans une perspective orientée individu dans la section 5.2. Afin d'évaluer les solutions à un problème SM nous utilisons la Théorie des Jeux et la Théorie du Choix Social en section 5.3. Dans la section 5.4, nous présentons la méthode Swing et nous identifions les situations de dilemme qui provoquent la non-terminaison. Notre algorithme Swing++ est présenté dans la section 5.5. Nous exposons brièvement les autres méthodes existantes pour la résolution du problème SM (cf section 5.6) puis nos expérimentations permettent d'évaluer et de comparer Swing, Swing++ et les autres méthodes (cf section 5.7). Dans la section 5.9, nous adaptons la méthode Swing au problème d'af-

fection des professeurs des écoles stagiaires. Ensuite, nous présentons en détail le problème réel que nous adressons (cf section 5.10) et nous comparons les solutions obtenues par Swing à celles des autres méthodes (cf section 5.11). Finalement, nous en tirons quelques conclusions puis nous dressons quelques perspectives (cf section 5.12).

5.2 Problème du mariage stable

Nous étudions ici le problème des mariages stables, en anglais *Stable Marriage Problem*, qui a été introduit par GALE et SHAPLEY (1962). Ce problème s'énonce simplement. On considère n individus d'une communauté (e.g la gent féminine) et n individus d'une autre communauté (e.g la gent masculine). Chaque individu a des penchants pour les individus de l'autre partition avec lesquels il souhaite être apparié.

Dans ce chapitre, on suppose que les relations de préférence sont des ordres stricts, ce qui signifie que chaque individu est en mesure de choisir entre deux partenaires (il n'y a ni *ex æquo*, ni indifférence). On appelle problème de mariage stable avec liste complète (SMC) un problème SM où toutes les préférences sont complètes (tous les individus sont classés). Par opposition, dans un problème de mariage stable incomplet (SMI), au moins l'une des listes de préférence est incomplète. Le fait qu'une liste de préférences soit incomplète signifie que l'individu préfère rester seul plutôt que d'être mal accompagné.

Résoudre un problème SM consiste à assortir ces communautés. On appelle appariement, en anglais *matching*, un ensemble de n mariages monogames et hétérosexuels. Dans un appariement complet, chaque individu est marié. Dans le cas d'un problème SMC, seuls les appariements complets sont intéressants. À l'inverse, pour un problème SMI, certains appariements complets sont irrationnels car ils consistent à marier un individu à un partenaire qui n'est pas dans sa liste de préférences.

Un appariement est stable s'il n'existe pas de couple qui préférerait être ensemble plutôt qu'avec leur partenaire actuel. En d'autres termes, il n'existe pas de relation extra-conjugale menaçant l'appariement. Ces relations prennent la forme de couples hypothétiques bloquants. Un appariement stable est une solution admissible à un problème SMC. Dans le cas d'un problème SMI, un appariement stable¹ est admissible s'il est rationnel, c'est-à-dire si chacun des individus est marié à un partenaire qui est dans sa liste de préférences.

L'algorithme de GALE et SHAPLEY (1962) est une preuve constructive de l'existence d'une solution admissible pour n'importe quelle instance de SM (complet ou non). Il peut être décrit comme un cérémonial au cours duquel les femmes attendent dans une salle et les hommes, qui sont à l'extérieur, rentrent à tour de rôle dans un ordre arbitraire. Quand un homme entre, il se présente à la femme qu'il préfère. Si la femme est libre et qu'elle le considère comme acceptable (i.e. il est dans sa liste de préférences), il devient son partenaire. Si elle a déjà un partenaire, il y a conflit et elle choisit l'homme qu'elle préfère. L'élu reste et l'autre, rejeté, sort de la pièce. Ce dernier se souvient de cet échec et reviendra plus tard pour se présenter à la femme qui suit dans l'ordre de ses préférences. On s'arrête quand tous les hommes qui ne sont pas libres sont désespérés (leur liste de préférences est vide).

Dans l'algorithme GS, les hommes proposent et les femmes disposent. Une exécution consiste en une séquence de propositions des hommes envers les femmes. En d'autres termes, les hommes jouent le rôle de proposant et les femmes jouent le rôle de disposant. On dit que GS est orienté homme. Il peut être facilement adapté pour être orienté femme, en inversant leur rôle respectif. Dans ce cas, les femmes proposent et les hommes disposent.

1. On peut remarquer que la notion de stabilité est différente de celle évoquée dans la section 2.4. Afin de rester cohérent avec la littérature, nous conserverons cette dénomination.

5.3 Évaluation sociale des appariements

Un problème de mariages peut avoir plusieurs solutions stables. Pour les distinguer les unes des autres, nous considérons ici différentes notions. Dans la section 5.3.1, nous considérons des concepts issus de la Théorie des Jeux qui soulignent que l'optimalité sociale permet de distinguer certaines solutions mais qui restent insuffisamment discriminants. Dans la section 5.3.2, nous considérons des concepts issus de la Théorie du Choix Social.

5.3.1 Optimalité sociale

Une fois qu'un appariement est établi, il est essentiel de déterminer à quel point cet appariement est acceptable pour la société d'agents. Dans cette section, nous utilisons la Théorie des Jeux pour évaluer les solutions admissibles à un problème SM (ROTH & OLIVEIRA SOTOMAYOR, 1990). Pour la clarté, nous nous restreignons dans la suite de ce chapitre au problème avec liste complète (SMC). Nous évaluons alors les solutions admissibles, i.e. celles qui sont stables.

Un appariement stable est le résultat d'un jeu entre deux communautés. Il peut être évalué du point de vue d'une communauté ou de l'autre à l'aide de la notion de Pareto-dominance (cf section 2.3). Donald Knuth a démontré dans (KNUTH, 1971) que l'exécution de l'algorithme GS orienté homme aboutit à une solution – un appariement stable – qui est mâle-optimal et que cette solution est une des pires du point de vue des femmes.

Comme l'espace des versions bien connu en apprentissage automatique (MITCHELL, 1997), l'espace des solutions stables est un treillis. L'espace que l'on envisage ici ne s'appuie pas sur une relation de généralisation mais sur la relation de Pareto-dominance (e.g. sur les femmes) et la frontière supérieure est constituée des mâle-optimaux alors que la frontière inférieure est l'ensemble des solutions femelle-optimales. Chacune des solutions frontières peut être atteinte par l'algorithme GS en modifiant l'ordre d'apparition des proposants. Toutes les autres solutions, hors des frontières, ne peuvent être atteintes par l'algorithme GS mais par un algorithme centralisé qui passe d'un appariement stable à un autre appariement stable en dégradant la satisfaction d'une communauté ou de l'autre communauté.

La stabilité est une condition suffisante à la Pareto-optimalité mais ce n'est pas une condition nécessaire (ROTH & OLIVEIRA SOTOMAYOR, 1990). On peut noter qu'un appariement Pareto-optimal n'est pas nécessairement stable.

En résumé, l'optimalité au sens de Pareto permet de distinguer parmi les solutions stables lesquelles sont meilleures du point de vue d'une communauté ou d'une autre, mais elle ne permet pas de distinguer lesquelles sont les meilleures du point de vue de la société globale.

5.3.2 Bien-être social

Afin d'évaluer la qualité d'une solution du point de vue collectif, la Théorie du Choix Social (MOULIN, 2002) propose d'autres outils que la Théorie des Jeux. Cette évaluation repose sur une mesure individuelle de la satisfaction des individus, qui est ensuite agrégée afin d'estimer la satisfaction de toute (ou partie de) la société.

Pour le problème SM, chaque individu (quelle que soit sa communauté) évalue sa satisfaction en fonction de ses préférences via son regret qui est : 0 s'il est marié avec le premier individu de sa liste; 1 s'il est marié avec le deuxième individu de sa liste, etc. Moins un individu a de regret, plus sa satisfaction est grande.

Comme nous l'avons vu dans la section 2.3.2, de nombreuses fonctions permettent d'agrèger les utilités des individus, comme par exemple le bien-être utilitaire. Nous adaptons ces mesures au problème SM afin d'évaluer le point de vue global, et nous introduisons la mesure de bien-être équitable pour évaluer les disparités entre communautés. Pour le bien-être utilitaire, masculin et féminin, plus

le bien-être est grand, plus la solution est optimale du point de vue de la communauté envisagée. Pour le bien-être équitable, plus le bien-être est grand, plus la solution est probe. Nous avons choisi d'envisager l'équité entre les communautés ainsi que l'équité au sein des couples plutôt que le bien-être égalitaire qui considère le bien-être de l'agent le plus insatisfait car cette notion est indépendante de la communauté. Pour une même instance de problème SM, l'agent le moins satisfait par une solution admissible peut être un homme et l'agent le moins satisfait par une autre solution admissible peut être une femme. La comparaison des solutions admissibles à l'aide de cette métrique n'est donc ni simple ni directe.

Exemple 5 *Considérons le problème SMC constitué de 3 hommes (x_1, x_2, x_3) et de 3 femmes (y_1, y_2, y_3) dont les préférences sont comme suit :*

- x_1 préfère y_2 à y_1 qu'il préfère à y_3 ;
- x_2 préfère y_3 à y_2 qu'il préfère à y_1 ;
- x_3 préfère y_1 à y_3 qu'il préfère à y_2 ;
- y_1 préfère x_2 à x_1 qu'elle préfère à x_3 ;
- y_2 préfère x_3 à x_2 qu'elle préfère à x_1 ;
- y_3 préfère x_1 à x_3 qu'elle préfère à x_2 .

L'appariement M tel que x_1 est avec y_2 , x_2 avec y_1 et x_3 avec y_3 est instable car (x_2, y_2) est un couple bloquant. À l'inverse, les trois appariements suivants sont stables :

- M_1 tel que x_1 est avec y_1 , x_2 avec y_2 et x_3 avec y_3 ;
- M_2 tel que x_3 est avec y_1 , x_1 avec y_2 et, x_2 avec y_3 ;
- M_3 tel que x_2 est avec y_1 , x_3 avec y_2 et x_1 avec y_3 .

Le résultat de l'algorithme GS orienté homme (respectivement orienté femme) est l'appariement M_2 (respectivement M_3). L'appariement M_1 ne peut être atteint par aucune exécution de GS qu'il soit orienté homme ou femme. De plus, l'appariement M_2 (respectivement M_3) est mâle optimal (respectivement femelle optimal) car les hommes (respectivement femmes) sont mariés avec leur partenaire préféré. L'appariement M_1 , où tous les individus sont mariés avec leur second partenaire préféré, est un optimum de Pareto comme M_2 et M_3 .

Les biens-être pour les différents appariements stables sont les suivants :

Bien-être	M_1	M_2	M_3
utilitaire	.5	.5	.5
utilitaire des hommes	.5	1	0
utilitaire des femmes	.5	0	1
égalitaire	1	0	0

5.4 Swing

Dans l'algorithme GS, une communauté joue le rôle de proposant, l'autre de disposant. Ainsi les proposant sont favorisés. Swing permet à chaque communauté d'adopter alternativement ces deux rôles afin d'atteindre un résultat plus équitable. Swing s'inspire du protocole de concession monotone présenté dans la section 2.2.3. À cette intention, les concessions des agents s'appuient sur le regret des agents. Chacun s'adresse à l'agent qu'il préfère. Si ce dernier refuse, l'agent concède en s'adressant au partenaire suivant dans sa liste de préférences. Pendant ce temps, les partenaires potentiels jouent le rôle de disposant en acceptant/refusant les propositions au regard de leur niveau de concession (noté κ).

Dans la méthode Swing, les hommes et les femmes se proposent alternativement. Chaque proposant envoie une demande à l'ensemble des individus acceptables. Ainsi quand une proposition est acceptée par un disposant :

1. si le proposant est marié, il divorce ;
2. si le disposant est marié, il divorce ;
3. le proposant et le disposant se marient ;
4. le niveau de concession du proposant et celui du disposant sont modifiés de sorte qu'ils ne peuvent divorcer que pour de meilleurs partenaires.

Quand tous les individus sont mariés, Swing se termine². Lorsqu'un agent divorce, son ancien partenaire devient célibataire et concède, c.-à-d. qu'il considère le partenaire suivant dans sa liste comme acceptable. Quand Swing se termine, il atteint une solution pour le problème SMC : un appariement stable.

Exemple 6 *Considérons l'exemple précédent. Les messages reçus/envoyés par les agents sont représentés dans la figure 5.1. Les négociations mènent ici à l'appariement M_1 où x_1 est avec y_1 , x_2 avec y_2 et x_3 avec y_3 . Lors de la première étape, les hommes proposent et les femmes disposent. Initialement, x_1 est libre et son niveau de concession est $\kappa = 1$. En conséquence, le seul partenaire acceptable est y_2 et donc x_1 envoie une proposition à y_2 . Pour des raisons similaires, x_2 envoie une proposition à y_3 et x_3 envoie une proposition à y_1 . Comme aucun de ces hommes n'est acceptable pour leur partenaire potentiel respectif, les femmes rejettent ces propositions. Lors de la seconde étape, les femmes proposent et les hommes disposent. y_1 est libre et son niveau de concession est $\kappa = 1$. En conséquence, le seul partenaire acceptable est x_2 et donc y_1 envoie une proposition à x_2 . Pour des raisons similaires, y_2 envoie une proposition à x_3 et y_3 envoie une proposition à x_1 . Comme aucune de ces femmes n'est acceptable pour leur partenaire potentiel respectif, les hommes rejettent ces propositions. Lors de la troisième étape, le niveau de concession de x_1 est $\kappa = 2$, y_1 et y_2 sont acceptables. Même si y_2 rejette la proposition, y_1 l'accepte car son niveau de concession est $\kappa = 2$. y_2 accepte la proposition de x_2 car leur niveau de concession est $\kappa = 2$. Pour des raisons similaires, y_3 accepte la proposition de x_3 .*

Malheureusement, la terminaison de Swing n'est pas garantie. Les cycles d'exécution de Swing sont provoqués par le souhait des hommes de se diriger vers les solutions mâle-optimales et les femmes vers les solutions femelle-optimales. Ces situations de dilemme prennent la forme de circuit dans le graphe associé. On peut remarquer que dans un tel circuit le nombre d'hommes menaçants et le nombre de femmes menaçantes est identique et donc la taille d'un circuit est toujours paire.

Exemple 7 *Considérons le problème SMC constitué de 3 hommes (x_1, x_2, x_3) et de 3 femmes (y_1, y_2, y_3) dont les préférences sont comme suit :*

- x_1 préfère y_3 à y_2 qu'il préfère à y_1 ;
- x_2 préfère y_3 à y_2 qu'il préfère à y_1 ;
- x_3 préfère y_1 à y_2 qu'il préfère à y_3 ;
- y_1 préfère x_2 à x_1 qu'elle préfère à x_3 ;
- y_2 préfère x_1 à x_3 qu'elle préfère à x_2 ;
- y_3 préfère x_3 à x_2 qu'elle préfère à x_1 .

Pour cette instance, les appariements stables sont :

- M_1 où x_1 est avec y_2 , x_2 avec y_3 et x_3 avec y_1 ;

2. Comme dit précédemment, le problème du mariage stable avec listes incomplètes est hors de portée de ce chapitre. Toutefois, on peut remarquer que Swing peut être adapté à ce type de problème en modifiant le test d'arrêt.

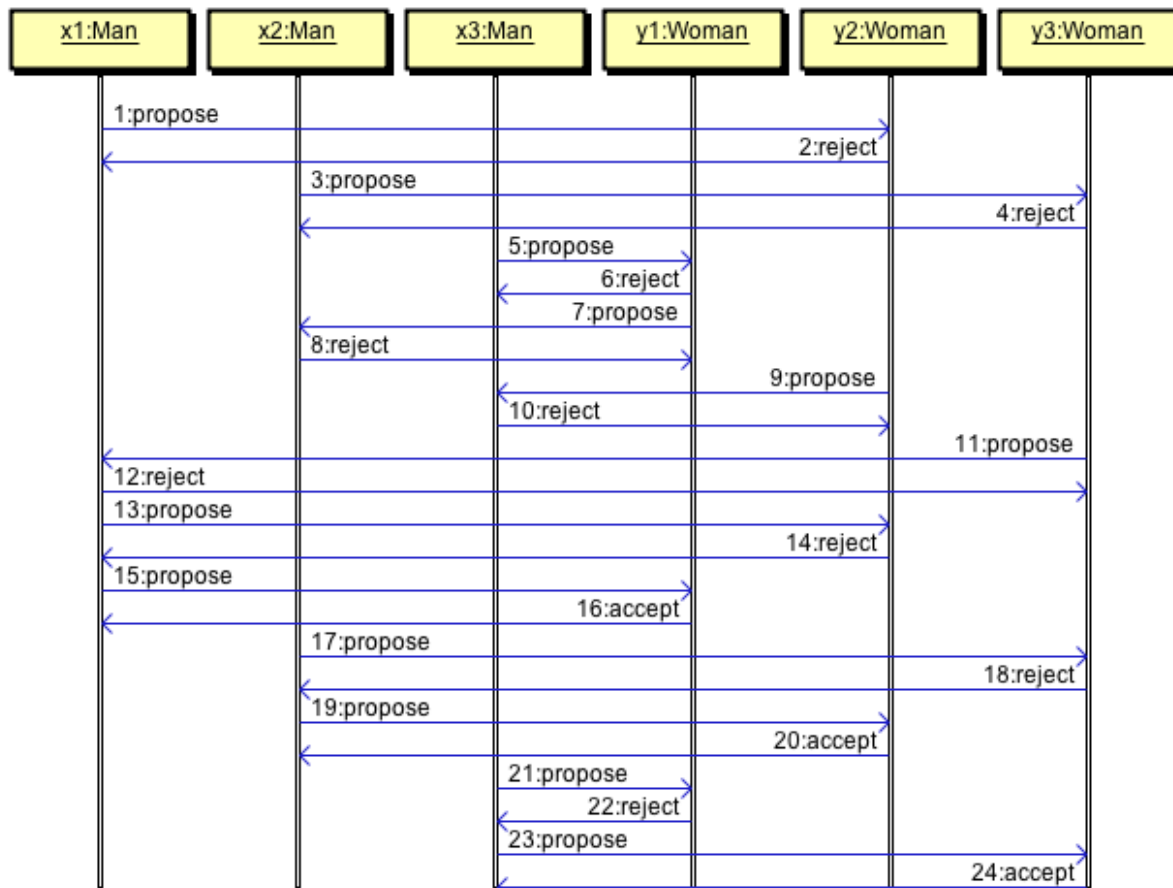


FIGURE 5.1 – Trace d'exécution Swing

- M_2 où x_1 est avec y_2 , x_2 avec y_1 et x_3 avec y_3 .

Alors que M_1 est mâle-optimal (le résultat de l'algorithme GS orienté homme), M_2 est femelle-optimal (le résultat de l'algorithme GS orienté femme). Ces solutions peuvent être représentées dans un graphe (cf figure 5.2) où chaque nœud correspond à un couple et un arc signifie que l'un des partenaires du couple origine menace le couple destination. Par exemple x_3 préfère y_1 et y_3 préfère être avec x_3 . Ce graphe contient le circuit $[x_3, y_3, x_2, y_1]$.

5.5 Swing++

Pour éviter les cycles d'exécution présentés dans la section précédente, nous proposons d'étendre Swing à Swing++ afin de détecter les dilemmes et de les traiter.

Dans Swing++, les agents qui sont impliqués dans un dilemme vont soit se sacrifier soit abandonner. Un sacrifice consiste à briser le circuit en n'envoyant pas de demande au partenaire potentiel mais à l'individu suivant dans la liste de préférences. Un abandon consiste à passer son tour de parole. À chaque divorce, un agent mémorise son nouveau partenaire, c.-à-d son amant. L'algorithme permet de détecter les circuits quel que soit le nombre d'agents impliqués. Il permet de déterminer si un couple est impliqué dans un circuit. Si le désir est réciproque, alors ce n'est pas le cas. Dans le cas contraire, si le premier agent n'a pas d'amant et il n'y a pas de circuit. Sinon on ajoute le nouvel amant au circuit. Finalement, on vérifie que le couple est compris dans cette liste.

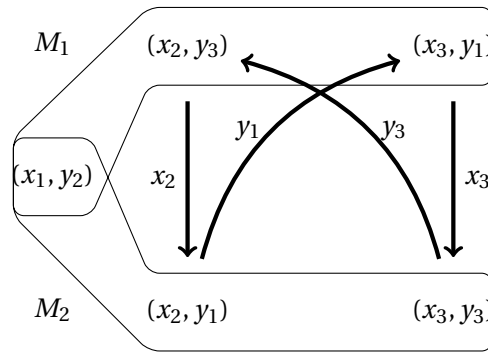


FIGURE 5.2 – Un exemple de circuit

Pour résoudre un dilemme, les agents responsables du circuit doivent ne pas envoyer de proposition, on parle de sacrifice. Ainsi, d'autres solutions sont explorées. Si le nombre d'agent ayant détecté le circuit est maximum, alors l'agent se sacrifie (et donc brise le circuit), le nombre de détection maximum incrémente afin qu'un autre individu se sacrifie dans le cas où le même circuit se reforme. Sinon, il abandonne et incrémente de 1 le nombre de détection du circuit.

On peut noter qu'il est possible, au cours de l'exécution de Swing++, que deux individus célibataires se sacrifient jusqu'à se marier en rendant l'appariement instable. Afin de s'assurer que l'appariement calculé soit stable, nous vérifions lorsqu'on ajoute un mariage à l'appariement, que celui-ci reste stable. Ainsi, ce test garantit la stabilité de la solution.

Exemple 8 Nous considérons ici le problème de l'exemple 7. Pour cette instance, l'exécution de Swing++ est identique à l'exécution de Swing pour les 7 premières étapes. Dans le tableau 5.1, nous représentons les états de E_7 à E_{10} avec les amants. Au cours de l'étape entre E_7 et E_8 , x_3 détecte le circuit $[x_3, y_3, x_2, y_1]$ et donc il se sacrifie pour le briser. Au cours de l'étape entre E_{10} et E_{11} , y_1 ne se propose pas à x_2 car il détecte un circuit. Comme c'est la seconde détection, y_1 abandonne. Ensuite, y_3 ne se propose pas à x_3 car il détecte de nouveau ce circuit. Étant le second agent à faire cette détection pour la seconde fois, y_3 concède et elle se propose à x_2 qui accepte. L'appariement obtenu est M_1 qui est stable.

En résumé, Swing++ est une méthode pour laquelle nous ne disposons pas à ce jour, de preuve théorique de terminaison ou de succès. Toutefois, comme nous allons le voir expérimentalement Swing++ propose les mêmes avantages que Swing et se termine pour l'ensemble des instances.

5.6 Travaux connexes

Parmi les méthodes existantes pour la résolution de problème SMC, on trouve : l'algorithme GS (GALE & SHAPLEY, 1962), l'algorithme SML2 (GELAIN, PINI, ROSSI, VENABLE & WALSH, 2010) et l'algorithme ZigZag (ZAVIDOVIQUE, SUVONVORN & SEETHARAMAN, 2005).

L'algorithme GS (GALE & SHAPLEY, 1962) permet d'atteindre un appariement stable. Il favorise une communauté au détriment d'une autre. Si on considère l'algorithme GS orienté homme, alors l'appariement est mâle-optimal et cette solution est une des pires du point de vue des femmes. L'agentification de cette résolution est simple et directe. C'est l'objet de la méthode DisEGS proposée par BRITO et MESEGUER, 2005 qui consiste à faire négocier deux types d'agents – les proposant et les disposants – pour faire émerger une solution admissible.

L'algorithme DisFC (BRITO & MESEGUER, 2005) est une adaptation de l'algorithme ABT (YOKOO, DURFEE, ISHIDA & KUWABARA, 1998). Comme ABT, l'algorithme DisFC est une méthode distribuée

État	a'	x_1	x_2	x_3	y_1	y_2	y_3
E_7	σ_a	\top	\top	\perp	\perp	\top	\top
	μ_a	y_2	y_3	λ	λ	x_1	x_2
	κ_a	1	0	2	2	0	1
	η_a	λ	y_3	y_1	x_2	λ	x_3
E_8	σ_a		\top	\perp	\perp		\top
	μ_a	//	y_3	λ	λ	//	x_3
	κ_a		0	3	2		1
	η_a		y_3	λ	x_2		x_3
E_9	σ_a		\perp	\top	\perp		\top
	μ_a	//	λ	y_3	λ	//	x_3
	κ_a		2	2	3		0
	η_a		y_3	λ	x_2		x_3
E_{10}	σ_a		\perp	\top	\top		\perp
	μ_a	//	λ	y_1	x_3	//	λ
	κ_a		2	0	1		2
	η_a		y_3	y_1	x_2		x_3
E_{11}	σ_a		\top	\top	\top		\top
	μ_a	//	y_3	y_1	x_3	//	x_2
	κ_a		0	0	2		1
	η_a		y_3	y_1	x_2		λ

TABLE 5.1 – Un exemple d'exécution de Swing++

de résolution de problèmes de satisfaction de contraintes – *Constraint Solving Problem (CSP)*. L'algorithme DisFC résout un CSP qui est la traduction d'un problème SM. Nous considérons ici que l'algorithme DisFC est une méthode de résolution décentralisée comme cela est communément admis pour ABT. Toutefois, on peut remarquer que cette méthode nécessite un ordonnancement entre les agents. Ainsi, certains agents capturent une grande partie de la complexité du problème et ceux-ci doivent résoudre une large partie (voire l'intégralité) du problème. Cette asymétrie peut impliquer des points de centralisation.

L'algorithme ZigZag (ZAVIDOVIQUE et al., 2005) consiste à parcourir une table dont les lignes (respectivement les colonnes) représentent les ordres de préférences des hommes (respectivement des femmes). Comme cet algorithme s'appuie sur cette table des mariages qui nécessite la connaissance complète du problème, cette méthode n'est pas distribuée. De plus, l'appariement obtenu n'est pas nécessairement stable.

SML2 (GELAIN et al., 2010) qui est un algorithme d'optimisation par recherche locale vise à minimiser le nombre de mariages non stables. Cet algorithme consiste, à partir d'un appariement aléatoire, à améliorer la stabilité de l'appariement en supprimant des mariages non stables. De par la notion de voisinage utilisée, cette méthode est centralisée. De plus, l'appariement obtenu n'est pas nécessairement stable.

Le problème qui consiste à trouver un appariement stable qui est optimal vis-à-vis du bien-être équitable ou égalitaire a été soulevé par KNUTH (1971). La plupart des algorithmes existants exploite la structure du treillis formé par les appariements stables et ils utilisent des méthodes de Théorie des Graphes comme l'algorithme en $\Theta(n^4)$ proposé par IRVING, LEATHER et GUSFIELD (1987).

5.7 Évaluation

L'ensemble des algorithmes évalués ici ont été implémentés en Java dans la librairie DSMP (PIETTE, MORGE & PICARD, 2014) et en Scala dans la librairie ScaMA (MORGE, 2014). Dans nos expérimentations, nous générons de manière pseudo-aléatoire des instances du problème SM. Les listes de préférences sont des permutations indépendantes de la liste des partenaires potentiels. Pour chaque taille de problème, nous présentons pour chaque métrique et pour chaque méthode la moyenne obtenue en fonction de la taille du problème. Nous commençons par évaluer Swing puis Swing++.

5.7.1 Évaluation de Swing

Ici, nous comparons Swing avec les méthodes décrites dans la section 5.6 : Gale-Shapley (GALE & SHAPLEY, 1962), ZigZag (ZAVIDOVIQUE et al., 2005), SML2 (GELAIN et al., 2010). On peut noter que les résultats présentés ne sont pas significativement sensibles à l'ordre des propositions, i.e. les résultats sont similaires si les femmes sont les premières proposantes.

Nos expérimentations confirment que l'algorithme GS orienté homme favorise les hommes par rapport aux femmes alors que Swing atteint des résultats similaires à ZigZag. Tandis que SML2 est légèrement meilleur que Swing pour le bien-être masculin, Swing est légèrement meilleur que SML2 pour le bien-être féminin. De plus, l'algorithme GS prône une communauté au détriment d'une autre alors que ZigZag, SML2 semblent plus équitables. Nos expérimentations confirment le fait que Swing est équitable.

En ce qui concerne le bien-être utilitaire, la méthode Swing est aussi bonne que ZigZag et SML2. Effectivement, les appariements stables qui sont optimaux ne sont pas nécessairement obtenus par l'algorithme GS, c.-à-d. des solutions mâle-optimales ou femelle-optimales. Nos expérimentations montrent ainsi que Swing améliore globalement la satisfaction de la société d'agents.

On vérifie également que SML2 ne garantit pas nécessairement la stabilité de la solution. ZigZag produit des appariements dit « quasi-stables ». Les autres méthodes produisent des solutions stables. Nous l'avons prouvé pour Swing. Nos expériences confirment ce résultat.

En ce qui concerne la complexité communicationnelle, Swing semble moins efficace que GS. C'est le prix de l'équité et de l'optimalité des solutions. Selon cette métrique, DisFC (BRITO & MEGUER, 2005) produit un nombre de messages bien supérieur à ceux générés par Swing dont la complexité communicationnelle semble approximativement n^3 (avec n individus).

En ce qui concerne la complexité algorithmique, Swing est moins performant que GS. C'est le prix de l'équité et de l'optimalité des solutions. Toutefois, si on le compare avec d'autres méthodes qui prônent l'équité, Swing est plus efficace que SML2 et un peu moins que ZigZag. Malheureusement, ces deux algorithmes ne garantissent pas la stabilité de la solution.

5.7.2 Évaluation de Swing++

Pour mémoire, Swing++ est une méthode qui a été conçue pour présenter les mêmes avantages que Swing et pour se terminer quelle que soit l'instance du problème considérée (même si nous ne disposons pas de preuve formelle à ce jour).

Pour évaluer le taux de terminaison de Swing et de Swing++, nous fixons expérimentalement un nombre maximal d'étape afin de nous assurer de la non-terminaison de Swing si celui-ci est atteint. On observe que la probabilité d'arrêt de la méthode Swing décroît avec la taille du problème. Alors que le taux de terminaison de Swing est élevé pour des problèmes de taille inférieure à 60 agents, le taux de terminaison est très faible pour des problèmes de grande taille. La probabilité qu'une instance fasse l'objet d'un dilemme ne semble pas linéaire par rapport à la taille du problème. Même si ce résultat ne constitue pas une preuve de terminaison, Swing++ se termine pour l'ensemble des 40 000 instances engendrées de manière (pseudo)-aléatoire.

Nos expérimentations indiquent que, comme SML2 et ZigZag, Swing++ est équitable contrairement à GS. De plus, les méthodes équitables sont plus optimales (du point de vue du bien-être utilitaire). En effet, les solutions optimales ne sont pas forcément explorées par GS. La différence entre les méthodes équitables et l'algorithme GS est significatif. Par contre, nous ne pouvons pas discriminer les méthodes équitables. Contrairement à Swing++, le temps d'exécution de SML2 explose avec la taille du problème. Toutefois, Swing++ est moins rapide que ZigZag ou GS. C'est le prix de l'équité (GS n'est pas équitable) et de la stabilité (les solutions de ZigZag ne sont pas stables). Swing++ est légèrement plus lent que Swing à cause de la résolution des dilemmes.

5.7.3 Synthèse

Comme le montre le tableau 5.2, Swing++ est la seule méthode permettant d'obtenir systématiquement un appariement équitable et optimal. Pour l'ensemble des 40 000 expériences réalisées, Swing++ se termine. Malheureusement, cette méthode n'est pas, comme GS ou Swing++, distribuable, c.-à-d. qu'elle ne peut pas être facilement implémentée à l'aide d'une plate-forme multi-agents. En effet, afin de s'assurer que l'appariement calculé soit stable, Swing++ vérifie lorsqu'on ajoute un mariage à l'appariement, que celui-ci reste stable.

Critère	GS	SML2	ZigZag	Swing	Swing++
Stabilité	✓	X	X	✓	✓
Terminaison	✓	X	✓	X	✓
Équité	X	✓	✓	✓	✓
Optimalité	X	✓	✓	✓	✓
Distribution	✓	X	X	✓	X

TABLE 5.2 – Synthèse pour la comparaison des méthodes de résolution pour SM

5.8 Problème des hôpitaux/résidents

Le problème des hôpitaux/résidents (HR) a été pour la première fois défini dans (GALE & SHAPLEY, 1962). Ce problème est une généralisation « un-à-plusieurs » du problème de mariage stable (cf section 5.2).

Une solution pour une instance de HR est une affectation de résidents à des postes dans chacun des hôpitaux. L'affectation d'un résident est un hôpital, éventuellement aucun. Les postes dans un hôpital sont des résidents, éventuellement l'ensemble vide. Évidemment, l'affectation est mutuelle. Un hôpital est sous chargé, plein ou surchargé si le nombre de résidents en poste est plus petit, égal ou plus grand que sa capacité, respectivement. Dans un appariement, aucun résident n'est affecté à un hôpital non-acceptable, aucun hôpital ne propose de poste à un résident non-acceptable, chaque résident est affecté à au plus un hôpital et aucun hôpital n'est surchargé.

Un appariement pour une instance de HR est stable s'il n'y a pas de couple dit bloquant qui menace l'appariement. Un couple bloquant préfère être assigné l'un à l'autre plutôt que selon l'appariement courant. Considérant un appariement, un couple hôpital-résident est bloquant si le résident préfère l'hôpital à son affectation courante et soit l'hôpital est sous chargé; soit il préfère le résident à au moins l'un de ses résidents en poste. Un appariement stable est appelé solution admissible.

L'algorithme GS proposé dans (GALE & SHAPLEY, 1962) s'adapte au problème HR pour constituer une preuve constructive de l'existence d'une solution admissible pour n'importe quelle instance : on parle de l'algorithme GS orienté résident (RGS). Dans RGS, chaque résident non-affecté, qui n'est pas désespéré (dont la liste de préférence n'est pas vide), envoie une proposition à l'hôpital qu'il préfère. Ce dernier accepte. S'il est surchargé, l'hôpital congédie le pire résident en poste. Si l'hôpital est plein,

il supprime de sa liste de préférence tous les résidents qui sont pires que les résidents en poste et réciproquement.

Une alternative à l'algorithme RGS, connu sous le nom d'algorithme de Gale-Shapley orienté hôpital (HGS) consiste à considérer les hôpitaux comme les proposant. Dans HGS, chaque hôpital, qui n'est pas désespéré (sous chargé avec au moins un résident dans sa liste de préférence n'étant pas en poste) se propose au résident qu'il préfère. Celui-ci accepte. Si ce dernier est déjà en poste dans un autre hôpital, il commence par démissionner. Ensuite, le résident supprime de sa liste de préférence tous les hôpitaux pires que son hôpital courant et réciproquement.

Dans (GUSFIELD & IRVING, 1989), Gusfield et Irving démontrent que toutes les solutions atteintes par RGS et HGS sont admissibles et que RGS (resp. HGS) atteint une solution résident-optimal (resp. hôpital-optimal). Les solutions admissibles d'une même instance de HR possèdent les propriétés suivantes : i) chaque hôpital possède le même nombre de résidents ; ii) les mêmes résidents sont non affectés ; iii) chaque hôpital sous chargé emploie les mêmes résidents. En d'autres termes, l'algorithme RGS se termine avec l'appariement qui est résident-optimal dans lequel chaque résident est affecté au meilleur hôpital auquel il peut prétendre dans n'importe quel appariement stable et chaque résident non-affecté est non-affecté dans tous les appariements stables. Cet appariement est le pire pour les hôpitaux. L'algorithme HGS se termine avec l'appariement stable hôpital-optimal dans lequel chaque hôpital plein est affecté avec ces meilleurs partenaires et chaque hôpital sous chargé est affecté au même ensemble de résidents dans tous les appariements stables. Dans cet appariement, chaque résident est affecté au pire hôpital auquel il peut prétendre dans tous les appariements stables.

5.9 Adaptation de Swing

Nous adaptons ici la méthode Swing (cf section 5.4) afin de résoudre le problème HR. Cette méthode permet d'atteindre des appariements stables qui ne peuvent pas être atteints avec les algorithmes RGS et HGS.

Dans les algorithmes RGS et HGS, chaque communauté est associée à un rôle : proposant ou disposant. Dans la méthode Swing, les agents (qu'ils soient des résidents ou des hôpitaux) jouent alternativement ces deux rôles dans de multiples négociations bilatérales à partir desquelles la solution émerge. Swing s'inspire du protocole de concession monotone présenté dans la section 2.2.3. À cette intention, les concessions des agents s'appuient sur le regret des agents. Chacun s'adresse à l'agent qu'il préfère. Si ce dernier refuse, l'agent concède en s'adressant au partenaire suivant dans sa liste de préférence. Pendant ce temps, les partenaires potentiels jouent le rôle de disposant en acceptant/refusant les propositions reçues au regard de leur niveau de concession. Quand tous les agents sont complètement affectés ou désespérés, la méthode s'arrête.

Dans la méthode Swing, les résidents et les hôpitaux se proposent alternativement. Au cours des étapes impaires, les hôpitaux jouent le rôle de proposant et les résidents le rôle de disposant. Au cours des étapes paires, les rôles sont inversés.

Chaque proposant envoie une demande à l'ensemble des individus acceptables. Ainsi quand une proposition est acceptée par un disposant :

1. si le proposant est affecté, il divorce ;
2. si le disposant est affecté, il divorce ;
3. le proposant et le disposant se marient ;
4. le niveau de concession du proposant et celui du disposant sont modifiés de sorte qu'ils ne peuvent divorcer que pour de meilleurs partenaires.

On peut noter que, à chaque étape, un proposant cesse d'envoyer des propositions dès qu'il est affecté. Si tous les répondants rejettent les propositions d'un proposant, alors celui-ci concède. Quand

tous les proposant sont affectés ou désespérés, la méthode Swing se termine. Un agent est désespéré si c'est un hôpital (resp. un résident) qui est sous-chargé (resp. non-affecté) et qu'il a atteint son niveau de concession maximal (la liste de préférence a été complètement explorée).

À chaque étape, le proposant et le disposant peuvent divorcer. Si un membre d'un nouveau couple est affecté, les précédents (ou l'un des précédents) partenaires sont libérés et donc concèdent.

5.10 Application pratique

Dans cette section, nous nous intéressons à l'affectation des stages aux Professeurs des Écoles (PE) dans l'Académie de Lille. Pour compléter la formation pratique des Professeurs des Écoles, l'IUFM³ (Institut Universitaire de Formation des Maîtres) répartit les stagiaires dans les différents établissements en fonction de leur desiderata et des contraintes des titulaires.

Pour faire face à l'augmentation du nombre de recrutement, l'IUFM souhaite automatiser cette procédure. Nous montrons ici que ce problème d'affectation peut être résolu à l'aide des méthodes présentées précédemment.

Les différentes données du problème sont représentées en figure 5.3 dans un Modèle Conceptuel des Données (MCD). Lors de sa formation à l'IUFM, chaque stagiaire doit effectuer trois stages, un par trimestre. Réciproquement, chaque titulaire suffisamment qualifié pour encadrer ces stages peut accueillir au plus deux stagiaires par trimestre. L'Académie est divisée en bassin. Chaque bassin regroupe un ensemble de villes dans lesquelles se trouvent les écoles. Pour être affecté, chaque stagiaire doit exprimer deux vœux, c.-à-d. sélectionner et classer deux bassins dans lesquels il souhaite effectuer ses stages. On retrouve en figure 5.3 ce découpage administratif et les vœux des stagiaires.

Les encadrants enseignent dans une classe qui regroupe un ou plusieurs niveaux, appelés sections qui sont regroupées en cycle. L'enseignement du premier degré (maternelle/élémentaire) est divisé en trois cycles⁴ :

- le cycle 1 comprenant la très petite section (noté TPS), la petite section (noté PS) et la moyenne section (noté MS) ;
- le cycle 2 comprenant la grande section (noté GS), le cours préparatoire (noté CP) et le cours élémentaire 1 (noté CE1) ;
- le cycle 3 comprenant le cours élémentaire 2 (noté CE2), le cours moyen 1 (noté CM1) et le cours moyen 2 (noté CM2).

Quand un titulaire enseigne dans l'une des sections d'un cycle, on considère qu'il enseigne dans ce cycle. Cependant, il existe un cas particulier concernant la section GS qui si elle est associée à MS ou PS est considérée comme cycle 1. Dans la figure 5.3, on retrouve l'école associée à un titulaire à l'aide de l'association `Travaille` mais également les sections dans lesquelles il enseigne avec l'association `Enseigne`.

L'objectif du problème est d'affecter un stage par trimestre à chaque stagiaire en respectant ses vœux comme le montre l'association `Affectation` de la figure 5.3. De plus, l'Académie de Lille veut minimiser les remboursements kilométriques des stages, c'est la raison pour laquelle nous avons ajouté dans le MCD les informations relatives aux distances. Nous avons fait le choix de différencier les domiciles des stagiaires, des villes contenant les écoles afin de minimiser le nombre de distances à calculer.

La priorité est donnée aux stagiaires ayant le plus d'enfants, puis à ceux travaillant à temps partiel et enfin à ceux n'ayant pas de voiture. Réciproquement, le niveau de qualification des titulaires (EMF, MAT CAFIPEMF ou MAT) permet de les prioriser. Finalement, l'Académie souhaite minimiser le nombre d'encadrants pour réduire les frais de gestions liés à leur dédommagement.

3. Les IUFM ont été remplacés en 2013 par les Écoles Supérieures du Professorat et de l'Éducation (ESPE).

4. Cette subdivision a été modifiée par le bulletin officiel spécial n°2 du 26 mars 2015.

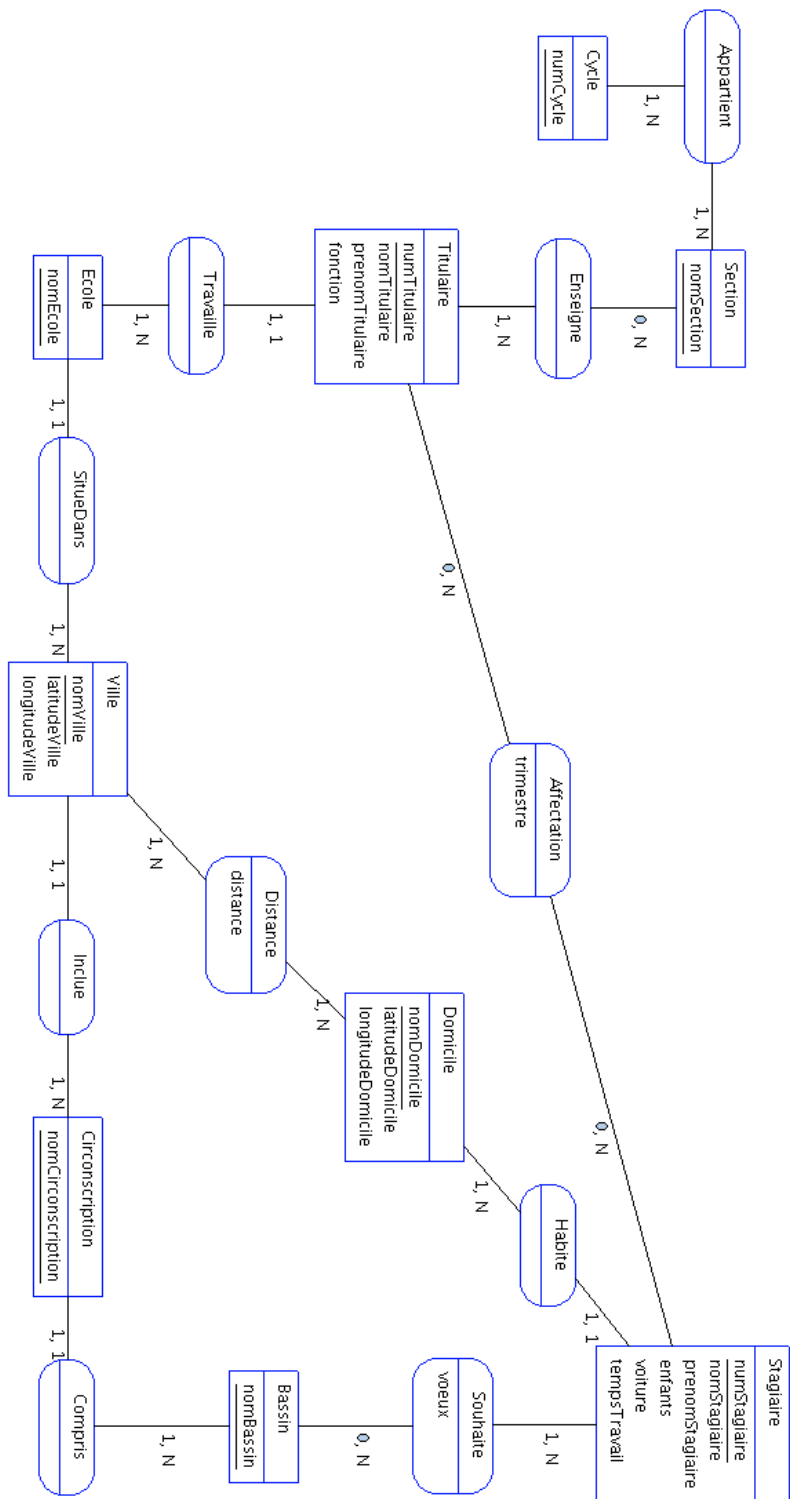


FIGURE 5.3 – MCD représentant les entrées et les sorties de la résolution d'un problème PE

Un problème PE se transforme en trois problèmes HR où chaque problème correspond à un trimestre de PE. Afin de réaliser l'affectation des stagiaires pour un trimestre, il est nécessaire de générer le problème HR correspondant et de le résoudre.

Considérons une instance d'un problème PE. Afin de déterminer l'appariement trimestriel, nous transformons ce problème PE en un problème HR pour en calculer un appariement stable avec les algorithmes présentés dans la section précédente. Afin de générer le problème HR correspondant, nous considérons la mise en correspondance illustrée dans le tableau 5.3.

PE	HR
Titulaire	Hôpital
Stagiaire	Résident
Préférences et contraintes	Listes de préférences
Capacité d'accueil d'un titulaire	Capacité d'un hôpital

TABLE 5.3 – Correspondance entre PE et HR

Lors de la génération du problème HR correspondant, nous établissons les listes de préférences des hôpitaux et des résidents afin de répondre aux exigences de l'IUFM et de respecter la priorisation des stagiaires et des titulaires.

La résolution d'un problème PE traite le problème par trimestre et initialise chaque appariement trimestriel. Pour chaque trimestre, un problème HR est généré puis résolu. L'appariement obtenu permet d'établir l'appariement trimestriel correspondant. Initialement, la correspondance entre les hôpitaux et les titulaires et entre les résidents et les stagiaires est réalisée. Par la suite on génère les listes de préférences des résidents et des titulaires.

Pour générer la liste de préférences d'un résident, on initialise la liste de préférences d'un résident avec les hôpitaux correspondants à ces vœux. Ensuite ces hôpitaux sont répartis par cycle. Les hôpitaux d'un même cycle sont triés par vœu, par diplôme et enfin par distance. Ces différentes listes sont alors concaténées dans la liste de préférences du résident afin de favoriser les cycles qui n'ont pas encore été réalisés. Finalement, on supprime de ses préférences l'ensemble des hôpitaux qui correspondent à des titulaires enseignants à des sections déjà pratiquées.

Pour effectuer le tri d'une liste d'hôpitaux correspondants à des titulaires d'un même cycle, on commence par répartir chaque hôpital dans deux listes correspondant à chacun des vœux. Chacune de ces listes est alors triée en fonction des qualifications des titulaires correspondants. Ensuite, les hôpitaux sont triés en fonction de la distance entre le domicile du stagiaire et l'école du titulaire correspondant. Finalement, les différentes listes sont concaténées.

Pour générer des listes de préférences pour les hôpitaux, cette liste de préférences est initialisée avec les résidents qui le considèrent. Les résidents sont classés en fonction du nombre d'enfants du stagiaire correspondant, puis en fonction de leur temps de travail et finalement en fonction du fait qu'il possède (ou non) un véhicule. Finalement, ces listes sont concaténées.

5.11 Expérimentations

Nous considérons ici la campagne 2012 d'affectation de stagiaires au sein de l'Académie de Lille. Lors de cette campagne, 356 stagiaires doivent trouver trois stages, chacun encadré par l'un des 783 titulaires. Nous avons transformé cette instance du problème PE en trois problèmes HR (comme décrit dans la section précédente) puis nous l'avons résolu avec l'algorithme RGS, l'algorithme HGS et la méthode Swing. On peut remarquer qu'aucune méthode d'optimisation comme SMP2 (GELAIN et al., 2010) ou DisFC (BRITO & MESEGUER, 2005) ne peut être appliquée pour des raisons de scalabilité. Nos trois méthodes permettent de résoudre ce problème en une vingtaine de minutes. 80 % du temps

de calcul est utilisé pour la génération des listes des préférences qui sont différentes pour chaque trimestre car : i) les encadrants sont triés selon les vœux des candidats, leur diplôme, la distance et les cycles non pratiqués ; iii) les stagiaires sont triés selon les enfants, le temps partiel et les voitures.

Comme dit précédemment, la priorité est donnée aux stagiaires ayant le plus d'enfants, puis à ceux travaillant à mi-temps et enfin à ceux n'ayant pas de voiture. En 2012, peu de stagiaires avaient des enfants. C'est la raison pour laquelle nous ne pouvons pas vérifier que la distance décroît quand le nombre d'enfants croît. Si on considère seulement les stagiaires n'ayant pas d'enfants, il semble que ceux travaillant à mi-temps sont favorisés. Comme les stagiaires n'ayant pas de voiture représentent les deux tiers de la cohorte, la règle correspondante ne peut pas être appliquée quelle que soit la méthode employée.

Le tableau 5.4 résume et compare les résultats obtenus par les trois méthodes. ++, +, et - signifient qu'un critère est complètement, partiellement ou n'est pas respecté, respectivement. D'après nos expérimentations, aucune méthode ne semble satisfaire tous les critères. Alors que RGS semble prioriser les stagiaires (couverture des cycles, distance moyenne, diplôme), HGS semble plus adapté pour prioriser les encadrants (i.e. nombre d'enfants, temps partiel, voiture). La méthode Swing pondère ces deux aspects. En réalité, cette méthode relaxe la contrainte relative aux souhaits. Comme cette contrainte est relâchée, Swing permet d'accroître la couverture des cycles et de décroître le nombre d'encadrants et donc le coût de gestion.

Critère	RGS	HGS	Swing
3 stages	++	++	++
3 cycles	+	-	++
Souhaits	+	+	-
Distance	++	-	+
Nb. d'encadrants	-	+	++
Priorité au diplôme	++	+	?
Priorité aux enfants	?	?	?
Priorité au temps partiel	+	+	+
Priorité au « sans voiture »	-	-	-

TABLE 5.4 – Comparaison des méthodes pour résoudre le problème PE

5.12 Conclusion

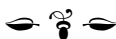
Dans ce chapitre, nous avons adopté une approche centrée individu pour la résolution du problème des mariages stables. L'algorithme distribuable de Gale-Shapley apporte une preuve de l'existence d'une solution stable pour chaque instance et il est distribuable. Toutefois, son asymétrie ne garantit pas d'atteindre de solutions équitables et optimales pour l'ensemble de la société. La méthode Swing, que nous avons étudiée ici, prône l'équité en échangeant les rôles des agents à chaque étape. De plus, les individus négocient à l'aide du protocole de concession monotone (cf section 2.2.4) afin d'obtenir un partenaire qui correspond à leurs préférences. Comme nous l'avons souligné ici, ces changements de rôles peuvent donner lieu à des cycles d'exécution de Swing. Alors que les hommes privilégient les solutions mâle-optimales lorsqu'ils sont proposants, les femmes privilégient les solutions femelle-optimales lorsque c'est à elles de proposer. Dans ce chapitre, nous avons étendu Swing pour détecter ces dilemmes et réintroduire de l'asymétrie pour atteindre systématiquement une solution sans nuire à l'équité.

Nous avons fait l'hypothèse que les relations de préférence sont des ordres stricts. Plus directement on peut considérer que les membres d'une des communautés ne sont plus appariés avec un

mais plusieurs partenaires. Grâce à cette adaptation, nous adressons un problème réel PE, celui de l'affectation des stages aux professeurs des écoles.

Ayant présenté les entrées et les sorties de ce problème, nous avons montré que celui-ci peut être capturé par trois instances d'un problème bien connu : celui des hôpitaux/résidents (HR). Nous avons présenté deux algorithmes qui permettent d'aboutir à des solutions stables pour ce problème abstrait où l'on suppose que les individus classent leurs partenaires potentiels : l'algorithme GS orienté résident (RGS) et l'algorithme de Gale-Shapley orienté hôpital (HGS). De plus nous avons adapté la méthode Swing au problème HR. La transformation d'un problème PE en trois problèmes HR nécessite la génération de listes de préférence complexes. La liste de préférence des résidents prend en compte les vœux du stagiaire correspondant, du diplôme des encadrants, de la distance entre eux, etc. La liste de préférence des hôpitaux prend en compte les priorités des stagiaires en fonction de leur nombre d'enfants, du fait qu'ils travaillent à mi-temps (ou pas) et du fait qu'ils possèdent une voiture (ou pas). Afin de résoudre le problème PE, nous avons essayé d'utiliser les trois méthodes précédemment citées. Nos expérimentations sont en adéquation avec le fait que Swing soit plus équitable. Cette méthode équilibre les critères prônés par les résidents (e.g. la distance moyenne) et les critères prônés par les hôpitaux (i.e la priorité entre les stagiaires). Comme la contrainte qui réside dans la satisfaction du premier vœu de chaque stagiaire a été relâchée, la méthode Swing décroît le nombre d'encadrants et donc le coût de gestion.

Nous avons vu ici que la difficulté majeure réside dans la transformation d'un problème concret en un problème abstrait. Cette transformation représente 80 % du temps de calcul et probablement 80 % du temps de travail consacré à ce problème. Toutefois, ce problème réel nous a permis de valider expérimentalement la pertinence de la méthode Swing. On peut remarquer qu'il est plus qu'improbable que deux tiers des stagiaires ne disposent pas de voiture. Il est à parier que ceux-ci mentent dans le but de défendre leur intérêt individuel. Malheureusement, ces mensonges empêchent, quelle que soit la méthode employée, de favoriser les stagiaires sans voiture.



Deuxième partie

Travaux en cours et perspectives

CHAPITRE 6

Négociations bilatérales concurrentes

Sommaire

6.1	Introduction	69
6.2	Affectation d'individus à des activités	70
6.3	Procédure centralisée d'appariement	73
6.4	Comportement d'agent	75
6.5	Évaluation empirique	76
6.6	Application pratique	78
6.7	Conclusion	78

Les travaux présentés dans ce chapitre ont été réalisés entre 2016 et 2018 dans le cadre du projet PartENS (Partir ENsemble) soutenu par le programme chercheur-citoyen de la région Nord Pas de Calais en collaboration avec Antoine NONGAILLARD et Yann SECQ (Université de Lille). Ils ont notamment fait l'objet des publications suivantes :

- MORGE, M. & NONGAILLARD, A. (2018a). Distributed Algorithm for Egalitarian Matching between Individuals and Activities with Additively Separable Preferences. In *IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)* (p. 739-746). B (core)
- MORGE, M. & NONGAILLARD, A. (2018b). Procédure décentralisée d'affectation d'individus à des activités. *Revue d'Intelligence Artificielle*, 32(5-6), 627-658

6.1 Introduction

Nos travaux s'inscrivent dans un projet de recherche plus large qui vise à comprendre et modéliser les dynamiques de rétroaction se produisant dans un collectif en interaction à la fois au sein d'un réseau social virtuel et d'un réseau social réel. Le cas d'usage sur lequel nous nous concentrons ici concerne un collectif de seniors qui entrent en relation afin d'effectuer des activités ponctuelles. Ce terrain d'expérimentation implique plusieurs milliers d'individus. L'objectif est ici de maximiser les activités partagées afin d'améliorer la cohésion sociale et de lutter contre l'isolement chez certains seniors. Nous souhaitons proposer un réseau social numérique afin que les utilisateurs forment des groupes pour pratiquer ensemble des activités. Ce système a pour objectif de suggérer à chaque utilisateur des individus avec lesquels pratiquer ces activités.

Dans ce chapitre, nous proposons une modélisation multi-agents pour la formation de coalitions qui correspond à notre cas d'usage. Un ensemble d'individus doivent être appariés pour pratiquer

une des activités qui leur est proposée en fonction des préférences des individus pour leurs alter-egos et pour les activités. Nous définissons des règles de décision pour l'élicitation des préférences individuelles et collectives vis à vis des coalitions en terme d'efficacité économique, d'équité, de stabilité et de cohésion sociale. En s'appuyant sur la structure bilatérale du problème formalisé, nous proposons ici deux procédures qui retournent des appariements où aucune activité n'est surchargée. La première procédure, dite « sélective », vise à maximiser la satisfaction globale des individus, i.e. le bien-être utilitaire. La seconde procédure, dite « inclusive », s'assure que tous les individus sont affectés autant que possible à une activité qu'ils préfèrent. La modélisation multi-agents nous permet de distribuer l'exécution de ces algorithmes décrits sous la forme de comportements d'agent qui consistent en de multiples négociations bilatérales concurrentes. Nos expérimentations montrent que les solutions atteintes par ces procédures sont de meilleure qualité en terme d'utilitarisme (respectivement d'égalitarisme) que celles obtenues avec des techniques classiques de recherche locale. De plus, la décentralisation des procédures d'appariement permet d'accélérer leur exécution.

Nous introduisons notre problème d'affectation dans la section 6.2. Nous proposons différentes procédures d'affectation dans la section 6.3. La distribution de ces algorithmes est présentée sous la forme de comportements d'agents dans la section 6.4. La section 6.5 exhibe nos résultats empiriques. Après un bref retour d'expérience sur notre application pratique (cf section 6.6), nous synthétisons nos travaux puis nous dressons quelques perspectives (cf section 6.7).

6.2 Affectation d'individus à des activités

Nous nous abstrayons ici de l'application pratique visée pour introduire un cadre formel pour la formation de coalition qui modélise dans notre cas d'usage : le problème, les solutions potentielles et leur qualité d'un point de vue individuel et collectif.

Nous introduisons ici le problème des individus/activités (IA). Dans une instance de problème IA, on considère n activités, chacune ayant une capacité maximale, et m individus, chacun étant muni d'une fonction d'intérêt pour évaluer les activités et d'une fonction d'affinité pour évaluer ses pairs. Les individus privilégient les activités qui leur plaisent avec les partenaires qu'ils apprécient. Intuitivement, l'activité nulle, qui correspond à ne rien faire, a une capacité infinie : tous les individus peuvent y être affectés quitte à être seul.

Les fonctions d'intérêt (respectivement d'affinité) permettent d'exprimer la répulsion, l'attraction, voire l'indifférence d'un individu vis-à-vis des activités (respectivement de ses pairs). Par convention, l'intérêt d'un individu pour l'activité nulle est neutre. On déduit de chaque fonction d'intérêt une relation de préférence individuelle sur les activités, c'est-à-dire une relation d'ordre réflexive, totale et transitive. De même, on déduit de chaque fonction d'affinité une relation de préférence individuelle sur les groupe de pairs, c'est-à-dire une relation d'ordre. La satisfaction pour un individu d'appartenir à un groupe pour pratiquer une activité dépend : (i) de son intérêt pour l'activité ; (ii) de son affinité vis-à-vis du groupe.

Nous souhaitons former des coalitions autour des activités. À cette intention, nous considérons une instance de problème où le nombre d'individus est considérablement plus important que le nombre d'activités. Par définition, l'affectation d'un individu est une activité, éventuellement l'activité nulle, et chaque individu est associé à un groupe auquel il appartient. Tous les individus qui sont affectés à l'activité nulle sont seuls. Tous les individus associés les uns aux autres ont la même activité et réciproquement tous les individus qui sont affectés à la même activité, à l'exception de l'activité nulle, sont associés les uns aux autres.

Exemple 9 *Nous considérons ici dans un cas jouet quatre individus (1, 2, 3 et 4) qui souhaitent faire du passing avec six objets. Disposant de six massues et de six balles, deux activités sont disponibles : les massues (a) et les balles (b). Chaque activité peut être réalisée par au plus deux individus. Selon*

	a	b	j =	1	2	3	4
v_1	.50	.25	$w_1(j)$		1.0	-0.5	-1.0
v_2	.50	.25	$w_2(j)$	1.0		0.5	-1.0
v_3	.50	.25	$w_3(j)$	1.0	0.5		-1.0
v_3	.50	.25	$w_4(j)$	1.0	1.0	-1.0	

$a \succeq_1 b$	$\{2, 1\} >_1 \{1\}$	$\{1\} >_1 \{3, 1\} >_1 \{4, 1\}$
$a \succeq_2 b$	$\{1, 2\} >_2 \{3, 2\} >_2 \{2\}$	$\{2\} >_2 \{4, 2\}$
$a \succeq_3 b$	$\{1, 3\} >_3 \{2, 3\} >_3 \{3\}$	$\{3\} >_3 \{4, 3\}$
$a \succeq_4 b$	$\{1, 4\} >_4 \{2, 4\} >_4 \{4\}$	$\{4\} >_4 \{3, 4\}$

FIGURE 6.1 – Préférences cardinales (en haut) et les préférences ordinales déduites (en bas) vis-à-vis des activités (à gauche) et des pairs (à droite).

l'intérêt des individus pour les activités et leur affinité pour leur pairs (cf figure 6.1), tous les individus préfèrent jongler avec les massues plutôt qu'avec les balles mais pas avec n'importe qui. 1 et 2 s'apprécient, ce n'est pas le cas pour tous. Par exemple, 3 et 4 aiment faire du passing ensemble avec des massues mais ils préfèrent rester seuls plutôt que faire du passing ensemble avec des balles. Au delà de la combinatoire du problème (63 appariements sont possibles dans notre exemple), la recherche d'une solution harmonieuse est un système complexe.

Chaque individu évalue la coalition à laquelle il appartient et donc l'appariement en fonction de son groupe et de son activité. Un individu considère qu'une coalition valide pour lui est individuellement rationnelle si et seulement si elle est attirante. L'existence d'un appariement valide IR est garantie.

Nous sommes maintenant en mesure d'évaluer les appariements valides du point de vue du collectif. Une première propriété désirable est la stabilité. Intuitivement, les individus d'une coalition dite « bloquante » souhaitent se séparer pour former leur propre coalition, ce qui rend l'appariement sous-jacent instable. Une coalition C valide et non-vide bloque fortement un appariement M si et seulement si tous les individus de la coalition la préfèrent strictement plutôt que d'être affectés selon M . Un appariement valide est stable de cœur (noté CS) si et seulement s'il n'y a pas de coalition valide et non-vide qui le bloque fortement. Par définition, un appariement qui est stable de cœur est individuellement rationnel.

Une coalition C valide et non-vide bloque faiblement un appariement M si et seulement si tous les individus de la coalition la préfèrent plutôt que d'être affectés selon M et au moins un individu de la coalition la préfère strictement. Un appariement valide est strictement stable de cœur (noté SCS) s'il n'y a pas de coalition valide et non-vide qui le bloque faiblement. Par définition, tous les appariements SCS sont *a fortiori* CS.

Un appariement valide M est stable au sens de Nash (noté NS) si aucun individu n'a intérêt à dévier unilatéralement vers une autre activité sous-chargée. Un tel appariement est à l'abri des mouvements individuels car les coalitions sont rationnelles pour tous les individus et au moins aussi bonnes que les autres coalitions valides atteignables par un seul mouvement.

Un appariement valide M est individuellement stable (noté IS) si et seulement si aucun individu n'a intérêt à dévier unilatéralement vers une autre coalition valide ou si c'est le cas, l'individu qui dévie n'est pas unanimement apprécié par ses nouveaux partenaires. Par définition, un appariement IS est *a fortiori* NS.

Un appariement valide M est contractuellement individuellement stable (noté CIS) si et seulement aucun individu ne peut bénéficier de son seul mouvement d'une coalition à une autre coalition sans aggraver la satisfaction de l'une ou de l'autre. Par définition, un appariement CIS est *a fortiori* IS.

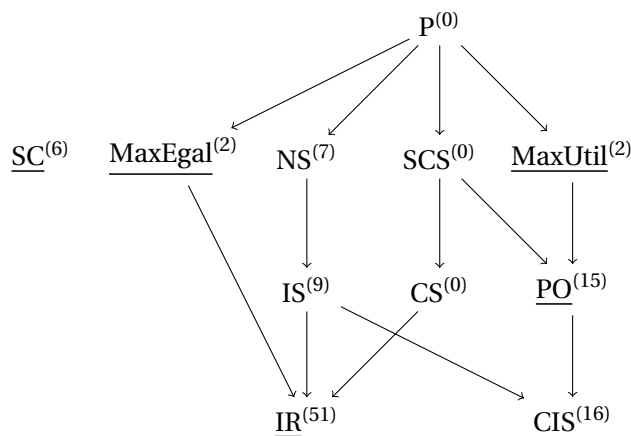


FIGURE 6.2 – Relations d’inclusion entre les concepts de stabilité, d’optimalité, d’équité et de cohésion pour le problème IA. L’existence des concepts soulignés est garantie. Les exposants indiquent la distribution des 63 appariements valides pour notre exemple (cf figure 6.1) en fonction de leurs propriétés.

On peut remarquer que, contrairement à un appariement IS, un appariement CIS n’est pas nécessairement IR.

Nous formulons maintenant d’autres propriétés désirables pour mesurer l’efficacité économique d’un appariement. Comme les précédentes, elles ne s’appuient pas sur des comparaisons inter-personnelles. Un appariement valide M est parfait (noté P) si la coalition pour chaque individu est l’une de ses préférées. Un appariement M' domine au sens de Pareto M si et seulement si M' est strictement préféré à M pour au moins un individu et pas pire pour tous les autres. Un appariement est optimal au sens de Pareto (noté PO) s’il n’est pas dominé au sens de Pareto par un appariement valide. En d’autres termes, un appariement est optimal au sens de Pareto s’il n’existe pas d’alternative valide pour laquelle tous les agents sont dans une position meilleure ou équivalente. La stabilité stricte de cœur est une condition suffisante pour l’optimalité au sens de Pareto.

Contrairement à l’approche parétienne, la théorie du choix social analyse les comparaisons inter-personnelles (SEN, 1970). Cette théorie réconcilie l’utilitarisme de Jeremy Bentham et la justice distributive prônée par John Rawls (MOULIN, 2002). Plus les bien-être sont élevés, plus l’appariement est bon. Un appariement valide est un maximum utilitariste (respectivement maximum égalitariste) - noté MaxUtil (respectivement MaxEgal) s’il maximise le bien-être utilitaire (resp. égalitaire). On peut noter que, si tous les individus et toutes les activités sont attractifs, le bien-être égalitaire est nul dès qu’un individu est inactif. Par définition, un appariement valide MaxUtil est *a fortiori* PO. De même, un appariement valide MaxEgal est *a fortiori* IR.

Finalement, en s’inspirant de la psychologie sociale (ZACCARO & LOWE, 1988), nous pouvons considérer que la cohésion est plus étroitement liée à l’intérêt des activités qu’aux affinités interpersonnelles. Un appariement valide M est socialement cohésif (SC) si et seulement si tous les individus sont affectés autant que faire se peut à une activité qu’ils préfèrent. L’existence d’un appariement valide SC est garantie, e.g. l’appariement trivial quand toutes les activités sont rebutantes. Ce concept est orthogonal aux précédents, et contrairement à la plupart des autres, un appariement valide SC n’est pas nécessairement valide.

Inspirée par AZIZ, BRANDT et SEEDIG (2013), la figure 6.2 représente sous la forme d’un treillis les relations d’inclusion entre les concepts présentés dans cette section. Par exemple, l’arc entre SCS et PO indique que le premier concept est une condition nécessaire du second mais pas suffisante.

Comme illustré par notre exemple, si la stabilité est une propriété désirable, il n’existe pas néces-

sairement de telles solutions. De plus, l'optimalité au sens de Pareto ne semble pas être suffisamment discriminante. En revanche, les concepts de cohésion et de bien-être sociaux sont de meilleurs candidats pour évaluer la qualité d'un appariement du point de vue de la société d'individus.

6.3 Procédure centralisée d'appariement

En s'appuyant sur la structure bilatérale du problème IA, nous proposons ici deux algorithmes qui retournent des appariements valides.

Nos algorithmes s'inspirent de l'algorithme d'acceptation différée orienté individu (GALE & SHAPLEY, 1962) - en anglais, *student-proposing deferred acceptance algorithm*. Chaque individu est associé à une liste de concessions, i.e. la liste des activités qui l'attirent classées par intérêt décroissant. D'un côté, un individu se propose aux activités par ordre de préférence quitte à concéder en cas de rejet. De l'autre côté, la coalition en tant qu'hôte, explicite sa contrainte de capacité et les objectifs des individus hébergés, c'est-à-dire une agrégation des bien-être individuels qui peut être ici : soit une règle sociale utilitaire; soit une règle sociale égalitaire.

Dans le premier algorithme, l'appariement initial est l'appariement trivial, i.e. tous les individus sont inactifs. Tour à tour, chaque individu libre i considère l'activité qui l'attire et qu'il préfère a . Si la coalition est vide, i est affecté. Sinon, cet algorithme tente d'améliorer la satisfaction de cette coalition, éventuellement en excluant les individus dont la présence pénalise la règle sociale (utilitaire ou égalitaire) du groupe. Les individus qui sont exclus, éventuellement i , doivent concéder et donc considérer l'activité attractive suivante. On peut noter que cet algorithme est dit sélectif car la règle sociale est évaluée à chaque proposition d'intégration à un groupe existant. Tant que la capacité de a n'est pas atteinte alors le groupe peut croître. Finalement, un individu qui est rejeté par toutes ses activités attrayantes est définitivement inactif.

Comme un groupe de k individus comporte $2^k - 1$ sous-groupes non-vides, nous proposons un algorithme d'approximation qui consiste à n'exclure qu'au plus un individu à chaque étape. Notre algorithme sélectif (éventuellement d'approximation) appliqué à une instance quelconque d'un problème IA se termine et l'appariement retourné est valide.

Dans le cas général, on peut noter que l'algorithme sélectif ne maximise pas nécessairement le bien-être utilitaire (respectivement égalitaire) en appliquant la règle sociale utilitaire (respectivement égalitaire) car les individus donnent la priorité à l'intérêt des activités plutôt qu'à l'affinité des groupes. Malgré cette priorité, l'appariement résultant n'est pas nécessairement socialement cohésif. C'est la raison pour laquelle nous proposons l'algorithme inclusif. Contrairement à l'algorithme précédent, la règle sociale d'une coalition, en tant qu'hôte, n'est évaluée que lorsque sa contrainte de capacité est violée et l'exclusion d'un seul individu est envisagée à chaque intégration. L'algorithme inclusif appliqué à une instance quelconque d'un problème IA se termine et l'appariement retourné est valide et socialement inclusif.

Exemple 10 *Considérons le problème représenté dans la figure 6.1. Quelle que soit la règle sociale adoptée, notre algorithme sélectif retourne l'appariement où 1 et 2 pratiquent a , 4 pratique b et 3 reste seul à ne rien faire. Cet appariement est Pareto-optimal mais pas socialement cohésif. La partie gauche de la figure 6.3 contient une trace d'exécution de cet algorithme.*

Quelle que soit la règle sociale adoptée, l'algorithme inclusif retourne l'appariement où 1 et 2 pratiquent a alors que 3 et 4 pratiquent b . Cet appariement est socialement cohésif mais pas individuellement rationnel. La partie droite de la figure 6.3 contient une trace d'exécution de cet algorithme.

```

1 M: a(2): [] b(2): [] θ: [1, 2, 3, 4]
2 Comme a est libre,
3 4 est affecté à a
4 M: a(2): [4] b(2): [] θ: [1, 2, 3]
5 3 choisit a avec [4]
6 [3].usum(a)=0,250
7 [4].usum(a)=0,250
8 [4, 3].usum(a)=0,167
9 Comme [3] est choisi,
10 4 est exclu de a
11 3 est affecté à a
12 M: a(2): [3] b(2): [] θ: [1, 2, 4]
13 2 choisit a avec [3]
14 [2].usum(a)=0,250
15 [3].usum(a)=0,250
16 [3, 2].usum(a)=0,667
17 Comme [3, 2] est choisi,
18 2 est affecté à a
19 M: a(2): [2, 3] b(2): [] θ: [1, 4]
20 1 choisit a avec [2, 3]
21 [3, 2].usum(a)=0,667
22 [1].usums(a)=0,250
23 [3].usum(a)=0,250
24 [2, 1].usum(a)=0,833
25 [2].usum(a)=0,250
26 [3, 1].usum(a)=0,583
27 Comme [2, 1] est choisi,
28 3 est exclu de a
29 1 est affecté à a
30 M: a(2): [1, 2] b(2): [] θ: [3, 4]
31 Comme b est libre,
32 3 est affecté à b
33 M: a(2): [1, 2] b(2): [3] θ: [4]
34 4 choisit b avec [3]
35 [4].usum(b)=0,125
36 [3].usum(b)=0,125
37 [3, 4].usum(b)=-0,083
38 Comme [4] est choisi,
39 3 est exclu de b
40 4 est affecté à b
41 M: a(2): [1, 2] b(2): [4] θ: [3]
42 3 est définitivement inactif
43 M: a(2): [1, 2] b(2): [4] θ: [3]

1 a(2): [] b(2): [] θ: [1, 2, 3, 4]
2 4 choisit a avec []
3 Since a n'est pas pleine,
4 4 est affecté à a
5 a(2): [4] b(2): [] θ: [1, 2, 3]
6 3 choisit a avec [4]
7 Since a n'est pas pleine,
8 3 est affecté à a
9 a(2): [3, 4] b(2): [] θ: [1, 2]
10 2 choisit a avec [3, 4]
11 [4, 3].umin(a)=0,083
12 [4, 2].umin(a)=0,083
13 [3, 2].umin(a)=0,333
14 Since [3, 2] est choisi,
15 4 est exclu de a
16 2 est affecté à a
17 a(2): [2, 3] b(2): [] θ: [1, 4]
18 1 choisit a avec [2, 3]
19 [3, 2].umin(a)=0,333
20 [3, 1].umin(a)=0,167
21 [2, 1].umin(a)=0,417
22 Since [2, 1] est choisi,
23 3 est exclu de a
24 1 est affecté à a
25 a(2): [1, 2] b(2): [3] θ: [4]
26 4 choisit b avec [3]
27 Since b n'est pas pleine,
28 4 est affecté à b
29 a(2): [1, 2] b(2): [3, 4] θ: []

```

FIGURE 6.3 – Trace d'exécution de l'algorithme sélectif exact avec la règle utilitaire (à gauche) et pour l'algorithme inclusif avec la règle égalitaire (à droite).

6.4 Comportement d'agent

Nous décrivons ici les procédures d'appariement présentées précédemment sous la forme de comportements d'agent afin de distribuer leur exécution.

Nous adoptons le modèle de programmation concurrente par passage de messages asynchrones, appelé modèle d'acteur, proposé dans (HEWITT, BISHOP & STEIGER, 1973). Afin de proposer un moteur d'appariement distribué à partir de ce modèle, nous distinguons trois types d'agents :

1. l'agent « solveur » qui crée les autres agents et enregistre les affectations;
2. les agents « individu » tous munis d'un même comportement mais avec des préférences hétérogènes;
3. les agents « coalition » tous munis d'un même comportement mais qui gèrent des contraintes de capacité différentes et des objectifs de groupe dissemblables.

Le comportement de l'agent « solveur » consiste à : i) créer les agents; ii) lancer la résolution; iii) enregistrer les affectations et les exclusions; iv) et renvoyer l'appariement quand tous les individus sont affectés ou définitivement inactifs.

Que la procédure soit sélective ou inclusive, le comportement d'un agent « individu » est identique. Il consiste à se proposer à l'activité préférée qui l'attire. Quand l'agent est affecté ou exclu, il en informe l'agent « solveur ». Comme nous le verrons dans la suite, lors d'une exclusion, l'agent « coalition » attend une confirmation de la part de l'agent exclu avant de procéder effectivement à l'intégration d'autres agents afin que l'appariement ne soit pas retourné prématurément par l'agent « solveur ». Si un agent « individu » devient libre, il concède, puis il se propose à l'activité suivante qui l'attire jusqu'à être définitivement inactif. Conformément au principe de subsidiarité, la reconfiguration d'une coalition peut être soumise à l'arbitrage de l'agent « coalition » qui requiert l'opinion des agents « individu » concernés.

Afin de décrire le comportement d'un agent « coalition », nous utilisons le patron de conception de programmation concurrente FSM (*Finite State Machine*) avec trois états :

1. l'état initial de disposant (*Disposing*), où chaque proposition est traitée;
2. l'état de reconfiguration (*Casting*), où l'opinion d'agents « individu » est sollicitée afin de déterminer le périmètre du groupe conformément à la règle sociale;
3. l'état de liquidation (*Firing*), où les agents « individu » exclus sont notifiés.

Les transitions entre ces états sont du type :

Événement *if* conditions → actions.

Elles sont déclenchées par un événement externe, e.g. la réception d'un message. L'automate est déterministe, i.e. les conditions de déclenchement des transitions sortantes d'un même état sont mutuellement exclusives. Ce patron de conception de comportement d'agent permet de gérer des dialogues multi-parties concurrents. Les propositions reçues par un agent « coalition » sont traitées une-à-une, quitte à être mises de côté (*stash()*) pour être évaluées plus tard (*unstash()*), mais leur acceptation/rejet dépendent de l'opinion des autres agents « individu » concernés.

Par exemple, la figure 6.4 décrit le comportement de l'agent représentant l'activité $\langle a, g \rangle$ pour la procédure sélective sans approximation. Selon ce comportement, une proposition d'un individu i reçue dans l'état *Disposing* est acceptée si la coalition est vide. Sinon, l'agent évalue cette proposition dans l'état *Casting* en sélectionnant un sous-groupe non-vide de $g \cup \{i\}$ qui respecte la contrainte de capacité et l'objectif de la coalition, i.e. la règle sociale. Ensuite, si le proposant n'est pas membre du groupe choisi alors la proposition est rejetée. Dans le cas contraire, la proposition est acceptée dès que, si des membres de g ont été écartés, ceux-ci confirment la notification de l'agent

« solveur » (dans l'état *Firing*). Pour les autres procédures d'affectation, le comportement des agents « coalition » est similaire à l'exception des conditions et des actions pour les transitions sortantes de l'état *Disposing*. Par exemple, pour la procédure inclusive, la reconfiguration de la coalition n'a lieu que lorsque la capacité est atteinte.

Finalement, on peut remarquer qu'une résolution peut atteindre des résultats différents à partir de la même configuration. Le non-déterminisme observable est nécessaire pour distribuer la résolution de problèmes au travers de programmes indépendants.

6.5 Évaluation empirique

Nos expérimentations visent à évaluer : i) l'exploitation des propriétés structurelles du problème par nos procédures ; ii) la qualité des solutions atteintes ; et iii) l'accélération due à leur distribution.

Notre prototype ScaIA (MORGE, 2018b) a été implémenté avec le langage de programmation Scala et la boîte à outils Akka¹. Cette dernière, en s'appuyant sur le modèle d'acteur (HEWITT et al., 1973), nous permet de réduire la distance entre les spécifications de nos comportements d'agent et leur implémentation.

Nos procédures exploitent la structure du problème IA qui, contrairement au jeu hédonique, est bilatérale. Afin d'en mesurer les mérites, d'une part, nous traduisons une instance de problème IA en un jeu hédonique (DREZE & GREENBERG, 1980) et nous en calculons une coalition contractuellement stable (CIS). D'autre part, nous traduisons une instance de problème IA sous la forme d'un problème quadratique mixte (en anglais, *Mixed Integer Quadratic Problem*) pour sa résolution à l'aide de l'outil IBM® ILOG® CPLEX®.

Tout d'abord, nousinstancions la définition de la fonction d'affinité et celle de la fonction d'utilité. Inspirés par notre application pratique, nous axiomatisons le principe *quo plus est eo hilarior, sed quo paucior eo melior cibus*² Cette affinité, qui est normalisée croit avec la taille du groupe à condition que i soit attiré par tous les membres du groupe. Cette fonction, qui évalue les préférences d'un individu vis-à-vis des 2^{m-1} groupes, est additivement séparable et donc sa représentation est linéaire par rapport au nombre d'individus.

Contrairement à l'optimisation mathématique, le temps d'exécution de notre procédure bénéficie des propriétés structurelles du problème IA. De plus, l'algorithme CIS est pénalisé par la génération des préférences (RLC) dont la représentation n'est pas linéaire comme dans un problème IA mais exponentielle. Même si notre procédure ne maximise pas toujours le bien-être utilitaire, nous constatons qu'elle est proche du maximum utilitariste calculé par optimisation mathématique (MIQP) tout en étant plus rapide. Le bien-être égalitaire obtenu par notre procédure est proche mais surpasse celui approximé par optimisation mathématique (MIQP) tout en étant plus rapide.

Même si nous n'avons pas de résultats théoriques pour la stabilité du résultat de la procédure sélective, nos expérimentations montrent que 95% des résultats de la procédure sélective avec approximation sont optimaux au sens de Pareto et 96% d'entre eux sont individuellement rationnels.

Comme dans notre application pratique, nous considérons que les n activités et les m individus sont attractifs et que ces derniers peuvent tous être affectés. Attendu que l'algorithme MIQP ne passe pas l'échelle, nous avons implémenté un algorithme de recherche locale (RUSSELL & NORVIG, 2003) pour le comparer à notre procédure d'affectation. Cet algorithme de montée en gradient, qui commence à partir d'un appariement valide et aléatoire (où tous les individus sont affectés à une activité), cherche à améliorer itérativement le bien-être égalitaire. Deux appariements sont voisins s'ils sont identiques à l'exception d'un individu affecté à une activité différente, éventuellement nulle. Si cette dernière est pleine, tous les échanges d'individus avec les participants de cette activité sont considérés.

1. <http://akka.io>

2. En français « plus on est de fous, plus on rit » et en anglais *the more the merrier, but the fewer the better fare*.

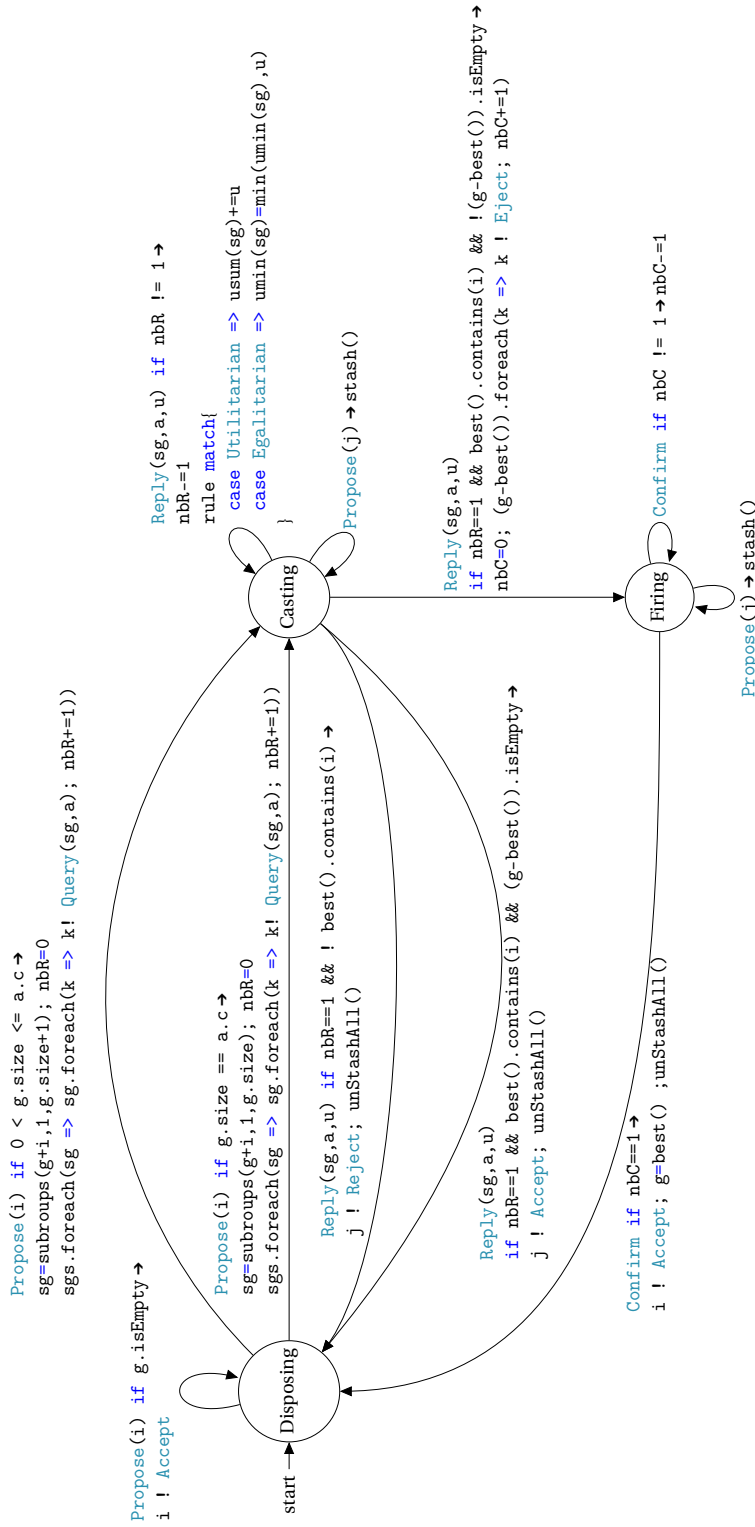


FIGURE 6.4 – Comportement de l'agent représentant la coalition $\langle a, g \rangle$ dans la procédure sélective sans approximation. subgroups (g, \min, \max) retourne les sous-groupes de g dont la taille est comprise entre \min et \max .

Tout d'abord, nous comparons le bien-être égalitaire de l'appariement atteint par notre algorithme avec celui atteint par la recherche locale. Notre procédure inclusive surpasse l'algorithme de recherche locale. En effet, le bien-être égalitaire, comme le bien-être utilitaire, est pour le problème IA une fonction qui possède de nombreux optima locaux.

Ensuite, nous comparons le temps d'exécution de l'algorithme inclusif de notre algorithme avec sa version décentralisée. Bien que l'algorithme centralisé soit plus rapide quand le nombre d'individus est faible, son temps d'exécution augmente rapidement avec le nombre d'individus alors que le temps d'exécution de la version décentralisée est inférieure. De plus, le temps d'exécution de la montée en gradient est trop important pour être représenté. On peut s'attendre à un temps d'exécution encore plus important si on adopte une méthode de recherche locale de type recuit simulé sans pour autant garantir d'obtenir un optimum global.

En résumé, la solution atteinte par notre procédure sélective (respectivement inclusive) semble proche du maximum utilitariste (respectivement égalitariste). De plus, ces algorithmes sont distribuables, ce qui permet d'accélérer leur exécution.

6.6 Application pratique

Notre application pratique concerne un collectif de seniors qui souhaitent entrer en relation afin d'effectuer des activités. L'objectif est ici de maximiser les activités partagées afin d'améliorer la cohésion sociale et lutter contre l'isolement chez certains seniors.

Selon la figure 6.5, qui représente le modèle conceptuel de données que nous avons adopté pour construire la base de données, chaque utilisateur (`HolidayMaker`) est décrit (`describesHolidayMakers`) par un ensemble de caractéristiques (`HolidayMakerFeature`) : son sexe, son âge, s'il est fumeur, ses hobbies, etc. Ainsi chaque utilisateur indique (ou non) les valeurs qui le caractérisent (`HolidayakerFeatureValue`) et il peut signaler son attraction voire sa répulsion (`attractionPartner`) vis-à-vis de ces valeurs pour ses co-vacanciers. C'est à partir de ces connaissances que sont agrégées les affinités entre utilisateurs (`computedPreferencePartner`). De manière similaire, les voyages (`Tour`) sont décrits (`describesTour`) par des valeurs (`TourFeatureValue`) de caractéristique (`TourFeature`) sur lesquels les utilisateurs expriment des préférences (`likesTourFeature`). C'est à partir de ces connaissances que sont agrégés les intérêts des utilisateurs pour les voyages (`computedPreferenceTour`).

Bien que ce terrain d'expérimentation implique plusieurs milliers d'individus, nous avons préalablement testé notre prototype avec 102 utilisateurs et 10 voyages. La table 6.1 résume les temps d'exécution et les bien-être atteints par tous les algorithmes, à l'exception des algorithmes par optimisation mathématique trop lents. Que cela soit pour le bien-être utilitaire ou le bien-être égalitaire, nos algorithmes s'avèrent plus efficaces et plus rapides que la montée en gradient. Les temps d'exécution des versions distribuées de nos algorithmes sont plus rapides quand le nombre d'intégration est suffisant.

6.7 Conclusion

Nous avons abordé ici le problème générique où des individus doivent être affectés à des activités en privilégiant les activités qui leur plaisent avec les partenaires qu'ils apprécient. Bien que souhaitable, la stabilité d'un appariement n'est pas garantie. À l'inverse, l'optimalité au sens de Pareto semble ne pas être suffisamment discriminante. C'est la raison pour laquelle nous avons évalué la qualité d'un appariement du point de vue de la société d'individus à l'aide des concepts issus de la théorie du choix social. D'autres concepts sont envisageables, qu'ils soient issus de l'Économie comme l'absence d'envie justifiée (en anglais, *envy-freeness*), la non-manipulabilité (en anglais,

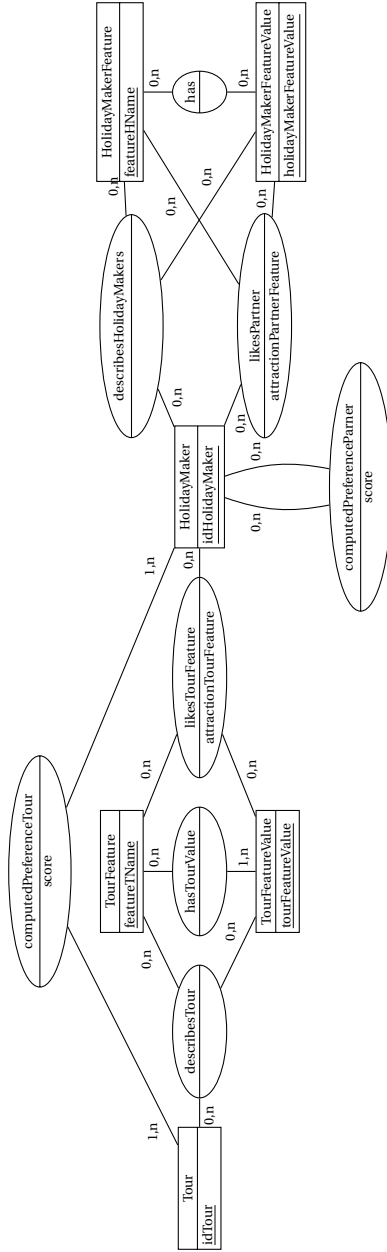


FIGURE 6.5 – Modèle conceptuel pour les données réelles collectées et traitées par nos algorithmes d'appariement. Les entités sont représentées par des rectangles et les associations par des ovales à bords arrondis.

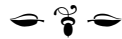
Algorithme	Bien-être	Temps d'exécution (ms)
Sélec. avec appr.	$U(M) = 0.154$	263
Dis. sélec. avec appr.	$U(M) = 0.156$	210
Montée en gradient	$U(M) = -0.066$	11 221
Inclusif	$E(M) = 0.0$	22
Dis. inclusif	$E(M) = 0.0$	44
Montée en gradient	$E(M) = -0.433$	20 643

TABLE 6.1 – Bien-être et temps d'exécution des différents algorithmes d'appariement sur les données réelles.

strategy-proofness) ou issue de la psychologie sociale comme le concept de cohésion sociale esquissé ici.

Même si une étude de la complexité dans le pire des cas est nécessaire, nos expérimentations, notamment celles concernant des données réelles, montrent que les bien-être sociaux sont pour ce problème des fonctions qui possèdent de nombreux optima locaux. C'est pourquoi nous avons proposé deux heuristiques pour chacune des notions de bien-être social. Elles retournent des appariements valides, selon une règle sociale (utilitaire ou égalitaire) pour la première et socialement cohésifs pour la seconde. Ces procédures s'appuient sur l'aspect organisationnel de la modélisation multi-agents en introduisant explicitement un niveau intermédiaire entre l'individu et la société, i.e. les coalitions d'individus qui partagent un intérêt commun pour une activité. Conformément au principe de subsidiarité, la reconfiguration d'une coalition est soumise à l'arbitrage d'un agent représentant les objectifs et contraintes d'un groupe d'individus.

Comme l'illustrent nos résultats empiriques, les méthodes de résolution multi-agents peuvent être plus efficaces que les méthodes générales pour l'optimisation de problème dont la formulation est inadaptée. De plus, au-delà d'une seule distribution qui comme ici accélère son exécution, la modélisation des comportements d'agents permet d'introduire des stratégies variées pour mettre en œuvre les contraintes et les objectifs des différentes coalitions d'agents.



CHAPITRE 7

Enchères multi-parties concurrentes

Sommaire

7.1	Introduction	81
7.2	MapReduce	82
7.3	Proposition	83
7.3.1	Processus de négociation	84
7.3.2	Multi-enchères	84
7.4	Expérimentations	85
7.5	Conclusion	86

Les travaux présentés dans ce chapitre ont été réalisés entre 2015 et 2018 dans le cadre du projet MAS4Data (*Multiagent systems for analyzing very large data sets*) soutenu par le défi CNRS MASTODONS en collaboration avec Anne-Cécile CARON et Jean-Christophe ROUTIER (Université de Lille), avec lesquels je co-encadre la thèse de Quentin BAERT. Ils ont notamment fait l'objet de la publication suivante :

- **BAERT, Q.**, CARON, A.-C., MORGE, M. & ROUTIER, J.-C. (2018). Fair Multi-Agent Task Allocation for Large Data Sets Analysis. *Knowledge and Information Systems – An International Journal*, 54(3), 591-615. h-index=52, Q2 (Scimago)

7.1 Introduction

La science des données vise à traiter de grands volumes de données pour y extraire de nouvelles connaissances (en anglais, *insight*). Comme le potentiel technologique et la demande sociale ont augmenté, de nouvelles méthodes, de nouveaux modèles, systèmes et algorithmes sont développés. L'analyse de ces données, en raison de leur volume et de leur vitesse d'acquisition, demande de nouvelles formes de traitements. À cette intention, le patron de conception MapReduce (DEAN & GHEMAWAT, 2004) est parallélisable et utilisable, par exemple pour mettre en œuvre l'algorithme PageRank, le calcul d'un index inversé, identifier les articles les plus populaires sur Wikipedia ou réaliser le partitionnement en k-moyennes. Le framework le plus populaire pour MapReduce est Hadoop (WHITE, 2015) mais de nombreuses autres implémentations existent comme le framework Spark (ZAHARIA et al., 2012) ou la base de données NoSQL distribuée Riak construite par Amazon Dynamo (DECANDIA et al., 2007). Dans ces approches, l'extraction des données ainsi que leur traitement sont distribués et exécutés sans échantillonnage.

Les données et les flux d'entrées peuvent faire l'objet de biais, de pics d'activités périodiques (quotidiens, hebdomadaires ou mensuels) ou de pics d'activités déclenchés par un événement particulier. Ces distorsions peuvent être particulièrement difficiles à gérer. Dans les frameworks existants, une affectation efficace des tâches (c.à.d. la répartition des clés) demande une connaissance *a priori* de la distribution des données. À l'inverse, nous défendons la thèse selon laquelle les systèmes multi-agents (SMA) sont particulièrement appropriés pour s'adapter à des données inconnues, à des flux qui évoluent constamment ou à un environnement informatique dynamique. Les systèmes multi-agents constituent un paradigme de premier ordre pour l'analyse, la conception et l'implémentation de systèmes composés d'entités interagissantes; le traitement de données distribuées avec le patron d'architecture logicielle MapReduce en est un exemple. Dans les SMA, l'autonomie des agents permet au système de s'adapter dynamiquement aux variations et perturbations de l'environnement.

Dans ce chapitre, nous proposons un système multi-agent adaptatif pour mettre en œuvre le patron de conception MapReduce afin d'analyser des données. Le traitement des données est distribué parmi deux types d'agents : (i) les agents mappers qui filtrent les données; (ii) les agents reducers qui agrègent les données. Afin d'équilibrer la charge de travail des reducers, les tâches sont dynamiquement re-allouées parmi les reducers durant le processus et sans échantillonnage. Pour y parvenir les reducers sont impliqués dans de multiples enchères concurrentes. Nos agents négocient les tâches sur la base de leur charge individuelle afin de faire diminuer celle de l'agent le plus chargé, c.-à-d. celui qui retarde le traitement des données. Nous prouvons que ce processus de négociation améliore l'équité qui mesure si le traitement est effectué au détriment du plus chargé des agents. Nous avons confronté notre système multi-agent à des données réelles et nos observations confirment la plus-value des négociations.

Ce chapitre est structuré comme suit. La section 7.2 introduit le patron de programmation MapReduce qui est le contexte d'application de nos travaux. La section 7.3 décrit le cœur de notre proposition. Puis nous présentons nos résultats expérimentaux dans la section 7.4. Finalement, la section 7.5 conclut.

7.2 MapReduce

Le patron de programmation MapReduce permet de traiter de grands volumes de données (DEAN & GHEMAWAT, 2004). Dans ce modèle, la fonction *map* filtre les données et la fonction *reduce* les agrège.

L'implémentation distribuée Hadoop permet aux développeurs sans expérience avec les systèmes parallèles et distribués d'utiliser facilement les ressources au sein d'un tel système. Les jobs MapReduce sont divisés en deux ensembles de tâches, les tâches *map* et les tâches *reduce*, qui sont distribuées sur une grappe de PCs. Le modèle de programmation MapReduce s'appuie sur ces deux fonctions fournies par l'utilisateur avec les signatures suivantes :

$$\begin{aligned} \text{map} &: (K1, V1) \rightarrow \text{list}[(K2, V2)] \\ \text{reduce} &: (K2, \text{list}[V2]) \rightarrow \text{list}[(K3, V3)] \end{aligned}$$

La figure 7.1 illustre le flux de données MapReduce :

1. le superviseur partage les données d'entrée entre les mappers;
2. les mappers appliquent la fonction *map* sur les données d'entrée et créent les couples intermédiaires clé-valeur (*key* : $K2$, *value* : $V2$);
3. une fonction de partitionnement est appliquée sur la sortie des mappers afin de les diviser en sous-ensembles, un sous-ensemble par reducer de sorte que tous les couples avec la même clé soient envoyés au même reducer. La fonction de partitionnement peut être personnalisée afin de spécifier les clés qui doivent être traitées ensemble par le même reducer;

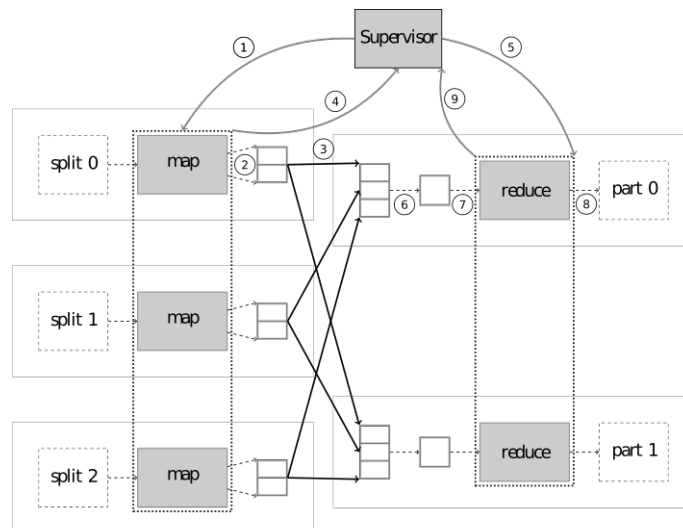


FIGURE 7.1 – Flux de données MapReduce

4. une fois qu'un mapper a traité son entrée, il en informe le superviseur;
5. lorsque tous les mappers ont traité leur entrée, le superviseur informe les reducers pour lancer la seconde phase;
6. les reducers agrègent les couples intermédiaires clé-valeur pour construire les couples $(K2, list[V2])$;
7. les reducers prennent les couples $(K2, list[V2])$ et exécutent la fonction *reduce* sur chaque groupe de valeurs associé à chaque clé;
8. les couples finaux clé-valeur $(K3, V3)$ sont écrits dans un fichier du système de fichiers distribué;
9. enfin, les reducers informent le superviseur de la localisation du résultat final. Lorsque tous les reducers ont terminé leurs tâches, le travail est terminé.

Qu'il soit réalisé par une fonction par défaut ou par une fonction spécifique, le partitionnement est fixé *a priori*. Par exemple, pour déterminer quel reducer traite une clé, la partition Hadoop effectue un modulo sur le nombre de reducers :

```
key.hashCode() % numberOfReducers.
```

Cette fonction est fixée sans connaissance *a priori* sur les données. Ainsi la charge de travail n'est pas nécessairement répartie uniformément entre les reducers.

Outre ce problème de partition de données, les nœuds qui exécutent un job MapReduce (et donc les reducers) n'ont pas nécessairement des performances homogènes. Leurs performances peuvent varier durant l'exécution. En conséquence, le temps de calcul est déterminé par le nœud le plus chargé voire le plus lent. Notre but ici est de résoudre ce problème lié aux biais des données (ou à l'hétérogénéité des performances) en réaffectant dynamiquement les tâches des reducers pendant le traitement.

7.3 Proposition

Dans notre système multi-agents, les reducers négocient les tâches en utilisant *Contract Net Protocol* (SMITH, 1980). Ainsi, ils équilibrent leur charge de travail qu'ils exécutent des tâches au cours

de la phase de reduce. Il est important de noter que le traitement des tâches et les négociations se déroulent simultanément. L'adaptation de la charge de travail est alors continue. Ici, nous nous focalisons sur la mise en œuvre d'enchères simultanées.

7.3.1 Processus de négociation

Rappelons les principes du processus de négociation par le biais d'un exemple de processus de négociation. Nous considérons ici une enchère particulière pour un seul job MapReduce.

Une tâche reduce représente une clé et toutes ses valeurs associées. Le coût d'une tâche est défini par le nombre de valeurs qu'elle contient. Ainsi, une tâche a le même coût pour tous les reducers du système. Nous appelons contribution d'un reducer la somme des coûts des tâches restantes à exécuter.

Dans notre exemple, nous supposons que l'étape de map est terminée et que les tâches de reduce sont initialement attribuées à quatre reducers, $\Omega = \{1, 2, 3, 4\}$. Nous nous concentrons sur l'allocation des tâches à l'instant t de telle sorte que les contributions individuelles soient $c_1(t) = 10$, $c_2(t) = 8$, $c_3(t) = 3$ et $c_4(t) = 5$, où $c_i(t)$ est la contribution de l'agent i à l'instant t (voir figure 7.2a). Chaque reducer dispose de croyances concernant les contributions de ses pairs. Ces croyances sont mis à jour grâce aux informations véhiculées par les messages de négociation.

Afin de diminuer leur contribution, tous les reducers initient des enchères. En particulier, le reducer #1 initie une enchère à propos de la tâche τ avec $c_\tau = 3$ par le biais d'un appel à propositions (cfp) envoyé aux pairs (voir figure 7.2b). Un cfp contient la contribution de l'initiateur ($c_1(t)$) et le coût de la tâche à négocier (c_τ).

Pour décider s'il peut prendre la tâche τ au temps $t + 1$, le reducer i ($i \in \Omega \setminus \{1\}$) doit satisfaire le critère d'acceptabilité suivante : $c_i(t) + c_\tau < c_1(t)$. Les reducers qui satisfont ce critère amélioreront l'équilibre de la charge puisqu'ils peuvent prendre la tâche τ . En conséquence, ils font une proposition pour τ alors que les autres refusent la délégation de tâche. Par exemple, le reducer #2. ne prend pas τ . Dans le cas contraire, sa contribution $c_2(t) + c_\tau$ serait supérieure à $c_1(t)$. Dans le même temps, le reducer #3 et le reducer #4 font des propositions pour τ en envoyant leurs contributions au reducer #1 (voir figure 7.2c).

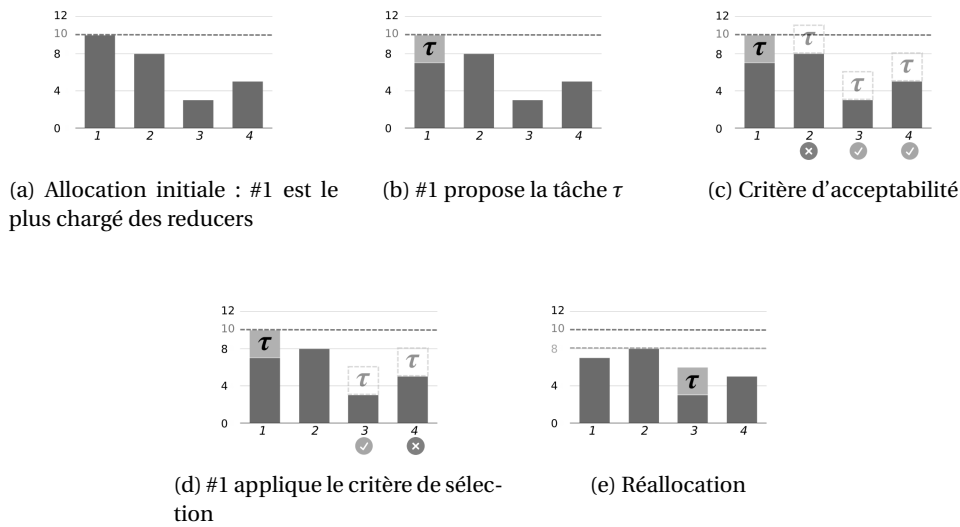
Le reducer #1 reçoit les propositions des agents $\Omega' = \{3, 4\}$. Il met à jour ses croyances au sujet des contributions de ses pairs pour de futures décisions. Ensuite, il choisit de déléguer τ à l'enchérisseur le moins chargé en appliquant le critère de sélection suivant : $\operatorname{argmin}_{j \in \Omega'} (c_j(t))$. De cette manière, le reducer #1 accepte la proposition du reducer #3 et il rejette celui du reducer #4 (voir figure 7.2d).

Après la négociation (à l'instant $t + 1$), nous observons que la tâche τ appartient au reducer #3. Les nouvelles contributions sont $c_1(t + 1) = 7$, $c_2(t + 1) = 8$, $c_3(t + 1) = 6$ et $c_4(t + 1) = 5$. La négociation conduit à une meilleure configuration, c'est-à-dire une répartition plus équitable des tâches (voir figure 7.2e).

L'attribution des tâches est dynamique et adaptative car les négociations se répètent. Par exemple, si un réducteur est ralenti pour certaines raisons, alors une allocation déséquilibrée apparaîtra, et donc une négociation sera déclenchée afin de déléguer une autre tâche et diminuer la contribution courante du reducer retardé. Plusieurs reducers peuvent simultanément initier des négociations, mais un reducer est soit un initiateur soit un enchérisseur à un instant t . Dans la section suivante, nous allons autoriser un enchérisseur à participer à plus d'une enchère à la fois.

7.3.2 Multi-enchères

Nous considérons ici un processus d'enchères concurrentes qui permet à un agent d'enchérir simultanément dans plusieurs enchères. De cette manière, l'écart entre le reducer le plus chargé et le moins chargé est diminué plus rapidement et donc, la réactivité du système multi-agents est améliorée.

FIGURE 7.2 – Processus de négociation étape par étape : le reducer #1 délègue la tâche τ .

Afin de gérer simultanément plusieurs enchères, un enchérisseur maintient un surcoût lié à sa contribution. Ce coût additionnel est la somme de tous les coûts des tâches pour lesquelles l'enchérisseur a fait une proposition. En d'autres termes, un enchérisseur enregistre les surcoûts correspondants aux enchères qu'il est susceptible de remporter. De cette manière, un enchérisseur peut faire des propositions pertinentes dans plusieurs enchères et faire une proposition seulement si le critère d'acceptabilité est susceptible d'être satisfait (en considérant sa contribution courante et le surcoût).

Soit c la contribution du reducer et o son surcoût. Initialement, $o = 0$. Lorsque le réducteur reçoit un cfp à propos d'une tâche τ avec le coût $cost$ c_τ de l'initiateur i avec la contribution c_i , l'enchérisseur adopte le comportement suivant :

- soit $c + c_\tau \geq c_i$, l'enchérisseur ne satisfait pas le critère d'acceptabilité et il décline le cfp ;
- soit $c + o + c_\tau < c_i$, l'enchérisseur fait la proposition avec sa contribution potentielle (i.e. $c + o$) comme le critère d'acceptabilité est satisfait quel que soit le résultat des enchères en cours. Donc, $o \leftarrow o + c_\tau$;
- soit $c + o + c_\tau \geq c_i$ et $c + c_\tau < c_i$, sa réponse dépend de l'issue des enchères en cours, il stocke donc le cfp afin de réévaluer ultérieurement le critère d'acceptabilité correspondant.

Quand une enchère se termine, l'enchérisseur :

1. met à jour le surcoût, i.e. $o \leftarrow o - c_\tau$, et éventuellement sa contribution $c \leftarrow c + c_\tau$ s'il remporte l'enchère ;
2. réévalue les cfps stockés avec sa contribution courante et le surcoût mis à jour. Selon le comportement précédent, l'enchérisseur peut répondre (ou pas) aux cfps stockés par une acceptation ou un refus.

Enfin, l'enchérisseur ne conserve que le dernier cfp de chaque pair afin de ne pas répondre à une enchère close.

7.4 Expérimentations

Nos expériences comparent notre proposition à la distribution classique de MapReduce. De plus, ils évaluent la valeur ajoutée du processus d'enchères multiples.

Nous avons implémenté notre prototype (BAERT, CARON, MORGE & ROUTIER, 2019) avec le langage de programmation Scala et la boîte à outil Akka¹. Cette dernière, basée sur le modèle d'acteur (HEWITT et al., 1973), permet de combler l'écart entre les spécifications du comportement des agents et leur implémentation. Afin d'évaluer notre proposition, nous avons réimplémenté la classique mise en œuvre distribuée du modèle de programmation MapReduce (DEAN & GHEMAWAT, 2004). Nous la comparons avec notre SMA. Rappelons que l'allocation initiale des tâches pour notre SMA correspond au partitionnement par défaut d'Hadoop. De plus, nous considérons l'allocation quasi-optimale des tâches obtenues par la règle LPT (*Longest Processing Time*) selon laquelle à $t = 0$, les n tâches les plus coûteuses sont affectées aux n reducers puis, chaque fois qu'un reducer est libre, il lui est assigné la tâche la plus coûteuse parmi celles restantes (GRAHAM, 1969). Il est important de noter que, sans connaissance préalable sur les données, cette dernière est une limite théorique qui ne peut être calculée que rétroactivement.

La comparaison est faite sur l'équité (le rapport entre la contribution du reducer le plus chargé et celle du moins chargé) et le taux d'accélération (le rapport entre le temps d'exécution du MapReduce classique et celui de notre SMA). Nous avons observé que la négociation équilibre la charge de travail entre les reducers car les tâches sont dynamiquement ré-allouées au cours du processus. Les reducers surchargés délèguent certaines tâches aux reducers qui sont moins chargés ou inoccupés.

Dans notre proposition, la phase de négociation et la phase reduce sont simultanées. Par conséquent, les tâches qui sont données au worker ne peuvent plus être négociées. C'est la raison pour laquelle il existe un écart entre l'équité de notre allocation dynamique de tâches et l'allocation quasi-optimale. Même si l'allocation des tâches initiale est presque équilibrée, la phase reduce réalisée par notre SMA est plus rapide que celle réalisée par le partitionnement classique des tâches. Ceci est dû au fait que notre SMA est toujours en mesure de produire une allocation de tâches plus équitable et que le surcoût lié à la négociation est faible.

De plus, nous avons comparé le processus d'enchères multiples simultanées qui permet à l'agent d'enchérir dans plusieurs enchères simultanées à un seul processus d'enchères où chaque reducer peut être impliqué en tant qu'enchérisseur dans au plus une enchère. Comme le processus d'enchères multiples améliore la réactivité, la phase de reduce avec un processus multi-enchère est plus rapide.

En résumé, nos expériences montrent que notre SMA bénéficie davantage du parallélisme que d'autres mises en œuvre. En particulier, la négociation des tâches diminue la durée d'exécution de la phase de reduce en améliorant l'équilibre de charge de travail. De plus, le processus d'enchères multiples améliore la réactivité du SMA.

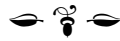
7.5 Conclusion

Les applications MapReduce sont complexes à optimiser car elles s'appuient sur des opérations définies par l'utilisateur qui nécessitent que le développeur d'application comprenne l'implémentation du framework qu'il utilise (Hadoop par exemple) mais aussi qu'il ait une connaissance *a priori* des propriétés des données ainsi que de l'environnement physique. En particulier, des biais dans les données peuvent entraîner une répartition inégale des charges due à un partitionnement fixé et statique. Notre SMA constitue un modèle de calcul qui est intrinsèquement adaptatif.

Dans ce chapitre, nous avons mis en œuvre une distribution du modèle de programmation MapReduce où l'allocation des tâches est le résultat de négociations multi-agents au cours de la phase reduce. Cette réallocation dynamique des tâches s'appuie sur des décisions locales des reducers incarnés par des agents, sans orchestration ni prétraitement des données et sans paramètre dépendant des données ou de l'environnement informatique. Afin d'équilibrer la charge de travail, ces agents négocient des tâches en fonction de leurs contributions pour diminuer celle des reducers les moins

1. <http://akka.io>

performants, c'est-à-dire ceux qui retardent la phase reduce. Nous avons prouvé que notre processus de négociation améliore l'équité de l'allocation des tâches. Nos expériences ont confirmé que notre proposition diminue le temps d'exécution de la phase reduce. De plus, afin d'améliorer la réactivité, nous avons proposé un processus d'enchères multiples qui permet à un agent d'enchérir dans plusieurs enchères simultanées. Il améliore la réactivité du SMA.



CHAPITRE 8

Projet de recherche

Sommaire

8.1 Synthèse	89
8.1.1 Verrous scientifiques	89
8.1.2 Verrous technologiques	90
8.2 Perspectives	91
8.2.1 Contexte scientifique	91
8.2.2 Périmètre scientifique	92
8.2.3 Verrous scientifiques	94
8.2.4 Verrous technologiques	96
8.2.5 Résumé analytique	96

8.1 Synthèse

L'ensemble des travaux (passés et en cours) qui sont présentés dans ce manuscrit, s'intègrent aux recherches menées par l'équipe SMAC du laboratoire CRISTAL dans le domaine de l'Intelligence Artificielle Distribuée. Mon projet de recherche vise à proposer des processus décentralisés pour la gestion des conflits entre des agents autonomes en interaction, en spécifiant leurs comportements. Je défends la thèse selon laquelle le mécanisme opératoire nécessaire pour résoudre un conflit pratique est la concession.

8.1.1 Verrous scientifiques

Mes travaux sont résumés dans la figure 8.1 reproduite ici. Ma grille d'analyse distingue la complexité du modèle de préférence (sur l'axe des abscisses) et l'arité du processus (sur l'axe des ordonnées). L'espace hachuré correspond au domaine de recherche que j'ai exploré. La flèche épaisse pointe la direction de recherche prise. Je travaille actuellement sur deux projets. D'une part, la mise en œuvre de multiples négociations bilatérales concurrentes et multi-niveaux permet d'adresser le problème général d'affectation comme dans le projet PartENS (cf chapitre 6), d'autre part, la mise en œuvre de multiples enchères concurrentes permet d'étendre les cadres de négociation envisagés afin d'adresser le problème général de ré-allocation dynamique de tâches pour l'appliquer au domaine des sciences des données à travers le projet MAS4Data (cf chapitre 7).

Le verrou scientifique auquel ce projet se confronte consiste à garantir la qualité de l'accord obtenu par le processus de négociation. Que l'accord obtenu soit optimal au sens de Pareto ou qu'il

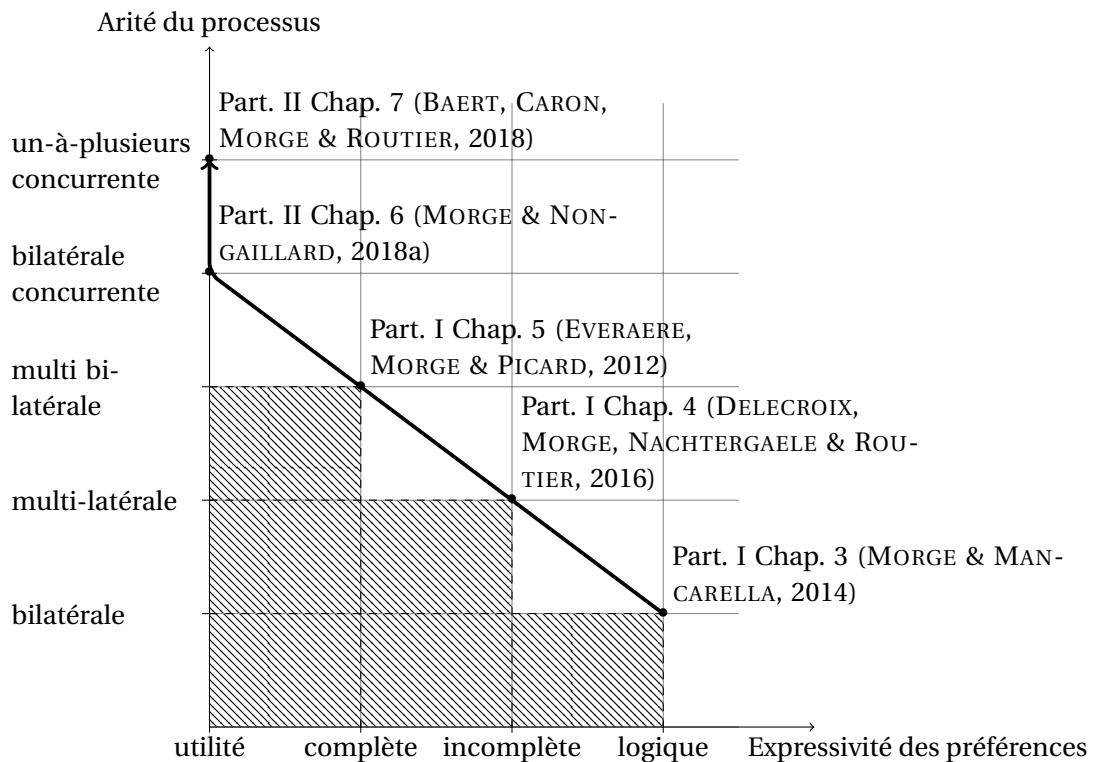


FIGURE 8.1 – Grille d’analyse des contributions passées et en cours

maximise le bien-être utilitaire, la difficulté réside dans le fait que l’information dont disposent les agents est imparfaite. De plus, les concessions, c’est-à-dire les efforts réalisés par les agents pour atteindre un accord doivent être équitablement répartis. Cela est d’autant plus difficile quand les échanges de propositions sont asynchrones (cf les chapitres 5, 6 et 7).

8.1.2 Verrous technologiques

S’inscrivant dans une approche empirique, l’ensemble des travaux présentés dans ce manuscrit ont fait l’objet d’implémentations.

MARGO (MORGE & MANCARELLA, 2006) est un moteur d’argumentation écrit en Prolog qui met en œuvre le cadre d’argumentation présenté dans le chapitre 3. MARGO est intégré dans la plate-forme multi-agents GOLEM (BROMURI & STATHIS, 2007).

NegoMaze (DELECROIX et al., 2014) est un simulateur écrit en Java qui contient un modèle d’environnement, un modèle d’agent et une interface graphique qui repose sur l’API Swing pour afficher les labyrinthes et visualiser le processus de négociation présenté dans le chapitre 4. Cette mise en œuvre a été couronnée par le troisième prix de la meilleure démonstration décerné par IBM à la conférence PAAMS en 2014 (DELECROIX et al., 2014).

DSMP (PIETTE et al., 2014) et **ScaMa** (MORGE, 2014) sont des bibliothèques écrites respectivement en Java et en Scala qui implémentent les algorithmes d’appariements évalués dans le chapitre 5.

ScaIA (MORGE, 2018b) est une bibliothèque écrite en Scala qui implémentent les algorithmes de formation de coalitions évalués dans le chapitre 5. Le modèle d’acteur (HEWITT et al., 1973) que nous adoptons nous permet de réduire la distance entre les spécifications de nos comportements d’agent et leur implémentation.

MAS4Data (BAERT et al., 2019) est un système distribué multi-agents qui permet de déployer le patron de conception MapReduce pour le traitement de jeux de données de grandes tailles sur une

grappe de serveurs (cf chapitre 7). Ce prototype a été implémenté en Scala avec la boîte à outils Akka¹ adaptée aux applications orientées messages, fortement concurrentes, distribuées et robustes.

Les principales difficultés rencontrées dans la conception de ces prototypes viennent du fait que :

- les agents ne partagent pas la même horloge. À partir de la même configuration, plusieurs exécutions peuvent atteindre des solutions différentes;
- les agents ne partagent pas les mêmes variables. Ils ne se synchronisent pas directement via un environnement commun spatialisé mais indirectement avec leurs propres accointances par des canaux de communication;
- les agents ne partagent pas un état global. Ils n'ont qu'une connaissance partielle de l'état du système. Les propriétés de stabilité des algorithmes multi-agents, comme leur terminaison, doivent être détectées.

8.2 Perspectives

Le problème d'équilibrage des charges et d'allocation des tâches dans un système distribué apparaît dans de nombreuses applications telles que l'informatique en nuage, les réseaux pair-à-pair, les réseaux sociaux et le traitement de données massives. Dans ce projet de recherche, j'aborde les problèmes applicatifs où (a) certaines ressources, par exemple des données, nécessaires à l'exécution d'une tâche sont distribuées sur différents nœuds du réseau, et (b) certains des nœuds sont susceptibles de rencontrer des perturbations lors de l'exécution, par exemple une chute de performance ou des latences réseau. Dans cette classe de problème, l'allocation de tâches peut être remise en cause lors de l'exécution des tâches.

8.2.1 Contexte scientifique

Afin d'aborder le problème d'équilibrage des charges et d'allocation des tâches dans des applications telles que celles qui motivent ce travail, les technologies multi-agents ont fait l'objet d'une grande attention, particulièrement celles qui portent sur l'ordonnancement à l'aide d'enchères (SHEHORY & KRAUS, 1998; TURNER, MENG, SCHAEFER & SOLTOGGIO, 2018; WALSH & WELLMAN, 1998). En plus de la décentralisation qui permet d'éviter les goulots d'étranglement en terme de performance, l'approche multi-agents pour l'allocation de tâches répond à deux autres exigences cruciales : (a) la co-occurrence de la réallocation des tâches et de leur exécution; (b) l'adaptation de l'allocation, c'est-à-dire le déclenchement d'une réallocation quand une perturbation se produit.

Les problèmes d'ordonnancement classiques ont fait l'objet d'un grand nombre d'études et de recherches. Ces dernières ont abouti à des ordonnanceurs hors-ligne pour des modèles simples (CHEN, POTTS & WOEGINGER, 1998). Le problème qui consiste à minimiser le *flowtime* (i.e. le temps d'attente moyen) de n tâches sur m machines avec des capacités différentes, dénotées $R||\Sigma C_j$, est polynomial (HORN, 1973). Le problème de la minimisation du *makespan* (le temps auquel la dernière tâche est achevée) pour n tâches sur m machines hétérogènes (avec des capacités différentes), appelé $R||C_{max}$, est NP-difficile (HOROWITZ & SAHNI, 1976). Les algorithmes pseudo-polynomiaux conçus pour ce problème incluent : l'heuristique *earliest completion time* (ECT) proposé par IBARRA et KIM, 1977, les heuristiques de recherche locale (HARIRI & POTTS, 1991), l'algorithme *branch and bound* (MARTELLO, SOUMIS & TOTH, 1997) et les heuristiques biphasées basées sur la programmation linéaire (LENS-TRA, SHMOYS & TARDOS, 1990). Même si ECT est un algorithme d'approximation qui atteint des résultats acceptables avec un coût computationnel faible, les algorithmes centralisés ne peuvent pas être appliqués à large échelle où les tâches sont nombreuses.

1. <http://akka.io>

L'ordonnancement multi-agents (AGNETIS, BILLAUT, GAWIEJNOWICZ, PACCIARELLI & SOUKHAL, 2014; Y. JIANG, 2016) a suscité beaucoup d'intérêt dans le cadre du problème d'équilibrage de charge pour les systèmes distribués. Ce problème est différent des problèmes d'ordonnancement classiques de par ses exigences :

- **passage à l'échelle.** Un contrôleur global constitue un goulot d'étranglement en terme de performance car il doit collecter les informations d'état de l'ensemble du système en temps réel (COULOURIS, DOLLIMORE & KINDBERG, 2012). Au lieu de cela, la répartition des tâches peut être négociée par des agents représentant les nœuds (AN, LESSER, IRWIN & ZINK, 2010; WALSH & WELLMAN, 1998);
- **réactivité.** Les problèmes d'ordonnancement classiques sont statiques. L'estimation inexacte du temps d'exécution des tâches, aggravée par des perturbations (consommation de tâches, ralentissement des nœuds, etc.), peuvent nécessiter d'importantes modifications de l'allocation existante pour qu'elle reste optimale (TURNER et al., 2018). Plutôt que de recalculer en continu une allocation optimale, quelques modifications locales au cours de l'exécution des tâches peuvent améliorer l'équilibre de charge (ATTIYA & HAMAM, 2006; KANG, HE, SONG & DENG, 2010; YIN, YU, WANG & WANG, 2007).

La plupart des travaux existants adoptent une approche fondée sur le marché (GARG, VENUGOPAL, BROBERG & BUYYA, 2013; NAJJAR, BOISSIER & PICARD, 2017; WALSH & WELLMAN, 1998) qui modélise l'équilibrage de charge comme un jeu non coopératif afin d'optimiser des métriques centrées sur l'utilisateur plutôt que des métriques centrées sur le système, telles que le temps d'achèvement global que nous utilisons dans cet article.

8.2.2 Périmètre scientifique

Mon projet de recherche consiste à formaliser le problème de **formation de coalition multi-agents pour l'allocation de tâches**. À travers la négociation multi-agents, je souhaite décentraliser l'allocation des tâches dans les systèmes distribués. À cette intention, les agents doivent opérer dans des négociations concurrentes pour former des coalitions afin de réaliser conjointement des tâches. La réallocation de tâches doit être dynamique grâce à un processus de négociation itéré et concomitant avec l'exécution des tâches. De plus, le système doit s'adapter localement aux événements perturbateurs.

La formalisation d'un tel problème et donc sa résolution s'appuie sur un grand nombre de paramètres et d'hypothèses sous-jacentes qui doivent être explicitées.

Répartition et partage de ressources. Dans les systèmes distribués, des ressources sont disséminées et placées sur certains nœuds. Ce sont des données, des capacités de calcul ou de stockage qui sont nécessaires pour l'exécution des tâches. La répartition de ces ressources éventuellement disputées dans le système influence l'allocation de tâches (Y. JIANG & JIANG, 2008).

Localisation et connaissance des agents. Comme les ressources, les agents sont disséminés et situés explicitement sur certains nœuds. Ils sont munis d'un comportement individuel et ils maintiennent leur propre état sans pour autant avoir accès à l'état global du système : les ressources disponibles, les tâches allouées et celles à réaliser ainsi que leur coût. Toutefois, ils peuvent avoir une connaissance locale et partielle du système, voire établir des croyances (éventuellement erronées) sur ce système par échange de messages ou par déduction.

Topologie du réseau. Dans un système distribué, les nœuds sont en réseau. Selon sa structure, chaque nœud peut communiquer directement avec ses voisins et la communication entre deux nœuds qui ne sont pas voisins doit être relayée par d'autres nœuds intermédiaires. En conséquence, la négociation (NONGAILLARD & MATHIEU, 2011) et l'exécution des tâches (Y.-C. JIANG & JIANG, 2005) sont fortement influencées par la topologie du réseau.

Type de tâches. Les tâches sont associées à un coût computationnel, voire à un certain nombre de contraintes concernant leur exécution. Qu'elles soient atomiques ou composites et divisibles, les tâches peuvent être associées à une date butoir, voire à une relation de précédence qui contraint l'ordre d'exécution. Une tâche est « préemptible » si son exécution peut être suspendue au profit d'une autre tâche pour être reprise plus tard. Une tâche n'est pas partageable si elle appartient à un unique agent à un instant donné. À l'inverse, l'affectation de tâches à des groupes d'agents est nécessaire si elles ne peuvent pas être exécutées par un unique agent comme envisagé par SHEHORY et KRAUS (1998).

Capacité des agents. Les agents peuvent avoir des capacités et des efficacités différentes. Par exemple, VINYALS, MACARTHUR, FARINELLI, RAMCHURN et JENNINGS (2014) parallélisent des tâches de calcul hétérogènes quand tous les agents ne sont pas en mesure d'exécuter toutes les tâches et quand ils ont des vitesses de traitement différentes pour des tâches différentes. Si plusieurs ressources sont nécessaires pour effectuer une tâche et comme la récupération des ressources consomme du temps, cette collecte représente un coût supplémentaire pour l'exécution de la tâche. En d'autres termes, les agents, qui sont situés sur des nœuds différents, ont des vitesses différentes pour des tâches différentes.

Préférences des agents. Contrairement au problème classique d'affectation de tâches (KUHN, 1955) qui se limite à l'affectation d'une seule ressource par agent, la plupart des travaux sur l'allocation multi-agents de tâches supposent qu'un lot de tâches peut être assigné à chaque agent. De même, le coût computationnel des tâches est souvent représenté par des fonctions. En d'autres termes, j'adopterai un modèle cardinal et quantitatif des préférences plutôt qu'une relation de préférence.

Objectifs des agents. On distingue principalement deux approches pour définir l'objectif d'un processus d'allocation de tâches dans un système distribué. D'une part, l'approche centrée sur le système vise un objectif global commun à l'ensemble des agents. D'autre part, l'approche centrée sur l'utilisateur se focalise sur l'équité entre les utilisateurs du système. Alors que dans le premier cas, il est nécessaire de considérer des agents coopératifs qui collaborent pour atteindre un objectif global, la plupart des travaux adoptant la seconde approche modélisent le problème comme un jeu non-coopératif. Qu'ils soient locaux ou globaux, on distingue deux types d'objectifs à l'allocation de tâches : l'équilibrage des charges et la fiabilité de l'allocation. Afin d'évaluer l'équilibrage des charges, on peut mesurer notamment le *makespan* qui évalue le temps écoulé entre le début de la première tâche à traiter et la fin de la dernière tâche exécutée. La fiabilité de l'allocation, quant à elle, correspond à la probabilité que les tâches soient exécutées avec succès en fonction des nœuds impliqués, des ressources nécessaires, des calculs réalisés ou des communications utilisés. On peut considérer de multiples objectifs à l'allocation de tâches qui à défaut de pouvoir être tous satisfaits simultanément nécessitent des compromis, voire des concessions.

En résumé, l'allocation multi-agents de tâches est un champ de recherche large et varié.

Exemple 11 *Afin d'illustrer les paramètres et hypothèses sous-jacentes à un problème d'allocation de tâches, nous considérons l'application abstraite du monde des colis (en anglais, Packet-World) introduit par WEYNS, HELLEBOUGH et HOLVOET (2005). Ce problème constitue un banc d'essai pour étudier, raffiner, expérimenter et évaluer les algorithmes multi-agents (STATHIS, 2018).*

Problème. *L'environnement que nous considérons est une grille de cellules (cf figure 8.2). Chaque cellule contient au plus une entité, qu'elle soit passive (un objet) ou active. Quel que soit leur poids, les colis dispersés comme la destination où ils doivent être déposés sont des objets. Une entité active est soit le corps d'un agent soit une foule, c.-à-d. un ensemble de corps d'agents. La taille d'une entité active, c'est-à-dire le nombre de corps qui la constitue, détermine le poids maximal des colis qu'elle peut transporter.*

Action. *À chaque pas de simulation, les actions des entités actives, éventuellement conflictuelles, sont simultanées. Chaque entité active soumet une influence, c.-à-d. une action qu'elle souhaite réali-*

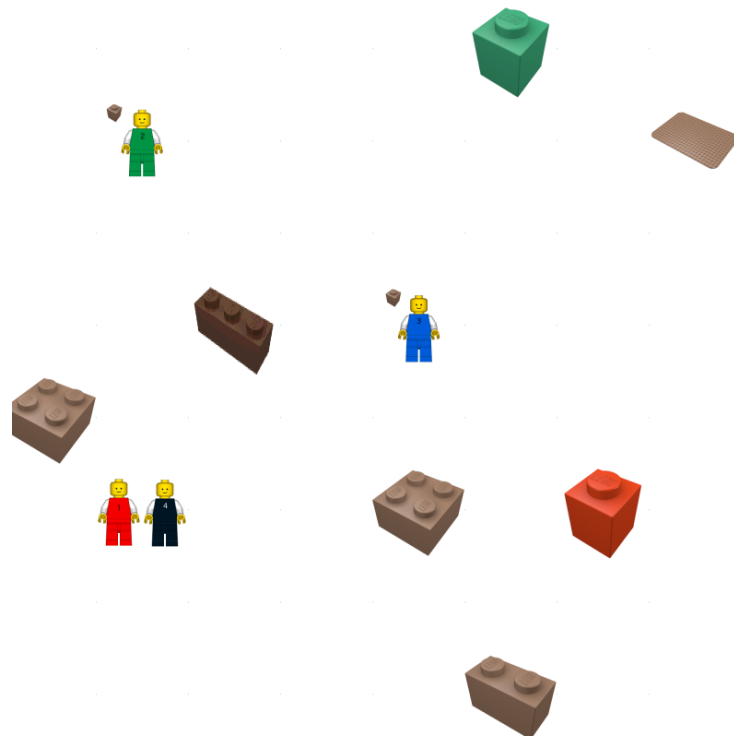


FIGURE 8.2 – Problème de la collecte multi-agents de colis (MORGE, 2019). Un colis ciblé par une entité est coloré.

ser (MICHEL, 2007). La réaction du simulateur, qui résout les conflits et (in)valide les influences, génère une nouvelle instance de problème ou clôt la simulation. Les actions réalisables par une entité active sont : (a) le déplacement vers l'une des cases adjacentes, (b) le chargement d'un colis si l'entité en a la capacité, (c) le dépôt d'un colis si l'entité chargée est adjacente à la destination, (d) l'agrégation avec une autre entité active consentante, ou (e) la dissolution.

Effort. Arbitrairement, chaque action a un coût unitaire qui représente l'effort nécessaire pour la réaliser. Le coût de la collecte d'un colis par une entité active correspond (a) à la distance la plus courte pour atteindre une case adjacente à ce colis, (b) au chargement, (c) à la distance la plus courte pour atteindre une case adjacente à la destination, et (d) au dépôt du colis. Le nombre d'actions accomplies par les corps d'agent correspond à l'effort que ces derniers ont réalisé et le nombre de pas de simulation aux efforts consentis par le corps le plus sollicité (en anglais, *makespan*).

Tâches. Pour être tous collectés, chaque colis doit être ciblé par une entité active. Il est important de noter que le coût d'une tâche (i.e. un colis à collecter) ne correspond pas nécessairement aux efforts effectivement observés. Certaines actions peuvent échouer (e.g. un déplacement vers une case occupée) ou être non sollicitées (e.g. une entité refoulée par une autre). Afin de réduire le *makespan*, les agents peuvent être amenés à échanger des tâches au cours de la collecte à cause de ces aléas d'exécution.

8.2.3 Verrous scientifiques

L'originalité de mon projet de recherche réside d'une part dans la modélisation multi-niveaux et d'autre part dans le processus de décision collective qui consiste en de multiples négociations concurrentes.

Modélisation multi-niveaux. Issue du domaine de la simulation multi-agents (PICAULT, 2013), la modélisation multi-niveaux permet d'enrichir la formalisation de problèmes pour leur résolution multi-agents (NONGAILLARD & PICAULT, 2017). Au delà de sa seule représentation, mon objectif est

de concevoir une **méthode générique de résolution au moyen d'un SMA multi-niveaux**. Cette approche est d'autant plus pertinente quand le problème abordé est intrinsèquement multi-niveaux comme dans le projet PartENS évoqué dans le chapitre 6 ou comme envisagé dans ce projet de recherche, i.e. la formation de coalition pour la réalisation de tâches qui ne peuvent pas être effectuées par un unique agent. L'idée consiste à réifier une organisation d'agents, qu'il s'agisse de sa structure (i.e. les rôles joués par les agents, les groupes formés et leurs relations) ou de sa fonction (i.e. les missions assignées aux agents), sous la forme d'un « macro-agent » qui agrège les comportements de micro-agents. En d'autres termes, on introduit explicitement un niveau intermédiaire, entre l'individu et la société, qui exhibe un comportement. Cette modélisation permet d'introduire des stratégies variées pour mettre en œuvre les contraintes et les objectifs des groupes d'agents. Conformément au principe de subsidiarité, la reconfiguration d'une organisation est soumise à l'arbitrage d'un macro-agent représentant les objectifs et contraintes du groupe.

Contrairement aux systèmes holoniques (FISHER, SCHILLO & SIEKMANN, 2004), cette approche n'est pas strictement hiérarchique. Un micro-agent peut être hébergé par plusieurs macro-agents. Au cours de la collecte de colis, un agent peut être membre de plusieurs groupes afin de participer à la collecte de plusieurs colis lourds. Les travaux qui traitent de la modélisation multi-niveaux pour la résolution multi-agents en sont à leur prémisses (MORGE & NONGAILLARD, 2018b; NONGAILLARD & PICAULT, 2017). Une piste prometteuse réside dans l'adaptation des patrons de conception proposés par MATHIEU, MORVAN et PICAULT (2018) pour la simulation multi-niveaux (zoom, marionnettiste, cohabitation, vue) afin de proposer un modèle opérationnel pour la résolution multi-niveaux de problème.

Négociations concurrentes. Je considère ici un processus de décision collective qui consiste en de multiples négociations concurrentes. Récemment, NIU, REN et ZHANG, 2018 a montré qu'une procédure concurrente de négociation permet d'améliorer le bien-être des agents et le taux de succès des négociations. De plus, comme je l'ai mentionné dans le chapitre 7, la co-occurrence de multiple négociations permet d'améliorer la réactivité du système multi-agents. L'objectif consiste à proposer un **mécanisme générique de coordination dynamique de multiples fils de négociation asynchrones**. La difficulté réside dans le fait qu'un même agent peut participer simultanément dans plusieurs négociations. Dans le contexte de l'allocation multi-agents de tâches, l'inter-dépendance entre les fils de négociation provient des contraintes sur les ressources et sur les tâches. D'une part, les ressources sont partagées, i.e. la réalisation d'une tâche requiert des ressources (e.g. de l'énergie dans le monde des colis) qui ne seront plus disponibles pour la réalisation d'une autre tâche. D'autre part, les tâches peuvent être dépendantes, i.e. la réalisation d'une tâche est conditionnée par la réalisation d'une sous-tâche ou d'une autre tâche qui doit la précéder.

Bien que la littérature sur la négociation multi-attributs est abondante (cf chapitre 2), plus rares sont, à ma connaissance, les travaux qui traitent de négociations interdépendantes. Proposer une stratégie d'offre pour qu'un agent coordonne plusieurs négociations simultanées consiste à résoudre le problème de « l'enchérisseur enthousiaste » (SCHILLO, KRAY & FISCHER, 2002). Si un agent utilise toutes ses ressources dans une négociation et que son offre est rejetée, alors ces ressources ne sont plus disponibles pour d'autres négociations. C'est le cas dans le chapitre 7 où nous avons adopté une approche conservatrice : un agent mobilise toutes ses ressources pour émettre une offre. Cette solution permet de garantir qu'une négociation couronnée de succès améliore strictement l'allocation de tâches et donc le processus de négociation converge. Toutefois, quand plusieurs agents envoient leur proposition au même opposant, seulement l'une d'entre elles est sélectionnée. La procédure n'est donc pas complète. Dans une approche plus libérale, on peut envisager d'ajouter une phase de pré-engagement pour retarder le moment de l'engagement définitif lié à une offre (FISCHER, MÜLLER, PISCHEL & SCHIER, 1995) ou on peut estimer le risque de rupture d'engagement (SCHILLO et al., 2002), comme le fait une compagnie aérienne qui pratique des surréservations. À ma connaissance, la conception d'une stratégie d'offre dans des négociations concurrentes pour l'allocation de tâches

composites à des coalitions d'agents est un problème ouvert.

8.2.4 Verrous technologiques

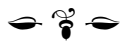
Ce projet de recherche doit aboutir à un catalogue de modules réutilisables pour réduire la taille du code applicatif. Je me focalise plus particulièrement sur :

- la modélisation de traits de comportement composables définis comme des automates à états finis pour la gestion des protocoles multi-agents (e.g. marchandages, enchères);
- la détection de la terminaison superposée aux algorithmes multi-agents sous-jacents;
- l'intégration des accusés de réception dans les protocoles de négociation (en anglais, *business-level agreements*) pour qu'ils soient robustes à la perte de messages possible dans les systèmes distribués.

La mise en œuvre de ce projet de recherche consiste à fournir des patrons de conception pour une approche comportementale de la résolution distribuée de problème.

8.2.5 Résumé analytique

D'une manière plus générale, mon projet de recherche se situe dans la continuité de ceux de EN-DRISS, MAUDET, SADRI et TONI, 2006; NONGAILLARD et MATHIEU, 2011; SANDHOLM, 1998. Il vise à proposer des méthodes de résolution multi-agents de problème basée sur la négociation. Au delà de l'évaluation de l'impact de la rationalité des agents, de la nature des transactions et de la topologie du réseau sur le résultat, je souhaite faire un pas de plus en proposant un processus de négociations décentralisées, concurrentes et multi-niveaux.



Publications choisies

Les noms des étudiants, doctorants et post-doctorants supervisés sont en gras.

Livres

- MORGE, M., PAUCHET, A. & SABOURET, N. (2014). Introduction. *Revue d'Intelligence Artificielle*, 28(1), 7-8.
- HASSAS, S. & MORGE, M. (2013). *Actes des 21èmes Journées Francophones sur les Systèmes Multi-Agents (JFSMA)*. Cépaduès éditions.
- BUISINE, S., MORGE, M. & SABOURET, N. (Éd.). (2010). *Actes du Workshop sur les Agents Conversationnels Animés*. Université Lille 1.

Chapitres de livre

- MORGE, M., MCGINNIS, J., BROMURI, S., MANCARELLA, P., STATHIS, K. & TONI, F. (2013). Argumentative Agents for Service-Oriented Computing. In M. GANZHA & L. C. JAIN (Éd.), *Multiagent Systems and Applications* (T. 45, p. 217-255). Intelligent Systems Reference Library. Springer Berlin Heidelberg.
- DELECROIX, F.**, MORGE, M. & ROUTIER, J.-C. (2012a). 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)* (p. 57-66). Advances in Intelligent and Soft-Computing. Salamanca : Springer.
- MORGE, M. (2011). Arguing over Goals for Negotiation : Adopting an Assumption-Based Argumentation Decision Support System. In C. JAO (Éd.), *Efficient Decision Support Systems - Practice and Challenges From Current to Future* (Chap. 12, p. 211-240). InTech.
- MORGE, M. (2010). Liaising using a Multi-agent System. In B. REBER & C. BROSSAUD (Éd.), *Digital Cognitive Technologies* (Chap. 21, p. 349-360). ISTE/John Wiley et Sons Inc.
- MORGE, M. (2007). Se concerter à l'aide d'un système multi-agents. In B. REBER & C. BROSSAUD (Éd.), *Humanités numériques* (T. 2, p. 117-126). Traité Information-Commande-Communication. Hermès science publication.

Articles dans des revues internationales

- BAERT, Q.**, CARON, A.-C., MORGE, M. & ROUTIER, J.-C. (2018). Fair Multi-Agent Task Allocation for Large Data Sets Analysis. *Knowledge and Information Systems – An International Journal*, 54(3), 591-615. h-index=52, Q2 (Scimago).
- DELECROIX, F.**, MORGE, M., NACHTERGAELE, T. & ROUTIER, J.-C. (2016). Multi-party negotiation with preferences rather than utilities. *Multiagent and Grid Systems – An International Journal*, 12(2), 139-165. h-index=16, Q2 (Scimago).
- MORGE, M. & MANCARELLA, P. (2014). Arguing over Goals for Negotiation : Adopting an Assumption-Based Argumentation Decision Support System. *Group Decision and Negotiation*, 23(5), 979-1012. h-index=50, Q2 (Scimago).
- MORGE, M. & ROUTIER, J.-C. (2007). Debating over heterogeneous descriptions. *Applied Ontology, Special issue on Formal Ontology for Communicating Agents*, 2(3-4), 333-349. h-index=24, Q1 (Scimago).
- MORGE, M. (2006). Collective decision making process to compose divergent interests and perspectives. *Special issue of Artificial Intelligence and Law on Argumentation in Artificial Intelligence and Law*, 75-92. h-index=30, Q1 (Scimago).

Articles dans des conférences internationales

- MORGE, M. & NONGAILLARD, A. (2018a). Distributed Algorithm for Egalitarian Matching between Individuals and Activities with Additively Separable Preferences. In *IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)* (p. 739-746). B (core).
- BAERT, Q.**, CARON, A., MORGE, M. & ROUTIER, J. (2017). Negotiation Strategy of Divisible Tasks for Large Dataset Processing. In *Multi-Agent Systems and Agreement Technologies - 15th European Conference, EUMAS 2017, and 5th International Conference, AT 2017, Évry, France, December 14-15, 2017, Revised Selected Papers* (p. 370-384). C (CORE).
- BAERT, Q.**, CARON, A.-C., MORGE, M. & ROUTIER, J.-C. (2016b). Fair Multi-agent Task Allocation for Large Data Sets Analysis. In Y. DEMAZEAU, T. ITO, J. BAJO & M. J. ESCALONAFI (Éd.), *Proc. of 14th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS), Sevilla, Spain*. (T. 9662). LNAI. C (CORE). Springer.
- DELECROIX, F.**, MORGE, M. & ROUTIER, J.-C. (2014a). Bilateral negotiation of a meeting point in a maze. In Y. DEMAZEAU, F. ZAMBONELLI, J. M. CORCHADO & J. BAJO (Éd.), *Proc. of 12th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS), Salamanca, Spain*. (T. 8473, p. 86-97). LNAI. C (CORE). Springer.
- PIETTE, E.** & MORGE, M. (2014). Practical Application of Matchmaking Problem : Trainee Allocation for Teachers. In Y. DEMAZEAU, F. ZAMBONELLI, J. M. CORCHADO & J. BAJO (Éd.), *Proc. of 12th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS), Salamanca, Spain*. (T. 8473, p. 195-206). LNAI. C (CORE). Springer.
- BROMURI, S. & MORGE, M. (2013a). Multiparty Argumentation Game for Consensual Expansion. In J. FILIPE & A. FRED (Éd.), *Proceedings of the fifth International Conference on Agents and Artificial Intelligence (ICAART)* (T. 1, p. 160-165). C (CORE). INSTICC Press.
- BROMURI, S. & MORGE, M. (2013b). Multiparty Argumentation Game for Consensual Expansion Applied to Evidence Based Medicine. In N. PEEK, R. MARIN MORALES & P. M. (Éd.), *Proc. of 14th Conference on Artificial Intelligence in Medicine (AIME), Murcia, Spain* (T. 7785, p. 33-37). Lecture Note on Artificial Intelligence. B (CORE). Springer-Verlag Berlin Heidelberg.
- DELECROIX, F.**, MORGE, M. & ROUTIER, J.-C. (2013a). A Virtual Selling Agent which is Persuasive and Adaptive. In S. OSSOWSKI (Éd.), *Agreement Technologies* (T. 8, p. 625-645). Law, Governance and Technology Series. Springer Netherlands.

- MORGE, M. & PICARD, G. (2011). Privacy-Preserving Strategy for Negotiating Stable, Equitable and Optimal Matchings. In *Proc. of of the 9th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 11)* (p. 97-102). Advances in Intelligent and Soft-Computing. C (CORE). Salamanca : Springer.
- MATT, P.-A., MORGE, M. & TONI, F. (2010). Combining statistics and arguments to compute trust. In *Proceedings of the ninth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'2010)* (p. 209-216). A* (CORE). IFAAMAS.
- MORGE, M., STATHIS, K. & VERCOUTER, L. (2009). Arguing over motivations within the V3A-architecture for self-adaptation. In *ICAART 2009 - Proceedings of the International Conference on Agents and Artificial Intelligence, Porto, Portugal, January 19 - 21, 2009* (p. 214-219). C (CORE).
- MORGE, M. & BEAUNE, P. (2004). A Negotiation Support System based on a Multi-agent System : specificity and preference relations on arguments. In *19th ACM Symposium on Applied Computing (SAC 2004)* (p. 474-478). B (CORE). Nicosia, Cyprus : ACM.

Lecture Notes in Computer Science

- MORGE, M., ABDEL-NABY, S. & BEAUFILS, B. (2011). Towards a dialectical approach for conversational agents in selling situations. In P. MCBURNEY, I. RAHWAN & S. PARSONS (Éd.), *Argumentation in Multi-Agent Systems, 7th International Workshop, ArgMAS 2010, Toronto, Canada, May 10, 2010. Revised Selected and Invited Papers* (T. 6614, p. 141-158). Lecture Notes in Artificial Intelligence. Berlin, Germany : Springer Verlag.
- MORGE, M. & MANCARELLA, P. (2010). Assumption-Based Argumentation for the Minimal Concession Strategy. In P. MCBURNEY, I. RAHWAN, S. PARSONS & N. MAUDET (Éd.), *Argumentation in Multi-Agent Systems, 6th International Workshop, ArgMAS 2009, Budapest, Hungary, May 12, 2009. Revised Selected and Invited Papers* (T. 6057, p. 114-133). Lecture Notes in Computer Science. Berlin, Germany : Springer Verlag.
- MORGE, M. (2008a). Computable Models of the Law : Languages, Dialogue, Games, Ontologies. In *Computing Argumentation for Decision Making in Legal Disputes* (T. 4884, p. 203-216). Lecture Notes in Artificial Intelligence. Springer-Verlag.
- MORGE, M. (2008b). The hedgehog and the Fox. An Argumentation-Based Decision Support System. In I. RAHWAN, S. PARSONS & C. REED (Éd.), *Argumentation in Multi-Agent Systems : Fourth International Workshop ArgMAS, Revised Selected and Invited Papers* (T. 4946, p. 114-131). Lecture Notes in Artificial Intelligence. Honolulu, Hawaii, USA : Springer-Verlag.
- TONI, F., GRAMMATIKOU, M., KAFETZOGLOU, S., LYMBEROPOULOS, L., PAPAVALASSILEIOU, S., GAERTNER, D., ... GUO, L. (2008). The ArguGRID Platform : An Overview. In *Proc. of the 5th International Workshop on Grid Economics and Business Models (GECON'08)* (T. 5206, p. 217-225). Lecture Notes in Artificial Intelligence. Las Palmas de Gran Canaria, Spain : Springer-Verlag.

Articles dans des revues nationales

- MORGE, M. & NONGAILLARD, A. (2018b). Procédure décentralisée d'affectation d'individus à des activités. *Revue d'Intelligence Artificielle*, 32(5-6), 627-658.
- BAERT, Q., CARON, A.-C., MORGE, M. & ROUTIER, J.-C. (2017). Allocation équitable de tâches pour l'analyse de données massives. *Revue d'Intelligence Artificielle*, 31(4), 401-426. h-index=12, Q4 (Scimago).
- DELECROIX, F., MORGE, M. & ROUTIER, J.-C. (2014c). Réduire l'arbitraire par la négociation quitte à concéder. *Revue d'Intelligence Artificielle*, 28(4), 433-462.
- EVERAERE, P., MORGE, M. & PICARD, G. (2012). Casanova : un comportement d'agent respectant la privacité pour des mariages stables et équitables. *Revue d'Intelligence Artificielle*, 5, 471-494.

MORGE, M. & MATHIEU, P. (2007). Mécanisme de rétribution pour les systèmes P2P d'échange de fichiers. *La revue d'Ingénierie des Systèmes d'Information (ISI), numéro spécial « Les systèmes d'information pair-à-pair », 12, 61-84.*

Articles dans des conférences nationales

- BAERT, Q.,** CARON, A., MORGE, M., ROUTIER, J. & STATHIS, K. (2019b). Stratégie situationnelle pour l'équilibrage de charge. In O. SIMONIN & S. COMBETTES (Éd.), *Actes des 27ème Journées Francophones sur les Systèmes Multi-Agents (JFSMA)* (p. 9-18). Acceptance rate= 40%. Cépaduès éditions.
- MORGE, M. (2018a). Négociation bilatérale par concession : un état de l'art. In G. PICARD, C. LANG & N. MARILEAU (Éd.), *Actes des 25ème Journées Francophones sur les Systèmes Multi-Agents (JFSMA)*. (p. 75-84). Acceptance rate= 22%. Cépaduès éditions.
- NAVETEUR, J., ANTOINE, P., SECQ, Y., ADAM, E., DELELIS, G., MORGE, M., ... NONGAILLARD, A. (2018). Partir Ensemble Séniors (PartEnS) : un projet interdisciplinaire centré utilisateur pour des vacances partagées. In *ERGO'IA 2018*. Acceptance rate= 40%, Bidart, France.
- MORGE, M. & NONGAILLARD, A. (2017). Affectation distribuée d'individus à des activités avec des préférences additivement séparables. In C. GARBAY & G. BONNET (Éd.), *Actes des 25ème Journées Francophones sur les Systèmes Multi-Agents (JFSMA)*. (p. 19-28). Acceptance rate= 27%. Cépaduès éditions.
- BAERT, Q.,** CARON, A.-C., MORGE, M. & ROUTIER, J.-C. (2016a). Allocation équitable de tâches pour l'analyse de données massives. In F. MICHEL & J. SAUNIER (Éd.), *Actes des 24ème Journées Francophones sur les Systèmes Multi-Agents (JFSMA)*. (p. 55-64). Acceptance rate= 32%. Cépaduès éditions.
- BAERT, Q.,** CARON, A.-C., MORGE, M. & ROUTIER, J.-C. (2016c). Stratégie de découpe de tâche pour le traitement de données massives. In C. GARBAY & G. BONNET (Éd.), *Actes des 25ème Journées Francophones sur les Systèmes Multi-Agents (JFSMA)*. (p. 65-75). Acceptance rate= 27%. Cépaduès éditions.
- DELECROIX, F.,** MORGE, M. & ROUTIER, J.-C. (2013b). Négociation bilatérale pour la recherche d'un compromis. In *Actes des 21èmes Journées Francophones sur les Systèmes Multi-Agents (JFSMA)* (p. 19-28). Acceptance rate 30%. Cépaduès éditions.
- GOETGHEBUER-PLANCHON, T.,** MORGE, M. & BROMURI, S. (2012). Jeu d'argumentation multi-parties pour l'enrichissement consensuel. In P. CHEVALIER & B. MERMET (Éd.), *Actes des 20ème Journées Francophones sur les Systèmes Multi-Agents (JFSMA)* (p. 75-84). Acceptance rate 29%. Honfleur : Cépaduès éditions.
- DELECROIX, F.,** MORGE, M. & ROUTIER, J.-C. (2011). Vers un agent dialogique proactif. In *Actes des 19ème Journées Francophones sur les Systèmes Multi-Agents (JFSMA)* (p. 203-212). Acceptance rate 33 %. Cépaduès éditions.
- MORGE, M., EVERAERE, P. & PICARD, G. (2011). Casanova : un comportement d'agent pour l'équité des mariages préservant la confidentialité. In *Actes des 19ème Journées Francophones sur les Systèmes Multi-Agents (JFSMA)* (p. 191-200). Acceptance rate 33 %. Cépaduès éditions.
- MORGE, M., STATHIS, K. & VERCOUTER, L. (2008). Argumentation sur les motivations propres dans l'architecture V3A pour des agents auto-adaptatifs. In *Actes des 16ème Journées Francophones sur les Systèmes Multi-Agents (JFSMA)* (p. 149-158). Acceptance rate 22 %. Cépaduès éditions.
- MORGE, M., MCGINNIS, J., BROMURI, S., TONI, E., MANCARELLA, P. & STATHIS, K. (2007b). Vers une architecture modulaire d'agent argumentatif pour la composition de services. In P. MATHIEU (Éd.), *Actes des 15ème Journées Francophones sur les Systèmes Multi-Agents (JFSMA)* (p. 33-42). Acceptance rate 32 %. Hermès.

- MORGE, M. & MATHIEU, P. (2006). Conception de mécanisme décentralisé pour les système P2P d'échange de fichiers. In V. CHEVRIER (Éd.), *Actes des 14ème Journées Francophones sur les Systèmes Multi-Agents (JFSMA)* (p. 145-158). Acceptance rate= 32%. Hermès.
- MORGE, M., ROUTIER, J.-C., SECQ, Y. & T., D. (2006). Comment atteindre un accord sur une représentation? In V. CHEVRIER (Éd.), *Actes des 14ème Journées Francophones sur les Systèmes Multi-Agents (JFSMA)* (p. 167-170). Hermès.

Posters et démonstrations

- BAERT, Q.**, CARON, A., MORGE, M., ROUTIER, J. & STATHIS, K. (2019a). Adaptive Multi-agent System for Situated Task Allocation. In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)* (p. 1790-1792). A* (CORE). IFAAMAS.
- MORGE, M. (2019a). Répartition des tâches pour la collecte de colis : démonstration. In O. SIMONIN & S. COMBETTES (Éd.), *Actes des 27ème Journées Francophones sur les Systèmes Multi-Agents (JFSMA)*. (p. 185-186). Cépaduès éditions.
- DELECROIX, F.**, MORGE, M. & ROUTIER, J.-C. (2014b). Bilateral negotiation of a meeting point in a maze : Demonstration. In Y. DEMAZEAU, F. ZAMBONELLI, J. M. CORCHADO & J. BAJO (Éd.), *Proc. of 12th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS), Salamanca, Spain.* (T. 8473). LNAI. C (CORE). Springer.
- EVERAERE, P., PICARD, G. & MORGE, M. (2013). Minimal Concession Strategy for Reaching Fair, Optimal and Stable Marriages (Extended Abstract). In *Proc. of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)* (p. 1319-1320). A* (CORE). IFAAMAS.
- DELECROIX, F.**, MORGE, M. & ROUTIER, J.-C. (2012b). A Virtual Selling Agent which is Proactive and Adaptive : Demonstration. In *Proc. of of the 10th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 12)* (p. 231-236). Advances in Intelligent and Soft-Computing. Salamanca : Springer.
- BROMURI, S., UROVI, V., MORGE, M., TONI, F. & STATHIS, K. (2009). A multi-agent system for service discovery, selection and negotiation (demonstration). In *8th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2009), Budapest, Hungary, May 10-15, 2009, Volume 2* (p. 1395-1396). A* (CORE).

Articles dans des workshops internationaux

- DELECROIX, F.**, MORGE, M. & ROUTIER, J.-C. (2012c). An algorithm for active learning of lexicographic preferences. In M. PIRLOT & V. MOUSSEAU (Éd.), *Proc. of the workshop from Multiple Criteria Decision Aiding to Preference Learning* (p. 115-122).
- MORGE, M., **ABDEL-NABY, S.** & BEAUFILS, B. (2010). Towards a dialectical approach for conversational agents in selling situations. In *The Seventh Proceedings of the seventh International Workshop on Argumentation in Multi-Agent Systems (ArgMAS'2010)* (p. 127-144). IFAAMAS.
- MORGE, M. & STATHIS, K. (2008). The Agent Argumentation Architecture Revisited. In *Proc. of the Sixth European Workshop on Multi-Agent Systems (EUMAS'08), Bath, UK* (p. 1-15).
- MORGE, M., MCGINNIS, J., BROMURI, S., TONI, F., MANCARELLA, P. & STATHIS, K. (2007a). Toward a Modular Architecture of Argumentative Agents to Compose Services. In *Proc. of the Fifth European Workshop on Multi-Agent Systems (EUMAS)* (p. 1-15). Hammamet, Tunisia.

Rapports

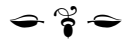
MORGE, M. (2005). *Système dialectique multi-agents pour l'aide à la concertation* (thèse de doct., École Nationale Supérieure des Mines de Saint-Étienne).

Logiciels

- BAERT, Q.,** CARON, A.-C., MORGE, M. & ROUTIER, J.-C. (2019). MAS4Data - Multiagent systems for analyzing very large data sets. Récupérée 21 mars 2019, à partir de <https://github.com/cristal-smac/mas4data>
- MORGE, M. (2019b). ScaSMATA - Scalable Situated Multi-Agent Task Allocation. Récupérée 21 mars 2019, à partir de <https://github.com/maximemorge/ScaSMATA>
- MORGE, M. (2018b). ScaIA - Scalable Individuals/activities Algorithms. Récupérée 21 mars 2019, à partir de <https://github.com/cristal-smac/ScaIA>
- DELECROIX, F.,** MORGE, M., **NACHTERGAELE, T.** & ROUTIER, J.-C. (2014). NegoMAZE - Negotiation game to meet in a MAZE. Récupérée 21 mars 2019, à partir de <http://cristal.univ-lille.fr/~morge/software.html#NegoMaze>
- MORGE, M. (2014). ScaMA - Scalable Matching. Récupérée 21 mars 2019, à partir de <https://forge.lifl.fr/ScaMa>
- PIETTE, E.,** MORGE, M. & PICARD, G. (2014). DSMP - Distributed resolution of Stable Marriage Problems. Récupérée 21 mars 2019, à partir de <https://forge.lifl.fr/DSMP>
- MORGE, M. & MANCARELLA, P. (2006). MARGO - A Multiattribute ARGumentation framework for Opinion explanation. Récupérée 21 mars 2019, à partir de <http://margo.sourceforge.net/>

Conférence invitée

MORGE, M. (2017). Affectation distribuée d'individus à des activités. Journée de modélisation et simulation en sciences humaines et sociales, Nancy.



Bibliographie

- AGNETIS, A., BILLAUT, J., GAWIEJNOWICZ, S., PACCIARELLI, D. & SOUKHAL, A. (2014). *Multiagent Scheduling - Models and Algorithms*. Springer. (Cf. p. 92).
- ALPERN, S. & GAL, S. (2003). *The theory of search games and rendezvous*. Springer. (Cf. p. 45).
- AMGOUD, L., DIMOPOULOS, Y. & MORAITIS, P. (2007). A Unified and General Framework for Argumentation-based Negotiation. In *Proc. of the International Joint Conference on Autonomous Agents and Multi-Agent Systems* (p. 963-970). (Cf. p. 24, 37).
- AMGOUD, L. & PRADE, H. (2006). Explaining qualitative decision under uncertainty by argumentation. In *Proc. of the National Conference on Artificial Intelligence* (p. 16-20). (Cf. p. 24).
- AN, B., LESSER, V., IRWIN, D. & ZINK, M. (2010). Automated Negotiation with Decommitment for Dynamic Resource Allocation in Cloud Computing. In *Proc. of the International Conference on Autonomous Agents and MultiAgent Systems* (p. 981-988). (Cf. p. 92).
- ARROW, K. J. (1951). *Social Choice and Individual Values*. John Wiley & Sons. (Cf. p. 17).
- ARROW, K. J. (1958). Utilities, Attitudes, Choices : A Review Note. *Econometrica*, 26(1), 1-23. (Cf. p. 13).
- ATTIYA, G. & HAMAM, Y. (2006). Task allocation for maximizing reliability of distributed systems : A simulated annealing approach. *Journal of parallel and Distributed Computing*, 66(10), 1259-1266. (Cf. p. 92).
- AYDOGAN, R., BAARSLAG, T., HINDRIKS, K. V., JONKER, C. M. & YOLUM, P. (2013). Heuristic-Based Approaches for CP-Nets in Negotiation. In *Complex Automated Negotiations* (p. 113-123). Springer. (Cf. p. 16, 19, 21).
- AYDOGAN, R., SANCHEZ, V., JULIAN, V., BROEKENS, J. & JONKER, C. (2014). Guest editorial : computational approaches for conflict resolution in decision making : new advances and developments. *Cybernetics and Systems : An International Journal*, 45(3), 217-221. (Cf. p. 12).
- AZIZ, H., BRANDT, F. & SEEDIG, H. G. (2013). Computing desirable partitions in additively separable hedonic games. *Artificial Intelligence*, 195, 316-334. (Cf. p. 72).
- BAARSLAG, T., FUJITA, K., GERDING, E. H., HINDRIKS, K., ITO, T., JENNINGS, N. R., ... WILLIAMS, C. R. (2013). Evaluating Practical Negotiating Agents : Results and Analysis of the 2011 International Competition. *Artificial Intelligence*, 198, 73-103. (Cf. p. 13, 19, 21).
- BENCH-CAPON, T. & PRAKKEN, H. (2006). Justifying actions by accruing arguments. In *Proc. of the International Conference on Computational Models of Argument* (p. 247-258). (Cf. p. 24).
- BOULWARE, L. (1969). *The Truth About Boulwarism : Trying to Do Right Voluntarily*. Bureau of National Affairs. (Cf. p. 16).
- BOUTILIER, C., BRAFMAN, R. I., DOMSHLAK, C., HOOS, H. H. & POOLE, D. (2004). CP-nets : A Tool for Representing and Reasoning with Conditional Ceteris Paribus Preference Statements. *Journal of Artificial Intelligence Research*, 21, 135-191. (Cf. p. 14, 16).

BIBLIOGRAPHIE

- BOUTILIER, C. & HOOS, H. H. (2001). Bidding languages for combinatorial auctions. In *Proc. of the International Joint Conference on Artificial Intelligence* (p. 1211-1217). (Cf. p. 14).
- BRACCIALI, A., DEMETRIOU, N., ENDRISS, U., KAKAS, A., LU, W. & STATHIS, K. (2006). Crafting the Mind of PROSOCS—Agents. *Applied Artificial Intelligence*, 20(2-4), 105-131. (Cf. p. 27).
- BRITO, I. & MESEGUER, P. (2005). Distributed Stable Matching Problems. In *Principles and Practice of Constraint Programming* (T. 3709, p. 152-166). Lecture Notes in Computer Science. Springer-Verlag. (Cf. p. 6, 50, 56, 58, 63).
- BROMURI, S. & STATHIS, K. (2007). Situating Cognitive agents in GOLEM. In *Proc. of the Engineering Environment-Mediated Multiagent Systems Conference (EEMMAS)* (p. 76-93). (Cf. p. 25, 27, 90).
- CHEN, B., POTTS, C. N. & WOEGINGER, G. J. (1998). Handbook of combinatorial optimization. (Chap. A review of machine scheduling : Complexity, algorithms and approximability, p. 1493-1641). Springer. (Cf. p. 91).
- CLEMEN, R. T. (1996). *Making Hard Decisions*. Duxbury Press. (Cf. p. 29).
- COULOURIS, G. F., DOLLIMORE, J. & KINDBERG, T. (2012). *Distributed systems : concepts and design. 5th Edition*. Pearson education. (Cf. p. 92).
- DAVIS, R. & SMITH, R. G. (1980). *Negotiation as a Metaphor for Distributed Problem Solving* (rapp. tech. N° AIM-624). Massachusetts Institute of Technology. (Cf. p. 12).
- DEAN, J. & GHEMAWAT, S. (2004). MapReduce : Simplified Data Processing on Large Clusters. In *Proc. of the Symposium on Operating System Design and Implementation* (p. 137-150). (Cf. p. 81, 82, 86).
- DECANDIA, G., HASTORUN, D., JAMPANI, M., KAKULAPATI, G., LAKSHMAN, A., PILCHIN, A., ... VOGELS, W. (2007). Dynamo : Amazon's Highly Available Key-value Store. In *Proc. of the ACM SIGOPS Symposium on Operating Systems Principles* (p. 205-220). (Cf. p. 81).
- DREZE, J. & GREENBERG, J. (1980). Hedonic coalitions : Optimality and stability. *Econometrica*, 48, 987-1003. (Cf. p. 76).
- DUNG, P. M., MANCARELLA, P. & TONI, F. (2007). Computing ideal sceptical argumentation. *Artificial Intelligence, Special Issue on Argumentation*, 171(10-15), 642-674. (Cf. p. 38).
- DURFEE, E. H. & LESSER, V. R. (1989). Distributed Artificial Intelligence. (Chap. Negotiating Task Decomposition and Allocation Using Partial Global Planning, T. 2, p. 229-243). Morgan Kaufmann Publishers. (Cf. p. 12).
- ENDRISS, U., MAUDET, N., SADRI, F. & FRANCESCA, T. (2006). Negotiating Socially Optimal Allocations of Resources. *Journal of Artificial Intelligence Research*, 25, 315-348. (Cf. p. 12).
- ENDRISS, U., MAUDET, N., SADRI, F. & TONI, F. (2006). Negotiating Socially Optimal Allocations of Resources. *Journal of Artificial Intelligence Research*, 25(1), 315-348. (Cf. p. 96).
- FARATIN, P., SIERRA, C. & JENNINGS, N. . (1998). Negotiation Decision Functions for Autonomous Agents. *Robotics and Autonomous Systems*, 24(3-4), 159-182. (Cf. p. 4, 13, 15, 19, 21).
- FATIMA, S. S., WOOLDRIDGE, M. & JENNINGS, N. R. (2006). Multi-Issue Negotiation with Deadlines. *Journal of Artificial Intelligence Research*, 27, 381-417. (Cf. p. 13, 15, 19, 20).
- FERBER, J. (1995). *Les systèmes multi-agents : vers une intelligence collective*. InterEditions. (Cf. p. 1).
- FISCHER, K., MÜLLER, J. P., PISCHEL, M. & SCHIER, D. (1995). A Model for Cooperative Transportation Scheduling. In *Proc. of the International Conference on Multiagent Systems* (p. 109-116). (Cf. p. 95).
- FISHER, K., SCHILLO, M. & SIEKMANN, J. (2004). Holonic multiagent system : A foundation for the organisation of multiagent system, Holonic and Multi-Agent System for Manufacturing. *Lecture Notes in Computer Science*, 2744, 71-80. (Cf. p. 95).
- GALE, D. & SHAPLEY, L. S. (1962). College Admissions and the Stability of Marriage. *The American Mathematical Monthly*, 69, 9-14. (Cf. p. 6, 12, 50, 51, 56, 58, 59, 73).

- GARG, S. K., VENUGOPAL, S., BROBERG, J. & BUYYA, R. (2013). Double auction-inspired meta-scheduling of parallel applications on global grids. *Journal of Parallel and Distributed Computing*, 73(4), 450-464. (Cf. p. 92).
- GARTNER, D. & TONI, F. (2007). CaSAPI : a system for credulous and sceptical argumentation. In *Proc. of ArgNMR* (p. 80-95). (Cf. p. 38).
- GATTI, N., DI GIUNTA, F. & MARINO, S. (2008). Alternating-offers Bargaining with One-Sided Uncertain Deadlines : an Efficient Algorithm. *Artificial Intelligence*, 172(8-9), 1119-1157. (Cf. p. 13, 15, 19, 20).
- GELAIN, M., PINI, M. S., ROSSI, F., VENABLE, K. B. & WALSH, T. (2010). Local search algorithms on the Stable Marriage Problem : Experimental Studies. In *Proc. of the European Conference on Artificial Intelligence* (p. 1085-1086). IOS Press. (Cf. p. 56-58, 63).
- GHANEM, M., AZAM, N., BONIFACE, M. & FERRIS, J. (2006). Grid-enabled workflows for industrial product design. In *Proc. of the International Conference on e-Science and Grid Computing*. (Cf. p. 24).
- GOODMAN, L. A. & MARKOWITZ, H. (1952). Social welfare functions based on individual rankings. *American Journal of Sociology*, 257-262. (Cf. p. 41).
- GRABISCH, M. (1997). k-order additive discrete fuzzy measures and their representation. *Fuzzy Sets and Systems*, 92(167-189), 165-114. (Cf. p. 13).
- GRAHAM, R. L. (1969). Bounds on multiprocessing timing anomalies. *SIAM journal on Applied Mathematics*, 17(2), 416-429. (Cf. p. 86).
- GUSFIELD, D. & IRVING, R. W. (1989). *The stable marriage problem : structure and algorithms*. Cambridge, MA, USA : MIT Press. (Cf. p. 60).
- GUTTMAN, R. H., MOUKAS, A. G. & MAES, P. (1998). Agent-mediated Electronic Commerce : a Survey. *Knowledge Engineering Review*, 13(2), 147-159. (Cf. p. 12).
- HARIRI, A. M. A. & POTTS, N., Chris. (1991). Heuristics for scheduling unrelated parallel machines. *Computers & operations research*, 18(3), 323-331. (Cf. p. 91).
- HARSANYI, J. C. (1977). *Rational Behavior and Bargaining with Incomplete Information*. Cambridge University Press. (Cf. p. 15).
- HAYES-ROTH, B. (1985). A blackboard architecture for control. *Artificial Intelligence*, 26(3), 251-321. (Cf. p. 2).
- HEWITT, C. (1977). Viewing control structures as patterns of passing messages. *Artificial Intelligence*, 8, 323-364. (Cf. p. 2).
- HEWITT, C., BISHOP, P. & STEIGER, R. (1973). A Universal Modular ACTOR Formalism for Artificial Intelligence. In *Proc. of the International Joint Conference on Artificial Intelligence* (p. 235-245). (Cf. p. 75, 76, 86, 90).
- HORN, W. (1973). Minimizing average flow time with parallel machines. *Operations Research*, 21(3), 846-847. (Cf. p. 91).
- HOROWITZ, E. & SAHNI, S. (1976). Exact and approximate algorithms for scheduling nonidentical processors. *Journal of the ACM*, 23(2), 317-327. (Cf. p. 91).
- IBARRA, O. H. & KIM, C. E. (1977). Heuristic algorithms for scheduling independent tasks on nonidentical processors. *Journal of the ACM*, 24(2), 280-289. (Cf. p. 91).
- IRVING, R. W., LEATHER, P. & GUSFIELD, D. (1987). An Efficient Algorithm for the "Optimal" Stable Marriage. *Journal of ACM*, 34(3), 532-543. (Cf. p. 57).
- ITO, T., KLEIN, M. & HATTORI, H. (2008). Multi-issue Negotiation Protocol Among Agents with Nonlinear Utility Functions. *Multiagent and Grid Systems*, 4(1), 67-83. (Cf. p. 13).
- JENNINGS, N., FARATIN, P., LOMUSCIO, A., PARSONS, S., WOOLDRIDGE, M. & SIERRA, C. (2001). Automated Negotiation : Prospects Methods and Challenges. *Group Decision and Negotiation*, 10(2), 199-215. (Cf. p. 3, 11).

BIBLIOGRAPHIE

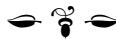
- JIANG, Y.-C. & JIANG, J. (2005). A multi-agent coordination model for the variation of underlying network topology. *Expert systems with applications*, 29(2), 372-382. (Cf. p. 92).
- JIANG, Y. (2016). A survey of task allocation and load balancing in distributed systems. *IEEE Transactions on Parallel and Distributed Systems*, 27(2), 585-599. (Cf. p. 92).
- JIANG, Y. & JIANG, J. (2008). Contextual resource negotiation-based task allocation and load balancing in complex software systems. *IEEE Transactions on Parallel & Distributed Systems*, (5), 641-653. (Cf. p. 92).
- KAKAS, A. & MORAITIS, P. (2003). Argumentative-Based Decision-Making for Autonomous Agents. In *Proc. of the second international joint conference on Autonomous agents and multiagent systems* (p. 883-890). (Cf. p. 24).
- KAKAS, A. & MORAITIS, P. (2006). Adaptive Agent Negotiation via Argumentation. In *Proc. of the International Joint Conference on Autonomous Agents and Multi-Agent Systems* (p. 384-391). (Cf. p. 24).
- KANG, Q.-M., HE, H., SONG, H.-M. & DENG, R. (2010). Task allocation for maximizing reliability of distributed computing systems using honeybee mating optimization. *Journal of Systems and Software*, 83(11), 2165-2174. (Cf. p. 92).
- KEENEY, R. L. & RAIFFA, H. (1976). *Decisions with Multiple Objectives*. John Wiley & Sons. (Cf. p. 13).
- KERSTEN, G. E. (2001). Modeling Distributive and Integrative Negotiations. Review and Revised Characterization. *Group Decision and Negotiation*, 10(6), 493-514. (Cf. p. 13).
- KERSTEN, G. E. & NORONHA, S. J. (1999). WWW-based negotiation support : design, implementation, and use. *Decision Support Systems*, 25(2), 135-154. (Cf. p. 12).
- KERSTEN, G. E., NORONHA, S. J. & TEICH, J. (2000). Are all e-commerce negotiations auctions? In *Proceedings of the International Conference on the Design of Cooperative Systems* (T. 58, p. 387). (Cf. p. 17).
- KERSTEN, G. & LAI, H. (2007). Negotiation Support and E-negotiation Systems : An Overview. *Group Decision and Negotiation*, 16(6), 553-586. (Cf. p. 12).
- KNUTH, D. (1971). *Marriage stables*. Montréal, Canada : Les Presses de l'Université de Montréal. (Cf. p. 50, 52, 57).
- KRAUS, S. (1997). Negotiation and cooperation in multi-agent environments. *Artificial Intelligence*, 94(1), 79-97. (Cf. p. 12).
- KRAUS, S. (2001). *Strategic Negotiation in Multiagent Environments*. MIT Press. (Cf. p. 13, 14).
- KRAUS, S., WILKENFELD, J. & ZLOTKIN, G. (1995). Multiagent Negotiation under Time Constraints. *Artificial Intelligence*, 75(2), 297-345. (Cf. p. 13).
- KUHN, H. W. (1955). The Hungarian method for the assignment problem. *Naval Research Logistics (NRL)*, 2(1-2), 83-97. (Cf. p. 93).
- LENSTRA, J. K., SHMOYS, D. B. & TARDOS, E. (1990). Approximation algorithms for scheduling unrelated parallel machines. *Mathematical programming*, 46(1-3), 259-271. (Cf. p. 91).
- LIN, R., KRAUS, S., BAARSLAG, T., TYKHONOV, D., HINDRIKS, K. & JONKER, C. M. (2012). GENIUS : an Integrated Environment for Supporting the Design of Generic Automated Negotiators. *Computational Intelligence*, 48-70. (Cf. p. 16).
- LOMUSCIO, A., WOOLDRIDGE, M. & JENNINGS, N. (2003). A Classification Scheme for Negotiation in Electronic Commerce. *Group Decision and Negotiation*, 12(1), 31-56. (Cf. p. 11).
- LOPES, F., WOOLDRIDGE, M. & NOVAIS, A. Q. (2008). Negotiation Among Autonomous Computational Agents : Principles, Analysis and Challenges. *Artificial Intelligence Review*, 29(1), 1-44. (Cf. p. 12).
- LYMBEROPOULOS, L., BROMURI, S., STATHIS, K., KAFETZOGLOU, S. & GRAMMATIKO, M. (2007). Towards a P2P Discovery Framework for an Argumentative Agent Technology assisted GRID. In *Proc. of the CoreGRID Workshop on Grid Programming Model, Grid and P2P systems Architectures, Grid Systems, Tools, and Environments*. (Cf. p. 24).

- MARTELLO, S., SOUMIS, F. & TOTH, P. (1997). Exact and approximation algorithms for makespan minimization on unrelated parallel machines. *Discrete applied mathematics*, 75(2), 169-188. (Cf. p. 91).
- MATHIEU, P., MORVAN, G. & PICAULT, S. (2018). Multi-level agent-based simulations : Four design patterns. *Simulation Modelling Practice and Theory*, 83, 51-64. (Cf. p. 95).
- MCBURMEY, P. & PARSONS, S. (2002). Games that agents play : A formal framework for dialogues between autonomous agents. *Journal of Logic, Language and Information*, 11(3), 315-334. (Cf. p. 24).
- MCGINNIS, J., STATHIS, K. & TONI, F. (2011). A formal model of agent-oriented virtual organisations and their formation. *Multiagent and Grid Systems*, 7(6), 291-310. (Cf. p. 38).
- MICHEL, F. (2007). Le modèle IRM4S. De l'utilisation des notions d'influence et de réaction pour la simulation de systèmes multi-agents. *Revue d'intelligence artificielle*, 21(5-6), 757-779. (Cf. p. 94).
- MITCHELL, T. (1997). *Machine learning*. McGraw-Hill. (Cf. p. 52).
- MOULIN, H. J. (2002). *Fair Division and Collective Welfare*. The MIT Press. (Cf. p. 17, 52, 72).
- MÜLLER, J. (1996). Foundations of Distributed Artificial Intelligence. (Chap. Negotiation Principles, p. 211-228). John Wiley & Sons. (Cf. p. 11).
- NAJJAR, A., BOISSIER, O. & PICARD, G. (2017). AQUAMan : An Adaptive QoE-Aware Negotiation Mechanism for SaaS Elasticity Management. In *Proc. of the International Conference on Autonomous Agents and MultiAgent Systems* (p. 1655-1657). (Cf. p. 92).
- NIU, L., REN, F. & ZHANG, M. (2018). Feasible Negotiation Procedures for Multiple Interdependent Negotiations. In *Proc. of the International Conference on Autonomous Agents and MultiAgent Systems* (p. 641-649). (Cf. p. 95).
- NONGAILLARD, A. & MATHIEU, P. (2011). Reallocation Problems in Agent Societies : A Local Mechanism to Maximize Social Welfare. *Journal of Artificial Societies and Social Simulation*, 14(3), 21. (Cf. p. 13, 92, 96).
- NONGAILLARD, A. & PICAULT, S. (2017). Modélisation multi niveau du bien-être social dans un SMA. *Revue d'intelligence artificielle*, 31(6), 709-734. (Cf. p. 94, 95).
- OSBORNE, M. & RUBINSTEIN, A. (1990). *Bargaining and Markets*. The Academic Press. (Cf. p. 14).
- OSBORNE, M. J. & RUBINSTEIN, A. (1994). *A Course in Game Theory*. MIT Press. (Cf. p. 17).
- OUERDANE, W., MAUDET, N. & TSOUKIAS, A. (2007). Arguing over Actions that Involve Multiple Criteria : A Critical Review. In S. BERLIN (Éd.), *Proc. of the European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty* (T. 4724, p. 308-319). Lecture Notes in Computer Science. (Cf. p. 24).
- ÖZTÜRK, M., TSOUKIAS, A. & VINCKE, P. (2012). Multiple Criteria Decision Analysis : State of the Art. (Chap. Preference Modelling). Springer. (Cf. p. 13).
- PARSONS, S., RODRIGUEZ-AGUILAR, J. A. & KLEIN, M. (2011). Auctions and Bidding : A Guide for Computer Scientists. *ACM Computing Surveys*, 43(2), 10 :1-10 :59. (Cf. p. 11, 12).
- PARSONS, S., SIERRA, C. & JENNINGS, N. R. (1998). Agents That Reason and Negotiate By Arguing. *Journal of Logic and Computation*, 8(3), 261-292. (Cf. p. 13).
- PICAULT, S. (2013). *From multi-agent simulation to multi-level simulation. Reifying the interactions*. (Habilitation à diriger des recherches, Université des Sciences et Technologie de Lille - Lille I). (Cf. p. 94).
- PRUITT, D. G. (1981). *Negotiation Behavior*. Academic Press. (Cf. p. 3, 12).
- RAHWAN, I. (2005). Guest Editorial : Argumentation in Multi-Agent Systems. *Autonomous Agents and Multiagent Systems*, 11(2), 115-125. (Cf. p. 24).
- RAHWAN, I., RAMCHURN, S. D., JENNINGS, N. R., MCBURNEY, P., PARSONS, S. & SONENBERG, L. (2004). Argumentation-based negotiation. *Knowledge Engineering Review*, 18(4), 343-375. (Cf. p. 24, 37).

BIBLIOGRAPHIE

- RANDALL, D. (1980). *Report on the Workshop on Distributed AI*. MIT Artificial Intelligence Laboratory. (Cf. p. 2).
- ROBERTSON, D. (2004). Multi-agent Coordination as Distributed Logic Programming. In *Proc. of the International Conference on Logic Programming* (p. 416-430). (Cf. p. 34).
- ROBU, V., SOMEFUN, D. J. A. & LA POUTRÉ, J. A. (2005). Modeling Complex Multi-issue Negotiations Using Utility Graphs. In *Proc. of the International Joint Conference on Autonomous Agents and Multiagent Systems* (p. 280-287). (Cf. p. 14).
- ROSENSCHEIN, J. & ZLOTKIN, G. (1994). *Rules of Encounter - Designing Conventions for Automated Negotiation among Computers*. MIT Press. (Cf. p. 4, 14, 15, 19, 20, 42).
- ROTH, A. E. & OLIVEIRA SOTOMAYOR, M. A. (1990). *Two-sided Matching; A Study in Game-theoretic Modeling and Analysis*. Cambridge University Press. (Cf. p. 50, 52).
- RUBINSTEIN, A. (1982). Perfect Equilibrium in a Bargaining Model. *Econometrica*, 50(1), 97-102. (Cf. p. 14, 15, 19, 20).
- RUBINSTEIN, A. (1985). A Bargaining Model with Incomplete Information About Time Preferences. *Econometrica*, 53(5), 1151-1172. (Cf. p. 15, 19, 20).
- RUSSELL, S. & NORVIG, P. (2003). *Artificial Intelligence : A Modern Approach*. (Chap. Chapter 4 : Informed Search and Exploration, p. 94-136). 2nd edition. Pearson Education. (Cf. p. 76).
- SANDHOLM, T. (1998). Contract types for satisficing task allocation. In *Proceedings of the AAAI spring symposium : Satisficing models* (p. 23-25). (Cf. p. 96).
- SANDHOLM, T. & VULKAN, N. (1999). Bargaining with Deadlines. In *Proc. of the National Conference on Artificial Intelligence* (p. 44-51). (Cf. p. 13, 15, 19, 20).
- SCHELLING, T. (1960). *The Strategy of Conflict*. Harvard University Press. (Cf. p. 12).
- SCHILLO, M., KRAY, C. & FISCHER, K. (2002). The eager bidder problem : a fundamental problem of DAI and selected solutions. In *Proc. of the International Conference on Autonomous Agents and MultiAgent Systems* (p. 599-606). (Cf. p. 95).
- SEN, A. K. (1970). *Collective Choice and Social Welfare*. North-Holland. (Cf. p. 17, 72).
- SHEHORY, O. & KRAUS, S. (1998). Methods for task allocation via agent coalition formation. *Artificial Intelligence*, 101(1-2), 165-200. (Cf. p. 91, 93).
- SMITH, R. G. (1980). The Contract Net Protocol : High-Level Communication and Control in a Distributed Problem Solver. *IEEE Transactions on computers*, 29(12), 1104-1113. (Cf. p. 83).
- STAHL, I. (1972). *Bargaining Theory*. Stockholm School of Economics. (Cf. p. 14).
- STATHIS, K. (2018). Enabling Decentralised Decision-Making in AI - an Agent Environments Perspective. In *Actes des journées Francophones sur les systèmes multi-agents* (p. 7). (Cf. p. 93).
- SYCARA, K. (1998). Multiagent Systems. *Artificial Intelligence Magazine*, 19(2), 79-92. (Cf. p. 2).
- TURNER, J., MENG, Q., SCHAEFER, G. & SOLTOGGIO, A. (2018). Distributed Strategy Adaptation with a Prediction Function in Multi-Agent Task Allocation. In *Proc. of the International Conference on Autonomous Agents and MultiAgent Systems* (p. 739-747). (Cf. p. 91, 92).
- VINYALS, M., MACARTHUR, K. S., FARINELLI, A., RAMCHURN, S. D. & JENNINGS, N. R. (2014). A Message-Passing Approach to Decentralized Parallel Machine Scheduling. *The Computer Journal*, 57(6), 856-874. (Cf. p. 93).
- WALSH, W. E. & WELLMAN, M. P. (1998). A market protocol for decentralized task allocation. In *Proc. of the International Conference on Multi Agent Systems* (p. 325-332). (Cf. p. 91, 92).
- WEISS, G. (Éd.). (2013). *Multiagent Systems : A modern Approach to Distributed Artificial Intelligence*. second edition. The MIT Press. (Cf. p. 1).
- WEYNS, D., HELLEBOUGH, A. & HOLVOET, T. (2005). The packet-world : A test bed for investigating situated multi-agent systems. In *Software Agent-Based Applications, Platforms and Development Kits* (p. 383-408). Springer. (Cf. p. 93).
- WHITE, T. (2015). *Hadoop : The Definitive Guide, 4th Edition*. O'Reilly Media. (Cf. p. 81).
- WOOLDRIDGE, M. (2002). *An Introduction to Multiagent Systems*. John Wiley & Sons. (Cf. p. 1).

- WURMAN, P. R., WELLMAN, M. P. & WALSH, W. E. (1998). The Michigan Internet AuctionBot : A configurable auction server for human and software agents. In *Proc. of the International Conference on Autonomous Agents* (p. 301-308). (Cf. p. 11).
- YIN, P.-Y., YU, S.-S., WANG, P.-P. & WANG, Y.-T. (2007). Task allocation for maximizing reliability of a distributed system using hybrid particle swarm optimization. *Journal of Systems and Software*, 80(5), 724-735. (Cf. p. 92).
- YOKOO, M., DURFEE, E., ISHIDA, T. & KUWABARA, K. (1998). The Distributed Constraint Satisfaction Problem : Formalization and Algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 10, 673-685. (Cf. p. 56).
- ZACCARO, S. J. & LOWE, C. A. (1988). Cohesiveness and Performance on an Additive Task : Evidence for Multidimensionality. *The Journal of Social Psychology*, 128(4), 547-558. (Cf. p. 72).
- ZAHARIA, M., CHOWDHURY, M., DAS, T., DAVE, A., MA, J., MCCAULEY, M., ... STOICA, I. (2012). Resilient distributed datasets : A fault-tolerant abstraction for in-memory cluster computing. In *Proc. of the USENIX conference on Networked Systems Design and Implementation* (p. 15-28). (Cf. p. 81).
- ZAVIDOVIQUE, B., SUVONVORN, N. & SEETHARAMAN, G. S. (2005). A Novel representation and algorithms for (quasi) stable marriages. In *Proc. of the International Conference on Informatics in Control, Automation and Robotics* (p. 63-70). INSTICC Press. (Cf. p. 56-58).
- ZEUTHEN, F. (1930). *Problems of Monopoly and Economic Welfare*. London : G. Routledge et Sons. (Cf. p. 15).



Index

- équilibre
 - de Nash, 17
- algorithme
 - de Gale-Shapley, 6, 50–53, 55, 56, 58, 59, 73
 - de Gale-Shapley orienté hôpital, 60, 63, 65
 - de Gale-Shapley orienté résident, 59, 63, 65
 - de Prim, 47
 - inclusif, 73
 - recursive backtracker, 47
 - sélectif, 73
 - SML2, 56, 58, 59
 - SMP, 63
 - ZigZag, 56, 58, 59
- alternative, 29, 45
 - équitable, 4
 - non-dominée, 41
- appariement, 53
- approche
 - argumentative, 3, 15, 24, 27, 37
 - heuristique, 15
 - Théorie des Jeux, 15
- argument, 27
- bien-être
 - égalitaire, 17, 53, 72
 - équitable, 4, 7, 59, 90
 - de Nash, 17
 - utilitaire, 4, 7, 17, 59, 72, 90
- cadre d'argumentation, 5, 29, 33, 36, 37, 90
- cohésion sociale, 72
- complexité
 - algorithmique, 17
 - communicationnelle, 18, 58
- concession, 3, 14, 89, 93
 - fonction, 16
 - minimale suffisante, 15
- couple bloquant, 53
- décision, 29, 34
- date butoir, 13
- distribution, 17, 59
- fonction d'utilité, 4, 7, 8, 13, 15, 93
 - multi-attributs, 13, 16
- garantie de succès, 4, 6, 7, 18, 36, 43, 45, 59
- garantie de terminaison, 58, 59
- graphe de préférences, 13
- méthode
 - Swing, 6, 7, 53, 54, 59, 60, 63
 - Swing++, 55, 59
- makespan, 91, 93, 94
- mariage stable, 6, 50
- négociation
 - à base de propositions, 39
 - argumentative, 37
 - bilatérale, 5, 7, 13, 23
 - mono-attribut, 6, 7, 13, 39
 - multi-attributs, 5, 13, 24
 - multi-parties, 13
 - par enchère, 7, 82, 84
- Pareto
 - dominance, 4, 36, 42, 52, 89
 - optimum de, 7, 16, 72, 73, 78
- parfait, 72
- problème

- de mariage stable, 59
- des hôpitaux/résidents, 50, 59
- profil de stratégies, 43
- protocole, 5, 14, 25, 33, 34, 37, 83
 - asymétrique, 14
 - d'offres alternées, 14, 15
 - de concession monotone, 4, 14, 15, 53, 60, 64
 - dynamique, 14
 - simultané, 14
 - symétrique, 6, 14, 42, 48
- rationalité individuelle, 17
- relation
 - d'incomparabilité, 40
 - de préférence, 6, 13, 39, 51, 64, 93
- stabilité, 17, 54, 59
 - au sens de Nash, 71
 - contractuellement individuelle, 71
 - de cœur, 71
 - individuelle, 71
- stabilité, 53
- stratégie, 14
 - coopérative, 43
 - de Zeuthen, 15
 - tactique, 16
 - Boulware, 16
 - Conceder, 16
 - temporisatrice, 43
- taux d'opposition, 17
- taux de réduction, 13
- valeur de réserve, 4, 13, 15

INDEX

Troisième partie

Pièces annexes



Maxime Morge

Maître de conférences
en Informatique

- 17 novembre 1977
- 4 rue du Pavé bleu
F-59650 Villeneuve d'Ascq
France
- +33 (0)3 20 33 77 12
- [https://pro.univ-lille.fr/
maxime-morge](https://pro.univ-lille.fr/maxime-morge)
- Maxime.Morge@univ-lille.fr

À propos

Maxime Morge est maître de conférences en Informatique à l'Université de Lille depuis 2009. Il est responsable du parcours BIMST (*Business Intelligence Management Skills Training*) du Master SIAD. Son activité de recherche concerne l'Intelligence Artificielle, en particulier la négociation multi-agents.

Encadrement

Sameh Abdel-Naby (PostDoc)

Quentin Baert (Doctorat + M2)

Fabien Delecroix (Doctorat + M2)

Eric Piette (M2)

Santiago Vilareal (M2)

Thomas Nachtergaele (M1)

Taux d'encadrement

Activité de recherche

Négociation multi-agents pour la résolution de problème.

Mots-clefs : Informatique, Intelligence Artificielle (IA), Système Multi-Agents (SMA).

Activité d'enseignement

Analyse et conception des systèmes d'information pour le traitement, l'analyse et la visualisation des données.

Cours : Traitement de données massives, Entrepôt de données, Analyse et conception des systèmes d'information, Datavisualisation.

Formation

2001-2005 Thèse de doctorat en Informatique Mines de Saint-Étienne
Système dialectique multi-agents pour l'aide à la concertation

Expérience

- Depuis 2009 Maître de conférence en Informatique Université de Lille
Faculté de Sciences Économiques et Sociales
Centre de Recherche en Informatique, Signal et Automatique de Lille
- 2006-2009 Chercheur contractuel Université di Pisa
Projet européen FP6 ArguGRID
(Argumentation as a foundation for the semantic grid)
- 2005-2006 ATER IUT A Université de Lille
Attaché Temporaire d'Étude et de Recherche
- 2004-2005 ATER Université Jean-Monnet
Attaché Temporaire d'Étude et de Recherche

Publications choisies

- Quentin Baert, Anne-Cécile Caron, Maxime Morge et Jean-Christophe Routier (2018). « Fair Multi-Agent Task Allocation for Large Data Sets Analysis ». In : *Knowledge and Information Systems – An International Journal* 54.3. h-index=52, Scimago Q2, p. 591-615
- Fabien Delecroix, Maxime Morge, Thomas Nachtergaele et Jean-Christophe Routier (2016). « Multi-party negotiation with preferences rather than utilities ». In : *Multiagent and Grid Systems – An International Journal* 12.2. h-index=16, Scimago Q2, p. 139-165
- Maxime Morge et Paolo Mancarella (2014). « Arguing over Goals for Negotiation : Adopting an Assumption-Based Argumentation Decision Support System ». In : *Group Decision and Negotiation* 23.5. h-index=50, Scimago Q2, p. 979-1012
- Maxime Morge et Jean-Christophe Routier (2007). « Debating over heterogeneous descriptions. » In : *Applied Ontology, Special issue on Formal Ontology for Communicating Agents* 2.3-4. h-index=24, Scimago Q2, p. 333-349
- Maxime Morge (jan. 2006). « Collective Decision making process to compose divergent interests and perspectives ». In : *Special issue of Artificial Intelligence and Law on Argumentation in Artificial Intelligence and Law*. h-index=25, Scimago Q3, p. 75-92

Projet

- 2018-2019 Projet bilatéral avec Royal Holloway University of London
- 2017 Projet MAS4Data soutenu par le défi CNRS Mastodons
- 2016-2018 Projet « Partir ensemble » Région Nord Pas-de-Calais
- 2015 Projet « Dynamiques de groupe au sein de réseaux sociaux numériques et spontanés » CNRS PEPS MoMIS Humain
- 2012-2014 Projet « Vendeur Virtuel Ubiquitaire » labellisé par le PICOM
- 2009-2012 Projet européen ARGUMENTATION as a foundation for the semantic Grid

Curriculum vitae détaillé

Maxime MORGE

Né le 17 novembre 1977 à Saint-Étienne

Nationalité : française

Situation familiale : marié, trois enfants



Contact

e-mail : Maxime.Morge@univ-lille.fr

web : <https://pro.univ-lille.fr/maxime-morge>

Adresse professionnelle

Université de Lille

CRISAL / Cité scientifique

F-59655 Villeneuve d'Ascq

Phone : +33 (0)3 20 33 77 12

Fax : +33 (0)3 28 77 85 37

Adresse personnelle

4 rue du Pavé bleu

F-59650 Villeneuve d'Ascq, France

Tel : +33 (0)6 33 14 11 37

FONCTIONS ASSURÉES

2009- Maître de conférence

Lieu : Université de Lille, Faculté de Sciences Économiques et Sociales.

2006-2009 Chercheur contractuel.

Lieu : Dipartimento di Informatica, Università di Pisa.

2008-2009 Vacataire

Lieu : Université Jean-Monnet de Saint-Étienne et Mines de Saint-Étienne.

2005-2006 Attaché Temporaire d'Étude et de Recherche

Lieu : Département d'Informatique de l'IUT A, Université Lille 1.

FORMATION

Cursus universitaire

2006 Qualification CNU aux fonctions de Maître de Conférence en Informatique (Section 27)

2001-2005 Thèse de doctorat spécialité informatique

Lieu : Mines de Saint-Étienne

Titre : *Système dialectique multi-agents pour l'aide à la concertation*

Mots clés : Informatique, IA, SMA, Argumentation

Financement :

— Action Concertée Incitative Ville (allocation de recherche fléchée)

— ADNT (projet région Rhône-Alpes)

Encadrants : Olivier BOISSIER, Philippe BEAUNE

Date de soutenance : 20 juin 2005

Mention : Très honorable

Composition du jury :

Rapporteurs :

Philippe MATHIEU Professeur d'Informatique, Université Lille 1
Jean-Paul SANSONNET Directeur de recherche CNRS, LIMSI-CNRS

Examineurs :

Philippe BEAUNE Maître assistant, Mines Saint-Étienne
Olivier BOISSIER Professeur d'Informatique, Mines Saint-Étienne
Jean-Luc KONING Professeur d'Informatique, INPG ESISAR Valence
Nicolas MAUDET Maître de conférences, Université Paris IX

Compétences techniques

Langages de programmation : AWK, C, CAML, COBOL, Java, JavaScript, Perl, PHP, Python, PostScript, Prolog, Scala, Scheme, SQL.

Langage de formatage : \LaTeX , HTML, CSS.

ACTIVITÉS DE RECHERCHE

Le travail réalisé s'intègre aux recherches menées dans l'équipe SMAC du laboratoire CRISTAL. Le travail concerne le codage de comportements intelligents pour des intelligences artificielles. Ce projet de recherche vise à proposer des processus computationnels décentralisés pour la gestion des conflits entre des agents autonomes en interaction en soutenant leur raisonnement, leur prise de décision et la négociation.

Mes travaux contribuent à l'étude des modèles de négociation multi-agents. La question à laquelle ils répondent est : « Quels sont les mécanismes opératoires à mettre en œuvre pour la négociation ? ». Cette question est abordée non pas d'un point de vue Mathématique, c'est-à-dire en se focalisant sur l'existence de stratégies qui exhibent un certain nombre de propriétés, mais d'un point de vue Informatique. Dans une perspective algorithmique, je construis des modèles de raisonnement, des protocoles et des stratégies. Les cadres de négociation proposés sont systématiquement implémentés, évalués de manière expérimentale et appliqués soit via des scénarios jouets ou pour des applications pratiques s'appuyant sur des données réelles ou des problèmes industriels.

Encadrement

Thèse de doctorat

[2016-] Quentin BAERT. Système multi-agents pour le traitement de données massives. 33 %

[2011-2015] Fabien DELECROIX. Dialoguer pour décider. 50 %

Stage de recherche

[2018] Simon VANDEPUTTE. Résolution multi-agents pour l'affectation d'individus à des activités. 50 %

[2016] Quentin BAERT. Allocation équitable de tâches pour l'analyse de données massives. 33 %

[2013] Eric PIETTE. Problème d'appariement : application à l'affectation de stages aux Professeurs des écoles. 100 %

[2011] Fabien DELECROIX. Agent dialogique proactif. 50 %

[2010] Santiago VILAREAL. Distributed Constraint-based Optimisation and Social Choice. 50 %

Projet de recherche

[2019] Quentin BRIAND et Florian LECOINTE Un pour tous, tous pour un. Projet encadré M1. 100 %

[2018] Julien COURTEVILLE. Affectation distribuée d'individus à des activités. Option label recherche L3. 100 %

[2016] Rémi GODVILLE. Réseaux sociaux numériques et environnements socio-technique simulés. Projet technique M2. 50 %

- [2015] Quentin BAERT. Allocation équitable de tâches pour l'analyse de données massives. Projet recherche M2. 50 %
- [2015] Thomas NACHTERGAELE. Négociation d'un point de rencontre. Projet de recherche M2. 50 %
- [2014] Thomas NACHTERGAELE. Négociation d'un point de rencontre. Projet encadré M1. 50 %
- [2012] Eric PIETTE. Sur la stabilité, l'équité et l'optimalité des mariages. Projet de recherche M2. 50 %
- [2012] Manuella DJAMBOU. Site web de covoiturage basé sur l'algorithme de Gale-Shapley. Projet de fin d'étude, Polytech'Lille. 100 %
- [2011] Thibaut GOETGHEBUER PLANCHON. Jeu argumentatif de demande d'information multi-parties. Projet de recherche M2. 100 %
- [2010] Fabien DELECROIX. Agent dialogique proactive. Projet de recherche M2. 50 %

Chercheur contractuel

- [2011] Isabelle PIALAPRAT. Conception de stratégies marketing pour un agent vendeur virtuel. Ingénieur de recherche. 50 %
- [2011] Adrien NOUVEAU. Développement et validation d'un agent vendeur virtuel. Ingénieur de recherche. 50 %
- [2009-2010] Sameh ABDEL-NABY. Business-Driven Negotiation Strategies for Proactive Embodied Conversational Agents in E-Commerce. Chercheur post-doctorant. 50 %

Animation scientifique

- 2016- Membre du comité de pilotage du collège SMA de l'AFIA
- 2013-2018 Membre du comité consultatif du groupe de travail "Affects, Compagnons Artificiels et Interactions" (ACAI) du GDR I3
- 2012-2013 Co-animateur du comité consultatif du groupe de travail "Affects, Compagnons Artificiels et Interactions" (ACAI) du GDR I3

Comité de programme

- 2015-2020 International Conference on Agents and Artificial Intelligence (ICAART)
- 2018-2019 International Conference on Tools for Artificial Intelligence (ICTAI)
- 2008-2019 Journées Francophones sur les Systèmes Multi-Agents (JFSMA)
- 2018 International Conference on Intelligent Technologies and Applications (INTAP)
- 2018 International Joint Conferences on Ambient Intelligence (AmI)
- 2018 1st Workshop on Actors, Agents, Assistants, Avatars (4A'18)
- 2016-2018 Conférence Nationale en Intelligence Artificielle (CNIA)
- 2012-2017 Workshop on Agent Based Computing : from Model to Implementation IX (ABC)
- 2012-2018 Workshop on Artificial Companions, Affects and Interaction (WACAI)
- 2015 Workshop on Personal Health Systems for Chronic Diseases (PHSCD) at Pervasive Health
- 2014 14th International Conference on Principles of Knowledge Representation and Reasoning (KR)
- 2014 7th international conference on Autonomous Agents and Multiagent Systems (AAMAS).
- 2011-2014 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT).
- 2012 18^{ème} conférence Reconnaissance des Formes et Intelligence Artificielle (RFIA).
- 2011 22th International Joint Conference on Artificial Intelligence (IJCAI).
- 2011 International meeting on Group Decision and Negotiation (GDN).

Comité d'organisation

2018 Journée du laboratoire CRISAL.

2013 Plate-forme d'Intelligence Artificielle (PFIA).

2010 Workshop sur les Agents Conversationnels Animés (WACA).

Relecteur associé

- Journal of Artificial Intelligence (2011-2017).
- 26th International Joint Conference on Artificial Intelligence (IJCAI 2017).
- International Journal of Conflict Management (2016).
- Special issue on "Argumentation in Multi-Agent Systems" in Argument & Computation (2016).
- Argument & Computation (2015).
- LNCS Transactions on Computational Collective Intelligence (2015).
- Journal of Autonomous Agents and Multi-Agent Systems (2015).
- Transactions on Autonomous and Adaptive Systems (2014).
- Numéro spécial de RIA consacré aux meilleurs papiers des JFSMA 2012 et JFSMA 2015.
- Numéro spécial de RIA consacré aux meilleurs papiers du WACAI 2010.
- Journal of Artificial Intelligence in Medicine (2012).
- Special issue on "Negotiation and Argumentation in Artificial Intelligence" for the International Journal on Artificial Intelligence (2010).
- Journal of Decision Support Systems (2010).
- Journal of Logic and Computation, Oxford Press (2009).
- The Computer Journal, Oxford Press (2009).
- IEEE Journal of Intelligent Systems (2007).
- 3rd International Conference on Agents and Artificial Intelligence (ICAART 2011).
- International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 2010).
- International Conference on Autonomous Agents and MultiAgent System (AAMAS 2010-2008).
- International Conference on Logic Programming (ICLP 2007).

Expertises

2018- Comité de suivi de thèse de Walid Rifi, LITIS, Rouen.

2016-2018 Évaluation de convention CIFRE auprès de l'ANRT

2012- Crédit Impôt Recherche (CIR)

2017 Innovational Research Incentives Scheme, The Netherlands Organisation for Scientific Research (NWO)

2016 Centre International de Mathématiques et d'Informatique de Toulouse (Labex CIMIT)

2016 Agence Nationale pour la Recherche (ANR)

2011 European network of national research funding agencies supporting the collaboration between European states in long-term research on ICT during the call "Beyond Autonomic Systems – the Challenge of Consciousness" (BASCC)

2011 European Coordinated Research on Long-term Challenges in Information and Communication Sciences & Technologies ERA-NET (CHIST-ERA)

Jury de thèse

- 2019** Membre du jury de thèse de doctorat de Jean-Baptiste Louvet « Collaboration humain-machine à l'aide de motifs dialogiques pour la réalisation d'une tâche complexe », LITIS/INSA, Rouen.
- 2016** Membre du jury de thèse de doctorat de Nadine Glas « Modélisation des stratégies verbales d'engagement dans les interactions humain-agent », Telecom ParisTech, Paris.
- 2014** Membre du jury de thèse de doctorat de Georges Dubus « Modélisation logique pour la personnalisation sur le Web », Supelec, Paris.
- 2013** Membre du jury de thèse de doctorat de Sabrina Campano « COR-E : un modèle pour la simulation d'agents affectifs fondé sur la théorie COR », LIP6/UPMC, Paris.
- 2011** Membre du jury de thèse de doctorat de Julien Dumont « Système multi-agents pour le diagnostic pluridisciplinaire », Mines Saint-Étienne.

Récompense

- 2014** Third Prize of IBM Demonstration Award @ PAAMS 2014 *Bilateral Negotiation of a Meeting Point in a Maze* by Fabien DELECROIX, Maxime MORGE and Jean-Christophe ROUTIER.
- 2011** Prix du meilleur papier des 19^{ème} Journées Francophones sur les Systèmes Multi-Agents (JFSMA'11) à Valenciennes pour l'article *Casanova : un comportement d'agent pour l'équité des mariages préservant la privacité* par Maxime MORGE, Patricia EVERAERE et Gauthier PICARD.

Projets

- 2018** Projet bilatéral avec Royal Holloway University of London
- porteur du projet et coordination du séjour de mobilité de thèse à l'étranger pour Quentin Baert
- 2017** Projet MAS4Data soutenu par le défi CNRS Mastodons
- porteur du projet et coordinateur entre le laboratoire Clersé (économétrie) et le laboratoire CRISTAL (informatique)
- 2016-2018** Projet « Partir ensemble » (PartENS) soutenu par la Région Nord Pas-de-Calais entre la mairie de Lille et les laboratoires SCALab (ergonomie), LAMIH/CRISTAL (informatique).
- chercheur principal pour le laboratoire CRISTAL
- 2015** Projet « Dynamiques de groupe au sein de réseaux sociaux numériques et spontanés » soutenu par le CNRS PEPS MoMIS Humain entre le laboratoire Clersé (sociologie) et le laboratoire CRISTAL (informatique)
- chercheur principal pour le laboratoire CRISTAL
- 2009-2012** Projet « Vendeur Virtuel Ubiquitaire » (VVU) labellisé par le pôle des industries du commerce (PI-COM) entre Decathlon, Orange, Becquet, Skema business school, INRIA, Mines Douai et LIFL
- Chercheur principal pour le laboratoire LIFL
- 2006-2009** Projet européen FP6 "ARGUmentation as a foundation for the semantic Grid" (ArguGRID) entre Imperial (UK), RHUL (UK), Inforsense (UK), COSMO-ONE (GR), ICCS (GR), GMV (SP), UNIFI (IT) et AIT (TH).
- Chercheur principal pour Università di Pisa, chef d'équipe WP2.
- 2005-2006** Contrat Plan Etat Région TAC de la région Nord-Pas de Calais et les fonds européens du FEDER
- chercheur

Publications

La figure 8.3 représente le nombre de mes publications depuis mon doctorat. On peut noter que la diminution de la production en 2010 puis 2015 correspondent à ma prise de fonction à l'Université Lille 1 en 2009 et à une charge d'enseignement plus importante en 2013/2014 respectivement, comme expliqué dans la section suivante. Cette figure montre mon implication constante et soutenue dans les activités de recherche.

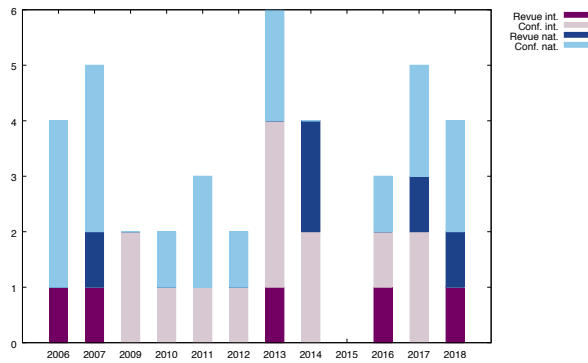


FIGURE 8.3 – Nombre de publications

ACTIVITÉS D'ENSEIGNEMENT

Mon projet pédagogique consiste à enseigner l'ensemble des techniques, outils et méthodes pour l'analyse et la conception des systèmes d'information (systèmes opérationnels, systèmes décisionnels) sous la forme d'entrepôt de données ou de lac de données pour le traitement, l'analyse et la visualisation des données. Les cours suivants s'intègrent dans le Master SIAD (Système d'Information et Aide à la Décision) de la Faculté des Sciences Économiques et Sociales que cela soit dans le parcours *Business Intelligence Management Skills Training* (BIMST), dans le parcours *Business Intelligence Technical Skills Training* (BITST) ou dans le parcours *Data Sciences* (DS).

La figure 8.4 représente le nombre d'heures (équivalent TD) que j'ai dispensé à Lille. On peut noter que le faible nombre d'heures réalisées en 2009 et en 2016 correspondent à une dispense et à un congé maladie. Le pic en 2013 correspond au remplacement d'un collègue décédé. Cette figure montre mon implication constante et soutenue dans les activités d'enseignement.

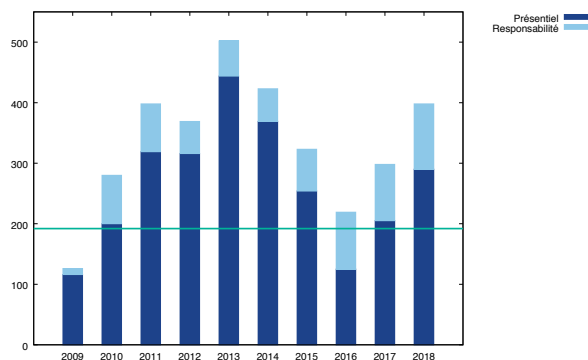


FIGURE 8.4 – Volume d'enseignement (eqTD)

Traitement des données massives

Résumé : Cet enseignement porte sur la problématique du traitement des données massives. Il a pour objectif d'introduire les technologies de l'éco-système Hadoop.

Public : M2 BIMST en S4

Volume : 20h de CM + 15h de TD soit 45h eqTD

Contenu : Après une brève introduction des langages de script orienté donnée (AWK), ce cours aborde le système de fichier HDFS (interface en ligne), le patron de conception MapReduce. Sont également étudiés le langage de manipulation de flot de données Pig Latin, le langage d'interrogation des données HiveQL et le framework de calcul distribué Spark.

Contrôle : QCM, Projet, rendu de TP / examen sur machine.

Pré-requis : Programmation orientée objet, Entrepôt de données.

Entrepôt de données

Résumé : Ce cours sert d'armature théorique aux autres matières de l'UE concernée (BO, Sunopsis, Projet décisionnel). Il concerne la problématique de modélisation multidimensionnelles des données.

Public : M1 DS en S2, M2 BIMST en S3 et M2 BITST en S4

Volume : 3x20h de CM soit 60h eqTD

Contenu : Dans ce cours, nous présentons les motivations et les objectifs d'un Datawarehouse (DWH). Nous introduisons les modèles en étoile, en flocon et en constellation. Nous définissons ce qu'est un cube OLAP et les différentes requêtes qui peuvent être réalisées sur un hypercube (dice, slice, drill down/up, ...). Le cours met l'accent sur la problématique de l'alimentation d'un DWH à partir de données relationnelles. L'ensemble des notions sont illustrées par des exemples réalisés à l'aide d'un outil de reporting (Qlikview) et en SQL.

Contrôle : QCM, rendu de TP / examen sur table.

Pré-requis : Modèle entité-relation (MERISE), Modèle relationnel, SQL.

Analyse et conception de systèmes d'information

Résumé : Cet enseignement porte sur la conception des bases de données relationnelle depuis le MCD jusqu'au MLD. Il s'agit du socle de connaissances requis pour tous les cours ayant ensuite trait aux systèmes d'information : SGBD, DWH, ETL, SQL etc.

Public : M1 DS en S1

Volume : 2*14h de CM + 3*2h de TD soit 48 eqTD

Pré-requis : Aucun.

Contenu : Ce cours présente les SGBDR et la méthode MERISE (MCD et MLD).

Contrôle : QCM, TP noté / examen sur feuille.

Datavisualisation

Résumé : Ce cours porte sur le couplage bases de données/web. Il consiste à donner aux étudiants les savoirs et les savoirs-faire pour créer un « site dynamique » pour la visualisation de données.

Public : M2 BITST eau S3 et M1 DS au S2

Volume : 2x20 h de CM soit 60 eqTD

Contenu : Le cours présente les fondements de l'Internet et du web. Sont présentés le langage de formatage HTML5, les feuilles de style en cascade (CSS), ainsi que les techniques de programmation (JavaScript, Python) pour la visualisation données (D3.js).

Contrôle : QCM, projet / examen sur machine.

Pré-requis : Algorithmique.

Responsabilité pédagogique

2013- Responsable du parcours BIMST du Master SIAD

2011- Tuteur académique des stages pour le parcours BIMST du Master SIAD

2018 Vice-président du comité de sélection pour le recrutement d'un maître de conférence, Université de Lille

2015 Vice-président du comité de sélection pour le recrutement d'un maître de conférence, Université Lille 1

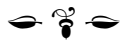
2013 Membre du comité de sélection pour le recrutement d'un maître de conférence, INSA Rouen

DIVERS

Langues : français, anglais (lu, parlé, écrit), italien (niveau moyen, parlé langage quotidien).

Hobbies : trompette, course à pied, tennis de table.

Autres : permis B.



Résumé

Mes travaux s'intègrent aux recherches menées dans l'équipe SMAC de CRISTAL en Intelligence Artificielle Distribuée. Mon projet de recherche vise à proposer des processus décentralisés pour la gestion des conflits entre des agents autonomes en interaction, en spécifiant leurs comportements. Je défends la thèse selon laquelle le mécanisme opératoire nécessaire pour résoudre un conflit pratique est la concession. La difficulté commune abordée par ces travaux réside dans le fait que l'information dont disposent les agents est imparfaite. Le verrou scientifique auquel ce projet se confronte consiste à garantir la qualité de l'accord obtenu.

Abstract

My work is part of the research done by the SMAC team in the laboratory CRISTAL in Distributed Artificial Intelligence. My research project aims to propose decentralized processes for handling conflict between interacting autonomous agents by specifying their behaviour. I defend the thesis that the operating mechanism required to solve a practical conflict is the concession. The main issue raised by this work lies in the fact that the information available to the agent is imperfect. The scientific challenge with this project is to warrant the quality of the agreement.