



جامعة العربي التبسي - تبسة
Université Larbi Tébessi - Tébessa

République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la
recherche scientifique

Université Larbi Tébessi - Tébessa



كلية العلوم الدقيقة وعلوم الطبيعة و الحياة
FACULTÉ DES SCIENCES EXACTES
ET DES SCIENCES DE LA NATURE ET DE LA VIE

Faculté des Sciences Exactes et des Sciences de la Nature et de la Vie

Département : Mathématiques et Informatique

Mémoire de fin d'étude

Pour l'obtention du diplôme de **MASTER**

Domaine : Mathématiques et Informatique

Filière : Informatique

Option : Systèmes d'information

Thème

Allocation de ressources multi agent sous information incomplète

Présenté Par :

Khelif Ahlem

Devant le jury :

<i>Mr. Hakim Bendjenna</i>	<i>Pr</i>	<i>Université Larbi Tébessa</i>	<i>Président</i>
<i>Mr. Abdelmalek Matrouh</i>	<i>MCB</i>	<i>Université Larbi Tébessa</i>	<i>Examineur</i>
<i>Mr. Ali-Abdelatif Betouil</i>	<i>MCB</i>	<i>Université Larbi Tébessa</i>	<i>Encadreur</i>

Date de soutenance : 22 /06/2019

Making it simple is probably the most sophisticated goal in the world

Steve Jobs

Dédicace

À mon père et ma mère.

À ma petite famille.

À mes sœurs et mon frère.

Remerciements

Je remercie, tout d'abord, Allah qui m'a éclairé le chemin du savoir. Ensuite, je tiens à remercier beaucoup mon encadreur le Docteur Ali-Abdelatif Betouil pour ses conseils, ses orientations et son aide.

Je remercie également les membres de jury, le Professeur Hakim Bendjenna, le docteur Abdelmalek Matrouh d'avoir accepté de juger ce travail.

Je remercie mes parents, ma famille et mes amis pour leurs encouragements.

ملخص

حظيت مشكلة تصميم الأنظمة متعددة الوكلاء باهتمام متزايد في السنوات الأخيرة. عندما نتحدث عن وكيل ، يتعين علينا أن نأخذ بشكل واسع ، لأن الوكيل يمكن أن يمثله الروبوت ، والشخص الافتراضي ، وبرنامج الكمبيوتر في مشروعنا ، ركزنا على تنفيذ نظام متعدد الوكلاء لتخصيص الموارد. كل وكيل لديه تفضيلاته الخاصة على مجموعة (من الموارد ، يمكن أن تكون هذه التفضيلات متناقضة) الصراعات بين الوكلاء يعد تخصيص الموارد مشكلة تكمن في تقاطع المجالين: الاقتصاد والمعلوماتية. يركز الباحثون في المجال الأول على تحديد الخصائص والصفات الجيدة لتخصيص الموارد. بالنسبة للمجال الثاني ، يركز الباحثون على الآليات اللازمة للحصول على هذه المخصصات. هو مجال بحث يهتم بدراسة آليات توزيع مجموعة من الموارد بين مجموعة (MARA) تخصيص الموارد متعدد الوكلاء من وكلاء البرامج. في هذا العمل ، سوف نقدم نهجنا الذي يتناول توزيع الموارد الموزع بين الوكلاء الأنايين لإعادة توزيع الموارد غير القابلة للتجزئة دون تعويض مالي. على وجه الخصوص ، لا يشارك الوكلاء تفضيلاتهم ويجب عليهم تحقيق بدل جيد النوعية يقاس من حيث الرفاه الاجتماعي. يتم اقتراح آليات تسمح للوكلاء بإدارة المخاطر وقبول العروض التي توصل الوكلاء إلى اتفاق.

الكلمات المفتاحية: الأنظمة متعددة الوكلاء ، تخصيص الموارد ، استراتيجية التداول ، الرعاية الاجتماعية

Résumé

Le problème de la conception de systèmes multi-agents a fait l'objet d'une attention croissante ces dernières années. Quand on parle d'un agent, il faut le prendre au sens large, car l'agent peut être représenté par le robot, la personne virtuelle, un programme informatique...

Dans notre projet, On a concentré sur la mise en œuvre du système multi-agents pour allocation des ressources. Chaque agent a ses propres préférences sur un ensemble de ressources, ces préférences peuvent être contradictoires (conflits entre les agents).

L'allocation de ressources est un problème qui se situe à l'intersection des deux domaines : L'économie et l'informatique. Les chercheurs du premier domaine concentrent leurs travaux sur la détermination des propriétés et les bonnes qualités des allocations de ressources. Pour le deuxième domaine, les chercheurs se focalisent sur les mécanismes nécessaires pour obtenir ces allocations.

L'allocation de ressources multi agent (MARA) est un domaine de recherche qui s'intéresse à l'étude des mécanismes de distribution d'un ensemble de ressources entre un ensemble d'agents logiciels.

Dans ce travail, on va présenter notre approche qui aborde l'allocation distribuée de ressources entre les agents égoïstes pour la redistribution des ressources indivisibles sans compensation monétaire. Particulièrement les agents ne partagent pas leurs préférences et doivent atteindre une allocation de bonne qualité mesurée en terme de bien-être social. On propose des mécanismes qui permettent aux agents la gestion des risques et d'accepter des offres qui permettent d'amener les agents à un accord.

Mot-clés : Systèmes multi-agents, Allocation de ressources, stratégie de négociation, bien être social.

Abstract

The problem of designing multi-agent systems has received increasing attention in recent years. When we talk about an agent, we have to take it in a broad sense, because the agent can be represented by the robot, the virtual person, a computer program...

In our project, we focused on the implementation of the multi-agent system for resource allocation. Each agent has their own preferences on a set of resources, these preferences can be contradictory (conflicts between the agents).

Resource allocation is a problem that lies at the intersection of the two domains: Economics and Informatics. Researchers in the first area concentrate their work on the determination of properties and the good qualities of resource allocations. For the second area, researchers focus on the mechanisms needed to obtain these allocations.

Multi-Agent Resource Allocation (MARA) is a research area that is interested in studying the distribution mechanisms of a set of resources between a set of software agents.

In this work, we will present our approach which addresses the distributed allocation of resources among the selfish agents for the redistribution of indivisible resources without monetary compensation. In particular, the agents do not share their preferences and must achieve a good quality allowance measured in terms of social welfare. Mechanisms are proposed that allow agents to manage risk and accept offers that bring the agents to an agreement.

Keywords: Multi-Agent Systems, Resource Allocation, Trading Strategy, Social Welfare.

Table des matières

Dédicace	ii
Remerciement.....	iii
ملخص	iv
Résumé	v
Abstract	vi
Table des matières	vii
Liste des figures	xi
Liste des tableaux	xii
Introduction générale.....	1
Chapitre 1 Allocation de ressources.....	3
1.1-Introduction.....	4
1.2Allocation de ressources: définition.....	4
1.3 Le ressources.....	4
1.3.1 Continues ou Discrètes.....	4
1.3.2 Divisible ou non.....	5
1.3.3 Partageable ou non.....	5
1.3.4 Dynamiques ou statiques.....	5
1.3.5 Mono-Item ou Multi-Item.....	5
1.3.6 Ressources ou tâches.....	5
1.4 Représentation de préférences.....	6
1.4.1 Définition de préférence	6
1.4.2 Structures de préférences.....	6
1.4.3 Préférences quantitatives.....	7
1.4.3.1 Forme explicite « Bundle forme »	7
1.4.3.2 Forme k-additive.....	7
1.5 Bien-être social.....	9
1.5.1 L'Optimalité de Pareto.....	9
1.5.2 Bien être social utilitaire.....	9
1.5.3 Bien être social égalitaire	10
1.5.4 Produit de Nash.....	10
1.5.5 Bien être social élitiste.....	10

1.5.6 L'absence d'envie.....	11
1.6 Procédures d'allocation	11
1.6.1 Centralisé vs distribué.....	12
1.7 Les transactions.....	13
1.7.1 Définition.....	13
1.7.2 Types de transaction.....	13
1.7.2.1 Les transactions simples.....	13
1.7.2.2 Les transactions groupées.....	14
1.7.2.3 Les transactions échange.....	14
1.7.2.4 Les transactions multi-agents.....	15
1.7.2.5 Les transactions combinées.....	15
Chapitre 2 La négociation dans les systèmes multi agent.....	16
2.1 Agents et Systèmes multi-agents.....	17
2.1.1 Les agents.....	17
2.1.2 Les modèles d'agents.....	17
2.1.2.1 Modèles d'agents réactifs.....	17
2.1.2.2 Modèles d'agents cognitifs	18
2.1.2.3 Modèles d'agents hybrides.....	18
2.1.2 Environnement.....	18
2.1.3 L'interaction.....	19
2.1.3.1 La communication.....	19
2.1.3.2 La coopération.....	19
2.1.3.3 La coordination.....	19
2.1.3.4 La négociation.....	20
2.1.4 L'organisation.....	20
2.1.4.1 Les topologies de l'organisation.....	20
2.1.4.2 L'émergence.....	21
2.1.4.3 L'auto-organisation.....	21
2.1.2 Système multi-agents.....	21
2.1.2.1 Définition système multiagents.....	21
2.1.2.2 Méthodologies de conception des systèmes multi-agents.....	22
2.1.2.2.1 Les méthodologies utilisant UML et constituant une extension des méthodes orientées objet.....	22
2.1.2.2.1.1 MaSE.....	22

2.1.2.2.1.2 AAI	22
2.1.2.3 Les méthodologies organisationnelle	23
2.1.2.3.1 Aalaadin	23
2.1.2.3.2 Gaia	23
2.1.2.4 Les méthodologies formelles	24
2.1.2.4.1 DESIRE	24
2.1.2.4.2 PASSI	24
2.1.3 Domaines d'application des systèmes multi-agents	25
2.2 La négociation	26
2.2.1 Définition	26
2.2.1.1 Négociation entre agents égo-centrés	26
2.2.2 Les grandes types de négociation	26
2.2.2.1 Négociation compétitive	26
2.2.2.2 Négociation coopérative	26
2.2.3 Composants du modèles de négociation	27
2.2.4 Théorie de la négociation	27
2.2.5 Les formes de négociation	27
2.2.6 Les protocoles de négociation	28
2.2.6.1 Définition	28
2.2.6.2 Classification des protocoles de négociation	28
2.2.6.3 Protocoles de négociation	29
2.2.6.3.1 Le protocole Contract Net	29
2.2.6.3.2 Le protocole Contract Net étendu	30
2.2.6.3.3 Le protocole Kasbah	30
2.2.6.3.4 Le protocole AuctionBot	31
2.2.7 Négociation basée sur la théorie des jeux	31
2.2.7.1 Efficacité Pareto	32
2.2.7.2 Rationalité individuelle (RI)	32
2.2.7.3 Stabilité	32
2.2.7.4 Equilibre de Nash	32
2.2.7.5 La dilemme du prisonnier	32
2.2.8 Enchères	33
2.2.8.1 Protocoles d'enchères	33
2.2.8.1.1 Enchère anglaise (premier-prix offre-publique) – chaque participant	33

2.2.8.1.2	Enchère premier-prix offre-cachée.....	34
2.2.8.1.3	Enchère hollandaise (descendante).....	34
2.2.8.1.4	Enchère Vickery (deuxième-prix offre-cachée).....	34
2.2.8.1.5	Enchères tous-payent.....	34
2.2.8.1.6	Problèmes avec les protocoles d'enchère.....	34
2.2.9	Négociation basée sur les heuristiques.....	35
2.2.10	Négociation basée sur l'argumentation.....	35
2.2.11	Négociation basé sur les transactions.....	36
Chapitre 3 Allocation de ressource sous information incomplète.....		37
3.1	Protocole de négociation.....	40
3.1.1	La première phase.....	40
3.1.2	Deuxième phase : 'Fonction d'utilité proposée'.....	41
3.2	Génération de problème.....	43
3.3	Les résultats obtenus.....	52
Conclusion.....		54
Perspectives.....		54
Bibliographie.....		56

Liste des figures

FIG1.1 : Transaction simple (Sandholm, 1998).....	14
FIG 2.1 : Protocole de négociation compétitifs et coopératifs.....	28
FIG 2.2 : Protocole de négociation unidirectionnels et bidirectionnels.....	29
FIG 2.3 Le protocole Contract Net.....	30
FIG 2.4 : Le protocole Kasbah	31
FIG 2.5 : Négociation heuristique.....	35
FIG 3.1 : Un fichier de données généré par CATS	47
FIG.3.2-Interface graphique de la plateforme JADE.....	49
FIG.3.3- Les Containers et les plateformes (Caire, 2009).....	50
FIG.3.4-Cycle de vie d'un agent dans JADE (Bellifemine et al, 2010).....	51
FIG 3.5 Le bien être social.....	53

Liste des tableaux

Tab 2.1 La dilemme du prisonnier.....	33
Tab 3.1 : Les agents et leurs préférences.....	38
Tab 3.2 : Les préférences de l'agent 1.....	39
Tab 3.3 : Les préférences de l'agent 2.....	39
Tab 3.4 : Les utilités pour l'agent 1.....	42
Tab 3.5 : Les utilités de l'agent 2.....	42
Tab 3.6 : Bien être de Nash obtenu %	52
Tab 3.7 : Bien Influence de la taille du paquet sur le bien être social et le temps de calcul....	52

Introduction générale :

L'allocation de ressources multi agents (MARA) est un domaine de recherche qui s'intéresse à l'étude des mécanismes de distribution d'un ensemble de ressources entre un ensemble des agents.

Le problème émergent de l'allocation de ressources se situe au carrefour de deux disciplines : pour l'informatique dans le domaine des systèmes multi-agents et pour l'économie dans le domaine du choix social. D'une façon générale, il existe deux grands types de procédures d'allocation de ressources: Les procédures de type centralisé et les procédures de type décentralisé. Dans une procédure de type centralisé, les agents s'en remettent à un arbitre supposé juste et impartial dont les décisions sont incontestables. L'idée est de rendre la justice de manière centralisée selon des principes démocratiquement consentis. Contrairement à cela, dans une procédure décentralisée, une règle du jeu est établie et acceptée par les agents. Ils interagissent et prennent librement leurs décisions de façon autonome, à condition que les règles du jeu soient respectées. Dans ce type de procédure, le partage est négocié entre les agents à partir d'une allocation initiale qui évolue au travers de négociations locales successives.

Les premières études basées sur les préférences des agents qui sont partagées et les critères d'acceptabilité ne tiennent pas compte de leurs impacts sur leurs intérêts dans le future (agent myope). Avec ces hypothèses, plusieurs solutions ont été proposées et développées comme la définition des classes de fonction d'utilités et les critères d'acceptabilités des offres.

Néanmoins, face à des agents égoïstes avec des préférences divergentes sur un ensemble de ressources corrélées indivisibles sans aucune connaissance partagée sur les préférences, les solutions existantes ne parviennent pas à des solutions de compromis de bonne qualité. En effet, les approches existantes fonctionnent en information complète qui permet aux agents de chercher ensemble un accord efficace. Les études sont concentrées sur la définition des classes d'utilité qui garantissent la convergence de la négociation en utilisant les transactions simples. Dans le cas des informations incomplètes sur les préférences des agents, on ne peut pas utiliser un agent myope mais on doit concevoir un agent avec la capacité de gestion des risques en tenant compte l'incertitude liée à l'information incomplète sur les préférences des agents. Dans notre proposition, les agents ont des préférences sur des paquets de ressources et non sur des ressources simples (l'agent ne peut pas savoir l'utilité d'une ressource unique).

L'objectif de notre proposition est d'identifier des heuristiques et des mécanismes pour pallier le manque d'informations et pour obtenir une allocation de bonne qualité.

Dans notre contribution on a essayé de proposer une stratégie au problème d'allocation de ressource sous cette information incomplète.

La suite de ce manuscrit est structurée en trois chapitres. Le premier chapitre représente le domaine de l'allocation de ressources. Le deuxième chapitre aborde le domaine de système multi-agents et les différentes négociations entre eux. Dans le troisième chapitre, on va présenter notre contribution qui aborde l'allocation de ressources entre les agents pour la redistribution des ressources indivisibles sans compensation monétaire.

Chapitre 1

Allocation de ressources

Ce chapitre représente tout sur le domaine de l'allocation de ressources. On définit c'est quoi l'allocation de ressources avec tout les éléments associé à ce domaine (les ressources, les préférences des différents agents, les utilités, les procédures d'allocations, les transactions..).

1.1 Introduction

L'allocation de ressources au sein d'un système d'agents, qui ont des préférences sur un ensemble de ressources, est un domaine de recherche passionnant à la croisée de l'informatique et de l'économie.

L'affectation des ressources est un sujet d'importance élevé en informatique et en économie. Pour confirmer le fait que les ressources sont réparties entre plusieurs agents et que ces agents font l'effet sur le choix de l'allocation, le terme appelé allocation de ressources multi-agents (MARA). Les informaticiens étudient comment trouver une allocation, Alors que les économistes concentrent sur les qualitatives qu'est-ce qui fait une bonne allocation.

1.2 L'allocation de ressources : définition

L'allocation de ressources est composée par trois compositions principales $\{R\}, \{A\}, \{U\}$; Alors A c'est l'ensemble d'agents, R représente l'ensemble de ressources, U centré autour de préférences des agents sur les ressources existants. Le but est l'identification d'une distribution de R sur A satisfaisant un objectif, selon les préférences U des agents. (Nongaillard, 2009).

1.3 Les ressources

Une ressource c'est un composant principal dans la négociation dans les problèmes d'allocation de ressources ; donc les ressources se sont des éléments qui sont distribués pour une allocation définit.

1.3.1 Continues vs Discrètes

La ressource comporte de deux types ; peut être continue ou bien discrète. Le type de ressource mentionné a la propriété d'intervenir dans le processus de négociation pour les échanges des ressources entre les agents. On peut distinguer les deux types à travers que la ressource continue comme gaz naturel ce dernier accepte la fragmentation, par exemple, un gouvernement veut distribuer également le gaz naturel entre les différentes villes (Cormas, 2001) (Nongaillard, 2009), ou bien autres pays ! , mais la ressource discrète est indivisible, par exemple, un sac à bandoulière faut l'acheter comme une entité entière et complet. Pour la ressource continue peut transformée en ressources discrètes.

1.3.2 Divisible ou non

Ici ce n'est pas le même cas que la ressource soit discrète ou bien continue se sont des propriétés Lieutenant de la ressource donc ce qui n'est pas pour la propriété de la ressource divisible ou indivisible. Cette propriété influe sur les mécanismes d'allocation. On concentre uniquement sur les ressources non indivisibles.

1.3.3 Partageable ou non

Plusieurs agents utilisent une ressource en même temps donc c'est une ressource partageable , par exemple, l'utilisation de satellite ou les images sont partagés aux plusieurs agents (Lemaître et al, 2001) (Bouveret, 2005). Donc notre cas étudie les ressources non partageables seulement.

1.3.4 Dynamiques ou statiques

La ressource soit dynamique soit statique ; quand son état reste inchangé donc on dit la ressource est statique. La ressource dynamique c'est la ressource ou son état changé avec le temps, par exemple, la nourriture ; un agent peut consommer avec le temps , il est changé donc dynamique. On étudie sauf les ressources statiques.

1.3.5 Mono-Item ou Multi-Item

Un environnement multi-item où chaque type de ressource a un ou plusieurs exemplaires mais un environnement mono- item où chaque type de ressource ne représente qu'une seule ressource. On peut dit qu'un environnement multi- item peut être transformé en un environnement mono-item en attribuant des nouveaux noms ou numéros qui permettent de différencier les ressources identiques. En général notre cas pour étudie seulement les environnements mono-item.

1.3.6 Ressources ou tâches

Toutefois, il faut prendre en compte que les tâches sont comme des ressources ayant une utilité négative vis-à-vis de l'agent (Estivie, 2006) ; donc l'allocation de ressources et l'allocation de tâche se sont convergé dans en le sens et similaire , alors on dit que la différence entre eux c'est que l'allocation de ressources est indépendante tandis que

l'allocation de tâches est dépendante où les tâches sont liés , nécessite le début d'exécution de processus de la tâche B quand fini l'exécution de processus de la tâche A.

Notre cas on étudie seulement les ressources.

D'après les propriétés de ressource mentionnées dans notre cas d'étude, alors les ressources sont discrètes, indivisibles, non partageables et statiques dans un environnement mono-item ; donc on dit Chaque ressource peut être allouée à un et un seul agent.

1.4 Représentation de préférences

La présentation de ressources c'est nécessaire pou l'allocation de ressources ainsi que les préférences sont très importants pour un agent pour exprimer leur conviction pour les ressources existantes dans un problème d'allocation de ressources.

1.4.1 Définition de préférence

Les préférences expriment la satisfaction relative ou absolue d'un individu face à un choix entre différentes alternatives.

Dans le contexte de MARA, ces alternatives sont les différentes allocations potentielles de ressources, ou plus concrètement, le groupe de ressources reçu par un agent pour chacune des allocations alternatives. Une structure de préférences représente les préférences d'un agent par rapport à un ensemble d'alternatives X.

1.4.2 Structures de préférences

Y a quatre familles de structures de préférences :

- La structure des préférences cardinales : généralement appelée utilité, elle consiste en une fonction d'évaluation ;

$$u: X \rightarrow \text{Val}$$

Où Val est soit un ensemble de valeurs numériques, soit une échelle totalement ordonnée de valeurs qualitatives ; la structure de préférences est appelée quantitative si non c'est qualitative.

- la structure des préférences ordinales : elle consiste en une relation binaire sur les alternatives, désignée par, réflexive et transitive
- la structure de préférence binaire : est une partition de X en un ensemble de bons et de mauvais états.

- la structure des préférences floues : est une relation floue sur X

$$X \rightarrow [0, 1]$$

Les préférences floues sont plus générales que les préférences ordinales et cardinales.

Par contre, en économie, les structures ordinales sont les plus utilisées (Mas-Colell et al, 1995), puisque les économistes s'intéressent dans la plupart du temps à des applications purement économiques ne considèrent que des agents humains (en informatique, les agents ne sont pas des humains dans la majorité des applications (Nongaillard, 2009).

Il existe plusieurs types de fonction d'utilité, parmi ces types on cite :

- Fonction non-négatives : pour modéliser les situations dont lesquelles on ne peut pas attribuer des utilités négatives.
- Fonctions monotones : dans ce genre de fonctions, on ne peut pas attribuer à un lot de ressources une valeur qui est inférieure à une valeur attribuée à l'un des sous-lots qui le compose.
- Fonction binaires : En utilisant ces fonctions, un agent peut dire seulement s'il veut ou non posséder une ressource.
- Fonctions additives : ces fonctions sont utilisées pour modéliser les cas où il n'existe pas de synergie (interaction) entre les ressources. Dans ce cas, l'utilité d'un lot de ressources est la somme des utilités des ressources qui composent ce lot.

1.4.3 Préférences quantitatives

Soit $R = \{r_1, \dots, r_m\}$ soit un ensemble de ressources indivisibles. Une structure de préférence quantitative pour un problème d'allocation de ressources est une fonction d'utilité.

On définit les utilitaires sur des bundles, nous supposons que les préférences des agents sont exemptes d'externalités d'allocation. C'est-à-dire que la valeur qu'un agent attribue à un groupe R ne dépend pas de l'allocation des ressources restantes entre les autres agents.

1.4.3.1 Forme explicite « Bundle forme »

Elle construit par définir des certains nombres pour chaque paquet(bundle) , définir un ensemble des paires $\wp, u_i(\wp)$ et on élimine tous les bundles tel que $u_i \wp = 0$. On peut définir et décrite chaque fonction d'utilité par la forme explicite parce qu'elle très utile et significatif

Exemple : R un ensemble de ressources donc on donne $R=\{r1,r2,r3,r4\}$, construit sous un forme explicite la fonction d'utilité d'un agent $i \in A$ avec les sous ensembles de ressources :

$$\begin{aligned}
 u_i(\{r1\}) &= 3 \\
 u_i(\{r2\}) &= 2 \\
 u_i(\{r3\}) &= 1 \\
 u_i(\{r4\}) &= 2 \\
 u_i(\{r1, r2\}) &= 5 \\
 u_i(\{r1, r3\}) &= 4 \\
 u_i(\{r1, r4\}) &= 3 \\
 u_i(\{r2, r3\}) &= 2 \\
 u_i(\{r2, r4\}) &= 4 \\
 u_i(\{r3, r4\}) &= 3 \\
 u_i(\{r1, r2, r3, r4\}) &= 5
 \end{aligned}$$

Alors la remarque c'est qu'il manque le nombre exponentiel malgré l'expression de l'exemple est expressive et aussi le calcul de nombre exponentiel est très élevée, c'est-à-dire pour déterminer la solution optimale on doit considérer explicitement tous les allocations possibles (Nongaillard, 2009).

1.4.3.2 Forme k-additive

Pour certains utilitaires, il est possible d'exploiter des régularités dans la structure de la fonction afin de construire des descriptions. On donne $k \in \mathbb{N}$, une fonction d'utilité u est dite k-additive si et seulement s'il existe un coefficient α_T pour chaque ensemble de ressources T de taille au plus k telle que :

$$u(R) = \sum_{T \subseteq R} \alpha_T$$

Le coefficient α_T représente la valeur synergique (le degré d'interaction) de la possession de tous les éléments de T ensemble, au-delà de l'utilité associée à l'un de ses sous-ensembles appropriés. Si une fonction d'utilité est présentée en termes de tels coefficients, on dit qu'elle est donnée sous forme k-additive.

La forme k-additive est également pleinement expressive, mais seulement dans le sens où elle peut décrire toute fonction d'utilité à condition que k soit choisi assez grand (pour tout k inférieur au nombre total de ressources, il existe des fonctions qui ne peuvent pas être représentées). Il est généralement beaucoup plus succinct que la simple forme explicite. La forme k-additive de représentation des fonctions d'utilité s'inspire des travaux de la théorie de la mesure floue. Il a été introduit dans le domaine MARA (Chevaleyre et al, 2006).

1.5 Bien-être social

L'objectif de MARA est de trouver une allocation optimale par rapport à une métrique qui dépend, d'une manière ou d'une autre, des préférences des agents individuels dans le système. L'agrégation des préférences individuelles peut souvent être modélisée en utilisant la notion de bien-être social telle qu'elle a été étudiée dans la théorie de l'économie du bien-être et du choix social. Ce point de vue est conforme à la métaphore largement utilisée des systèmes multi-agents en tant que «sociétés d'agents».

En supposant que les agents individuels modélisent leurs préférences en utilisant des ensembles de ressources de fonctions d'utilité en valeurs numériques, le concept de protection sociale utilitaire, défini comme la somme d'utilitaires individuels, peut être utilisé pour mesurer la qualité d'une allocation du point de vue du système dans son ensemble .

1.5.1 L'Optimalité de Pareto

Une allocation P est dominée par Pareto par une autre allocation Q si et seulement si les choses suivantes :

- $P \leq_i Q$ pour tous les agents $i \in A$
Et
- $P <_i Q$ pour au moins un agent $i \in A$.

Une allocation est optimale selon Pareto si et seulement si elle n'est dominée par aucune autre allocation. C'est-à-dire qu'une allocation est optimale selon Pareto si et seulement si il n'est pas possible d'améliorer le bien-être individuel d'un agent sans aggraver la situation des autres.

L'optimalité de Pareto est généralement considérée comme le critère le plus fondamental de l'efficacité. Notez que le concept d'optimalité de Pareto est purement ordinal: il ne nécessite pas que les préférences soient numériques, pas même comparables sur le plan interpersonnel. Notez également que la notion de dominance de Pareto ne donne lieu qu'à un ordre partiel (et non complet) par rapport aux allocations alternatives.

1.5.2 Bien être social utilitaire

Le bien-être social utilitaire est défini comme la somme des services publics individuels comme suit :

$$sw_u (P) = \sum_{i \in A} u_i (P)$$

Le bien être utilitaire est utilisé pour fournir une mesure appropriée pour le bénéfice global (ainsi que moyen) dans une gamme d'applications de commerce électronique.

Il est indépendant des zéros des utilitaires individuels.

1.5.3 Bien être social égalitaire

Le bien-être social égalitaire est donné par l'utilité de l'agent qui est actuellement la somme suivante :

$$sw_e (P) = \min \{u_i (P) \mid i \in A\}$$

Le bien être social égalitaire offre un niveau d'équité et peut constituer un indicateur de performance approprié lorsque nous devons satisfaire les besoins minimaux d'un grand nombre de clients.

1.5.4 Produit de Nash

Le produit Nash est défini comme le produit de l'individu utilitaires comme suit:

$$sw_v (P) = \prod_{i \in A} u_i (P)$$

Cette notion de bien-être social favorise à la fois l'augmentation de l'utilité globale et les redistributions visant à réduire les inégalités. En ce sens, il peut être considéré comme un bon compromis entre les agendas utilitariste et égalitaire.

Le produit Nash ne peut fournir une mesure significative du bien-être social que si tous les services publics individuels ne sont pas négatifs (ou mieux, s'ils sont tous positifs)

1.5.5 Bien être social élitiste

Le bien-être social élitiste est donné par l'utilité de l'agent c'est actuellement le mieux comme suit:

$$sw_e^l (P) = \max \{u_i (P) \mid i \in A \}$$

Le bien être social élitiste n'est clairement pas une mesure juste pour le bien-être social, mais il peut être utile dans les applications basées sur la coopération dans lesquelles un seul agent est nécessaire pour atteindre ses objectifs.

1.5.6 L'absence d'envie

Une allocation est exempte d'envie si et seulement si chaque agent est au moins aussi satisfait de sa part que de l'un quelconque des ensembles attribués à l'un des autres agents. Autrement dit, une allocation P est sans envie si et seulement si $P(j) \leq_i P(i)$ valable pour tous les agents i et j . L'absence d'envie est une propriété qui n'exige pas la comparabilité des utilitaires de différents agents.

Si on exige que tous les éléments soient alloués, alors une allocation sans envie n'existe pas toujours (considérons, par exemple, un problème d'allocation avec une ressource unique souhaitée par tous les agents du système). Mais même lorsque tous les éléments ne doivent pas être alloués, il est bien connu qu'il existe des problèmes d'allocation pour lesquels il n'existe aucune allocation à la fois optimale de Pareto et sans envie. On pourrait donc chercher à trouver des allocations (Pareto optimales) qui minimiseraient au moins autant que possible le «degré d'envie» global.

Non garantie d'existence de ces allocations sans argent, mais garantie dans notre cadre de travail avec argent (Alkan, 91)

Il existe plusieurs définitions de candidats pour une envie minimale. Deux approches possibles seraient les suivantes:

- Réduisez le nombre d'agents envieux.
- Minimiser le degré moyen d'envie (la distance à parcourir) le concurrent le plus envié) de tous les agents envieux

1.6 Procédures d'allocation

D'une manière générale, la procédure d'allocation utilisée pour trouver une allocation appropriée de ressources pourrait être centralisée ou distribuée. Dans le cas centralisé, une seule entité décide de l'affectation finale des ressources entre les agents, éventuellement après avoir obtenu les préférences des autres agents du système. Des exemples typiques d'approche centralisée sont les enchères combinatoires. Ici, l'entité centrale est le commissaire-priseur et la déclaration des préférences prend la forme d'enchères. En revanche, dans les approches

réellement distribuées, les allocations apparaissent à la suite d'une séquence d'étapes de négociation locales. De telles négociations locales sont souvent limitées aux accords bilatéraux, mais des systèmes permettant des échanges multilatéraux de ressources entre plus de deux agents sont également possibles.

Une enquête complète sur les procédures d'allocation de ressources pour le MARA irait au-delà de la portée du présent document. Toute enquête de ce type devrait traiter au moins les trois problèmes suivants:

- Protocoles: à ce niveau, nous devons aborder les problèmes ontologiques (quels types d'opérations sont possibles?) Et élaborer un protocole de communication en conséquence (quels messages les agents doivent-ils échanger pour s'entendre sur un accord de ce type?).
- Stratégies: lors de la conception d'agents individuels, nous devons concevoir des stratégies qui leur permettent d'exploiter au mieux un protocole de négociation donné. Cela peut également fournir des informations en retour au premier niveau: Dans la mesure du possible, les protocoles doivent être conçus de manière à inciter les agents de négociation à adopter un profil de comportement souhaitable (conception du mécanisme).
- Algorithmes: à ce niveau, nous devons fournir des algorithmes permettant de résoudre les problèmes de calcul rencontrés par les agents lors de la négociation. Cela inclut à la fois des algorithmes pour décider comment répondre à une proposition dans un scénario de négociation distribuée et des algorithmes de détermination des gagnants pour les enchères combinatoires. Encore une fois, ce niveau peut fournir des informations en retour aux deux autres niveaux: Si un problème de calcul particulier s'avère trop difficile à résoudre de manière raisonnable.

Cela peut nécessiter une simplification du protocole de négociation (ou de la stratégie).

1.6.1 Centralisé vs distribué

L'approche centralisée et l'approche distribuée du MARA ont leurs avantages et leurs inconvénients. L'argument le plus important en faveur des mécanismes basés sur des enchères concerne peut-être la simplicité des protocoles de communication nécessaires à la mise en œuvre de tels mécanismes. Une autre raison de la popularité des mécanismes centralisés est la poussée récente dans la conception d'algorithmes puissants pour les enchères combinatoires qui, pour la première fois, fonctionnent assez bien dans la pratique. Bien entendu, de telles techniques sont, en principe, également applicables dans le cas distribué, mais les recherches

dans ce domaine n'ont pas encore atteint le même niveau de maturité que pour les enchères combinatoires.

Le modèle distribué semble également plus naturel dans les cas où il est impossible (sur le plan informatique) de trouver des allocations optimales, mais même de petites améliorations par rapport à l'allocation initiale des ressources seraient considérées comme un succès. Les améliorations progressives par rapport au statu quo sont naturellement modélisées dans un cadre de négociation réparti.

1.7 Les transactions

1.7.1 Définition

Une transaction est une paire $d = (A, A')$ où A et A' sont des allocations de ressources avec $A \neq A'$. L'interprétation voulue de cette définition est que l'opération $= (A; A')$ ne s'applique que dans la situation A et aboutira à la situation A' . Il spécifie pour chaque ressource du système si elle doit rester où elle est ou où elle doit être déplacée, respectivement nos agents sont intéressés dans le sens où ils ne proposeront ou n'accepteront que des offres qui augmentent strictement leur propre bien-être.

1.7.2 Types de transaction

1.7.2.1 Les transactions simples

Les transactions les plus simples sont des transactions une ressource à la fois, dans le cadre desquelles une ressource unique est transmise d'un agent à un autre. Cela correspond à la forme "classique" d'un contrat que l'on trouve généralement dans le protocole de contrat (Smith, 1980).

De toute évidence, si l'agent qui se sépare de la ressource en question lui attribue une valeur quelconque, un tel accord «une ressource unique» ne peut être conclu que si les deux agents peuvent s'accorder sur un prix approprié.

Définition (O-Contract): Un O-Contract est défini par le couple $\{r_{ij}, p_{ij}\}$ où r_{ij} est la ressource que l'agent i va donner à l'agent j , et p_{ij} est la somme que j va donner à i en échange. (Estivie, 2006).

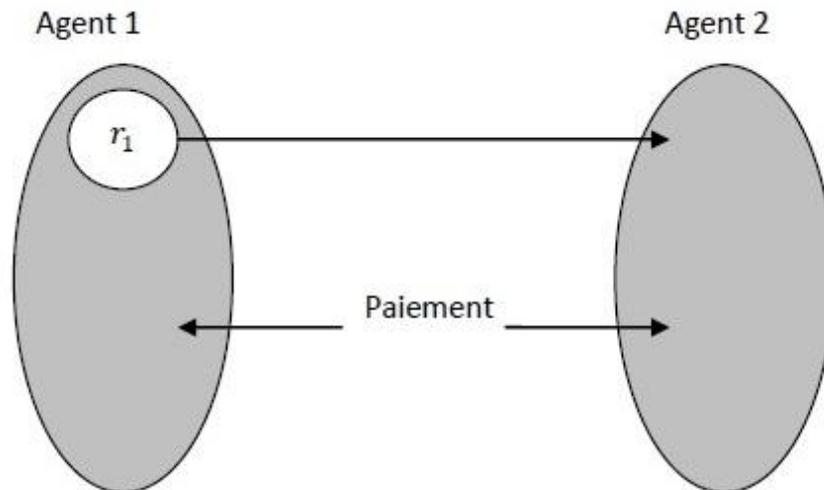


FIG1.1 : Transaction simple (Sandholm, 1998)

1.7.2.2 Les transactions groupées

Les transactions dans lesquelles un agent transmet un ensemble de ressources à un autre agent sont appelées **Cluster contracts**. Dans l'exemple ci-dessus, passer $\{r_1, r_2\}$ de l'agent 1 à l'agent 2 serait individuellement rationnel si l'agent 2 versait à l'agent 1 un montant de, disons, r_2 . Ce problème a été étudié depuis longtemps dans des problèmes d'optimisation centralisée (Waters, 1987). Dans le domaine des négociations automatiques, Sandholm et Lesser sont parmi les premiers chercheurs qui se sont intéressés au problème des transaction groupées (Sandholm, 1993), (Sandholm et Lesser, 1997).

1.7.2.3 Les transactions échange

Les transactions dans lesquelles un agent donne un seul élément à un autre agent qui retourne un autre élément sont appelées **Swap deals** (swap contracts ou S-contracts). Un échange peut être avantageux pour les deux agents impliqués, même si aucun argent n'est échangé et même si les deux agents attribuent une valeur à l'une ou l'autre des ressources.

Définition 2.13 (S-Contract): Un S-Contract est défini par le 4-tuple $\{r_{ij}, r_{ji}, \rho_{ij}, \rho_{ji}\}$ où

- ❖ r_{ij} , est la ressource que l'agent i va donner à l'agent j
- ❖ r_{ji} , est la ressource que l'agent j va donner à l'agent i
- ❖ ρ_{ij} est la somme que j va donner à i
- ❖ ρ_{ji} est la somme que i va donner à j .

1.7.2.4 Les transactions multi-agents

Parfois, il peut également être nécessaire d'échanger des ressources entre plus de deux agents. Une transaction multi-agents est une transaction pouvant impliquer un nombre quelconque d'agents, chaque agent transmettant au plus une ressource à chacun des autres agents participants. De manière similaire à l'exemple ci-dessus, nous pouvons également construire des scénarios dans lesquels des accords de swap ou des contrats multi-agents sont nécessaires (c'est-à-dire où les contrats en groupe ne seraient pas suffisants pour garantir un bien-être social maximal).

1.7.2.5 Les transactions combinées

D'après (Sandholm, 1998), les transactions combinant les fonctionnalités du cluster et le type d'opérations multi-agents sont appelées offres combinées. Celles-ci peuvent impliquer un nombre quelconque d'agents et de ressources.

En d'autres termes, chaque transaction d est une transaction combinée. L'ontologie des types de transaction présentée ici n'est bien entendu pas exhaustive. Il peut, par exemple, Il serait également intéressant d'envisager la catégorie d'opérations qui implique exactement deux agents mais n'importe quel nombre de ressources.

Chapitre 2

La négociation dans les systèmes multi agent

Dans ce chapitre on représente la définition de l'agent et les systèmes multi agents avec la négociation et ses types et les protocoles de négociation et les domaines d'application des systèmes multi agents, et ses protocoles et aussi la négociation qui est basée sur heuristique, argumentation et les transactions.

2.1 Agents et Systèmes multi-agents

2.1.1 Les agents

Actuellement, il n'existe pas, une seule définition pour le concept d'agent. La plupart des travaux font référence à la définition fournie par (J Ferber ,2000) qui stipule qu'un agent est une entité physique ou virtuelle :

- qui est capable d'agir dans un environnement,
- qui peut communiquer directement avec d'autres agents,
- qui est mu par un ensemble de tendance (sous la forme d'objectifs individuels ou d'une fonction de satisfaction, voire de survie, qu'elle cherche à optimiser),
- qui possède des ressources propres,
- qui est capable de percevoir (mais de manière limitée) son environnement,
- qui ne dispose que d'une représentation partielle de cet environnement (et éventuellement aucune),
- qui possède des compétences et offre des services,
- qui peut éventuellement se "reproduire",
- qui a un comportement,
- qui tend à satisfaire ses objectifs, en tenant compte des ressources et des compétences dont elle dispose, et en fonction de sa perception, de ses représentations et des communications qu'elle reçoit.

2.1.2 Les modèles d'agents

On présente les principaux modèles d'agents :

2.1.2.1 Modèles d'agents réactifs

La structure des agents purement réactifs tend à la simplicité. Ce sont des agents qui fonctionnent selon un mode stimuli/réponse. Dès qu'ils perçoivent une modification de leur environnement, ils répondent par une action programmée. L'agent réactif ne possède pas une représentation complète de son environnement et n'est pas capable de tenir compte de ses actions passées.

De ce fait, il ne peut avoir des capacités d'apprentissage. Ainsi, les agents réactifs sont souvent considérés comme n'étant pas "intelligents" par eux mêmes. Chaque individu pris séparément possède une représentation faible de l'environnement et n'a pas de buts. Mais ces derniers peuvent être capables d'actions évoluées et coordonnées

2.1.2.2 Modèles d'agents cognitifs

Les agents cognitifs quant à eux possèdent une mémoire et sont capables de représentations symboliques. Ils possèdent une connaissance propre qui comprend une représentation (partielle) de leur environnement, des autres agents, ainsi que de leur savoir-faire, ce qui leur permet de gérer leurs interactions avec l'environnement et les autres agents. On parle aussi agents d'intentionnels, délibératifs ou rationnels car ils possèdent des représentations explicites de leurs buts sur lesquelles ils sont capables de raisonner afin de produire des plans d'actions.

2.1.2.3 Modèles d'agents hybrides

Ce sont des agents ayant à la fois des capacités cognitives et réactives. Ils conjuguent la rapidité de réponse des agents réactifs ainsi que les capacités de raisonnement des agents cognitifs. Ce modèle a été proposé par plusieurs auteurs afin de palier aux problèmes liés au temps de décision et au temps d'action relativement élevé pour les agents réactifs.

2.1.2 Environnement

L'environnement désigne le monde réel ou bien l'univers virtuel dans lequel les agents évoluent. L'environnement d'un agent est donc constitué de tout ce qui l'entoure : les objets et/ou les autres agents. La description d'un environnement dépend fortement de la nature d'un problème. On peut distinguer pour un environnement les propriétés suivantes :

- ✓ *accessible versus inaccessible*: un environnement est dit accessible si les agents ont accès à l'intégralité de son état. Inaccessible dans le cas contraire.
- ✓ *déterministe versus indéterministe*: un environnement est déterministe si un changement de son état est déterminé uniquement par son état courant et les actions des agents. Un environnement indéterministe pourra produire un résultat différent pour une même action.
- ✓ *statique versus dynamique*: un environnement est dit statique si il ne change d'état que sous l'effet des actions des agents. Au contraire, un environnement dynamique possède ses propres processus d'évolution qui peuvent modifier son état sans intervention des agents.
- ✓ *discret versus continu*: on parle d'environnement discret lorsque le nombre de perceptions et d'actions possibles est limité. Si ce n'est pas le cas, l'environnement est alors continu.

2.1.3 L'interaction

Le simple fait de placer un ensemble d'agents dans un même environnement n'est pas suffisant pour définir un système multi-agent (SMA). Les différents agents doivent être en mesure d'interagir et de se comprendre mutuellement afin de pouvoir se coordonner et éventuellement coopérer. L'étude des mécanismes d'interaction est donc primordiale dans la conception d'un SMA.

(J Ferber, 1995) donne la définition suivante de l'interaction: «Pour un agent, interagir avec un autre constitue à la fois la source de sa puissance et l'origine de ses problèmes. C'est en effet parce qu'ils coopèrent que des agents peuvent accomplir plus que la somme de leurs actions, mais c'est aussi à cause de leur multitude qu'ils doivent coordonner leurs actions et résoudre des conflits».

On distingue différents types d'interaction que les agents peuvent adopter comme :

La communication, la coopération, la négociation et la coordination.

2.1.3.1 La communication

La communication est un des éléments importants du système multi-agents. Elle permet l'échange des informations, la coopération et la coordination. Dans la communication agent, l'intention de communiquer est de produire un effet sur les destinataires: ils exécuteront probablement une action demandée par l'émetteur ou répondront à une question.

2.1.3.2 La coopération

La coopération se traduit par le fait qu'un ensemble d'agents travaillent ensemble pour satisfaire un but commun ou individuel. L'ajout ou la suppression d'un agent influe considérablement sur la performance du groupe. Le besoin de faire coopérer des agents, vient essentiellement du fait qu'un agent ne peut atteindre son objectif individuellement et a, par conséquent besoin de l'aide des autres agents du système.

2.1.3.3 La coordination

La coordination est présente lorsqu'il existe une interdépendance dans les actions des agents, leurs buts ou même les ressources qu'ils utilisent. A ce stade, coordonner les activités des agents devient un aspect essentiel afin d'éviter des problèmes dans le comportement global du système. En effet, la coordination met de l'ordre dans le processus global effectué par des agents. Elle consiste à synchroniser leurs activités ou à régler les conflits qui existent entre eux.

2.1.3.4 La négociation

Un système multi-agents est composé de plusieurs agents. Ces derniers peuvent entrer en conflit pour plusieurs raisons: conflits d'intérêts ou de buts, accès à des ressources ou proposition de plusieurs solutions différentes à un seul problème. Afin de résoudre ces conflits et de trouver une situation qui satisfasse tout le monde, les agents négocient entre eux en faisant des concessions ou en cherchant des alternatives.

2.1.4 L'organisation

L'organisation permet de structurer les différentes composantes du système ainsi que leur mode de fonctionnement global. L'organisation s'intéresse aussi aux dimensions sociales des agents. Elle peut être considérée comme une structure décrivant les interactions et autres relations qui existent entre les membres de la dite organisation (dans le but d'assouvir un objectif commun). L'organisation peut donc apparaître comme une structure de coordination et de communication.

Quand on parle d'organisation au niveau des systèmes multi-agents, différents aspects sont étudiés :

2.1.4.1 Les topologies de l'organisation

Les topologies décrivent les structures organisationnelles en fonction des besoins du système. On retrouve dans la littérature différents types d'organisation. L'organisation des agents peut être :

- ✓ les topologies à structure hiérarchique: ces organisations ont un mode d'adaptation fixe, le contrôle est centralisé sur un agent qui communique les ordres aux autres agents (principe maître/esclave). Les autres agents se contentent d'exécuter l'ordre reçu,
- ✓ les topologies de type marché: ces organisations sont composées d'agents coordinateurs et d'agents d'exécutants. L'allocation de tâches est régie sous la forme d'un appel d'offres.
- ✓ les topologies de type communauté: les agents dans ce type d'organisation possèdent les mêmes capacités et le contrôle est fortement distribué (ils communiquent les résultats des problèmes aux autres membres de la communauté),

- ✓ les topologies de type société: le contrôle est décentralisé dans ce type d'organisation, les agents poursuivent chacun un objectif opérationnel et le comportement global s'ajuste par des principes de négociations.

2.1.4.2 L'émergence

Un système multi-agents est un ensemble d'agents situés dans un même environnement et qui interagissent. De ces interactions peuvent émerger différents phénomènes. L'émergence traite donc de l'apparition soudaine non programmée et irréversible de phénomènes dans un système.

2.1.4.3 L'auto-organisation

L'émergence et l'auto-organisation sont décrites conjointement car elles décrivent les changements caractéristiques du système. En réalité, l'auto-organisation est une technique d'émergence qui consiste à changer les interactions et les places dans l'organisation des entités, afin de s'adapter globalement à l'environnement (et aux éventuelles perturbations de l'environnement). Il en résulte une évolution dynamique en fonction du contexte permettant de satisfaire au mieux les objectifs du système. Pour aboutir à ce phénomène, il faut agir sur les agents afin d'adapter le comportement global. L'auto-organisation est donc une technique complexe pour la résolution de problèmes.

2.1.2 Système multi-agents

2.1.2.1 Définition système multi-agents

L'agent est le composant principal des systèmes multi-agents. Un système multi-agents est un système composé des éléments suivants (J Ferber, 1995) :

- ✓ Un environnement E , c'est-à-dire un espace disposant généralement d'une métrique.
- ✓ Un ensemble d'objets O . Ces objets sont situés, c'est-à-dire que, pour tout objet, il est possible, à un moment donné, d'associer une position dans E . Ces objets sont passifs, c'est-à-dire qu'ils peuvent être perçus, créés, détruits et modifiés par les agents.
- ✓ Un ensemble A d'agents, qui sont des objets particuliers, lesquels représentent les entités actives du système.
- ✓ Un ensemble de relations R qui unissent des objets (et donc des agents) entre eux.
- ✓ Un ensemble d'opérations Op permettant aux agents de A de percevoir, produire, consommer, transformer et manipuler des objets de O .
- ✓ Des opérateurs chargés de représenter l'application de ces opérations et la réaction du

monde à cette tentative de modification, que l'on appellera les lois de l'univers.

2.1.2.2 Méthodologies de conception des systèmes multi-agents

De nombreuses méthodologies ont été proposées pour le développement des systèmes multi-agents, on peut les classer en trois catégories :

2.1.2.2.1 Les méthodologies utilisant UML et constituant une extension des méthodes orientées objet

2.1.2.2.1.1 MaSE

Multi-agent Software Engineering est une approche complète pour le développement des systèmes multi-agents de l'analyse au déploiement. L'objectif de MaSE est d'aider le concepteur à analyser, concevoir et implémenter un SMA à partir d'un cahier de charges initial.

Elle utilise de nombreux modèles graphiques pour décrire le type d'agents et les interfaces inter-agents. En effet, de nombreux diagrammes sont utilisés dans cette méthodologie. Ils sont très proches de ceux utilisés dans la conception orientée objet, avec quelques caractéristiques de plus et quelques modifications de la sémantique du paradigme objet pour pouvoir capter les concepts d'agent et les comportements coopératifs des agents.

2.1.2.2.1.2 AAI

Australian Artificial Intelligence Institute a été développée en se basant sur l'expérience accumulée dans les travaux effectués pour la gestion du trafic aérien. Elle constitue un mélange de concepts des méthodologies orientées objet avec certains concepts orientés agents. Cette méthodologie vise à construire un ensemble de modèles qui définissent des spécifications du système d'agent à base d'architecture BDI (Belief, Desires, Intentions - Croyance, Désirs, Intentions). AAI fournit deux modèles:

- ✓ Le modèle externe (le point de vue externe): il a pour but de définir les relations entre les classes d'agents, l'organisation, les rôles, les services et les responsabilités.
- ✓ Le modèle interne (le point de vue interne): il représente les comportements des agents.

2.1.2.3 Les méthodologies organisationnelles

2.1.2.3.1 Aalaadin

Est une approche organisationnelle pour le développement des SMA fondée sur le concept AGR (Agent, Groupes et Rôles). Il s'agit, d'une part, d'un cadre de développement des systèmes multi-agents, fournissant des indications méthodologiques et d'autre part, d'un environnement de prototypage et d'exécution pour des agents reposant sur les notions du modèle AGR.

Un agent est une simple entité autonome qui joue des rôles dans des groupes. Il peut avoir plusieurs rôles et être membre de plusieurs groupes. L'agent dans Aalaadin n'a aucune contrainte ou pré-requis ni sur l'architecture interne de l'agent ni sur son modèle de comportement.

Un agent est identifié dans son groupe par un rôle qui est une représentation abstraite d'une fonction ou d'un service qui définit un ensemble de contraintes que l'agent jouant ce rôle doit respecter. Un rôle peut être joué par plusieurs agents. Un groupe est un regroupement d'ensembles d'agents partageant une caractéristique. La communication entre deux agents appartenant aux groupes différents n'est pas possible.

2.1.2.3.2 Gaia

La méthodologie Gaia utilise une approche centrée sur l'organisation pour analyser et concevoir un système multi-agents. Elle est considérée comme une méthodologie générique permettant le développement de tous types de systèmes. Les concepts principaux de GAIA sont : Rôle, Permission, Responsabilité, Protocole, Propriété d'animation et Propriété de sécurité.

- ✓ Un rôle est défini par quatre attributs : responsabilités, permissions, activités et protocoles.
- ✓ Une responsabilité détermine la fonctionnalité principale liée à un rôle. Elle est décrite par des propriétés d'animation, décrivant le cycle de vie du rôle et des propriétés de sécurité, qui sont des conditions que doivent vérifier certaines variables afin de garantir la survie du rôle.
- ✓ Les permissions sont les droits de l'agent sur les ressources utilisées par un rôle. Ces droits peuvent être de type lecture, modification ou génération. Les permissions énoncent les limites de ressources dans lesquelles l'exécuteur de rôle doit opérer.

- ✓ Une activité représente une opération interne de l'agent, sans qu'il n'ait nécessairement recours à une interaction avec d'autres agents.
- ✓ Les protocoles définissent la manière dont les rôles peuvent agir avec d'autres rôles.

2.1.2.4 Les méthodologies formelles

2.1.2.4.1 DESIRE

Design and Specification of Interacting Reasoning components est directement issue de l'ingénierie des connaissances. Elle se base sur un modèle qui traite la connaissance, l'interaction, la coordination des tâches et les possibilités de raisonnement dans les SMA. DESIRE est basée sur trois concepts clés :

- ✓ la représentation de connaissances sous forme de simples graphes noeuds/liens,
- ✓ le modèle générique d'agents: un agent est considéré comme un composant décomposé en tâches,
- ✓ le modèle de composition de tâches : permet de définir les tâches et leurs relations. Les tâches sont décomposées en sous tâches et sont présentées sous forme de composants avec des informations d'entrée et de sortie.

2.1.2.4.2 PASSI

Process for Agent Societies Specification and Implementation est une méthodologie se basant sur des concepts provenant de l'ingénierie orientée objet et de l'intelligence artificielle. PASSI est une méthodologie générique pouvant s'appliquer à n'importe quel domaine. PASSI a la particularité de prendre en compte la modélisation des agents mobiles. Cette modélisation respecte les spécifications FIPA.

Dans PASSI, chaque agent est censé satisfaire ses besoins fonctionnels exprimés sous la forme de cas d'utilisation. Ces besoins sont attribués à l'agent dans la phase d'identification d'agent. Il peut jouer plusieurs rôles impliqués dans des scénarios et pouvant fournir des services à la société d'agents.

Le rôle que joue l'agent remplit au moins une tâche. Ces rôles sont aussi des médiums d'information par message. Chaque message contient la connaissance échangée et le langage utilisé pour décrire son contenu.

Le processus de PASSI est composé de cinq phases :

- ✓ La phase de définition des besoins du système qui permet d'exprimer les besoins et de décrire le contexte du système.
- ✓ La phase de définition de la société d'agents qui permet d'identifier les interactions sociales et les dépendances entre les agents.
- ✓ La phase d'implémentation qui permet de définir la structure et le comportement du SMA et ceux de chaque agent.
- ✓ La phase de codage qui permet d'identifier les modules réutilisables en cas de développement préexistants. Elle permet de rendre exécutable le code des agents.
- ✓ La phase de déploiement qui permet de spécifier la configuration du système.

2.1.3 Domaines d'application des systèmes multi-agents

Les domaines d'application des SMA sont particulièrement riches : Support et aide à la décision, commerce électronique, systèmes manufacturiers, télécommunications, supervision de réseaux, robotique, simulation de systèmes sociaux et naturels, applications embarquées et sans fil, applications Internet, filtrage et recherche d'information, gestion des transports et du trafic, ingénierie médicale, jeux, e-Learning. En effet, les SMA ont attiré l'attention à cause de leur grande souplesse d'utilisation puisqu'il n'y a aucune limite fixée sur les comportements des agents ni sur leur manière de s'exprimer mais aussi à cause de leur pouvoir de représentation d'un univers intuitif où chaque composante d'un système peut être modélisée en un agent. Il existe cependant quelques grandes catégories d'applications des SMA:

- La résolution de problèmes qui concerne toutes les situations dans lesquelles des agents logiciels, purement informatiques, accomplissent des tâches utiles aux êtres humains.
- La robotique distribuée porte sur la réalisation d'un ensemble de robots qui coopèrent pour accomplir une mission et utilise des agents concrets qui se déplacent dans un environnement réel.
- La simulation multi-agents : la simulation est la démarche scientifique qui consiste à réaliser une reproduction artificielle, appelée modèle, d'un phénomène réel que l'on désire étudier, à observer le comportement de cette reproduction lorsqu'on en fait varier certains paramètres, et à en induire ce qui se passerait dans la réalité sous l'influence de variations analogues. La simulation multi-agents offre un outil puissant

pour expliquer les changements au sein d'un système naturel ou artificiel en fonction des changements des individus qui le composent.

2.2 La négociation

2.2.1 Définition

Dans les SMA, la négociation = processus par lequel des agents résolvent leur conflits (Bussmann et Muller , 92)

La négociation est un processus de communication d'un groupe d'agents permettant d'atteindre un accord mutuellement accepté

Interactions entre agents en conflits : négociation

- agents sont en conflits, antagonismes (agents ego-centrés)
- la coordination des interactions entre les agents se basent principalement sur la négociation :
 - ✓ utilisée comme méthode de coordination et de résolution de conflits
 - ✓ utilisée pour l'allocation de tâches et les changements de plans
 - ✓ difficile à définir de façon précise

2.2.1.1 Négociation entre agents égo-centrés :

Résolution décentralisée de conflits par les agents :

- ✓ possédant une rationalité limitée
- ✓ possédant des informations incomplètes
- ✓ pouvant s'échanger de manière itérative des propositions et des contre-propositions

2.2.2 Les grandes types de négociation

2.2.2.1 Négociation compétitive :

Les agents d'intérêts différents tentent un choix de groupe sur des alternatives bien définies. (Rosenschein & Zlotkin , 94)

2.2.2.2 Négociation coopérative :

Les agents ont un but commun unique pour le système. (Smith & Davis, 81)

2.2.3 Composants du modèles de négociation

Le modèle de négociation se compose de cinq composants sont comme suit :

- langage de négociation
- protocole de négociation
- objets de négociation
- processus de décision
- La cardinalité des participants de la négociation

2.2.4 Théorie de la négociation

Elle établie des propriétés particulières pour certains domaines de négociations, pour élaborer des protocoles spécifiques :

"Nous sommes intéressés à créer des environnements sociaux pour des machines ayant un comportement bénéfique et optimal. Plutôt que de décrire un phénomène social, nous sommes faire de l'ingénierie sociale, aider les concepteurs à établir des règles de société automatisées".
(Rosenschein et Zlotkin, 94)

Elle formule 5 hypothèses :

1. maximisation des utilités
2. absence d'effets de mémoire des négociations passées
3. comparaison des utilités
4. capacités symétriques
5. absence de transfert d'utilité entre agents

2.2.5 Les formes de négociation

Selon (MH.VERRON, 1995) , les types de négociation sont:

- Les systèmes de vote
- Le take it or leave it offer
- Les négociations multi-attributs
- Les négociations multi-niveaux
- La négociation à base d'argumentation
- Les enchères.

2.2.6 Les protocoles de négociation

2.2.6.1 Définition

Chaque forme de négociation possède son propre protocole, qui définit le déroulement du processus de négociation, c'est-à-dire les actes de langage utilisés et leur séquençement.

2.2.6.2 Classification des protocoles de négociation

Elle compose de deux classes comme suit

- Première classe :

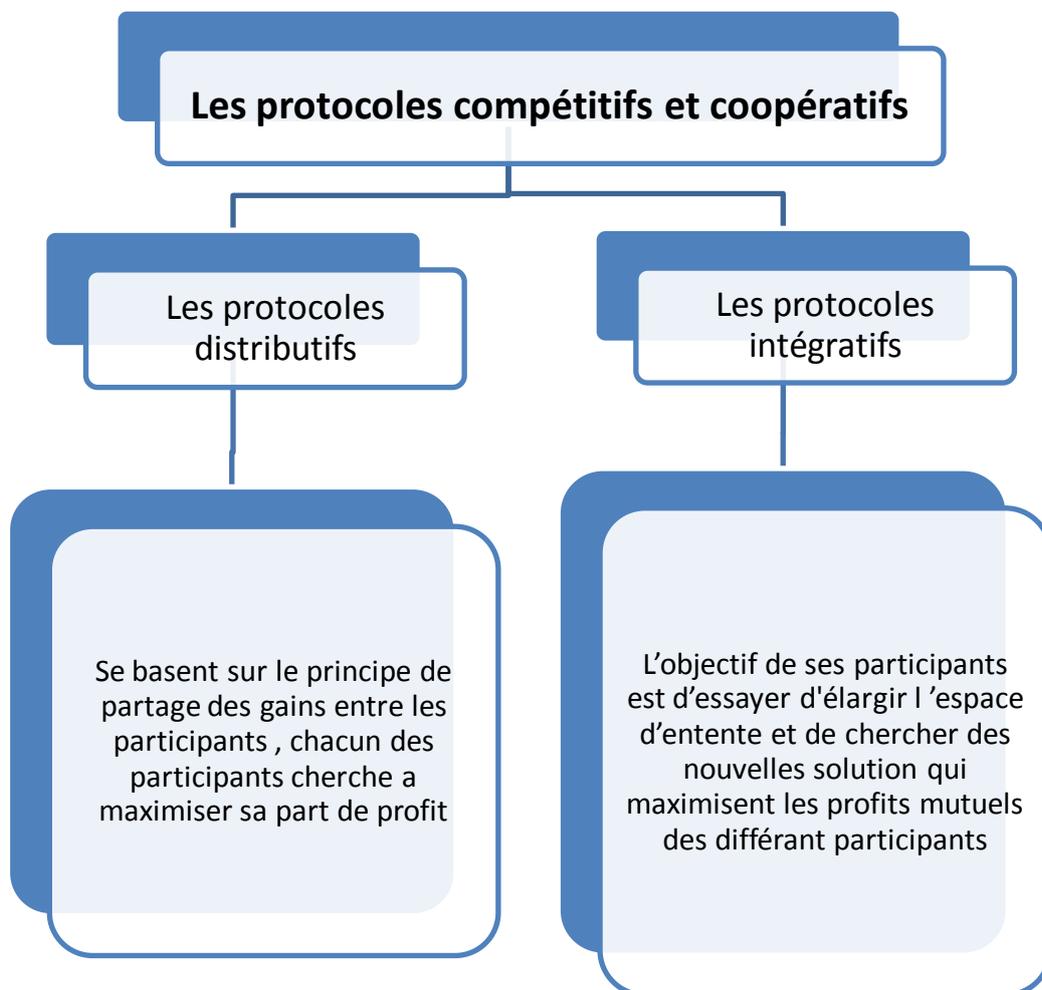


FIG 2.1 : Protocole de négociation compétitifs et coopératifs

- Deuxième classe :

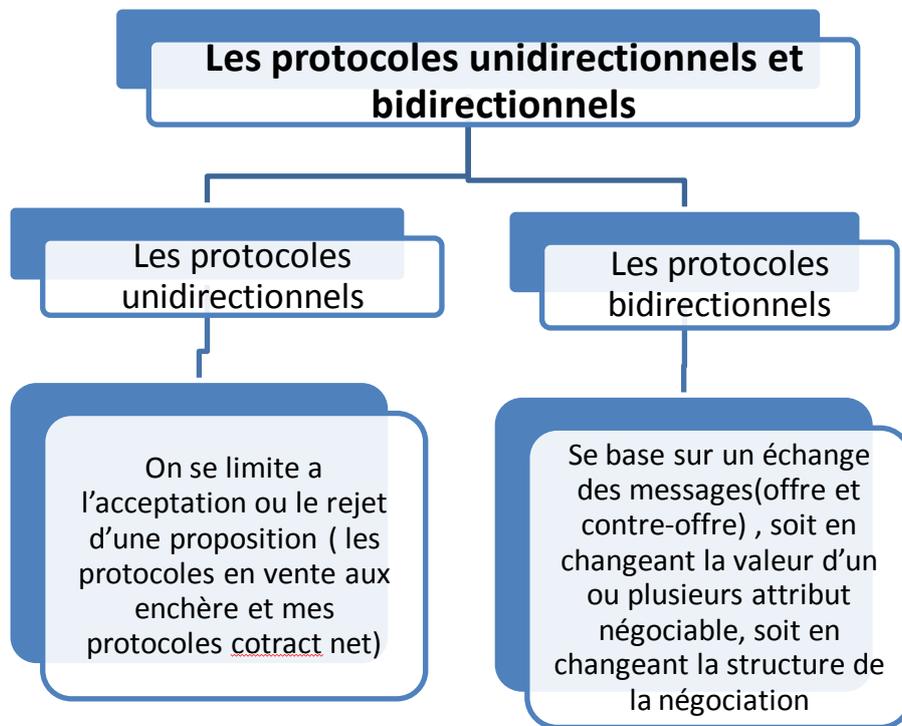


FIG 2.2 : Protocole de négociation unidirectionnels et bidirectionnels

2.2.6.3 Protocoles de négociation

2.2.6.3.1 Le protocole Contract Net

Modèle proposé par (Davis et Smith, 81,83), s'inspire de l'économie de marchés.

Principe général de la coordination par contrat : "Contract-net":

- les agents peuvent prendre 2 rôles: manager et contractant
- un agent manager décompose un problème en sous problèmes
- il annonce ensuite chaque sous-problème à un ensemble d'agents « contractants potentiels »

- les agents contractants évaluent l'annonce : ils peuvent échanger avec l'agent manager les informations lors de l'accomplissement cette évaluation
- les agents contractants qui ont les ressources et capacités nécessaires font des offres
- l'agent manager évalue les soumissions et accorde le contrat à l'agent contractant qui a fait la meilleure offre.

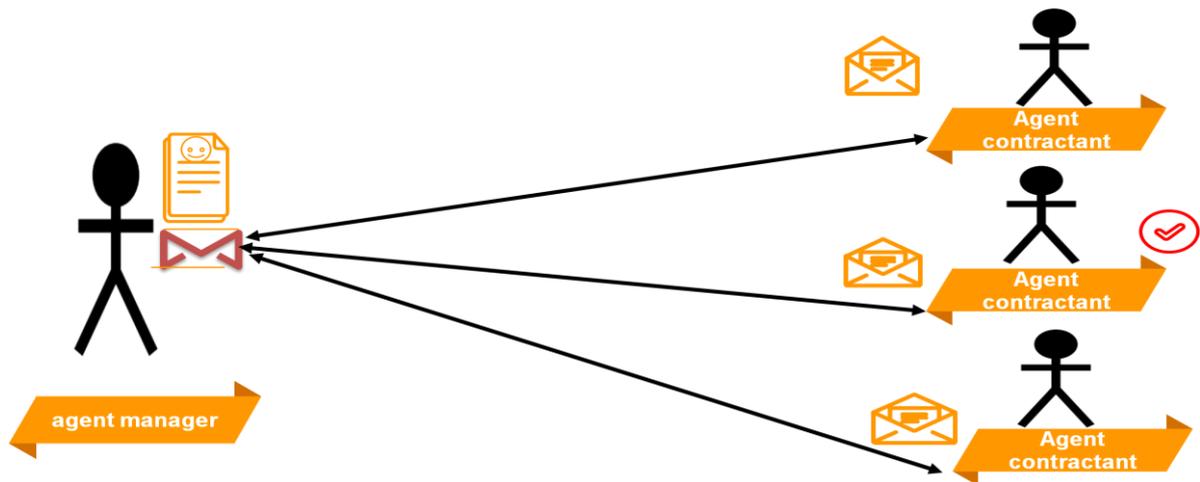


FIG 2.3 Le protocole Contract Net

2.2.6.3.2 Le protocole Contract Net étendu

Le modèle Contract Net étendu est basé sur une décomposition décentralisée de tâches. L'agent gestionnaire fractionne une tâche en plusieurs sous-tâches et les annonce à un agent ou à un groupe d'agents soumissionnaires.

Mais contrairement au Contract Net original où une tâche ne peut être qu'accordée ou rejetée, dans le Contract Net étendu, une tâche peut être accordée temporairement, rejetée temporairement, accordée définitivement et rejetée définitivement par les agents.

2.2.6.3.3 Le protocole Kasbah

C'est un système où les utilisateurs créent des agents pour négocier la vente et l'achat de biens pour leur compte sur Internet. Ces biens sont classifiés, reprenant ainsi l'idée des petites annonces classées par type.

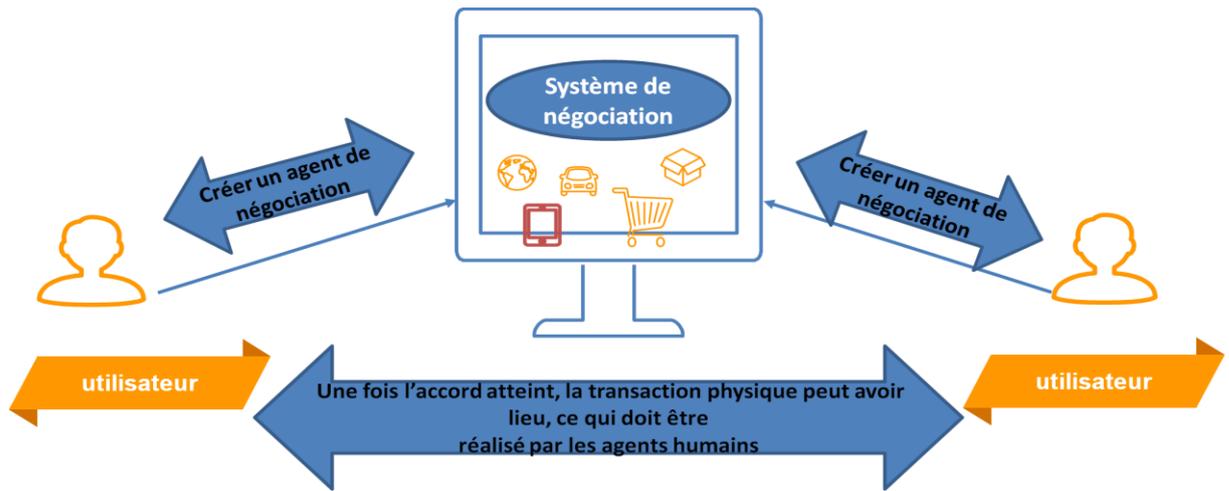


FIG 2.4 : Le protocole Kasbah

2.2.6.3.4 Le protocole AuctionBot

C'est un serveur d'enchères expérimental développé et opérationnel au laboratoire d'intelligence artificielle de l'université du Michigan par (Michael P Wellman et Peter R, 1998).

Son but est de permettre à n'importe quel internaute de participer aux enchères sur le net. AuctionBot est un framework utile à la fois pour le commerce et pour la recherche au sens où il propose une large variété de types d'enchères et une API pour créer ses propres agents qui participeront à la place de marché d'AuctionBot.

Son architecture est asynchrone, il stocke les enchères dans une base de données et il peut en gérer plusieurs simultanément.

2.2.7 Négociation basée sur la théorie des jeux

Avec les critères d'évaluation suivants :

- Critères pour évaluer les protocoles de négociation entre plusieurs agents egocentrés
- Comportement rationnel = un agent préfère une utilité (un profit) plus grande à une plus petite
- La maximisation du profit: profits individuels, profits du group, ou bien-être social
- Bien-être social: La somme des utilités (profits) des agents dans une solution.

2.2.7.1 Efficacité Pareto

- Mesure le bien-être global, demande pas la comparaison d'utilités
- Bien-être social \subset efficacité Pareto

2.2.7.2 Rationalité individuelle (RI)

RI de la participation d'un agent = Le profit de l'agent dans la solution négociée n'est pas plus petit que celui obtenu si l'agent n'aurait pas négocié

Un mécanisme est RI si la participation est RI pour tous les agents.

2.2.7.3 Stabilité

Un protocole est stable si une fois que les agents ont trouvé une solution, ils ne la changent pas.

Stratégie dominante c'est l'agent obtient le profit maximal avec une stratégie indépendant des stratégies utilisées par les autres

On dit qu'une stratégie $S1 = \{r11, r12, \dots, r1n\}$ domine une autre stratégie $S2 = \{r21, r22, \dots, r2m\}$ si n'importe quel résultat $r \in S1$ est préféré à n'importe quel résultat $r' \in S2$.

2.2.7.4 Equilibre de Nash

On suppose deux stratégies, $S1$ de l'agent A et $S2$ de l'agent B, sont dans un équilibre Nash si:

□ dans le cas où l'agent A adopterait $S1$ l'agent B ne peut pas faire mieux que d'utiliser $S2$

Et,

□ dans le cas où l'agent B adopterait $S2$ l'agent A ne peut pas faire mieux que de d'utiliser $S1$.

Problèmes :

- Si pas d'équilibre de Nash
- Si plusieurs équilibres de Nash
- garanti la stabilité seulement au début du jeu

2.2.7.5 La dilemme du prisonnier

- ✓ Ce genre de situation se rencontre fréquemment dans la vie: concurrence entre deux entreprises, conflits de personnes,

- ✓ Ce modèle très simple de la théorie des jeux semble appréhender en miniature les tensions entre cupidité individuelle et intérêts de la négociation collective

Joueur A\ Joueur B	Se tait	dénonce
Se tait	1 an ; 1 an	10 ans ; 0 an
dénonce	0 an ; 10 ans	5 ans ; 5 ans

Tab 2.1 La dilemme du prisonnier

2.2.8 Enchères

La théorie des enchères C'est les protocoles et les stratégies des agents dans les enchères

- L'initiateur veut vendre un objet au plus grand prix et les participants veulent l'acheter au plus petit prix possible
- Un protocole centralisé, inclut un initiateur et plusieurs participants L'initiateur annonce un objet pour la vente
- Les participants font des soumissions (offres). Cela peut être fait plusieurs fois, en fonction du type d'enchère
- L'initiateur choisi le gagnant

2.2.8.1 Protocoles d'enchères

2.2.8.1.1 Enchère anglaise (premier-prix offre-publique) – chaque participant

C'est un annonce publiquement son offre. Le participant avec la plus grande soumission gagne l'objet au prix de son offre.

- Dans les enchères à valeurs privées la stratégie dominante est de tjrs faire une offre avec un peu plus grande que la plus grande offre actuelle et s'arrêter quand la valeur privée est atteinte.
- Dans les enchères à valeurs corrélées, le participant augmente le prix à une rate constante ou à une rate qu'il considère appropriée.

2.2.8.1.2 Enchère premier-prix offre-cachée

Chaque participant soumet une offre sans savoir les offres des autres. Celui qui fait la plus grande soumission gagne l'objet et paye le montant de son offre.

- Pas de stratégie dominante
- Offrir moins que sa vraie évaluation, mais cela dépend des autres soumissions qui sont pas connues.

2.2.8.1.3 Enchère hollandaise (descendante)

L'initiateur diminue tout le temps le prix jusqu'à ce qu'un des participants achète l'objet au prix actuel.

- Équivalente du point de vue stratégique avec l'enchère premier-prix offre-cachée.
- Efficiente en temps réel.

2.2.8.1.4 Enchère Vickery (deuxième-prix offre-cachée)

Chaque participant soumet une offre sans savoir les offres des autres. Celui avec la plus grande offre gagne, mais au prix de la deuxième plus grande offre.

- La stratégie dominante du participant est d'offrir sa vraie évaluation

2.2.8.1.5 Enchères tous-payent

Chaque participant doit payer le montant de son offre (ou un autre montant) à l'initiateur.

2.2.8.1.6 Problèmes avec les protocoles d'enchère

- ✓ Des blocages peuvent apparaître
- ✓ Initiateur menteur
 - Problème dans l'enchère anglaise – l'initiateur utilise des faux participants dans l'enchère pour augmenter l'évaluation de l'objet par les autres participants
 - L'initiateur offre le deuxième plus grand prix pour obtenir son prix réservé – il est possible qu'il arrive à garder l'objet
 - Les enchères avec des valeurs communes peuvent être soumises à la *malédiction du gagnant*

2.2.9 Négociation basée sur les heuristiques

Les agents pouvaient accepter ou repousser une proposition (offre). Si les agents peuvent uniquement accepter ou rejeter les propositions d'autres agents, alors la négociation peut être très longue et inefficace puisque l'auteur d'une proposition n'a pas le moyen de vérifier pourquoi la proposition est inacceptable, ni si les agents sont proches d'un accord, ni dans quelle direction une proposition peut être changée pour convenir à l'autre agent.

Pour améliorer l'efficacité de la négociation, les agents doivent fournir des réactions plus utiles aux propositions qu'ils reçoivent. Ces réactions peuvent prendre la forme d'une critique ou d'une contre-proposition (proposition refusée ou modifiée). Une critique est un commentaire sur la partie de la proposition que l'agent accepte ou refuse. Une contre-proposition est une proposition alternative engendrée en réponse à une proposition. À partir de telles réactions, l'auteur doit être capable d'engendrer une proposition qui est probablement plus apte à mener à un accord.

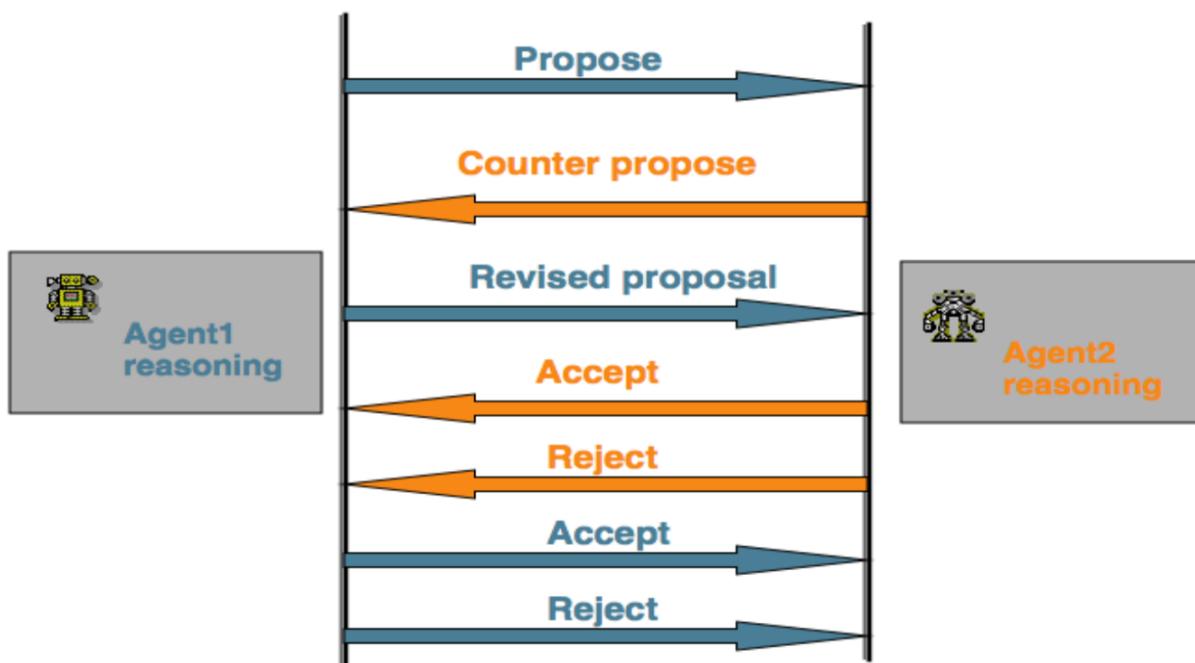


FIG 2.5 : Négociation heuristique

2.2.10 Négociation basée sur l'argumentation

Arguments utilisés pour convaincre l'autre d'accepter la proposition faite :

- Différents types d'arguments
- Chaque type d'argument définit des pré-conditions pour son utilisation.

- L'agent a besoin d'une stratégie pour décider quel argument utiliser
- On utilise un modèle BDI.

L'argument permet de :

- modifier la région d'acceptabilité
- Les seuils de récepteurs
- augmenter la probabilité d'un accord

2.2.11 Négociation basé sur les transactions

Une phase de négociation entre les agents est nécessaire pour échanger les ressources. Par exemple : si agent 1 donne ressource r1 à l'agent 2 et l'agent 2 donne la ressource r2 à l'agent 1 avec t d'argents donc si cette résultat augmente du bien être pour les deux agent 1 et 2 alors ils l'acceptent et les ressources seront changées (une transaction). Les éléments principaux qui caractérisent une transaction selon la négociation sont : le nombre d'agents et le nombre de ressource qui a été échangée entre eux .

- ❖ **One-to-one (Bilatéral)** : selon la négociation, ce type se concentre sur deux agents uniquement et plusieurs ressources passent d'un agent à l'autre.
- ❖ **Many-to-one** : Comme les enchères on a plusieurs agents qui négocient avec un seul agent.
- ❖ **Many-to-many** : On a plusieurs agents qui négocient avec plusieurs agents. C'est très complexe mais il y a un exemple dans (Friedman et Rust, 1993).

Chapitre 3

Allocation de ressources multi agent sous information incomplète

Dans ce chapitre, on va présenter notre contribution qui est autour de l'allocation de ressources entre les agents pour la distribution des ressources indivisibles (information incomplète) et aussi sans la dédommagement financier, et surtout les agents de notre cas ils ne partagent pas préférences et utilités pour atteindre une allocation de bonne qualité pour le bien-être social.

L'allocation de ressources multi agent (MARA) est un domaine de recherche qui s'intéresse à l'étude des mécanismes de distribution d'un ensemble de ressources entre un ensemble d'agents.

Pour notre étude on prend que les agents sont myopes , un agent myope , myope comme une taude ça veut dire chercher son profit au même moment et pas à long terme, on disons donc qu'il regarde ce qu'il a en main dans le moment et ne calcule pas le gain futur (utilité).

Les approches existantes concentrées sur la définition des classes d'utilité qui garantissent la convergence de la négociation en utilisant les transactions simples.

Il est actuellement supposé que les travaux existants contiennent des agents ont des préférences pour une ressource particulière, mais la ressource importe peu parfois si celle-ci est liée de manière fiable à un autre ressource tel que le téléphone et que la batterie ne peut pas être trouvée sans l'autre.

Notre cas étudie les préférences sur des paquets des ressources et non pas sur des paquets simples donc les ressources sont liées et en corrélation. On ne peut pas utiliser l'agent myope dans notre cas des paquets des ressources ou l'information est incomplète mais on a besoin d'un agent avec la capacité de gestion des risques qu'il liées avec l'information incomplète sur les préférences des agents. Si on a utilisé les approches existants pour l'allocation de ressources (chapitre 1) dans un cas où les ressources est liées sur un paquet, ces approches échoue à déterminer une solution pour allocation des ressources et les agents myopes aussi donc il faut faire un agent ou il a des mécanismes pour déterminer allocation des ressources sous information incomplètes.

Faire un exemple 3.1 : dans le tableau suivant qui représente deux agents 1 et 2 $A\{1,2\}$ et des préférences sur un ensemble de ressources $R = \{r_1, r_2, r_3\}$ comme suit :

Ressource	Agent1/Préférence	Agent2/Préférences
r1	5	3
r2	3	4
r3	2	8
r1r2	9	5
r1r3	7	9
r2r3	6	10
r1r2r3	11	11

Tab 3.1 : Les agents et leurs préférences

Quand on utilise le protocole de négociation pour faire notre négociation, on dit que le protocole de négociation ne permet pas de l'utilisation de l'argent et on peut utiliser l'échange comme un seul type de transaction par l'échange de ressource par ressource

D'abord on définit l'allocation initiale comme suit : l'agent 1 a les ressources r_1 et r_3 et l'agent 2 a la ressource r_2 , $A_0 = [\{r_1, r_3\}, \{r_2\}]$. On suppose que l'agent 1 soit le premier à proposer une offre, il fait une consultation de sa table d'utilités et il trouve que la seule transaction qui augmente son bien être est faire l'échange des ressources r_3 et r_2 , alors il propose cette offre à l'agent 2 qui trouve aussi que ça augmente son bien être et alors accepte ça, dans ce cas là, les deux agents échangent les ressources.

Dans notre cas principal, on suppose que les agents ont des préférences sur des paquets de ressources et pas sur des simples ressources (exemple précédent) avec 2 agents et 6 ressources représentées comme suit :

Ressource	Utilité
r_2r_7	4
r_5r_1	7
$r_5r_6r_3$	8

Tab 3.2 : Les préférences de l'agent 1

Ressource	Utilité
r_3r_8	6
$r_5r_1r_4$	9

Tab 3.3 : Les préférences de l'agent 2

Dans ce cas, un agent ne peut pas proposer une offre puisque il ne peut pas savoir l'utilité d'une seule ressource pour faire l'échange. Les approches existantes ne peuvent pas résoudre des problèmes de ce genre puisque ne peuvent pas répondre à plusieurs problématiques où il existe un manque d'informations : Comment un agent détermine la ressource à demander ou à proposer ? Comment un agent décide si une offre est profitable pour lui ou non.

Dans le cas des informations incomplètes sur les préférences agents, on ne peut pas utiliser un agent myope mais on doit concevoir un agent avec la capacité de gestion des risques en tenant compte l'incertitude liée à l'information incomplète sur les préférences des agents. Y a beaucoup de travaux pour réduire un peu la complexité du ce problème une transaction bilatérale en échangeant un certain nombre de ressources d'un agent à un autre. Comme le travail de contribution de (Betouil et Belleili-Souici, 2012).

Le problème principal dans la stratégie de négociation est la détermination de l'ordre dans lequel les ressources sont demandées. Dans notre proposition, on concentre sur la représentation des préférences de l'agent (information interne) avec les messages échangés pendant la négociation.

3.1 Protocole de négociation

Le protocole qu'on adopte dans notre travail est un protocole à deux phases. La première phase vise à localiser les ressources et la seconde phase du protocole est composée d'un ensemble de messages (demande/réponse) visant à atteindre une allocation de bonne qualité.

3.1.1 La première phase

Cette première phase est concernée par le protocole de négociation lui-même. On opte pour un protocole qui est une extension des offres alternatives bilatérales « *bilateral alternating offers* » (Rubinstein, 1982) qui est largement utilisé dans la négociation bilatérale. Les messages échangés entre deux agents, visent à échanger des ressources sans compensation monétaire.

Les agents suivent un ordre pour faire les offres. Une offre est faite à un seul agent choisi par moyen d'une phase de décision stratégique. Un agent qui reçoit une offre doit répondre par un message d'acceptation ou d'un rejet pour permettre à l'agent suivant de prendre son tour.

On définit trois types de messages possibles:

- ❖ *Offer (Asked-Res, Offered-Res)* : Ce message est envoyé à l'agent choisi pour lui demander la ou les ressources nécessaire (s) (*Asked-Res*) en proposant une ou des ressources (*Offered-Res*). Dans une offre, un agent tente de satisfaire un paquet et demande toutes les ressources associées, si elles sont possédées par le même agent, à l'intérieur d'une seule offre. Dans le cas où les ressources nécessaires du paquet à satisfaire appartiennent à différents agents, l'offre sera faite à un agent choisi.
- ❖ *Accept/Reject (Asked-Res, Offered-Res)* : La réponse à une offre peut être une

« acceptation » ou un « rejet » où le contenu du message est le même pour le contenu du message de l'offre « *Offer* ».

Pour obtenir un échange de ressources, lorsque deux agents sont d'accord pour une offre.

On détermine une transaction comme deux messages d'échange : le premier est le message de l'offre le second est le message de réponse.

Le modèle de négociation utilisé est « one-to-one ». Les transactions sont basées sur l'échange de ressources qui peut impliquer plus d'une ressource à la fois.

3.1.2 Deuxième phase : 'Fonction d'utilité proposée'

Pour la représentation des préférences, on a utilisé la représentation explicite ou *bundle form* (Chevalyere et al, 2006a) qui attribue pour chaque paquet (bundle) de ressources nécessaire une valeur positive.

Dans cette phase chaque agent calcule l'ordre de ses ressources nécessaires afin de faire l'importance de chaque ressource qu'il a , le résultat de cette phase est que chaque agent peut savoir la classement de besoin de ses ressources , sans aucune information sur l'importance de la ressource et leur utilité et préférence de l'autre agent donc on dit il est complètement floue pour les deux agents (information incomplète).

Comment chaque agent peut extraire l'utilité de chaque seule ressource depuis son bundle (paquet).

Cette information peut être formulée comme suit:

Soit f une fonction donne pour chaque ressource, chaque agent qui a besoin de cette ressource relativement avec son utilité pour un agent

A partir de la fonction f chaque agent peut calculer le taux d'utilité de l'importance de chaque ressource pour lui même relativement à ses paquets d'utilité des ressources pour lui même (chaque agent) :

$$\psi_{r_j} = \frac{\text{Prj} * S}{n} + 1$$

/ On ajoute 1 si notre ressource calculée est retrouve première paquet d'utilité utilisé pour chaque agent

- Prj : représente le nombre de paquet ou il y a notre ressource calculée.
- S : représente la somme des utilités ou la ressource calculée trouvée déjà.

- n : représente le nombre de paquets d'utilités utilisé pour chaque agent.

On suppose que les fonctions d'utilités sont aléatoires.

On prend un exemple comme suit, y a deux agents A1 et A2 chaque agent a trois paquets de ressources :

Ressource	Utilité
r1r2r4	8
r1r5	7
r2r3	2

Tab 3.4 : Les utilités pour l'agent 1

Ressource	Utilité
r1r4r6	9
r1r3r4	7
r1r2	5

Tab 3.5 : Les utilités de l'agent 2

On calcule la valeur de chaque seule ressource d'une chaque paquet de l'agent 1 et l'agent 2 avec la fonction d'utilité proposée et ça c'est information interne ou chaque l'agent peut obtenir le taux d'utilité de chaque ressource et faire leur classement de ressource et ses ressources préférées.

Donc pour l'agent 1 :

$$u(r1,r2,r4)=8$$

$$u(r1,r5)=7$$

$$u(r2,r3)=2$$

Donne le classement des ressources préférées comme suit :

$$r1 > r2 > r4 > r5 > r3$$

Et pour L'agent 2 :

$$u(r1,r4,r6)=9$$

$$u(r1,r3,r4)=7$$

$$u(r1,r2)=5$$

Et donne ce classement des ressources préférées comme suit :

$$r1 > r4 > r6 > r3 > r2$$

L'allocation initiale est :

$$A0(a1)=\{(r1,r2,r4)\}, A0(a2)=\{(r3,r5,r6)\}$$

Quand A1 et A2 rentrent dans la négociation dans ce cas là et dans notre exemple, la négociation est bloqué parce que l'agent A1 au niveau d'allocation initiale définit son paquet avec les ressources préférées.

Allocation initiale

Notre proposition s'inscrit dans le cadre des négociations distribuées, dans ce genre de problèmes, comme il a été évoqué précédemment (chapitre 2), le processus de résolution se commence par une allocation initiale tel que toutes les ressources sont affectées aux agents. On a choisit de générer l'allocation initiale de façon aléatoire. Ainsi, au début des négociations, chaque ressource est affectée à un et un seul agent.

3.2 Génération de problème

On a testé notre approche sur un grand nombre de scénario, variant n le nombre d'agents entre [2,5] et m le nombre de ressources entre [5,20]. Pour chaque n (nombre d'agents) et m (nombre de ressources), on a généré 10 000 exemples aléatoires de préférences des agents. On a utilisé pour cela le logiciel CATS (Leyton-Brown et Shoham, 2006) une référence standard pour générer les cas de test pour les enchères combinatoires, qu'on va présenter dans la section suivante.

CATS (Combinatorial Auctions Test Suite)

Un problème voisin du notre est apparu il y a quelques années dans le domaine des enchères combinatoires où plusieurs chercheurs travaillent sur des algorithmes de détermination du gagnant. Le problème majeur qui se pose à ces chercheurs est la génération des lots de ressources qui seront proposés aux acheteurs à un certain prix. Pour la validation et la comparaison des différents algorithmes de détermination du gagnant proposés dans ce domaine, les chercheurs ont besoin de plusieurs données pour faire des tests, ces données ne

doivent pas être générées aléatoirement, mais au contraire doivent reproduire ce que l'on pourrait rencontrer dans le monde réel. Une première idée a été proposée, elle consiste à créer des enregistrements de données stockant le comportement d'enchérisseurs, mais cela est impossible à cause de la taille de ces données (Estivie, 2006). Pour résoudre ce problème, Leyton-Brown et al ont créé un générateur d'offres pour les enchères combinatoires appelé CATS (Leyton-Brown et al., 2000). Ce générateur permet de générer des suites de tests très proches des données que l'on retrouve dans le monde réel. Le générateur CATS a été utilisé par plusieurs chercheurs qui travaillent sur les algorithmes de détermination du gagnant (Boutilier, 2002) (Zurel et Nisan, 2001). Dans la phase de la génération des données, CATS respecte les caractéristiques suivantes (Estivie, 2006) :

- ❖ Certaines ressources ont plus tendance à apparaître avec certaines que d'autres.
- ❖ Le nombre de ressources contenues dans un lot dépend du type de ressources contenues dans ce terme.
- ❖ Les coefficients (prix) associés à chaque lot dépendent de la composition de ce lot (type de ressource et nombre de ressources)

Aussi, CATS a un avantage qui permet aux chercheurs de faire des « paramétrisations » pour tester leurs algorithmes, c'est-à-dire, ils peuvent choisir le nombre de ressources et le nombre des offres, cette caractéristique permet de varier la taille du problème.

Génération avec CATS

Le mécanisme de génération est basé sur la construction d'un graphe qui permet de représenter les différents liens entre les ressources. Cette construction des graphes s'appuie des principes issus de l'économie, permettant d'extraire des relations de complémentarité entre les ressources.

Ce générateur propose plusieurs distributions qui permettent de générer des différentes données pour certains domaines d'application. D'une façon générale, il existe deux familles de distribution : les *legacy* et celles propres à des domaines d'application précis.

La première famille (les *legacy*) contient un ensemble de distribution qui génère des données de façon aléatoire en utilisant des différentes lois (normale, exponentielle, binomiale,...), ces données sont utilisées par plusieurs chercheurs qui ont développé des algorithmes de détermination du gagnant, elles sont utiles pour la comparaison des nouveaux résultats avec les résultats existants. Les quarts distributions qui ont été conçues pour la modélisation des domaines particuliers sont représentées comme suit :

- ❖ *paths in space* : cette distribution permet de générer des données très similaires à celles que l'on pourrait rencontrer dans des problèmes de connexion entre deux points tels que le problème de la gestion des réseaux de gazoducs (Rassenti et al, 1994), aussi, le cas de l'allocation de bande passante.
- ❖ *proximity in space* : dans un cas où la complémentarité entre les ressources résulte de la proximité dans un espace bidimensionnel, cette distribution est utilisée. On peut prendre comme exemple la vente de morceaux adjacents de bien immobilier (Quan, 1994).
- ❖ *temporal matching* : dans le cas où la complémentarité entre les ressources est basée sur un rapport temporel (entre les ressources), cette distribution est utilisée. Un exemple typique est le problème de la gestion des pistes d'atterrissage dans un aéroport (Rassenti et al, 1982).
- ❖ *arbitrary relationships* : dans certain cas, la relation de complémentarité entre les ressources n'est pas aussi simple (comme dans la proximité géographique ou le rapport temporel entre les ressources), cette relation de complémentarité peut être un peu complexe et dépend de certaines propriétés des ressources, comme dans le cas des enchères combinatoires où les ressources sont différentes et indivisibles avec l'existence des relations de complémentarité entre ces ressources.

Le choix d'une distribution dépend fortement du domaine d'application. Dans notre cas, on a choisi la distribution *arbitrary relationship*, puisque cette distribution représente le plus notre domaine d'application. Elle permet aussi d'exprimer les complémentarités et les corrélations entre les ressources. Les données générées par cette distribution sont très similaires aux données rencontrées dans des domaines d'application tel que l'allocation de bande passante, l'approvisionnement industriel.

Pour chaque instance du problème tel que les données sont générées par CATS, on obtient un fichier de données. Ce fichier contient un ensemble d'offres qui sont représentées par des paquets de ressources et d'une valeur pour chaque paquet. Le type utilisé pour la génération des fonctions d'utilité des agents est les fonctions XOR. Puisque ces fonctions XOR sont considérées comme un standard (Sandholm et Boutilier, 2006) pour exprimer les préférences des agents dans le domaine des enchères combinatoires, Leyton-Brown et al ont choisit ces fonctions pour générer les préférences dans CATS.

La figure 4.4 présente un exemple d'ensemble d'offres généré par CATS.

Dans cette figure, les vingt premières lignes du fichier de données représentent les paramètres du générateur. Il existe deux genres de paramètres :

- ❖ Le premier genre concerne les paramètres que l'on choisit tels que :
 - Goods : représente le nombre de ressources de notre système
 - Bids : représente le nombre d'offres à générer

Distribution : représente le type de la distribution choisie (Dans notre cas c'est *arbitrary relationship*).

- ❖ Le deuxième genre concerne les paramètres qui sont déduits ou tirés aléatoirement par le générateur (tel que seed, ...) (Estivie, 2006).

```
%% File generated by CATS v.2.1Wed Dec 5 00:27:41 2012

%% The CATS webpage is http://robotics.stanford.edu/CATS

%% PARAMETER SETTINGS:
% Goods: 12; Bids: 22; Distribution: arbitrary-upv; Runs: 1; Seed:
1354665925
% Arbitrary Distribution Parameters:
% Max Substitutable Bids = 5
% Removing HV Neighbor = 0.1
% Adding DIAG Neighbor = 0.2
% Additional Location = 0.9
% Additivity = 0.2
% Budget Factor = 1.5
% Resale Factor = 0.5
% Uniform Private Value:
% Deviation = 0.5
% Max Good Value = 100

goods 12
bids 24
dummy 4
```

0	176.20	13	6	10	12	#						
1	243.713	3	5	6	7	10	12	#				
2	214.185	2	3	5	6	7	12	#				
3	205.60	1	3	4	10	12	#					
4	140.135	0	1	2	7	8	12	#				
5	120.727	4	#									
6	321.716	3	4	5	11	13	#					
7	328.953	0	4	5	11	13	#					
8	317.341	3	4	5	9	13	#					
9	277.519	3	4	7	9	13	#					
10	136.811	2	3	7	11	13	#					
11	408.639	0	1	2	3	4	5	6	7	8	9	
	10 11	#										
12	277.197	0	1	2	4	6	7	8	9	10	11	
	14 #											
13	368.14	0	1	2	3	4	5	7	8	9	11	
	14 #											
14	350.352	0	1	2	4	5	6	7	8	9	11	
	14 #											
15	345.406	0	1	4	5	6	7	8	9	10	11	
	14 #											
16	319.888	0	1	2	4	5	6	7	8	10	11	
	14 #											
17	294.984	0	1	2	3	4	7	8	9	10	11	
	14 #											
18	377.574	1	4	5	10	11	15	#				
19	376.488	1	3	5	6	11	15	#				
20	362.418	0	1	2	5	7	15	#				
21	340.529	1	3	7	10	11	15	#				
22	287.977	2	4	5	8	11	15	#				
23	238.537	2	5	6	7	8	15	#				

FIG 3.1 : Un fichier de données généré par CATS

Les systèmes multi-agents (SMA) et la plateforme JADE

Notre approche est basée sur la négociation et la communication entre plusieurs entités appelées « agents » qui forment un groupe ou une société appelée « système multi-agents ». Un système multi-agents est représenté par un ensemble organisé d'agents. Il est composé d'une ou plusieurs organisations qui structurent les règles de communication et de travail collectif. Un agent peut appartenir à plusieurs organisations, dans un même système. Les systèmes multi-agents sont basés, principalement, sur la communication entre les agents, c'est-à-dire, un agent doit être capable à communiquer avec les autres agents. Cette phase de négociation nécessite que les agents doivent utiliser un langage commun. Pour échanger les informations et les connaissances, les agents utilisent des ACL (*Agent Communication Language*). Généralement, il existe deux types de communication:

- Communication indirecte : ce type est basé sur le partage d'informations via l'environnement.

- Communication directe : ce type est basé sur l'envoi des messages entre les différents agents.

Les interactions entre les agents sont représentées peuvent appartenir à l'un de ces 3 types :

- La coordination : ce type consiste à organiser la résolution du problème à fin de bénéficier des interactions utiles et d'éviter les interactions inutiles.
- La coopération : ce type est utilisé dans un cas où les agents ont un but commun. Il consiste à décomposer l'ensemble des tâches en sous-tâches, puis, à les allouer entre les différents agents.
- La négociation : ce type est utilisé dans un cas où les agents doivent prendre des décisions communes, sachant que chaque possède son propre but et tente de le réaliser.

Pour implémenter des systèmes multi-agents, plusieurs plateformes existent, comme : JADE, NetLogo, Madkit, VDL, CLAIM. Dans notre travail, on a choisit la plateforme JADE.

JADE (**J**ava **A**gent **D**evelopment framework), est un « framework » développé en java et permet de construire et d'implémenter des systèmes multi-agents adaptés aux spécifications FIPA, ces spécifications définissent le modèle de référence d'une plate forme d'agents et l'ensemble des services qui doivent être fournis pour une réalisation interopérable des systèmes multi-agent. Cette plateforme est open-source et distribuée par Telecom Italia sous la licence LGPL (Lesser General Public License).

JADE offre aux développeurs des systèmes multi-agents une interface graphique (la figure 4.5). Cette interface contient plusieurs outils graphiques qui facilitent la gestion des agents, comme : Dummy Agent pour l'envoi et la réception des messages ACL (« Agent Communication Language ») et le Sniffer Agent pour le contrôle des communications entre les différents agents.

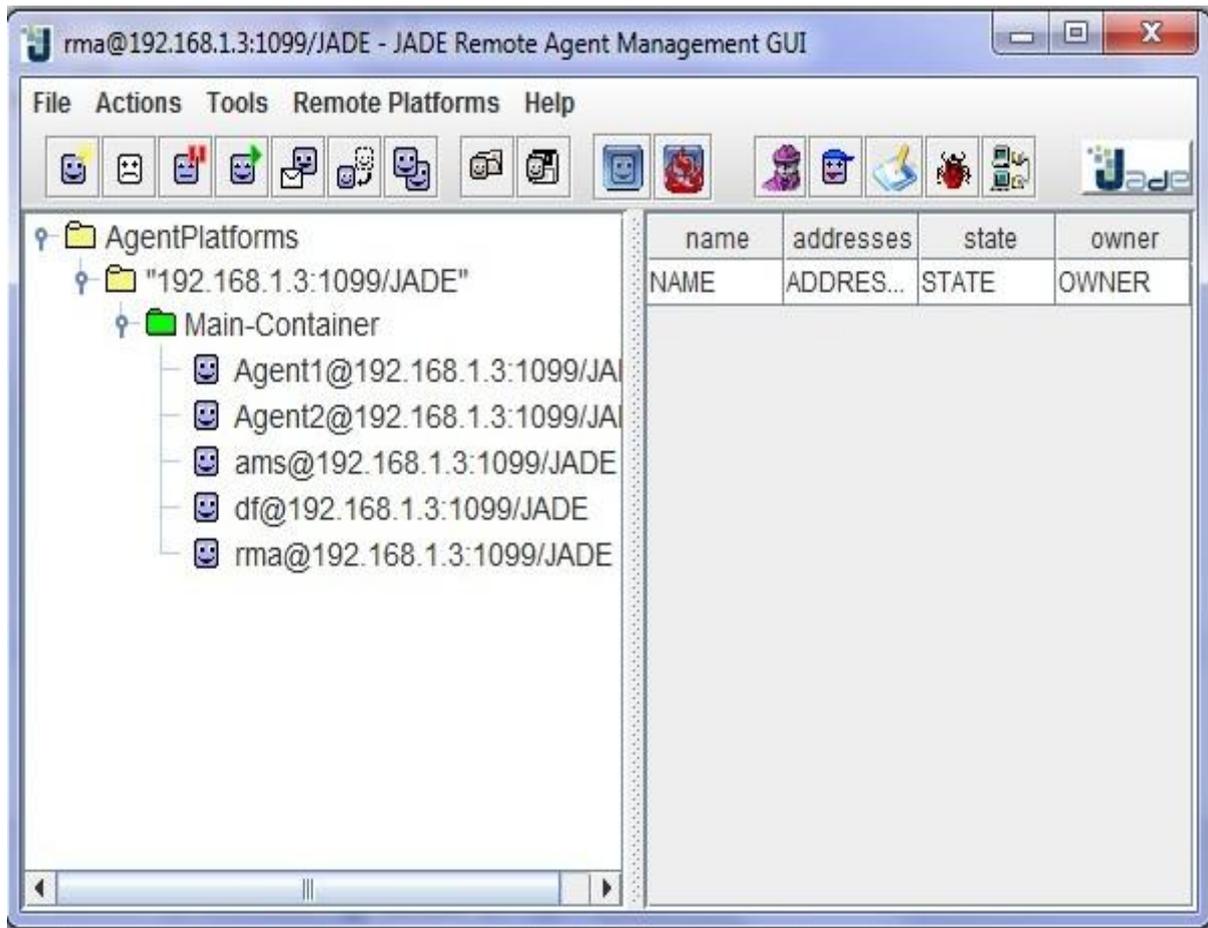


FIG.3.2-Interface graphique de la plateforme JADE

JADE contient des classes Java qui sont nécessaires pour le développement des applications d'agents, ainsi que le run-time qui fournit les services de base. A fin d'utiliser la plateforme et de bénéficier de ces avantages, il faut connaître quelques termes importants :

- ❖ Conteneur : une instance de jade run-time. Un Conteneur (figure 4.6) contient les agents (les agents vivent sur un Conteneur). Dans une plateforme, on peut trouver plusieurs Containers mais il existe un seul qui s'appelle Main Container qui gère les autres Container dans la plateforme.
- ❖ Plateforme : une plateforme est définie comme l'ensemble de tous les Containers du système.
- ❖ AMS (Agent Management System) : c'est un agent qui est responsable à la gestion du système.
- ❖ DF (Directory Facilitator) : le but de cet agent consiste à aider les agents à chercher leurs partenaires.

- ❖ Les agents : des entités développées par les programmeurs à fin de réaliser certaines tâches. Ils vivent dans les Containers.

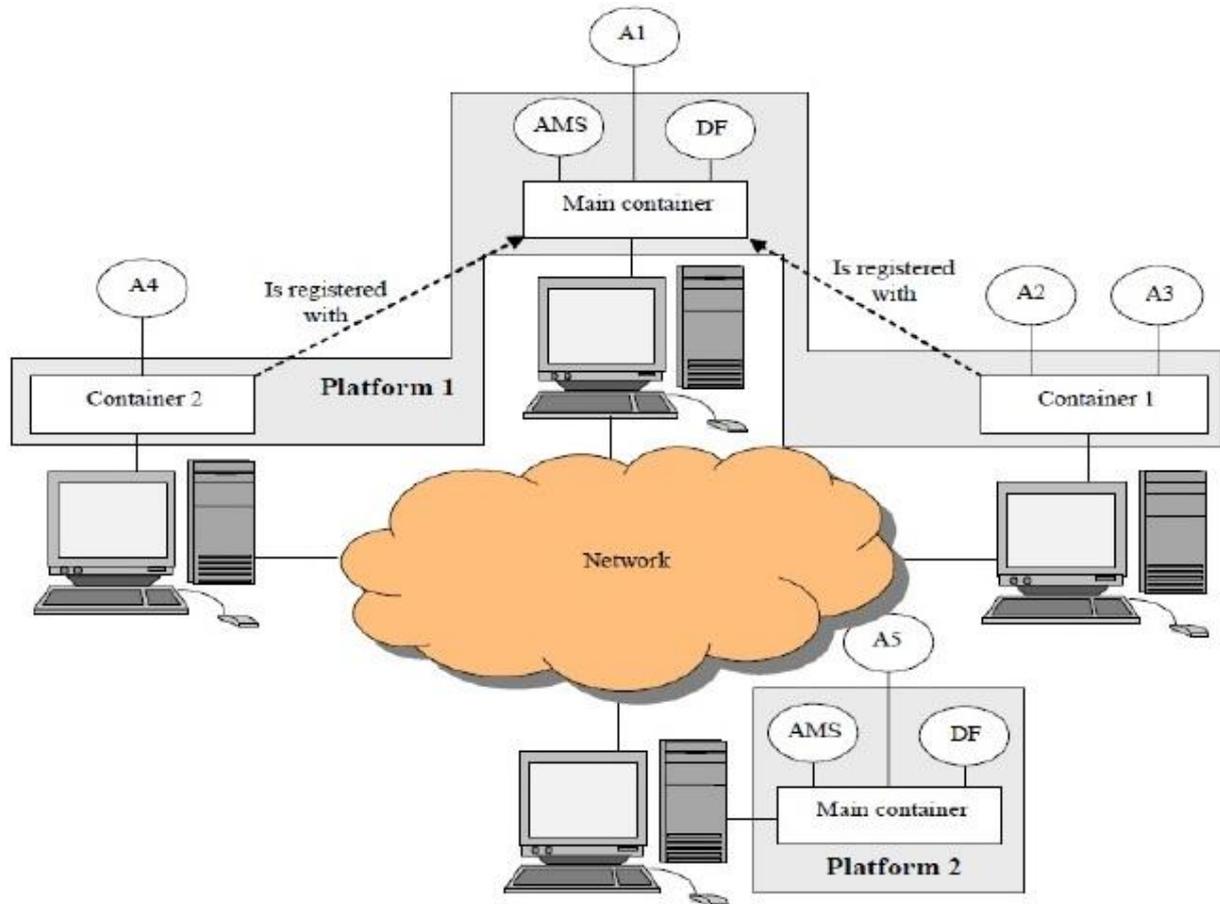


FIG.3.3- Les Containers et les plateformes (Caire, 2009)

La plateforme JADE est composée de plusieurs paquets :

- ❖ Jade.core : représente le noyau du système. Il contient la classe « Agent » qui permet de définir les tâches et le comportement des agents.
- ❖ Jade Content : Pour la définition des ontologies.
- ❖ Jade.lang.acl : Chargé pour la communication entre les agents en utilisant ACL, un message ACL est représenté comme un objet JAVA (String).
- ❖ Jade.gui : chargé pour la gestion d'interfaces graphiques.
- ❖ Jade.proto : ce paquet contient des classes permettant la définition des protocoles d'interaction standard de la spécification FIPA comme fipa-request, fipa-query, fipacontract-net.

- ❖ Jade.wrapper : ce paquet est important est consiste à utiliser JADE comme une bibliothèque qui permet aux applications de définir leurs agents et les conteneurs de ces agents.

Les agents dans JADE Dans JADE, généralement, un agent possède les états suivants (Bellifemine et al, 2010) :

- ❖ Initial (initiated) : Initialisation de l'agent, cet état consiste à annoncer, seulement, la naissance de l'agent qui n'est pas encore inscrit chez l'agent AMS, ne possède pas de nom et d'adresse et ne peut pas communiquer avec les autres agents.
- ❖ Actif (active) : Indique que l'agent est activé. Il est inscrit chez l'agent AMS et possède un nom et une adresse.
- ❖ Suspendu (suspended) : Pour indiquer que l'agent est arrêté.
- ❖ En-attente (waiting) : l'agent est bloqué pour attendre quelque chose (réponse, son tour,...)
- ❖ Inconnu (unknown) : Indique que l'agent n'existe plus dans le système (supprimé définitivement).
- ❖ Transfert (transit) : Pour indiquer qu'un agent mobile se déplace à une nouvelle place.

La figure suivante représente les différents états d'un agent dans JADE.

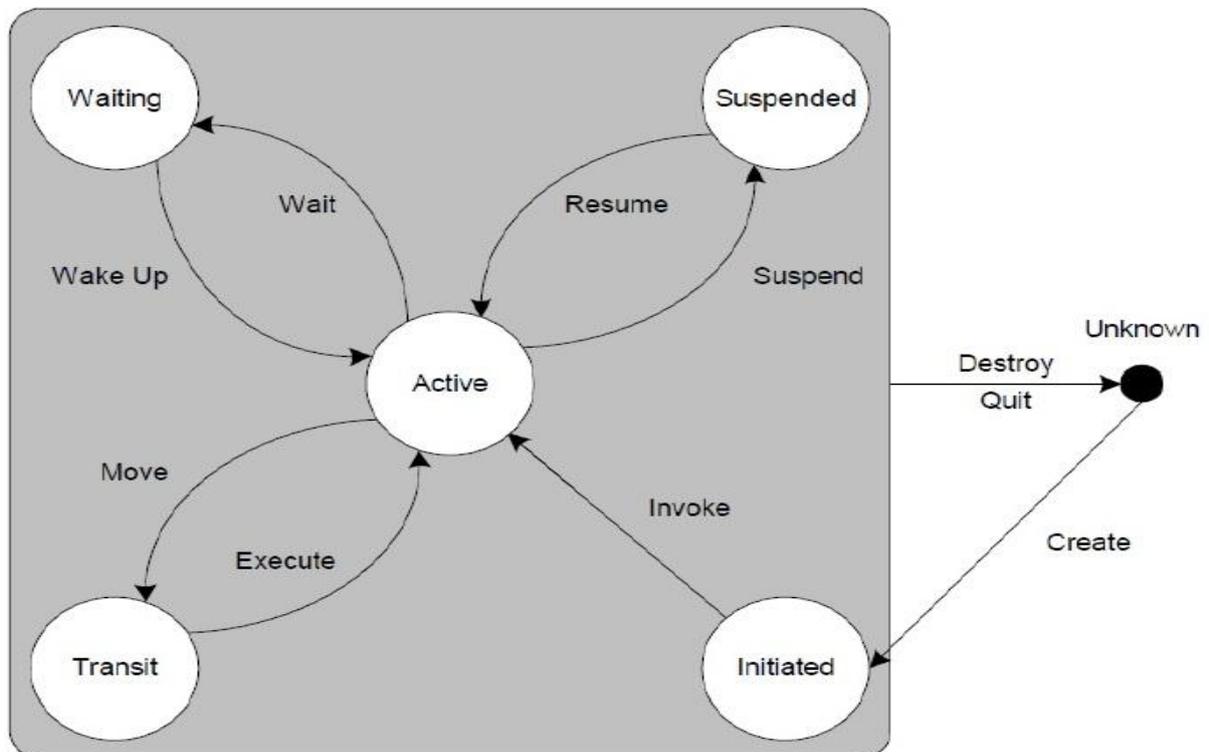


FIG.3.4-Cycle de vie d'un agent dans JADE (Bellifemine et al, 2010)

La figure 3.4 présente les différents états possibles dans Jade et les différents cas pour passer d'un état à l'autre.

3.3 Les résultats obtenus

On a testé notre approche sur un grand nombre de scénarios, on a varié le nombre d'agents (m) entre [2,5] et le nombre de ressource entre [5,20]. On a généré 10000 exemples aléatoires.

n	m	décentralisé		centralisé
		t(ms)	SW	SW
5	2	33	96,71%	98,66%
		37	96,22%	98,41%
10	3	37	92,13%	96,51%
	2	35	94,45%	97,53%
15	3	41	91,78%	95,01%
	4	47	89,47%	93,38%
	2	42	98,06%	98,73%
20	3	48	94,36%	97,91%
	4	57	90,53%	95,62%
	5	64	89,87%	91,74%
	2	42	98,06%	98,73%

Tab 3.6 : Bien être de Nash obtenu %

Le tableau suivant montre les résultats lorsque nous avons fixé le nombre de ressources à 20 et le nombre d'agents à 5 et on a varié la taille du paquet entre [2,5]. Cependant, plus la taille du paquet augmente plus le bien être social diminue et le temps d'exécution augmente, on peut expliquer ces résultats par: plus la taille du paquet augmente, plus il devient difficile d'obtenir ce paquet et le temps nécessaire augmente.

Taille du paquet	Temps (ms)	SW
2	61	93,88%
3	68	90,21%
4	73	87,63%
5	82	84,03%

Tab 3.7 : Bien Influence de la taille du paquet sur le bien être social et le temps de calcul.

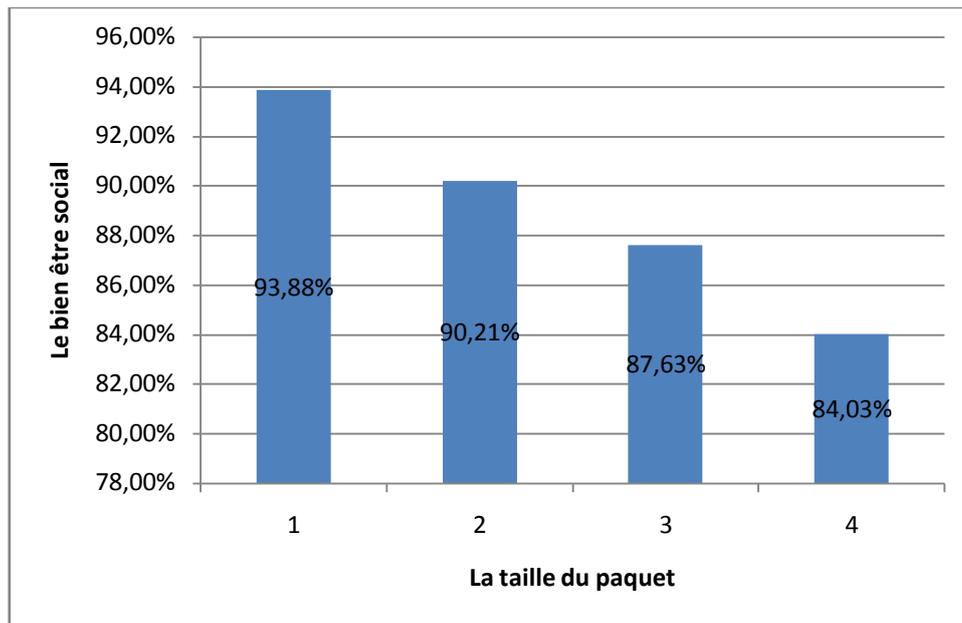


FIG 3.5 Le bien être social

D'après la table précédente, on remarque que plus la taille du paquet est grande plus le bien être social diminue et le temps de calcul augmente. On peut expliquer ce résultat comme suit : plus la taille du paquet est grande, il devient difficile d'obtenir toutes les ressources de ce paquet, ce que résulte une augmentation du temps de calcul.

Conclusion

La problématique abordée dans ce mémoire se situe dans le cadre de l'allocation de ressources basée sur un système multi-agents (MARA). Notre contribution consiste à proposer une stratégie d'allocation de ressources dans le cas des agents égoïstes. Les agents ont des préférences sur des paquets de ressources. La stratégie proposée basée sur l'échange des ressources sans aucune compensation monétaire. On a introduit la notion de la fonction d'utilité proposée comme une information interne d'une ressource à travers des paquets de ressources avec l'incertitude. Cette incertitude étant principalement due à l'information incomplète dans laquelle les agents évoluent. L'approche proposée dans ce travail se compose de deux phases : la première phase qui consiste à proposer fonction d'utilité qui permet d'estimer l'utilité de chaque ressource et la deuxième phase c'est le protocole de négociation visant à atteindre une allocation de bonne qualité.

Perspectives

Les travaux qu'on a menés et les résultats qu'on a obtenus laissent entrevoir des perspectives de recherche qui on semble devoir être explorées dans un futur proche. On les évoque dans cette section.

On a utilisé le bien être de Nash pour évaluer la qualité de l'allocation. Il existe d'autres mesures du bien être comme le bien être utilitaire, le bien être égalitaire et le bien être élitiste. Il est donc intéressant de poursuivre ces travaux par une étude de différentes notions du bien être social.

On a utilisé dans cette thèse une représentation explicite des préférences (bundle forme). Il est donc intéressant de se concentrer sur les classes de fonctions d'utilité et les types de transactions nécessaires pour chaque classe permettant de garantir la convergence.

On a présenté les différentes caractéristiques des ressources, puis on a fait des choix. Par exemple, on a choisi d'étudier seulement les ressources non partageables, discrètes,... La stratégie d'allocation proposée dans cette thèse est adaptée à ces types restreints des ressources, il est utile d'améliorer cette stratégie pour prendre en compte les différents types de ressources.

La taille du paquet est un paramètre très important et a une grande influence sur le problème étudié. Pour cela, il est intéressant d'élaborer une étude expérimentale pour déterminer la taille du paquet qui minimise le risque.

Bibliographie

Baligand F (2008). « Une Approche Déclarative pour la Gestion de la Qualité de Service dans les Compositions de Services ». Ecole Nationale Supérieure des Mines de Paris. Juin 2008.

Belleili-Souici H. and Betouil A-A (2013). « Risk management for multi-agent resource allocation under incomplete information ». In International International Journal of Systems, Control and Comuunication (2013). Inderscience Publisher.

Belleili-Souici, H, and Betouil, A-A (2011). « Using Lexicographic Preorder for Pareto Search in QoS-Aware Web Service Composition” : .In the IEEE conference :International Symposium on Innovation in Information & Communication Technology 2011. ISIICT 2011. Amman, Jordan. Bellifemine F, Caire G, Trucco T, Rimassa, G. « JADE Programmer’s Guide ». Documentation officielle de JADE.

Betouil A-A and Belleili-Souici H. (2012). « A strategy for multi-agent resource allocation under incomplete information ». In International Conference on Complex System (ICCS’2012).

Boutilier C, Goldszmidt M, Sabata B (1999) « Sequential Auctions for the Allocation of Resources with Complementarities ». In: IJCAI’99 - International Joint Conference on Artificial Intelligence, Morgan Kaufmann, Sweden, Stockolm, vol 16, pp 527–534

Chevaleyre Y, Endriss U, Lang J, Maudet N (2005) « Negotiating over Small Bundles of Resources ». In: AAMAS’05 - Autonomous Agents and multiagent systems, ACM Press, EU, The Netherlands, Utrecht, pp 296–302

Caire, G (2008). « JADE TUTORIAL- JADE Programming for Beginners » Documentation officielle de JADE.

Chevaleyre Y, Dunne P, Endriss U, Lang J, Lemaitre M, Maudet N, Padget J, Phelps S, Rodriguez-Aguilar J, Sousa P (2006a) « Issues in Multiagent Resource Allocation ». Informatica 30(1):3

Chevaleyre, Y., Endriss, U. and Maudet, N. (2006b) « On maximal classes of utility functions for efficient one-to-one negotiation », Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI-2005), Morgan Kaufmann Publishers.

Chevaleyre Y, Endriss E, Estivie S, Maudet N (2007) « Reaching Envy-free States in Distributed Negotiation Settings ». In: IJCAI’07 - International Joint Conference on Artificial Intelligence, AAAI Press, India, Hyderabad, pp 1239–1244

Chevaleyre, Y., Endriss, U., Estivie, S., et Maudet, N. (2007a). « Multiagent resource allocation in k-additive domains : Preference representation and complexity. Annals of Operations Research.

Chevaleyre Y, Endriss U, Maudet N (2009) « Simple Negotiation Schemes for Agents with Simple Preferences: Sufficiency, Necessity and Maximality ». *Journal of Autonomous Agents and Multiagent Systems*

Dunne P (2005) « Extremal behaviour in multiagent contract negotiation ». *Journal of Artificial Intelligence Research* 23:41–78

Dunne, P. E., Wooldridge, M., et Laurence, M. (2005). « The complexity of contract negotiation ». *Artificial Intelligence*, 164(1–2) :23–46.

Endriss, U., Maudet, N., Sadri, F., et Toni, F. (2003). « On optimal outcomes of negotiations over resources ». In *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2003)*. ACM Press.

Endriss U, Maudet N, Sadri F, Toni F (2006) « Negotiating Socially Optimal Allocations of Resources ». *Journal of Artificial Intelligence Research* 25:315–348
Endriss U. (2013) « Reduction of Economic Inequality in Combinatorial Domains ». In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS-2013)*, May 2013
Estivie, S., (2006). « Allocation de ressources multi-agents : théorie et pratique ». Thèse de doctorat en France.

Engel, Y et Wellman, M.P (2006) « CUI networks : A graphical representation for conditional utility independence », dans *Proc. of AAI*, pp. 83_112.

Feldman, A. M. (1973). « Bilateral trading processes, pairwise optimality, and pareto optimality ». *The Review of Economic Studies*, 40(4) :463–473.
Ferber J (1999) « Multi-Agent System: An Introduction to Distributed Artificial Intelligence ». AddisonWesley Longman

Golden, B et Perny, P (2010) « Infinite order lorenz dominance for fair multiagent optimization », dans *Proc. of AAMAS*

Klemperer, P. (1999). « Auction theory : A guide to the literature ». *Journal of Economic Surveys*, 13(3).

Chevaleyre Y, Dunne P, Endriss U, Lang J, Lemaitre M, Maudet N, Padget J, Phelps S, Rodriguez-Aguilar J, Sousa P (2006a) « Issues in Multiagent Resource Allocation ». *Informatica* 30(1):3

Nongaillard A, (2009) « An agent-based approach for distributed resource allocations ». Thèse de doctorat en France et Canada.

Sandholm, T (1998) « Contract Types for Satisficing Task Allocation I : Theoretical Results ». In: *AAAI Spring Symposium: Satisficing Models*, AAAI Press, USA, California, Stanford University, vol 99, pp 68–75

Endriss U, Maudet N, Sadri F, Toni F (2006) « Negotiating Socially Optimal Allocations of Resources ». *Journal of Artificial Intelligence Research* 25:315–348

Betouil, A-A., Belleili, H. A Strategy for Multi-Agent Resource Allocation under Incomplete

Information . In the IEEE conference : International Conference on Complex Systems (ICCS 2012). Agadir, Morocco.

Belleili,H.,Betouil A-A (2013),. Risk management for Multi-Agent Resource Allocation under

Incomplete Information. In «International Journal of Systems, Control and Communication (IJSCC) vol 5. Inderscience Publishers, Switzerland. DOI: 10.1504/IJSCC.2013.058181

Ramezani S, Endriss U (2009) « Nash Social Welfare in Multiagent Resource Allocation ». In: AMEC'09 - International Workshop on Agent-Mediated Electronic Commerce

Ramezani S et Endriss U (2010). « Nash Social Welfare in Multiagent Resource Allocation ». In Agent-Mediated Electronic Commerce: Designing Trading Strategies and Mechanisms for Electronic Markets, Lecture Notes in Business Information Processing, volume 59, pages 117-131, Springer-Verlag, 2010.

A. Selma, “Approche dirigée par les modèles pour le développement de systèmes multiagents”,

Thèse de doctorat en informatique, université de Savoie, 2007

H. Joumaa, ‘Analyse des performances d'un système multi-agents par visualisation’, thèse de doctorat, Université de Grenoble, 2010.

J Ferber, Multi-agent systems: an introduction to distributed artificial intelligence, article publié 25/02/1999.

F Med.El-Kabir, “Une approche basée agents pour l’allocation des ressources dans le cloud computing”, mémoire en vue de l’obtention du diplôme de magister en informatique, Université Mohamed Khider-Biskra, 2015.

M-R.HABES, « Allocation de ressources dans le cloud computing », Thèse de doctorat en informatique, Université de Annaba, 2016.

Olivier Gutknecht, Jacques Ferber, The MADKIT Agent Platform Architecture, ConférenceWorkshop on Infrastructure for Scalable Multi-Agent Systems at the International Conference on Autonomous Agents, 2000

Ramezani S, Endriss U (2009) « Nash Social Welfare in Multiagent Resource Allocation ». In: AMEC'09 - International Workshop on Agent-Mediated Electronic Commerce

Sandholm, T. W. et Boutilier, C. (2006). « Preference elicitation in combinatorial auctions ». In Cramton, P., Shoham, Y., et Steinberg, R., editors, Combinatorial Auctions, chapter 10. MIT Press.

Sen, A. K. (1970). « Collective Choice and Social Welfare ». Holden Day.

Woolridge M (2001) « Introduction to Multiagent Systems ». JohnWiley & Sons

