

وزارة التعليم العالي و البحث العلمي

Ministère de l'enseignement Supérieur et de la Recherche Scientifique

BADJI MOKHTAR UNIVERSITY – ANNABA
UNIVERSITE BADJI MOKHTAR – ANNABA



جامعة باجي مختار – عنابة

Faculté des Sciences de l'Ingéniorat

Année : 2015/2016

Département d'informatique

THESE

Présentée en vue de l'obtention du diplôme de :

DOCTORAT 3^{ème} Cycle LMD

ALLOCATION DE RESSOURCES DANS LE CLOUD COMPUTING

Option

Science de la Technologie de l'Information et de la Communication (STIC)

Par

Mohamed Raouf HABES

DIRECTEUR DE THESE : Habiba BELLEILI MCA U.Badji-Mokhtar Annaba (UBMA)

DEVANT LE JURY

PRESIDENT: M.T. KHADIR	Professeur	U.Badji-Mokhtar Annaba (UBMA)
EXAMINATEURS : M.REDJIMI	Professeur	U. 20 Août 1955-Skikda
H.SERIDI	Professeur	U.Badji-Mokhtar Annaba (UBMA)
N.AZIZI	MCA	U.Badji-Mokhtar Annaba (UBMA)

Faire simple est probablement l'objectif le plus sophistiqué du monde

Steve Jobs

Remerciements

Je remercie Dieu avant toute chose, qui m'a donné l'énergie et la patience nécessaire pour finaliser le long travail de thèse.

Je tiens à remercier sincèrement madame Habiba Belleili-Souici pour l'encadrement qu'elle a apporté à ce travail, l'expérience qu'elle a pu me transmettre et les responsabilités qu'elle a bien voulu me confier. Plusieurs des outils utilisés lui doivent d'exister et ses remarques ont eu un poids décisif au cours de l'évolution de ce travail.

Je désire également remercier Monsieur Tarek Khadir pour sa sympathie et l'honneur qu'il m'a fait en acceptant de présider le jury de ma thèse.

Mes plus sincères remerciements vont aux trois personnes qui ont accepté d'être rapporteurs de ce travail. Monsieur Mohamed Redjimi, Madame Hassina Seridi et Madame Nabiha Azzizi.

J'adresse ma profonde reconnaissance à Monsieur Laurent Vercouter de Rouen pour toute l'aide qu'il m'a apportée.

Je remercie vivement l'équipe de LabGED pour leur écoute et leur contribution efficace à la réalisation de ce travail.

Pour finir je tiens à adresser le plus grand merci à ma famille pour son soutien inconditionnel.

ملخص

تلقت مشكلة تصميم نظام متعدد الوكيل اهتماما متزايدا في السنوات الأخيرة. عندما نتحدث عن وكيل، يجب أن يؤخذ بمعناه الواسع، لأن الوكيل يمكن أن يمثله الروبوت، الشخص الظاهري، برنامج يتصرف نيابة عن شخص آخر ... في هذه الأطروحة، ركزنا على تطبيق النظام متعدد الوكلاء لتخصيص الموارد في الحوسبة السحابية. كل وكيل يمكن أن يكون له تشكيل به من حيث القدرات، والأفضليات، الهدف والقيود. بالإضافة إلى ذلك، التفاوض على الموارد لأغراض التأجير هي بطبيعتها أكثر تعقيدا من أغراض شراء ، مما يجعل المشكلة أكثر صعوبة للعثور على الحل الأمثل.

التفاوض الآلي هو نهج مشترك في مجال تخصيص الموارد عند إستعمال نظام متعدد الوكيل. ولكن بأخذ بعين الإعتبار لمشاكل العالم الحقيقي مما ينطوي عليه من عدم اليقين، وكلاء المصلحة الذاتية، ووجود تعدد الشركاء والمنافسين التجاريين، يسبب أن توليد الحل الأمثل في ظل هذه المعلومات غير ممكن فيما يتعلق بالموارد الحوسبة الحسابية في الوقت الحاضر. فلهذا السبب لجأنا في هذه الأطروحة إلى مناهج تقريبية التي هي أقل تطلبا في الموارد وتسمى نهج اللامركزية. حيث المشتري والبائع المشاركين في العقد يجب أن يتفقا على السعر في الموارد المتفاوض عليها وكذلك عقوبة فك العقد في حالة التي يكون فيها أحد من الطرفين المعنيين في العقد لم يعد راغب فيه. وقد اقترح ثلاثة مساهمات في هذا الاتجاه، وكان أول منهم يتمثل في الاعتماد على مؤشر ثقة الطرف الآخر من العقد عند اختيار الشريك التجاري، والثاني في استكشاف أكثر بعمق على الفضاء اتفاق للمشتري عندما يأتي به الأمر على اقتراح عرض والثالث والأخير هو على تصميم استراتيجية تصنيف ذو جودة أفضل للبائع عند حصوله بعروض المشتريين من أجل الحفاظ على العروض التي تعتبر الأكثر إثارة للاهتمام.

نحن قارنا تجريبيا نهجنا باستخدام سيناريوهات تمثيلية وأعباء العمل المختلفة ضد مناهج ذات الصلة المعروفة في الأدب وأظهرنا أن نماذجنا تحقق نتائج أفضل في عدة مقاييس الأداء.

الكلمات الرئيسية : الحوسبة السحابية، أنظمة متعددة وكيل، استراتيجية التفاوض والسوق الإلكترونية.

Résumé

Le problème de la conception de système multi-agents a reçu une attention croissante ces dernières années. Quand on parle d'agent, il faut le prendre au sens large, car les agents peuvent être sous-forme de robot, personne virtuelle, programme agissant pour le compte d'une tierce personne ... Dans cette thèse, nous nous sommes focalisés sur l'application des systèmes multi-agents pour l'allocation de ressources dans le cloud. Chaque agent peut avoir sa propre configuration en termes de capacités, préférences, objectifs et contraintes. De plus, la négociation de ressources pour des fins de locations en elle-même est plus complexe par rapport à celle de l'achat, ce qui rend une solution automatisée optimale à ce problème difficile à réaliser.

La négociation automatisée est l'approche la plus répandue dans le domaine d'application pour l'allocation de ressources multi-agent. Mais vue que les problèmes pratiques réels impliquent l'incertitude, des agents à intérêt personnels, et l'existence de multiple partenaires et compétiteurs commerciaux, la génération d'une solution optimale en vertu de tels paramètres n'est pas envisageable par rapport aux ressources informatiques de nos jours. C'est pour cela, que dans cette thèse nous avons eu recours à des approches approximatives moins gourmandes en ressources qui sont des approches de négociation de type décentralisées. Où l'acheteur et le vendeur impliqués dans le contrat doivent se mettre d'accord sur le prix des ressources négociées ainsi que sur pénalité de désengagement en cas où une des deux parties ne serait plus intéressée par le contrat en question. Trois contributions ont été proposées dans ce sens, la première consistait à se fier à l'indicateur de confiance de l'autre partie du contrat lors du choix du partenaire commercial, la deuxième à explorer plus en profondeur l'espace d'accords de l'acheteur lors de la formulation de l'offre de ce dernier et la troisième est la stratégie de classification des offres reçues par le vendeur afin d'en retenir que celles considérées comme les plus intéressantes.

Nous avons comparé expérimentalement nos approches, en utilisant des scénarios représentatifs et des charges de travail différentes par rapport à des approches bien connues de la littérature qui sont relatives à la nôtre et nous avons montré que nos modèles réalisent un meilleur résultat dans plusieurs mesures de performances.

Mot-clés : Cloud Computing, Système Multi-agents, Stratégie de négociation et Marché électronique.

Abstract

The problem of designing multi-agent system has received an increasing attention in recent years. When we talk about agent, it must be taken in a broad sense, because agents can be represented by robot, virtual person, program acting on behalf of another person ... In this thesis, we focused on the application of multi-agent systems for the allocation of resources in the cloud. Each agent can have its own configuration in terms of capabilities, preferences, objectives and constraints. In addition, the negotiation of resources for leasing purposes is inherently more complex than buying purposes, which makes the problem even more difficult to find out an optimal solution.

Automated negotiation is the common approach in the domain of resource allocation with multi-agent system. But in accounting for real world problems involving uncertainty, self-interested agents, and the existence of multiple commercial partners and competitors, generating an optimal solution under such parameters is not possible with respect to the computational resources of nowadays. This is why, in this thesis we have resort to approximate approaches which are less demanding in resources and are called decentralized approaches. Where the buyer and seller involved in the contract have to agree on the price of the resources negotiated as well as the disengagement penalty in case where one of the two parties involved in the contract would no longer being interested by it. Three contributions have been proposed in this direction, the first one had consisted on relying to the indicator of trustworthiness of the other party of the contract when choosing a commercial partner, the second one in exploring more in-depth the agreement space of the buyer when this latter comes to propose an offer and the third and last one is on designing a better seller strategy classification for the offers received by the seller in order to retain only offers which are considered the most interesting ones.

We experimentally compared our approaches using representative scenarios and different workloads against well-known related approaches in literature and we showed that our models achieve better results in several performance measures.

Key-words : Cloud Computing, Multi-Agent systems, Negotiation Strategy and Electronic Marketplace.

Table des matières

Remerciements	ii
ملخص	iii
Résumé	iv
Abstract	v
Table des matières	vi
Liste des tableaux	xii
Liste des figures	xiii

PARTIE 1 INTRODUCTION GENERALE

Chapitre 1 Introduction	2
1.1 Introduction	2
1.2 Exemple d'application de négociation automatisée sur les ressources	3
1.2.1 Collaborating, Autonomous Stream Processing Systems (CLASP).....	3
1.2.2. Global Environment for Network Innovations (GENI)	5
1.3 Les contributions principales	8
1.4 Organisation de la thèse.....	9
Chapitre 2 Présentation du Cloud Computing	10
2.1 Introduction	10
2.2 Définition du Cloud computing	10
2.3 Modèle de déploiement du Cloud.....	10
2.4 Grid computing vs Cloud computing	11
2.5 Service du Cloud	11
2.5.1 Software as a Service	11

2.5.2 Platform as a Service	11
2.5.3 Infrastructure as a Service.....	11
2.6 Les caractéristiques du Cloud.....	12
2.6.1 Les aspects techniques	12
2.6.2 Les aspects qualitatifs	13
2.6.3 L'aspect économique	14
2.7 Projets du Cloud	14
2.7.1 Produit commercial	14
2.7.2 Projets de recherche	15
2.8 Comparaison générale des simulateurs de Cloud computing.....	17

Partie 2 ALLOCATION DE RESSOURCES

Chapitre 3 L'allocation de ressources : une Cartographie21

3.1 Introduction	21
3.2 L'allocation de ressources : définition.....	22
3.3 Les ressources.....	24
3.3.1 Continues vs Discrètes	24
3.3.2 Divisibles ou non	24
3.3.3 Partageables ou non	24
3.3.4 Statiques ou non	25
3.3.5 Mono-Item vs Multi-Items.....	25
3.3.6 Ressources ou tâches.....	26
3.4 Outils : Les différentes mesures du bien-être social.....	26
3.4.1 Bien-être social Utilitaire	26
3.4.2 Bien-être social égalitaire.....	27
3.4.3 Produit de Nash.....	27

3.4.4 Bien-être social élitiste.....	27
3.4.5 L'absence d'envie	28
3.4.6 L'optimalité au sens de Pareto.....	28
3.5 Exemples d'applications.....	28
3.5.1 Des applications à caractère égalitaire	29
3.6 Revue de littérature sur l'allocation de ressources dans le Cloud	31

PARTIE 3 MODELE CENTRALISE ET DECENTRALISE D'ALLOCATION DE RESSOURCES

Chapitre 4 Taxonomie de l'enchère34

4.1 Théorie de l'enchère :	34
4.1.1- Enchère à un côté pour une seule ressource	35
4.1.2. Enchère à deux côtés pour une seule ressource	37
4.1.3. Les enchères combinatoires	39

Chapitre 5 Négociation automatisée.....48

5.1 Introduction	48
5.2 Le problème de négociation	48
5.2.1 L'espace d'accords de la négociation	48
5.2.2 Mécanisme de négociation.....	50
5.3 Approches de négociation basée sur la théorie des jeux.....	51
5.3.1 Un aperçu de la théorie des jeux non coopératifs	51
5.3.2. La théorie des jeux pour la négociation automatisée	53
5.3.3 Les limitations de la théorie des jeux.....	53
5.4 Approches basées heuristique pour la négociation.....	55
5.4.1. Heuristiques pour la négociation automatisée	55
5.4.2. Limitations des approches heuristiques	56

5.5	Approches de négociation argumentatives	56
5.5.1.	Aperçu sur les modèles heuristiques et la théorie des jeux.....	56
5.5.2.	Les arguments dans la négociation	57
5.6	Négociation automatisée pour les problèmes d'allocation de ressources complexe	58
5.6.1.	Négociation avec incertitude.....	59
5.6.2.	Négociation un-à-plusieurs et plusieurs-à-plusieurs	60
5.6.3	Négociation Multi-Ressources.....	60
5.6.4	Négociation avec désengagement pour l'allocation de ressource dynamique dans le Cloud.....	63

PARTIE 4 CONTRIBUTIONS

Chapitre 6 : Allocation de ressources basée confiance.....67

6.1	Introduction	67
6.1.1	Description du problème.....	69
6.1.2	Protocole de négociation.....	71
6.1.3	Disposition de l'axe de temps	74
6.2	Stratégie de négociation de l'acheteur.....	75
6.2.1	Stratégie de négociation de l'acheteur concernant l'ensemble des accords.....	76
6.2.2	Stratégie de confirmation de l'acheteur	76
6.2.3	Stratégie de désengagement de l'acheteur	76
6.2.4	Stratégie de préférence de l'acheteur sur les vendeurs	78
6.2.5	Stratégie de fréquence de négociation de l'acheteur.....	81
6.2.6	Pseudo algorithme de la stratégie de l'acheteur.....	82
6.3	La stratégie du vendeur.....	83
6.4	Evaluation expérimentale	87
6.4.1	Paramètres expérimentaux	87

6.4.2 Résultats expérimentaux	89
6.5 Conclusion	97
Chapitre 7 : Stratégie de négociation d'acheteur dynamique et hybride pour l'allocation de ressources	98
7.1 Introduction	98
7.2 Description du problème	100
7.2.1 Etapes de l'offre	101
7.2.2 Vue d'ensemble du problème de négociation	102
7.3 Stratégie de négociation de l'acheteur	102
7.3.1 La stratégie d'ensemble d'accords de l'acheteur	107
7.3.2 La stratégie de confirmation	107
7.3.3 Stratégie du mode de négociation de l'acheteur	108
7.3.4 Stratégie de planification de l'acheteur	112
7.3.5 Stratégie d'abandon	113
7.4 Stratégie du vendeur	115
7.5 Evaluation expérimentale	116
7.5.1 Paramètres de Simulation	116
7.5.2 Résultats expérimentaux	118
7.6 Conclusion	126
Chapitre 8 : Stratégie de vendeur améliorée dans une négociation décentralisée plusieurs-à-plusieurs	127
8.1. Introduction	127
8.2. Contrat d'engagement à niveau	130
8.2.1 Format de soumission de l'offre	130
8.2.2 Les phase d'accord dans la négociation d'engagement à niveau	130

8.3. La stratégie du vendeur.....	131
Le processus de résolution WDPS :.....	134
8.4. Evaluation expérimentale	136
8.4.1. Configuration expérimentale.....	136
8.4.2. Observations et analyse des résultats	137
8.5. Conclusions	142

PARTIE 5 CONCLUSION

Chapitre 9 Conclusion et Recherches futures..... 145

9.1 Contributions	145
9.2 Recherches futures.....	146

Référence bibliographiques 147

Publications Internationales 158

Communication Internationale 158

Annexe A : CloudSim : Outil de simulation 159

A.1 Introduction.....	159
A.2 Les fonctionnalités générales du CloudSim	159
A.3 Enregistrement et communication entre les entités	160
A.4 Système de simulation du CloudSim.....	161
A.5 Politique d’approvisionnement de la ressource	162
A.5.1 Allocation des ressources de stockage, bande passante et Ram.....	163
A.5.2 Allocation d’élément de traitement (Pe)	163
A.6 Cas d’utilisation.....	166

Liste des tableaux

Tableau 1 Comparatif des outils de Simulation du Cloud	17
Tableau 2 Taxonomie de la conception de l'enchère	35
Tableau 3 Dilemme du prisonnier.....	52
Tableau 4 Variables de la proposition 1.....	87
Tableau 5 Symboles Utilisés	104
Tableau 6 Paramètre de prix de l'acheteur Tableau 7 Statut de négociation des vendeurs	108
Tableau 8 Variables de la deuxième proposition	117
Tableau 9 Symboles Utilisés	132
Tableau 10 Variables.....	136

Liste des figures

Figure 1 Exécution d'une tâche distribuée consistant en trois sous-tâches [21]. Le propriétaire du Site 3 exécute une sous-tâche, et dispatche deux sous-tâches au Site 1 et 2 pour exécution. Site 4 supervise Sites 1 et 2.	4
Figure 2 Partage de ressource par le biais d'un consommateur et un fournisseur [3].....	5
Figure 3 Partage de ressource par le biais de plusieurs consommateurs et fournisseurs [3].....	6
Figure 4 Allocation de ressources [38].....	23
Figure 5 Allocation de ressources satellitaires [20]	29
Figure 6 Exemple d'un carnet de commande de "periodical double-sided auction"	39
Figure 7 Machine à état fini pour le protocole de négociation [16]	71
Figure 8 Agencement de l'axe de temps	74
Figure 9 Stratégie de négociation de l'acheteur sur l'ensemble d'accords.	76
Figure 10 Stratégie de désengagement de l'acheteur	77
Figure 11 Tendance d'annulation de l'acheteur et compétition en ressources.....	90
Figure 12 Tendance de désengagement du vendeur et compétition en ressources	91
Figure 13 taux de succès et compétition en ressources	92
Figure 14 Transactions exécutées et compétition en ressources	93
Figure 15 Bien être sociale et compétition en ressources.....	94
Figure 16 Utilité de l'acheteur et compétition en ressources	95
Figure 17 Utilité du vendeur et compétition en ressources	96
Figure 18 Machine à état finie des étapes de l'offre [16].....	101
Figure 19 Disposition de l'axe de temps	102
Figure 20 Atteignabilité du prix et les chances de succès de l'offre de l'acheteur	111
Figure 21 Issue d'une négociation avec inter-blocage entre l'acheteur 2 et 3.	113
Figure 22 Ratio du bien être compétition en ressources	119
Figure 23 Taux de succès des acheteurs et compétition en ressources	119
Figure 24 Temps de démarrage de la tâche d'exécution et compétition en ressources.....	120
Figure 25 Utilité de l'acheteur et compétition en ressources	122
Figure 26 Ratio du bien-être social et compétition en ressources	123
Figure 27 Taux de succès de l'acheteur et compétition en ressources	123

Figure 28 Temps de départ de l'exécution de la tâche de l'acheteur et compétition en ressources	124
Figure 29 Utilité de l'acheteur et compétition en ressources	124
Figure 30 Revenu de pénalité de l'acheteur et compétition en ressources	125
Figure 31 Machine à état finie des phases du contrat [16]	131
Figure 32 Bien-être social et compétition en ressources	138
Figure 33 Utilité de l'acheteur en moyenne et compétition en ressources	138
Figure 34 Acheteurs servis et compétition en ressources.....	139
Figure 35 Revenu moyen du vendeur et compétition en ressources	140
Figure 36 Temps d'exécution de la stratégie du vendeur et compétition en ressources	141
Figure 37 Taux d'utilisation et compétition en ressources	141
Figure 38 Flux de donné de Simulation [23].....	161
Figure 39 Diagramme de Classe du noyau du système de simulation CloudSim.....	161
Figure 40 Les différents effets des politiques d'approvisionnements sur l'exécution de la tâche [23]	166
Figure 41 Diagramme de cas d'utilisation pour la négociation de ressources en utilisant CloudSim.....	166



PARTIE 1 :

INTRODUCTION

GÉNÉRALE

Chapitre 1 Introduction

1.1 Introduction

Parmi les domaines de recherche qui ont prêté une attention particulière au problème d'allocation de ressources, se trouve l'économie, la recherche opérationnelle et l'informatique. De plus le problème d'allocation de ressources est pertinent à une large gamme d'applications, tel que le commerce électronique, la chaîne d'approvisionnement, les réseaux de capteurs, la composition de service grid/web, le flux de travail, et l'intégration d'application d'entreprise. Les systèmes d'agents autonomes font appel à l'informatique et l'économie pour résoudre les problèmes d'allocation de ressources. La tâche de résolution du problème d'allocation de ressources peut être gérée de façon centralisée ou distribuée. Elle est centralisée comme le cas des enchères combinatoire où c'est le commissaire-priseur qui émet les décisions d'allocation de ressources [26, 65]. Ou bien décentralisée via la négociation automatisée, est sa deviendra le rôle des participants (agents) du marché à contribuer dans les décisions d'allocation de ressources. La négociation automatisée est un domaine de recherche qui fait appel à l'intelligence artificielle, la théorie des jeux et l'économie. Elle a reçu une attention importante ces dernières années [54] et son importance est largement reconnue due au fait que les agents intelligents qui négocient entre eux pour le compte d'utilisateurs humains sont censés produire des négociations plus efficaces [95].

Dans la littérature multi-agents pour la négociation automatisée, des mécanismes basés sur l'analyse de la théorie des jeux [59, 99], les approches heuristiques [39, 40] et sur l'argumentation [58, 101] ont été proposées. Bien qu'on puisse classifier les approches relatives à la négociation automatisée en plusieurs classes, la classification adoptée dans cette thèse, suit la classification de Jennings et al. [54]. Parmi les classes les plus répandues figure le marchandage, où l'accord ne peut être conclu que par l'approbation des deux parties concernées.

Dans les systèmes d'agents intelligents comme les marchés électroniques, la préoccupation majeure des agents est de maximiser leurs propres biens sans se préoccuper des autres. Pour cela l'agent doit trouver la meilleure stratégie à employer étant donné le protocole de négociation mis en place. La tâche de la stratégie à adopter est rendue difficile par la dynamique qu'implique les marchés de négociation plusieurs-à-plusieurs du cloud et l'incertitude dans les contrats d'engagement à niveaux [96]. Ces types de contrats bien qu'ils rendent difficile la conception de

stratégies d'agents optimales permettent aux approches de négociation décentralisées de générer une meilleure approximation.

Dans cette thèse, nous nous intéressons aux stratégies de négociation des agents dans des environnements où : 1) La négociation est de type plusieurs-à-plusieurs, c'est-à-dire que la concurrence se fait aussi bien du côté de l'acheteur que du côté du vendeur ; 2) La négociation est aussi multi-items, dans le sens où chaque agent requière ou vend chaque item en zéro ou plusieurs copies ; 3) les contrats négociés sont destinés à la location de ressources avec engagement à niveau ; 4) les agents ont une information incomplète au sujet des autres agents.

1.2 Exemple d'application de négociation automatisée sur les ressources

Cette thèse va présenter deux exemples pratiques qui traitent du problème d'allocation de ressources dans le cloud computing. Cela permettra de motiver les contributions apportées par la suite dans le domaine d'allocation de ressources dans le cloud d'un point de vue pratique.

1.2.1 Collaborating, Autonomous Stream Processing Systems (CLASP)

Collaborating, Autonomous Stream Processing Systems (CLASP) [21] est un middleware pour la coopération des sites de traitement de flux de données, qui a été désigné et prototypé dans le cadre du projet Système S [73] au sein d'IBM Research pour permettre le traitement de flux sophistiqués. Il y a plusieurs sites qui font tourner le Système S, chacun avec sa propre administration et ses objectifs. Chaque site peut seulement avoir des capacités limitées de traitement, ainsi la coopération entre ces sites peut souvent dégager un bénéfice mutuel et une telle coopération leur permet de passer à l'échelle, d'augmenter la largeur, la profondeur et la fiabilité de l'analyse au-delà des capacités de traitement disponibles au sein d'un seul site.

Dans le système S, une tâche est une unité d'exécution qui accomplit certains travaux par l'analyse du flux. Un travail prend la forme d'un graphe de traitement, comprenant des ressources, à savoir, les ressources de données et les éléments de traitement (PE), qui sont reliés entre eux d'une certaine manière. Ces ressources peuvent être situées sur plusieurs sites différents. En raison du nombre potentiellement grand des sources de données et des PEs nécessaires pour l'accomplissement des travaux complexes, ainsi que l'existence de graphes de traitement fonctionnellement équivalents, il est impossible pour les utilisateurs humains de construire et d'identifier le meilleur graphe alternatif. Système S a une composante de planification qui peut

construire automatiquement des graphes de traitement à partir de descriptions de haut niveau des résultats escomptés.

La Figure 1 illustre un exemple de l'exécution d'une tâche distribuée [21]. Site 3 est responsable de l'exécution d'une tâche, qui peut être décomposée en trois sous-tâches, chacune d'elles contient une tâche dite "normale" (pour le traitement de données) et des tâches à tunnel PE (pour le transport de données). Le coordinateur de l'exécution à distance (REC) du site propriétaire 3 exécute la troisième sous-tâche et distribue les deux autres sous-tâches vers les sites 1 et 2 pour l'exécution. Site 4 surveille l'état d'exécution du Site 1 et 2. Le REC du site propriétaire maintient un tableau de sous-tâches sur lequel on trouve quelle sous-tâche est en train de s'exécuter sur quel autre site. Ce tableau est utilisé pour la récupération de sous-tâches sur des sites défaillants. Le REC exécutant la sous-tâche analyse d'abord le langage de description de la tâche (JDL) pour identifier une tâche normale et de multiples tâches PE à tunnel. Un thread est lancé pour traiter chacune d'elles.

Beaucoup de ressources requis dans les plans générés par les tâches sont accessibles de façon exclusive. Afin qu'un site réserve une ressource limitée d'un autre site, il doit établir un accord avec celui-là, en précisant le prix de partage de la ressource. Ensuite, le site négocie avec d'autres sites pour acquérir les ressources nécessaires à l'aide de sa composante de gestion de négociation.

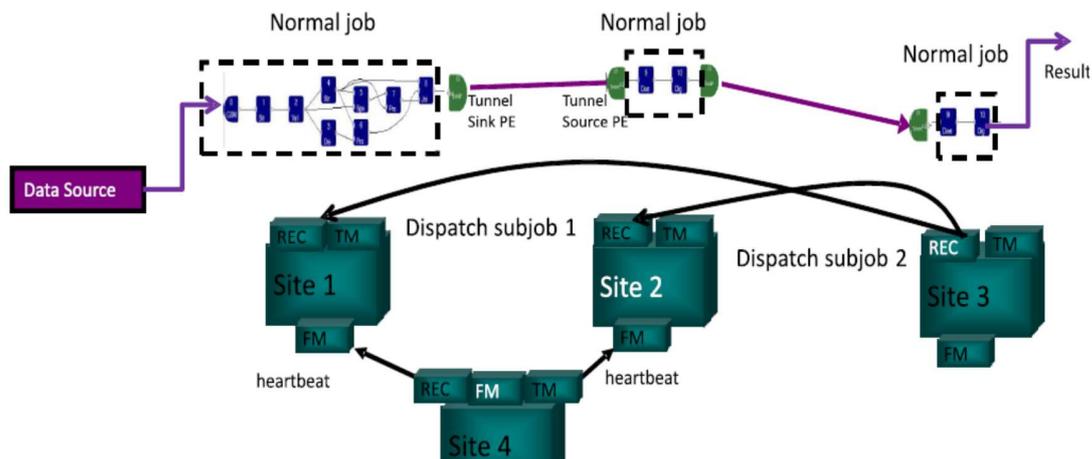


Figure 1 Exécution d'une tâche distribuée consistant en trois sous-tâches [21]. Le propriétaire du Site 3 exécute une sous-tâche, et dispatche deux sous-tâches au Site 1 et 2 pour exécution. Site 4 supervise Sites 1 et 2.

Pour chaque ressource, il peut y avoir un ou plusieurs fournisseurs et le site devra alors négocier avec eux et choisir le partenaire avec qui il conclura des accords. Le choix du partenaire avec qui le contrat sera conclu dépendra des situations de négociation de chaque ressource

demandée, où chaque ressource peut être en négociation avec plusieurs fournisseurs. De plus, il devra considérer le prix total qu'il doit payer pour l'ensemble des ressources requises par son plan.

1.2.2. Global Environment for Network Innovations (GENI)

Vu d'un angle d'allocation de ressources, le cloud computing permet aux consommateurs de louer de façon programmable de multiples ressources accessibles à travers l'internet. Cela est rendu possible grâce à la technologie de virtualisation du Cloud, qui permet de faire partitionner une ressource physique en plusieurs ressources virtuelles isolées l'une de l'autre selon les demandes de l'utilisateur. Ces ressources peuvent être de différent type tel que les machines, les outils de stockage, les détecteurs, et les liaisons de réseaux.

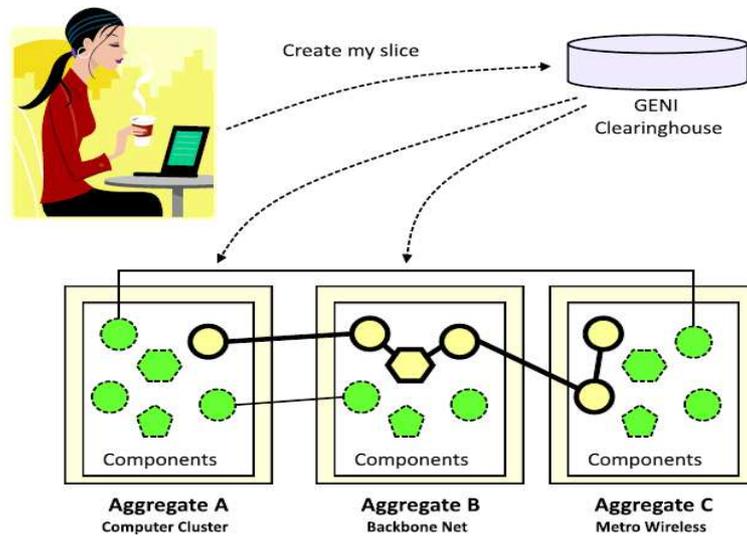


Figure 2 Partage de ressource par le biais d'un consommateur et un fournisseur [3]

En effet, le Cloud computing n'applique pas seulement le schéma « pay-as-you-go » (payer seulement ce que vous avez utilisé), mais aussi fournit les exigences propres à vos besoins. On peut citer comme exemple commerciaux du Cloud computing : Elastic Compute Cloud (EC2) et Elastic Block Store (EBS). Une solution académique du cloud computing nommé GENI « the Global Environment for Network Innovations [3] » a été initié par NSF (National Science Foundation) et disponible sans frais pour des fins académiques et recherches scientifiques.

GENI vise à fournir une plateforme hautement programmable, où par exemple on peut décider sur la façon sur laquelle les serveurs seront distribués, virtualisable afin que le consommateur peut réserver un espace virtuelle (slice) à travers les multiples composantes hétérogènes de la

plateforme. De plus GENI est un banc d'essai fédéré ce qui signifie que les ressources dans GENI sont détenues et exploitées par différentes organisations.

Le système de réservation de GENI permet aux consommateurs de louer des ressources à partir d'une large gamme. La collection de ressources réserver par le consommateur peut former un tout cohérent où elles peuvent se mettre en interaction pour l'accomplissement d'une tâche donnée. Par exemple, la Figure 2 d'au-dessus montre un consommateur réservant une collection de ressources composée de trois types de ressources : une grappe de calcul, un réseau de base et un réseau sans fils métropolitain. Cette collection de ressources est empaquetée dans un slice afin de l'isolée des autres slices et éviter tout mal fonctionnements. Une fois que le slice est établi, l'utilisateur peut faire tourner ces expérimentations sans se soucier de ce qui peut se passer dans les autres slices.

Dans la pratique, la plateforme GENI sera sollicitée par plusieurs consommateurs et fournisseurs comme cela est illustré dans la Figure 3 ci-dessous. L'entité fournisseur peut représenter un magasin, une compagnie privée ou même une nation et cela est vrai pour les consommateurs.

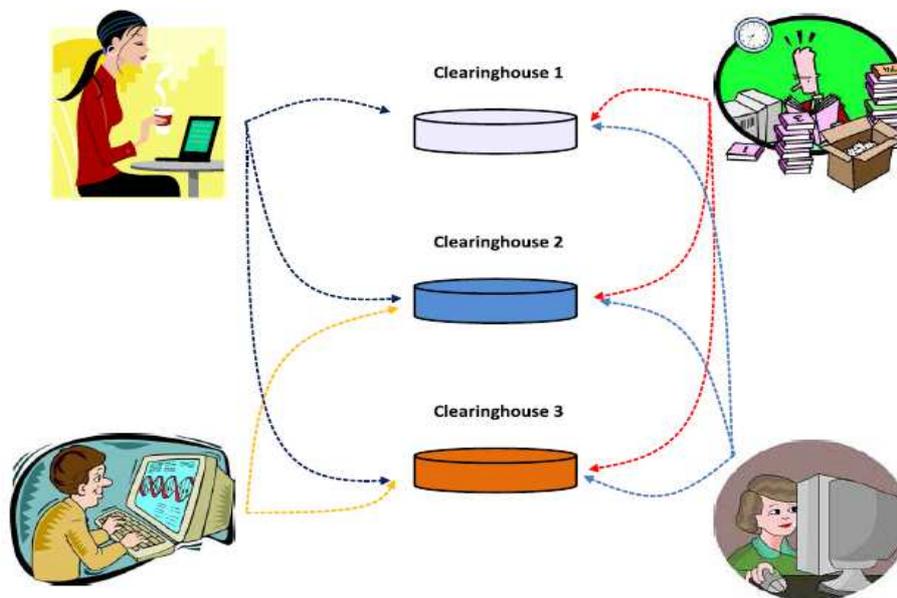


Figure 3 Partage de ressource par le biais de plusieurs consommateurs et fournisseurs [3]

Puisque la plateforme GENI accepte que plusieurs fournisseurs opèrent en même temps, le consommateur aura le choix de choisir celui le moins disant. Il est aussi possible que le

consommateur doit faire recourt à plusieurs fournisseurs afin de satisfaire ces besoins dans le cas où aucun fournisseur à lui seul peut satisfaire toutes ces demandes. Une coordination entre les multiples fournisseurs doit être établit afin de se conformer aux contraintes du consommateur qui demande parfois que les ressources doivent être fournies au même moment. Chaque consommateur qui parvient à réaliser sa tâche avec succès générera une certaine utilité qui exprime ses objectifs et ses préférences.

Le problème d'allocation de ressources intervient dans l'application GENI, puisque les ressources allouées par ce dernier sont limitées. De plus chaque agent que ce soit le consommateur ou le fournisseur agit rationnellement et pour son propre compte. Ainsi, une régulation sur le comportement des consommateurs et fournisseurs est nécessaire par l'établissement de contrat qui a comme principale élément le prix de vente. La régulation peut se faire à travers la négociation automatisée où se seront des agents (programmes) qui travailleront pour le compte des acheteurs ou vendeurs.

Les caractéristiques du problème de négociation décrit par les deux exemples cités auparavant sont extraites de [12] :

- Chaque agent est restreint dans sa négociation par une date limite. L'agent ne peut négocier au-delà de cette date limite et doit parfois faire de plus en plus de concession chaque fois qu'il voit que sa date se rapproche.
- Les agents sont non-coopératifs et d'intérêts personnels.
- Les agents ont des informations incomplètes au sujet des autres agents. Par exemple, un fournisseur de ressource ne connaît pas le prix de réserve d'un autre site, c'est-à-dire, le prix maximal que ce site est disposé à payer. De plus, puisque ces ressources s'allouent pour une location non pas pour un achat, une dynamicité émergera où les consommateurs entrent dans le marché dès qu'ils génèrent un plan et sortent dès que la durée d'utilisation de ressources demandées s'est écoulée.
- Plusieurs fournisseurs peuvent mettre en vente la même ressource, ce qui permet au consommateur de lancer des négociations simultanées et contracter la ressource par le fournisseur le moins disant.

- La concurrence se fait aussi au niveau des consommateurs (la compétition du marché). Ce qui indique qu'un agent doit prendre en compte la situation du marché lors de son processus de prise de décision.
- L'acheteur ne peut exécuter sa tâche que lorsque l'ensemble des ressources demandées soit satisfait. Il devra alors éviter l'effondrement de sa négociation en consultant la situation de négociation avec chaque fournisseur.
- Dans le cadre de ce travail, nous avons adopté le désengagement dans le processus de négociation. Dans ce contexte, l'acheteur fera face à de nouveaux problèmes de négociations qui consistent à décider sur le nombre d'accords qui devra contracter afin qu'il puisse faire face à l'incertitude du désengagement des vendeurs envers ces accords. Dans certaines situations, il est important de négocier à la fois le prix de l'offre et la pénalité de désengagement.

1.3 Les contributions principales

Le travail décrit dans cette thèse fait un certain nombre de contributions à l'état de l'art dans le domaine de la négociation méditée par l'agent en proposant des approches de négociations plus complexifiées vues d'un angle heuristique.

Les principales contributions de ce travail peuvent être résumées comme suit étant donné les caractéristiques du problème décrit au-dessus :

- Notre première proposition consistait à étendre l'approche de [16] en intégrant la confiance afin de rendre le mécanisme plus équitable en termes d'indemnisation des agents victimes du désengagement d'accords.
- La deuxième contribution consistait à explorer de façon plus judicieuse l'espace d'accords de l'acheteur afin d'accroître ces chances de décrocher des contrats et d'exécuter sa tâche.
- Finalement, notre troisième contribution consistait à élaborer une stratégie vendeur pour trouver le meilleur rapport revenu/complexité-computationnelle de la stratégie du vendeur.

1.4 Organisation de la thèse

La thèse se structure en cinq parties. La première partie évoque l'introduction générale, où on trouve deux chapitres, ce premier chapitre et le chapitre 2 qui fournit une introduction du Cloud computing. La deuxième partie contient seulement le chapitre 3 qui est consacré à l'allocation de ressources dans les systèmes multi-agents et présente certains concepts essentiels de l'économie du bien-être avant d'énumérer certaines instances réelles d'allocation de ressources dans les systèmes multi-agents où ces concepts seront utilisés comme mesure de performances. Ensuite, la partie 3 qui est la plus documentée, porte sur les modèles centralisé et décentralisé d'allocation de ressources. Cette dernière est composé de deux chapitres : le premier sur les enchères et le deuxième sur la négociation automatisée qui se distingue principalement des enchères par le fait qu'on peut mener les ventes sans avoir à intégrer un commissaire-priseur. Puis, les trois contributions que nous avons apporté dans le domaine d'allocation de ressources via la négociation automatisée sont présentées dans la quatrième partie. Ces contributions se trouvent dans les chapitres 6,7 et 8. Enfin nous terminerons cette thèse avec une conclusion qui représente la dernière et cinquième partie et qui résume les contributions apportées et offre des orientations futures de nos recherches. A la fin de cette thèse, nous avons jugé utile de présenter les fonctionnalités du simulateur sur lequel nous avons mené nos simulations pour évaluer les différentes approches de négociations.

Chapitre 2 Présentation du Cloud Computing

2.1 Introduction

Le Cloud computing a permis de réaliser un rêve de longue date qui était de transformer l'usage des ressources computationnelles sous-forme d'utilité comme c'est le cas pour l'électricité. Les développeurs ayant des idées innovatrices peuvent lancer leurs commerces sans avoir recours à un large capital d'investissement au préalable. Ils n'auront pas à se préoccuper du surapprovisionnement d'un service si ce dernier est en sous-utilisation par rapport aux prévisions établit au préalable et ils n'auront pas aussi à se préoccuper du scénario inverse. Néanmoins, la confusion demeure sur exactement ce que le Cloud est et quand il est utile. En effet, il existe de nombreuses définitions et interprétations du cloud computing qui sont consultables à partir de différentes sources.

2.2 Définition du Cloud computing

Cloud computing, souvent appelé simplement « le cloud », est l'approvisionnement sous forme de service à la demande des ressources informatiques passant par les applications aux centres de données via l'internet. Le paiement dans le modèle du cloud se fait au prorata de l'utilisation des ressources par le consommateur, ce modèle de paiement est appelé "pay-as-you-go".

Dans la littérature, on retrouve plusieurs définitions qui sont basées à partir de différents aspects du cloud, mais il n'y a pas eu encore de consensus.

2.3 Modèle de déploiement du Cloud

Il existe trois modèles de déploiement du Cloud communément utilisé : privé, public, et hybride. Un modèle additionnel est le Cloud de communauté, qui est moins répandu par rapport aux trois modèles précédemment cités.

- **Cloud privé** a une portée délimitée par l'organisation où il est implémenté.
- **Cloud public** est un ensemble de ressources fournit par des organisations tierces. Parmi les plus populaires on peut citer : Amazon Web Services, Google AppEngine, et Microsoft Azure.

- **Cloud hybride** est un ensemble de ressources computationnels fournis par des clouds publics et privés.
- **Cloud de communauté** partage les ressources computationnelles à travers plusieurs organisations appartenant à une communauté spécifique.

2.4 Grid computing vs Cloud computing

De nombreux experts diront que le cloud computing dérive du grid computing. Cependant se sont deux implémentations qui ont certes des similitudes mais aussi des différences. Parmi ces différences, il y a le fait que les grids sont déployés historiquement pour les grands travaux de calcul et ne sont pas originellement destinés pour être un modèle de service à usage utilitaire comme c'est le cas pour le cloud computing.

2.5 Service du Cloud

Le modèle d'approvisionnement du cloud se fait sous-forme de service. Le service peut prendre la forme de logiciel, de plateforme ou d'infrastructure.

2.5.1 Software as a Service

Software as a Service (SaaS) est un modèle d'approvisionnement d'application où les applications sont hébergées au sein du vendeur ou d'un fournisseur de service. L'accès à de telles applications se fait à travers un réseau généralement l'internet. Grâce à ce modèle, l'utilisateur n'a pas à se préoccuper de l'ensemble des éléments essentiels pour l'exécution de l'application demandée : Infrastructure sous-jacente, système d'exploitation, etc.

2.5.2 Platform as a Service

Platform as a Service (PaaS) est un modèle du cloud computing. Dans le modèle de PaaS, le fournisseur délivre tout ce qui est nécessaire afin que l'utilisateur puisse développer l'application désirée. Dans ce modèle, l'utilisateur a le contrôle sur l'application ainsi que l'environnement de développement de l'application, mais tout ce qui est attribué à l'infrastructure sous-jacente est géré par le fournisseur.

2.5.3 Infrastructure as a Service

Infrastructure as a Service (IaaS) fournit des ressources virtualisées, accessible à travers l'internet. Les ressources fournies sont virtualisées dans le sens où ces ressources sont une

agrégation de plusieurs ressources fournit par plusieurs centres de données qui peuvent être géographiquement distribués. Grâce donc à la technologie de virtualisation, l'utilisateur aura l'impression qu'il a en possession un bloc de ressources, hors en réalité ces ressources peuvent être une agrégation de ressources distribuées géographiquement.

2.6 Les caractéristiques du Cloud

Dans cette section, nous allons présenter les aspects techniques, qualitatifs et économiques du cloud computing.

2.6.1 Les aspects techniques

Dans cette sous-section nous allons présenter les aspects techniques du cloud.

- **La virtualisation** et le Cloud computing vont de pair pour fournir différents types de services. Le principe de la virtualisation est de diviser une pièce matérielle physique en plusieurs segments. Ces segments fonctionnent indépendamment bien qu'ils partagent la même ressource physique. Par exemple, la partition d'un processeur de 3 GHz ne peut être divisée de façon physique pour produire deux sous-processeurs opérant indépendamment. La solution est alors d'utiliser la virtualisation qui permet dans ce cas de faire partager le processeur en deux sous-processeurs virtuels. Ces sous-processeurs virtuels opèrent indépendamment mais fonctionnent à la base sur un processeur physique commun. De l'autre côté, la virtualisation peut être utilisée pour fusionner une ou plusieurs ressources physiques et leur donner l'apparence d'une seule ressource. Cela a été utilisé dans le concept du grid computing où plusieurs ressources inactives du parc informatique ont été fusionnées virtuellement afin de former un super ordinateur.
- **Multi-tenant** L'architecture Multi-tenant d'une application est une architecture où une seule instance est partagée par plusieurs utilisateurs. Par contre, une architecture multi-instance est une architecture où chaque client possède sa propre instance. Les bénéfices d'une architecture Multi-tenant sur multi-instances se traduit par une meilleure efficacité de la maintenance de l'application car le développeur de l'application pourra mettre à niveau son application une seule fois au lieu de la faire pour chaque instance comme c'est le cas pour une architecture multi-instance. Le mode de consommation en mode multi-tenant est considéré comme économique vu que les coûts de développement et de maintenance sont partagés par les clients utilisant l'application. Le cloud computing rend

plus efficace le modèle multi-tenant car le cloud grâce à la technologie de virtualisation permet aux utilisateurs utilisant la même instance d'application de contrôler leurs données situées sur la même base de donnée de manière indépendante.

- **La sécurité**¹ est l'une des préoccupations du Cloud computing non seulement pour le fournisseur mais aussi pour le consommateur. Le fournisseur doit s'assurer que les données de ses clients ainsi que leurs applications sont protégées. Le consommateur de son côté doit sécuriser ses données à travers des mots de passes et des mesures d'authentification élaborées. La responsabilité sécuritaire concernant le fournisseur est accrue car l'accès physique aux serveurs hébergeant les données de l'entreprise cliente sont situées chez le fournisseur et non pas implanté physiquement chez l'entreprise cliente. Le fournisseur du cloud doit donc doubler de vigilance et prendre les mesures sécuritaires nécessaire afin d'éviter des attaques provenant des employés même du fournisseur.
- **L'environnement de programmation** Il devrait être capable d'adresser des préoccupations comme les multiples domaines administratifs, les grandes variations de l'hétérogénéité des ressources, la stabilité des performances, la gestion des exceptions dans des environnements très dynamiques, etc [106].

Grâce aux API prédéfinies, les utilisateurs peuvent accéder, configurer et programmer les applications du Cloud.

2.6.2 Les aspects qualitatifs

Parmi les caractéristiques qualitatives du Cloud qui se réfèrent aux propriétés du modèle du cloud et non pas aux exigences technologiques de ce dernier. Cette sous-section a été inspirée de l'article [106].

- **Élasticité** dans le Cloud peut être comparée à la propriété physique de l'élastique, c'est-à-dire sa capacité à s'élargir puis à reprendre sa forme originale. Avec cette propriété, on peut satisfaire les besoins de l'utilisateur de manière dynamique en lui fournissant les ressources requises qui peuvent varier à travers le temps.
- **La disponibilité** fait référence à une capacité pertinente qui répond aux exigences spécifiques des services de la sous-traitance (externalisation). Les indicateurs de qualité de

¹ La propriété sécuritaire du Cloud a été inspiré de l'article Wikipédia [9]

service comme le temps de réponse et le débit doivent être garantis, de sorte à satisfaire les garanties de qualités avancées aux utilisateurs du cloud computing.

- **La fiabilité** représente la capacité d'assurer le fonctionnement du système de manière constante sans interruption. Grâce à l'utilisation des sites redondants, la possibilité de perdre les données et le code est diminuée de façon spectaculaire. Ainsi le cloud computing est adapté pour la continuité des activités et de la restauration après sinistre.
- **L'agilité** est une exigence de base pour le cloud computing. Les fournisseurs du cloud sont capables de réagir en ligne à l'évolution de la demande des ressources et des conditions environnementales.

2.6.3 L'aspect économique

Parmi les caractéristiques du cloud computing, on retrouve le modèle économique « pay-as-you-go ».

Le modèle **Pay-as-you-go** suit le modèle économique de la facturation des services publics comme l'électricité et le gaz. Vous payez seulement la quantité que vous avez utilisée ainsi que sa durée correspondante. Avant la venue de ce modèle économique, les entreprises devaient d'abord estimer la charge de travail maximale qui doit être satisfaite avant de lancer leurs projets. Cela n'est plus requis par le modèle « pay-as-you-go » puisque le consommateur va payer seulement les ressources utilisées et les ressources peuvent s'agrandir grâce à l'élasticité que le cloud possède comme caractéristique.

2.7 Projets du Cloud

Plusieurs projets du Cloud ont été lancés par les compagnies commerciales telles que Amazon, Google et Microsoft ainsi que par des institutions académiques dans le but de promouvoir le Cloud computing.

2.7.1 Produit commercial

On citera dans cette sous-section certain projets commerciaux ainsi que certains projets de recherche portant sur le cloud computing.

Amazon EC2 [4] est un service web qui fournit des capacités de calcul redimensionnable, accessible et configurable à travers le web. Amazon EC2 réduit le temps requis pour lancer de nouvelles instances serveurs rendant ainsi plus rapide le processus de mise en échelle des

applications web. Le service fournit aussi les outils nécessaires pour bâtir des applications robustes aux pannes. Amazon permet à ces utilisateurs de créer une machine virtuelle nommée instance via "Amazon Machine Image". Ces instances ou machine virtuelles sont classées en 6 catégories : Standard, micro, haute-mémoire, haut-CPU, cluster-GPU et cluster compute.

Google App Engine (GAE) [6] sortit en 2008 en version « preview » et ce n'est qu'en 2011 qu'une version finale a été créée. GAE est écrit en langage python, java, Go et PHP. Les applications s'exécutant dans GAE sont mises dans une sandbox et exécuté à travers plusieurs serveurs. GAE est un service de Google gratuit pour une certaine ressource et devient payant si des ressources supplémentaires viennent à être consommées.

MapReduce [29] est un modèle de programmation et une implémentation associés pour le traitement et la génération de large ensemble de données. Les utilisateurs spécifient une fonction "map" qui traite des paires clé/valeur pour générer un ensemble de paires intermédiaires clé/valeur, et une fonction "reduce" qui fusionne toutes les valeurs associées ayant la même clé intermédiaire. En résumé, MapReduce est le produit de deux opérations : l'opération « map » qui est une opération de filtrage et triage et l'opération « reduce » qui est une opération d'agrégation. Les programmes écrits dans ce style fonctionnel sont automatiquement parallélisés et exécutés sur un large cluster de machines. Le système d'exécution prend en compte les détails du partitionnement des données d'entrée, la planification de l'exécution du programme à travers un ensemble de machine, le traitement des défaillances de la machine et la gestion de la communication inter-machine, nécessaire. Cela permettra aux programmeurs ayant aucune expérience avec les systèmes distribués et parallèles de facilement utilisés les ressources d'un large système distribué.

2.7.2 Projets de recherche

Des projets universitaires ont eu lieu afin de contribuer dans la promotion du Cloud computing. Dans cette sous-section on citera certains projets les plus actifs en Europe et en Amérique du Nord.

XtreemOS [8] XtreemOS est un système d'exploitation basé sur Linux. La principale particularité du XtreemOS est qu'il fournit aux Grids ce qu'un système d'exploitation traditionnel offre à un ordinateur personnel : la transparence du matériel et le partage de ressources sécurisé entre les différents utilisateurs. Il simplifie ainsi le travail des utilisateurs en leur donnant l'illusion d'utiliser un ordinateur traditionnel tout en éliminant le fardeau de la gestion des ressources complexe dans

un environnement de grid typique. Lorsqu'un utilisateur exécute une application sur XtremOS, le système d'exploitation détecte automatiquement toutes les ressources nécessaires à l'exécution, configure les informations d'identification de l'utilisateur sur les ressources sélectionnées et lance l'application. Il fournit trois services qui sont : la gestion de l'exécution de l'application, la gestion des données et la gestion de l'organisation virtuelle.

Le projet était coordonné par l'INRIA (Rennes) et regroupait 19 partenaires industriels et académiques d'Europe et de Chine

OpenNebula [7] offre une solution flexible et riche en caractéristiques pour construire et gérer les cloud des entreprises et les centres de données virtualisés. OpenNebula est conçue pour être simple. Les administrateurs peuvent l'installer, le mettre à jour et le gérer simplement et les utilisateurs peuvent facilement l'utiliser. OpenNebula fonctionne avec MySQL, Ceph, LVM, GlusterFS, Open vSwitch, Ceph, LDAP ... Cela permet de délivrer un gestionnaire de cloud léger, flexible et robuste. Il a été créé par l'Université Complutense de Madrid en 2005, et a sorti son premier logiciel en 2008.

FutureGrid [10] c'est un projet coopératif qui a débuté en 2010. Il a été mené par l'Université d'Indiana et financé par la National Science Foundation (NSF) afin de développer un grid de haute performance qui permettra aux scientifiques à développer collaborativement et tester des approches innovatrices sur les grid et les cloud computing. FutureGrid fournira l'infrastructure aux chercheurs pour leur permettre d'effectuer leurs propres expériences de calcul en utilisant des systèmes distribués. Le but est de rendre plus facile aux scientifiques de mener de telles expériences d'une manière transparente. Les utilisateurs FutureGrid seront en mesure de déployer leurs propres configurations matérielles et logicielles sur un cloud public/privé, et exécuter leurs expériences. Ils seront en mesure de sauvegarder leurs configurations et exécuter leurs expériences en utilisant les outils à leurs dispositions. Le banc d'essai FutureGrid est composé d'un réseau à grande vitesse, reliant des clusters distribués composés d'ordinateurs de haute performance. FutureGrid emploiera la technologie de virtualisation qui permettra au banc d'essai de soutenir un large éventail de systèmes d'exploitation.

2.8 Comparaison générale des simulateurs de Cloud computing

Dans le tableau 1, un état comparatif de trois simulateurs est représenté : MDCSim [66], CloudSim [2] et GreenCloud [57]. Il montre les différents paramètres et sur la base de ces paramètres se caractérisent les différents simulateurs [57]. Cette section s'appuie sur l'article [88].

Tableau 1 Comparatif des outils de Simulation du Cloud

Paramètres	MDCSim [66]	CloudSim [2]	GreenCloud [57]
Communication Réseau	Limité	Limité	Complet
Support graphique	Aucun	Limité (Cloud Analyst)	Limité(NewtworkAnimator)
Disponibilité	Commerciale	OpenSource	OpenSource
Plateforme	CSIM	SimJava	NS2
Modèle d'application	Calcul	Calcul et transfert de donné	Calcul, transfert de donné et date limite d'exécution
Temps de Simulation	Secondes	Secondes	Dizaine de minutes
Language/Script	C++/Java	Java	C++/OTcl
Modèle physique	Aucun	Aucun	Disponible en utilisant plug in
Modèle d'énergie	Approximatif	Aucun	Précis (serveurs + réseaux)
Support de TCP/IP	Aucun	Aucun	Complet
Modèle économique d'énergie	Aucun	Aucun	DVFS, DNS et à la fois

1. **Réseau de communication** - Cloudsim et MDCSim [66] exécutent un modèle de communication limitée, qui fait ces calculs généralement sur la base du retard de transmission et de la bande passante. Le réseau de communication supporté par le simulateur Green Cloud est complet car il permet de conquérir les dynamiques des protocoles de communication réseau largement utilisés tel que TCP, UDP, IP, etc.
2. **Support graphique** - Fondamentalement, il n'y a pas de simulateur parmi ceux supervisée dans le tableau 1 implémentant un GUI avancé. Le GreenCloud peut être activé pour produire des fichiers de trace reconnus qui permettent de visualiser une topologie simulée et un flux de paquets après que la simulation a été terminée. Cependant, aucun outil GUI est disponible pour paramétrer un réglage de simulation ou afficher des graphes de

simulation d'une manière conviviale. Dans le cas du simulateur CloudSim, un outil externe CloudAnalyst [110] a été développé à cet effet. Il visualise seulement les paramètres de simulation de haut-niveau ciblant les applications de Cloud à échelle globale. Le simulateur MDCSim ne fournit aucune interface graphique.

3. **Disponibilité** – Le simulateur MDCSim n'est pas disponible actuellement pour le public, tandis que les simulateurs CloudSim et Green Cloud sont publiés sous la licence open source GPL.
4. **Plate-forme**- Les simulateurs MDCSim et CloudSim sont des simulateurs basé évènements. Le simulateur Green Cloud est en fait conçu comme une extension du simulateur NS2 qui est codé en C++ avec une couche supplémentaire OTcl implémenté au-dessus.
5. **Modèles d'application** - Les trois simulateurs implémentent des modèles d'applications utilisateur comme des objets simples décrivant les exigences de calcul pour l'application. En outre, GreenCloud et CloudSim précisent les exigences de communication des applications en termes de la quantité de données à transférer, avant et après l'achèvement de la tâche. Le modèle d'application implémenté dans le CloudSim s'accorde bien avec le calcul haute performance (HPC). Les tâches HPC, étant à calcul intensif et n'ayant pas de délai de réalisation spécifique sont des applications typiques pour les réseaux grid. Dans le cloud computing, les exigences de qualité de service pour l'exécution des requêtes utilisateur sont définies dans la SLA. Par conséquent, GreenCloud étend le modèle d'application utilisateur par l'ajout d'un délai d'exécution prédéfinie.
6. **Temps de Simulation** - Le temps nécessaire à la simulation dépend de nombreux facteurs comme le scénario simulé ou le matériel utilisé pour l'exécution du logiciel de simulation. En général, CloudSim et MDCSim, étant des simulateurs à base d'événements, sont plus rapides et s'étendent à un plus grand nombre de nœuds de centres de données. Néanmoins, le simulateur GreenCloud réalise encore des temps de simulation raisonnables. Il est de l'ordre de plusieurs dizaines de minutes pour une heure de temps de simulation lorsqu'il simule un centre de données typique avec quelques milliers de nœuds. En dehors du nombre de nœuds, la durée de la simulation est fortement influencée par le nombre de paquets de communications produits, ainsi que le nombre de fois où ils sont traités par des routeurs de réseau pendant leurs transferts. En conséquence, un datacenter typique simulé dans

GreenCloud peut être composé de milliers de nœuds tandis que le CloudSim et MDCSim Java peuvent simuler des millions d'ordinateurs.

7. **Language/Script** - Le langage/script des simulateurs qui signifie la plateforme dans laquelle les simulateurs vont effectuer leurs implémentations.
8. **Modèle physique** - Il n'y a pas de prise en charge direct pour simuler les processus physiques dans CloudSim et MDCSim. Pour les simulateurs Green Cloud, les modèles physiques utilisent les plug-ins disponible.
9. **Modèle d'énergie** – Le MDCSim effectue l'estimation approximative de tous les modèles énergétiques et le simulateur CloudSim ne tient pas compte de l'énergie dépensée dans les modèles. Le GreenCloud Simulator donne une estimation précise à l'aide des serveurs et des réseaux.
10. **Support TCP/IP** - Le MDCSim et le simulateur CloudSim ne supporte pas TCP/IP alors que le simulateur Green Cloud prend intégralement en charge le TCP/IP.
11. **Modes économes d'énergie** - Les simulateurs MDCSim et CloudSim ne prennent pas en charge tous les types de modes d'économie d'énergie alors que le simulateur Green Cloud prend en charge trois modes d'économie d'énergie à savoir DVFS, DNS et même DVFS+DN.



PARTIE 2 : ALLOCATION DE
RESSOURCES

Chapitre 3 L'allocation de ressources : une Cartographie²

3.1 Introduction

L'allocation de ressources est utilisée pour assigner les ressources disponibles de manière économique. Dans le domaine de gestion de projet, l'allocation de ressources est la planification des activités ainsi que les ressources demandées par ces activités tout en satisfaisant les contraintes qui y sont associées : la capacité limitée en ressources, le planning du projet ... Dans le cadre d'enchère combinatoire, l'allocation de ressources est la distribution des marchandises par le commissaire-priseur à plusieurs acheteurs selon leurs offres soumissent dans l'enchère. Dans d'autre domaine d'applications, l'allocation de ressources est un problème d'optimisation combinatoire tel que le problème de la patrouille multi-agent, où les ressources sont les régions à patrouiller et les agents sont les contrôleurs et le but du problème est de trouver la meilleure allocation des régions aux contrôleurs afin que l'ensemble des régions soit contrôlé le plus fréquemment possible.

Tous les problèmes cités auparavant ont été adressés de manière centralisée, c'est-à-dire que la charge computationnelle demandée pour adresser le problème était gérée par une entité centrale. Donc la résolution du problème d'allocation de ressources on y procédant par une approche centralisée repose sur une seule entité centrale. Cependant, en cas où la charge de calcul du problème devient très grande, l'entité centrale peut ne pas adresser le problème adéquatement, car l'approche centralisé n'est pas une approche scalable. Une approche scalable qui peut adresser le problème d'allocation de ressources où il y a plusieurs ressources à allouer et plusieurs agents est l'approche distribuée. Une approche distribuée distribue la charge computationnelle du problème à l'ensemble des agents au sein du système multi-agent au lieu de l'assigner à une seule entité centrale comme c'est le cas pour les approches centralisées. Plus précisément, la tâche de la prise de décision sera répartie à l'ensemble des agents.

Puisque nous nous intéressons dans cette thèse aux problèmes d'allocation de ressources distribuées, les décisions de négociations seront basées sur des critères locaux de rationalité.

² Ce chapitre s'appuie sur l'article Chevalere et al. "Issues in Multiagent Resource Allocation" [24].

Chaque agent via ces propres transactions effectuées va permettre à l'état globale de l'allocation de ressource de passer d'un état à un autre jusqu'à atteindre l'allocation finale où le processus du problème de l'allocation de ressources se termine. De plus les agents participant à la résolution du problème d'allocation de ressources sont considérés comme des agents d'intérêt personnels où les externalités ne sont pas prises en compte, c'est-à-dire que les agents ne se soucient pas de ce que les autres agents ont acquis.

L'évaluation de la qualité de l'allocation finale se fait via différentes mesures de performances. Parmi ces mesures on peut citer le bien-être social (somme des bien-être individuels), la complexité computationnelle de l'approche...

Dans ce chapitre nous allons donner un aperçu global de l'allocation de ressources dans les systèmes multi-agents et nous terminons ce chapitre en citant des instances réelles de l'allocation de ressource dans les systèmes multi-agents.

3.2 L'allocation de ressources : définition

Dans les systèmes multi-agent, la ressource signifie tout objet qui peut être acquis par l'agent via le processus de négociation. Chaque sous-ensemble possible de l'ensemble des ressources disponible aura un effet particulier sur l'utilité de l'agent selon les préférences de ce dernier. Une tentative de définition de l'allocation de ressources au sein des systèmes multi-agents pourrait être comme suit :

Définition : *Une allocation de ressource au sein d'un système multi-agents, est une partition particulière de ces ressources, de sorte que chaque partie de cette partition serait allouée à un agent.*

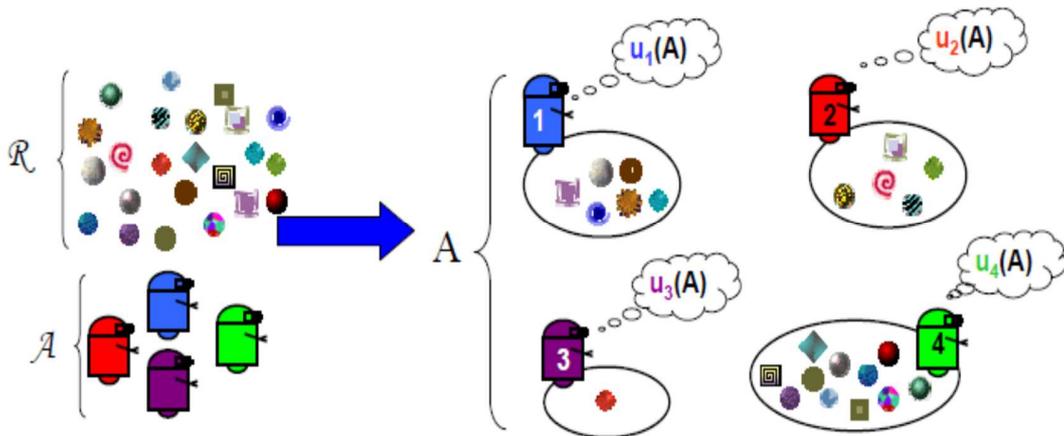


Figure 4 Allocation de ressources [38]

Comme on l'a dit dans l'introduction de ce chapitre, il existe deux grandes catégories pour adresser le problème d'allocation de ressources : soit de façon centralisée ou de façon distribuée. Dans les procédures centralisées, les décisions finales relatives à l'allocation de ressources sont entreprises par une entité centrale qui doit être impartial afin que les agents acceptent de collaborer avec cette dernière. D'autre part, une procédure décentralisée permet à l'ensemble des agents de participer à la prise de décision concernant l'allocation de ressources. Dans cette procédure, les décisions d'allocation de ressources se font à travers l'interaction des agents entre eux et non pas à travers une entité centrale qui raisonne à travers les informations des différents agents. Tout cela doit être fait en respectant les contraintes du problème qu'on doit adresser et parmi ces contraintes il y a les règles du protocole de négociation à respecter. Dans cette thèse, nos contributions sont des procédures décentralisées pour les avantages suivant qu'elles procurent [38] :

- La complexité de la négociation est répartie sur l'ensemble des agents du système.
- Evite la problématique de divulguer des informations privées à une entité centrale.
- Permet de laisser une certaine autonomie aux agents.
- Semble plus appropriée à certains types de problèmes, spécifiquement lors du passage à l'échelle.

Dans ce qui suit, nous allons porter notre attention uniquement aux procédures d'allocation de ressources décentralisées.

3.3 Les ressources

Cette section est dédiée aux différentes propriétés des ressources qui sont l'objet de négociation dans les problèmes d'allocation de ressources.

3.3.1 Continues vs Discrètes

La ressource peut être allouée sur une échelle dénombrable ou infiniment dénombrable si on a affaire à une ressource de type discret, comme par exemple une pomme où la plus petite unité indivisible lors du marchandage est une pomme elle-même. Par contre, une ressource continue est une ressource allouée sur une échelle non dénombrable comme par exemple l'énergie. Une représentation d'une allocation de ressources de type continue peut-être représentée par un vecteur de réel positifs (ou, alternativement, des numéros compris dans l'intervalle $[0,1]$ qui désignent la proportion d'une ressource particulière détenue par l'agent recevant le paquet de ressources). D'un autre côté, les paquets de ressources discrètes peuvent être représentées par un vecteur d'entiers non-négatifs ou par un vecteur binaire comprenant des valeurs $\{0, 1\}$ s'il y a au plus une ressource dans le marché pour chaque type de ressource. Les méthodes pour les problèmes d'allocation de ressources de type discret sont applicables aux types continus après discrétisation. La discrétisation d'une ressource continue se fait par la désignation de la plus petite unité que cette ressource peut être divisée lors de son marchandage. Par exemple, dans le cas où 10.000 litres de jus seront mis pour négociation, et après discrétisation de cette ressource, l'unité de cette ressource sera spécifiée à 50 litres, alors, ni l'acheteur, ni le vendeur ne peuvent se permettre de négocier une quantité de cette ressource qui n'est pas multiple de l'unité de cette dernière : 100 litres est accepté, 120 litres n'est pas accepté comme quantité à négocier.

3.3.2 Divisibles ou non

Le fait que la ressource soit discrète ou continue relève de la propriété inhérente de la ressource, ce qui n'est pas le cas pour la propriété de la divisibilité de la ressource : divisible ou indivisible. La spécification de cette propriété de la ressource est spécifiée au niveau du mécanisme d'allocation de ressource. Dans cette thèse, nous nous concentrons seulement sur les ressources indivisibles.

3.3.3 Partageables ou non

On dit qu'une ressource est partageable lorsqu'elle peut être partagée ou utilisée par plusieurs agents en même temps. Un exemple d'application est l'observation de la terre par satellite, où

l'image satellitaire est partagée par plusieurs agents. Le cas canonique, considère que la ressource est non partageable et nous nous conformons à ce principe dans le reste de cette thèse.

3.3.4 Statiques ou non

Une ressource est dite statique lorsque son statut reste inchangé. En général, les ressources ne peuvent pas être considérées comme statiques, mais dans les systèmes multi-agent elles sont supposées l'être. Les ressources non-statiques sont des ressources dont leurs statuts varient avec le temps. C'est des ressources consommable et périssable comme les carburants et la nourriture. Dans le reste de cette thèse nous nous conformons aux ressources statiques.

3.3.5 Mono-Item vs Multi-Items

Une configuration dites multi-items est une configuration où chaque type de ressource est fournie en un ou plusieurs exemplaires. Par contre dans une configuration mono-item, chaque type de ressource ne représente qu'une seule ressource. En fait, tout problème Multi-Item peut être transformé en un problème mono-item en catégorisant les ressources appartenant à la même catégorie (type de ressource). D'un autre côté, tout problème mono-item peut être transformé en un problème multi-item, car une configuration mono-item est un type de configuration multi-item dégénéré.

Dans la configuration multi-items, il est possible d'avoir plusieurs ressources de même type et de s'y référer en utilisant le même nom. Supposons, par exemple, qu'il existe un certain nombre de bouteilles de limonades disponibles dans le système. Dans une configuration multi-item, le concepteur va assigner l'ensemble des limonades à une ou plusieurs catégories, et donc les agents ne pourront distinguer individuellement chaque bouteille appartenant à la même catégorie. D'autre part, dans une configuration Mono-item chaque élément doit être distinguable. A cet effet, chaque bouteille doit porter un nom unique afin qu'elle soit identifiable. En optant pour une représentation multi-item on gagne en lisibilité et en compacité mais d'un autre côté, le langage devient plus riche car le domaine des variables sera étendu à un domaine qui prend des valeurs entières non négatives plutôt que des valeurs binaires pour le cas du Mono-Item, car le mono-item est une configuration multi-item où chaque catégorie ne peut contenir qu'un seul objet.

3.3.6 Ressources ou tâches

À un niveau suffisamment élevé d'abstraction, un problème d'allocation de tâches peut être réduit à un problème d'allocation de ressources. Toutefois, il faut prendre en compte que les tâches sont comme des ressources ayant une utilité négative vis-à-vis de l'agent. De plus, les tâches se distinguent des ressources par le fait que ces dernières sont couplées à des contraintes définissant les combinaisons cohérentes qui peuvent leur être autorisées. Comme par exemple l'exécution d'une tâche comme précondition pour l'exécution d'une autre tâche.

Dans cette thèse, cependant, nous nous concentrons sur les problèmes d'allocation de ressources en générale et non pas de tâches.

3.4 Outils : Les différentes mesures du bien-être social

L'économie du bien-être se définit par la problématique suivante : « Etant donné les bien-être individuels des membres d'une société, quelles seront les mesures à prendre pour évaluer l'efficacité du bien-être de la société dans sa globalité ? » [38].

Pour essayer d'adresser ce problème, nous allons présenter un aperçu de l'ensemble des notions du bien-être social et dont ils s'avéreront utile pour la conception de protocole d'allocation de ressources multi-agents.

Dans ce qui va suivre, on optera pour la notation suivante : \mathcal{A} l'ensemble des agents qui composent la société d'agents, \mathcal{R} l'ensemble des ressources en marchandage et u_i est la fonction d'utilité de l'agent i appartenant à \mathcal{A} .

3.4.1 Bien-être social Utilitaire

La mesure utilitariste du bien-être sociale dans une société multi-agent est la mesure la plus répandue et est considérée comme l'une des métriques les plus importantes dans l'évaluation de la qualité du bien-être de la société d'agents. Elle peut être exprimée par la formule suivante :

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

Où sw_u est le bien être social de la société d'agent, qui n'est autre que la somme de l'ensemble des utilités individuelles des agents. L'utilité de chaque agent est exprimée par la fonction d'utilité

$u_i(A)$ qui prend comme paramètre la répartition des ressources à l'ensemble des agents, noté par A .

3.4.2 Bien-être social égalitaire

La mesure du bien-être social égalitaire n'est autre que l'utilité de l'agent le plus pauvre de la société. Elle est exprimée par la formule suivante :

$$sw_e(A) = \min\{u_i(A) | i \in \mathcal{A}\}$$

Où $sw_e(A)$ est le bien-être social égalitaire. La maximisation du bien-être social égalitaire permet d'avoir une certaine équité au sein de la communauté d'agents.

3.4.3 Produit de Nash

Une autre mesure relative au bien-être d'une société est le produit de Nash. Ce dernier est un compromis entre les deux mesures précédentes : bien-être social utilitaire et égalitaire. Il est défini par la formule suivante :

$$sw_N(A) = \prod_{i \in \mathcal{A}} u_i(A)$$

Où $sw_N(A)$ est le produit de Nash pour l'allocation de ressources A . Une maximisation de ce dernier revient à avoir une équité plus forte entre les agents et un bien être sociale élevé.

3.4.4 Bien-être social élitiste

Une mesure qui est à l'opposé du bien-être social égalitaire se nomme : bien être sociale élitiste. Cette dernière représente l'agent ayant l'utilité la plus élevée au sein de la société. Elle est définie par la formule suivante :

$$sw_{el} = \max\{u_i(A) | i \in \mathcal{A}\}$$

Où sw_{el} représente le bien être social élitiste, qui est clairement une mesure non-équitable. Elle est pertinente dans des applications à caractère coopérative où le but est d'achever des objectifs qui peuvent être accomplis à l'aide d'un seul agent.

3.4.5 L'absence d'envie

Une allocation de ressources est dite dépourvue d'envie lorsque cette dernière possède la caractéristique suivante : chaque agent se sent au moins aussi satisfait avec sa part qu'il le serait avec n'importe quelle autre part appartenant à un autre agent.

De manière plus formelle, on dit qu'une allocation de ressources A est dépourvue d'envie si et seulement si pour chaque pair d'agents $i, j \in \mathcal{A}$, on a $u_i(A(i)) \geq u_i(A(j))$.

Comme on peut le constater, ce n'est pas toutes les allocations de ressources qui peuvent avoir une telle propriété. Cependant, si on ne peut pas trouver d'allocation de ressources ayant cette propriété, on peut toujours diminuer l'envie d'une allocation en lui définissant une mesure.

3.4.6 L'optimalité au sens de Pareto

L'optimalité de Pareto est une allocation de ressources dans laquelle il est impossible de changer l'allocation de sorte que chaque agent ait le même bien-être ou un meilleur bien-être qu'il eut avec l'allocation originale.

Soit $P \preceq_i G$ signifie que l'agent i a une préférence à l'allocation G qui est supérieur ou égale à celle de l'allocation de ressource P . On dit alors qu'une allocation P est *Pareto dominée* par une allocation G si les deux conditions suivantes sont réunies :

- $P \preceq_i G$ pour chaque agent $i \in \mathcal{A}$
- $P \prec_i G$ pour au moins un agent $i \in \mathcal{A}$

De plus on parle de Pareto optimal pour une allocation donnée, lorsque cette dernière n'est pas Pareto dominée par n'importe quelles autres allocations possibles.

3.5 Exemples d'applications

Il existe de nombreux exemples d'applications réelles qui peuvent être traduit sous forme de problème d'allocation de ressources au sein d'un système multi-agents, ce qui nous permettra de les adresser en se référant à l'ensemble de résultats déjà obtenu dans ce domaine. Après avoir traduit une instance d'application réelle en un problème d'allocation de ressources, en désignant entre autres quelles seront les objets qui seront traités comme des ressources et ceux qui seront traités comme des agents, il faut ensuite lui définir les objectifs les plus appropriés pour ce

problème tel que la maximisation du bien-être social utilitaire, égalitaire ou un compromis entre les deux.

3.5.1 Des applications à caractère égalitaire

Allocation de ressources satellitaires

Etant donné que le prix des satellites est exorbitant, l'acquisition de ce dernier s'opère par un cofinancement des différents organismes ou pays qui seront représentés par des agents dans la modélisation du problème d'allocation de ressources. Les objectifs à atteindre dans ce problème sont :

- Efficacité : Le satellite ne doit pas être sous-exploité
- Equité : Chaque agent doit avoir un retour sur investissement qui doit être proportionnel à son financement pour l'acquisition du satellite.

Le satellite est opéré par un centre de programmation et de traitement d'image comme le montre la Figure 5.

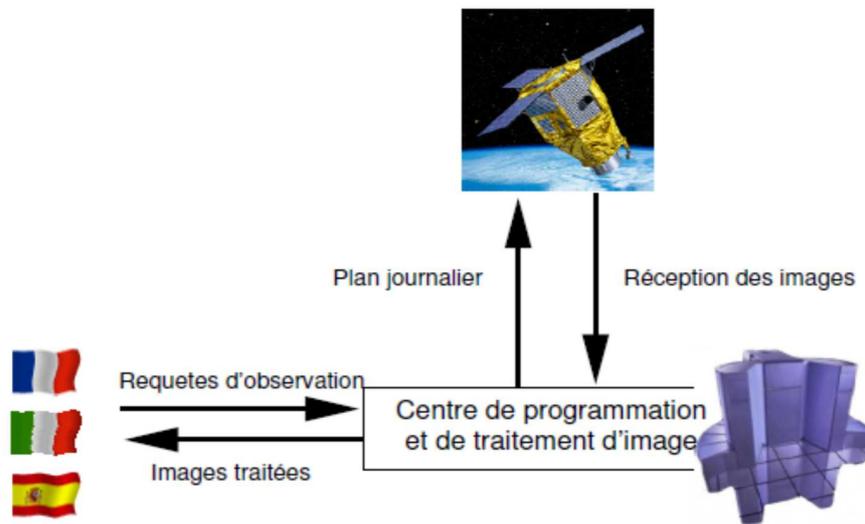


Figure 5 Allocation de ressources satellitaires [20]

On peut considérer ce problème comme un problème d'allocation d'un ensemble de ressources indivisible à certains agents qui ne peuvent interagir entre eux dans le but d'échange et de négociation des ressources car ce rôle est assigné à une entité centrale. C'est un problème typique de partage de ressources [24].

La patrouille multi-agents

Le problème de la patrouille est très répandu dans la communauté multi-agent. Le problème peut être formulé comme suit : étant donné un ensemble d'agents qui ne peut couvrir à la fois un territoire donné pour le surveiller, l'objectif de ce problème est de concevoir le plan le plus optimal qui permet aux agents d'émettre le moins de temps possible pour se déplacer d'un territoire à un autre afin que la surveillance de l'ensemble du territoire soit la plus efficace possible.

Récemment Almeida et al., [11], ont modélisé le problème de la patrouille en considérant les parties du territoire comme des ressources sous forme de nœud de graphe. Chaque agent, ayant son propre lot de ressources (zones) à patrouiller, aura comme objectif de trouver le chemin le plus court pour les parcourir. L'efficacité à laquelle l'agent remplit sa mission se reflétera sur son utilité. De plus, les agents sont autorisés à faire des échanges de nœud à travers une procédure de négociation dont le but n'est autre que d'améliorer leurs performances de patrouille et donc leurs bien-être utilitaires respectifs.

Grid Computing

Une étude menée par Dataquest [38] a constaté qu'il y a une sous-utilisation du parc informatique puisqu'une utilisation de l'ordre de «5 à 20% des capacités d'ordinateurs des entreprises est observée pendant les heures ouvrables et pratiquement zéro le reste du temps ».

Au lieu d'acheter un méga ordinateur pour mener à bien des tâches computationnelles lourdes, une alternative plus efficace se nomme *Grid Computing* qui est l'agrégation virtuelle des ressources inactives du parc informatique afin de former une machine ayant les capacités d'un méga ordinateur.

On peut faire le parallèle de ce problème avec le problème d'allocation de ressources. Les agents seront les ordinateurs du parc informatique et les ressources seront les ressources computationnelles inactives de ce parc.

L'objectif de ce problème est de trouver le meilleur compromis entre le bien-être utilitaire par la réalisation d'un maximum de calcul possible et le bien être égalitaire par le transfert de ressources de manière virtuelle d'un agent qui est en surplus à un agent nécessiteux à ces ressources.

3.6 Revue de littérature sur l'allocation de ressources dans le Cloud

Dans cette section, nous allons présenter les travaux qui sont relatives à l'allocation de ressources dans le Cloud. J.Ejarque [34] a présenté un processus d'allocation de ressources distribué qui se nomme SERA (Semantically Enhanced Resource Allocation), lequel combine les bénéfices du web sémantique pour rendre plus facile l'intégration entre plusieurs fournisseurs de ressources dans le Cloud, et les technologies d'agent pour la coordination et l'adaptation de l'exécution à travers les différents fournisseurs. Le processus d'allocation est basé sur la négociation des différents agents, lequel autorise la combinaison des politiques des fournisseurs et consommateurs pour obtenir des résultats de planification qui satisfassent les deux parties. G.Jang et K.M.Sim [55] partent du principe que les consommateurs et les data centers sont distribués géographiquement. Afin d'assurer une bonne qualité de service en termes de rapidité du temps d'allocation et du temps de réponse d'exécution, sa problématique s'est portée sur le choix du data center approprié au milieu des data centers qui sont distribués géographiquement afin de produire la meilleure qualité de service possible, celle-ci est basée sur la distance géographique qui sépare le consommateur du fournisseur et la charge de travail de ce dernier. L.Chimakurthiet et M.K.SD [25] proposent de s'appuyer sur le mécanisme de la colonie de fourmi afin de minimiser la consommation d'énergie tout en respectant le SLA (Service Level Agreement). Y.O.Yazir et.al [112] adressent le problème de la sous-utilisation et/ou la surutilisation des ressources en se basant sur une nouvelle approche de gestion de ressources qui est autonome et dynamique. La contribution de ce travail est double : Premièrement, l'adoption d'une architecture distribuée où la gestion d'une ressource est décomposée à des tâches indépendantes, chacune d'elle est effectuée à des Nœuds d'Agents Autonome (NAs) qui sont fortement couplés avec les machines physiques (PMs) dans un data center. Deuxièmement, les Nœuds d'Agents Autonomes exécutent des configurations en parallèle à travers l'Analyse de Décision avec des Critères Multiples. M.de Assuncao et.al [19] ont proposés de prouver le bénéfice apporté par les services d'un cluster distant et externe pour résister au période de pic de demande afin que la qualité de service reste inchangée, tout en générant du bénéfice à travers cette opération. Un algorithme de redirection qui est associé à différentes stratégies de planification a été implémenté pour dire quand il serait nécessaire d'emprunter les ressources du Cloud lorsque les ressources locales sont surchargées de travail. D.Sun et al. [105] ont présentés un nouvel algorithme nommé NECDA où le commissaire-priseur trie les offres dans l'ordre croissant et les demandes en ordre décroissant, où l'offre est la proposition d'achat de

l'acheteur, et la demande est la proposition de vente émise par le vendeur. Ensuite, l'offre la plus élevée sera liée à la demande la plus faible. S. Kraus et. al [58] proposent une troisième partie en plus des parties acheteur et vendeur, qui sera impliquée dans chaque transaction qui n'est autre que le courtier, ce qui permet au consommateur de ne plus avoir besoin de connaître l'identité de son fournisseur ni même de savoir où les ressources résident. C'est une approche d'allocation de ressources basée courtier dont le consommateur peut demander plus d'un type de ressource dans un environnement de Cloud fédéré.



PARTIE 3 : MODELES

CENTRALISE ET DECENTRALISE D'ALLOCATION DE RESSOURCES

Chapitre 4 Taxonomie de l'enchère

4.1 Théorie de l'enchère :

L'enchère est un type de mécanisme où le prix est déterminé par le processus de surenchère, ce qui n'est pas le cas avec la négociation où se sont les agents qui décident du prix de l'offre et de la pénalité de désengagement. Par l'émergence du e-commerce et les transactions électroniques à faible coût sur internet, un intérêt croissant sur le design de nouvelles enchères sur ces plateformes a été constaté. Plusieurs sites Web ont réussi dans le domaine des enchères via internet comme eBay. Aussi de nombreux chercheurs en informatique, en économie et la théorie des jeux ont présenté de nombreuses études intéressantes sur le sujet de l'e-commerce, e-négociations, des ventes aux enchères en ligne, et la théorie des enchères. Les défis que posent les mécanismes d'enchères sont considérés comme un large domaine.

McAfee et McMillan définissent l'enchère comme suit [111] : " An auction is a market institution with an explicit set of rules determining resource allocation and prices on the basis of bids from the market participants". Dans le cadre de ventes aux enchères, un mécanisme d'enchère peut être défini comme une fonction qui prend comme entrée l'ensemble des stratégies des participants, et en fonction de cet ensemble de stratégie retourne un assignement d'items pour chaque participant ainsi que le paiement correspondant. La fonction qui fait correspondre les stratégies des participants aux résultats de l'enchère est basée sur une fonction objective qui aura pour but de maximiser les revenus du vendeur ou l'acheteur si on a affaire à une enchère inversée, la maximisation de l'efficacité sociale de l'allocation ou tout autre objectif.

Comme le souligne Rothkopf et Park dans [89], la conception du marché est un effort multidisciplinaire composée de contributions de l'économie, la recherche opérationnelle, l'informatique, et bien d'autres disciplines. Selon Abrache J. et.al [52], la contribution de l'informatique réside surtout dans (a) le développement d'architectures et d'outils logiciels appropriés pour le déploiement de ventes aux enchères ; (b) la conception d'agents logiciels capables d'interagir de façon intelligente en étant soit coopératifs ou compétitifs ; et (c) la conception et la mise en œuvre de plates-formes de simulation pour l'évaluation des mécanismes d'enchères dans des environnements artificiels contrôlés.

De manière plus concise, les enchères permettent à plusieurs acheteurs de soumettre leurs propositions à un vendeur où à un commissaire-priseur travaillant pour le compte du vendeur qui décidera en conséquence. Les enchères inverses partagent le même principe que les enchères mais au lieu que la compétition se passe entre les acheteurs pour contracter des ressources, elle se passe entre vendeurs pour décrocher un contrat d'appel d'offre lancé par l'acheteur. Le commissaire-priseur fixera les règles de l'enchère et agira comme un arbitre. Le processus d'enchère ne s'arrête qu'après avoir atteint un prix d'équilibre, c'est-à-dire que les règles de l'enchère vont servir de régulateur à l'offre et la demande en se basant sur la compétition du marché.

Tableau 2 Taxonomie de la conception de l'enchère

#Vendeurs : #Acheteurs			
# Marchandises		1 : N ou M : 1	M : N
	Une	Enchère à un côté, une seule ressource (Section 4.1.1)	Enchère à deux côtés, une seule ressource (Section 4.1.2)
	Plusieurs	Combinatoire (Section 4.1.3)	

L'enchère est considérée comme un outil puissant pour résoudre le problème de demandes concurrentielles sur des ressources limitées en quantité qui ne peuvent satisfaire l'ensemble des demandes. Il faut noter qu'il y a pratiquement une infinité de variété de mécanisme d'enchères dans le monde. Cette thèse les classe comme le montre le **Error! Reference source not found.** à partir de deux points de vue : le nombre de bien mis en enchère et le nombre de participant des deux côtés : acheteur et vendeur.

4.1.1- Enchère à un côté pour une seule ressource

a. Enchère anglaise :

Puisque l'enchère anglaise est l'enchère la plus répandue dans les marchés, le lecteur peut la confondre avec le mot « enchère ». C'est une enchère où un vendeur met en enchère une marchandise et où plusieurs acheteurs sont intéressés par cette dernière. Les principes de l'enchère anglaise sont simples : 1) Le commissaire-priseur déclare un prix initiale, 2) Les acheteurs

surenchériront de manière croissante, c'est-à-dire que chaque acheteur doit surenchérir par une offre supérieure à l'offre dernièrement reçue par le commissaire-priseur, 3) Le commissaire-priseur arrête l'enchère lorsqu'il n'y a plus d'acheteur voulant soumettre une proposition plus élevée que celle dernièrement reçue par le commissaire-priseur. Le dernier enchérisseur sera alors l'acquéreur de la marchandise mise en enchère et doit payer le prix qu'il a annoncé, ce qui n'est pas toujours le cas dans tous les mécanismes d'enchères comme on va le montrer plus tard.

b. Enchère Japonaise :

La différence entre l'enchère japonaise par rapport à l'enchère anglaise est que c'est le commissaire-priseur qui fait grimper les prix et non pas les acheteurs. Le prix augmentera de manière continue et l'acheteur aura deux actions à choisir : soit rester dans l'enchère, soit se retirer et dans ce cas il n'est plus autorisé à retourner à l'enchère. Le critère d'arrêt de l'enchère intervient lorsque l'avant dernier participant décide de se retirer et le dernier participant sera alors déclaré vainqueur. Le prix de l'enchère s'arrête aussitôt que le processus d'enchère s'arrête, et le vainqueur acquerra l'objet au prix final.

c. Enchère Hollandaise :

Dans les enchères Hollandaises, le prix de l'enchère est descendant. C'est-à-dire, le commissaire-priseur déclare un prix maximal initial qui décroît tout au long de l'enchère. Le premier acheteur qui décide d'accepter le prix pour la marchandise mise en enchère est déclaré vainqueur et achète l'objet à ce prix-là. Si aucun acheteur ne se déclare avant que le prix d'enchère se décroît jusqu'à atteindre le prix de réserve du vendeur, alors l'enchère se terminera en ayant aucun acheteur pour la marchandise en question. L'enchère hollandaise est nommée pour son meilleur exemple d'utilisation qui est l'enchère de tulipes hollandaise.

d. Enchère à sous-pli cacheté :

Contrairement aux enchères citées précédemment, les enchères sous-pli cachetés comme leurs noms l'indiquent sont des enchères où les soumissions de chaque participant sont traitées comme information privées. En d'autres termes, chaque participant ne connaît pas le prix de soumission des autres participants et doit faire avec cette incomplétude d'information pour établir sa stratégie. Le déroulement de l'enchère scellé sera comme suit : 1) les acheteurs soumettent leurs enchères « scellées » au commissaire-priseur, 2) Le commissaire-priseur désigne comme vainqueur celui qui soumis l'offre la plus élevée après avoir évalué l'ensemble des enchères soumis.

Les variantes de l'enchère scellée se basent sur le prix que le vainqueur doit payer : a) dans les ventes aux enchères premier prix, le gagnant doit payer le prix qu'il a soumis, b) Les enchères deuxième prix, le gagnant doit payer la deuxième offre la plus élevée ou bien l'offre la plus élevée qui a été rejetée et c) en parle d'enchère au $k^{\text{ième}}$ prix, dans ce cas, le gagnant doit payer la $k^{\text{ième}}$ proposition de l'enchère.

e. Enchère inversée :

Toutes les enchères auparavant avaient un point en commun : Une enchère à un vendeur et plusieurs acheteurs. L'enchère inversée est tout simplement le contraire de cela : Une enchère à un acheteur et plusieurs vendeurs. Cela veut dire que la concurrence se passera du côté vendeur où plusieurs vendeurs se concurrenceront les uns avec les autres afin de satisfaire l'appel d'offre de l'acheteur et en tirer un profit. Comme les enchères, les enchères inverses suivent le même principe, les théories sur les enchères non-inverse peuvent être appliquées aux enchères inverses en inversant les signes de prix (un paiement négatif signifie un gain) et échangeant les mots : "vendeur" pour "acheteur", "plus haut" pour "plus bas" et ainsi de suite, sans perte de généralité.

4.1.2. Enchère à deux côtés pour une seule ressource

Les enchères à deux côtés se caractérisent par le fait que la concurrence est présente dans les deux côtés : acheteur et vendeur. L'enchère à deux côtés pour une seule ressource signifie qu'il y a plusieurs acheteurs et vendeurs où chaque vendeur peut vendre qu'un seul type de ressource. Le processus de fonctionnement d'une enchère à deux côtés est comme suit : Les acheteurs et les vendeurs soumettent leurs offres et demandes respectivement au commissaire-priseur. Après évaluation des offres et demandes, le commissaire-priseur fixe un prix p : tous les vendeurs demandant à vendre à un prix moins que p vont pouvoir effectivement vendre leurs marchandises, et tous les acheteurs dont leurs offres dépassent le prix p achèteront effectivement la marchandise pour laquelle ils étaient en négociation. Un exemple typique des enchères à deux-côtés sont les marchés boursiers.

On peut distinguer deux catégories d'enchère à deux-côtés pour une seule ressource : "the continuous double auction" (CDA) et "the periodic double auction" (PDA). Dans les deux enchères CDA et PDA, un participant passe des commandes à tout moment et autant de fois qu'il le veut et c'est le commissaire-priseur qui détermine le résultat de l'enchère. Le résultat de l'enchère comprend le prix de vente et la quantité que chaque vendeur va vendre à chaque acheteur.

La différence entre le CDA et le PDA réside sur le moment où le processus de correspondance entre les offres des acheteurs et les demandes des vendeurs se produit. Dans le CDA, il se produit dès qu'une nouvelle commande est enregistrée, par contre dans le PDA, il se produit à chaque période d'intervalle prédéfinie. Chaque commande formulée que ce soit dans le CDA ou le PDA, l'acheteur va exprimer le prix unitaire maximum pour l'achat de la ressource demandée ainsi que sa quantité, de son côté le vendeur va exprimer le prix unitaire minimale pour la vente d'une seule unité de ressource ainsi que le nombre qu'il puisse en vendre.

Dans les enchères à deux-côtés, le processus de correspondance effectué par le commissaire-priseur ne se limite pas à une correspondance d'un seul vendeur à un seul acheteur, mais peut prendre la forme d'un ou plusieurs vendeurs à un ou plusieurs acheteurs. Dans son processus de correspondance, le commissaire-priseur doit satisfaire en premier lieu la contrainte que le prix d'offre doit être supérieur ou égale au prix de demande avant d'envisager toute correspondance possible. Ensuite, le commissaire-priseur va classer les offres et les demandes et procède par l'offre la mieux classé dont il va essayer de la correspondre avec la demande la mieux classé. Dès qu'une offre ou une demande est satisfaite il passe à la suivante en suivant la classification qu'il a adoptée. De plus la commande d'achat ou de vente peut ne pas être complètement satisfaite. Par exemple, si une commande d'achat de 08 unités est reçue par le commissaire-priseur, ce dernier peut la faire correspondre avec une commande de vente de 04 unités et une autre commande de vente de 20 unités. On remarque que la deuxième commande de vente va rester avec 16 unités non encore vendue qui resteront dans le carnet de commande du commissaire-priseur pour une correspondance ultérieure.

Dans la littérature, on peut citer le travail de D.Sun et al. [105] où est présenté un nouvel algorithme nommé NECDA, dans lequel la proposition du vendeur est déterminée par la charge de travail de ce dernier, et la valeur de l'offre de l'acheteur est déterminée en fonction de ces ressources manquantes et le temps restants avant la date limite de l'exécution de sa tâche. Ensuite, le commissaire-priseur trie les offres des acheteurs dans l'ordre croissant et les propositions des vendeurs par ordre décroissant, puis, il fera correspondre l'offre la plus élevée avec la proposition la plus faible et déterminera le prix de vente, qui sera la moyenne des deux valeurs de cette paire de correspondance.

Comme on l'a souligné au-dessus, le commissaire-priseur ne déclenche le processus de correspondance dans le PDA que périodiquement et non pas continuellement comme c'est le cas pour le CDA. L'algorithme Itayose est souvent utilisé pour adresser ce genre d'enchère où les commandes d'achat sont classées en fonction de leurs prix d'offres et les commande de ventes en fonction de leurs prix de ventes, puis déterminera le point où l'offre rencontre la demande.

Avant				Après		
# Vendeur	Prix	#Acheteur		# Vendeur	Prix	#Acheteur
7	\$15	5		5	\$15	
3	\$12	6			\$12	1
0	\$8	3		0	\$8	3
4	\$6	0			\$6	0
0	\$5	4		0	\$5	4
3	\$4	6			\$4	6
0	\$3	3		0	\$3	3

Figure 6 Exemple d'un carnet de commande de "periodical double-sided auction"

La Figure 6 présente un instantané d'un carnet de commandes avant et après le processus de correspondance déclenché par le commissaire-priseur. Dans cet exemple, le commissaire-priseur commence à correspondre l'offre la plus élevée, en l'occurrence l'offre à laquelle il a fixé le prix d'achat à \$15 avec la demande à qui il a assigné le prix de vente le moins élevé, à savoir \$4, et partiellement avec celle de \$6. Cependant la deuxième offre de \$12 aura une correspondance avec les unités restantes de la demande de vente de \$6 et une correspondance partielle avec la demande de \$12, ce qui au total ne satisfait pas l'offre de \$12 car seulement cinq de ses six unités demandées sont satisfaites et doit attendre des commandes de vendeurs potentiels ultérieurs pour satisfaire ses unités restantes. L'enchère à deux-côtés s'arrête dès que les commandes de ventes demandent un prix unitaire supérieur aux commandes d'achat.

4.1.3. Les enchères combinatoires

L'enchère combinatoire est la forme la plus générique de ventes aux enchères où l'acheteur a l'occasion d'enchérir sur un ensemble de types de ressources ainsi que leurs quantités respectives. En effet plusieurs applications du monde réel demandent plusieurs types de ressources pour accomplir leurs tâches, tel que les ventes aux enchères du spectre radioélectrique, l'énergie, les

chemins de transport et l'approvisionnement de l'entreprise [100]. Donc dans certain problème d'allocation de ressources, l'acheteur va valoriser un ensemble de biens qui lui permettra de réaliser sa tâche et n'attribuera aucunes préférences aux ressources prises individuellement. Un exemple concret est l'usage d'un téléphone. Vous ne pouvez faire usage de ce dernier que si vous obtenez l'appareil téléphonique ainsi qu'un abonnement téléphonique. L'enchère combinatoire a été motivée par le fait qu'elle évite à l'acheteur le problème d'exposition qui est que l'acheteur se retrouve avec un sous-ensemble de ces demandes générant ainsi une utilité négative. En effet, dans ce cas il y a une utilité négative qui sera générée par l'acheteur car ce dernier sera contraint de payer pour le sous-ensemble qu'il a contracté, mais en contrepartie ne pourra pas exécuter sa tâche dont cette dernière lui permettra de générer une utilité afin de compenser le paiement des ressources requis pour l'accomplissement de sa tâche. Si par exemple le teneur du marché décida de tenir plusieurs enchères séparées pour chacune de ces ressources, le participant pourrait bien sûr se présenter sur les enchères proposant les ressources qu'il demande, mais dans ce cas, l'acheteur n'a pas l'occasion de se désister des ressources gagnées sur ces enchères si ces dernières ne forment qu'un sous-ensemble des demandes de l'acheteur. Par contre, en utilisant l'enchère combinatoire, l'acheteur est assuré de l'une des deux issues : soit l'acquisition de l'ensemble de ces ressources par le paiement de ces derniers, soit il n'acquiert rien et il a rien à payer. Donc par l'utilisation de l'enchère combinatoire l'acheteur est assuré de ne pas générer une utilité négative et par conséquent éviter le problème d'exposition. I.Fujiwara et al. [43] ont proposé une vente aux enchères combinatoire à deux côtés qui permet de commander une combinaison de services dans un ordre spécifique pour les flux de travail et de co-allocations dans un marché à terme/future. Ils emploient la programmation entière mixte pour maximiser strictement le bien-être social. Le courtier peut composer de nombreuses ressources primitives dans une application sophistiquée pour la mettre en vente. Une des mesures de sécurité intéressantes est que le marché n'assignera pas de ressources à l'utilisateur s'il ne peut pas acheter l'ensemble des ressources demandées dans le même paquet.

Les agents opérant dans des enchères combinatoires ont une fonction d'utilité non-additive. On peut prendre l'exemple décrit précédemment où on note par A l'appareil téléphonique, et B l'abonnement téléphonique. On aura dans ce cas : $u(A + B) > u(A) + u(B)$. Ceci s'explique par le fait que ni A, ni B pris individuellement ne peuvent faire passer des appels téléphoniques ce qui n'est pas le cas lorsque A et B sont réunis à la fois. Il y a deux types de non-additivité : substituabilité et la complémentarité.

La *substituabilité* signifie que la valeur combinée de plusieurs biens est inférieure à la somme des valeurs individuelles. Cela se passe lorsque plusieurs biens ont la même fonction comme par exemple des produits de lavages qui ne se diffèrent que par la marque qui les a produits. Dans ce cas, si le consommateur veut acquérir un produit de lavage pour achever une tâche ménagère et il se retrouve avec deux ou plus de produits de lavage, alors l'utilité d'un produit de lavage équivaut à deux ou plusieurs produits de lavage car dans les deux cas la tâche de l'acheteur est satisfaite de manière équivalente. Il y a deux types de substitut, les substituts partiels où l'acheteur exprime une préférence vis-à-vis d'un ensemble de ressources, ce qui veut dire que l'acheteur pourra exécuter sa tâche en ayant obtenu une des ressources parmi cet ensemble, mais l'utilité générée sera différente selon la ressource que l'acheteur a choisie parmi cet ensemble. Le deuxième type de substitut est le substitut strict pour un ensemble de ressources où l'acheteur n'exprime pas de préférences par rapport aux articles de cet ensemble. Dans ce cas, l'acheteur pourra traiter cet ensemble de ressources comme des unités multiples d'un type de ressource.

La *complémentarité* est une autre fonction non-additive où la valeur combinée d'un ensemble de biens est supérieure à la somme de leurs valeurs individuelles. Par exemple, imaginez que vous voulez voyager à partir d'Annaba pour aller à Oran et que les seules destinations possibles offertes par la compagnie sont Annaba-Alger et Alger-Oran. Donc, pour l'acheteur l'utilité Annaba-Alger et Alger-Oran combinée sera supérieur que la somme des utilités des destinations de Annaba-Alger et Alger-Oran considérées individuellement.

Plus formellement, on pourrait modéliser l'enchère combinatoire comme suit : soit un ensemble d'agent $N = \{1, \dots, n\}$ et un ensemble de biens misent en négociation dénotés par $M = \{1, \dots, m\}$. Soit le vecteur $v = (v_1, \dots, v_n)$ dénote l'ensemble des fonctions d'utilités des agents où v_i est la fonction d'utilité de l'agent i . La fonction d'utilité i est défini comme suit : $v_i : 2^M \rightarrow \mathbb{R}$, c'est-à-dire elle prend comme entrée un sous-ensemble de M et comme sortie l'utilité qui lui est associée. Dans ce qui va suivre concernant les enchères combinatoires, on ne s'intéresse qu'aux problèmes qui ne traitent pas d'externalité ; ce qui signifie que nous ne considérons pas d'agent qui s'occupe aussi des allocations et des paiements d'autres agents. Le processus de déroulement de l'enchère combinatoire est comme suit : l'ensemble d'agent N va soumettre leurs offres contenant des demandes de paquet de ressources au commissaire-priseur qui va ensuite en décider sur la répartition des ressources ainsi que leurs prix d'acquisitions respectifs entre les participants selon

une fonction objective suivie par le commissaire-priseur. A l'issue de l'enchère, les agents qui auront obtenu une allocation de ressources sont ceux auquel le commissaire-priseur a accepté leurs offres et les agents qui n'ont rien obtenu comme ressources sont ceux pour lesquels le commissaire-priseur a rejeté leurs offres. Tout cela doit se faire en respectant les contraintes prédéfinies : le prix d'offre doit être supérieur au prix de réserve du vendeur, l'allocation doit être faisable ... Le problème de l'enchère combinatoire est parfois appelé le problème de détermination du gagnant (*winner determination problem*) qui est connu pour être NP-difficile et même difficile à approcher [94].

a. Enchère combinatoire Un-à-Plusieurs :

Nous allons considérer dans cette sous-section l'enchère combinatoire de type un-à-plusieurs (un vendeur et plusieurs acheteurs) où le commissaire-priseur résout le problème pour le compte du vendeur. Nous supposons qu'il n'y a pas d'externalité et que les offres sont indivisibles (l'offre doit être complètement satisfaite ou rien ne sera alors fournie). Nous notons l'ensemble de ressources par G contenant m ressources misent en enchère à n acheteurs potentiels. L'offre formulée par l'acheteur j prend la forme de tuple : $(S, p_{j,S})$ où $S \subseteq G$ et $p_{j,S}$ est le prix d'offre pour le paquet de ressources correspondant. Dans le modèle du problème de détermination du gagnant on utilise des variables de décisions $x_{j,S}$ qui prennent la valeur 1 lorsque la collection de ressources S est allouée à l'acheteur j et 0 sinon. Le modèle (E1) peut être défini comme suit :

$$\max \sum_{1 \leq j \leq n} \sum_{S \subseteq G} p_{j,S} x_{j,S} \tag{1}$$

$$\text{s.t. } \sum_{1 \leq j \leq n} \sum_{S \subseteq G} \delta_{i,S} x_{j,S} \leq 1, \forall i \in G, \tag{2}$$

$$\sum_{S \subseteq G} x_{j,S} \leq 1, \forall j, 1 \leq j \leq n, \tag{3}$$

$$x_{j,S} \in \{0,1\}, \forall S \subseteq G, \forall j, 1 \leq j \leq n, \tag{4}$$

Où $\delta_{i,S} = 1$ si $i \in S$, et 0 sinon. La contrainte (2) est une contrainte qui permet pour le commissaire-priseur résolvant ce problème de générer une allocation faisable en n'attribuant pas une même ressource à plus d'une offre. La contrainte (3) s'assure qu'aucun acheteur n'obtient plus d'un paquet dans le cas où ce dernier émet plus d'une offre. En prenant compte des contraintes : (2), (3) et (4), le commissaire-priseur a pour objectif de maximiser le revenu du vendeur en maximisant la formule (1).

Le modèle (E1) correspond au problème « set-packing » [31]. La première description classique du problème par Rothkopf et al., date de 1998 [88], a été de proposer un algorithme de programmation dynamique qui peut déterminer une maximisation des revenus d'allocation en $O(3^m)$ itérations. L'algorithme est basé sur la remarque simple qui, étant donné un sous-ensemble S d'items, le revenu maximum qui peut être atteint par la vente de S vient de l'offre $(S, p_{j,S})$ de S lui-même, ou par la vente de deux sous-ensembles S_1 et S_2 qui forment une partition de S . Les auteurs ont aussi pris en considération plusieurs restrictions relatives aux offres admissibles pour rendre le problème de calcul gérable. Par conséquent, ils montrent que le problème de détermination du gagnant peut être résolu en temps polynomial si les offres ont une structure imbriquée (N'importe quelle pair de paquets doit soit être disjointe, ou l'un d'eux est un sous-ensemble de l'autre), certaines restrictions qui demandent qu'une certaine cardinalité soient imposée (par exemple, autoriser uniquement les paquets de deux éléments ou moins), ou bien que les offres ont une structure géométrique inhérente (notamment lorsque des éléments peuvent être totalement ordonné et les paquets peuvent contenir seulement des éléments qui sont adjacents l'un à l'autre). Cependant, en limitant les paquets que chaque participant est autorisé à soumettre, cela peut conduire certain d'entre eux à des problèmes d'exposition comme on l'a expliqué auparavant dans le cas où plusieurs enchères sont lancées indépendamment les uns des autres. Regan et al [86] ont proposé une enchère scellée dans laquelle la décision sur ce qui peut être négocié est déléguée aux soumissionnaires eux-mêmes. Les soumissionnaires déposent une liste prioritaire de combinaisons sur lesquelles ils veulent soumissionner au début de la vente aux enchères et le commissaire-priseur utilisera le plus grand nombre possible de ces combinaisons lors de son processus de décision. Plus récemment, (Müller, 2006 [72]) a fourni une étude des problèmes gérables pour le problème de détermination du gagnant sur la base de deux restrictions : les restrictions de sous-ensembles et les restrictions de type de préférences.

Selon [52], les algorithmes de recherche qui ont été proposées dans la littérature (e.g [44, 94]) capitalisent sur l'observation que lorsque le nombre d'items est grand, les soumissionnaires sont susceptibles de formuler des offres uniquement sur un petit sous-ensemble de tous les ensembles possibles qui est par opposition à l'analyse du pire cas (Rothkopf et al., 1998 [88]). En particulier, (Sandholm, 2002 [94]) propose une représentation arborescente de l'espace de solutions dans laquelle les items sont indexés judicieusement de sorte qu'une allocation faisable peut être représentée qu'une seule fois. (Fujushima et al., 1999 [44]) suggèrent dans leur algorithme de CASS

(Combinatorial Auction Structured Search) une procédure de recherche structurée qui procède par profondeur d'abord dans laquelle deux idées fondamentales sont mises en avant pour éviter les calculs inutiles. La première idée, est l'identification des sous-ensembles de soumissions mutuellement incompatibles (les « bacs »), c'est à dire, qui ne peut être exécutés simultanément en raison d'un conflit sur un item, ce qui permet d'interrompre l'exploration d'une solution dès que deux éléments dans un même "bac" sont rencontrés. La deuxième idée, inspirée de la programmation dynamique, est l'utilisation des résultats intermédiaires pour l'élagage de l'arbre de recherche. Supposons que nous connaissons déjà le revenu maximum r_C^* qui peut être obtenu de la vente de $C \subseteq G$. Considérons une allocation partielle faisable du sous-ensemble $F \subseteq G$ à une étape donnée de la recherche de telle sorte que $G \setminus F \subseteq C$. Alors, si $r_C^* + r_F$ est inférieure au revenu de la meilleure allocation faisable trouvée jusqu'à ce point, alors il n'est pas nécessaire d'explorer l'arbre au-delà de F .

D'après [52], la première méthode approximative pour le problème général du « set packing » a été proposée par Hoos et Boutilier, 2000 [50] qui ont conçu l'algorithme Casanova. Casanova est un algorithme de recherche stochastique utilisant dans son exploration de l'espace d'allocation, un concept simple de voisinage. Plus précisément, une seule soumission non exécutée dans la solution correspondante à l'allocation faisable actuelle des items est choisie pour être exécutée dans la prochaine allocation faisable. Le choix se fait en fonction du "score" de l'offre, qui est représenté par le ratio du prix de l'offre au nombre d'items que l'offre comporte.

Le modèle (E1) est destiné pour les enchères combinatoires mono-items. Le modèle (E2) est un modèle multi-item et donc une généralisation du modèle (E1). Chaque ressource i peut être fournie en M_i unités. Une offre dans le modèle (E2) prend la forme $b = (\{a_{b,i}\}_{i \in G}, p_b)$, où $a_{b,i}$ est le nombre d'unités de ressource i qui sont demandées par l'offre b et p_b représente le prix d'offre de l'acheteur vis-à-vis de la collection de ressources demandées. L'acheteur ne peut soumissionner qu'un seul type de ressource par offre, et devra donc soumettre autant d'offres que de type de ressources demandées dans ce modèle (E2). Soit B l'ensemble de toutes les offres soumissionnées par les acheteurs et x_b est une variable de décision qui prend 1 si l'offre est acceptée et 0 sinon. Le modèle (E2) peut être décrit par ces trois formules suivantes :

$$\max \sum_{b \in B} p_b x_b \tag{5}$$

$$s. t. \sum_{b \in B} a_{b,i} x_b \leq M_i, \forall i \in G \quad (6)$$

$$x_b \in \{0,1\}, \forall b \in B \quad (7)$$

Le modèle (E2) est un problème de sac à dos 0-1 multidimensionnel pour lesquels des méthodes exactes ainsi qu'heuristiques ont été conçus (Martello et Toth, 1997 [67]). Parmi les contributions importantes dans le cadre d'enchères combinatoires on peut citer : (Leyton-Brown et al., 2000 [63]) et (Leyton-Brown, 2003 [64]) qui a présenté l'algorithme CAMUS ("Combinatorial Auction Multi-Unit Search").

b. Enchère combinatoire Plusieurs-à-Un :

L'enchère combinatoire plusieurs-à-un est une sorte d'enchère inversée traitant du problème combinatoire de l'enchère, d'où son appellation parfois d'enchère combinatoire inverse. Dans l'enchère combinatoire plusieurs-à-un, il y a plusieurs fournisseurs pour un seul acheteur qui désire acquérir un ensemble de ressources notée par G . Dans ce type d'enchère c'est le vendeur qui soumit son offre et non pas l'acheteur dont sa forme peut être définie comme : $b = (S_b, p_b)$ où S_b est un sous-ensemble de G et p_b est son prix demandé. Soit l'ensemble de toutes les soumissions des vendeurs représentées par B et soit x_b représente la variable de décision qui prend la valeur 1 si l'offre est acceptée et 0 sinon. Le problème de détermination du gagnant dans ce cas sera de trouver la solution qui satisfait les demandes de l'acheteur tout en minimisant leurs coûts d'acquisition. Le modèle à ce problème est noté par (E3) :

$$\min \sum_{b \in B} p_b x_b \quad (8)$$

$$s. t. \sum_{b \in B} \delta_{i,S_b} x_b \geq 1, \forall i \in G, \quad (9)$$

$$x_b \in \{0,1\}, \forall b \in B, \quad (10)$$

Où $\delta_{i,S_b} = 1$ si $i \in S_b$ et 0 sinon. Le modèle (E3) est un problème de couverture par ensemble, en anglais : « set cover problem » qui est NP-Difficile. Une supposition implicite de « free disposal » -- l'acheteur peut s'en procurer des ressources qui lui sont extras et le vendeur peut garder certain ressources invendu—est faite à l'égard du modèle (E3). Si cette supposition ne peut être assumée dans un contexte de marché particulier, alors la contrainte (9) doit passer d'une équation d'inégalité à une équation d'égalité.

c. Enchère combinatoire plusieurs-à-plusieurs :

Dans les enchères combinatoires plusieurs-à-plusieurs, il y a plusieurs acheteurs et vendeurs. Dans ce cas la compétition se passera dans les deux côtés : acheteurs et vendeurs comme dans le cas de l'enchère à deux côtés pour une ressource, mais la différence est que l'acheteur a la possibilité de négocier un paquet de différentes ressources et de même pour le vendeur. Les offres soumissionnées par l'acheteur ou le vendeur auront la forme suivante : $b = (\{q_{b,i}\}_{i \in G}, p_b)$ où $q_{b,i}$ est la quantité de ressources demandées par l'acheteur et dans ce cas elle prend une valeur positive ou proposé par le vendeur et dans ce cas elle prend une valeur négative. p_b est le prix d'offre qui est exprimé par un nombre positive si c'est l'acheteur qui a émis l'offre et par un nombre négative si c'est le vendeur qui a émis la proposition. Soit B l'ensemble de toutes les offres des acheteurs et vendeurs et soit la variable de décision x_b qui prend la valeur de 1 en cas où l'offre a été conclue entre les deux parties concernées et 0 dans le cas inverse. Assumant que les offres sont indivisibles, i.e., l'ensemble des ressources qui sont contenus dans l'offre sont négociées ou rien du tout, le problème de détermination du gagnant peut être formulé comme le modèle (E4-a) :

$$\max \sum_{b \in B} p_b x_b \quad (11)$$

$$s. t. \sum_{b \in B} q_{b,i} x_b \leq 0, \forall i \in G, \quad (12)$$

$$x_b \in \{0,1\}, \forall b \in B, \quad (13)$$

Le modèle (E4-a) a pour objectif de maximiser le surplus total du marché exprimé par la formule (11) en respectant les contraintes (12) et (13). De plus ce modèle suppose implicitement la supposition de "free disposal" et lorsque le contexte ne le permet pas on doit changer l'inégalité de (12) par une contrainte d'égalité.

Lorsque les offres sont considérées comme divisibles, c'est-à-dire que l'offre de l'acheteur où la demande du vendeur peut ne pas être satisfaite à 100%. Dans ce cas la variable de décision x_b désignera la proportion de l'exécution de l'offre b où $b \in B$, et où la variable x_b sera compris entre l'intervalle réel 0 à 1. Le prix $p_b(x_b)$ représente le montant que l'acheteur va payer ou le vendeur va recevoir au cas où x_b de l'offre b soit exécutée. Le problème d'allocation peut être formulé comme un modèle (E4-b) :

$$\max \sum_{b \in B} p_b(x_b) \quad (14)$$

$$s. t. \sum_{b \in B} q_{b,i} x_b \leq 0, \forall i \in G, \quad (15)$$

$$0 \leq x_b \leq 1, \forall b \in B \quad (16)$$

(Sandholm et Suri, 2003 [97]) suggèrent l'algorithme de BOB, dans lequel ils adaptent diverses techniques de recherche précédemment proposées pour le modèle d'allocation un-à-plusieurs (E1).

Les applications d'enchères combinatoires plusieurs-à-plusieurs ont été proposées pour l'équilibre du marché dans les processus d'industries [56]. Ce dernier est particulièrement intéressant, car le modèle considéré par les auteurs considère les offres de demandes et de fourniture sur les produits simples, mais avec différents niveaux de qualité. Le fait que le modèle tolère la substitution entre les produits ayant différents niveaux de qualité donne lieu à des contraintes supplémentaires sur le processus de correspondance entre les ordres d'achats et de ventes.

Chapitre 5 Négociation automatisée

5.1 Introduction

Dans cette section, nous allons discuter des différents modèles de négociation automatisée dans les systèmes multi-agents.

Pour commencer, nous allons donner une définition de la négociation : *la négociation permet de répartir des ressources au sein d'agents ayant des conflits d'intérêt mais un désir de coopérer. Cette coopération permettra de conclure des contrats par l'approbation mutuelle des parties concernées dans les contrats qui en résultera par la répartition des ressources au sein de la communauté d'agent.*

La négociation automatisée devient pertinente lorsque les agents ont des intérêts conflictuels, c'est-à-dire des revendications concurrentielles sur les ressources disponibles et qui ne peuvent être toutes satisfaites à la fois. Le mot « ressource » est à prendre au sens large, pouvant prendre la forme de produits, services, temps, argent etc.

Ce chapitre est organisé comme suit. Dans la section suivante, nous présenterons les concepts de base du problème de négociation automatisée. Ensuite, nous discuterons et critiquerons les approches de la négociation automatisée basée sur la théorie des jeux à la section 5.3, suivie d'une discussion sur les approches heuristiques dans la section 5.4. Dans la section 5.5, nous donnerons un aperçu de l'idée de la négociation automatisée basée sur l'argumentation. Enfin dans la section 5.6 nous discuterons sur la négociation automatisée pour ressources à structure complexe qui est liée directement à nos contributions apportées dans ce domaine.

5.2 Le problème de négociation

L'objectif de cette section est de décrire certains concepts fondamentaux relatifs à la négociation automatisée et qu'on retrouve dans la littérature de la négociation automatisée.

5.2.1 L'espace d'accords de la négociation

L'objectif principale de la négociation est qu'à l'issue de la négociation, les ressources seront réparties par l'approbation de l'ensemble des parties concernées. La négociation peut être considérée comme une « recherche distribuée à travers un espace d'accords potentiels » [54]. Un

espace d'accords est défini comme l'ensemble des solutions possibles à un problème d'allocation de ressources.

Plus formellement, un espace d'accords est caractérisé par un ensemble d'attributs : A_1, \dots, A_n où chaque attribut A_i représentera une dimension donnée dans l'espace euclidien multidimensionnel. De plus, chaque attribut A_i sera compris entre : a_1^i, \dots, a_m^i où $s(A_i)$ représente le nombre de valeurs possibles que l'attribut A_i peut prendre. L'espace d'accords est le produit de l'ensemble des $s(A_i)$'s, en d'autres termes : $\prod_{i=1}^n s(A_i)$. Ceci est illustré par l'exemple suivant :

Exemple 1. *Supposons un acheteur voulant acheter une voiture, où son budget est limité à \$20 000 . Avec ce budget-là, l'acheteur peut s'en procurer une voiture dont son prix est compris entre \$10,000 et \$20,000. Trois marques de voiture peuvent être envisageables : Renault, Toyota ou Ford. Dans ce cas, l'espace d'accords est représenté par deux attributs : Prix et Marque, dont le premier a 10 001 possibilités et le second a 3 ce qui en résulte à 30 003 combinaison possibles formant l'espace d'accords.*

A partir de l'exemple 1, on remarque que l'espace d'accords s'accroît par l'accroissement des deux facteurs suivants : le premier est le nombre d'attributs qui le compose et le second l'intervalle de valeurs que chaque attribut peut prendre qui n'est autre que le domaine de ce dernier.

Notez que dans l'exemple 1, nous avons évalué à 30 003 possibilités le nombre de combinaisons possible que peut comprendre cet espace d'accords. En fait, l'espace d'accords tient en compte aussi des combinaisons possibles des paires d'identités acheteurs-vendeurs possibles lorsqu'il y a plus d'un acheteur et un vendeur, ce qui fait étendre encore plus l'espace d'accord. La nécessité de mentionner l'identité de l'agent est illustrée dans l'exemple qui suit :

Exemple 2. *L'administration du transport qui alloue pour chaque camion une destination précise, doit non seulement définir la tâche de transport à effectuer mais aussi l'identité du camion qui sera en charge de l'effectuer. Supposons donc, que camion1 aura pour rôle de transporter la marchandise vers les destinations1 et 2, tandis que camion2 vers la destination3. L'espace d'accords sera alors représenté comme suit :*

$$((\text{camion1:}\{\text{destination1, destination2}\}), (\text{camion2:}\{\text{destination3}\}))$$

Dans ce problème on ne peut pas mentionner de manière implicite l'identité de l'agent qui va accomplir la tâche comme c'était le cas dans le premier exemple où il y avait comme agents qu'un seul acheteur et vendeur.

5.2.2 Mécanisme de négociation

Un mécanisme de négociation est l'ensemble des règles qui définissent le déroulement de la négociation tel que les règles d'acceptation et de refus de l'offre. Un exemple d'un mécanisme est l'enchère anglaise qui a comme règles : 1) chaque participant doit soumettre une offre plus élevée que celle dernièrement déposée 2) le gagnant sera celui dont l'offre n'a pas été surenchérie par un autre compétiteur.

Différents mécanismes peuvent avoir des propriétés différentes. Voici la liste décrite par Rosenschein et Zlotkin [87] sur les caractéristiques désirables pour les mécanismes de négociation :

1. *Simplicité* : Un mécanisme qui nécessite moins de traitement est préférable à celui qu'il ne l'est pas.
2. *Efficacité* : Un mécanisme est efficace s'il produit un bon résultat. Qu'entend-t-on par « bon résultat » ? peut varier d'un domaine à un autre. Un des critères communs est l'optimum de Pareto, où aucun agent ne pourrait être mieux dans une autre allocation sans que d'autres agents soient moins bien.
3. *Distribution* : Il est préférable d'avoir un mécanisme de négociation qui n'implique pas un décideur central. Centralisation peut conduire à des goulots d'étranglement en communication ou conduit à la diminution de la fiabilité du mécanisme en raison de son point central.
4. *La symétrie* : cette propriété implique que le mécanisme ne devra pas être biaisé pour ou contre un certain agent sur la base de critères inappropriés. Encore une fois, ce qui constitue un critère « inapproprié » dépend du domaine en question.
5. *Stabilité* : Un mécanisme est stable si aucun agent n'a un intérêt à dévier d'une ou d'un ensemble de stratégies convenues. Dans une vente aux enchères, par exemple, nous pouvons exiger qu'aucun agent ne mente en faisant une fausse offre, ou qu'aucun groupe d'agents ne peuvent former des coalitions stratégiques pour créer un désavantage aux autres agents.

6. *Flexibilité* : par cette propriété, nous entendons que le mécanisme devrait conduire à un accord, même si les agents ne disposent pas d'informations privées complètes et correctes par rapport à leurs propres décisions et leurs propres préférences.

Une grande variété de mécanismes a été discutée dans la littérature, où chacun a ses propres avantages et inconvénients. Dans ce qui va suivre nous allons discuter de grande famille de mécanismes.

5.3 Approches de négociation basée sur la théorie des jeux

Parmi les approches de négociation, il y a ceux qui sont basées sur la théorie des jeux.

5.3.1 Un aperçu de la théorie des jeux non coopératifs

La théorie des jeux est une branche de l'économie spécialisée dans l'étude de la prise de décision. C'est un ensemble d'outils qui permet d'analyser des situations où l'ensemble des agents doit être rationnel et où la stratégie optimale d'un agent va en dépendre des anticipations des stratégies des autres agents. Récemment, elle a été largement utilisée pour étudier et concevoir l'interaction entre les agents computationnels d'intérêts personnels [99].

La théorie des jeux fournit des outils pour mener deux types d'analyses [84] :

- L'analyse du comportement (ou stratégie) optimale des individus ou des organisations étant donné un mécanisme d'allocation de ressources sous-jacent ;
- Analyser comment on peut concevoir des mécanismes optimaux, étant donné que les agents se comportent de manière stratégique.

Dans ce qui va suivre nous allons seulement décrire la première analyse.

Concepts pour l'analyse du comportement stratégique :

Quand on parle de jeu dans la théorie des jeux, on parle alors d'objet mathématique bien-définie. Un jeu est complètement spécifié lorsque les éléments suivants lui ont été définis : les joueurs du jeu, l'information et les actions disponibles pour chaque joueur à chaque point de décision et l'utilité que chaque joueur peut gagner pour chaque résultat possible du jeu. Le mécanisme d'une négociation basé sur la théorie des jeux précise : 1) l'ensemble des actions disponible pour chaque joueur dans chaque point de décision ; 2) Le résultat de la rencontre effectuée sur la base des actions/stratégies entrepris par les différents joueurs.

L'exemple le plus populaire de la théorie des jeux est celui du dilemme du prisonnier, introduit par Poundstone, 1993 [83]. C'est un jeu à deux joueurs où ces derniers ont été détenus par la police. Les deux joueurs sont placés dans des pièces séparés afin qu'ils ne puissent communiquer entre eux. Cela se traduit par le fait que chaque joueur ne connaît l'action de l'autre joueur lorsque ce dernier devra agir et choisir entre : *avouer* ou *ne pas avouer*. C'est un jeu dont sa représentation est matricielle comme le montre le tableau 3 ci-dessous.

Tableau 3 Dilemme du prisonnier

	Ne pas avouer	Avouer
Ne pas avouer	(5, 5)	(0, 10)
Avouer	(10, 0)	(3, 3)

Il y a donc deux joueurs, le premier où ces actions sont au long des lignes de la matrice et le second sont au long des colonnes de la matrice. Le résultat de la rencontre est sous forme de tuple dont le premier représente l'utilité du premier joueur et le second représente celle du second joueur. On peut remarquer à partir du tableau 3 qu'il y a quatre scénarios, dont deux sont symétriques :

1. (Joueur 1 : Ne pas avouer, Joueur 2 : Ne pas avouer) : Dans ce cas les deux seront libérés et partageront le butin moitié moitié entre eux.
2. (Joueur 1 : Ne pas avouer, Joueur 2 : Avouer) ou (Joueur 1 : avouer, Joueur 2 : Ne pas avouer) : Ce sont deux scénarios symétriques, où le joueur qui avoue sera libéré et obtient l'intégralité du butin. Le sort de l'autre joueur sera une utilité nulle étant donné qu'il n'a pas voulu coopérer avec la police et que cette dernière a un témoignage contre lui ce qui en résultera pour lui une longue période de détention.
3. (Joueur 1 : avouer, Joueur 2 : Avouer) : Dans ce dernier cas, les deux joueurs confessent et doivent purger une durée réduite en prison.

Un concept central de la théorie des jeux est celui de l'*équilibre*. Le type d'équilibre répondue dans le domaine de la théorie des jeux est l'*équilibre de Nash* où la déviation de la stratégie de l'agent lui générera au plus l'utilité qu'il a généré en gardant sa stratégie définit dans l'équilibre de

Nash. Dans l'exemple qu'on a cité au-dessus, les stratégies « avouer, avouer » forment un équilibre de Nash.

5.3.2. La théorie des jeux pour la négociation automatisée

La théorie des jeux offre un outil puissant pour concevoir les stratégies d'agents opérant dans le domaine de la négociation automatisée [84]. En effet, la théorie des jeux est utilisée non-seulement pour la conception de stratégie d'agent mais aussi pour les mécanismes de négociations. Le domaine de la conception du mécanisme computationnel [24] utilise des techniques de conception de mécanisme afin de construire des mécanismes d'allocation de ressources dans les systèmes multi-agents.

5.3.3 Les limitations de la théorie des jeux

Une évaluation adéquate de la théorie des jeux dans les approches de négociations dépasse le cadre de cette thèse. Nos discussions seront concentrées sur les questions pertinentes à la négociation automatisée.

Dans l'analyse de la théorie des jeux, les chercheurs tentent généralement de déterminer la stratégie optimale en analysant l'interaction comme un jeu entre les participants identiques, et cherchant son équilibre [107]. La stratégie déterminée par ces méthodes peut être optimale pour ces participants à condition de se conformer aux règles du jeu et à certaines suppositions.

La théorie des jeux classique suppose, notamment, que les agents [84] :

1. Disposent de ressources de calcul illimitées
2. Ont une connaissance complète de l'espace de résultat, et
3. Sont des optimiseurs d'utilité dans le sens de la théorie du choix rationnel,

D'un point de vue pratique, ces suppositions impliquent des suppositions irréalistes sur les agents logiciels de la négociation.

La première supposition est une forme de relaxation du problème de négociation en supposant que les ressources sont illimitées. Cependant tous les environnements computationnels réalistes ont des capacités limitées en traitement de l'information et en communication. Les modèles de la théorie des jeux n'offrent donc pas le moyen de tenir compte de cette limitation réelle.

La deuxième supposition mentionne que l'agent a une connaissance complète de l'espace de résultat et cela est possible lorsqu'on spécifie à l'agent l'ensemble des préférences de l'utilisateur, ce qui n'est pas toujours possible à réaliser.

La troisième supposition indique que l'agent optimise l'utilité au sens de la théorie du choix rationnel car la théorie des jeux exige cela. En effet, dans la théorie des jeux, les agents doivent raisonner de façon optimale en démarrant par le fait que leurs adversaires vont adopter la meilleure stratégie. Mais comme les agents logiciels peuvent avoir des ressources limitées ne leurs permettant pas de générer des réponses optimales, suivre alors les *prédictions* de la théorie des jeux qui reposent sur des suppositions impraticables en termes de capacité de ressources va guider l'agent à des prescriptions erronées.

La théorie des jeux évolutionnaire :

Il est intéressant de souligner qu'un sous-domaine de la théorie des jeux émerge, appelé la théorie des jeux évolutionnaire [93], dont son utilisation est motivée par certaines motivations évoquées ci-dessus. La restriction de la théorie des jeux évolutionnaire est la relaxation de la supposition de la rationalité illimitée. Au lieu de calculer les stratégies optimales, le jeu est joué à plusieurs reprises et les différentes stratégies possibles sont testées par un processus d'apprentissage essais-erreurs afin de trouver les meilleurs d'entre elles à adopter dans le jeu. Aussi la théorie des jeux évolutionnaire est utilisée pour la modélisation de ce qui est appelé la « rationalité limitée » en capturant explicitement les éléments du processus de choix, tel que la mémoire limitée, la connaissance limitée, les préférences approximatives (qui ignorent les différences mineures entre les options), etc [91]. Ces frameworks sont plus réalistes car ils décrivent mieux les scénarios économiques et sociaux réels.

Dans le volet relatif à notre thèse, G. Wei et.al [109] ont proposé une approche d'allocation de ressource équitable basée sur la théorie des jeux évolutionnaire dans l'environnement du cloud computing, dans lequel les demandeurs de service ont l'intention de résoudre des problèmes de calcul parallèle compliqués. Une solution approximée pratique à deux étapes a été proposée à cet effet : Premièrement, chaque participant résout son problème optimal de manière indépendante, sans prendre en considération les assignements de ressources multiplexées où la ressource multiplexé est assignée à une ou plusieurs sous-tâches. Une méthode de programmation d'entier binaire est proposée pour résoudre l'optimisation indépendante. Deuxièmement, un mécanisme

évolutionnaire est désigné, lequel change les stratégies multiplexées des solutions optimale initiales des différents participants tout en minimisant les pertes efficacement. Les algorithmes dans le mécanisme évolutionnaire prennent en compte à la fois l'optimisation et l'équité. Selon les auteurs, il a été démontré que l'équilibre de *Nash* existe toujours si le jeu d'allocation de ressources a des solutions faisables.

5.4 Approches basées heuristique pour la négociation

Lorsque les concepteurs d'agent essayent d'adresser certains problèmes qui sont caractérisés par certaines relaxation de certaines hypothèses de la théorie des jeux, en particulier la relaxation de l'hypothèse de la rationalité illimitée car des prédictions sur les problèmes du monde réel, qui sont basées sur cette hypothèse peuvent être erronées. Au lieu de se baser sur la théorie des jeux pour la conception de stratégies afin d'adresser le problème, on peut approximer le problème via l'emploi d'heuristique qui sont des solutions approximatives au problème. L'avantage d'heuristique contrairement aux algorithmes qui trouvent des solutions exactes est qu'ils sont moins gourmands en ressources. Le support pour une heuristique particulière est habituellement basé sur des tests empiriques et l'évaluation de la comparaison avec d'autres heuristiques [59].

5.4.1. Heuristiques pour la négociation automatisée

Heuristiques pour la négociation :

Un certain nombre de méthodes heuristiques ont été employées dans un cadre de négociation "service-oriented" présenté par Faratin [39]. Dans ce cadre, différentes fonctions de décisions heuristiques sont utilisées pour évaluer et générer des offres dans des négociations multi-attributs. Au lieu d'explorer toutes les offres possibles, les agents échangent des offres basées sur les fonctions heuristiques qui dépendent sur les dates limites et la disponibilité de la ressource. De plus, afin d'améliorer la convergence vers un accord, les auteurs présentent une méthode qui permet à un agent de générer des offres qui sont « similaires » aux offres précédentes faites par l'autre partie de la négociation (où, la représentation de "similarité" est basée sur des techniques de logique floue). L'intuition derrière cela est que ces offres sont plus susceptibles d'être acceptées par la contrepartie.

Kowalczyk et Bui [60] présentent un modèle de négociation avec les procédures de décision basées sur la satisfaction de contraintes distribuées. Ce framework a ensuite été étendu pour permettre de multiples négociations concurrentielles [85].

5.4.2. Limitations des approches heuristiques

Les méthodes heuristiques surmontent un grand nombre de lacunes provenant des approches basées sur la théorie des jeux. Cependant, ils ont aussi un certain nombre d'inconvénients [54]. Tout d'abord, les modèles peuvent parfois conduire à des résultats qui sont sous-optimaux car entre autres ils se basent sur une rationalité limitée. Une autre limitation est la supposition faite aux agents : les agents *savent ce qu'ils veulent*, en d'autres termes, les agents ont la capacité d'attribuer correctement les utilités selon leurs préférences à chaque objet mise en négociation. Comme nous allons le montrer dans la section suivante, cette supposition ne peut être satisfaite dans certains contextes, et dans de tels cas des techniques alternatives seraient nécessaires.

5.5 Approches de négociation argumentatives

On rappelle que dans 5.2.2, l'une des propriétés souhaitables d'un mécanisme de négociation est la *flexibilité* : la capacité à conduire à un accord même dans le cas où les agents ne disposent pas d'informations privées complètes et correctes par rapport à leurs propres décisions et préférences. Dans la prochaine sous-section, nous allons argumenter que les approches de la théorie des jeux et heuristiques ne peuvent satisfaire cette propriété et comment des frameworks basé argumentation peuvent la satisfaire. Ces frameworks ont été nommés "argumentation-based negotiation (ABN) frameworks".

5.5.1. Aperçu sur les modèles heuristiques et la théorie des jeux

Selon Rahwan Iyad [84], la plupart des approches basée heuristiques et théorie des jeux ont comme hypothèse que les préférences et les utilités des agents sont définies avant l'interaction, ce qui permet à l'agent de comparer différentes propositions. Ceci est envisageable en pratique lorsque les propositions à comparer s'évaluent en termes d'une métrique numérique. Comme exemple : si l'utilité de l'agent est basée sur la tarification par heure de la consommation électrique, il peut alors facilement comparer les propositions de différentes compagnies d'électricité en se référant à la métrique de la tarification par heure de la consommation électrique de chacune. Toutefois, dans certaines situation complexe, tels que les négociations syndicales, l'utilité qui modélise les

préférences de l'agent est difficile à définir et l'agent doit donc acquérir l'information relative à ces préférences de manière dynamique à travers le processus de négociation.

Chez les humains, le processus de révision de préférences a souvent eu lieu au cours de la négociation. Cependant les frameworks basés sur les heuristiques et la théorie des jeux ne peuvent avoir cette flexibilité d'ajuster leurs préférences au cours des négociations pour les raisons citées dans [84] :

- Dans la plupart des modèles de théorie des jeux et heuristiques, les agents échangent des propositions (accords potentiels ou transactions potentielles). Les agents ne sont pas autorisés à échanger n'importe quelle autre information supplémentaire autre que ce qui est exprimé dans la proposition elle-même.
- Les préférences des agents par rapport aux propositions sont supposées être "correctes" dans le sens où ils reflètent le véritable bénéfice que l'agent reçoit en cas où ces préférences sont satisfaites. Par exemple, un agent qui essaie d'acheter une voiture, peut assigner une valeur élevée à une marque particulière basée sur une fausse croyance que cette marque fabrique des voitures plus sûres que les autres marques. Dans ce cas, les préférences ne reflètent pas correctement le gain réel de l'agent si ce dernier achète cette voiture.
- Les approches heuristiques et de la théorie des jeux supposent que les utilités ou préférences des agents sont fixes. Un agent rationnel modifierait ses préférences seulement à la réception de nouvelles informations, et les mécanismes de négociations automatisées traditionnels ne facilitent pas l'échange de telles informations.

Par contre, les approches fondées sur l'argumentation ont la capacité de faire ajuster leurs préférences à travers les arguments échangés par les agents.

5.5.2. Les arguments dans la négociation

Dans le contexte de la négociation, l'argument est une pièce d'information qui permet à l'agent de [84] :

1. *Justifier* sa position de négociation ; ou
2. *Influencer* la position de négociation d'un autre agent.

Les justifications sont utilisées afin de véhiculer des « conseils utiles » lors de la négociation.

Prenons l'exemple suivant :

ACHETEUR : *Je voudrais passer des vacances d'une durée de 08 jours s'il vous plaît.*

VENDEUR : *Il y a un hôtel qui vous permet de séjourner pendant 8 jours à \$200.*

ACHETEUR : *J'ai un budget limité de \$100. Quel est le meilleur que vous pouviez faire ?*

Dans ce cas, l'acheteur justifie son rejet par le fait que son budget ne lui permet pas de dépasser la barre des \$100. Le vendeur dans ce cas a reçu plus qu'un rejet car il a aussi les causes du rejet de l'acheteur ce qui lui permet de ne plus proposer à cet acheteur des offres au-delà du prix de \$100.

Les arguments ont aussi une fonction d'influencer les préférences d'un autre agent en fournissant à ce dernier de nouvelles informations. Le dialogue suivant illustre un cas de changement de préférence :

ACHETEUR : *Je voudrais passer des vacances d'une durée de 08 jours s'il vous plaît.*

VENDEUR : *Il y a un hôtel qui vous permet de séjourner pendant 8 jours à \$200.*

ACHETEUR : *C'est beaucoup !*

VENDEUR : *Mais vous ne trouverez pas moins cher sur le marché, parce que c'est la saison estivale !*

Ici, bien que l'acheteur a les moyens de s'offrir la proposition du vendeur, il l'a refuse en premier lieu, la considérant comme une mauvaise alternative. Mais, vu que le vendeur lui a apporté des informations justifiant le prix de l'offre proposé, les préférences de l'acheteur pourraient être modifiées, ce qui permettra de faire tourner un rejet de l'acheteur vis-à-vis du vendeur à un accord avec ce dernier.

5.6 Négociation automatisée pour les problèmes d'allocation de ressources complexe

On considère qu'il y a un certain nombre de problèmes non-résolus au sujet du problème d'allocation de ressources à structure complexe en se basant sur les systèmes multi-agents. Cette thèse a pour rôle d'adresser les problèmes relatifs à l'allocation de ressources à structure complexe,

où la négociation se passe en mode plusieurs-à-plusieurs. Plus spécifiquement, cette thèse va investiguer sur les deux questions suivantes :

- Quand on offre à l'agent la possibilité de se désengager d'un contrat, quelle sera alors la meilleure stratégie à lui faire adopter en termes de détermination du prix d'offre et de pénalité de désengagement afin qu'il maximise ses utilités ?
- Quelle sera le modèle de négociation le plus adapté pour la plateforme cloud computing due à la dynamique inhérente dans les marchés opérés dans une telle plateforme ?

5.6.1. Négociation avec incertitude

Le problème de trouver des stratégies rationnelles à un problème d'allocation de ressources où l'information n'est pas toujours connue par l'ensemble des agents, comme c'est le cas pour certaine information qui sont d'ordre privée, alors il n'y a généralement pas d'algorithmes applicables [41]. Bo.An et.al [17] ont proposé une négociation bilatérale avec incertitude à un-côté sur les prix de réserves. Pour cela, ils se sont inspirés du protocole de négociation de Rubinstein [90]. Ce problème a été modélisé comme un « Bayesian extensive-form game » avec un nombre infini de stratégies étant donné que le prix est une valeur continue. Le concept de solution approprié pour ce genre de problème a été *l'équilibre séquentiel* qui spécifie une paire : un système de croyance et les stratégies. Cette paire doit satisfaire une circularité qui consiste à ce que : Les stratégies doivent être séquentiellement rationnelles étant donné le système de croyances et le système de croyance doit être consistant par rapport aux stratégies. L'algorithme en question combine ensemble l'analyse théorique du jeu avec les techniques de recherche de l'espace d'état, et il est considéré comme solide et complet selon l'auteur. Ils emploient une approche vers l'arrière pour trouver les équilibres séquentiels dans le contexte d'un processus de recherche vers l'avant : pour calculer la stratégie d'équilibre des agents dans un jeu continu avec une certaine croyance, ils cherchent en avant pour trouver les stratégies d'équilibres des agents dans un jeu continu avec des croyances différentes et considèrent tous les choix de règles possibles des agents ainsi que les règles de mise à jour de la croyance. Dans le même temps, ils dérivent théoriquement les stratégies optimales des agents en appliquant "Bayesian extension of backward induction" et vérifie les conditions d'existence d'équilibre.

5.6.2. Négociation un-à-plusieurs et plusieurs-à-plusieurs

Une hypothèse courante dans la littérature est qu'un agent ne peut poursuivre qu'une seule négociation à la fois. Ceci est pertinent lorsque l'agent juge qu'une autre alternative est meilleure que le contrat qui est entre ces mains. La présomption que l'agent ne peut poursuivre qu'une seule négociation semble restrictive à l'agent.

La négociation devient de plus en plus considérée de nos jours et son champ d'application inclut : l'e-commerce, la planification de la fabrication et l'acheminement de véhicule de manière distribuée, etc. La prolifération et l'acceptation des technologies d'internet ont rendu le remplacement de certaines transactions effectuées manuellement par une négociation automatisée basée sur les mécanismes d'enchères non seulement possible mais aussi efficace. Mais l'avantage des mécanismes de négociation par rapport aux enchères est leur capacité à adresser des problèmes où les attributs de services et de marchandises sont mal définis et qu'il y a des critères autre que le prix (par exemple, la réputation, la confiance, la relation et les contrats futurs) [47]. En outre, aucun tiers comme un commissaire-priseur n'est nécessaire à la négociation, ce qui permet de rendre les décisions de négociations distribuées.

5.6.3 Négociation Multi-Ressources

Sur les marchés de commerce électronique où les agents sont généralement d'intérêt-personnel, leurs besoins parfois demandent plusieurs ressources à acquérir. Chacune de leurs ressources peut être disponible dans plusieurs fournisseurs, ce qui fait que l'agent va lancer plusieurs négociations concurrentielles. Pour illustration, on peut se référer aux exemples de la section 1, comme CLASP [21] où un site génère un plan dans son exécution requière un ensemble de ressources et donc un ensemble de négociations concurrentielles doivent être déployées à cet effet. Effectivement, si les multiples négociations ne sont pas toutes réussies, il y aura effondrement de la négociation, et l'agent ne gagne rien. Il s'agit d'une forme de négociation « multi-chainée » où les ressources sont indépendantes individuellement mais interdépendantes si on les considère comme un ensemble. Les ressources sont indépendantes individuellement dans le sens où le fonctionnement d'une ressource ne dépend pas de l'autre ressource demandée par l'acheteur, mais elles sont interdépendantes car l'acheteur ne peut faire exécuter sa tâche que lorsque l'ensemble des ressources demandées soit satisfait.

Le problème de la négociation que nous considérons présente les trois caractéristiques suivantes :

1. Les agents auront la possibilité de se désengager des accords provisoires en payant une certaine pénalité à l'autre partie du contrat qui sera considéré comme une forme de compensation à ce dernier. Cela est intéressant si l'agent voit venir une opportunité plus prolifique en termes de revenue que le contrat courant ou ce dernier devient infaisable par rapport aux contraintes de l'agent.
2. Les agents de négociation auront des informations incomplètes sur les autres agents. Ces informations qui ne sont pas connues par l'ensemble de la communauté d'agents pour des raisons stratégiques sont appelées informations privées. Parmi les informations privées, il y a la stratégie de l'agent qui n'est connue que par l'agent lui-même, le statut de négociation opéré par chaque vendeur (le nombre de propositions qu'il a reçu).
3. L'acheteur négocie concurrentiellement avec les vendeurs, c'est-à-dire lance plusieurs threads de négociation en parallèle chacune destinées à un fournisseur précis.

Puisque consommateurs et fournisseurs ont des objectifs concurrentiels sur des ressources limitées, des préférences, des intérêts et des politiques différentes, le problème de trouver une allocation de ressources optimale a été considéré comme intraitable à la fois en termes de calcul [33] et de communication [37]. La négociation multi-ressources qu'on adresse dans cette thèse est plus complexe du fait du désengagement.

Le problème de négociation multi-ressources est différent de la négociation multi-attributs [15, 61] dans laquelle les négociations sont bilatérales. Sim et Shi [102] ont proposé une stratégie de coordination pour la négociation multi-ressources, où un agent peut négocier avec plusieurs agents comme dans cette thèse. Chaque acheteur connaît le prix de réserve (prix d'offre maximale) de chaque ressource à l'avance et l'acheteur doit simplement décider de la stratégie de concession de chaque négociation un-à-plusieurs pour une ressource. En outre, un acheteur dans [102] ne fait qu'un accord provisoire, mais dans ce travail, un acheteur peut faire plus d'un accord provisoire.

Nguyen et Jennings [74, 75] fournissent et évaluent un modèle d'engagement pour la négociation simultanée. Cependant, le nombre maximal d'accords provisoires est déterminé avant la négociation. Dans notre travail, le nombre maximal d'accords provisoire est déterminé par la

situation du marché et changera dynamiquement lors de la négociation. En outre, notre travail étudie un problème de négociation multi-ressources, plutôt que la négociation d'une seule ressource comme dans [74, 75]. En outre, Nguyen et Jennings [74, 75] font des hypothèses très restrictives sur l'information disponible des agents.

Selon Bo An et.al [16], la position de négociation d'un agent à chaque tour est déterminée par de nombreux facteurs tels que la concurrence sur le marché, les dates limites de négociation, l'ensemble d'accords courants, les propositions des partenaires commerciaux, et la dynamique du marché. La proposition qu'ils ont adoptée est qu'au lieu de modéliser explicitement ces facteurs interdépendants et ensuite déterminer les meilleures décisions de chaque agent par une optimisation combinée infaisable, ils ont essayé de connecter ces facteurs interdépendants indirectement et développer un ensemble d'heuristiques pour approximer la prise de décision des agents au cours de la négociation. En adoptant ces heuristiques comme approximation, une des caractéristiques distinctives des agents de négociation, c'est qu'ils sont conçus avec la flexibilité d'ajuster le montant de la concession en réagissant à i) l'évolution des conditions du marché, et ii) le statut de négociation courant de tous les threads de négociations simultanés.

Dans le travail de recherche de Bo.An et.al [15], les agents utilisent une stratégie de négociation en fonction du temps dont le prix de réserve de chaque ressource est déterminé dynamiquement par 1) la probabilité que la négociation ne sera pas réussie (la *probabilité de conflits*), 2) le *prix de l'accord attendu* de la ressource, et 3) *le nombre attendu des accords finaux* étant donné l'ensemble des accord temporaires fait jusqu'ici. Enfin, la date limite de négociation de chaque ressource est déterminée par sa rareté.

Dans les simulations que nous avons fait tourner avec le simulateur Cloudsim [2], plusieurs scénarios ont été testés en évaluant plusieurs configurations différentes en termes de dates limites, nombre de ressources à acquérir ou vendre et à différent ratio offre/demande pour chaque ressource. Les résultats expérimentaux montrent des tendances favorables à nos contributions apportées dans ce domaine par rapport aux stratégies relatives.

5.6.4 Négociation avec désengagement pour l'allocation de ressource dynamique dans le Cloud

Les plateformes de cloud computing permettent aux consommateurs de louer par programmation les ressources distantes qu'ils désirent. Ces plateformes se basent sur la technologie de virtualisation qui est une des composantes principales du cloud pour faire apparaître les ressources au consommateur comme des composantes matériels brutes, tels que les machines, les périphériques de stockage, les capteurs ou les liens de réseau. Amazon a investi sur des solutions cloud computing et a défini comme objectif de réaliser des profits en louant les ressources pour moins qu'il en coûte aux consommateurs de les acheter.

Alors que GENI [3] est initialement utilisée comme une plate-forme à but non lucratif qui a des objectifs portés par des raisons académiques. Elle permet d'allouer les ressources à partir de plusieurs fournisseurs, ce qui rend l'allocation centralisée difficile à gérer lorsque le nombre de fournisseurs passent à l'échelle.

Dans ce travail, nous nous adressons à des problèmes d'allocations de ressources qu'on retrouve dans les plateformes de cloud computing, traités auparavant par Bo.A et.al [16]. Ces problèmes sont caractérisés par les caractéristiques suivantes : 1) de multiples agents d'intérêt personnel fournissent ou consomment plusieurs types de ressources, 2) les consommateurs entrent et quittent dynamiquement le marché, 3) les consommateurs ont une certaine flexibilité limitée quant au moment où ils ont besoin de ressources et 4) un seul fournisseur peut ne pas satisfaire les exigences en ressources des consommateurs. Les trois premières caractéristiques sont répondues dans les problèmes d'allocation de ressources au sein de plateforme de cloud computing. La motivation de 4) est une conséquence de l'infrastructure de GENI qui fournit ses ressources aux consommateurs à partir de fournisseurs tiers répartis à travers le territoire des Etats Unis.

Comme nous traitons des problèmes d'allocation de ressources dans les plateformes de cloud computing, les contrats seront négociés à des fins de locations de ressources et non pas d'achat. A cet effet, parmi les termes du contrat défini, l'agent doit spécifier la durée sur laquelle il utilisera les ressources demandées ainsi que le temps de début d'approvisionnement. De plus les agents négocieront leurs ressources de façon concurrentielles, et peuvent se désengager des accords temporaires pour poursuivre des alternatives potentiellement plus lucratives. Notez qu'un tel type de négociation implique plusieurs paramètres à spécifier, ce qui est difficile de les fixer de façon

exogène tout en générant à l'issue de la négociation une solution optimale. Au lieu de cela, Bo.A et.al [16] ont considéré le rôle de fixer les pénalités de désengagement ainsi que le prix d'offre du contrat par les agents concernés par la négociation, ce qui permet de faire distribuer la charge de travail du problème de l'allocation de ressources à l'ensemble des participants du marché.

En se basant sur ce mécanisme de négociation [16], nous avons proposé trois variantes qui ne sont qu'une extension de ce dernier. A savoir, un mécanisme basé-confiance qui prend en compte un paramètre supplémentaire lors de sa décision, qui est une sorte d'identité que chaque agent bâti au fil des expérimentations. Le champ d'application de ce mécanisme est approprié spécifiquement lorsque la transaction effectuée implique un coût qui revient au propriétaire de la plate-forme, de ce fait le contexte encourage les agents à restreindre au minimum possible leurs fréquences de négociations. Une deuxième proposition qui est aussi basée sur le même mécanisme consiste à explorer l'espace d'accords de l'acheteur de manière plus approfondie en générant et évaluant plus d'une offre à chaque vendeur. Et finalement, une troisième proposition qui adresse le problème du WDP au sein d'une négociation automatisée itérative en proposant un meilleur rapport utilité/complexité computationnelle.

La négociation avec incertitude est à la fois un des problèmes le plus difficile dans la littérature de la négociation [45], et est l'application clé de la négociation pour des problèmes réels tels que le cloud computing. La littérature a fourni auparavant des négociations impliquant de l'incertitude, mais c'était majoritairement avec les négociations bilatérales qui considèrent un seul type d'incertitude tel que la date limite [45] ou le prix de réserve [13]. Ce n'est que récemment que des travaux relatifs aux notre ont été adressés [16], où la négociation est de type plusieurs-à-plusieurs impliquant plusieurs types d'incertitudes tels que le prix de réserve et la date limite de l'acheteur, la stratégie de l'acheteur et le vendeur et la situation de négociation du vendeur. Ces incertitudes ne font que complexifier le problème d'allocation de ressources, ajouter à cela la dynamique inhérente qui exige les marchés du cloud computing. Face à la complexité de ce problème, Bo.A et.al [16] ont proposé des heuristiques qui génèrent des approximations de qualité dont on s'est basée afin de concevoir des heuristiques à notre tour qui génèrent globalement de meilleur résultats dans la phase expérimentale par rapport aux approches relatives. Ces heuristiques ont été motivées par le constat que modéliser les facteurs interdépendants de ce problème et déterminer les meilleures décisions que chaque agent doit entreprendre de manière exogène revient à une

optimisation combinée intraitable. Donc ces heuristiques ont été considérées comme une meilleure alternative en connectant les facteurs interdépendants du problème de manière indirect afin d'approximer la prise des décisions des agents, comme par exemple faire des offres en réagissant à l'évolution de l'état de la négociation tout en tenant compte des contraintes de temps, la concurrence des ressources, et le coût des ressources.



PARTIE 4 :

CONTRIBUTIONS

Chapitre 6 : Allocation de ressources basée confiance

La négociation automatisée pour les contrats de location de ressources dans le Cloud pose de nombreux défis. Puisque nous incorporons une identité à chaque participant, les utilités futures joueront un rôle majeur dans les stratégies de ces derniers au sein du marché du cloud computing. En effet, à la différence des stratégies d'agents traditionnels, nos stratégies doivent prendre en considération deux éléments supplémentaires, à savoir, les utilités futures impliquant la réputation des identités des agents ainsi que les frais de transactions dans un marché destiné à l'allocation de ressources dans le cloud computing. Le but de cette proposition est de concevoir de telles stratégies qui empêchent toutes pratiques qui peuvent causer une injustice en raison de certains inconvénients dans le protocole d'allocation des ressources, aidant ainsi certains participants à prendre des avantages injustes sur les autres. Des expériences comparatives avec le modèle non-confiance montrent l'efficacité de notre approche en termes d'équité et de bien-être social.

6.1 Introduction

L'incorporation de la confiance dans le mécanisme de négociation a été motivée par les inconvénients trouvés dans le protocole de négociation du cloud proposé dans [16], qui sont énumérés et résolus comme suit :

1. Réduire le nombre de transactions effectuées dans le mécanisme puisque chaque transaction implique un coût.
2. Réduire le taux d'annulation des accords [16] de l'acheteur car il pourrait subir des pertes par rapport au vendeur.
3. Inciter la réduction du taux de désengagement [16] effectuée du vendeur puisque ce dernier pourrait ne pas compenser entièrement l'autre partie du contrat.

La première raison sera réalisée en laissant les acheteurs et les vendeurs ajuster leurs fréquences de négociation en fonction de leurs réputations publiques qui seront décrites plus loin sur quoi elles sont basées, plutôt que d'avoir une fréquence de négociation ordinaire comme c'est le cas dans [16]. En effet, une fréquence régulière dans le marché qui va facturer chaque transaction effectuée pourrait être très pénalisante et aura comme répercussion la diminution du bien être sociale.

Notre deuxième motivation à inclure la confiance dans le mécanisme de négociation provient de l'incertitude qui est autour du contrat. Comme il arrive que le contrat utilisé dans ce protocole est un contrat d'engagement à niveau [96], le vendeur ne sait pas lequel des contrats contractés sera désengagé par l'autre partie et ceux qui ne le seront pas. Naïvement, traiter chaque contrat en fonction de leurs revenus potentiels sans tenir compte de cette incertitude pourrait conduire le vendeur à des pertes importantes. Plus formellement, puisqu'il y a deux types de contrats, l'un sans engagement appelé "accord temporaire" qui pourrait être désengagé des deux parties gratuitement, et l'autre avec un engagement où un désengagement encourt une pénalité, de tels accords sont appelées accords finals [16]. En raison de la capacité limitée des ressources à la disposition du vendeur, il doit parfois choisir de garder l'un d'eux et de se désengager de l'autre. S'il décide de garder un accord final et d'annuler un accord temporaire, ce sera une stratégie d'aversion au risque d'un côté, mais en revanche, il pourrait laisser une meilleure opportunité lui échapper. De l'autre côté, si le vendeur décide de se désengager d'un accord final en vue de garder un accord temporaire ayant un revenu potentiel supérieur dans l'espoir que ce dernier sera conclu par la suite. Cela pourrait se révéler être un mauvais choix dans le cas où cet accord provisoire serait désengagé par son soumissionnaire gratuitement. Donc, dans ce scénario, le vendeur a perdu deux contrats : l'un avait été désengagé par lui-même en payant une pénalité et l'autre gratuitement de la part de l'acheteur. Pour faire face à cette incertitude quant à savoir si oui ou non le contrat sera désengagé, la réputation de l'acheteur aidera le vendeur dans son processus de décision sur quel accord garder et lequel libérer. La réputation de l'acheteur sera une indication de son comportement en termes de son taux de désengagement.

La troisième et dernière motivation est que la politique de compensation adoptée dans [16] dans le cas où le vendeur vient à se désengager d'un accord ne serait pas considérée comme une indemnisation totale vis-à-vis de l'acheteur. En effet, comme nous supposons que les acheteurs sont « single-minded » (uniques d'esprits), ils doivent avoir toutes leurs ressources exigées contractées avant de démarrer leurs tâches, ce qui est un indicateur qu'ils ont négociés avec succès leurs ressources. Le désengagement d'un vendeur pourrait être la cause de l'acheteur à ne pas contracter toutes ses ressources. Dans de tels cas, l'acheteur a deux options : dans le cas où l'acheteur a atteint la date limite, il doit alors quitter la négociation avec une utilité négative puisque les ressources contractées jusqu'à maintenant n'ont pas été suffisantes pour démarrer la tâche et ont servi donc à rien. La deuxième option est lorsque l'acheteur a atteint la date d'approvisionnement

mentionnée dans le contrat mais pas encore atteint la date limite finale de la négociation. Dans ce cas, puisqu'il ne peut pas démarrer sa tâche directement, alors, selon sa stratégie, il devra se désengager de toutes les ressources contractées jusqu'à maintenant et les renégocier pour un temps d'approvisionnement ayant lieu plus tard. Dans les deux cas, cela coûtera à l'acheteur de se désengager de plusieurs contrats dont leurs sommes est généralement supérieur à la somme de pénalité que devra payer le vendeur qui lui a causé l'échec de sa négociation. Afin de diminuer de telles occurrences, l'approche proposée permet à l'acheteur de ternir la réputation du vendeur s'il estime que ce dernier a abusé de l'utilisation de la caractéristique de désengagement, ou inversement, évaluer sa réputation de manière positive s'il n'a pas abusé. Nous avons choisi CloudSim [3] comme notre banc d'essai afin de mener nos expériences, en raison des nombreuses fonctionnalités qui lui sont intégrées. Entre autres, il a une API conviviale qui gère la communication entre les entités (agents).

6.1.1 Description du problème

Le problème que nous adressons consiste à un ensemble d'acheteurs et de vendeurs. Chaque acheteur n'entrera sur le marché de négociation qu'après avoir générer sa tâche χ [16] qui a comme attributs :

- ❖ R_b : précise tous les types de ressources avec leurs quantités demandées requis par l'acheteur.
- ❖ Temps de génération $tg(b)$: indique le temps de génération de la tâche χ .
- ❖ $q(R_b, r)$: La quantité de ressource $r \in R_b$, demandée par l'acheteur.
- ❖ La période $pd(b)$: représente le temps estimé par l'agent acheteur nécessaire à l'utilisation des ressources demandées R_b .
- ❖ Le temps de début au plus tôt $est(b)$: où l'agent acheteur ne peut pas démarrer la tâche avant $est(b)$.
- ❖ La date limite $dl(b)$: qui indique le dernier délai pour lancer la tâche. Si $dl(b) > est(b)$, cela permet au vendeur d'avoir un intervalle de temps $[est, dl]$ afin de choisir le temps de début d'approvisionnement de ressources qui lui convient le plus.

- ❖ La valeur $v_b(t)$: représente la valeur de l'acheteur b qui lui est attribué à la complétion de sa tâche. Cette dernière est une fonction du temps de complétion t qui décroît, linéairement, au fur et à mesure que le temps passe.

Notre mécanisme de négociation proposé prend en compte des attributs additionnels qui sont extraits de l'approche personnalisée [114] ce qui permet d'avoir quatre catégories d'agents dans le processus de négociation : acheteur, vendeur, conseiller acheteur et conseiller vendeur. Le conseiller de l'acheteur est un acheteur qui fournit les notations du vendeur quand il lui est demandé auprès de son compétiteur, et le même principe est appliqué pour le conseiller vendeur. En d'autres termes, chaque acheteur devra jouer à la fois le rôle de l'acheteur et de conseiller et la même chose pour le vendeur.

Le calcul de la confiance des vendeurs est inspiré du travail réalisé dans [114]. Voici la notation utilisée dans cette approche :

$R_{pub}(s)$: représente la réputation publique du vendeur s au sein de la communauté des acheteurs.

$R_{pri}(s)$: représente la réputation privée du vendeur s chez l'acheteur b .

$Tr(s)$: représente la confiance du vendeur s chez b .

$R_{pub}(b)$: représente la réputation publique de l'acheteur b au sein de la communauté des vendeurs.

$R_{pri}(b)$: représente la réputation privée de l'acheteur b chez le vendeur s .

$Tr(b)$: représente la confiance de l'acheteur b chez le vendeur s .

$R_{pub}(A)$: représente la réputation publique du conseiller (acheteur/vendeur) par rapport à la communauté des (acheteurs/vendeurs).

$R_{pri}(A)$: représente la réputation privée du conseiller (acheteur/vendeur) chez le (acheteur/vendeur).

$Tr(A)$: représente la confiance du conseiller (acheteur/vendeur) par rapport à le (acheteur/vendeur).

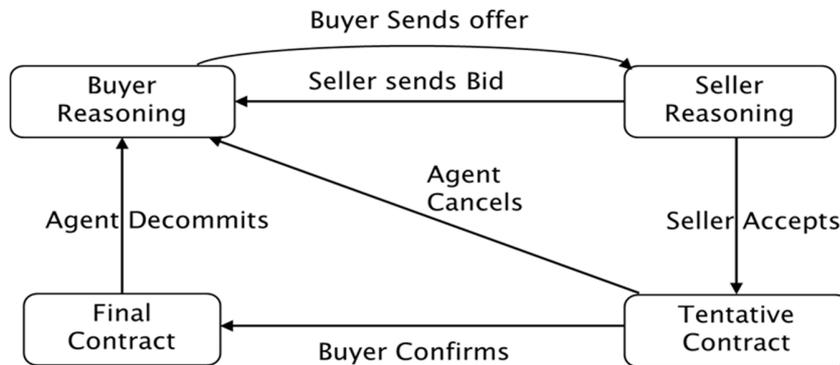


Figure 7 Machine à état fini pour le protocole de négociation [16]

6.1.2 Protocole de négociation

Les messages du protocole adopté de [16] sont :

- *RESOURCE_CHARACTERISTICS_REQUEST* : quand l'acheteur génère sa tâche, il lance ce message afin de recueillir tous les identifiants des différents vendeurs afin qu'il puisse communiquer avec eux par la suite. Ce message n'est envoyé qu'une seule fois durant la phase de négociation de l'acheteur.
- *RESOURCE_CHARACTERISTICS* : représente les caractéristiques d'un vendeur sur le marché.
- *BUYER_OFFER(o)* : représente l'offre de l'acheteur. L'offre contiendra les attributs suivant : $\langle pr, pe, \mathcal{R}, est, pd, dl \rangle$. Où pr est le prix d'offre, pe est la pénalité de désengagement, \mathcal{R} représente les ressources demandées par l'offre, est est le temps de début le plus tôt d'approvisionnement, pd est le temps d'usage des ressources et dl est la date limite d'approvisionnement.
- *ACCEPT_OFFER(o)* : Après que l'offre soit acceptée du côté du vendeur, l'acheteur évaluera l'offre à travers un algorithme d'évaluation, et décidera par la suite de confirmer ou pas le message d'acceptation du vendeur, si c'est le cas, l'accord deviendra final.
- *QUOTES(o)* : Lorsque le prix d'offre proposé par l'acheteur ne dépasse pas le prix de réserve du vendeur. Dans ce cas, le vendeur enverra pour chaque type de ressource demandée son prix de réserve.

- *REJECT_BID(o)* : Lorsque le prix de l'offre de l'acheteur est acceptable pour le vendeur, mais ne peut être gardé par ce dernier dû à son classement par rapport aux autres offres acceptées par le vendeur et à la limitation des ressources de ce dernier.
- *DECOMMIT(o)* : indique qu'un agent enverra un message de désengagement à l'autre agent correspondant, disant que l'offre o n'est plus considérée comme un accord final. Ce qui est à noter est que le désengagement implique de payer une pénalité à l'autre partie sous la condition que le temps courant t ne doit pas dépasser $est(o)$.
- *Cancel(o)* : si l'offre est dans un état d'accord temporaire, n'importe quel agent se désengagera sans avoir à payer une charge.
- *Confirm(o)* : Il est utilisé lorsque l'acheteur veut confirmer le message d'acceptation du vendeur de sorte que l'accord deviendra un accord final.
- *RUNNING_FINAL_AGREEMENT(o)* : est un message envoyé par le vendeur pour informer l'acheteur que o est devenue *RUNNING_FINAL_AGREEMENT*. Noter que, un *RUNNING_FINAL_AGREEMENT* est un accord final qui n'a pas été désengagé des deux côtés jusqu'à $est(o)$ et qui ne peut être désengagé par la suite.
- *CONTINUE_NEGOTIATION* : représente la période cyclique dans laquelle l'acheteur devra faire son prochain tour de négociation.
- *AGRT_TERMINATED* : c'est un message indiquant que la période de location des ressources a été écoulee.
- *END_OF_SIMULATION* : est un message indiquant à l'acheteur de quitter le marché. Elle se produit lorsque la période de négociation se termine.

La Figure 7 montre une description de la négociation impliquant les deux parties principales qui sont l'acheteur b et le vendeur s . Cette description est extraite du travail de recherche [16] :

L'état initial est "raisonnement acheteur", dans lequel b décide de la façon de présenter son offre. Après, b envoie l'offre à s , l'état se transforme en "raisonnement vendeur". Après, si s accepte l'offre, un *accord temporaire* sera alors établi. Dans le cas contraire, le vendeur envoie un « *quote* » (les prix acceptables) à l'acheteur. Si b confirme l'accord temporaire, la négociation passe à l'état "accord final". Les autres alternatives qui restent à b sont soit qu'il annule l'accord temporaire sans payer quoique ce soit ou se désengager de l'accord

final en payant une pénalité prédéterminée, leur négociation alors échouera et *b* redémarrera à l'état initial pour reformuler à nouveau son offre.

En intégrant l'action d'annulation dans le protocole, l'acheteur peut réserver plus de ressources qu'il pourrait réellement s'en procurer et tout cela sans rien dépenser, puisque l'acheteur pourrait annuler toutes les offres qu'il juge inutile aussi longtemps que ces dernières n'ont pas atteint leurs états finaux. Mais nous avons réglementé le taux d'annulation de l'accord en intégrant la confiance sur les identités des acheteurs et modifiant leurs stratégies par conséquence. Par ce moyen, les acheteurs seront moins encouragés à faire usage de la fonction d'annulation car ils verront leurs réputations publiques diminuées, et donc leurs prix d'offres seront moins valorisés parmi la communauté des vendeurs. En revanche, si l'acheteur vient à se désengager d'un accord final, sa réputation ne sera pas ternie puisque le paiement de la pénalité envers la victime (vendeur) sera considéré comme une compensation entière.

Comme conséquence à la permission de l'acheteur de prioriser ces partenaires commerciaux (vendeurs) en fonction de leurs confiances dans le cas où deux d'entre eux pourraient offrir la même proposition, les vendeurs seront encouragés à réduire leurs taux de désengagements de sorte qu'ils seront considérés plus prioritaires dans la communauté des acheteurs. En effet, la confiance du vendeur sera une indication de son comportement récent en termes de taux de désengagements, puisque le désengagement du vendeur pourrait causer des dommages qui ne sont pas totalement indemnisés sous le protocole de négociation de [16]³. Donc, pour rendre l'approche de négociation plus équitable que dans [16], en plus que le vendeur doit payer une pénalité en cas de désengagement, il verra sa réputation publique diminuée s'il abuse sur l'utilisation de cette fonctionnalité.

Enfin, comme nous supposons que les opérations sont effectuées sous une plate-forme qui enlève une partie de la somme d'argent transférée dans les opérations de négociations (fixé à 10 pour cent dans cette contribution), et que chaque accord d'engagement à niveau en lui-même détient une certaine incertitude, le défi sera alors comment faire face à cette incertitude tout en dépensant le moins possible de frais de transactions. De notre côté, nous nous sommes basé sur la confiance afin d'adresser ce défi au lieu d'opter pour la négociation de ressources redondantes [16] pour faire

³ Le troisième motif pour intégrer la confiance qui a été décrit dans la première section explique pourquoi le désengagement du vendeur pourrait causer des dommages qui ne seront pas entièrement compensés par les frais de pénalité.

face à l'incertitude du contrat. À la section résultat de cette contribution, nous avons montré que par l'incorporation de la confiance dans la stratégie de l'agent, nous pourrions réduire de façon significative le nombre de transactions, générant un meilleur bien être sociale et rendre l'approche proposée relativement plus équitable par rapport à [16].

6.1.3 Disposition de l'axe de temps

Avant de décrire l'agencement du temps de cette approche, nous soulignons deux différents types d'accords jouant un rôle important dans la façon dont les ressources doivent être approvisionnées du côté du vendeur :

- *L'accord partiel* : est un accord qui ne satisfait qu'une fraction des demandes de l'acheteur, ce qui contraint l'acheteur à synchroniser l'ensemble des offres partiaux afin que l'ensemble des ressources soient approvisionnées en même temps. Pour ce faire, un temps d'approvisionnement commun est requis, noté *estp*, qui est l'acronyme de « earliest start time for partial agreements ». Le *dip*, qui est l'acronyme de « deadline for partial agreements » devra être égale à *estp* pour restreindre le choix d'approvisionnement du vendeur à un seul point donné.
- *L'accord complet* : est un accord qui satisfait toutes les demandes de l'acheteur. Aucune synchronisation dans ce cas n'est requise du fait que l'offre en elle-même est indépendante. L'acheteur proposera comme temps d'approvisionnement l'intervalle [*est*, *dl*] au vendeur en cas où un contrat vient se concrétiser entre les deux partis.

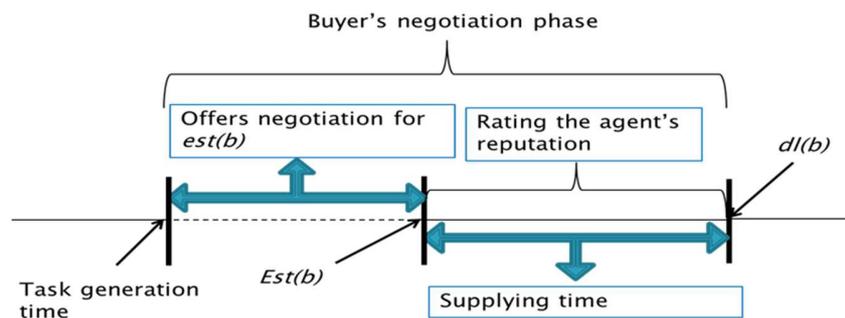


Figure 8 Agencement de l'axe de temps

La tâche principale de l'agent est d'avoir toutes ses ressources contractées par des accords finaux au temps *est(b)* comme cela est illustré par la figure 8, ce qui indique la conclusion du

contrat entre les deux parties (à ce stade, aucune des deux parties ne peut se désengager). Après que le contrat soit conclu, les ressources seront approvisionnées instantanément si l'accord est partiel puisque $estp = dlp$, où à un certain temps dans l'intervalle $[est, dl]$ au choix du vendeur. L'intervalle d'approvisionnement se rétrécit au fur et à mesure que l'acheteur re-planifie son $est(b)$. Une replanification du $est(b)$ sera automatiquement suivi par le désengagement et l'annulation de l'ensemble des offres partiaux déjà contractées jusqu'à maintenant.

En ce qui concerne les évaluations entre les agents, une note négative est donnée à l'une des parties si elles dépassent un certain seuil en termes de taux de désengagement côté vendeur et en termes de taux d'annulation côté acheteur. Une note positive est donnée si les deux parties sont parvenues à conclure un contrat en ne dépassent pas le seuil défini. Si les deux parties ne dépassent pas le seuil et ne concluent pas le contrat, une note nulle serait mutuellement attribuée pour les deux parties, ce qui signifie qu'aucune information relative à la réputation de l'agent ne peut être extraite de cette rencontre.

6.2 Stratégie de négociation de l'acheteur

Nous allons décrire la stratégie de l'acheteur, qui est décomposée en cinq sous-stratégies :

1. Stratégie concernant l'ensemble d'accords.
2. Stratégie de confirmation d'offres
3. Stratégie de désengagement
4. Stratégie de préférences envers les vendeurs
5. Stratégie de fréquence de négociation

La première est liée à combien d'ensemble d'accords l'acheteur doit négocier et quels types d'accords y inclure dans ces ensembles.

La deuxième stratégie locale a trait sur les conditions sous lesquelles un accord se verra confirmer par l'acheteur.

La troisième stratégie définit les conditions sous lesquelles le désengagement s'opère.

La quatrième stratégie concerne la préférence de l'acheteur envers les vendeurs matérialisés par leurs confiances respectives.

Et finalement, la cinquième stratégie spécifie la fréquence de négociation de l'acheteur vis-à-vis des vendeurs qui va dépendre de sa réputation publique.

6.2.1 Stratégie de négociation de l'acheteur concernant l'ensemble des accords

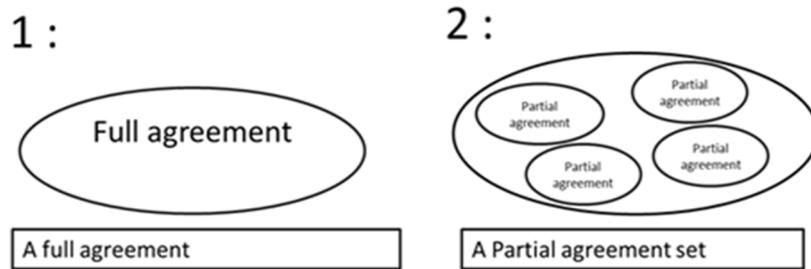


Figure 9 Stratégie de négociation de l'acheteur sur l'ensemble d'accords.

Comme décrit par la Figure 9 ci-dessus, l'acheteur aura comme choix en termes de constitution d'ensemble d'accords deux alternatives, chacune d'elles doit satisfaire l'ensemble des ressources demandées. La première est un accord complet qui peut se formuler en cas où l'ensemble des ressources de l'acheteur se trouve chez un seul fournisseur parmi l'ensemble des fournisseurs, et la deuxième possibilité qui s'offre à lui est lorsqu'il ne peut pas contracter toutes ses ressources via un seul fournisseur, dans ce cas, il contractera un ensemble d'accords partiels ayant un « *estp* » commun.

6.2.2 Stratégie de confirmation de l'acheteur

L'acheteur après avoir formulé une offre pourrait recevoir un message d'acceptation de la part du vendeur comme il est montré dans la Figure 7, pour lequel l'acheteur doit décider de confirmer ou non cette offre acceptée. Dans cette proposition, la stratégie de confirmation de l'acheteur est la même que celle dans [16]. Elle consiste à confirmer chaque offre si son contenu est disjoint avec le contenu de toutes les offres précédemment confirmées. Notez que l'acheteur enverra des offres simultanément dans chaque tour de négociation à l'ensemble des vendeurs qui vendent une partie des ressources manquantes nécessaires à l'acheteur.

6.2.3 Stratégie de désengagement de l'acheteur

Dans le prolongement de la section précédente, l'une des causes qui permettent à l'acheteur d'annuler une offre est quand il applique sa sous-stratégie de confirmation. En fait, si l'acheteur n'a

pas pu confirmer un accord temporaire lorsque le contenu de ce dernier n'est pas disjoint avec l'ensemble des agréments finaux, il doit alors l'annuler.

Une autre condition pour laquelle l'acheteur devra se désengager de l'offre proposée est lorsque le temps de négociation a atteint le point $est(b)$ et que toutes ces ressources n'ont pas encore été liées avec des accords finaux.

De plus, l'acheteur fixera des points intermédiaires nommés T_{bk} 's [16] dans lesquels les mêmes conditions de désengagements définis dans les points $est(b)$'s leurs seront appliquées.

$$T_{bk} = \frac{(dl(b)-t)}{k} + t; \tag{17}$$

Cette formule a comme paramètre k lequel est une variable qui doit être préalablement prédéfinie (fixé à 4 dans cette contribution) et qui se décrémente chaque fois que le temps courant de simulation atteint T_{bk} .

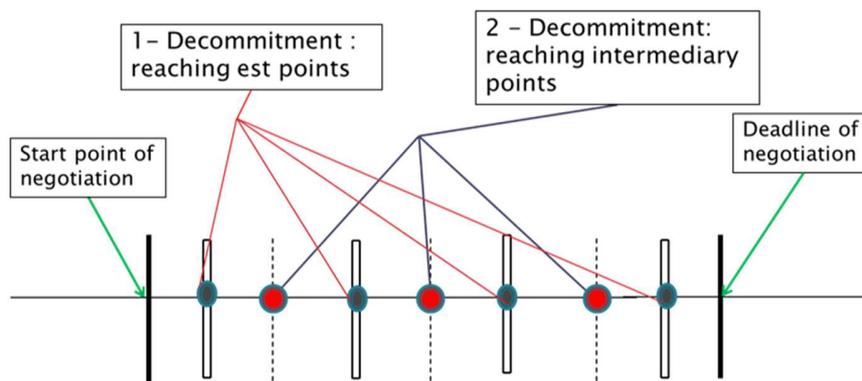


Figure 10 Stratégie de désengagement de l'acheteur

Comme la Figure 10 le montre, les points bleus représentent les différents est 's possible de l'acheteur. Les auteurs dans [16] ont conçus les points intermédiaires pour être planifiés entre deux est 's consécutives montrés par des points rouges (T_{bk} 's) dans la figure d'au-dessus. Ces points rouges sont planifiés de tel sorte afin de permettre à l'acheteur d'anticiper l'échec de sa négociation en faisant une re-planification "backtraking" anticipée sur l'exécution de la tâche courante si les mêmes conditions de désengagement définies dans les points est 's soient satisfaites dans ces points.

La mise à jour du $estp(b)$ [16], en cas où le temps de négociation atteigne soit un point bleu et la tâche ne peut être exécutée ou soit un point rouge et l'acheteur n'a pas encore acquis l'ensemble des ressources par le moyen d'accords finaux, est comme suit :

$$estp = \max\{t + \sigma, est(b)\} \quad (18)$$

Où σ est un paramètre non-négative.

6.2.4 Stratégie de préférence de l'acheteur sur les vendeurs

Dans l'article [16], l'acheteur envoie une offre à chaque vendeur qui n'est pas précédemment lié avec lui par un contrat. L'acheteur générera une demande de ressources consistant par l'intersection entre l'ensemble des ressources requis jusqu'à maintenant avec les ressources disponibles offerte par le vendeur. Le premier vendeur envoyant un message d'acceptation à l'acheteur sera automatiquement confirmé, tandis que les messages d'acceptation qui suivent devront ne contenir que de ressources non déjà confirmées pour être acceptées à leurs tours.

Nous avons étendu le mécanisme décrit dans [16], pour la classification des préférences de l'acheteur vis-à-vis des vendeurs en utilisant l'approche dans [114]. De cette façon, l'acheteur priorisera les vendeurs avec une forte confiance que les vendeurs avec une faible confiance, acceptant les derniers comme dernier ressort ce qui lui permettra d'ajouter plus de sécurité lors de la réservation de ces ressources.

La confiance de chaque vendeur est calculée selon [114] en suivant les formules suivantes :

$$Tr(S) = w'R_{pri}(S) + (1 - w')R_{pub}(S) \quad (19)$$

$$w' = \begin{cases} \frac{N_{all}^B}{N_{min}} & \text{si } N_{all}^B < N_{min} \\ 1 & \text{sinon} \end{cases} \quad (20)$$

$$N_{min} = -\frac{1}{2\varepsilon^2} \ln \frac{1-\gamma}{2} \quad (21)$$

Le poids w' mesure la fiabilité de la réputation privée du vendeur par rapport à l'acheteur qui l'évalue. N_{all}^B donne le nombre totale d'annotations fournie par l'acheteur B pour le vendeur S , laquelle est la somme des annotations pour chaque fenêtre de temps élémentaire. Nous avons pris pour notre part les cinq dernières fenêtres de temps élémentaires en considération, lesquelles représentent dans notre cas, les cinq dernières exécutions du processus de négociation.

N_{min} représente le nombre minimale de notations requises pour l'acheteur B pour être confiant de la valeur de la réputation privée du vendeur S , où ε est le niveau maximal d'erreur qui peut être accepté par l'acheteur B , et γ est la mesure de confiance. On constate que si la quantité d'information extraite de la réputation privée du vendeur est considérée comme significative, ou en d'autre terme, si $N_{all}^B > N_{min}$, l'acheteur se fiera entièrement à cette dernière pour évaluer sa confiance.

$$R_{pri}(S) = \frac{\sum_{i=1}^n N_{pos,i}^B \lambda^{i-1} + 1}{\sum_{i=1}^n (N_{pos,i}^B + N_{neg,i}^B) \lambda^{i-1} + 2} \quad (22)$$

Où $N_{pos,i}^B$ représente le nombre d'annotations positives de l'acheteur B en faveur du vendeur S dans la fenêtre de temps élémentaire i . Nous considérons une note positive, ayant une valeur de « 1 », si les deux parties du contrat ont conclu un contrat sans dépasser un certain seuil de désengagements/annulation, noté *threshold* :

$$threshold = \mu * ((dl - est) / avIter) \quad (23)$$

μ est une variable prédéfinie, que nous avons fixé dans nos simulations à 0.8 et *avIter* est la valeur moyenne que la variable itération pourrait prendre, dans notre cas, la valeur 3. Une note négative est obtenue quand l'une des parties impliquées dans le contrat dépasse le seuil défini ci-dessus. Le dernier cas possible est une notation "nulle" qui se produit lorsque l'une des parties impliquées ne dépasse pas le seuil prédéfini et aucun contrat n'a été conclu entre eux.

Le facteur λ est un facteur d'oubli qui traite du changement possible du comportement du vendeur sur le temps en assignant moins de poids pour les annotations anciennes que les annotations récentes.

$$R_{pub}(S) = \frac{\left[\sum_{j=1}^k \sum_{i=1}^n D_{pos,i}^{Aj} \lambda^{i-1} \right] + 1}{\left[\sum_{j=1}^k \sum_{i=1}^n (D_{pos,i}^{Aj} + D_{neg,i}^{Aj}) \lambda^{i-1} \right] + 2} \quad (24)$$

$$D_{pos,i}^{Aj} = \frac{2Tr(A_j)N_{pos,i}^{Aj}}{(1-Tr(A_j))(N_{pos,i}^{Aj} + N_{neg,i}^{Aj}) + 2} \quad (25)$$

$$D_{neg,i}^{Aj} = \frac{2Tr(A_j)N_{neg,i}^{Aj}}{(1-Tr(A_j))(N_{pos,i}^{Aj} + N_{neg,i}^{Aj}) + 2} \quad (26)$$

Où $D_{pos,i}^{A_j}$ et $D_{neg,i}^{A_j}$ sont des fonctions d'actualisation. $Tr(A_j)$ est la confiance du conseiller j dans l'esprit de l'acheteur ou le vendeur en question et qui est calculée comme suit :

$$Tr(A) = wR_{pri}(A) + (1 - w)R_{pub}(A) \quad (27)$$

$$w = \begin{cases} \frac{N_{all}^A}{N_{min}^A} & \text{si } N_{all}^A < N_{min}^A \\ 1 & \text{sinon} \end{cases} \quad (28)$$

$$N_{all}^A = \sum_{i=1}^m N_{S_i}^A \quad (29)$$

$$R_{pri}(A) = \frac{\alpha}{\alpha + \beta} \quad (30)$$

$$\alpha = N_{pos}^A + 1 \quad (31)$$

$$\beta = N_{all}^A - N_{pos}^A + 1 \quad (32)$$

Où $N_{S_i}^A$ représente le nombre d'annotations de paires correspondantes entre le conseiller A et l'acheteur B pour un vendeur S_i durant les cinq dernières fenêtres du temps élémentaires. Dans notre cas, nous considérons deux annotations comme paire correspondante, si la note du conseiller et celle de l'acheteur sont dans la même fenêtre de temps élémentaire pour un vendeur communément annoté. N_{pos}^A représente le nombre d'annotations positives du conseiller A durant les cinq dernières fenêtres de temps élémentaires. Une annotation du conseiller est positive par rapport à un acheteur donné si son évaluation correspond à celle de l'acheteur dans la même fenêtre de temps élémentaire.

$$R_{pub}(A) = \frac{\alpha'}{\alpha' + \beta'} \quad (33)$$

$$\alpha' = N_f^A + 1 \quad (34)$$

$$\beta' = N_{all}^A - N_f^A + 1 \quad (35)$$

Où N_{all}^A représente toute les évaluations fournies par le conseiller A , et N_f^A sont le nombre d'évaluations considérées comme équitables. Une évaluation est considérée comme équitable si elle est proche de la moyenne des annotations liées de la communauté des conseillers. Dans notre cas, nous assumons que tous les conseillers reporteront véridiquement leurs annotations. Par conséquent, la réputation publique de chaque conseiller sera toujours égale à « 1 ».

6.2.5 Stratégie de fréquence de négociation de l'acheteur

La réputation publique de l'acheteur reflétera son comportement récent en termes d'annulation d'accord, plus précisément, dans ses cinq dernières fenêtres de temps élémentaires, et il augmentera/diminuera sa fréquence de négociation selon l'augmentation/diminution de sa réputation.

Deux paramètres sont liés à la fréquence de négociation de l'acheteur et donc à sa réputation publique, qui sera mise à jour à la fin de chaque expérience de négociation de chaque acheteur. Le premier est σ :

$$\sigma = \sigma + \left(PubRep_{diff} * (\sigma/1.5) \right) \quad (36)$$

Où :

$$PubRepB_{diff} = preB_{PubRep} - curB_{PubRep} \quad (37)$$

$curB_{PubRep}$ signifie la réputation publique de l'acheteur et $preB_{PubRep}$ la réputation publique précédente de ce dernier.

$$iteration = iteration + \left(PubRepB_{diff} * (iteration/1.8) \right) \quad (38)$$

Où σ et $iteration$ sont limitées dans les intervalles suivants :

$$1 \leq iteration \leq 6 \quad (39)$$

$$4 \leq \sigma \leq 12 \quad (40)$$

Nous signifions par la variable $iteration$ le pas de temps pour lequel l'agent attendra pour passer à la prochaine itération de négociation et la variable σ jouera le rôle de réajustement du est de l'acheteur.

Les équations (36) et (38) signifient que l'acheteur va augmenter sa fréquence de négociation si sa réputation publique a été augmentée aussi, ce qui est une stratégie risquée d'une part, puisque l'acheteur pourrait facilement dépasser le seuil d'annulation de contrats et par conséquent il sera pénalisé moralement en voyant sa réputation publique diminuée. Mais l'avantage de l'utilisation de cette stratégie risquée est d'entraîner une diminution de la période de désengagement, diminuant ainsi le temps pour les concurrents commerciaux de proposer des offres plus élevées, ce qui

résultera d'une diminution des chances de l'offre de l'acheteur de se voir désengager par le vendeur.

Toutefois, si la réputation publique de l'acheteur décroît, l'acheteur adoptera une stratégie d'aversion au risque afin d'être moins exposé aux notations négatives sur sa réputation publique d'une part, mais ses chances de réussir à conclure un contrat seront plus compromettante d'autre part, puisque la période de désengagement permise au vendeur deviendra plus large.

6.2.6 Pseudo algorithme de la stratégie de l'acheteur

Dans cette sous-section, nous présentons le pseudo-algorithme qui résume en un seul endroit la stratégie de l'acheteur.

Algorithme 1 : Stratégie de négociation de l'acheteur

o : représente une offre simple

$\mathcal{T}o$: représente un accord temporaire

$\mathcal{F}o$: représente un accord final

// *Phase d'initialisation* : doit être exécutée qu'une fois durant l'entrée de l'acheteur sur le marché

$\mathcal{L}\mathcal{A}_b = \emptyset$ représente l'ensemble d'accords finaux complet

$\mathcal{L}\mathcal{T}\mathcal{A}_b = \emptyset$ représente l'ensemble d'accords temporaires complet

$\mathcal{P}\mathcal{A}_b = \emptyset$ représente l'ensemble d'accords finaux partiels

$\mathcal{P}\mathcal{T}\mathcal{A}_b = \emptyset$ représente l'ensemble d'accords temporaires partiels

Soit t représente le temps de simulation (initialement, $t = \mathbf{t}g(\mathbf{b})$).

Soit vt le temps d'évènement par lequel l'acheteur peut exécuter sa stratégie (initialement, $vt = \mathbf{t}g(\mathbf{b})$).

Soit $estp = est(b)$; $dlp(b) = dl(b)$; $k = 4$;

$\mathcal{C}\mathcal{R}_b = \mathcal{R}_b$: où $\mathcal{C}\mathcal{R}_b$ est la quantité courante de ressources demandées par l'acheteur.

task : une variable booléenne, initialisée à faux pour montrer que la tâche n'a pas encore démarrée, sinon elle prend la valeur vraie.

\mathcal{R}_s : les ressources disponibles pour la vente par le vendeur

LsTr : La liste des vendeurs triée par ordre décroissant de leurs valeurs $Tr(s)$.

// *Fin de la phase d'initialisation*

DecCntFrS (id) : une fonction qui prend comme paramètre l'identité du vendeur et retourne le nombre de désengagement fait par ce dernier.

CondCtrS (id) : une fonction booléenne qui prend comme paramètre l'identifiant du vendeur id, et retourne vrai s'il existe un contrat qui a été conclu avec ce dernier, sinon il retourne faux.

InNegWithSeller (id) : est une fonction qui retourne vrai chaque fois que l'acheteur a réalisé un contrat avec un vendeur identifié par le paramètre id.

// *La boucle principale de la stratégie de l'acheteur.*

tant que $(t < dl(b))$ **et** $(vt = t)$ **et** $(not\ task)$ **faire**

pour chaque $s \in LsTr$ **tel que** $\mathcal{R}_s \cap \mathcal{C}\mathcal{R}_b \neq \emptyset$ **et** $InNegWithSeller(s) = faux$ **faire**

 envoyer une offre **contenant** $(\mathcal{R}_s \cap \mathcal{C}\mathcal{R}_b)$ ressources au vendeur s .

fin

si vendeur s envoie quotes **alors**

 mettre à jour le « bid set »;

fin

```

si  $s$  accepte  $o$  ou annule a  $\mathcal{T}o$  alors
| mettre à jour l'ensemble d'agrément temporaires selon le type d'offre, i.e., complet ou partiel
fin
si  $s$  se désengage d'un  $\mathcal{F}o$  alors
| mettre à jour DecCntFrS( $s$ );
| si  $DecCntFrS(s) > threshold$  alors
| | évalué la réputation de  $s$  négativement;
| fin
fin
si  $s$  envoi  $\mathcal{T}o \in \mathcal{LTA}_b$  et  $(\mathcal{PA}_b \cup \mathcal{LA}_b) = \emptyset$  alors
| enlever  $\mathcal{T}o$  de  $\mathcal{LTA}_b$  et le placer à  $\mathcal{LA}_b$  ;
|  $\mathcal{CR}_b = \mathcal{CR}_b - \mathcal{T}o$ ;
fin
sinon si  $s$  envoi  $\mathcal{T}o \in \mathcal{PTA}_b$  et  $\mathcal{LTA}_b = \emptyset$  et  $(\mathcal{T}o \cap \mathcal{PA}_b = \emptyset)$  alors
| ajouter  $\mathcal{T}o$  à  $\mathcal{PA}_b$ ;
|  $\mathcal{CR}_b = \mathcal{CR}_b - \mathcal{T}o$ ;
fin
si  $(t > estp)$  alors
| se désengager et annuler les accords finaux et temporaires ;
| fixer  $\mathcal{LA}_b = \mathcal{LTA}_b = \mathcal{PA}_b = \mathcal{PTA}_b = \emptyset$ ;  $\mathcal{CR}_b = \mathcal{R}_b$ ;
| fixer  $estp = \max\{t + \sigma, est(b)\}$ ;
fin
si  $(t > T_{bk})$  alors
|  $k --$ ;  $T_{bk} = (dl(b) - t)/k + t$ ;
| fixer  $\mathcal{LA}_b = \mathcal{LTA}_b = \mathcal{PA}_b = \mathcal{PTA}_b = \emptyset$ ;  $\mathcal{CR}_b = \mathcal{R}_b$ ;
| fixer  $estp = \max\{t + \sigma, est(b)\}$ ;
fin
 $vt = t + iteration$ ;
fin
Annuler l'ensemble des accords temporaires ;
Se désengager de chaque agrément final jugé inutile : la pénalité de désengagement est moins élevée que son prix d'offre ;
pour chaque  $s \in LsTr$  tel que  $CondCtrS(s) = true$  faire
| si  $DecCntFrS(s) \leq threshold$  alors
| | évalué la réputation de  $s$  positivement;
| fin
fin
 $\sigma = \sigma + (PubRep_{diff} * (\sigma/1.5))$ ;
 $iteration = iteration + (PubRepB_{diff} * (iteration/1.8))$ ;

```

6.3 La stratégie du vendeur

Le processus de classification des offres est lancé chaque fois que le vendeur reçoit une nouvelle offre à partir d'un acheteur qui doit cependant excéder le seuil $\phi(o)$ [16] définit ci-dessous, sinon l'offre sera simplement rejetée :

$$\phi(o) = \sum_{r \in R(o)} q(R(o), r) c_s(r) (1 + \psi(r)) pd(o) \quad (41)$$

En ce qui concerne le processus de tri des offres en lui-même, au lieu d'adopter une stratégie myopique comme décrit dans [16], le vendeur tiendra compte au-delà des revenus potentiels de l'offre, sa valeur de réputation publique au sein de la communauté des acheteurs, symbolisée par $R_{pub}(S)$, ce qui indique son récent comportement en termes de taux de désengagement et qui a une relation sur le degré de son hiérarchisation auprès de la communauté des acheteurs.

La deuxième chose qu'il prend en considération est le niveau de confiance du soumissionnaire au sein de la communauté des vendeurs, notée par $Tr(B)$, en d'autres termes, quel est le taux potentiel de dommages non-intégralement indemnisés récents causés par l'acheteur B envers les vendeurs ?

Les offres sont ensuite classées par ordre décroissant :

$$rv(o) = \begin{cases} (pr(o) - cost(o)) * ((Tr(B) + \tau)/R_{pub}(S)) + ((Tr(B) + R_{pub}(S) + \tau + 2) * pe(o)) & \text{si } o \in \mathcal{A}_s \text{ et } pr > pe \\ (pr(o) - cost(o)) * ((Tr(B) + \tau)/R_{pub}(S)) + ((Tr(B) + R_{pub}(S) + \tau + 1) * pe(o)) & \text{si } o \in \mathcal{A}_s \text{ et } pr \leq pe \\ (pr(o) - cost(o)) * (Tr(B) + R_{pub}(S)) + ((Tr(B) + R_{pub}(S)) * pe(o)) & \text{sinon} \end{cases} \quad (42)$$

Où \mathcal{A}_s est l'ensemble des accords finaux du vendeur, et toutes les autres offres sont traitées comme des accords non-finaux (nouvelles offres ou accord temporaires). Comme décrit par l'équation ci-dessus, si le vendeur verra sa réputation publique $R_{pub}(S)$ augmenter, il aura tendance à adopter une stratégie risquée en diminuant l'avantage des accords finaux par rapport aux accords non-finaux. Cet avantage est ajusté par les valeurs de réputation publique et l'utilisation de la variables τ propres à chaque vendeur. En effet, en diminuant la marge séparant les accords finaux sur les non-finaux revient à adopter une stratégie risquée puisque les valeurs des nouvelles offres pourraient facilement dépasser ceux des offres finales ce qui contraindra les vendeurs à se désengager des accords finaux et accepter les nouvelles offres dépourvues de toute sécurité en cas de désengagement, en espérant que ces dernières seront confirmées par les acheteurs correspondants, sinon le vendeur générera des pertes.

En revanche, si le vendeur voit son $R_{pub}(S)$ diminué, il adoptera une stratégie d'aversion au risque, en favorisant encore plus les accords finaux par rapport aux accords non-finaux afin de

diminuer son exposition d'être moralement pénalisé (ternissement de sa réputation) en dépassant le seuil de désengagement, ce qui permet d'augmenter progressivement sa réputation publique au sein de la communauté des acheteurs. En revanche, cette stratégie pourrait diminuer son chiffre d'affaires potentiel en laissant quelques offres intéressantes se faire contractées par ses concurrents commerciaux.

Un autre scénario pour lequel le vendeur pourrait générer une perte, c'est quand il se désengage d'un contrat final pour faire un nouvel accord avec un autre acheteur, qui devient par la suite une offre finale après que l'acheteur l'est confirmé, le vendeur par la suite se désengage de cette offre pour la remplacer par une autre nouvelle offre générant un revenue potentiel supérieure que la précédente, et si nous supposons que le vendeur va répéter le même processus, il pourrait se retrouver avec une série de pénalités à payer qui excèdent le profit du dernier contrat conclu. Pour cette raison, il est utile pour le vendeur de valoriser encore plus les offres finales ayant des pénalités élevées, à savoir, des pénalités dépassant même le prix d'offre de l'accord qui se produit lorsque le bénéfice net potentiel de l'offre est élevée.

La variable τ mentionnée dans l'équation 43 sera mise à jour selon l'évolution de la réputation publique du vendeur comme c'est le cas pour l'acheteur avec ses deux variables *iteration* et *sigma*, de la façon suivante :

$$\tau = \tau + \left(PubRepS_{diff} * (\tau/\rho) \right) \quad (43)$$

$$PubRepS_{diff} = preS_{PubRep} - curS_{PubRep} \quad (44)$$

Où τ est limité dans l'intervalle suivant :

$$0.5 \leq \tau \leq 1.5 \quad (45)$$

Dans laquelle $curS_{PubRep}$ signifie la réputation publique courante du vendeur et $preS_{PubRep}$ signifie la réputation publique du vendeur lors de la négociation précédente. La variable ρ est propre à chaque vendeur qui permet une mise à jour personnalisée de la variable τ et donc, il y aura des préférences appropriées pour les accords finaux pour chaque vendeur.

Voici les trois étapes principales de l'algorithme de classification du vendeur, déclenchées dès qu'une nouvelle offre supérieure au prix de réserve du vendeur se présente :

ALGORITHME DE CLASSIFICATION

- 1- Dé-allouer les ressources assignées aux différentes offres
 - 2- Retrier les offres par ordre décroissant selon l'équation (42)
 - 3- Ré-allouer les ressources aux offres par ordre décroissant de leurs valeurs
-

En ré-allouant les ressources selon le nouvel ensemble d'accords mise à jour, toutes les offres qui verront leurs statuts changés, seront notifiées comme suit :

En cas de réponse positive, deux types de notification peuvent être envoyés à l'acheteur [16] :

1. Envoi d'un message d'acceptation en cas d'une simple offre
2. Envoi d'un message « d'accord final en état d'exécution » signifiant que l'accord est définitif et qu'aucune partie n'est désormais autorisée à se désengager.

D'autre part, quand un message de refus est reçu, il y a trois types de notification qui peuvent être envoyée :

1. Si l'offre est simple, un message de rejet de l'offre est envoyé.
2. Si l'offre est un accord temporaire, un message d'annulation sera envoyé
3. Si l'offre est finale, un message de désengagement sera envoyé avec une pénalité.

Finalement, nous terminons notre section en présentant le pseudo-algorithme de la stratégie du vendeur dans l'Algorithme 2 suivant :

Algorithme 2 : Stratégie de négociation du vendeur

// La phase d'initialisation : doit être exécutée qu'une seule fois durant l'entrée du vendeur dans le marché

Fixer $\mathcal{A}_s, \mathcal{T}\mathcal{A}_s = \emptyset$;

$\mathcal{C}\mathcal{R}_s = \mathcal{R}_s$; où $\mathcal{C}\mathcal{R}_s$ signifie l'ensemble des ressources courant disponible pour la vente.

ListSorted : Une liste d'offre triée.

o.bidder : est un champ de l'offre o qui révèle l'identité du soumissionnaire.

//Fin de la phase d'initialisation

CanCntFrB (id) : une fonction qui prend comme paramètre l'id de l'acheteur et retourne le nombre d'annulation annulées par ce dernier à l'encontre du vendeur s .

SortedListO($o, \mathcal{T}\mathcal{A}_s, \mathcal{A}_s$) : une fonction qui retourne une liste triée d'offres des différents ensembles d'offres passées comme paramètres par ordre décroissant en utilisant l'équation 42.

si b envoi o tel que $o > \phi(o)$ **alors**

ListSorted = **SortedListO**($o, \mathcal{T}\mathcal{A}_s, \mathcal{A}_s$);
 $\mathcal{C}\mathcal{R}_s = \mathcal{R}_s$;

```

pour chaque  $o \in \text{ListSorted}$  faire
|   si ( $\mathcal{CR}_s$  satisfait  $o$ ) alors
|   |    $\mathcal{CR}_s = \mathcal{CR}_s - o$ ;
|   |   si  $o$  est une offre ou un accord temporaire alors l'acceptée ou l'a confirmée respectivement ;
|   |   fin
|   |   sinon
|   |   |   si  $o$  est un accord temporaire ou final alors l'annulée ou la désengagée respectivement ;
|   |   |   fin
|   |   fin
|   fin
fin
si  $b$  se désengage  $\mathcal{F}o$  alors
|   enlevé  $\mathcal{F}o$  de  $\mathcal{A}_s$ ;
fin
si  $b$  annule  $\mathcal{T}o$  alors
|   enlevez  $\mathcal{T}o$  de  $\mathcal{T}\mathcal{A}_s$ ;
|   mettre à jour  $\text{CanCntFrB}(b)$ ;
|   si  $\text{CanCntFrB}(b) > \text{theshold}$  alors
|   |   évaluez la réputation de  $b$  négativement;
|   fin
fin
si  $b$  confirme  $\mathcal{T}o$  alors
|   enlevez  $\mathcal{T}o$  de  $\mathcal{T}\mathcal{A}_s$  et placer le dans  $\mathcal{A}_s$ ;
fin
si n'importe quelle  $\mathcal{F}o$  est conclu et  $\text{CanCntFrB}(\mathcal{F}o.\text{bidder}) < \text{threshold}$  alors
|   évaluez la réputation de  $\mathcal{F}o.\text{bidder}$  positivement;
fin

```

6.4 Evaluation expérimentale

6.4.1 Paramètres expérimentaux

Nous avons exécuté une série d'expérimentations en utilisant différentes combinaisons de paramètres à partir du tableau 4 ci-dessous :

Tableau 4 Variables de la proposition 1

Variabes	Valeurs
Nombre de vendeurs	[5, 20]
Nombre de type de ressources par vendeur	[2, 8]
Quantité de ressources par vendeur	[2, 20]
Coût d'unité à la ressource	[10, 100]
Nombre de type de ressource par acheteur	[2, 6]
Quantité de ressource par acheteur	[2, 8]
Valeur/coût ratio	[12,50]

$tg(b)$	1
$pd(b)$	[10, 50]
$est(b)$	20
$dl(b)$	75
ratio demande/ressources $\psi(r)$	[0.2, 6]
Sigma dans [16]	4
Sigma	[4, 12]
Itération pour Trust Model	[1, 6]
Itération pour Non-Trust Model	1
Tau	[1.0, 1.8]
Rho	[2.0, 3.0]
Gamma	0.8
Epsilon	0.1
Lambda	0.9

Voici en bref les étapes suivantes que nous avons menées pour générer les données expérimentales :

Exigences de l'acheteur : l'acheteur requièrent 2 à 6 ressources pour une quantité spécifique allant de 2 jusqu'à 8 copies de chaque ressource demandée.

Les temps de l'acheteur : Avant d'entrer dans le marché, l'acheteur générera les attributs suivants représentant ces différentes périodes sur son axe de temps de négociation, lesquels sont : $tg(b)$, $est(b)$, $pd(b)$ et $dl(b)$. $tg(b)$ est fixé à 1. $pd(b)$ est généré uniformément à partir de l'intervalle [10, 50]. $est(b)$ est fixé à 20 pour tous les acheteurs. $dl(b)$ est fixé à 75 pour tous les acheteurs aussi.

Prix des acheteurs : Après avoir généré les exigences des acheteurs, chaque acheteur se verra assigner différentes probabilités de distribution suivant l'intervalle [10, 100], que nous utiliserons pour la génération du prix de réserve de l'acheteur pour chaque instance du problème. Ainsi, son prix d'offre au début de la négociation sera son prix de réserve et aussitôt que la négociation progresse, l'acheteur fera de plus en plus de concession jusqu'à atteindre son prix maximal en proposant "valeur/coût ratio" fois son prix de réserve comme son prix maximal à la fin de la négociation.

Approvisionnements du vendeur : Le vendeur de son côté, peut fournir de 2 à 8 types de ressources avec une quantité de chaque ressource qui peut aller de 2 à 20.

Ratio demande/ressource : Pour refléter chaque valeur du ratio demande/ressource qui provient du tableau 4, nous avons commencé par générer premièrement les approvisionnements du vendeur, et en se basant sur ça, nous avons ensuite généré les ressources de l'acheteur jusqu'à atteindre la valeur du ratio désiré.

Sigma : Les valeurs sigma variant entre [4, 12] pour l'approche proposée et de 4 pour l'approche [16].

6.4.2 Résultats expérimentaux

Nous avons conduit 1206 expériences sur les différents ratios de demande/ressource. Les expériences conduites à moyenne-échelle nous ont données les résultats suivants :

Observation :

Nous avons observé que notre modèle surpasse clairement le modèle de non-confiance décrit dans [16] sous les paramètres mentionnées dans le tableau 4. Pour des raisons de simplicité, nous avons appelé notre mécanisme « Trust model » indiqué avec la courbe bleu et l'autre dans [16] « Non-Trust model » indiqué par la courbe rouge.

1- La tendance d'annulation des acheteurs :

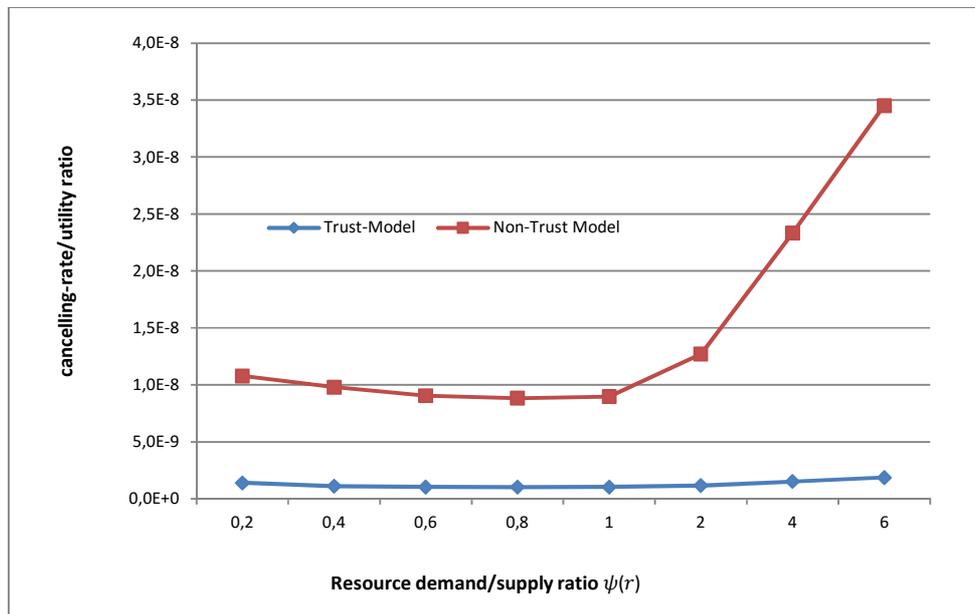


Figure 11 Tendence d'annulation de l'acheteur et compétition en ressources

Nous observons sur la Figure 11 que la tendance d'annulation de l'acheteur, ou en d'autres termes, le ratio d'annulation moyen de l'acheteur par rapport à son utilité est relativement bas dans le "trust model" que ce qui est dans le "Non-trust model", ce qui représente un de nos objectifs puisque cela permet de rendre le mécanisme potentiellement plus équitable parce que les taux des dommages causés par l'acheteur sont relativement bas par rapport à son utilité.

Le mécanisme proposé sur lequel l'acheteur ajuste sa fréquence de négociation selon le changement de sa réputation publique a permis de décroître le taux d'annulation de façon plus accrue que dans le cas où la fréquence de négociation est uniforme dans [16].

2- Tendence de désengagement du vendeur :

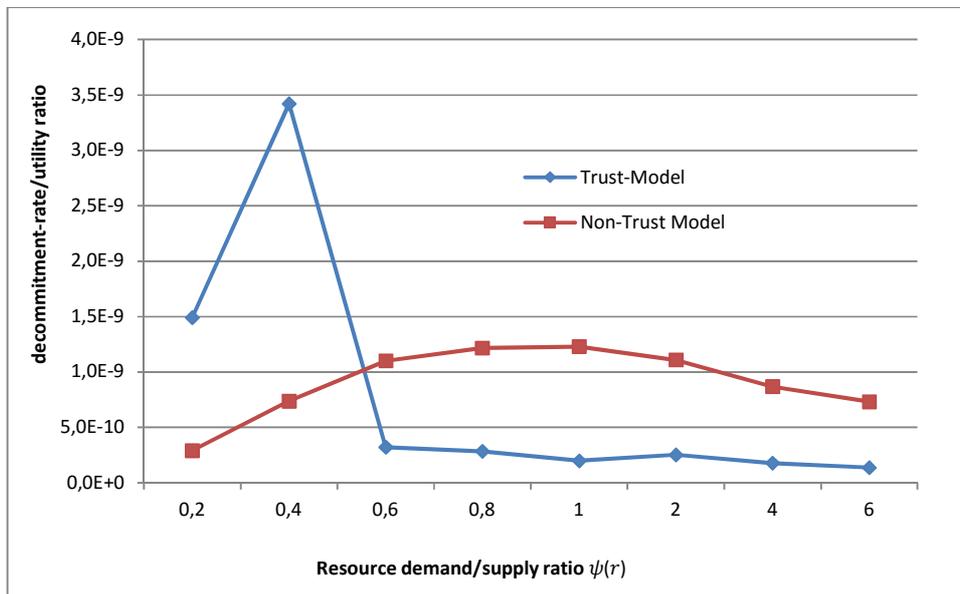


Figure 12 Tendence de désengagement du vendeur et compétition en ressources

La Figure 12 montre que la tendance de désengagement du vendeur dans le modèle proposé est globalement faible par rapport au modèle "Non-trust model" sauf lorsque la compétition en ressource est faible, i.e., 0,2 et 0,4. Comme nous l'avons illustré dans la section 1, le désengagement du vendeur peut causer des dommages non intégralement indemnisés par le mécanisme proposé dans [16]. Nous déduisons donc qu'un faible taux de désengagement du vendeur par rapport à son utilité rendra le mécanisme potentiellement plus équitable. Nous avons atteint cet objectif selon la figure ci-dessus en priorisant moins les offres venant de vendeurs ayant une faible confiance qui indique un taux élevé de désengagement commis de leurs parts et qui sont potentiellement non-intégralement indemnisés.

3-Taux de succès de l'acheteur :

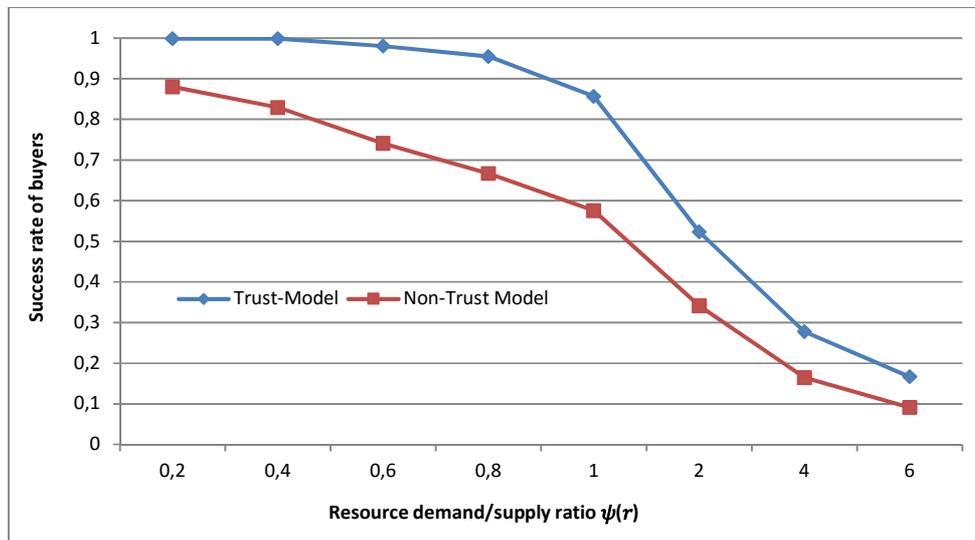


Figure 13 taux de succès et compétition en ressources

La Figure 13 montre que le mécanisme proposé surpasse le modèle "non-trust" [16] puisque le mécanisme proposé s'appuie sur la confiance de l'agent qui émerge au sein du réseau social au lieu de contracter des ressources en double pour faire face à l'incertitude. Deux inconvénients pourraient être cités en optant pour une stratégie d'acheteur à deux ensembles (non-trust model). Le premier est que l'acheteur pourrait obtenir des ressources en duplicata pour lesquelles d'autres acheteurs ne les ont même pas acquises en tant que ressources principales. Et le deuxième inconvénient est que chaque acheteur devra effectuer le double des opérations que s'il a opté pour une stratégie à un ensemble, ce qui s'avère une stratégie coûteuse dans un environnement où chaque transaction implique un coût.

En outre, le fait d'ajouter des critères de confiance peut aider parfois à accroître le taux de succès des acheteurs comme il peut l'aider à le décroître. Il pourrait aider à le faire augmenter dans les situations où l'acheteur est proche d'acquérir la totalité de ses ressources et le facteur de confiance agira comme un surplus d'avantage qui lui permettra d'acquérir les ressources nécessaires manquantes, pour qui sans ce surplus l'acheteur ne pourrait réussir à les acquérir dans sa négociation. D'autre part, il pourrait aider à diminuer le taux de réussite de l'acheteur, lorsque l'acheteur est plus proche d'acquérir la totalité de ses ressources, mais en raison de sa faible valeur

de confiance, cette dernière va agir comme un obstacle l'empêchant d'acquérir les ressources restantes nécessaires pour l'exécution de sa tâche dont il pouvait les acquérir si le facteur de confiance n'a pas été pris en compte.

Une autre observation de la Figure 13 est que le taux maximal de réussite atteint par le modèle "non-trust" est inférieur à 100%, même si la concurrence est très faible, ce qui pourrait s'expliquer par le fait que l'acheteur insistera toujours à essayer de faire un accord complet dans le cas où il y a certains vendeurs qui pourraient les satisfaire avant d'opter pour la deuxième alternative qui consiste à négocier sur un ensemble d'accords partiels. Enfin, il est parfois préférable de négocier sur peu de ressources avec un vendeur donné que d'une plus grande quantité puisque l'acheteur se verra accroître le nombre potentiel de concurrents commerciaux.

4-Nombre moyen de transactions :

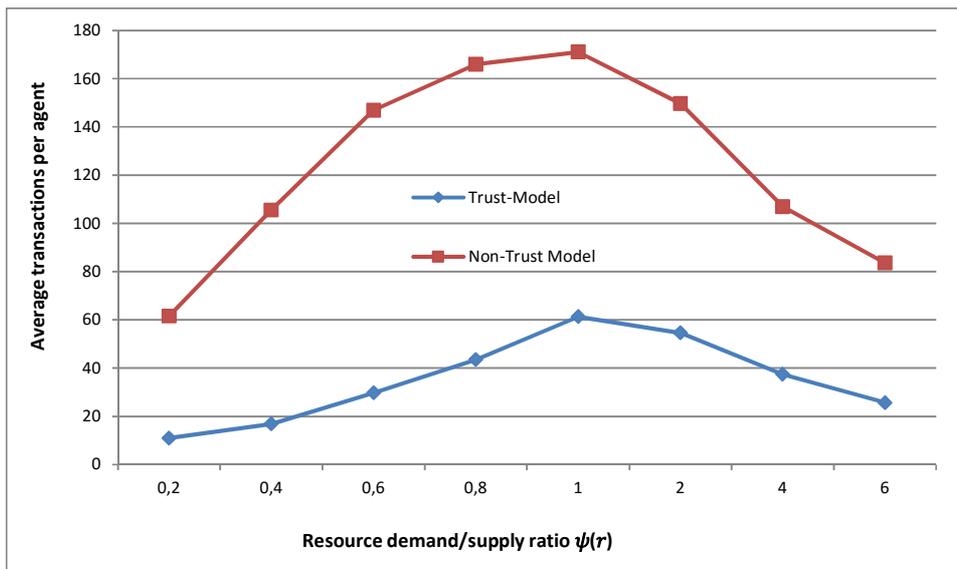


Figure 14 Transactions exécutées et compétition en ressources

Etant donné que la fréquence de transaction de l'acheteur sur le modèle proposé varie entre [1, 6] et est ajustée selon la réputation publique de l'agent au lieu d'être fixe pour l'ensemble des acheteurs à 1 dans le "Non-trust model", il est évident qu'il en résulte par moins de transactions en comparaison avec le modèle [16]. Trois types d'opérations sont prises en compte dans la Figure 14 (l'annulation, désengagement, et la conclusion de contrat). De plus, nous pouvons observer que les transactions effectuées en moyenne par l'agent décroîtront dans les deux mécanismes quand $\psi(r)$

passé à des valeurs 2, 4 et 6, et cela peut être expliqué par l'augmentation des offres rejetées puisque le prix de réserve du vendeur s'accroît avec l'augmentation de la compétition en ressource.

5-Moyenne du bien être sociale :

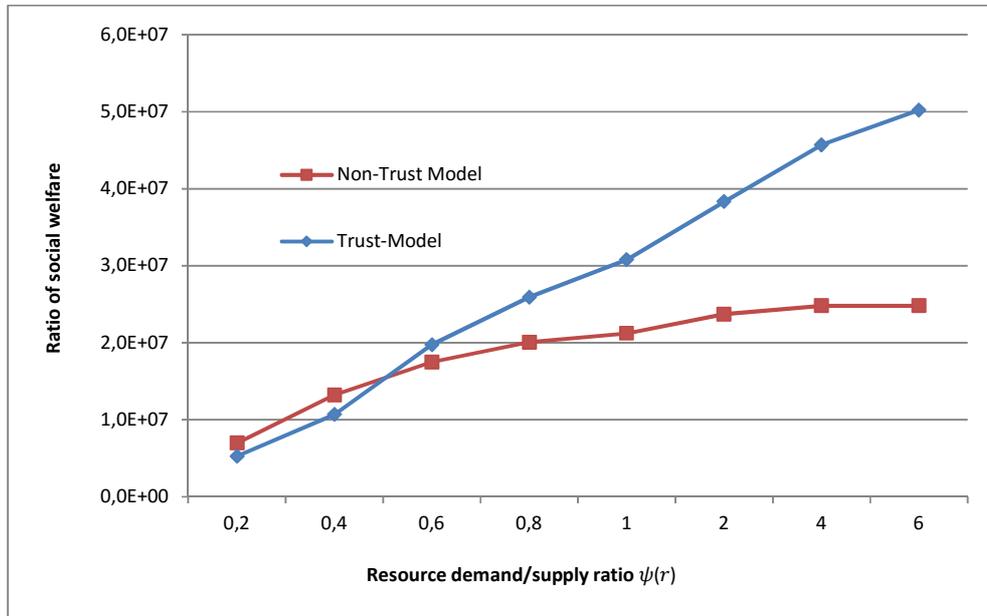


Figure 15 Bien être social et compétition en ressources

D'après les observations précédentes nous avons remarqué que le taux de réussite de l'approche proposée par l'acheteur est légèrement plus élevé tout en effectuant moins de transactions ce qui signifie moins de frais sont dépensés. Ainsi, il est évident qu'à partir de ces deux observations, le bien-être social généré par le modèle de confiance sera plus élevé que le modèle de non-confiance comme cela est montré dans la Figure 15. Toutefois, lorsque ces observations précédentes sont un peu similaires quand la compétition en ressource est faible, i.e., 0.2 et 0.4, le modèle « non-trust » fournit un meilleur résultat par rapport au modèle proposé en termes de bien-être social utilitariste parce que comme cela sera mentionné plus tard dans la Figure 17, la moyenne de l'utilité du vendeur dans le modèle « non-trust » est légèrement plus élevée.

6-Utilité Moyenne de l'acheteur :

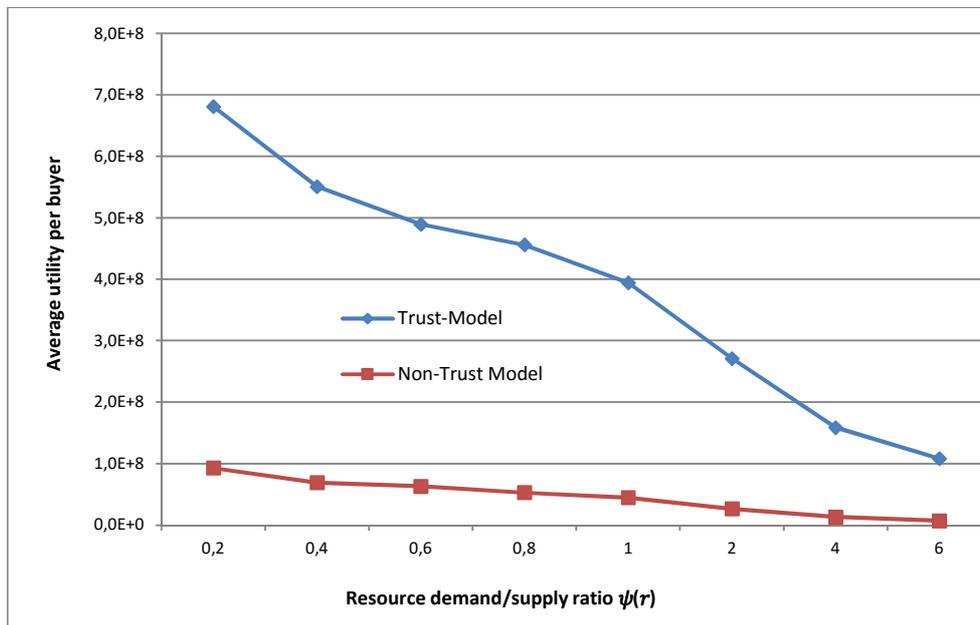


Figure 16 Utilité de l'acheteur et compétition en ressources

Nous pouvons observer à partir de la Figure 16, que l'utilité des acheteurs dans les deux mécanismes diminue par l'augmentation de la compétition en ressources puisque les ressources se feront plus rare, de plus les acheteurs feront de plus en plus de concessions pour satisfaire leurs exigences en ressources en augmentant le nombre de transactions pour les acquérir tout en augmentant leurs prix d'offres, ce qui entraîne une diminution dans le ratio des pénalité reçue/payé par l'acheteur et un accroissement de paiement de contrats non-nécessaire puisque ces derniers ne peuvent satisfaire seulement qu'une partie des exigences des acheteurs ce qui est insuffisant étant donné qu'ils sont unique d'esprit (« single-minded »).

Cependant, les raisons qui rendent notre mécanisme proposé meilleur en termes d'utilité de l'acheteur sont dues à un nombre de transactions effectuées moins élevés tout en maintenant un meilleur taux de succès de l'acheteur comparé au modèle [16].

En outre, nous pouvons observer un taux de décroissement plus net dans le modèle proposé comparé avec celui de non-confiance et cela peut être expliqué par le fait que les pénalités d'accords complet sont plus représentatives dans notre modèle que celui du modèle « non-trust ».

Par conséquent, une diminution du ratio des pénalités reçus/payés par l'acheteur due à l'augmentation de la compétition en ressources se reflétera par un décroissement plus net sur la courbe d'utilité de l'acheteur dans le modèle proposé par rapport au modèle de non-confiance.

7- Moyenne de l'utilité du vendeur :

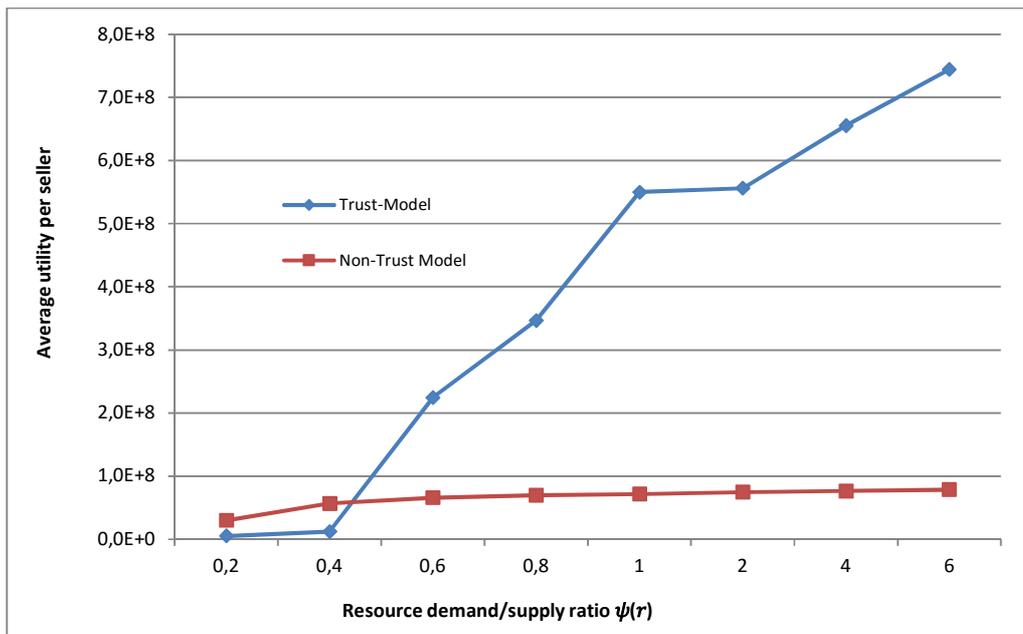


Figure 17 Utilité du vendeur et compétition en ressources

L'observation obtenue de la Figure 17 ci-dessus est que la méthode proposée débute avec une utilité du vendeur inférieure que le modèle « non-trust » lorsque la compétition en ressources est relativement faible, à savoir 0,2 et 0,4, mais dès que les ressources demandées dépassent la moitié des ressources fournies, l'utilité du vendeur du modèle de confiance s'accroît brusquement surpassant l'utilité du vendeur du modèle « non-trust ». Cela pourrait s'expliquer par le fait intuitif que la stratégie de l'acheteur de la méthode proposée est d'un seul ensemble, ce qui signifie qu'une augmentation dans le ratio des pénalités reçues/payés du côté du vendeur génèrera une augmentation plus accrue en terme d'utilité comparé au modèle « non-confiance » due aux mêmes raisons précédemment citées sur la courbe d'utilité de l'acheteur, à savoir, les pénalités d'accords complet sont plus représentatives dans le modèle de confiance que dans le modèle Non-Trust.

6.5 Conclusion

Dans cette proposition, nous avons identifié des défis relatifs au mécanisme de négociation pour le contrat de location des ressources qui est bien adapté dans l'environnement cloud avec son modèle de tarification « pay-as-you-go ». Nous avons proposé de répondre à ces questions en s'appuyant sur la confiance de l'agent qui émerge à partir de son réseau social. Cette confiance a eu pour rôle de rendre le mécanisme de négociation plus équitable en diminuant le taux des dommages non-intégralement indemnisés qui est causé à la fois par les acheteurs et les vendeurs. Ensuite, nous avons fait une étude comparative entre notre approche qui renforce sa stratégie de réservation en s'appuyant sur la confiance des agents au lieu de négocier sur des ressources duplicatas comme moyen de se protéger contre l'incertitude [16]. Nous avons montré à travers les résultats précédents que la méthode proposée pourrait nous donner des résultats prometteurs, et rendre le mécanisme de négociation plus équitable tout en générant un bien-être social plus élevé dans des environnements incluant des frais de transaction.

Chapitre 7 : Stratégie de négociation d'acheteur dynamique et hybride pour l'allocation de ressources

La négociation décentralisée plusieurs-à-plusieurs pour l'allocation des ressources du Cloud et des systèmes multi-agents présentent de nombreux défis, y compris ceux liés à la stratégie de l'acheteur qui est l'objet de la présente proposition. Les approches actuelles qui dérivent leurs demandes en ressources que chaque offre doit comporter ne sont pas dans tous les cas de marché un choix optimal. Pour cette raison, nous avons proposé une stratégie de négociation hybride consistant à une combinaison de deux modes de stratégie de négociation qui permettent de générer les ressources demandées de chaque offre qui sont soumises en parallèle aux vendeurs. La première est une stratégie de négociation fixe existante et la seconde est une stratégie d'apprentissage de sélection sur l'espace d'accord de l'acheteur. En outre, la stratégie de l'acheteur va avoir un comportement dynamique dans la place du marché en ajustant de façon appropriée les temps d'approvisionnement en ressources de l'acheteur et lance des appels à proposition d'abandonner les ressources contractées afin d'éclater les interblocages qui peuvent survenir entre les tâches des acheteurs. Les avantages apportés par ces caractéristiques ont été démontrées à travers les résultats de la simulation qu'elles permettent de réaliser de meilleures performances à la fois en termes de bien-être social et d'utilité d'acheteur.

7.1 Introduction

La négociation automatique est un domaine de recherche actif au cours des dernières années, où les deux aspects fondamentaux de la négociation automatisée sont les protocoles de négociation qui sont un ensemble de règles qui contrôlent l'interaction entre les agents et la stratégie de négociation qui est un modèle de prise de décision employée par chaque participant sur le marché pour atteindre ces objectifs en ligne avec le protocole de négociation.

Dans cette proposition, nous nous intéressons à une approche de négociation plusieurs-à-plusieurs décentralisée où aucune entité centrale ne raisonnera sur les décisions d'allocation de ressources tel que le commissaire-priseur et dans laquelle les participants du marché sont des agents d'intérêt personnels rationnels. Le processus de négociation sera composé de plusieurs tours de

négociations dans lequel un nombre limité d'offres sera autorisées à y être déposées pour chaque acheteur à chaque tour. Par conséquent, l'acceptation de l'offre de chaque acheteur par les vendeurs sera basée sur trois facteurs :

1. Le prix d'offre doit respecter la contrainte budgétaire de l'acheteur.
2. L'identité du vendeur.
3. Le montant des ressources exigées qui doit être disponible à la vente par ce vendeur.

Puisque la principale source d'utilité des acheteurs sera l'exécution de leurs tâches en réussissant à obtenir toutes les ressources nécessaires et en partie par les paiements reçus par d'autres agents. Ainsi, maximiser le bien être social doit passer par la maximisation du taux de réussite des acheteurs qui est le taux des acheteurs ayant réussi à exécuter leurs tâches. A cet effet, nous avons proposé une stratégie de négociation hybride de l'acheteur qui combine deux approches pour générer les ressources que l'acheteur doit demander dans chaque offre à chaque vendeur (i.e., le deuxième et le troisième facteur). La première [16] dérivera les ressources de l'offre par l'intersection des besoins de l'acheteur avec l'ensemble des ressources disponibles de chaque vendeur et la deuxième approche proposée les dérivera à partir de la région acceptable de l'espace d'accords de l'acheteur. En outre, étant donné que chaque acheteur maximisera de manière individuelle ses chances d'exécuter sa tâche, cela pourrait ne pas conduire à un taux de réussite globale maximale en raison de certains interblocages qui pourraient surgir entre les ressources des acheteurs. Pour cette raison, nous avons proposé la caractéristique "abandon" qui consiste à permettre à certains acheteurs de faire un appel à proposition pour se désengager des ressources contractées et tous les participants intéressés soumettront une proposition à cet effet. Ensuite, si la somme des propositions s'avère satisfaisante pour l'acheteur, il remettra ces ressources contractées au marché en se désengageant d'eux. Ces ressources sont nécessaires pour compléter les tâches de certains acheteurs qui peuvent créer une situation "gagnant-gagnant" aux acheteurs impliqués dans l'interblocage. En outre, l'acheteur pourrait maximiser son revenu en pénalité au détriment du vendeur en fixant stratégiquement les délais d'approvisionnement de ses ressources qui doivent être coordonnées de telle sorte que l'ensemble des ressources seront fournies en même temps.

Pour un jeu de négociation dynamique avec des informations incomplètes, le concept de solution approprié pour une telle classe de jeu est l'équilibre séquentiel [42], spécifiant une paire : un système de croyances qui prescrit la manière dont les croyances des agents doivent être mises à

jour au cours du jeu et les stratégies qui prescrivent comment les agents doivent agir. Mais comme les problèmes de négociation plusieurs-à-plusieurs sont souvent complexe du fait que les vendeurs et les acheteurs peuvent avoir des objectifs, des préférences et des politiques différentes, l'allocation optimale en termes d'efficacité de Pareto dans de telles situation a été prouvée qu'elle était intraitable à la fois en termes de quantité de calcul [33] et de communication [37] nécessaires. Une bonne alternative à ces problèmes serait alors de les adresser de façon décentralisée et qui est motivé par des applications réelles telles que la plate-forme GENI [3], qui est une organisation à but non lucratif reposant sur la technologie de virtualisation du cloud computing [22] pour allouer les ressources fournies provenant de plusieurs fournisseurs. Puisque le schéma de tarification du cloud computing est basé sur les services utilisés plutôt que sur toute une infrastructure, appelée par le modèle "Pay-as-you-go", les négociations opérant sous de tels environnements doivent être destinées aux contrats de ressources de location qui sont en adéquation avec les spécifications du cloud. De plus, la négociation décentralisée pourrait être utilisée dans des environnements tels que CLASP [21] pour des contrats d'achat ou de location où chaque système dispose de sa propre administration et objectifs, et que l'interopérabilité entre les sites pourrait leur permettre d'en tirer des avantages mutuels.

7.2 Description du problème

Notre problème traitera d'un ensemble d'agents répartis en deux types principaux : les acheteurs et les vendeurs.

Plus formellement, notre problème est présenté comme suit :

$$B = \{b_1, b_2, \dots, b_n\}$$

$$O = \{o_1, o_2, \dots, o_l\}$$

$$R_o = \{r_{o1}, r_{o2}, \dots, r_{oq}\}$$

$$S = \{s_1, s_2, \dots, s_m\}$$

$$R_s = \{r_1, r_2, \dots, r_t\}$$

Nous avons un ensemble d'acheteurs B et un ensemble de vendeurs S , dans lequel, chaque acheteur b générera une seul tâche τ qui représentera son point d'entré dans le processus de

négociation et qui définit les besoins de l'acheteur R_b qui seront satisfait par le moyen d'un ensemble d'offres O chacun d'eux contient R_o ressources fournit par un seul vendeur s_j qui peut vendre jusqu'à R_s ressources.

7.2.1 Etapes de l'offre

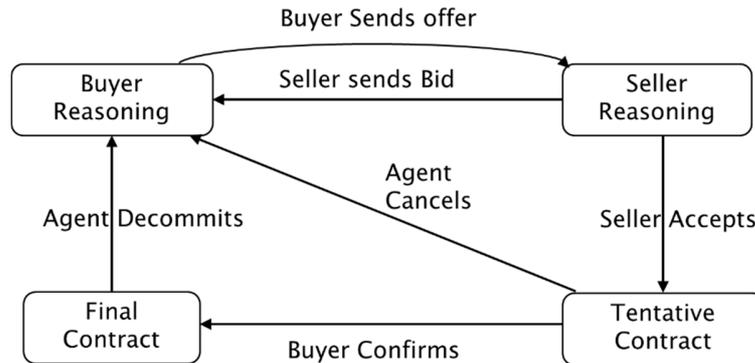


Figure 18 Machine à état finie des étapes de l'offre [16]

La Figure 18⁴ montre une description [16] de la négociation qui implique les deux parties principales qui sont l'acheteur b et le vendeur s :

L'état initial est « Buyer reasoning », dans lequel b décide de la façon de présenter son offre. Après, b envoie l'offre à s , l'état se transforme en « Seller reasoning ». Si s accepte l'offre, un *accord temporaire* sera alors établi. Dans le cas contraire, le vendeur envoie un « *quote* » (les prix acceptables) à l'acheteur. Si b confirme l'accord temporaire, la négociation est dans l'état « accord final ». Les autres alternatives qui restent à b sont soit qu'il annule l'accord temporaire sans rien payer ou se désengage de l'accord final en payant une pénalité prédéterminée, leurs négociations alors échouera et b redémarrera à l'état initial pour reformuler à nouveau son offre.

⁴ La figure 18 représente la figure 7 mentionnée auparavant et ceux pour faciliter la lecture.

7.2.2 Vue d'ensemble du problème de négociation

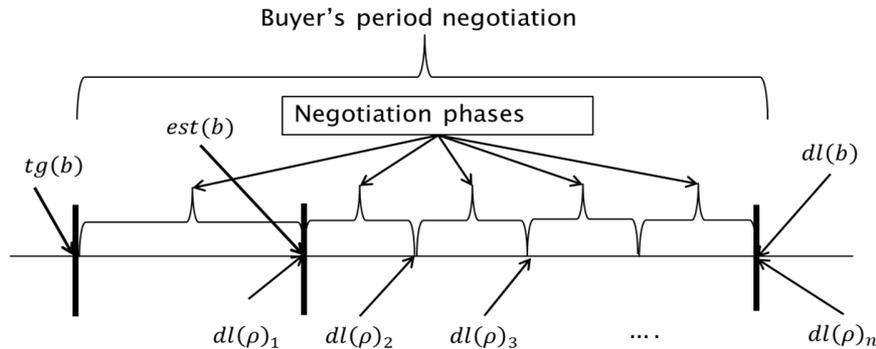


Figure 19 Disposition de l'axe de temps

Comme le montre la Figure 19, l'acheteur générera une tâche τ au temps $tg(b)$, puis il va définir dynamiquement de nouvelles phases de négociation avec des dates limites $dl(\rho)_i$ où $i = 1 \dots n$, qui représente le temps requis pour l'approvisionnement des ressources et des points de départ potentiel de τ . En effet, l'acheteur prendra en considération ces points comme des points de contrôles pour vérifier si l'ensemble des ressources requises ont été satisfaites aux moyens d'accords finaux et si c'est le cas, l'acheteur démarrera sa tâche, sinon, il établit une nouvelle date limite pour la phase de négociation $dl(\rho)_{i+1}$ après avoir désengager toutes ses ressources déjà contractées (i.e., ayant un temps d'approvisionnement $dl(\rho)_i$). Nous remarquons à partir de la Figure 19 que la première date limite de la phase de négociation est toujours égale à $est(b)$ qui signifie le temps de départ le plus tôt de τ . Enfin, l'acheteur mettra fin à sa négociation soit s'il a démarré sa tâche, $dl(\rho)_i \leq dl(b)$ où $dl(b)$ est la date limite de négociation de l'acheteur, soit il a accepté une offre intéressante pour remettre ses ressources contractées au marché afin qu'elles soient contractées par d'autres compétiteurs, ou il échoue à contracter ces ressources avant la date limite de négociation $dl(b)$. Dans le cas où l'acheteur échoue à exécuter sa tâche, il ne se désengagera pas systématiquement des accords finaux restants puisque ces derniers peuvent inclure une pénalité de désengagement supérieure au prix d'offre du contrat.

7.3 Stratégie de négociation de l'acheteur

La stratégie globale de l'acheteur est composée de cinq sous-stratégies :

1. Stratégie d'ensemble d'accords
2. Stratégie de confirmation

3. Stratégie de mode de négociation
4. Stratégie de planification
5. Stratégie d'abandon

La première concerne le nombre d'ensemble de ressources à contractées pour lesquelles chacun d'eux doit satisfaire l'ensemble des demandes de l'acheteur. La deuxième stratégie concerne les conditions sous lesquelles l'acheteur confirme les offres acceptées. La troisième est liée aux conditions qui conduiront l'acheteur à activer la stratégie d'apprentissage de sélection qui sera exécutée en parallèle avec la stratégie fixe, générant ainsi plus d'offres. Et finalement, la quatrième et la cinquième stratégie rendent le comportement de négociation de l'acheteur dynamique puisqu'ils sont responsables de régler stratégiquement la durée des phases de négociation et de décider sur les affaires d'abandons (retrait de la négociation) respectivement.

Dans ce qui suit nous présentons les notations utilisées et les algorithmes proposés :

- L'algorithme « **MAIN BUYER STRATEGY** » est l'algorithme principal de l'acheteur.
- L'algorithme « **UPDATE_NEGOTIATION_PHASE** » est l'algorithme qui met à jour la phase de négociation de l'acheteur d'après la phase précédente.
- L'algorithme « **DECIDING_CFP_ABANDON** » définit les conditions sous lesquelles l'acheteur lance des appels d'offres pour délaissé ses ressources contractées.
- L'algorithme « **FORMULATING_ABAND_PRICE** » définit la proposition de l'acheteur en réponse à un appel d'offre d'un autre acheteur qui veut remettre ces ressources contractées en négociation.
- L'algorithme « **EVALUATE_ABANDONING_PROPOSALS** » est employé lors de la prise de décision sur la question de remettre ou pas les ressources contractées de l'acheteur en négociation.

Tableau 5 Symboles Utilisés

\mathcal{MR}_b	Représente les demandes actuelles en ressources de b .
\mathcal{CR}_b	Représente les ressources actuelles contractées par b .
$perAcc$	Pourcentage d'accomplissement qui décrit le ratio des ressources demandées de b aux ressources contractées de b .
Bid_b	Un ensemble d'offres sélectionnées en utilisant le compromis d'exploration exploitation [30] à partir de la région d'accords de l'acheteur contenant les offres qui sont des sous-ensembles de \mathcal{MR}_b .
$v_b(t)$	L'utilité générée quand b exécute τ . Cette utilité se décroît linéairement avec le temps.
$c_b(r)$	Le prix de départ de l'acheteur à la ressource r .
$pd(b)$	Le temps d'usage de l'acheteur pour \mathcal{R}_b .
nb_{Bids}	Le nombre d'offres contractées dans la phase de négociation actuelle
lim_{Bids}	Le nombre maximum d'offres autorisées pour chaque tour de négociation
Sel_{Neg}	Prend la valeur vraie lorsque la stratégie de sélection d'apprentissage est activée.
CFP_b	L'identité de l'acheteur qui est derrière l'appel de proposition $CFP(\mathcal{CR}_b)$.
$CFP(\mathcal{CR}_b)$	L'appel de proposition de l'acheteur pour remettre son \mathcal{CR}_b .
ab_{offer}	L'offre de l'acheteur en réponse à $CFP(\mathcal{CR}_b)$
Pen_{Gains}	Le revenu actuel en pénalité de l'agent.
Pe_{Rec}	Le montant de pénalité reçue par l'agent
Pe_{Paid}	Le montant de pénalité payé par l'agent
$Ab_{Expense}$	Les dépenses de l'acheteur sur $CFP(\mathcal{CR}_b)$'s.
$Ab_{propositions}$	La somme de ab_{offer} de l'ensemble des compétiteurs en réponse au $CFP(\mathcal{CR}_b)$ de l'acheteur

$success_{Max}$	Les gains maximaux attendus g�n�r�s par l'ex�cution de la t�che.
$success_{Min}$	Les gains minimaux attendus g�n�r�s par l'ex�cution de la t�che.
inc_{Pres}	Une fonction croissante d�finit dans l'intervalle $[\lambda, 1]$ qui s'accro�t lin�airement de λ � 1 durant la p�riode entre $est(b)$ et $dl(b)$.
$Thresh_{Aband}$	Le prix de seuil de l'acheteur pour CFP_b afin de remettre son \mathcal{CR}_b .
$c(\mathcal{R}_b)$	Le c�t attendu de l'acheteur aux ressources \mathcal{R}_b .
$RP(dl(\rho_i), dl)$	La valeur attendue de la terminaison de la t�che de l'acheteur.

ALGORITHME 1 : MAIN BUYER STRATEGY

$nb_{Bids} = 0$; $\lim_{Bids} = 15$; $t' = t$ o  t repr sente le temps de simulation.

Pendant ($t < dl(b)$) et la t che n'a pas encore d marr e **faire**

$Sel_{Neg} = false$;

si ($t > ((dl(\rho)/2) + t')$) et ($|\mathcal{MR}_b| < 1/2 |R_b|$) **faire**

$Sel_{Neg} = true$;

fin

si ($t \geq dl(\rho) + t'$) **alors**

Se d sengager et annuler les accords finals et temporaires respectivement ;

$perAcc = 1 - (|\mathcal{MR}_b|/|R_b|)$; $t' = t$;

UPDATE_NEGOTIATION_PHASE ();

si ($perAcc < 1$) **alors**

DECIDING_CFP_ABANDON ();

fin

fin

si b re oit un $CFP(\mathcal{CR}_b)$ d'un comp titeur **alors**

FORMULATING_ABAND_PRICE ($CFP(\mathcal{CR}_b)$);

fin

pour chaque $s \in S$ tel que ($R_s \cap \mathcal{MR}_b \neq \emptyset$) et ($nb_{Bids} < \lim_{bids}$) **faire**

G n rer l'offre ($R_s \cap R_b$)   s ;

$nb_{Bids} ++$;

fin

si (Sel_{Neg}) **alors**

pour chaque $bid \in Bid_b$ and $nb_{Bids} < \lim_{bids}$ **faire**

G n rer l'offre ($bid \cap \mathcal{MR}_b$) pour le vendeur mentionn  dans le contrat ;

$nb_{Bids} ++$;

fin

fin

$t = t + dl(\rho)$; // Passer   la prochaine it ration

$nb_{Bids} = 0$;

fin

Annuler tous les accords temporaires.

Se désengager de chaque accord final inutile dont la pénalité de désengagement est inférieure à son prix d'offre.

ALGORITHME 2 : UPDATE_NEGOTIATION_PHASE ()

// La phase de négociation est délimitée entre $Neg_{MinPhase}$ et $Neg_{MaxPhase}$ et α est une constante (e.g., 0,8)
si ($|\mathcal{MR}_b| < 1/2 |\mathcal{R}_b|$) **alors**
 | $dl(\rho) = \min(Neg_{Phase} - \alpha, Neg_{MinPhase})$;
fin
sinon
 | $dl(\rho) = \max(Neg_{Phase} + \alpha, Neg_{MaxPhase})$;
fin

ALGORITHME 3 : DECIDING_CFP_ABANDON ()

si ($perAcc \times success_{Max} \times (1 - inc_{Pres}) < success_{Min}$) **alors**
 | envoyé **CFP**(\mathcal{CR}_b) à tous les autres acheteurs;
fin

ALGORITHME 4 : FORMULATING_ABAND_PRICE (CFP(\mathcal{CR}_b))

si ($perAcc = 1$) **ou** ($\mathcal{MR}_b - CFP(\mathcal{CR}_b) \neq 0$) **alors**
 | envoyé nulle à CFP_b
fin
sinon
 | envoyé ($inc_{Pres} \times (success_{Max} - success_{Min})$) à CFP_b
fin

ALGORITHME 5 : EVALUATE_ABANDONING_PROPOSALS ()

$avSuccess = ((success_{Max} + success_{Min})/2.0)$;
 $Thresh_{Aband} = perAcc \times avSuccess \times (1 - inc_{Pres})$;
si ($Thresh_{Aband} \leq (Ab_{propositions} + Pen_{Gains} - Ab_{Expense})$) **alors**
 | envoyé un message d'acceptation pour chaque acheteur ayant contribué à $Ab_{propositions}$;
fin
sinon
 | envoyé un message de refus pour chaque acheteur contribuant à $Ab_{propositions}$;
fin

Les gains minimaux et maximaux attendue de l'exécution de la tâche sont calculés comme suit :

$$success_{Max} = v_b(t) - pr(\mathcal{R}_b, dl(\rho_i), dl(b), t) + Pen_{Gains} - Ab_{Expense} \quad (46)$$

$$success_{Min} = v_b(dl(b)) - pr(\mathcal{R}_b, dl(b), dl(b), t) + Pen_{Gains} - Ab_{Expense} \quad (47)$$

Où $pr(\mathcal{R}_b, dl(\rho_i), dl(b), t)$ est le prix d'offre de \mathcal{R}_b défini comme [16] :

$$c(\mathcal{R}_b) + \left(RP(dl(\rho_i), dl(b)) - c(\mathcal{R}_b) \right) \left(\frac{t - tg(b)}{dl(b) - tg(b)} \right) \quad (48)$$

$$\text{Où : } c(\mathcal{R}_b) = \sum_{r \in \mathcal{R}_b} c_b(r) pd(b) \quad (49)$$

$$RP(dl(\rho_i), dl) = \begin{cases} \frac{\int_{dl(\rho_i)}^{dl(b)} v_b(t') dt'}{dl(b) - dl(\rho_i)} & \text{si } dl(b) \neq dl(\rho_i) \\ v_b(dl(\rho_i)) & \text{sinon} \end{cases} \quad (50)$$

Le revenu en pénalité du participant est calculé comme suit :

$$Pen_{Gains} = Pe_{Rec} - Pe_{Paid} \quad (51)$$

Et finalement, l'utilité de l'acheteur est calculée suivant l'expression suivante :

$$u_b(t) = \begin{cases} v_b(t) + Pen_{Gains} - Ab_{Expense} & \text{if } b's \text{ task is finished} \\ Pen_{Gains} - Ab_{Expense} & \text{otherwise} \end{cases} \quad (52)$$

7.3.1 La stratégie d'ensemble d'accords de l'acheteur

L'acheteur négociera sur deux ensembles d'accords, chacun doit satisfaire les exigences de \mathcal{R}_b . Nous avons opté pour une stratégie à deux ensembles puisqu'elle est considérée comme la meilleur [16] afin de faire face à l'incertitude de n'importe quel désengagement possible provenant du vendeur sans pour autant dépenser beaucoup dans Pe_{Paid} .

7.3.2 La stratégie de confirmation

Selon le pseudo algorithme "MAIN BUYER STRATEGY" ci-dessus, l'acheteur ne doit confirmer que les offres acceptées qui contiennent des ressources mutuellement disjointes sinon il devra les annulées. En outre, les offres générées à partir de la stratégie fixe ($\bigcup_{S=1}^n (R_S \cap \mathcal{M}\mathcal{R}_b)$) seront évaluées en premiers avant ceux générées par la stratégie de sélection d'apprentissage (Bid_b). En effet, changer l'ordre d'évaluation des offres peut changer le cours de la négociation comme il est démontré par l'exemple suivant :

Supposons les besoins de l'acheteur sont : $\{A, B, C, D, E, F\}$; et les offres acceptées (un tuple qui contient l'identité du vendeur et les ressources demandées) sont : $O1(S1; A, B)$; $O2(S2; A, C)$; $O3(S3; B, E)$ et $O4(S4; C, F)$. Supposons que l'acheteur a décidé de les évaluer dans l'ordre suivant : $O1, O2, O3$ et $O4$, il confirmera les offres suivantes : $O1$ et $O4$ et se désengagera des autres. Le $\mathcal{M}\mathcal{R}_b$ devient alors : $\{D, E\}$. Toutefois, si l'acheteur

choisit de les évaluer dans l'ordre suivant : $O2, O3, O1, O4$, il confirmera les offres : $O2$ et $O3$ et se désengagera des autres. Le \mathcal{MR}_b devient alors : $\{D, F\}$ ce qui conduit à un résultat différent que si l'acheteur a opté pour le premier ordre d'évaluation.

En outre, l'ordre d'évaluation spécifique des offres générées par la stratégie d'apprentissage sera déterminé en fonction des scores obtenus à partir d'un algorithme d'exploration-exploitation adaptée [30] qui sera décrit dans la dernière sous-section.

7.3.3 Stratégie du mode de négociation de l'acheteur

Cette stratégie locale définit les conditions sous lesquelles l'acheteur activera la stratégie d'apprentissage de sélection en parallèle avec la stratégie de négociation fixe pour accroître ces chances de succès à contracter les ressources dans laquelle la stratégie fixe a échoué en augmentant le nombre d'offres générées de $\cup_{s=1}^n (R_s \cap \mathcal{MR}_b)$ à $\cup_{s=1}^n (R_s \cap \mathcal{MR}_b) \cup Bid_b$ comme cela est indiqué dans le pseudo algorithme "MAIN BUYER STRATEGY" ci-dessus.

Nous avons illustré nos propositions par l'exemple suivant :

Tableau 6 Paramètre de prix de l'acheteur

Prix de l'acheteur Ressources de l'acheteur	Prix de départ	Prix de réserve
A	1000	3000
B	2000	4000

Tableau 7 Statut de négociation des vendeurs

Vendeur 1 : A, B, C
Offre 1 : A, C : 7 500
Offre 2 : B : 3 500
Vendeur 2 : A, B, C
Offre 1 : B, C : 7 300
Offre 2 : A : 2 200

En exécutant notre algorithme sur cet exemple, en relaxant la contrainte de la limitation d'offres et en assumant que le prix d'offre de l'acheteur est basé sur son prix de réserve et que la stratégie d'apprentissage est activée, les offres suivantes seront générées par la stratégie fixe : $\{(A, B), S1, 7000\}; \{(A, B), S2, 7000\}$ et les autres par la stratégie d'apprentissage : $\{(A, S1, 3000); (A, S2, 3000); (B, S1, 4000); (B, S2, 4000)\}$.

Dans le processus d'évaluation, l'acheteur évaluera en premier les offres générées par la stratégie fixe :

- $((A, B), S1, 7000); ((A, B), S2, 7000)$: Aucune confirmation d'offres sera faite puisque les deux offres entrent en compétition et ne peuvent proposer mieux que l'offre 1 des vendeurs 1 et 2 respectivement.

Ainsi, \mathcal{MR}_b reste inchangé et le mécanisme évaluera les quatre autres offres de Bid_b :

- $(A, S1, 3000)$ et $(B, S2, 4000)$: elles seront refusées pour la même raison que les autres offres précédentes.
- $(A, S2, 3000)$ et $(B, S1, 4000)$: Chacune de ces offres sera acceptées puisqu'elles entrent en compétition et proposent une meilleur offre que l'offre 2 des vendeurs 2 et 1 respectivement.

Ce cas montre que notre mécanisme a pu réussir dans ce que l'autre mécanisme [16] a échoué. Noter que si l'acheteur reçoit plus d'offres acceptées qu'il en a besoin, il peut les annuler gratuitement puisqu'ils ne représentent à ce stade que des accords temporaires. Cependant, générer plus d'offres peut avoir un effet secondaire même si annuler les offres se fait gratuitement. En effet, nous savons que l'acheteur se désengagera de tous ces contrats finaux à chaque point $dl(\rho)_i$ à moins que ces contrats finaux peuvent satisfaire l'ensemble des exigences de l'acheteur qui déclenchera l'exécution de la tâche de l'acheteur. A partir de ce fait, nous pouvons conclure que plus l'acheteur se rapproche du succès, plus de contrats finaux devront être désengagés, ce qui implique un paiement de pénalités plus conséquent pour pouvoir faire une nouvelle tentative de négociation, et puisque générer plus d'offres peut probablement conduire à confirmer plus d'offre, l'acheteur peut générer plus de pertes.

La dernière remarque qu'on peut donner sur l'exemple sus-cité, c'est qu'il montre une situation statique du marché dans le sens que le nombre de participants est fixe, ce qui n'est pas généralement le cas dans les ventes aux enchères menées dans un environnement cloud computing où les participants entrent et sortent du marché dynamiquement. En raison de la dynamique propre du cloud, une offre rejetée dans le tour actuel pourra être acceptée dans le prochain tour dans le cas où certains acheteurs ont libéré les ressources demandées après avoir terminés leurs tâches et le contraire reste vrai dans le cas où certains acheteurs arrivent avec des offres supérieures que l'offre de l'acheteur en question.

Afin de générer des offres dans la région acceptable de l'acheteur et ne pas dépasser lim_{Bids} , l'espace d'accords a été réduit d'un espace de complexité de $n2^m$ offres, où n est le nombre de

fournisseurs et m représente tous les types de ressources qui pourraient être fournis par eux, à la région acceptable de l'acheteur contenant des offres qui sont des sous-ensembles de \mathcal{MR}_b . Puis, en s'appuyant sur un algorithme adapté qui est un compromis de l'exploration-exploitation, Bid_b est alors formé par la sélection des offres ayant les scores les plus élevés, i.e., un maximum de cinq dans notre cas, ce qui est considéré comme la *phase d'exploitation*, et de certain autre offres aléatoirement choisies et non encore sélectionnées dans la phase d'exploitation, i.e., un maximum de cinq dans notre cas, ce qui sera considéré dans cette approche comme *la phase d'exploration*, faisant un totale d'ensemble de 10 offres contenues dans Bid_b . Ensuite, en termes d'ordre d'évaluation au sein de ces offres, les offres avec la note maximale de 1 seront toujours priorisées en les évaluant les premiers, puisqu'ils ont satisfait les besoins de l'acheteur pendant un certain temps avant que de nouvelles offres avec des revenus plus élevées les ont empêchées de réussir à les garder jusqu'à $dl(\rho)_i$. S'il n'y a pas d'offres d'une valeur 1, leurs positions dans Bid_b indiquant leurs emplacements d'évaluation seront mélangées pour explorer plusieurs combinaisons d'offres qui pourraient réussir à satisfaire les besoins de l'acheteur.

Le compromis exploration-exploitation [30] a été modifié dans les termes d'évaluations des offres candidates qui seront dans ce contexte leurs probabilités de réussite au lieu de leurs récompenses finales.

Nous notons par M_o le score passé moyen pour l'offre o et \tilde{R} le score courant du dernier tour de négociation. Les scores M_o de chaque offre sont calculés après avoir reçue un rejet, une annulation ou un désengagement de la part du vendeur comme suit :

$$M_o = M_o + (\tilde{R} - M_o) \quad (53)$$

$$\tilde{R} = \left[\left(\frac{sp_1^2 - or_1}{sp_1(c_1 + sp_1)} \right) + \left(\frac{sp_2^2 - or_2}{sp_2(c_2 + sp_2)} \right) + \dots + \left(\frac{sp_n^2 - or_n}{sp_n(c_n + sp_n)} \right) \right] * Pr(\sigma) \quad (54)$$

$$c_i = \frac{nb_{rbid}}{nb_{cbid}} \quad (55)$$

$$Pr(\sigma) = (Pr(\sigma_1) + Pr(\sigma_2) + \dots + Pr(\sigma_n)) / n \quad (56)$$

$$Pr(\sigma_i) = \begin{cases} 0 & \text{si } prMax_b(R) < pr(l) \\ \left(1 - \left(\frac{pr(l)}{prMax_b(R)} \right) \right) & \text{sinon} \end{cases} \quad (57)$$

Si une offre reçoit des “Quotes” à partir du vendeur, son \tilde{R} sera 0, cependant, si elle reçoit un message d'acceptation à partir du vendeur suivi de confirmation ou d'annulation de l'acheteur, elle obtiendra un \tilde{R} de 0.9 excepté pour les accords finaux qui ont succédés à satisfaire l'ensemble des besoins courants de l'acheteur dans un tour de négociation donné pour lesquels leurs \tilde{R} vaudra 1. c_i est le ratio des compétiteurs commerciaux direct/indirect pour la ressource r_i d'une offre donnée. Les compétiteurs commerciaux directs notés par $nb_{r_{bid}}$ sont ceux dans lesquels la ressource r_i est parmi leurs demande, et les compétiteurs indirects par nb_{cbid} dont la ressource r_i n'est pas parmi leurs demandes, mais ils ont en communs au moins une ressource en négociation avec l'offre o . Le sp_i est la quantité de ressource r_i fournie par le vendeur et or_i est la quantité de ressource r_i demandée par l'acheteur.

$Pr(\sigma_l)$ représente la probabilité d'atteignabilité du prix sur l'offre du compétiteur commercial l ayant une offre supérieure à l'offre o et $pr(l)$ représente son prix d'offre. $Pr(\sigma)$ est la probabilité moyenne d'atteignabilité du prix et $prMax_b(R)$ signifie le prix maximale que l'acheteur b est disposé à offrir pour l'ensemble des ressources R .

Pour récapituler, l'évaluation de la probabilité de réussite de chaque offre sera basée sur les ressources fournies par le vendeur, le niveau de compétition, et la quantité de ressources demandées par l'acheteur.

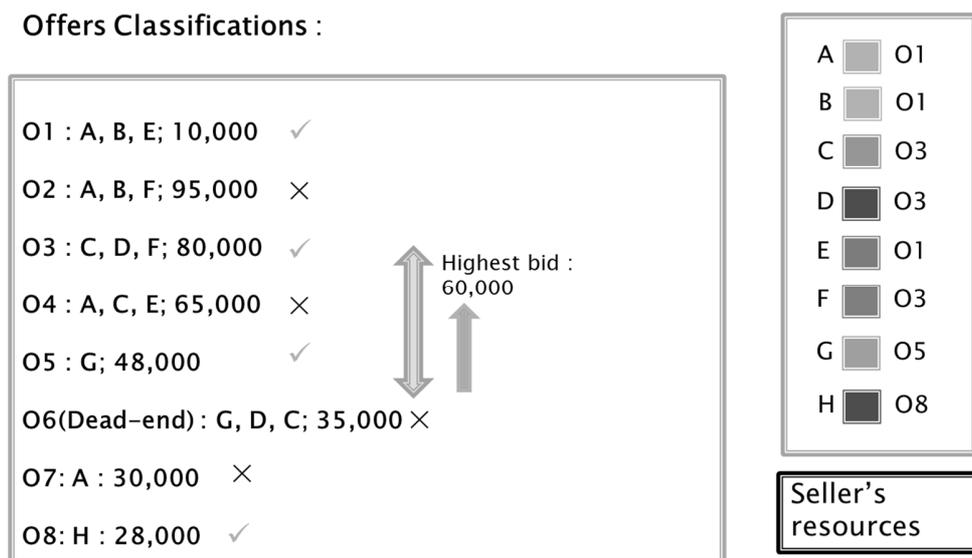


Figure 20 Atteignabilité du prix et les chances de succès de l'offre de l'acheteur

L'atteignabilité du prix des offres des compétiteurs impacte sur la probabilité de réussite d'une offre donnée. Par souci d'illustration, la Figure 20 ci-dessus montre que l'offre O6 ayant un prix de départ de 35 000 dans la situation actuelle n'est pas en mesure de réussir à obtenir un contrat auprès du vendeur. Mais puisqu'une négociation comporte plusieurs gagnants et les prix d'offres évoluent de plus en plus dans les cycles de négociations ultérieures, la probabilité que l'offre O6 pourrait être acceptée plus tard est possible si O3 et O4 se retirent de la négociation, et que O6 a la capacité de surenchérir plus que O5 sur les tours suivants de négociations ce qui implique une prédiction sur comment le prix maximal O5 (actuellement 48 000) dépassera le prix de réserve de O6 de 60 000.

7.3.4 Stratégie de planification de l'acheteur

Cette sous-stratégie a pour rôle d'ajuster dynamiquement la période de la phase de négociation au lieu qu'elle soit fixée préalablement comme dans [16].

Comme cela est mentionné dans le « MAIN BUYER STRATEGY » pseudo algorithme d'au-dessus, l'acheteur se désengagera de tous les contrats finaux à chaque point $est(b)$ à moins que ces contrats finaux puissent satisfaire les demandes de l'acheteur ce qui déclenchera l'exécution de la tâche de l'acheteur. Le vendeur de son côté se désengagera de n'importe quelle offre si une nouvelle offre arrive avec un prix d'offre supérieur à laquelle le vendeur ne peut satisfaire les deux demandes à la fois.

Partant de ces deux constats, nous pouvons conclure que l'acheteur peut tirer bénéfice d'un contrat en attendant suffisamment longtemps pour que son offre soit surenchérie par d'autres offres provenant des compétiteurs afin que le vendeur soit le premier à se désengager du contrat (stratégie d'aversion au risque). Cependant, laisser la possibilité à certaines offres de surenchérir fera certainement augmenter les revenus de l'acheteur en termes de pénalités reçues, mais en même temps il fera décroître ces probabilités de succès. Par contre, opter pour une stratégie risquée (phases de négociation courtes) aura pour effet d'accroître la fréquence de désengagement de l'acheteur d'un côté, mais l'acheteur va décroître le temps permis à ces compétiteurs de le surenchérir ce qui permettra d'accroître ces probabilités de succès à l'exécution de sa tâche.

Pour réaliser un bon compromis entre ces deux stratégies, la période de phase de négociation a été réduite si l'acheteur a contracté plus que la moitié des ressources demandées et sera étendue

dans le cas contraire comme cela est montré dans les pseudos algorithmes « MAIN BUYER STRATEGY » et « UPDATE_NEGOTIATION_PHASE » ci-dessus.

7.3.5 Stratégie d'abandon

A la fin de chaque phase de négociation, l'acheteur peut faire une $CFP(C\mathcal{R}_b)$ afin de remettre les ressources contractées en échange d'une récompense ($Ab_{propositions}$) qui est la somme des contributions de tous les autres participants intéressés et qui doit dépasser son prix de seuil. Cette opération permet parfois d'éclater certains interblocages qui empêchent certains acheteurs à exécuter leurs tâches, mais aussi pourrait même leurs permettre de la démarrer plutôt que prévue générant ainsi une meilleure utilité.

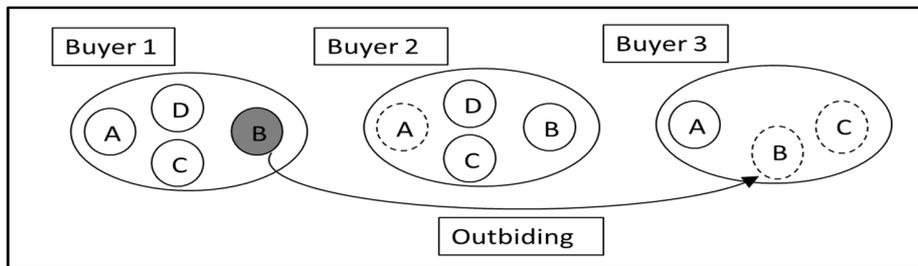


Figure 21 Issue d'une négociation avec inter-blocage entre l'acheteur 2 et 3.

La Figure 21 montre un résultat de négociation donné où les ressources A, B, C et D sont encerclées soit avec des lignes solides qui représentent leurs acquisitions ou en pointillés qui représentent leurs exigences. Trois observations peuvent être remarquées à partir de la Figure 21: D'abord, un interblocage de ressources a eu lieu entre les acheteurs uniques d'esprit 2 et 3. Deuxièmement, la ressource B représentée en gris signifie qu'elle a été acquise après avoir surenchéri d'autres compétiteurs l'ayant contractée auparavant. En effet, ce phénomène peut se produire quand un acheteur commence par surenchérir avec un prix de départ moins élevé que ses compétiteurs, mais aussi longtemps que le vendeur n'a pas encore conclu d'offres sur ces ressources, l'acheteur en question peut surenchérir dans les tours de négociations suivant sur ces compétiteurs si son prix de réserve est supérieur à ces derniers. La troisième observation est que seul l'acheteur 1 a réussi à exécuter sa tâche.

Nous pourrions améliorer l'issue de cette négociation si par exemple l'acheteur 3 aurait reçu un $Ab_{propositions}$ acceptable pour remettre ses ressources contractées au marché ce qui permet à l'acheteur 2 de contracter la ressource manquante A et démarrer sa tâche. En outre, l'acheteur 1

pourrait même démarrer sa tâche plus tôt si les ressources contractées ont été remises au marché avant qu'il surenchère l'acheteur 3 sur la ressource B .

Notez que les acheteurs malhonnêtes pourraient être tentés de tirer profit à partir de $CFP(\mathcal{CR}_b)$ en participant de façon continue sur le marché. Pour surmonter ce problème, le modérateur de la plate-forme de vente aux enchères pourrait réguler le nombre de $CFP(\mathcal{CR}_b)$ que chaque acheteur est autorisé à faire.

Enfin, avec cette fonctionnalité, il y a trois inconvénients qui pourraient affecter l'agent :

1. Dans les enchères de cloud computing, les acheteurs entrent et quittent le marché dynamiquement ce qui en résulte à un changement dynamique des ressources disponibles à la vente. Par conséquent, tous les interblocages pourraient avoir un effet temporaire sur les ressources des acheteurs. Toutefois, le fait d'éclater les interblocages à un stade précoce peut être bénéfique pour les acheteurs, même si cela pourrait conduire à un taux de réussite plus bas dans le cas où l'avantage supplémentaire apporté par l'exécution des tâches est supérieure aux pertes générées par le décroissement du taux de succès.
2. Les dépenses de l'acheteur seront constituées d'un élément supplémentaire $Ab_{Expense}$ en plus du paiement potentiel des ressources et des pénalités, qui pourrait provoquer des pertes supplémentaires à l'acheteur en cas d'échec d'exécution de la tâche, en d'autres termes, lorsque les ressources abandonnées seront acquises par d'autres concurrents que lui.
3. Le vendeur pourrait faire moins de profit si les acheteurs exécutaient leurs tâches plus tôt, puisque leurs prix d'offres grimpent tout au long de la négociation due au facteur de pression de temps puisque les acheteurs font de plus en plus de concession au fil du temps. Mais parfois, la fonctionnalité proposée pourrait augmenter l'utilité du vendeur car une plus grande quantité de ressources sera achetée en raison d'une augmentation du taux de tâches exécutées et pourrait même diminuer les pertes du vendeur en termes de Pe_{Paid} si ce dernier augmente de façon relativement plus élevée que Pe_{Rec} au fil de la négociation.

7.4 Stratégie du vendeur

Le processus de classification des offres est lancé chaque fois que le vendeur reçoit une nouvelle offre à partir d'un acheteur qui doit cependant excéder le seuil $\phi(o)$ [16] définit comme suit :

$$\phi(o) = \sum_{r \in R(o)} q(R(o), r) c_s(r) (1 + \psi(r)) p d(o) \quad (58)$$

Où $\psi(r)$ est le ratio de la demande/offre, $q(R(o), r)$ est la quantité demandée pour la ressource r et $c_s(r)$ est le prix de réserve du vendeur pour la ressource r .

Le vendeur classera les offres reçues selon leurs revenus par ordre décroissant selon la manière suivante :

$$rv(o) = \begin{cases} pr(o) - cost(o) + pe(o) & \text{si } o \in \mathcal{A}_s \\ pr(o) - cost(o) & \text{sinon} \end{cases} \quad (59)$$

L'accord final sera avantagé par rapport aux accords temporaires puisque le vendeur a une certaine garanti en cas où l'acheteur veut se désengager, une pénalité de désengagement devra ainsi être payée, ce qui n'est pas le cas avec l'accord temporaire et l'offre simple.

Ci-dessous les trois étapes principales de l'algorithme de classification lancé à chaque fois qu'une nouvelle offre excédant le prix de réserve du vendeur est reçue :

ALGORITHME DE CLASSIFICATION

- 1- Dé-allouer les ressources assignées aux différentes offres
 - 2- Retrier les offres par ordre décroissant selon l'équation (59)
 - 3- Ré-allouer les ressources aux offres par ordre décroissant de leurs valeurs
-

En réallouons les ressources selon le nouvel ensemble d'accords misent à jour, toutes les offres lesquelles verront leurs statuts changés, seront notifiées comme suit :

En cas de réponse positive, deux types de notification peuvent être envoyées à l'acheteur [16] :

1. Envoi d'un message d'acceptation en cas d'une simple offre

2. Envoi d'un message « d'accord final en état d'exécution » signifiant que l'accord est définitif et qu'aucune des parties n'est désormais autorisée à se désengager lorsque le point $dl(\rho)$ a été atteint.

D'autre part, quand un message de refus est reçu, il y a trois types de notification qui peuvent être envoyée :

1. Si l'offre est simple, un message de rejet de l'offre est envoyé.
2. Si l'offre est un accord temporaire, un message d'annulation sera envoyé.
3. Si l'offre est finale, un message de désengagement sera envoyé, lequel implique le paiement d'une pénalité qui est mentionnée dans le contrat.

7.5 Evaluation expérimentale

Dans cette section, nous avons comparé la stratégie de l'acheteur proposé, référencée par "Stratégie hybride" à son homologue [16] référencée par « stratégie fixe » et le schéma de prix fixe d'Amazon [4] en termes de bien-être social, taux de réussite, le temps de complétion de la tâche de l'acheteur et l'utilité moyenne de l'acheteur. Puisque *Amazon* applique *un schéma de prix fixe*, leur principal inconvénient réside dans le fait qu'ils ne sont pas économiquement efficace [108], ils ne garantissent pas que l'offre avec le revenu le plus élevée obtient les ressources. De plus, parce qu'il n'y a pas d'opération de désengagement dans l'enchère *Amazon*, l'acheteur terminera la négociation si l'une des trois conditions est vraie : l'exécution de la tâche, le budget a été consommé ou la période de négociation s'est écoulée.

7.5.1 Paramètres de Simulation

Nous avons réalisé une série de simulations en utilisant différentes combinaisons de paramètres du tableau 8 comme entrée pour les différents mécanismes de négociation. Chaque paramètre est tiré au hasard à partir de son intervalle respectif. L'étude comparative menée dans cette proposition est composée de deux environnements : le premier consistera en une négociation décentralisée plusieurs-à-plusieurs destinée à l'achat de contrats de ressources, où le nombre d'acheteurs et vendeurs est fixe, tandis que la seconde sera une négociation décentralisée plusieurs-à-plusieurs destinée à des contrats de locations de ressources qui sont appropriées aux spécifications du Cloud où les acheteurs entrent dynamiquement sur le marché avec un nombre de vendeurs fixe, ce qui correspondra aux environnements cloud du monde réel. Dans le deuxième

environnement la sous-stratégie d'abandon a été rejetée en raison de la dynamique inhérente du cloud.

Tableau 8 Variables de la deuxième proposition

Variabes	Values
Nombre de vendeur	[5, 8]
Nombre de type de ressources par vendeur	[2, 8]
Quantité de ressource par vendeur	[2, 6]
Coût d'unité à la ressource	[10, 100]
Prix initiale de soumission pour une unité de coût	[700, 800]
Nombre de type de ressource par acheteur	[2, 6]
Quantité de ressource par acheteur	1
Valeur/coût ratio	[1.2,5.0]
$tg(b)$	1, [1, 20]
$pd(b)$	-, [30, 70]
$est(b)$	$tg(b)$
$dl(b)$	$est(b) + 21$
$\psi(r)$	[0.2, 6]
$dl(\rho)$ dans [1], [4]	1.2
$dl(\rho)$ dans la stratégie proposée	[1.2, 3.6]
Tour de négociation	0.2
lim_{bids}	15
λ	0,4

Exigences de l'acheteur : Les exigences de chaque acheteur seront générées à partir du tableau 8, dans lequel un acheteur requiert 2 à 6 ressources.

Les temps de l'acheteur : Avant d'entrer dans le marché, l'acheteur générera les attributs suivants représentant ces différentes périodes sur son axe de temps de négociation, lesquels sont : $tg(b)$, $est(b)$, $pd(b)$ et $dl(b)$. Notez que $pd(b)$ a été rejeté dans le premier environnement alors qu'il est

compris dans l'intervalle $[30, 70]$ dans le deuxième environnement. $tg(b)$ est fixé à 1 dans le premier environnement tandis qu'il est compris dans l'intervalle $[1, 20]$ dans le deuxième environnement.

Prix des acheteurs : Chaque acheteur et vendeur évaluera initialement chaque unité entre $[700, 800]$, et à chaque fois que la négociation progresse dans le temps. Le *ratio valeur/coût* est utilisé pour générer la valeur maximale et minimale de chaque ressource de l'acheteur.

Approvisionnements du vendeur : Le vendeur de son côté, peut fournir de 2 à 8 types de ressources avec une quantité de chaque ressource qui va de 2 à 6.

Ratio demande/ressource : Pour refléter chaque valeur du ratio demande/ressource qui provient du tableau 8, nous avons commencé par générer premièrement les approvisionnements du vendeur, et en se basant sur ça, nous avons ensuite généré les ressources de l'acheteur jusqu'à atteindre la valeur du ratio désiré.

7.5.2 Résultats expérimentaux

En raison de ressources informatiques limitées dont nous disposons au sein d'un ordinateur de bureau équipé d'un processeur i5, nous avons effectué seulement 1002 simulations sur les différents ratios des demandes de ressources. Puisque le bien être sociale des différents mécanismes pourrait être considérablement disparates, nous rapportons le ratio du bien-être social de [16] et du mécanisme Amazon au bien être sociale du mécanisme proposé. Il sera suivi par deux autres mesures de performances, à savoir le taux de succès des acheteurs et le temps de démarrage de la tâche. Enfin, nous terminons par rapporter l'utilité moyenne de l'acheteur de chaque approche et le revenu de pénalité de l'acheteur de la stratégie fixe et hybride.

a Environnement 1 :

Dans le premier environnement, $tg(b)$ et $dl(b)$ signifions le début et la fin de négociation puisque le marché est supposé ouvrir périodiquement dès qu'une quantité de ressources suffisante est disponible à la vente et où le nombre de participants et des vendeurs est fixe.

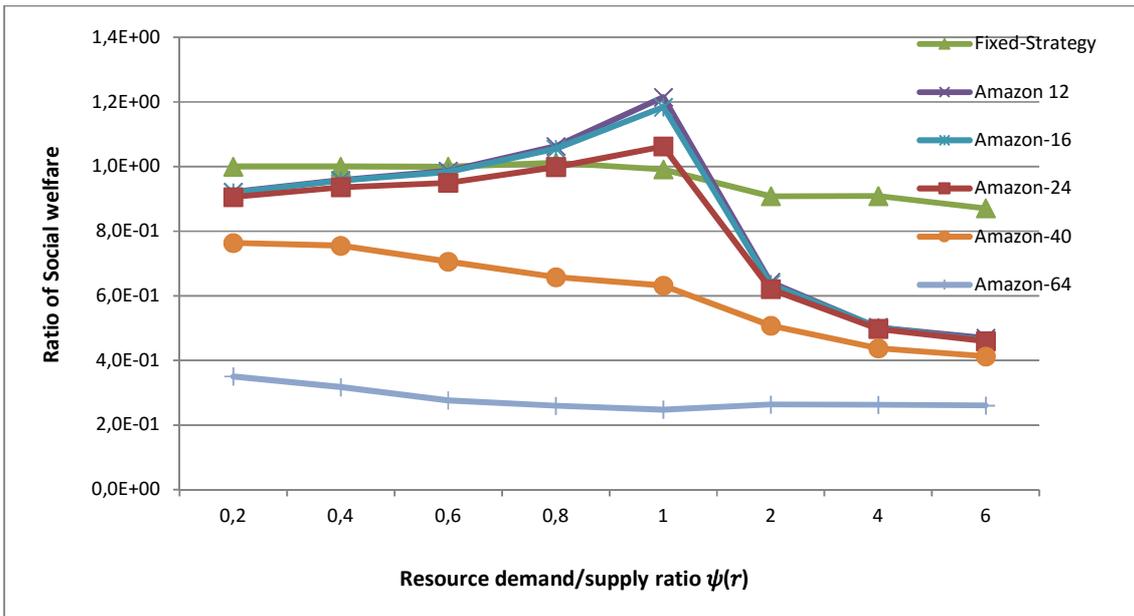


Figure 22 Ratio du bien être compétition en ressources

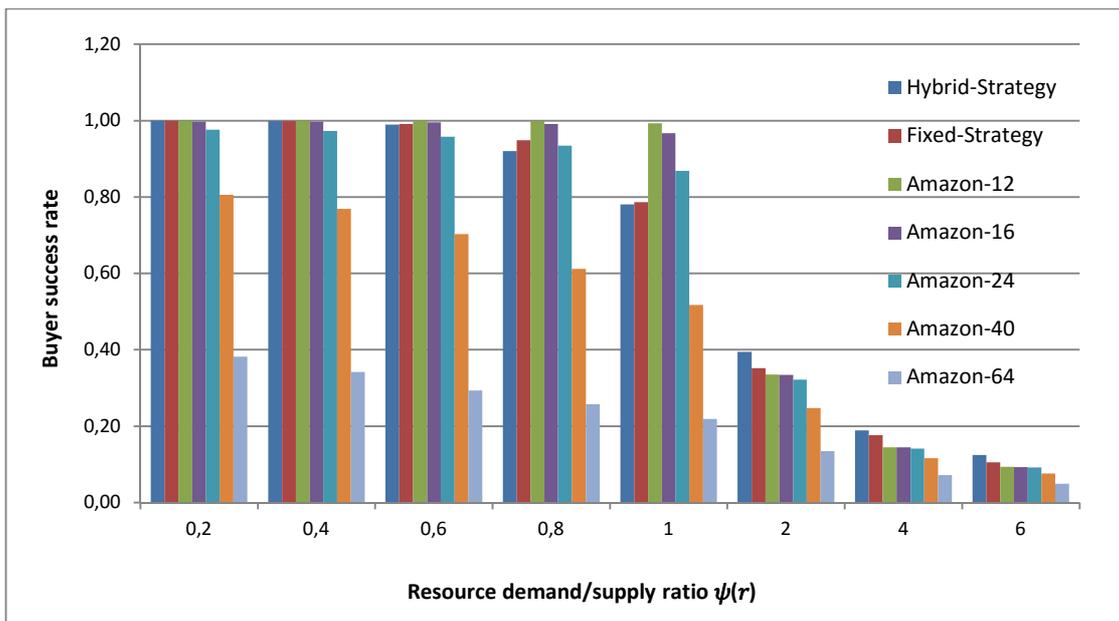


Figure 23 Taux de succès des acheteurs et compétition en ressources

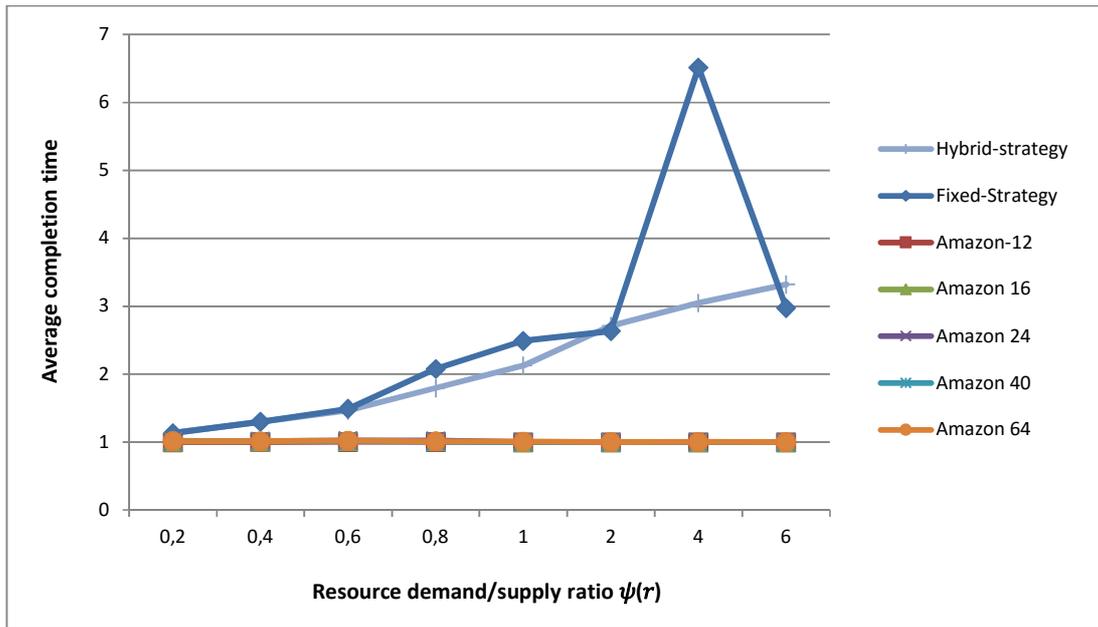


Figure 24 Temps de démarrage de la tâche d'exécution et compétition en ressources

A partir de la Figure 22, le bien-être social généré par l'approche proposée est relativement plus élevé que tous les autres mécanismes, sauf lorsque $\psi(r)$ a des valeurs de 0,8 et 1,0 dans laquelle le schéma d'Amazon à prix inférieur génère un bien-être social plus élevé en raison d'un taux de réussite plus élevé comme le montre la Figure 23. De plus, lorsque $\psi(r)$ a des valeurs élevées, l'avantage relatif pris par l'approche hybride s'accroît en raison de l'augmentation du taux de réussite qui se fait en éclatant les interblocages.

De plus, comme il n'existe pas de désengagement sous le schéma d'Amazon, il n'existera pas beaucoup d'interactions entre les agents avant que la négociation se termine. Pour cette raison, toutes les tâches s'exécutent aux alentours du point $est(b)$ comme cela est montré dans la Figure 24, ce qui leur permet de générer la valeur maximale en utilité que les tâches peuvent obtenir. Cependant, si les acheteurs ont été censés entrer sur le marché de manière continue comme c'est le cas dans un environnement Cloud, alors leur temps d'exécution sera systématiquement plus élevé que $est(b)$ fixé à la valeur 1 dans ce premier environnement.

Une autre observation de la Figure 22 montre que le schéma d'Amazon à bas prix est toujours plus élevé que ceux du schéma d'Amazon à prix élevé lorsque la compétition en ressources est large on contraste de ce qu'il a été observé dans [16]. Il est vrai que le bien-être social pourrait être augmenté en cas où le vendeur hausse son prix de vente s'il s'aperçoit que la compétition en

ressources est large. Mais pour cela, le taux de succès des acheteurs doit rester entre temps inchangé et cela ne peut se produire que lorsque le taux de succès des acheteurs à haut revenu s'accroît significativement. A partir de cette observation, les résultats finaux dépendront sur la disposition des budgets des acheteurs, ce qui signifie que si la proportion des acheteurs à haut revenu est plus élevée que celles à faible revenu, il est alors utile de fixer un prix de vente supérieur afin de générer un bien-être social supérieur lorsque la compétition en ressources est large.

Dans la Figure 23, on observe le taux de réussite de l'acheteur à travers les différentes valeurs $\psi(r)$. Trois observations ont été faites : La première est que le schéma d'Amazon avec des prix bas a toujours un taux de succès supérieur que ceux avec des prix élevés, ce qui peut s'expliquer par la proportion relativement élevée d'acheteurs qui sont apte financièrement à acquérir leurs demandes auprès des vendeurs ayant des schémas à bas-prix que ceux à prix élevés. La deuxième observation est que la stratégie hybride a un taux de succès supérieur à la stratégie fixe dans tous les niveaux de compétition puisque la stratégie proposée peut réussir dans des cas similaires à l'exemple montré par les tableaux 6 et 7 tandis que la stratégie fixe ne le peut pas. De plus, notre approche a la capacité d'augmenter encore plus le taux de succès lorsque la compétition en ressources est large en éclatant certaines interblocages. Finalement, la troisième observation est que le taux de succès de la stratégie hybride est moins bon que certains schémas d'Amazon avec des prix bas lorsque la ressource en compétition est basse (e.g., 0,6 ; 0,8 and 1,0) ce qui est dû au fait que l'agent n'acceptera pas ou n'offrira pas moins que ses gains attendus.

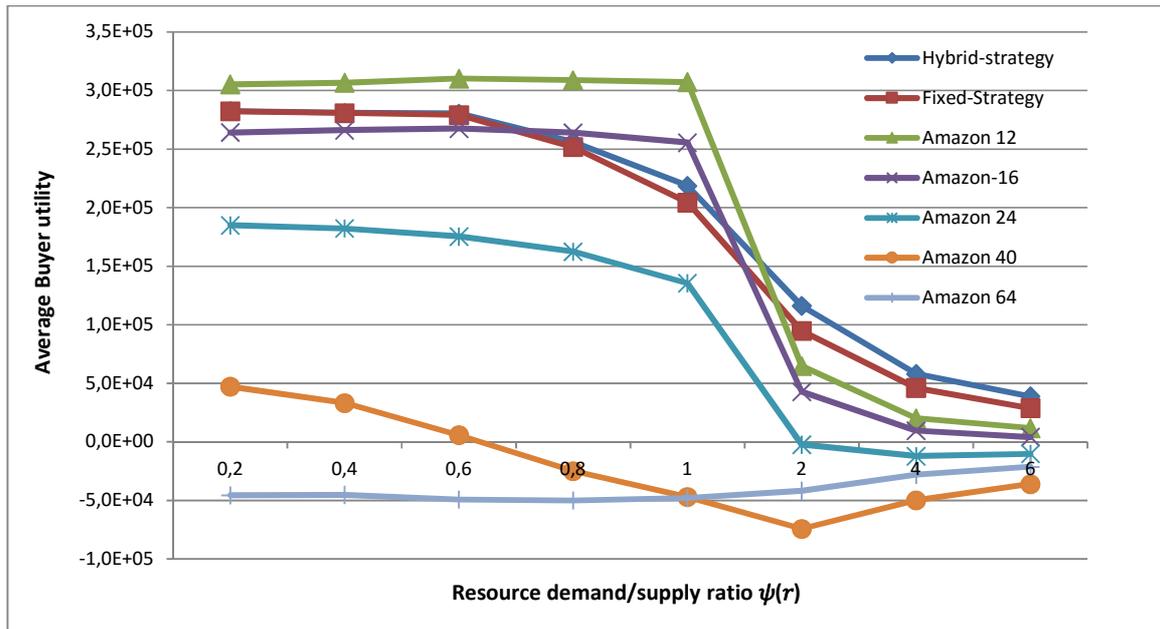


Figure 25 Utilité de l'acheteur et compétition en ressources

Enfin, la Figure 25 illustre l'utilité de l'acheteur dans les différentes approches à travers les différentes valeurs $\psi(r)$. Une observation importante tirée de cette figure pourrait être que l'utilité de l'acheteur de la stratégie hybride est globalement supérieure que celle obtenue par la stratégie fixe, cependant, cette dernière a générée un taux de succès légèrement supérieur lorsque $\psi(r)$ a été fixé à des valeurs de 0,8 et 1,0. Ceci pourrait être expliqué d'une part par le fait que le temps de départ d'exécution de la tâche des acheteurs dans la stratégie hybride s'est opérée plus tôt que la stratégie fixe comme cela est montrée par la Figure 24 et dans une perspective générale par le fait que la stratégie de l'acheteur proposé cherche à maximiser tout d'abord et avant tout l'utilité de l'acheteur en essayant de trouver le meilleur compromis entre l'utilité attendue de l'exécution de la tâche, la caractéristique d'abandon et le revenu de pénalité reçu par les vendeurs.

b Environnement 2 :

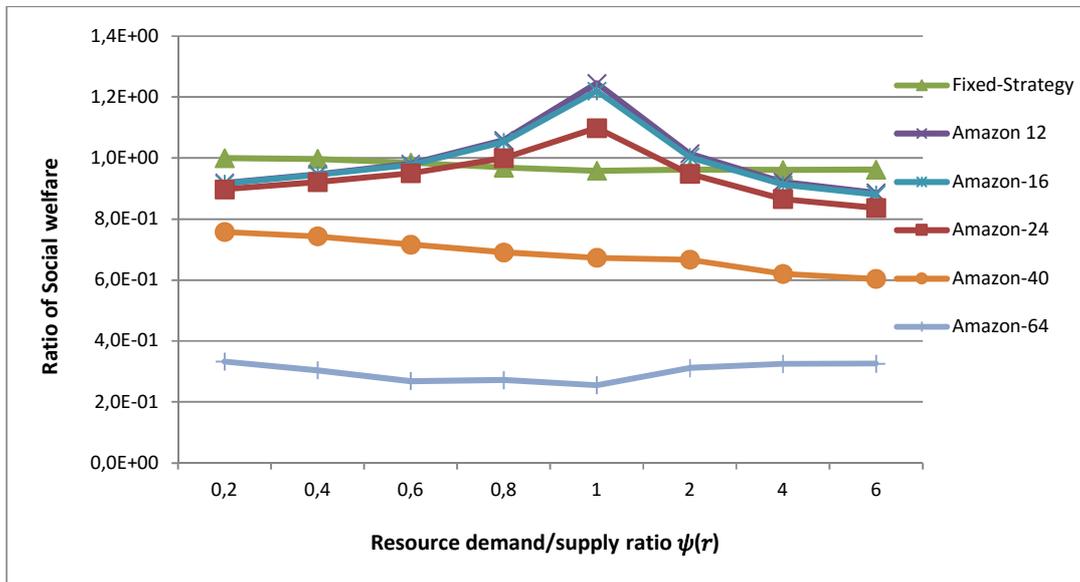


Figure 26 Ratio du bien-être social et compétition en ressources

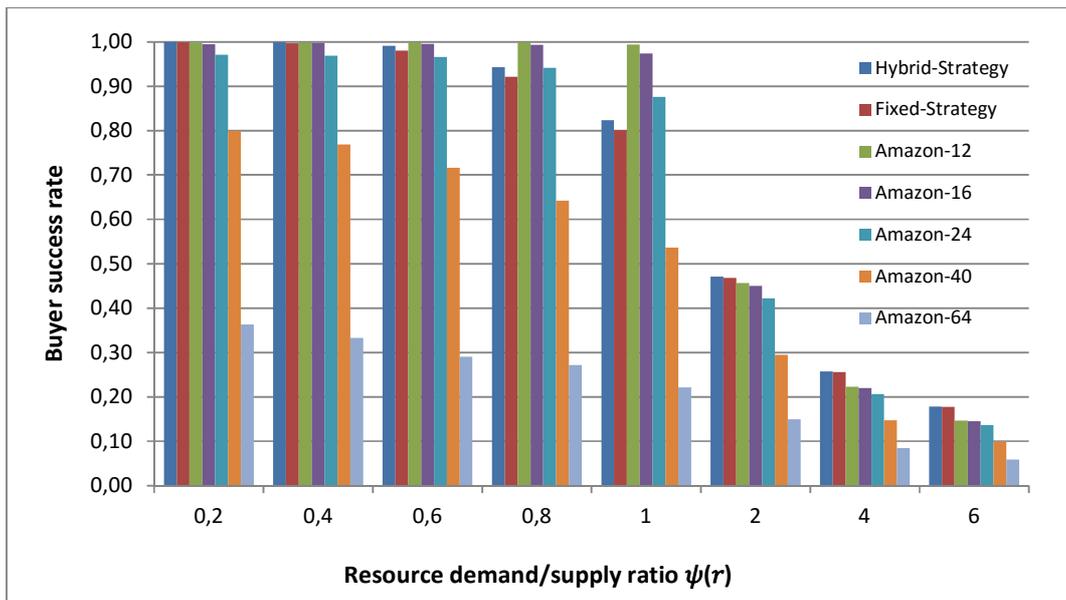


Figure 27 Taux de succès de l'acheteur et compétition en ressources

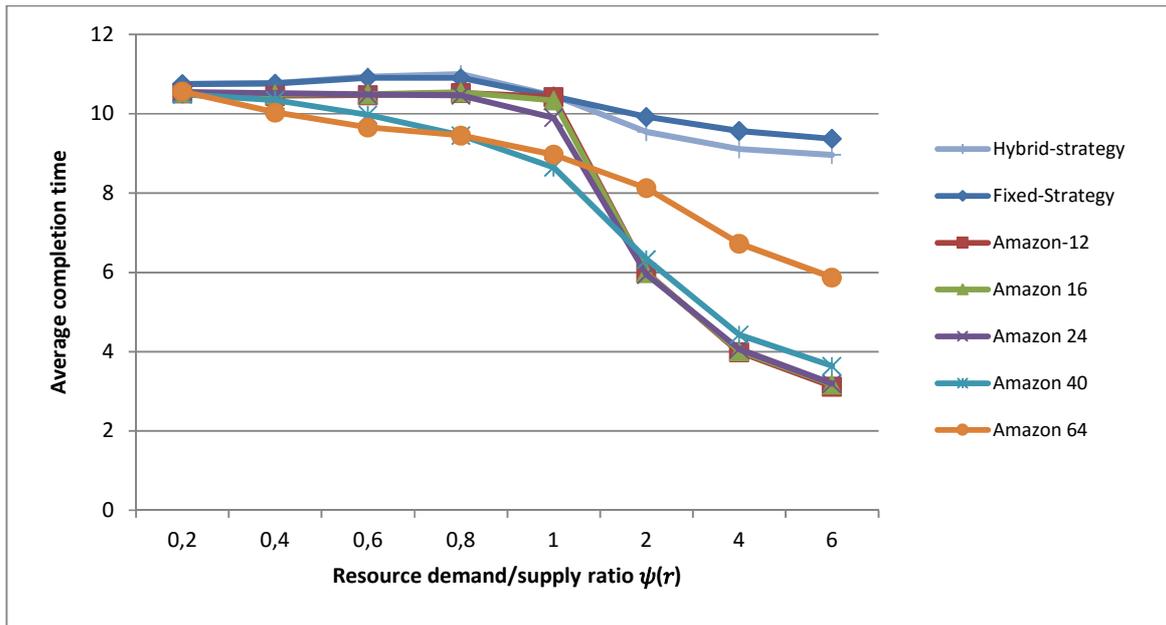


Figure 28 Temps de départ de l'exécution de la tâche de l'acheteur et compétition en ressources

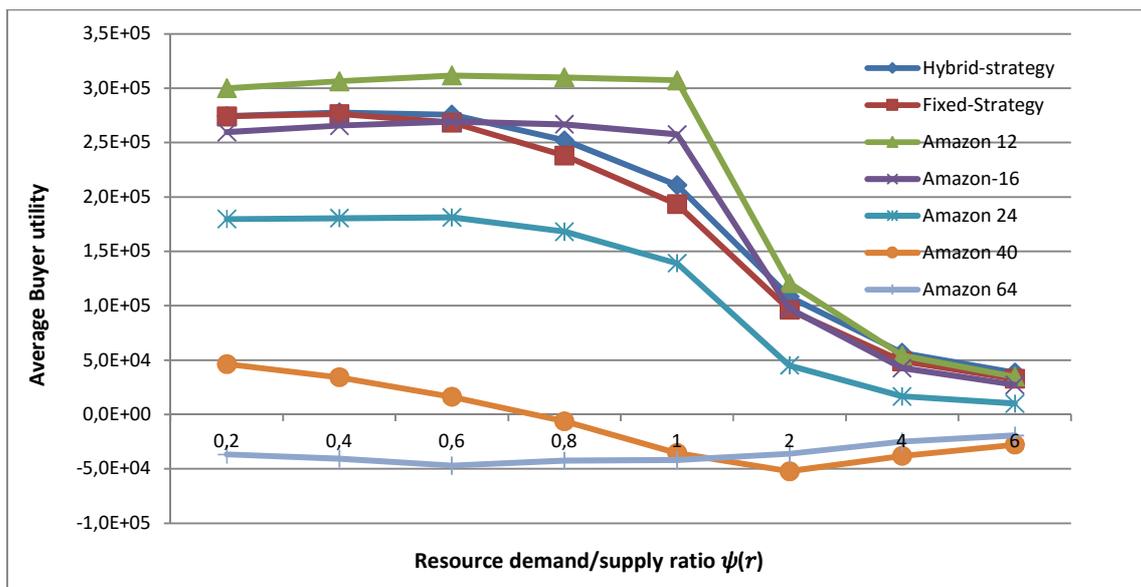


Figure 29 Utilité de l'acheteur et compétition en ressources

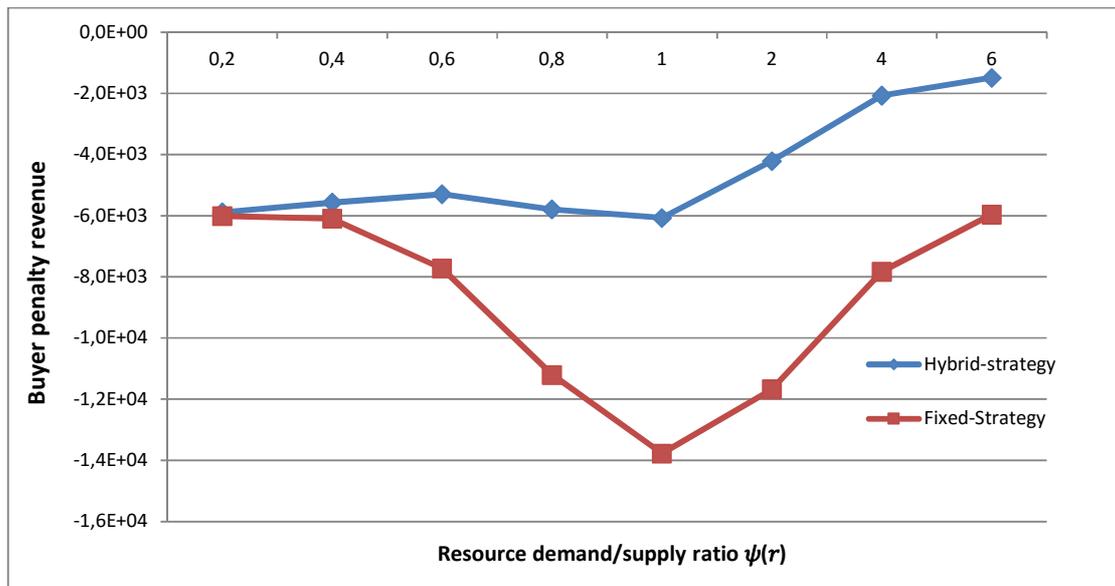


Figure 30 Revenu de pénalité de l'acheteur et compétition en ressources

Dans ce second environnement, d'après la Figure 29, l'utilité de l'acheteur dans la stratégie hybride est plus élevée que n'importe quel autre mécanisme, sauf lorsque $\psi(r)$ prend comme valeur 0,8 et 1,0 dans laquelle le schéma d'Amazon à prix inférieur génère un bien être sociale meilleur en raison d'un taux de succès plus élevé comme le démontre les figures Figure 26 et Figure 27. Les mêmes déductions peuvent être tirées pour le schéma de tarification d'Amazon par rapport au premier environnement. D'après la Figure 27, nous avons pu constater que le taux de succès de la stratégie hybride est plus élevé que la stratégie fixe, et à partir de la Figure 28 que le temps moyen pour démarrer la tâche de la stratégie hybride s'opère plus tôt que celui de la stratégie fixe ce qui s'explique intuitivement par le fait que la stratégie d'acheteur proposée envoie plusieurs propositions au vendeur que la stratégie fixe puisqu'elle génère ses offres à partir d'une plus grande portion de l'espace d'accord de l'acheteur, ce qui permet à l'acheteur de trouver la solution au problème de négociation plus tôt. Enfin nous avons pu constater d'après la Figure 30 que les revenus de pénalité moyenne de l'acheteur à partir de la stratégie hybride sont meilleurs, même si les deux approches génèrent des valeurs négatives, ce qui montre que la sous-stratégie de planification de l'acheteur s'avèrent utile, car elle diminue globalement les pertes de l'acheteur en termes de paiement de pénalité de désengagement.

7.6 Conclusion

Cette deuxième contribution propose une stratégie de négociation de l'acheteur appropriée pour les contrats de location de ressources qui prend un avantage par rapport aux approches existantes en explorant plus profondément l'espace d'accords de l'acheteur afin de déterminer la quantité de ressources demandées à chaque vendeur. Un autre aspect dans lequel l'acheteur pourrait générer des profits provient des revenus de pénalité pour lesquels un bon ordonnancement des délais d'approvisionnement des ressources pourrait générer une plus grande utilité à l'acheteur au détriment du vendeur. Comme les acheteurs sont unique d'esprit (« single-minded »), ces derniers peuvent tomber dans certains interblocages qui peuvent décroître leur bien-être social généré. Nous avons alors fait une extension du protocole existant [16] en introduisant la fonction « abandon » qui incite les acheteurs d'intérêt personnel à coopérer entre eux afin de parvenir à un accord sur un compromis mutuellement bénéfique entre les acheteurs impliqués dans ces interblocages. Toutefois, en raison de la nature même du cloud computing, un interblocage peut avoir un effet temporaire et des pertes potentielles peuvent alors survenir. Pour cette raison, nous avons seulement activé cette fonctionnalité dans les négociations relatives aux contrats d'achat de ressources dans lequel le marché s'ouvre périodiquement et qui a été démontré à travers les résultats expérimentaux que cela apporte de meilleurs résultats de négociation en termes de bien être sociale et d'utilité d'acheteur.

Chapitre 8 : Stratégie de vendeur améliorée dans une négociation décentralisée plusieurs-à-plusieurs

Dans le modèle de négociation automatisé plusieurs-à-plusieurs pour les contrats de location de ressources sous l'environnement du Cloud computing, la principale préoccupation était de trouver une bonne approche d'approximation au problème d'allocation de ressources, puisqu'une solution optimale au problème serait intraitable vue la complexité inhérente de la négociation de location de ressources sous l'environnement du Cloud. Une négociation décentralisée à engagement à niveau, ainsi que des stratégies d'agent non-gourmand aux ressources computationnelles ont été implémentés afin de générer des décisions de négociations qui ont un temps de réponse rapide tout en générant une bonne approximation au problème d'allocation de ressources plusieurs-à-plusieurs. Pour cette raison, nous avons conçu un algorithme "composite-greedy" pour la stratégie du vendeur afin d'adresser le problème "special winner determination problems" (SWDP) qui est le problème principal du vendeur dans un tel contexte et dont on va le décrire dans les sections suivantes. La stratégie du vendeur "composite-greedy" est une stratégie à deux-étapes : la première construit une classification selon le revenu de l'offre et la seconde selon sa densité. Le vendeur après avoir évalué ces deux classifications aura à choisir celle qui maximise le plus son utilité attendue. Les résultats expérimentaux montrent l'efficacité de notre approche par rapport à certaines autres approches relatives à la nôtre sur différentes mesures de performance.

8.1. Introduction

Le Cloud computing permet aux consommateurs de personnaliser les caractéristiques de la machine ainsi que le temps d'utilisation par rapport aux exigences de sa tâche. Le Cloud computing atteint cet objectif en s'appuyant sur la technologie de virtualisation qui permet de séparer un appareil physique informatique en un ou plusieurs appareils virtuels. Le déploiement du Cloud computing comme une solution d'entreprise avait été motivé par le fait qu'il n'est pas nécessaire d'avoir un capital initial, ce qui n'est pas le cas précédemment avec les infrastructures traditionnelles où les locaux, machines et leurs exigences de maintenabilité devait être satisfaites dès le départ avant de projeter le déploiement de n'importe quel plan.

Commercialisé les ressources sous un environnement du Cloud computing est caractérisé par trois difficultés : Tout d'abord, un grand nombre de participants du marché (acheteurs et vendeurs). Deuxièmement, la structure complexe de la transaction due à la complexité inhérente du contrat de location de ressources, et finalement, une enchère continue doit être implémentée dans un tel environnement où les ressources pourraient être demandées à tout moment et n'importe où afin de se conformer aux normes du Cloud computing. Dans de tels cas, concevoir des stratégies optimisées pour l'acheteur ou le vendeur va se révéler intraitable en termes de calcul puisque de nombreux facteurs (tels que la dynamique du marché, de multiples possibilités de marchés) sont interdépendants ainsi que des problèmes de confiance peuvent être soulevés par les participants du marché sur l'intégrité de l'agent central qui émet les décisions d'allocation de ressources, car les agents sont considérés comme étant d'intérêt personnel et rationnel. Pour cette raison, quelques bonnes approximations ont été proposées par Bo An, et al [16], dans lesquelles les auteurs avaient incorporé l'accord d'engagement à niveau dans le protocole de négociation, ce qui rendait possible de se défaire de l'agent de contrôle (commissaire-priseur), permettant aux agents de prendre des décisions de négociation rapides et locaux, ce qui s'est révélé une bonne approximation.

Actuellement, l'objectif de notre recherche est de traiter des questions que le vendeur doit être confronté dans ce contexte. À cet égard, le vendeur devra faire face à une série de soi-disant "special winner determination problem" (SWDP) que nous avons appelé ainsi en raison du fait que les décisions peuvent ne pas être définitives, ce qui engendre un troisième résultat qui vient s'ajouter aux deux présents dans les traditionnelles WDP, i.e. acceptation ou rejet de l'offre, qui est le fait de mettre en attente l'offre. Ce dernier résultat « en attente » avait été conçu pour résoudre SWDP pour la simple raison que chaque contrat n'est pas considéré comme définitif et les partis du contrat pourraient renier leurs engagements fait dans ce contrat, tant que son délai d'approvisionnement n'a pas encore été atteint. En fait, l'intervalle de temps qui sépare le temps de soumission de l'offre du temps d'approvisionnement, est le temps par lequel les deux parties du contrat peuvent se désengager s'ils le veulent. Donc, dès que le temps d'approvisionnement de l'accord a été atteint, le contrat sera scellé et devient comme les contrats traités dans le WDP standard, pour lesquels les deux parties du contrat (vendeur et acheteur) n'ont pas le droit de se désengager de leurs engagements et au lieu de cela ils doivent le respecter. En outre, la différence entre SWDP et le WDP standard vient du fait que dans le SWDP, les accords peuvent avoir des temps de soumission différents. Ceci s'explique par le fait que les offres ayant un temps de soumission antérieur ont

préalablement passé avec succès les précédents SWDP's leur permettant d'être en compétition encore avec les nouvelles offres dans le SWDP courant. Ces offres doivent maintenir ce même statut jusqu'à atteindre leur temps d'approvisionnement afin de faire passer leurs statut "en attente" en un statut de "succès".

L'événement qui déclenche la procédure de SWDP est la réception d'une nouvelle offre, puisque le vendeur ne peut satisfaire d'offre au-delà de ses capacités et devra donc vérifier si oui ou non il peut satisfaire la nouvelle offre qu'il a reçu en plus des offres qu'il a déjà contractées. C'est une enchère continue qui est tenue par chaque vendeur sur ces propres ressources et chaque offre se verra acceptée si elle passe avec succès plusieurs SWDP's tenu par le vendeur en question. Puisque le SWDP sera traité autant de fois que de nouvelles offres seront reçues par le vendeur, la nécessité d'implémenter une stratégie légère au vendeur est nécessaire. Celle-ci doit être efficace en termes de temps de réponse dans un environnement cloud computing. Dans cette proposition, nous avons conçu une stratégie « composite-greedy » qui évalue chaque SWDP avec deux classifications avant de générer n'importe quelles décisions de négociation qui concerne le vendeur. La première repose sur le prix de soumission de l'offre comme dans [16], et l'autre sur la densité de l'offre qui consiste à calculer le ratio prix-d'offre/ressources-demandées de l'accord. Ensuite, le vendeur va choisir la meilleure classification qui maximise le plus son utilité attendue. La classification retenue générera ses propres décisions de négociation qui doivent être alors suivies par le vendeur.

La négociation plusieurs-à-plusieurs est aussi motivée par des applications du monde réel, tels que CLASP [21], qui est un middleware assurant la communication dans les systèmes distribués à grande échelle, dans lequel chaque système autonome pourrait exiger l'exécution de certains plans dont leurs demandes dépassent parfois les ressources locales de ce dernier. Cela va créer la nécessité d'externaliser certaines tâches de calcul du système à d'autres systèmes qui disposent de certaines ressources inactives, créant ainsi une situation gagnant-gagnant entre les systèmes appartenant à la même organisation virtuelle.

Notez que les schémas commerciaux de prix fixe tel que Amazon EC2 spot instances [4] ont leurs propres inconvénients. D'abord, ils ne sont pas économiquement efficaces [108] dans le sens que l'acheteur avec l'offre la plus élevée n'est pas assuré d'obtenir les ressources demandées puisque ces approches sont basées sur le principe du premier arrivé, premier servi, et deuxièmement, les

schémas de prix fixe ne reflètent pas nécessairement les prix d'équilibre de l'offre et la demande du marché et ont comme seul avantage leurs performances en termes de temps de fonctionnement.

8.2. Contrat d'engagement à niveau

Dans cette section, nous nous intéresserons au format de soumission de l'offre ainsi que les phases possibles par lesquelles peut passer cette dernière sous le protocole de négociation d'engagement à niveau.

8.2.1 Format de soumission de l'offre

Chaque offre doit se conformer aux exigences du contrat de location qui inclut entre autre : la période de location qui sera appelée dans cette contribution le temps d'utilisation pour les ressources demandées pd , la pénalité de désengagement pe , le temps de démarrage le plus tôt pour l'exécution de la tâche de l'acheteur est auquel il n'acceptera aucun approvisionnement de ressources antérieures à ce temps-là et dl pour lequel il n'acceptera aucun approvisionnement de ressource ultérieure à ce temps-là.

Plus formellement, le format de l'offre sera présenté au vendeur comme suit : $\langle pr, pe, \mathcal{R}, est, pd, dl \rangle$ où pr signifie le prix d'offre du contrat et \mathcal{R} les ressources demandées du contrat ainsi que leurs quantités correspondantes.

8.2.2 Les phase d'accord dans la négociation d'engagement à niveau

Avant de décrire la stratégie du vendeur qui traite des contrats d'engagement à niveau, nous allons donner un bref aperçu de la façon dont ce concept est intégré dans le protocole de négociation. Dans l'ensemble, les contrats d'engagement à niveau, contrairement aux contrats d'engagement entiers auxquels aucune des deux parties n'est autorisée à se désengager, permet au contracteur de se libérer du contrat [96]. Le désengagement du contrat va s'opérer soit de manière gratuite ou de manière payante en payant une pénalité et cela va dépendre de la phase de l'offre dans laquelle l'ensemble des phases seront représentées dans la **Figure 31** Machine à état finie des phases du contrat [16]Figure 31 ci-dessous.

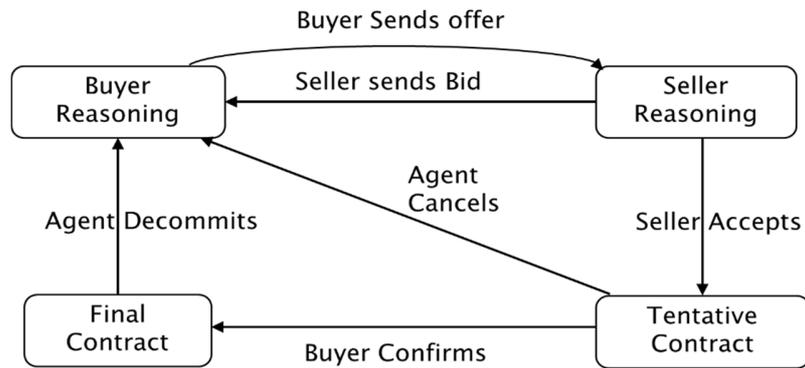


Figure 31 Machine à état finie des phases du contrat [16]

La Figure 31⁵ montre que l'offre peut être dans l'une des trois phases, où chaque phase de l'offre symbolise l'étendue de l'engagement des deux parties concernées par l'accord. Premièrement, "buyer reasoning" est l'état initial où l'offre est sur le point d'être formulée par l'acheteur et envoyé au vendeur. Ensuite, elle sera traitée par la composante du vendeur "seller reasoning" qui accepte des offres dépassant son prix de réserve, les transformant en des accords temporaires, sinon le vendeur va les rejeter en envoyant les « quotes » acceptable par ce dernier. Dans l'état " accord temporaire " de la figure 31, l'acheteur a la possibilité de confirmer l'accord en le transformant en un accord final. L'autre possibilité est que les deux parties de l'accord peuvent annuler gratuitement cet accord temporaire. Dans le cas où l'accord est devenu final, le désengagement de l'accord sera payant, sinon si l'accord est dans une autre phase que celle-ci, alors le désengagement est sans frais.

Notez que le désengagement n'est valide que lorsque l'une des parties de l'accord voulant se désengagé se désengage avant le temps d'approvisionnement mentionné dans l'accord. Après ce temps-là, l'accord devient scellé et appelé dans cette contribution « accord final définitif ».

8.3. La stratégie du vendeur

Avant de détailler la stratégie du vendeur, voici une liste dans le tableau 9 qui résume les symboles utilisés dans les formules et les pseudo-algorithmes qui vont suivre :

⁵ La figure 31 représente la figure 7 mentionnée auparavant et ceux pour faciliter la lecture.

Tableau 9 Symboles Utilisés

t	Le temps courant
\mathcal{DA}_s	L'ensemble des agréments finaux définitifs du vendeur s .
\mathcal{A}_s	L'ensemble des agréments finaux du vendeur s .
\mathcal{TA}_s	L'ensemble des agréments temporaires du vendeur s .
σ	L'offre nouvellement reçu qui déclenche le WDPS.
\mathcal{R}_s	L'ensemble des ressources du vendeur s disponibles pour la vente.
$pd(o)$	Le temps demandé pour l'utilisation des ressources demandées par le contrat o .
$q(r)$	La quantité demandée pour la ressource r .
$c_s(r)$	Le coût de revient du vendeur pour la ressource r
$\phi(o)$	Le prix de réserve du vendeur pour les ressources demandées dans l'offre o .
$\psi(r)$	Le ratio offre/demande de la ressource r
$\mathcal{R}(o)$	L'ensemble des ressources demandées par l'offre o .
$pr(o)$	Le prix d'offre de l'offre o .
$cost(o)$	Le coût de l'offre o pour le vendeur.
$pe(o)$	La pénalité de désengagement de l'offre o .
$v(o)$	La valeur de l'offre o par rapport à la classification spécifique appliqué par le vendeur.
$stage(o)$	La phase courante de l'offre o .
$rw dps(o)$	Le résultat de l'offre o par rapport au dernier WDPS.
$bidder(o)$	L'identité du soumissionnaire de l'offre o .

Dans cette contribution, nous allons nous passer de la stratégie de l'acheteur puisque nous nous focaliserons dans cette contribution seulement sur la conception de la stratégie du vendeur qui doit effectuer les meilleures performances dans la partie évaluation.

Essentiellement, le processus de classification des offres pour résoudre le problème du WDPS sera déclenché chaque fois que le vendeur reçoit une nouvelle offre qui dépasse le prix de réserve du vendeur, dénoté par $\phi(o)$ qui est défini comme suit :

$$\phi(o) = \sum_{r \in R(o)} q(r) c_s(r) (1 + \psi(r)) pd(o) \quad (60)$$

Chaque fois que le vendeur procède à classifier ces offres selon une classification donnée dans le but de résoudre WDPS, il va le faire en deux temps, de manière gloutonnière. Les classifications gloutonnières considèrerons en premier lieu les offres qui sont les mieux classées selon les spécifications de la classification en question, puis considèrerons les autres offres. Cela résultera à ce que les offres ayant le meilleur classement seront les premiers à être servis en ressources, puis sera le tour des offres qui les suivent dans le classement. Il faut noter que si une offre a été rejetée, cela ne va pas signifier que les offres qui la suivent dans le classement seront à leurs tours rejetées. Puisque ces dernières peuvent demander des ressources autres que l'offre qui a été rejetée et qui occupe un meilleur classement.

Ci-dessous, nous allons présenter la principale stratégie du vendeur dans l'algorithme 1. Il donne un aperçu sur les activités principales entretenues par le vendeur lors de sa négociation, et il sera suivi par l'algorithme 2 qui décrit quelles sont les directives à suivre par le vendeur après qu'il ait résolu le WDPS.

Algorithme 1 : Stratégie du vendeur

```

// phase d'initialisation : doit être exécuter qu'une seule fois lors de l'entrée du vendeur dans le marché.
Assigner  $\mathcal{DR}_s, \mathcal{A}_s, \mathcal{TA}_s, \sigma, o_{dl}, o_{rd} = \emptyset$ ;
si  $b$  se désengage d'un accord final  $o$  alors
| enlevez  $o$  de  $\mathcal{A}_s$ ;
fin
si  $b$  annule un accord temporaire  $o$  alors
| enlevez  $o$  de  $\mathcal{TA}_s$ ;
fin
si  $b$  confirme  $o \in \mathcal{TA}_s$  alors
| enlevez  $o$  de  $\mathcal{TA}_s$  et le placer dans  $\mathcal{A}_s$ ;
fin
// Extraire l'offre ayant la date limite la plus proche parmi l'ensemble des offres finaux contracté jusqu'à
// maintenant par le vendeur  $s$ 
 $o_{dl} = \underset{o \in \mathcal{A}_s}{arg\ min}(dl(o))$ 
// Vérifier si l'accord  $o_{dl}$  a été conclu (accord final définitif)
si  $(t = dl(o_{dl}))$  alors
| enlevez  $o_{dl}$  de  $\mathcal{A}_s$  et placer le dans  $\mathcal{DA}_s$ ;
|  $\mathcal{R}_s = \mathcal{R}_s - \mathcal{R}(o_{dl})$ ; // Retirez les ressources de  $o_{dl}$  de la procédure WDPS
fin
// Vérifier si le temps accordé pour un accord de location a été écoulé afin de mettre à nouveau en négociation les
// ressources loué par cet accord.
 $o_{rd} = \underset{o \in \mathcal{A}_s}{arg\ min}(dl(o) + pd(o));$ 

```

```

si ( $o_{rd} \neq null$  et  $t = (dl(o_{rd}) + pd(o_{rd}))$ ) alors
|   enlevez  $o_{rd}$  de  $\mathcal{DA}_S$ ;
|    $\mathcal{R}_S = \mathcal{R}_S + \mathcal{R}(o_{rd})$ ; // Libérez les ressources réservées et les remettre à nouveau en négociation
fin
si  $b$  envoie une offre  $\sigma$  alors
|   si  $pr(\sigma) < \phi(o)$  alors // si le prix d'offre est inférieur au prix de réserve du vendeur
|   |   envoyez à  $b$  les quotes acceptable pour les ressources demandées ;
|   fin
|   sinon
|   |   WDPS_RESOLUTION ( $\mathcal{R}_S, \mathcal{A}_S, \mathcal{TA}_S, \sigma$ );
|   |   WDPS_DIRECTIVES ( $\mathcal{R}_S, \mathcal{A}_S, \mathcal{TA}_S, \sigma$ );
|   fin
fin

```

Algorithme 2 : *WDPS_DIRECTIVES* ($\mathcal{R}_S, \mathcal{A}_S, \mathcal{TA}_S, \sigma$)

```

// Traitez chaque offre selon sa phase et son résultat du WDPS
pour ( $o \in (\sigma \cup \mathcal{TA}_S \cup \mathcal{A}_S)$ ) faire
|   si ( $rw dps(o) = success$ ) alors
|   |   si ( $stage(o) = "Offer"$ ) alors
|   |   |   envoyez un message d'acceptation au  $bidder(o)$ ;
|   |   |    $stage(o) = "accord temporaire"$ ;
|   |   fin
|   |   // si ( $rw dps(o) = succès$  ET ( $stage(o) = "accord temporaire"$  Ou " $accord final$ ")) // alors l'offre  $o$  est dans la
|   |   // phase « en attente », ce qui signifie qu'aucune action ne va être entreprise
|   fin
|   sinon //  $rw dps(o) = échec$ 
|   |   si  $stage(o) = "Offer"$  OU  $stage(o) = "accord temporaire"$  alors
|   |   |   envoyez un message de désengagement au  $bidder(o)$  impliquant aucun frais;
|   |   |   enlevez  $o$  de  $\mathcal{TA}_S$ ;
|   |   fin
|   |   sinon //  $stage(o) = "accord final"$  qui implique de payer une pénalité
|   |   |   envoyé un message de désengagement au  $bidder(o)$  avec la pénalité de désengagement correspondante ;
|   |   |   enlevez  $o$  de  $\mathcal{A}_S$ ;
|   |   fin
|   fin
fin

```

Au lieu de décrire la procédure de sous-jacente du **WDPS_RESOLUTION** ($\mathcal{R}_S, \mathcal{A}_S, \mathcal{TA}_S, \sigma$) sous le format d'un pseudo algorithme, nous avons choisi de le décrire à travers une description textuelle et par l'utilisation des équations 61, 62 et 63.

Le processus de résolution WDPS :

Comme mentionné dans l'algorithme 1, le processus **WDPS_RESOLUTION** ($\mathcal{R}_S, \mathcal{A}_S, \mathcal{TA}_S, \sigma$) prend quatre paramètres, qui sont : Le premier est \mathcal{R}_S qui représente les ressources disponibles

qui peuvent être commercialisées par le vendeur, car ce dernier doit se conformer à la contrainte qui est de ne pas négocier plus que cette quantité de ressources. Autrement dit, le vendeur doit satisfaire toutes les offres acceptées des ensembles \mathcal{DA}_s , \mathcal{A}_s et \mathcal{TA}_s après avoir exécuter WDPS. Le deuxième et troisième paramètres \mathcal{A}_s et \mathcal{TA}_s sont les offres qui sont encore sous la considération du vendeur et qui ont au moins passé avec succès un WDPS. Et finalement, le quatrième paramètre représente la nouvelle offre reçue σ qui est à l'origine de l'exécution du WDPS par le vendeur.

La stratégie WDPS entrepris par le vendeur consiste à appliquer deux classifications glouttonnières et choisir celle qui maximise le plus l'utilité attendue du vendeur. La première n'est autre que l'approche glouttonnière proposée par Bo An et.al [16], où les offres seront classifiées et évaluées selon leurs revenus respectifs. Plus formellement, la valeur de chaque offre pour la première classification est calculée comme suit :

$$v(o) = \begin{cases} pr(o) - cost(o) + pe(o) & \text{si } o \in \mathcal{A}_s \\ pr(o) - cost(o) & \text{sinon} \end{cases} \quad (61)[16]$$

Où « sinon » réfère à l'offre simple σ ou un des accords temporaire \mathcal{TA}_s . Dans l'équation 16, les accords finaux ont été avantagé par rapport aux accords non-finaux en ajoutant la valeur de pénalité de désengagement à leurs revenus. Cela s'explique du fait que le désengagement de l'acheteur d'un accord final va causer à ce dernier de payer une pénalité au vendeur ce qui n'est pas le cas pour les autres accords où l'acheteur peut s'en désengager gratuitement.

Après avoir évalué la première classification, le vendeur évaluera la deuxième classification glouttonnière qui consiste à évaluer chaque offre selon sa densité. La densité de l'offre est déterminée par le ratio du prix-offre/ressources-demandées. Le vendeur va assigner ses ressources de façon glouttonnière, en essayant en premier lieu de faire assigner ses ressources aux offres ayant les valeurs les plus élevées selon cette classification, puis passera aux offres avec les valeurs les moins élevées. La formule pour calculer l'évaluation de chaque offre sous cette classification est comme suit :

$$v(o) = \begin{cases} (pr(o) - cost(o) + pe(o))/\sqrt{\sum_{i=1}^n c_s(r_i) * q(r_i)} & \text{si } o \in \mathcal{A}_s \\ (pr(o) - cost(o))/\sqrt{\sum_{i=1}^n c_s(r_i) * q(r_i)} & \text{Sinon} \end{cases} \quad (62)$$

Où n est le nombre de types de ressources demandées par l'offre o .

Après avoir évalué les deux classifications, le vendeur va choisir celle qui maximise le plus l'utilité attendu du vendeur définit comme suit :

$$\sum_{o \in \mathcal{O}_s \cup \mathcal{T}_s \cup \mathcal{A}_s} (pr(o) - cost(o)) + \sum_{o \in \mathcal{A}_s} pe(o) \quad (63)$$

La classification que le vendeur adoptera par la suite va générer ses propres directives que ce dernier doit suivre. Ces directives sont décrites dans le pseudo algorithme 2. En résumé, les directives de la classification instruisent le vendeur de garder les offres qui ont passé avec succès le WDPS, et les autres doivent être désengager selon leurs type (final ou non-final).

8.4. Evaluation expérimentale

Pour des raisons de lisibilité, nous avons étiqueté la stratégie gloutonnière composé sur les figures suivantes comme « Composite-Greedy », l'algorithme gloutonier [16] comme « Greedy » et les schémas de prix fixe d'Amazon [4] comme "Amazon-X" où X représente le ratio prix/coût du vendeur. Les données utilisées dans l'étude comparative ont été inspirées par la conception actuelle de l'infrastructure GENI [3].

8.4.1. Configuration expérimentale

Tableau 10 Variables

Variabes	Valeurs
Nombre de vendeurs	[5, 12]
Nombre de type de ressources par vendeur	[2, 8]
Quantité de ressources par vendeur	[2, 15]
Coût unitaire de la ressource	[10, 100]
Prix de départ pour l'unité de ressource	[300, 400]
Nombre de types de ressources par acheteur	[2, 6]
Quantité de ressources par acheteur	[2, 6]
Ratio valeur/coût	[1.2, 5.0]

$dl(b)$	[22, 41]
$pd(b)$	[30, 70]
$\psi(r)$	[0.2, 6.0]
Tour de négociation	0.2

Exigences de l'acheteur : Les exigences de chaque acheteur seront générées à partir du tableau 10, dans lequel un acheteur requiert 2 à 6 ressources. Le nombre d'instances pour chaque ressource demandée par l'acheteur sera compris entre 2 et 6.

Les temps de l'acheteur : Le temps d'usage de chaque ressource sera compris entre 30 et 70 tandis que la date limite de chaque acheteur sera comprise dans l'intervalle [22, 41].

Prix des acheteurs : Chaque acheteur évaluera initialement chaque unité entre [300, 400], et fera de plus en plus de concessions au fil que la négociation progresse. Le *ratio valeur/coût* est utilisé pour générer la valeur maximale et minimale de chaque ressource.

Approvisionnements du vendeur : Le vendeur de son côté, peut fournir de 2 à 8 types de ressources avec une quantité de chaque ressource qui va de 2 à 8.

Ratio demande/ressource : Pour refléter chaque valeur du ratio demande/ressource qui provient du tableau 10, nous avons commencé par générer premièrement les approvisionnements du vendeur, et en se basant sur ça, nous avons ensuite généré les ressources de l'acheteur jusqu'à atteindre les différentes valeurs du ratio $\psi(r)$.

8.4.2. Observations et analyse des résultats

En raison des ressources computationnelles limitées dont nous disposant sur notre ordinateur de bureau équipé avec un processeur i5, nous avons mené seulement 1002 simulations sur les différents ratios d'offre et de demande. Puisque le bien être sociale des différents mécanismes pourraient être considérablement disparates, nous reportons le ratio du bien être sociale quand les vendeurs déploient la classification gloutonnière [16] et le schéma de prix fixe d'Amazon [4] au bien-être social où les vendeurs déploient la classification « composite-greedy » ainsi que l'utilité des acheteurs en moyenne. Elle sera suivie par cinq autres mesures de performances lesquelles sont représentatives des performances de la stratégie du vendeur, à savoir, le pourcentage des acheteurs

servis (taux de succès), le revenu (utilité) de la stratégie du vendeur, le temps d'exécution de l'approche ainsi que le taux d'utilisation en ressources.

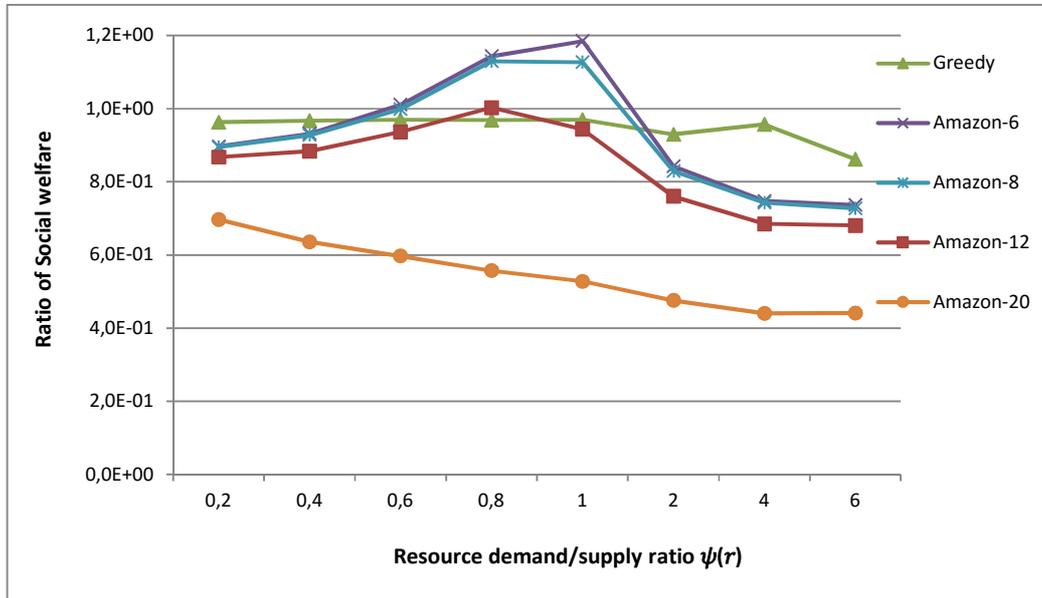


Figure 32 Bien-être social et compétition en ressources

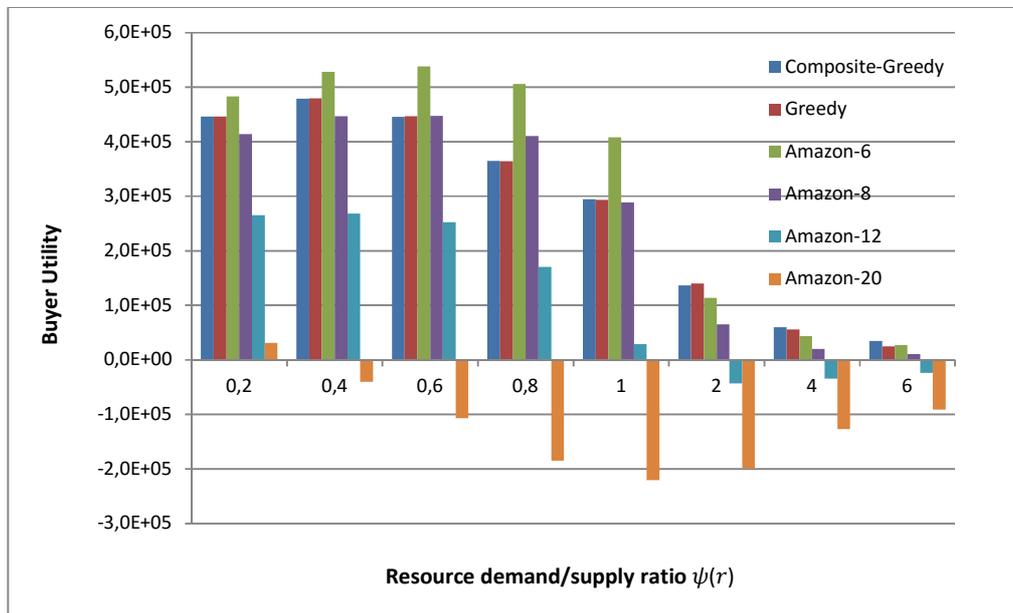


Figure 33 Utilité de l'acheteur en moyenne et compétition en ressources

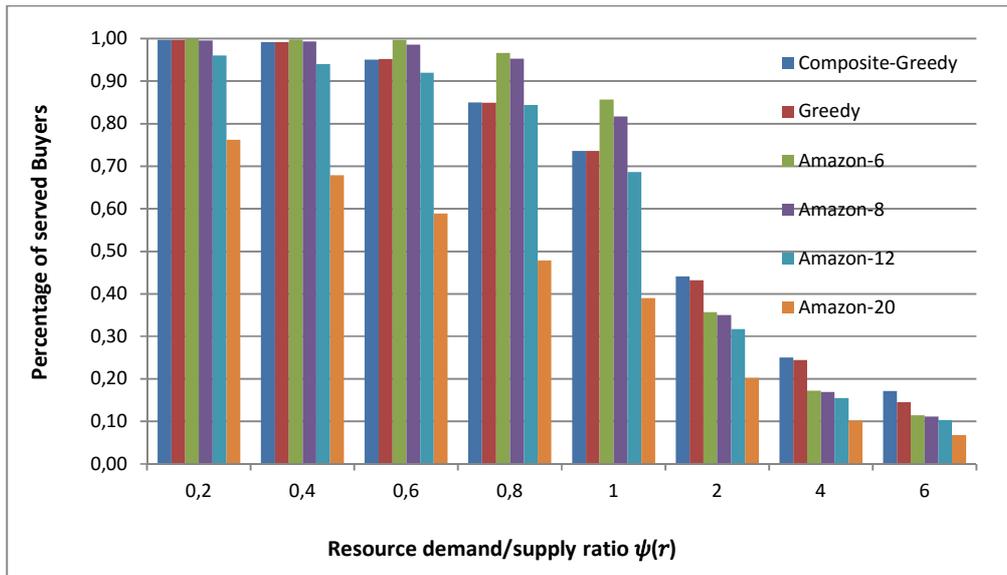


Figure 34 Acheteurs servis et compétition en ressources

A partir de la figure 32, la stratégie de négociation du vendeur proposé génère le bien-être social le plus élevé que tous les autres mécanismes évalués. Toutefois, les schémas de prix fixe Amazon ayant les prix les plus bas génèrent le bien-être social le plus élevé lorsque $\psi(r)$ prend les valeurs 0,8 et 1,0, car ils génèrent le taux d'acheteur servi le plus élevé, comme cela est montré dans la figure 34. En effet, nous pouvons observer à partir de la figure 34 que les deux stratégies gloutonnières, i.e., simple et composé, génèrent un taux d'acheteur servi relativement inférieur lorsque $\psi(r)$ vaut (0,6 ; 0,8 ou 1,0). Cela se passe puisque chaque agent n'acceptera ou n'offrira aucune offre moins que ses attentes. En outre, puisque le schéma de prix fixe d'Amazon n'inclut pas la possibilité de désengagement, les acheteurs "single-minded" seront condamnés de garder leurs ressources contractées inutilisées dans le cas où ils échouent à acquérir toutes leurs ressources requises pour démarrer leurs tâches, générant ainsi une utilité négative comme cela est montré dans la figure 33. De tels situations vont réduire le taux d'acheteur servi comme le montre la figure 34. En effet, ces deux effets négatifs qui se reflètent dans les figures 33 et 34 du côté de l'acheteur lors de l'application du schéma de prix fixe d'Amazon, vient du fait que ces schémas sont dépourvus de la caractéristique de désengagement. Cela implique que les acheteurs vont garder les ressources inutilisées et inutiles dans le cas où ils ne peuvent démarrer leurs tâches, ce qui pourrait même diminuer encore plus le taux d'acheteur servi représenté dans la figure 34 puisque ces ressources inutilisées pourraient être exploités par d'autre acheteurs pour les aider à démarrer leurs tâches. Nous pouvons aussi citer un troisième effet négatif de l'utilisation d'un protocole dépourvu de la

caractéristique de désengagement. Cet effet est le résultat de la diminution du bien-être social due à une diminution du taux d'acheteur servi qui est causé à son tour par la non-existence de la fonction de désengagement. La diminution du bien-être social est en corrélation avec la diminution du taux d'acheteur servi, puisque l'exécution des tâches des acheteurs représentent la principale source d'utilité du bien être sociale.

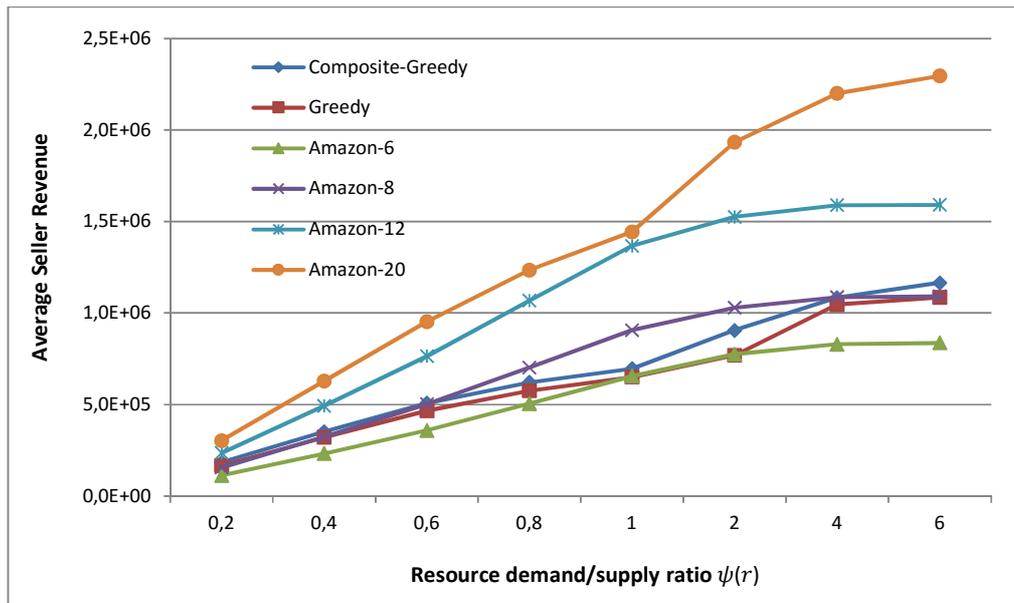


Figure 35 Revenu moyen du vendeur et compétition en ressources

Nous pouvons observer à partir de la figure 35, que la stratégie "composite-greedy" génère en moyenne une utilité plus élevée que la stratégie "greedy" [16]. Ceci est assez évident, puisque la stratégie vendeur proposée, inclut la stratégie vendeur défini dans [16], et va essayer de générer de meilleurs décisions en faisant une exploration plus approfondi à travers l'évaluation d'une classification additionnelle qui est basée sur la densité de l'offre. Ensuite le vendeur va choisir la meilleur parmi les deux, à chaque fois qu'il a à résoudre le WDPS. De plus, à partir de la figure 35, nous pouvons observer qu'il y a une relation entre le prix appliqué dans les schémas de prix fixe d'Amazon et le revenu généré par le vendeur. Ce dernier augmente par l'augmentation du prix appliqué par le schéma qui peut générer des revenus qui sont plus élevés que n'importe quel autre schéma dès que le ratio prix/coût atteint la valeur 12. Cependant, en observant les figures 33 et 35, on constate que l'augmentation de l'utilité du vendeur sur les schémas de prix fixe d'Amazon n'est pas sans effets secondaires. En effet, cela se fait au détriment de l'utilité de l'acheteur, le menant même parfois à générer une utilité négative en moyenne lorsque le ratio prix/coût du schéma

d'Amazon atteint un certain seuil. Cela pourrait être simplement expliqué par le fait que les schémas de prix fixe d'Amazon sont dépourvus de la caractéristique de désengagement, condamnant les acheteurs qui n'ont plus besoin des ressources déjà contractées de les garder et de les acheter, les menant ainsi à générer des pertes qui s'élèvent avec l'augmentation du ratio prix/coût du schéma d'Amazon.

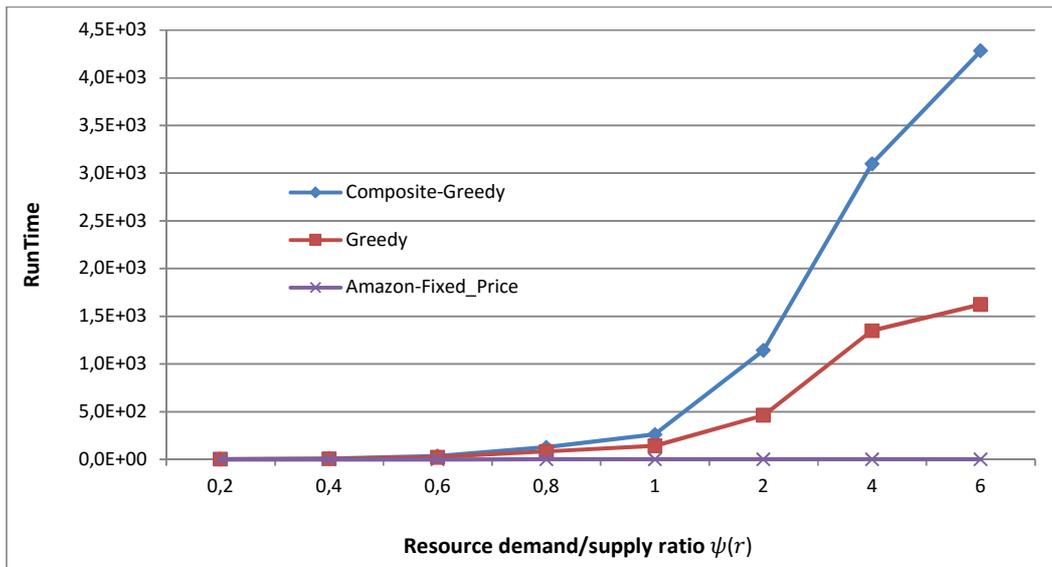


Figure 36 Temps d'exécution de la stratégie du vendeur et compétition en ressources

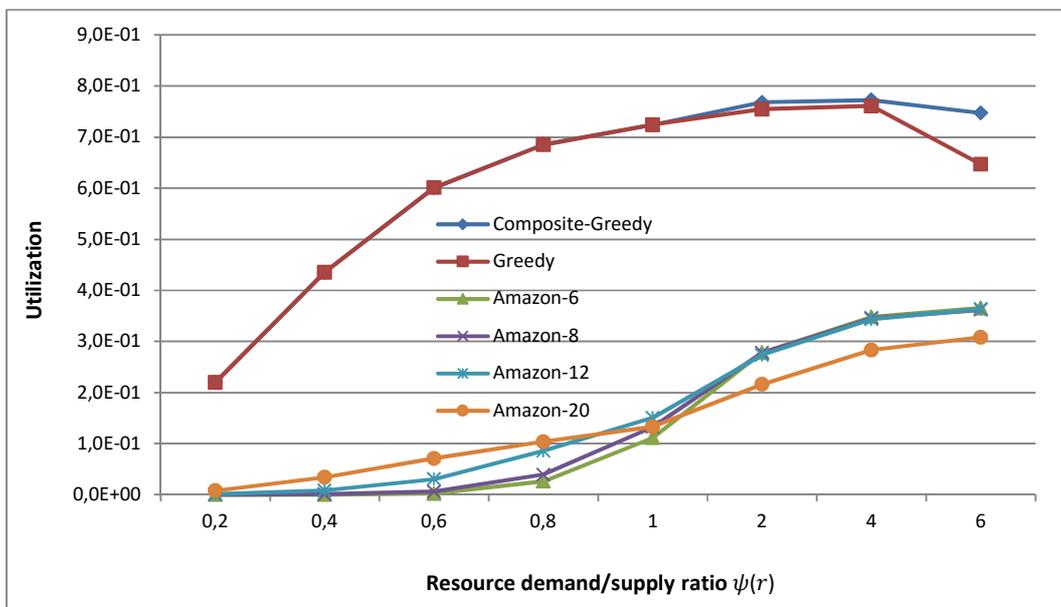


Figure 37 Taux d'utilisation et compétition en ressources

La figure 37 ci-dessus montre que la stratégie "composite-greedy" produit les meilleures performances dans cette métrique (taux d'exploitation des ressources), suivie par la stratégie "greedy" et les schémas de prix fixe d'Amazon. Ceci est en partie parce que la stratégie "composite-greedy" réalise une meilleure approximation que l'approche glouttonnière en essayant de maximiser l'utilité attendue du vendeur (eq. 63) grâce à l'évaluation de multiple stratégies (i.e., dans notre cas, il y a deux classifications qui sont évaluées contre une seule dans l'approche glouttonnière), et comme conséquence, cela mène à une utilisation de ressources encore plus efficace qui est reflétée à travers la figure 37. L'augmentation du taux d'utilisation de la ressource est en partie en corrélation avec l'augmentation du taux d'acheteurs servi, représenté dans la figure 34. Cette affirmation est prouvée par la relation observée entre les figures 34 et 37, ce qui est logique puisque le taux d'utilisation des ressources n'est rien d'autre que la proportion des ressources allouées à la communauté d'acheteurs. Cependant, il y a une certaine irrégularité dans cette relation lorsque $\psi(r)$ prend les valeurs (0,6; 0,8 ou 1,0) où des approches ayant un taux d'utilisation des ressources plus faible (schéma d'Amazon) parviennent à obtenir un taux d'acheteur servi plus élevé; Ceci est expliqué par le fait que la figure 34 ne montre seulement que le taux d'acheteurs servi de ceux qui ont démarré leurs tâches avec succès en ignorant ceux qui ont échoué, bien que ces derniers ont inutilement contractés une partie significative des ressources du marché qui a contribué à l'augmentation du taux d'utilisation de la ressources dans la figure 37.

En termes de temps d'exécution illustré par la figure 36, l'approche la plus efficace dans cette métrique est évidemment le schéma de prix fixe d'Amazon puisqu'elle alloue les ressources aux participants sur une base triviale : le premier arrivé est le premier servi, suivi par l'approche glouttonnière et l'approche "composite-greedy". Cette dernière réalise la mauvaise performance dans cette métrique due au fait qu'elle procède par deux classifications glouttonnières au lieu d'une seule pour [16] afin de déterminer la classification des offres des acheteurs.

8.5. Conclusions

Cette contribution présente une stratégie de négociation du vendeur adaptée aux problèmes d'optimisation combinatoire intraitable. Dans ces problèmes, de nombreux acheteurs et vendeurs interagissent entre eux en employant "negotiation thread", qui leurs permet d'effectuer plusieurs négociations en parallèle. Une approximation de ces problèmes consiste à décentraliser la charge de travail en faisant contribuer l'ensemble des participants aux décisions de négociations, en

d'autres termes, les décisions de négociations seront entreprises au niveau local. Cela signifie en partie que chaque vendeur tiendra sa propre vente aux enchères sur la base de ses propres ressources disponibles au lieu de se référer à une troisième entité centrale. Ajouter à cela, le vendeur doit prendre en compte le degré de l'engagement de chaque offre lorsqu'il mène l'enchère ; Le concept d'engagement à niveau combiné avec le protocole de négociation décentralisé nous permet de réaliser une meilleure approximation pour un tel problème d'allocation de ressources. Cependant, même si on applique un protocole de négociation décentralisé pour réduire la charge de travail du problème d'allocation de ressources, les acheteurs et vendeurs doivent adoptés des stratégies de calculs efficaces afin qu'aucun retard inacceptable ne se produit dans le système du marché. Après avoir comparé la stratégie du vendeur proposé "composite-greedy" aux autres stratégies relatives déployées dans le Cloud computing, nous déduisons que la stratégie proposée est le meilleur choix quand l'objectif est d'obtenir des revenus des vendeurs plus élevés et une meilleure utilisation des ressources tout en ne compromettant pas l'efficacité computationnelle de la stratégie du vendeur.



PARTIE 5 :

CONCLUSION

Chapitre 9 Conclusion et Recherches futures

Dans ce dernier chapitre, nous allons résumer nos contributions dans le domaine de la négociation plusieurs-à-plusieurs pour ressources à structure complexe. Dans la fin de ce chapitre nous allons aussi discuter des orientations futures de recherche.

9.1 Contributions

Le travail décrit dans cette thèse a réalisé des améliorations dans l'état de l'art dans le domaine de la négociation basée agent, en étendant les approches heuristiques concernant les négociations plusieurs-à-plusieurs dans le cloud par l'apprentissage de l'identité de l'agent afin de rendre les décisions de négociations plus équitables, l'exploration plus approfondie de l'espace d'accords afin d'augmenter les chances de l'acheteur à contracter ces ressources demandées, et finalement en concevant des stratégies vendeur qui ont le meilleur rapport revenu-généré/complexité-computationnelle.

Nous nous sommes basés sur le travail de Bo.A et.al [16] qui consistait à un mécanisme de négociation distribué approprié pour les problèmes d'allocation de ressources au sein des plateformes cloud computing. Dans le modèle de négociation traité, plusieurs acheteurs et vendeurs peuvent négocier simultanément entre eux où chaque agent a un intervalle de temps où il est possible pour lui de se désengager d'un contrat. De plus, les termes du contrat qui inclut : la quantité de ressources, le temps d'usage, le prix d'offre et la pénalité de désengagement seront négocié par les agents impliqués dans le contrat et non pas par une entité tierce tel que le commissaire-priseur comme c'est le cas avec les enchères. Une des caractéristiques distinctives des agents de négociation de ce modèle c'est qu'ils sont conçus avec la flexibilité nécessaire pour ajuster le taux de concession en réagissant à i) l'évolution des conditions du marché et ii) l'état de la négociation courant de tous les "threads" de négociation concurrentielles.

Nos contributions interviennent dans l'extension de ce mécanisme dont elles se résument comme suit :

- Dans la première proposition, les agents vont mutuellement s'évaluer pour mesurer la confiance de chaque agent qui sera un indicateur par la suite du taux de fidélité de l'agent

sur ces engagements, tandis que la deuxième proposition porte sur l'évaluation des différentes offres possibles que l'acheteur peut formuler aux vendeurs. C'est-à-dire permettre à l'acheteur d'explorer plus en profondeur son espace d'accords, ce qui permet au mécanisme d'atteindre certaines allocations de ressources inaccessibles si on vient à employer des approches relatives à [16].

- De plus, des situations mutuellement bénéfiques peuvent être trouvées entre les agents impliqués dans certain interblocages de négociation en recourant à des appels à proposition pour le désistement des ressources contractées.
- Enfin, nous avons amélioré la stratégie du vendeur dans les milieux de négociation décentralisée en proposant une stratégie gloutonnière composite qui fait mieux en termes de revenu que celle proposé dans [16] et fournit le meilleur apport revenue-complexité computationnelle.

9.2 Recherches futures

Les recherches futures en théorie de la négociation vont essayer d'adresser les problèmes de négociation complexes qui sont liés à *la multiplicité des issues, l'incomplétude de l'information*, et *la négociation impliquant plusieurs agents*. De notre côté, dans le future, nous souhaitons étudier les problèmes d'allocation de ressources relatif à ceux traités dans cette thèse mais qui ont la particularité d'avoir des dates limites plus longues et où il y a beaucoup plus d'acheteurs et de vendeurs.

Comme orientation future, nous pourrons considérés l'allocation des ressources dans de nouveaux champs d'applications telle que le grid. Aussi, nous pourrons considérer le rôle de retarder la prise de décisions des agents. Enfin, minimiser la diminution des pertes de l'acheteur en employant des approches qui permettent de se retirer du processus de négociation lorsque l'acheteur a peu de chances de réussir est un domaine intéressant qui serait considéré aussi comme l'une de nos perspectives d'avenir.

Référence bibliographiques

- [1] http://www-igm.univ-mlv.fr/~dr/XPOSE2006/Jolly_Laskri/principesdebase.html
- [2] <http://www.cloudbus.org/cloudsim/>
- [3] GENI System Overview. <http://groups.geni.net/geni/wiki/GeniSysOvrvw>.
- [4] Amazon ec2. <http://aws.amazon.com/ec2/>.
- [5] The Network Simulator Ns2 (2010) Available at: <http://www.isi.edu/nsnam/ns/>
- [6] Google app engine. <http://code.google.com/appengine/>.
- [7] Opennebula. <http://dev.opennebula.org/>.
- [8] Xtreemos. <http://www.xtreemos.eu/>.
- [9] https://en.wikipedia.org/wiki/Cloud_computing_security
- [10] Futuregrid. <https://portal.futuregrid.org/>.
- [11] Almeida A., Ramalho G. L., Santana H. P., Tedesco P., Menezes T. R., Corruble V., et Chevalyre Y., (2004). Recent advances on multi-agent patrolling. In *Advances in Artificial Intelligence SBIA 2004 : 17th Brazilian Symposium on Artificial Intelligence, Sao Luis, Maranhao, Brazil, Lecture notes on Artificial Intelligence*.
- [12] An B., (2011). Automated negotiation for complex multi-agent resource allocation. PhD thesis, University of Massachusetts Amherst.
- [13] An Bo, Nicola, Gatti, and Victor Lesser,. Bilateral bargaining with one-sided two-type uncertainty. In *Proc. of the 9th IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT) (Sep. 2009)*, pp. 403–410.
- [14] An B., Gatti N., and Lesser V. Extending alternating-offers bargaining in one-to-many and many-to-many settings. In *Proc. of the 9th IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT) (Sep. 2009)*, pp. 423–426.
- [15] An B.; Lesser V.; M. Sim K. Strategic Agents for Multi-resource Negotiation. *Autonomous Agents and Multi-Agent Systems*, Volume 23, Number 1, Springer Netherlands, pp. 114-153. 2011.

- [16] An B.; Lesser. V., Irwin D., and Zink M. Automated negotiation with decommitment for dynamic resource allocation in cloud computing. In Proceedings of the ninth international joint conference on autonomous agents and multi-agent systems, pages 981–988. Citeseer, 2010.
- [17] An, Bo, Nicola Gatti, and Victor Lesser. "Bilateral bargaining with one-sided uncertain reserve prices." *Autonomous agents and multi-agent systems* 26.3 (2013): 420-455.
- [18] Arrow K. J., Sen A. K., et Suzumura K., editors (2002). *Handbook of Social Choice and Welfare*, volume 1. North-Holland.
- [19] Assuncao A.C.M., Buyya R.: Evaluating the cost-benefit of using cloud computing to extend the capacity of clusters. In: Kranzlmüller, D., Bode, A., Hegering, H.-G., Casanova, H., Gerndt, M. (eds.) *11th IEEE International Conference on High Performance Computing and Communications, HPCC 2009*. ACM, New York (2009).
- [20] Bouveret S. (2005). *Partage de ressource : modélisation, classification et applications concrètes*. Technical report, Séminaires de l'unité Conduite et Décision du Département Commande des Systèmes et Dynamique du vol.
- [21] Branson M., Douglis, F., Fawcett, B., Liu, Z., Riabov, A., & Ye, F. (2007). CLASP: Collaborating, autonomous data stream processing systems. In Proceedings of the ACM/IFIP/USENIX 8th international middleware conference (pp. 348–367).
- [22] Buyya R., Yeo C.S., Venugopal S., Broberg J., and Brandic I. *Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility*. *Future Generation Computer Systems*, 25(6): 599-616, Elsevier Science, June 2009.
- [23] Calheiros R. N., Ranjan R., Beloglazov A., De Rose C. A. F., and Buyya R., *CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms*, Wiley Press, New York, USA, January, 2011.
- [24] Chevaleyre, Yann, et al. "Issues in multiagent resource allocation." *Informatica* 30.1 (2006).
- [25] Chimakurthi L. and SD M. K., "Power efficient resource allocation for clouds using ant colony framework" *CoRR*, vol. abs/1102.2608, 2011.

- [26] Cramton P., Shoham Y. and Steinberg R. *Combinatorial Auctions*. MIT Press, 2006.
- [27] Crawford E. and Veloso M., “Learning to select negotiation strategies in multi-agent meeting scheduling,” in *Working Notes of the AAI Workshop on Multiagent Learning*, 2005.
- [28] Dash R.K., Parkes D. C., and Jennings N. R. Computational mechanism design: A call to arms. *IEEE Intelligent Systems*, 18(6):40–47, 2003.
- [29] Dean J. and Ghemawat S. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.
- [30] De Farias D. P. and Megiddo N. How to combine expert (and novice) advice when actions impact the environment ? In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.
- [31] de Vries, S. and Vohra, S. (2003). *Combinatorial Auctions: A Survey*. *INFORMS Journal on Computing*, 15(3):284–309.
- [32] Dumas M., Aldred L., Governatori G., and A. Hofstede. “Probabilistic Automated Bidding in Multiple Auctions,” *Electronic Commerce Research*, vol. 15, no. 1, pp. 23–47, Jan. 2005.
- [33] Dunne P. E. Extremal behaviour in multiagent contract negotiation. *Journal of Artificial Intelligence Research* 23 (2005), 41–78.
- [34] Ejarque J. A Multi-agent Approach for Semantic Resource Allocation. 2010 IEEE Second International Conference on Cloud Computing Technology and Science, pp. 335- 342, 2010.
- [35] Endriss U., Maudet N., Sadri F., and Toni F. 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.
- [36] Endriss U., Maudet N., Sadri F., and Toni F., (2006). Negotiating socially optimal allocations of resources. *Journal of Artificial Intelligence Research*, 25 :315–348.
- [37] Endriss U. and Maudet N. On the communication complexity of multilateral trading: extended report. *Journal of Autonomous Agents and Multi-agent Systems* 11, 1 (2005), 91–107.

- [38] Estivie S., (2007), “Allocation de Ressources Multi-Agents : Théorie et Pratique,” Ph.D. thesis, Université Paris Dauphine, LAMSADE.
- [39] Faratin P. Automated Service Negotiation Between Autonomous Computational Agents. PhD thesis, University of London, Queen Mary and Westfield College, Department of Electronic Engineering, 2000
- [40] Fatima S. S., Wooldridge M., and Jennings N. R. Multi-issue negotiation with deadlines. *Journal of Artificial Intelligence Research* 27 (2006), 381–417.
- [41] Foster I., Y. Zhao, I. Raicu, and S. Lu. Cloud computing and grid computing 360-degree compared. In *Proceedings of Grid Computing Environments Workshop*, pages 1–10, 2008.
- [42] Fudenberg D., and Tirole J., *Game Theory*. MIT Press, Cambridge, MA, 1991.
- [43] Fujiwara I., Aida K. and Ono I. Applying Double-sided Combinatorial Auctions to Resource Allocation in Cloud Computing. In *Conference 10th IEEE/IPSJ International Symposium on Applications and the Internet (SAINT)*, pages 7 – 14. IEEE, 2010.
- [44] Fujishima Y., Leyton-Brown K., and Shoham Y. (1999). Taming the Computational Complexity of Combinatorial Auctions. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI-99)*, pages 548–553.
- [45] Gatti N., Giunta F. D. and Marino S. Alternating-offers bargaining with one-sided uncertain deadlines: an efficient algorithm. *Artificial Intelligence* 172, 8-9 (2008), 1119–1157.
- [46] Ghani R. and Simmons H., “Predicting the end-price of online auctions”, *Decision Support Systems* 44 (2008) 970-982, 2004.
- [47] Gimpel H., Jennings N. R., Kersten G., Okenfels A., and Weinhardt C. *Negotiation, Auctions, and Market Engineering*. Springer, 2008, ch. Market engineering : a research agenda, pp. 1–15.
- [48] Goudarzi H. and Pedram M., "Maximizing profit in the cloud computing system via resource allocation," *Int'l workshop on Data Center Performance*, Minneapolis, MN, Jun. 2011.

- [49] Haresh M. V.; Kalady S.; Govindan V. K. Agent based dynamic resource allocation on federated clouds. Recent Advances in Intelligent Computational Systems (RAICS), 2011 IEEE, pp. 111- 114, 2011.
- [50] Hoos H., and Boutilier C., (2000). Solving Combinatorial Auctions Using Stochastic Local Search. In Proceedings of the Seventeenth National Conference on Artificial Intelligence, pages 22–29.
- [51] Hurwitz, J., Bloor, R., Kaufman, M., & Halper, F. (2010). Cloud computing for dummies. John Wiley & Sons.
- [52] Jawad A., et al. "Combinatorial auctions." Annals of Operations Research 153.1 (2007): 131-164.
- [53] Jennings N. R., Norman T. J., Faratin P., O'Brien P., and Odgers B. Autonomous agents for business process management. Applied Artificial Intelligence 14, 2 (2000), 145–189.
- [54] Jennings N. R., Faratin P., Lomuscio A. R., Parsons S., Sierra C., and Wooldridge M. Automated negotiation: prospects, methods and challenges. Group Decision and Negotiation 10, 2 (2001), 199–215.
- [55] Jung G.; Sim K. M. Agent-Based Adaptive Resource Allocation on the Cloud Computing Environment. 2011 40th International Conference on Parallel Processing Workshops (ICPPW), pp. 345- 351, 2011.
- [56] Kalagnanam J., Davenport A., and Lee H., (2001). Computational Aspects of Clearing Continuous Call Double Auctions with Assignment Constraints and Indivisible Demand. Electronic Commerce Research Journal, 1(3) : 221–237.
- [57] Kliazovich D., Bouvry P., and Khan S., “Greencloud: a packet-level simulator of energy-aware cloud computing data centers,” The Journal of Supercomputing, 2010.
- [58] Kraus S., Sycara K., and Evenchik A. Reaching agreements through argumentation: A logical model and implementation. Artificial Intelligence, 104(1–2) : 1–69, 1998.
- [59] Kraus S., Strategic Negotiation in Multi-Agent Environments. MIT Press, Cambridge MA, USA, 2001.

- [60] Kowalczyk R. and Bui V. On constraint-based reasoning in e-negotiation agents. In F. Digham and U. Cortés, editors, *Agent-Mediated Electronic Commerce III*, volume 2003 of *Lecture Notes in Computer Science*, pages 31–46. Springer Verlag, Berlin, Germany, 2001.
- [61] Lai, G., Sycara, K., & Li, C. (2008). A decentralized model for automated multi-attribute negotiations with incomplete information and general utility functions. *Multiagent and Grid Systems*, 4(1), 45–65.
- [62] Lee C. and Snavely A. Precise and realistic utility functions for user-centric performance analysis of schedulers. In *Proc. of the 16th Int. Symp. on High Performance Distributed Computing*, pages 107–116, 2007.
- [63] Leyton-Brown, K., Shoham, Y., and Tennenholz, M. (2000). An Algorithm for Multi-unit Combinatorial Auctions. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence*, pages 56–61.
- [64] Leyton-Brown, K. (2003). *Resource Allocation in Competitive Multiagent Systems*. Doctoral dissertation, Stanford University.
- [65] Liad B., and Nisan N. *Algorithmic Game Theory*. Cambridge University Press, 2007, ch. Combinatorial Auctions.
- [66] Lim S-H, Sharma B, Nam G, Kim EK, Das CR (2009) MDCSim: a multi-tier data center simulation, platform. In: *IEEE international conference on cluster computing and workshops (CLUSTER)*.
- [67] Martello S., and Toth P., (1997). *Knapsack Problems: Algorithms and Computer Implementation*. Wiley & Sons.
- [68] Medernach E., (2011), “Allocation de ressources et ordonnancement multi-utilisateurs : une approche basée sur l’équité”, Ph.D. thesis, Université Blaise Pascal - Clermont-Ferrand II.
- [69] Mell P. and Grance T. *The NIST Definition of Cloud Computing (Draft)*. National Institute of Standards and Technology, 53:7, 2010.

- [70] Mihailescu M. and Teo Y. Strategy-Proof Dynamic Resource Pricing of Multiple Resource Types on Federated Clouds. *Algorithms and Architectures for Parallel Processing*, pages 337–350, 2010.
- [71] Moulin H., (1988). *Axioms of Cooperative Decision Making*. Cambridge University Press.
- [72] Müller R., (2006). Tractable Cases of the Winner Determination Problem. In P. Cramton, Y. Shoham, R. S., editor, *Combinatorial Auctions*, chapter 13. MIT Press.
- [73] Navendu J., Lisa A., Andrade H., Richard K., Yoonho P., Philippe S., and Chitra V. Design, implementation, and evaluation of the linear road benchmark on the stream processing core. In *Proc. of the 25th ACM SIGMOD International Conference on Management of Data (2006)*, pp. 431–442.
- [74] Nguyen T. D. and Jennings N. R. Coordinating multiple concurrent negotiations. In *Proc. of the 3rd International Conference on Autonomous Agents and Multi-Agent Systems (2004)*, pp. 1064–1071.
- [75] Nguyen T. D. and Jennings N. R. Managing commitments in multiple concurrent negotiations. *Electronic Commerce Research and Applications* 4, 4 (2005), 362–376.
- [76] Nisan N. Bidding languages for combinatorial auctions. In P. Cramton et al., editors, *Combinatorial Auctions*. MIT Press, 2006.
- [77] Núñez, A., Vázquez-Poletti, J. L., Caminero, A. C., Castañé, G. G., Carretero, J., & Llorente, I. M. (2012). iCanCloud: A flexible and scalable cloud infrastructure simulator. *Journal of Grid Computing*, 10(1), 185-209.
- [78] Ostermann, S., Plankensteiner, K., Prodan, R., & Fahringer, T. (2011, January). GroudSim: an event-based simulation framework for computational grids and clouds. In *Euro-Par 2010 Parallel Processing Workshops* (pp. 305-313). Springer Berlin Heidelberg.
- [79] Pawan Kumar & Gaganjot Kaur , “Study of Comparison of Various Cloud Computing Simulators”, IITT College of Engineering & Technology, 2nd National Conference in Intelligent Computing & Communication, GCET Greater Noida, India
- [80] Plummer D. C., Cearley D. W. and D. M. Smith. Cloud computing confusion leads to opportunity. Technical report, Gartner Research, 2008.

- [81] Park, S. and Rothkopf, M. (2005). Auctions with Bidder-Determined Allowable Combinations. *European Journal of Operational Research*, 161:399–415.
- [82] Ponka I. and Jennings N.R., "Changing Circumstances and Leveled Commitment: A Compensatory Approach to Contracting," *cec-eee*, pp.185-191, The 9th IEEE International Conference on E-Commerce Technology and The 4th IEEE International Conference on Enterprise Computing, E-Commerce and E-Services (CEC-EEE 2007), 2007.
- [83] Poundstone W. *Prisoner's Dilemma*. Anchor, New York, USA, 1993.
- [84] Rahwan, Iyad. *Interest-based negotiation in multi-agent systems*. University of Melbourne, Department of Information Systems, 2005.
- [85] Rahwan I., Kowalczyk R., and Pham H. H. Intelligent agents for automated one-to-many e-commerce negotiation. In M. Oudshoorn, editor, *Proceedings of the 25th Australasian conference on Computer science*, pages 197–204. Australian Computer Society Press, 2002.
- [86] Regan K., Poupart P., and Cohen R. Bayesian reputation modeling in E-marketplaces sensitive to subjectivity, deception and change, *proceedings of the 21st national conference on Artificial intelligence*, p.1206-1212, July 16-20, 2006, Boston, Massachusetts.
- [87] Rosenschein J. and Zlotkin G.. *Rules of Encounter: Designing Conventions for Automated Negotiation among Computers*. MIT Press, Cambridge MA, USA, 1994.
- [88] Rothkopf, M., Pekeć, A., and Harstad, R. (1998). Computationally manageable combinatorial auctions. *Management Science*, 44:1131–1147.
- [89] Rothkopf, M. and Park, S. (2001). An Elementary Introduction to Auctions. *Interfaces*, 31(6):83–97.
- [90] Rubinstein, A. Perfect equilibrium in a bargaining model. *Econometrica* 50, 1 (1982), 97–109.
- [91] Rubinstein R A. *Modeling Bounded Rationality*. MIT Press, Cambridge MA, USA, 1997.
- [92] Sakr S.; Liu A.; Batista D.M., and Alomari M. A Survey of Large Scale Data Management Approaches in Cloud Environments. In *Communications Surveys & Tutorials*, IEEE, v13 i.3, p. 311 – 336. IEEE 2011.

- [93] Samuelson L. *Evolutionary Games and Equilibrium Selection*. MIT Press, Cambridge MA, USA, 1998.
- [94] Sandholm T., “Algorithm for optimal winner determination in combinatorial auctions,” *Artificial Intelligence*, vol. 135, no. 12, pp. 1–54, 2002.
- [95] Sandholm T. Agents in electronic commerce: component technologies for automated negotiation and coalition formation. *Autonomous Agents and Multi-Agent Systems* 3, 1 (2000), 73–96.
- [96] Sandholm T., and Lesser V. Leveled commitment contracts and strategic breach. *Games and Economic Behavior* 35 (2001), 212–270.
- [97] Sandholm T., and Suri S., (2003). BOB: Improved Winner Determination in Combinatorial Auctions and Generalizations. *Artificial Intelligence*, 145:33–58.
- [98] Sandholm T. W., (1993). An implementation of the contract net protocol based on marginal cost calculations. In *Proceedings of the 11th National Conference on Artificial Intelligence (AAAI-1993)*. AAAI Press.
- [99] Sandholm R T. eMediator: A next generation electronic commerce server. *Computational Intelligence, Special issue on Agent Technology for Electronic Commerce*, 18(4):656 676, 2002a.
- [100] Shoham Y. and Leyton-Brown K., *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge Univ Pr, 2009.
- [101] Sierra C., Jennings N. R, P. Noriega, and Parsons S. A framework for argumentation-based negotiation. In M. Singh, A. Rao, and M. J. Wooldridge, editors, *Intelligent Agent IV: 4th International Workshop on Agent Theories, Architectures and Languages (ATAL-1997)*, volume 1365 of *Lecture Notes in Artificial Intelligence*, pages 177–192. Springer Verlag, Berlin, Germany, 1998.
- [102] Sim K. M., and Shi B.. Adaptive commitment management strategy profiles for concurrent negotiations. In *Proc. of The first International Workshop on Agent-based Complex Automated Negotiations (2008)*, pp. 16–23.

- [103] Sim, K. M., & Shi, B. (2010). Concurrent negotiation and coordination for grid resource co-allocation. *IEEE Transaction on Systems, Man and Cybernetics, Part B*, 40(3), 753–766.
- [104] Slade C. Reasons to buy: The logic of advertisements. *Argumentation*, 16:157– 178, 2002.
- [105] Sun D.; Chang G.; Wang C.; Xiong Y. and Wang X. Efficient Nash equilibrium based cloud resource allocation by using a continuous double auction. 2010 International Conference on Computer Design and Applications (ICCD), Vol. 1, pp. 94-99, 2010.
- [106] Teng F., “Resource Allocation and Scheduling Models for Cloud Computing”, tel-00659303, version 1- 12, 2012.
- [107] Von Stengel B. Computing equilibria for two-person games. In R. Aumann and S. Hart, editors, *Handbook of game theory*, volume 3, chapter 45, pages 1723–1759. Elsevier Science Publishers (North-Holland), Amsterdam, Netherlands, 2002.
- [108] Wang R. Auctions versus posted-price selling. *The American Economic Review*, vol. 83, no. 4, pp. 838–851, 1993.
- [109] Wei G., Vasilakos A. V., Zheng Y. and Xiong N. "A game-theoretic method of fair resource allocation for cloud computing services", *J. Supercomput.*, vol. 54, pp.252-269 2010.
- [110] Wickremasinghe B., Calheiros R. and Buyya R. CloudAnalyst: A CloudSim-based visual modeller for analysing cloud computing environments and applications. *Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications (AINA 2010)*, Perth, Australia, 20–23 April 2010; 446–452.
- [111] Wurman, Peter R., Michael P. Wellman, and William E. Walsh. "The Michigan Internet AuctionBot: A configurable auction server for human and software agents." *Proceedings of the second international conference on Autonomous agents*. ACM, 1998.
- [112] Yazir Y. O.; Matthews C.; Farahbod R.; Neville S.; Guitouni A.; Ganti S.; Coady Y. Dynamic Resource Allocation in Computing Clouds Using Distributed Multiple Criteria Decision Analysis. 2010 IEEE 3rd International Conference on Cloud Computing (CLOUD), pp. 91-98, 2010.

- [113] Zaman Sharrukh and Grosu Daniel. Combinatorial auction-based allocation of virtual machine instances in clouds. In *CloudCom*, pages 127–134, 2010.
- [114] Zhang J. and Cohen R.: Evaluating the trustworthiness of advice about selling agents in e-marketplaces: A personalized approach. *Electronic Commerce Research and Applications* 7(3), 330–340 (2008)
- [115] Zhang J. and Cohen R., “A framework for trust modeling in multi-agent electronic marketplaces with buying advisors to consider varying seller behavior and the limiting of seller bids,” *ACM Trans. Intell. Syst. Technol.*, vol. 4, no. 2, pp. 24:1–24:22, 2013.
- [116] Zhang Q., Gürses E., Boutaba R. and Xiao J. Dynamic resource allocation for spot markets in clouds, *Proceedings of the 11th USENIX conference on Hot topics in management of internet, cloud, and enterprise networks and services*, March 29, 2011, Boston, MA.

Publications Internationales

Mohamed Raouf HABES, Habiba Belleili-Souici. (2015). Towards a Fairer Negotiation for Dynamic Resource Allocation in Cloud by Relying on Trustworthiness. *International Journal of Grid and Utility Computing* (accepted).

Habes, M. R., Belleili-Souici, H., & Vercouter, L. (2014). A dynamic hybrid buyer bidding strategy for decentralized and cloud-based many-to-many negotiation. *Multiagent and Grid Systems*, 10(3), 165-183.

Communication Internationale

HABES Mohamed Raouf. *Dynamic Resource Allocation on Federated Clouds with decommitment and confidence factors*. 14th European Agent Systems Summer School. 28 May to 01 Juin, 2012 Valencia, Spain.

Annexe A : CloudSim : Outil de simulation

A.1 Introduction

Parce que l'utilisation des bancs d'essai réel limitent le champ d'expérience aux ressources physiques et qu'une étude à plus large échelle n'est pas possible à moins de les en procurées réellement, l'évaluation théorique classique s'avère donc extrêmement coûteuse et ne produira pas les résultats escomptés en termes de performance de service et de génération de revenu. Des approches alternatives plus sophistiquées ont été conçus pour faire face à ses problèmes et avancer le développement des technologies du cloud Computing. Les approches basé-simulation offrent quant à eux des bénéfices significatifs, en permettant de tester leurs services gratuitement de façon répété, contrôlable, et scalable avant de les déployer sur des clouds réels.

De notre côté, nous avons opté comme outil de simulation pour le cloud computing le « CloudSim » [2] qui a été inventé à partir du projet CloudBus par l'Université de Melbourne, Australie, puisqu'on voit en lui une plateforme qui est généralisable et extensible, et procure aux développeurs une modélisation, simulation et expérimentation correcte des infrastructures et des services d'application du cloud. Enfin, les détails de bas niveau sont entièrement prisent en charge par le système du CloudSim lui-même. Cette section décrira les principales fonctions du CloudSim, étant donné que c'est le simulateur que nous avons utilisé pour mener l'ensemble des expériences présentées dans cette thèse.

A.2 Les fonctionnalités générales du CloudSim

Parmi les fonctionnalités importantes du CloudSim, on peut citer [2] :

- Support pour la modélisation et la simulation des Datacenters du cloud computing à large échelle.
- Support pour la modélisation et la simulation de serveurs host virtualisés, avec des politiques d'approvisionnement en ressources aux machines virtuelles ajustables.
- Support pour la modélisation et la simulation de ressources de traitement dont on peut suivre leurs consommations en énergie.

- Support pour la modélisation et la simulation des topologies réseaux de data center et la transmission de messages.
- Support pour la modélisation et la simulation des clouds fédérées.
- Support pour l'insertion dynamique d'élément de simulation, l'arrêt et la reprise de la simulation.
- Support pour les politiques d'approvisionnement sur l'allocation des hosts aux machines virtuels.

A.3 Enregistrement et communication entre les entités

La communication qui implique l'enregistrement, la réservation et l'achat de la ressource se déroule entre trois entités : Datacenter, CloudInformationService et DatacenterBroker [2].

L'entité *Datacenter* qui regroupe un ensemble de Host qui sont composés d'un ensemble de ressources physique tel que (Mips, Ram, Bande passante, et le stockage) et des spécifications logique tel que (architecture, Vmm, Os, fuseau horaire, tarification à la seconde du processeur, tarification par octet pour la Ram, Bande passante, et le stockage). L'entité Datacenter se charge aussi de la facturation des dettes à ses consommateurs.

Concernant l'entité *courtier*, nommée *DatacenterBroker* dans le système CloudSim, il joue à la fois, le rôle d'acheteur, puisque c'est la Classe Broker qui règle la facture et propose le prix d'achat, et aussi le rôle de courtier en allant s'informer auprès de la classe CloudInformationService sur les différentes caractéristiques des Datacenters qui sont présent dans le marché.

Enfin la classe CloudInformationService (CIS) a pour fonction d'enregistrer chaque Datacenter qui vient de faire son entrée dans le marché. Côté utilisateur, il sert de registre pour consulter les différentes caractéristiques des Datacenters qui sont présent dans le marché.

Voici un schéma qui décrit ce que nous avons écrit jusque-là :

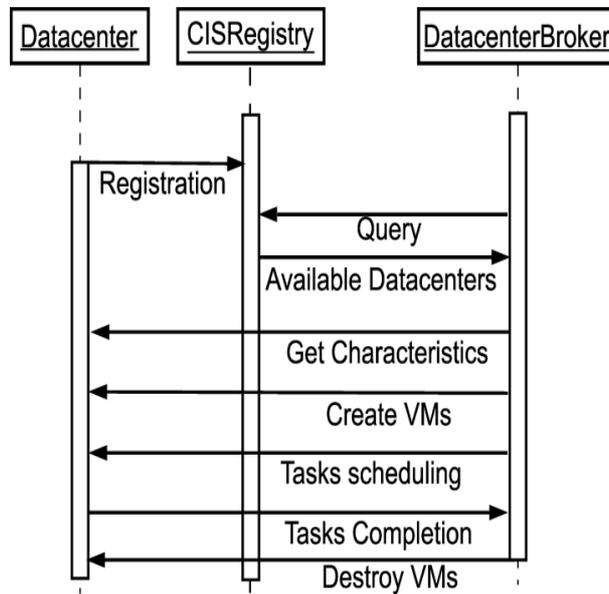


Figure 38 Flux de données de Simulation [23]

A.4 Système de simulation du CloudSim

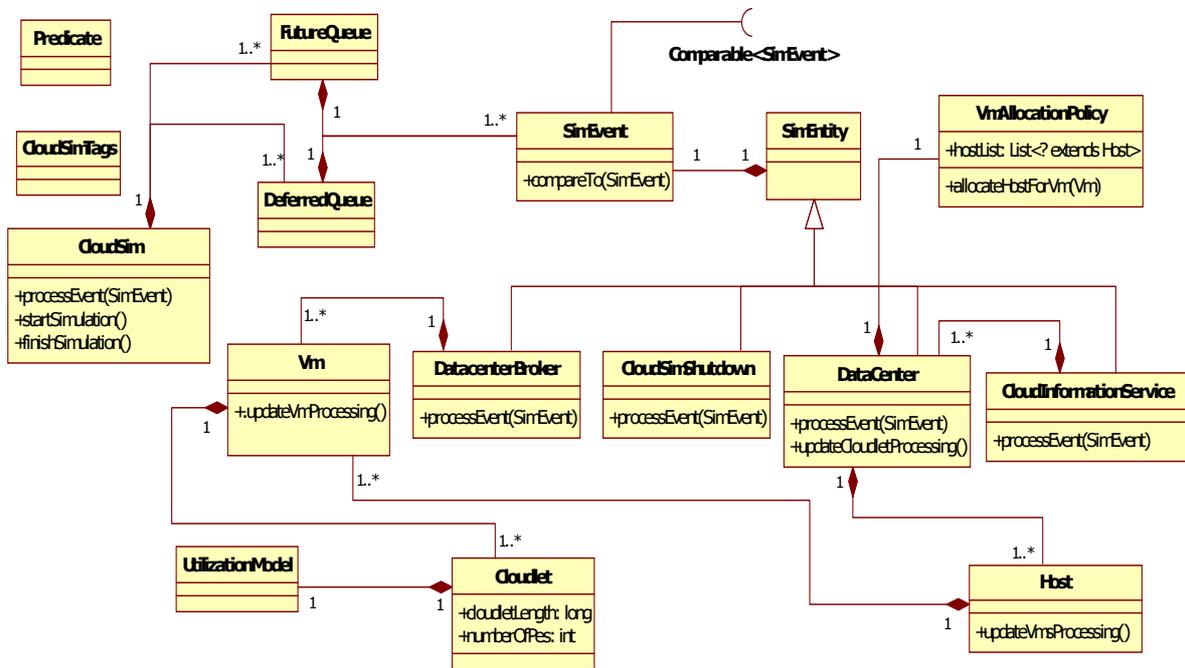


Figure 39 Diagramme de Classe du noyau du système de simulation CloudSim

Puisque la communication entre les différentes entités du CloudSim se passe par l'usage d'événements. Ces événements doivent porter toutes les informations nécessaires pour la communication, et l'une des informations importantes que le message doit détenir est son ordonnancement. En effet chaque réseau réel a un peu de retard qui est causé par le trafic, et il est exprimé en simulation en indiquant juste le temps que le paquet doit mettre pour qu'il arrive à sa destination. Pour gérer ces événements (arrivé et réception de messages) qui sont planifiés pour se produire à différentes ordre chronologique, la classe noyau de CloudSim a été fourni avec deux listes appelées `DeferredQueue` et `FutureQueue` [2]. C'est dans ces deux listes, que tous les événements qui transitent dans le système seront stockés, triés, et transmis à leurs destinataires respectives. Le traitement d'évènements se passe en deux étapes. La première étape consistera à formuler le message à envoyer, en remplissant toutes les informations nécessaires dont la syntaxe générale est comme suit :

Message (`Id_Entité_Src` ; `Id_Entité_Dest` ; `Délai` ; `CloudSimTags` [2] ; `Objet`)

`Id_Entité_Src` : représente l'identifiant de l'entité émettrice.

`Id_Entité_Dest` : représente l'identifiant de l'entité destinatrice.

`Délai` : représente le délai de transmission (latence) qui est causé par le réseau.

`CloudSimTag` : représente le type d'action à entreprendre par l'entité réceptrice.

`Objet` : représente l'objet à transporter par le message.

A.5 Politique d'approvisionnement de la ressource

Dans CloudSim, quatre types prédéfinis de ressources peuvent être louées : élément de Traitement (Pe), mémoire, disque dur et bande passante.

Avant d'attribuer n'importe quelles ressources, la classe `Datacenter` entreprendra deux étapes sous-jacentes auparavant :

1. Désigner quelles seront les centres de données qui doivent prendre en charge les ressources de l'utilisateur,

2. Ensuite, à l'intérieur de chaque centre de données choisi, sélectionner l'hôte qui va prendre en charge les machines virtuelles de chaque utilisateur (Exigences de l'utilisateur). La composante VmAllocationPolicy [23] du Datacenter est chargée à cet effet.

A.5.1 Allocation des ressources de stockage, bande passante et Ram

L'approvisionnement de Ram, bande passante sont effectuées par les composantes RamProvisionner et BwPronisionner dans lesquelles l'utilisateur peut exprimer et ajuster ses exigences. Ces deux éléments représentent la politique d'approvisionnement de Ram et de bande passante respectivement envers les machines virtuelles qui sont hébergées dans des Hosts.

A.5.2 Allocation d'élément de traitement (Pe)

Pour allouer les ressources Pe, dans lequel l'unité de mesure est estimée par le Mips (millions d'instructions par seconde), deux étapes de politique d'approvisionnement devront être entreprises auparavant :

1. Attribution des Pes de l'hôte à ses VMs
2. Attribution des Pes des machines virtuelles aux cloudlets correspondantes.

Concernant la première politique d'approvisionnement qui se fait à travers la composante « VmScheduler » [23] responsable de l'approvisionnement de Pes aux machines virtuelles, où les hôtes jouent le rôle de fournisseurs envers les consommateurs dont leurs demandes sont exprimées en termes de VMs.

Après que chaque VM ait reçu sa part de ressource, il devient à son tour un fournisseur envers les Cloudlets qui lui sont assignées. La composante qui est en charge de l'attribution des Pes aux Cloudlets au sein du VM est appelée « CloudletScheduler » [23].

De plus, le système CloudSim propose pour ces deux allocations, deux politiques différentes : les politiques d'espaces partagé et temps partagé. Au niveau de l'hôte, ces deux politiques sont prisent en charge respectivement par les deux composantes VmSchedulerTimeShared et VmSchedulerSpaceShared du système CloudSim, et au niveau de la machine virtuelle, ils sont implémentées par CloudletSchedulerTimeShared et CloudletSchedulerSpaceShared [23]. Ces deux politiques sont décrites ci-dessous.

a Politique d'espace partagé :

La politique d'espace partagé est caractérisée par le fait que l'allocation de ressources se fasse de façon exclusive aux entités qui les ont réservées. En d'autres termes, dès que les demandes excèdent les ressources, le fournisseur fournira ces ressources aux premiers demandeurs et met les autres dans une file d'attente sur laquelle chaque consommateur attendra son tour. Chaque fois, qu'un consommateur termine sa tâche, il va libérer des ressources qui seront disponibles aux consommateurs misent en attente. Le temps de complétion estimé d'une tâche p avec un nombre d'instructions exprimé par rl , est noté par $eft(p)$ et est défini comme suit :

$$eft(p) = est + \frac{rl}{capacity \times cores(p)}, \quad (64)[23]$$

Le temps de début de p est noté par $est(p)$ est juste un temps estimé puisque la tâche obtiendra son tour juste après que l'utilisateur de cette ressource n'a plus besoin d'elle. Dans cette politique, la complétion de la tâche dépend sur le nombre de Pes requis " $cores(p)$ " par p , et sur " $la\ capacité$ " de l'hôte laquelle est exprimée par le nombre moyen de Mips par Pe qui est contenu dans ce dernier. Plus formellement, si nous supposons que l'hôte a np Pes, " $la\ capacité$ " de l'hôte est définie par la formule suivante :

$$capacity = \sum_{i=1}^{np} \frac{cap(i)}{np}, \quad (65) [23]$$

Où « $cap(i)$ » est la puissance de traitement en Mips du Pe de l'hôte.

b Politique du temps partagé :

Il y a une autre politique qui est plus flexible, la politique dite de temps partagé. Nous utilisons à cet effet un ordonnanceur dont le rôle est d'allouer pour un certain temps l'usage des ressources à une certaine entité, et mettre les autres sur une file d'attente pendant un certain temps. Après que le temps d'utilisation a été écoulé par l'entité en question, cette dernière sera à son tour mise en file d'attente et on choisira une autre qui était en file d'attente pour prendre sa place. En appliquant cette politique, toutes les entités seront exécutées pratiquement au même moment, ce qui est très utile dans le cas où la tâche a besoin que toutes les ressources doivent être disponibles en même temps. Étant donné que toutes les tâches seront exécutées presque simultanément, l'heure de début sera juste l'heure actuelle de la simulation, noté ct , et le temps de complétion estimé est calculé comme suit :

$$eft(p) = ct + \frac{rl}{capacity \times cores(p)}, \quad (66)[23]$$

Puisque l'exécution se déroule en mode multitâche, la capacité de l'hôte sera exprimée par la formule suivante :

$$capacity = \frac{\sum_{i=1}^{np} cap(i)}{\max(\sum_{j=1}^{cloudlets} cores(j), np)}, \quad (67) [23]$$

Notez que cette politique sera utilisée lorsque les *Pes* requis par les cloudlets "*cores(j)*" dépassent le nombre de *np*, sinon, la politique se révélera être une politique d'espace partagé ayant un liste d'attente.

c Visualisation de la politique temps et espace partagé :

Afin de visualiser les effets de la politique temps-partagé et espace-partagé sur l'exécution d'une unité de tâche, nous avons pris comme exemple le scénario suivant extrait de [23] :

Un hôte disposant de deux cœurs de processeurs reçoit une demande pour l'hébergement de deux VMs, tel que chacune requière deux cœurs et planifie d'héberger quatre tâches tel que chaque tâche demande un cœur. Plus spécifiquement, les tâches t1, t2, t3 et t4 seront prisent en charge par VM1, tandis que t5, t6, t7 et t8 seront prisent en charge par VM2. Pour récapituler, il y a deux VMs dans lesquels leurs exigences s'élèveront à huit cœurs pour un hôte ne disposant que de deux seulement.

Note :

Le nombre de cœurs de processeurs requis par une seule VM ne doit pas dépasser ceux de l'hôte, même si la somme de leurs exigences est supérieure à ce dernier, et cela est également vrai pour les Cloudlets par rapport à leurs VMs.

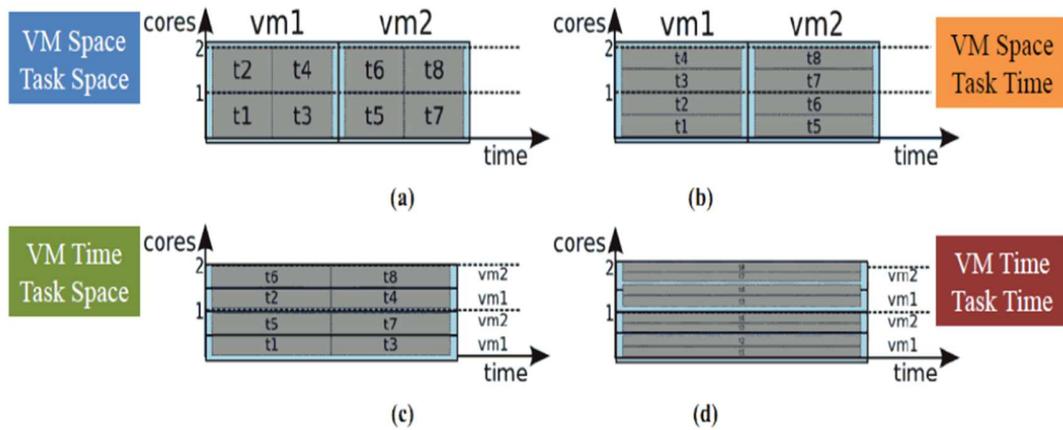


Figure 40 Les différents effets des politiques d'approvisionnements sur l'exécution de la tâche [23]

A.6 Cas d'utilisation

Ce cas d'utilisation implique un certain nombre d'acheteurs et un certain nombre de vendeurs engagés dans un marché électronique pour la négociation de ressources dans un environnement de cloud computing.

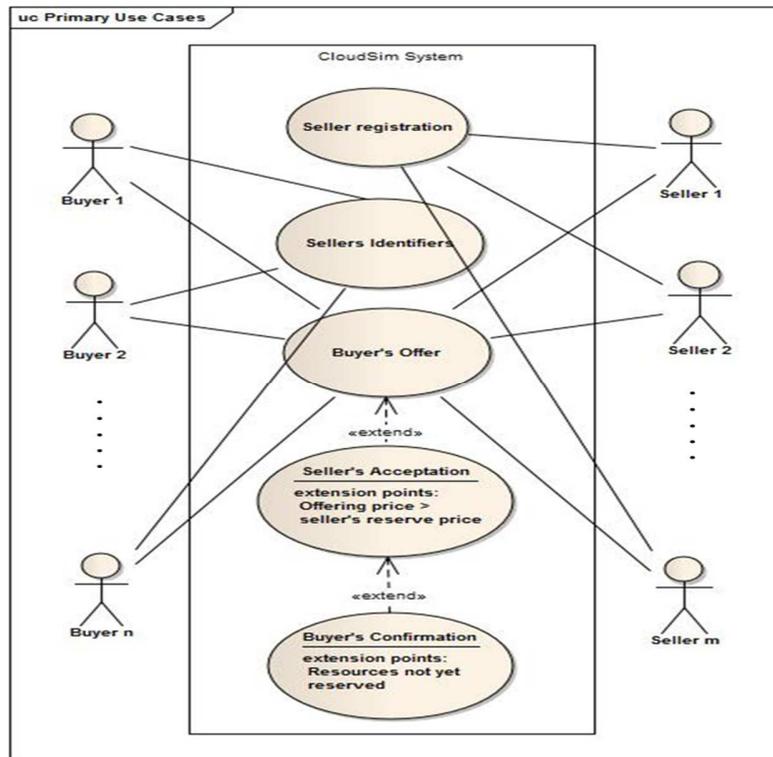


Figure 41 Diagramme de cas d'utilisation pour la négociation de ressources en utilisant CloudSim

Comme il est indiqué dans la Figure 41 ci-dessus, le système CloudSim permet aux vendeurs d'enregistrer leurs identités afin qu'ils soient joignables par n'importe quelle entité intéressée à marchander avec cette dernière. Les acheteurs peuvent effectivement consulter les identités des vendeurs pour choisir leurs partenaires commerciales appropriées. De même, les vendeurs connaissent l'identité de leurs acheteurs potentiels puisque ces derniers doivent la dévoiler dans chaque proposition formulée.