Master 2 Recherche Informatique mention Intelligence Artificielle

IRIT, Université Paul Sabatier
118, route de Narbonne
31062 Toulouse CEDEX

Un modèle formel pour raisonner sur des structures de coalitions

Par

Grégory Bonnet

Soutenu le 20 Juin 2005

Responsable de stage : Leila Amgoud

Résumé

Les agents autonomes travaillant dans des environnements multi-agents doivent coopérer afin de réaliser des tâches. Généralement, un agent ne peut pas exécuter seul une tâche et a besoin de l'aide des autres agents. Une des solutions à ce problème est de rechercher des groupes d'agents qui peuvent exécuter les tâches désirées plus efficacement. Différents algorithmes d'allocation de tâches via la formation de coalitions ont été proposés où chaque agent présente en vu d'une négociation différentes solutions appelées structures de coalitions. Ce mémoire fournit un cadre formel unifié pour raisonner sur les structures de coalitions. Inspiré du travail en théorie d'argumentation, en particulier du système de Dung, ce cadre prend en entrée un *ensemble de coalitions* dont les structures sont abstraites et fournit en sortie quatre types de structures de coalitions. Nous proposons également une théorie de la preuve dont le but est de déterminer si une coalition donnée sera dans une structure ou non pour l'agent sans avoir à calculer la structure entière. Deux instanciations de notre cadre sont données et leurs propriétés étudiées.

Table des matières

Chapitre 1

Introduction

Plusieurs applications logicielles ont été conçues et implémentées en utilisant la notion d'*agent autonome*. Ces applications varient du commerce électronique à de larges applications industrielles. Dans tous ces cas disparates, la notion d'autonomie est employée pour dénoter le fait que l'agent a la capacité de décider par lui-même des buts qu'il devra atteindre et de la manière dont ces buts seront réalisés. Au sein de ces systèmes, chaque agent œuvre donc pour la satisfaction de ses buts, ou encore pour maximiser son profit.

Cependant dans certaines situations, un agent peut ne pas pouvoir réaliser un but ou une tâche donnée seul, et décide donc de s'associer avec un ou plusieurs autres agents. Ainsi, ce groupe d'agents a formé une coalition temporaire pour réaliser la tâche en question.

Plusieurs chercheurs se sont penchés sur ce problème de formation des coalitions et cela dans des optiques différentes. Les premiers travaux ont été fait dans le cadre de la théorie des jeux [11]. Ensuite, afin de modéliser la notion de coordination dans les systèmes multi-agents, d'autres travaux [2, 5, 12, 14, 15, 17] ont été développés. L'idée de base est d'associer des tâches à des groupes d'agents. Les agents d'un groupe donné ont pour but soit de maximiser leur propre gain et on parlera alors d'*agents comp étitifs*, soit de maximiser le gain global du système, et on parlera d'*agents coop ératifs*.

Au vu de ces travaux et de ce que font remarquer Tohmé et Sandholm dans [17], la formation des coalitions est définie comme un processus en trois étapes :

1. Générer des structures de coalitions, c'est-à-dire l'ensemble de toutes les coalitions pouvant réaliser les tâches considérées. Par exemple, supposons que le système contient trois agents a_1, a_2, a_3 et que les tâches à réaliser sont t_1 et t_2. Supposons aussi que les deux agents a_1 et a_2 peuvent réaliser t_1 tandis que a_1 et a_3 peuvent réaliser t_2. Ainsi, l'ensemble $\{\{a_1,a_2\},\{a_1,a_3\}\}$ est une structure de coalition.
2. Choisir la structure qui sera retenue par tous les agents. Les structures générées à la 1ère étape sont différentes pour chaque agent en fonction de ses préférences. Il faut donc que ces agents discutent et négocient une unique solution.

3. Partager les gains obtenus par chaque coalition de la structure choisie entre les agents qui la compose.

La première phase consiste donc en la construction de l'ensemble des ensembles de coalitions possibles permettant une partition de l'ensemble des agents du système. La seconde, quant à elle, détermine quel ensemble de coalitions est optimal pour le problème donné et le sélectionne. Cette phase requière généralement une négociation entre les agents étant donné qu'ils peuvent ne pas avoir les mêmes préférences. Les structures que les agents génèrent peuvent aussi être différentes et, dans ce cas, il doivent se mettre d'accord sur celle qui sera adoptée. Quant à la dernière phase, il s'agit de répartir équitablement les gains de chaque coalition entre ses membres. Notons que lorsque les agents sont coopératifs, le partage des gains est inutile car les agents ne s'intéressent pas à leur gain individuel. Ainsi, le processus se réduit aux deux premières étapes uniquement.

Dans la littérature consacrée à la formation des coalitions, deux catégories de travaux existent :

- Quelques travaux sur la formation des structures de coalitions ($1^{\text{ère}}$ étape).
- Des travaux s'intéressant exclusivement au partage des gains ($3^{\text{ème}}$ étape).

La plupart des travaux en théorie des jeux sur la formation des coalitions se concentrent sur le partage des gains en supposant qu'une structure de coalition a été formée. La division des gains permet de s'assurer que cette structure est stable. Cette notion de stabilité se fonde sur le fait qu'aucun agent du système ne sera tenté de quitter la structure une fois qu'elle aura été calculée. Des schémas de transferts ont aussi été proposés pour transformer une structure instable en structure stable. Cependant, ceci suppose qu'une structure adéquate a été calculée.

Dans ce mémoire, nous nous intéressons à la première étape du processus de formation des coalitions, à savoir la génération des structures de coalitions. En effet, celle-ci se trouve en amont de tout le processus. De ce fait, si la structure générée est mauvaise ou n'est pas optimale, les gains à partager ne seront pas optimaux eux non plus. D'autre part, générer ces structures est un problème d'une complexité temporelle exponentielle. Or dans les systèmes multi-agents, le temps de calcul et de communication est coûteux. Pouvoir générer aisement de *bonnes* structures de coalitions permettrait donc d'éviter des traitements inutiles et de diminuer la complexité globale du processus.

Le peu de travaux qui se sont intéressés à la génération des structures de coalitions sont guidés par des applications. De ce fait, les structures générées dépendent largement de l'application considérée. Dans certaines par exemple, on peut exiger que les tâches soient indépendantes. Dans d'autres, on peut exiger qu'un agent ne devrait appartenir qu'à une seule coalition en même temps.

Différents algorithmes d'allocation de tâches via la formation de coalitions ont étés proposés [2, 3, 6, 8, 10, 9, 12, 13, 14, 15], et par conséquent ce sont des algorithmes qui générent des structures de coalitions. Chacun d'eux essaie de résoudre un problème particulier avec des contraintes particulières.

Inspiré du travail en théorie de l'argumentation, en particulier du puissant système d'argumentation développé par Dung dans [7], nous proposons un cadre formel *g énéral* et *unifié* pour produire des structures de coalitions. Ce cadre est défini en termes d'*ensembles de coalitions* considérées comme entités abstraites, de rapports de *conflits* entre ces coalitions et de relations de *préférence* entre les coalitions. Trois *sémantiques* pour ces structures de coalitions sont données : la sémantique de *base* qui renvoie une structure unique de coalitions, une sémantique *stable* et une sémantique *pr éférée* qui sont deux améliorations différentes de celle de base. Ces deux dernières sémantiques peuvent retourner plusieurs structures de coalitions à la fois. Chacune d'elle représente une façon de réaliser les différentes tâches par les agents. Nous montrons que ces sémantiques, définies à l'origine dans un cadre d'argumentation, ne sont pas toujours satisfaisantes dans notre contexte qu'est la formation des coalitions pour l'allocation de tâches. En particulier, nous montrons que ces sémantiques ne retournent pas toujours la solution optimale. Pour pallier cette limite, nous proposons une quatrième sémantique basée uniquement sur la notion de *sans-conflit*.

Nous proposons également une théorie de la preuve dans le cas de la sémantique de base. L'idée est la suivante : au lieu de calculer toute la structure de coalitions afin de savoir si une coalition donnée en fait partie ou non, nous pouvons vérifier directement s'il s'agit d'un membre de la structure ou pas. Cela permet aux agents de vérifier de façon dynamique la qualité d'une coalition. Ce travail est de grande importance, puisqu'il permet à des agents de raisonner sur les coalitions, et réduit au minimum la négociation entre eux dans la deuxième étape du processus de formation de coalition. Par ailleurs, ce cadre est assez général pour rendre compte des différentes propositions faites dans la littérature.

Nous proposons deux instanciations de notre cadre. La première considère le cas d'agents coopératifs, c'est-à-dire visant à maximiser le gain global du système, et cela dans le cas où les coalitions sont disjointes où un agent ne peut appartenir qu'à une seule coalition à la fois. Dans la seconde instanciation, nous considérons toujours des agents coopératifs, par contre nous nous intéressons au cas des coalitions recouvrantes où un agent peut participer à plus d'une coalition.

Ce mémoire est organisé comme suit. Le chapitre 2 présente un rappel sur la théorie d'argumentation. Le chapitre 3 introduit le cadre abstrait suggéré et fournit une théorie de la preuve vérifiant si une coalition donnée sera dans la structure de coalitions. Le chapitre 4 présente les deux instantiations de notre cadre. Le chapitre 5 présente brièvement les travaux existants dans la littérature sur la formation des coalitions. Le chapitre 6 est consacré à quelques remarques et perspectives de conclusion.

Chapitre 2

Introduction à l'argumentation

L'argumentation est une approche du raisonnement révisable. En logique classique, des valeurs de vérité contradictoires pour un même fait donné mènent à une indécidabilité. L'argumentation tente de résoudre ce problème en n'évaluant non pas un fait par sa valeur de vérité mais sur les raisons de sa vérité.

L'argumentation est un processus dynamique basé sur la construction d'arguments et de contre-arguments, l'évaluation de ces arguments et enfin la conclusion.

Un argument A *supporte* une conclusion lorsqu'il s'exprime en sa faveur. Tant qu'aucun autre argument B ne vient le contredire, sa conclusion est dite justifiée. Dans [4], le système abstrait de Dung a été étendu en prenant en compte la *force intrinsèque* des arguments. Cette dernière permet de comparer les arguments deux à deux. Ainsi, un système d'argumentation est défini par quatre éléments essentiels :
- Une notion d'argument.
- Une relation de préférence entre les arguments.
- Une relation de contrariété entre les arguments.
- Une notion d'acceptabilité des arguments.

Définition 1 *Un* système d'argumentation *est un triplet* $<\mathcal{A},\mathcal{R},\succeq>$ *où* $\mathcal{A}$ *est un ensemble d'arguments,* $\mathcal{R}$ *une relation binaire représentant la contrariété et* $\succeq$ *une relation de préférence entre arguments. Ainsi* $A\ \mathcal{R}\ B$ *signifie que l'argument* A *contrarie l'argument* B *et* $A \succeq B$ *signifie que l'argument* A *est préféré à l'argument* B *selon* $\succeq$.

La notion d'argument

La notion d'argument est au cœur du processus d'argumentation. Un argument peut être vu comme une raison de croire en une donnée, ou une raison de faire un choix donné, ou encore une raison de se comporter d'une certaine manière. En effet, dans un cadre de négociation, les agents peuvent échanger des menaces. Ces dernières sont considérées comme des arguments expliquant pourquoi l'agent qui les reçoit doit agir d'une certaine manière.

La relation de préférence

Au sein d'un même système d'argumentation, les différents arguments peuvent ne pas avoir la même *force*. Par exemple, un argument basé sur des informations certaines est meilleur qu'un autre argument basé, lui, sur des informations incertaines. Si on en vient à les comparer, le premier argument sera préféré au second. Ainsi, l'introduction de la notion de préférence permet de faire face à l'incomplétude et l'incertitude des croyances dans le cas des bases de connaissances.

D'une manière générale, les préférences sont déterminées par des critères d'évaluation qui dépendent de l'application considérée, que ce soit la fiabilité de l'information, une fonction d'utilité ou une mesure de probabilité. Cette préférence s'exprime sous la forme d'un pré-ordre (partiel ou total) sur l'ensemble des arguments du système.

La relation de contrariété

Un argument n'est pas une preuve car il ne peut pas toujours justifier sa conclusion, dans le sens où d'autres arguments peuvent venir le contrarier. En fait, la relation de contrariété est très vaste car représente toutes les formes de conflits qui peuvent exister entre les arguments. La notion de contrariété est donc fonction de la sémantique donnée aux arguments.

Dans le cadre de bases de connaissances par exemple, on trouve la réfutation et l'attaque. La réfutation consiste à nier la conclusion d'un argument en présentant un autre argument avec une conclusion contraire, et l'attaque consiste à nier un élément ou une prémisse de l'argument en présentant un argument avec une conclusion contraire à cette prémisse. On peut remarquer que l'attaque n'affecte pas directement la conclusion de l'argument et, donc, qu'un argument attaqué n'est pas nécessairement réfuté.

Définition 2 (Contrarier) *Soient A, $B \in \mathcal{A}$. A* contrarie *B si et seulement si $A \mathcal{R} B$ et $A \succeq B$.*

La notion d'acceptabilité

Une fois les notions précédentes définies, il doit être possible de conclure, à savoir déterminer le statut des arguments : lesquels seront jugés acceptables et lesquels ne le seront pas.

Une conclusion dépend de la qualité des arguments la supportant. Ainsi, une conclusion supportée par un argument non contrarié est justifiée jusqu'à preuve du contraire, c'est-à-dire jusqu'à l'arrivée d'un contrariant. Les contrariétés se répondent entre elles en une chaîne. Par exemple, si A contrarie B qui contrarie C, C sera jugé acceptable (donc justifié) car B aura été jugé injustifié par l'action de A.

L'acceptabilité pour un agent repose sur le principe de *défense*. L'ensemble des arguments qu'un agent rationnel peut accepter doit être défendu contre toute contrariété. Cela signifie que si un argument de l'agent est contrarié par un argument B, l'agent doit produire un nouvel argument contrariant cet argument B.

Dung a défini différentes sémantiques pour l'acceptabilité des arguments en se basant sur la notion d'ensemble d'arguments sans-conflits. Nous les rappelons dans cette section.

Définition 3 (Défense / Sans-conflit) *Soient $\mathcal{A}$ l'ensemble des arguments d'un système d'argumentation et S un sous-ensemble de $\mathcal{A}$.*

- S défend *un argument A si et seulement si chaque argument qui contrarie A est contrarié par un argument de S.*
- S *est* sans conflit *si et seulement si* $\nexists\ A_i, A_j \in S$ *tels que* A_i *contrarie* A_j.

Définition 4 (Sémantiques d'acceptabilité) *Soient S un ensemble sans-conflit d'arguments,*
$\mathcal{F} : 2^{\mathcal{A}} \longmapsto 2^{\mathcal{A}}$ *une fonction telle que* $\mathcal{F}(S) = \{A|S \text{ défend } A\}$ *et*
$\mathbf{G} : 2^{\mathcal{A}} \longmapsto 2^{\mathcal{A}}$ *une fonction telle que* $\mathbf{G}(S) = \{A|A \text{ n'est pas contrarié par un argument de } S\}$.

- S *est* admissible *si et seulement si* $S \subseteq \mathcal{F}(S)$.
- S *est une* extension complète *si et seulement si* $S = \mathcal{F}(S)$.
- S *est une* extension préférée *si et seulement si S est une extension complète maximale pour l'inclusion.*
- S *est une* extension stable *si et seulement si* $S = \mathbf{G}(S)$.
- S *est une* extension de base *si et seulement S est la plus petite des extensions complètes pour l'inclusion.*

L'extension complète se base sur le fait qu'un agent rationnel défend les arguments qu'il accepte et il accepte tous les arguments qu'il peut défendre. Le principe des extensions préférées est qu'un agent rationnel peut défendre tous les arguments qu'il accepte. L'extension stable est un ensemble dont tout argument qui n'en fait pas partie est contrarié par un argument de cet ensemble. L'extension de base, elle, est une vision plus prudente de l'acceptabilité car on peut remarquer qu'il n'y en a qu'une seule : elle contient l'ensemble des arguments qui ne sont pas contrariés ainsi que tous les arguments défendus directement ou indirectement par cet ensemble.

Chapitre 3

Un modèle formel pour le raisonnement sur les structures de coalitions

3.1 Les concepts de base

Le problème d'allocation de tâches par formation de coalitions peut être défini comme un ensemble fini $\mathcal{N}$ d'agents devant réaliser un ensemble fini $\mathcal{T}$ de tâches. Chaque agent vise à maximiser sa propre satisfaction ou bien la satisfaction globale du système multi-agents dont il est membre. Dans la littérature consacrée à la formation de coalitions, chaque agent est supposé être équipé d'une fonction qui renvoie son degré de satisfaction pour chaque coalition.

Un cadre générant des structures de coalitions est défini comme un triplet se composant d'un *ensemble de coalitions*, d'une relation binaire représentant la relation de *contrariété* entre ces coalitions, et pour finir d'une relation de *préférence* entre les coalitions. Ainsi, la structure d'une coalition n'est pas connue. Elle peut être, par exemple, n'importe quel sous-ensemble de $\mathcal{N}$, ou un sous-ensemble de $\mathcal{N}$ qui réalise une tâche donnée. On notera $\mathcal{C}$ l'ensemble de toutes les coalitions possibles.

Au niveau des conflits, ceux-ci devraient rendre compte des contraintes imposées par le problème étudié. Par exemple, si l'application considérée impose qu'un agent appartient à une coalition unique, alors deux coalitions contenant au moins un agent commun sont en conflit.

Classiquement, la théorie des jeux coopératifs dote les coalitions d'une utilité globale appelée *valeur de coalition* à partir d'une *fonction caratéristique* car les agents sont en mesure d'*évaluer* chaque coalition. Par exemple, une coalition peut avoir un *coût* et un *gain*. En ce cas, on peut imaginer la valeur d'une coalition comme étant son gain moins son coût. Les valeurs des coalitions permettent de les comparer. Généralement, un agent ne rejoint une coalition que si et seulement si il en tire au moins autant de bénéfice que s'il était resté seul et un agent est bénéficiaire s'il peut remplir ses buts ou recevoir un gain qui compense sa perte de ressource ou le non-accomplissement de ses objectifs. Ceci est appelé *rationalité individuelle*.

Dans ce qui suit, nous supposons que les valeurs des coalitions sont des nombres réels.

Définition 5 (Valeur de coalition) *La* valeur *d'une coalition $C \in \mathcal{C}$ est donnée par une fonction caractéristique :*

$$Valeur : \mathcal{C} \longmapsto \mathbb{R}$$

On note donc $Valeur(C)$ la valeur *de la coalition C.*

Définition 6 (Préférence entre les coalitions) *Soient deux coalitions C_1 et $C_2 \in \mathcal{C}$. C_1 est* préférée *à C_2, noté $C_1 \succeq C_2$, si et seulement si $Valeur(C_1) > Valeur(C_2)$.*

Suivant l'application considérée, la fonction $Valeur$ peut satisfaire certaines propriétés, comme la superadditivité ou la subadditivité.

Définition 7 (Coalitions disjointe / recouvrantes) *Deux coalitions sont dites* disjointes *si et seulement si elles n'ont pas d'agent en commun, sinon le deux coalitions sont dites* recouvrantes*.*

Définition 8 (Superadditivité) *Un système est* superadditif *si et seulement si $\forall$ C_1, $C_2 \in \mathcal{C}$ telles que C_1 et C_2 sont disjointes, la coalition conjointe $C_3 = C_1 \oplus C_2$ possède une valeur $Valeur(C_3) \geq Valeur(C_1) + Valeur(C_2)$. $C_1 \oplus C_2$ dénote la coalition contenant tous les agents de C_1 et tous les agents de C_2.*

Définition 9 (Subadditivité) *Un système est* subadditif *si $\forall$ C_1, $C_2 \in \mathcal{C}$ telles que C_1 et C_2 sont disjointes, la coalition conjointe $C_3 = C_1 \oplus C_2$ possède une valeur $Valeur(C_3) \leq Valeur(C_1) + Valeur(C_2)$.*

Les problèmes de formation de coalitions dans ces deux types de systèmes sont triviaux pour la génération des structures de coalitions. En effet, les agents d'un système superadditif préfèrent les coalitions qui sont composées de l'ensemble des agents du système, appelée la *grande coalition*, à toutes les autres. Les agents d'un système subadditif préfèrent les coalitions qui sont uniquement composées d'eux-même, des *coalitions singletons*, à toutes les autres. Cependant, rares sont les systèmes qui satisfont totalement ces propriétés. C'est pourquoi la recherche en formation de coalitions s'intéressent essentiellement à des systèmes non superadditifs et non subadditifs.

3.2 Le cadre général

Ainsi, le problème de formation de coalitions peut alors être représenté comme un processus en cinq étapes :

1. Construction des coalitions.

Argumentation	Formation des coalitions
Arguments	Coalitions
Relation de contrariété	Conflits entre coalitions
Relation de préférence	Relation de préférence basée sur les valeurs de coalitions
Extensions d'arguments	Structures de coalitions

FIG. 3.1 – Analogie entre argumentation et formation des coalitions

2. Définition des relations de contrariété et de préférence entre ces coalitions.
3. Définition des structures de coalitions.
4. Choisir la structure qui sera retenue par tous les agents.
5. Partager les gains.

Les trois premières étapes servent à générer des structures de coalitions pour chaque agent.

Au vu de ce qu'est un problème de formation de coalitions, on peut remarquer l'*analogie* avec l'argumentation. Une coalition joue le rôle d'un argument, les contraintes du problème étudié définissent la relation de contrariété et les valeurs des coalitions servent à définir la relation de préférence. Ainsi donc, on peut utiliser les sémantiques d'acceptabilité de Dung pour calculer et raisonner sur les structures de coalitions.

Définition 10 (Cadre formel) *Un* cadre de génération de structures de coalitions, *noté CGS, est un triplet* $<\mathcal{C}, \mathcal{R}, \succeq>$ *où* $\mathcal{C}$ *est un ensemble de coalitions,* $\mathcal{R}$ *une relation binaire représentant une relation de contrariété entre coalitions,* $\mathcal{R} \subseteq \mathcal{C} \times \mathcal{C}$*, et* $\succeq$ *un pré-ordre (partiel ou total) sur* $\mathcal{C}$ *(*$\succeq \subseteq \mathcal{C} \times \mathcal{C}$*).*

Définition 11 *Soient* C_1, C_2 *deux coalitions de* $\mathcal{C}$*, et* $S \subseteq \mathcal{C}$*.*
- C_1 attaque[1] C_2 *si et seulement si* $C_1 \; \mathcal{R} \; C_2$ *et* $C_1 \succeq C_2$*.*
- C_1 disqualifie C_2 *si et seulement si* C_1 *attaque* C_2 *et non (*C_2 *attaque* C_1*)*

Définition 12 *Un cadre (CGS) est* fini *si et seulement si pour chaque coalition* C*, il existe un nombre fini de coalitions qui contrarient* C*.*

Notons que le CGS est toujours fini car les deux ensembles $\mathcal{N}$ et $\mathcal{T}$ le sont aussi.

Différentes définitions de la relation de contrariété ($\mathcal{R}$) et de la relation de préférence ($\succeq$) mènent à différents systèmes qui peuvent ne pas retourner les mêmes structures de coalitions. Calculons maintenant ces structures de coalitions en utilisant les différentes extensions d'arguments définies par Dung, et commençons d'abord par la sémantique basique.

Intuitivement, il est clair qu'une coalition qui n'est pas du tout contrariée appartiendra à la structure de coalitions. Dans toute la suite, l'ensemble $\mathcal{C}_{\mathcal{R}}$ sera l'ensemble de toutes les coalitions non-contrariées au sens de la relation $\mathcal{R}$.

[1] Notons que le terme *attaque* est utilisé en argumentation pour désigner le conflit entre deux arguments où la conclusion de l'un vient contradire une prémisse de l'autre.

Exemple 1 *Soit <$\mathcal{C}$, $\mathcal{R}$, $\succeq$> un CGS tel que $\mathcal{C}$ = $\{C_1, C_2, C_3, C_4, C_5\}$, $\mathcal{R}$ = {(C_3, C_4), (C_4, C_3), (C_1, C_5)} et $C_3 \succeq C_4$, alors $\mathcal{C}_\mathcal{R}$ = $\{C_1, C_2\}$.*

Cette sémantique n'est pas suffisante et est très restrictive. Nous la raffinons en acceptant des coalitions contrariées à condition qu'elles soient préférées à leurs contrariantes. L'idée ici est de favoriser des coalitions *fortes*. Si par exemple une coalition C_1 dont le gain est V_1 (en supposant que le gain d'une coalition est égal à la valeur de cette coalition) est en conflit avec une autre coalition C_2 dont le gain est V_2 avec $V_2 > V_1$, il serait plus judicieux de conserver la coalition avec le plus grand gain, soit C_2.

Définition 13 *Soient <$\mathcal{C}$, $\mathcal{R}$, $\succeq$> un CGS et C_1, C_2 deux coalitions de $\mathcal{C}$ telles que C_1 $\mathcal{R}$ C_2. C_2* se défend seule *contre C_1 si et seulement si $C_2 \succeq C_1$.*
Une coalition se défend seule si et seulement si elle est préférée selon $\succeq$ à chacune de ses contrariantes.
$\mathcal{C}_{\mathcal{R},\succeq}$ dénote l'ensemble des coalitions qui se défendent seules contre leurs contrariantes.

Notons que $\mathcal{C}_{\mathcal{R},\succeq}$ est exactement l'ensemble des coalitions qui ne sont pas attaquées (voir Définition 11).

Exemple 2 *Dans l'exemple 1, comme $C_3 \succeq C_4$ alors C_3 se défend seule contre C_4. Par conséquent, $\mathcal{C}_{\mathcal{R},\succeq}$ = $\{C_1, C_2, C_3\}$.*

L'ensemble $\mathcal{C}_{\mathcal{R},\succeq}$ contient aussi les coalitions qui ne sont pas contrariées (au sens de la relation $\mathcal{R}$).

Propriété 1 *Soit <$\mathcal{C}$, $\mathcal{R}$, $\succeq$> un CGS. $\mathcal{C}_\mathcal{R} \subseteq \mathcal{C}_{\mathcal{R},\succeq}$.*

L'ensemble $\mathcal{C}_{\mathcal{R},\succeq}$ est également trop restreint car il rejette des coalitions qui peuvent apparaître «bonnes ». Intuitivement, si une coalition C_1 n'est pas autant préférée que son contrariant C_2, elle est alors affaiblie. Mais le contrariant C_2 lui-même peut être affaibli par une autre coalition C_3 qui le contrarie et lui est préférée. Dans ce cas, nous voudrions accepter C_1 parce qu'elle est défendue par C_3. Ceci correspond à la notion de défense conjointe utilisée en théorie d'argumentation.

Définition 14 *Soit $S \subseteq \mathcal{C}$. Une coalition C_1 est* défendue *par S si et seulement si $\forall$ C_2 $\in \mathcal{C}$, si C_2 attaque C_1 alors $\exists$ $C_3 \in S$ telle que C_3 attaque C_2.*

La structure basique de coalitions est alors caractérisée par une fonction *monotone* $\mathcal{F}$ qui retourne pour chaque ensemble de coalitions l'ensemble de toutes les coalitions qui sont défendues par cet ensemble.

Définition 15 *Soit $S \subseteq \mathcal{C}$. $\mathcal{F}(S)$ = $\{C \in \mathcal{C} \mid C$ est défendue par $S\}$.*

Puisque la fonction $\mathcal{F}$ est monotone, la structure basique de coalitions est définie en tant que son plus petit point fixe. Par ailleurs puisque le cadre CGS est fini, la fonction $\mathcal{F}$ est *continue* et, dans ce cas, son plus petit point fixe peut être obtenu par application itérative de $\mathcal{F}$ à l'ensemble vide.

Définition 16 (Structure basique de coalitions) *Soit* $<\mathcal{C}, \mathcal{R}, \succeq>$ *un CGS fini. La* structure basique *de coalitions est :*

$$\underline{\mathcal{S}}_{\mathcal{R},\succeq} = \bigcup \mathcal{F}^{i>0}(\emptyset)$$

En appliquant la fonction caractéristique $\mathcal{F}$ à l'ensemble vide, nous obtenons exactement l'ensemble de coalitions se défendant seules contre leurs contrariants. Plus formellement :

$$\mathcal{F}(\emptyset) = \mathcal{C}_{\mathcal{R},\succeq}.$$

Ainsi,

$$\underline{\mathcal{S}}_{\mathcal{R},\succeq} = \bigcup \mathcal{F}^{i>0}(\emptyset) = \mathcal{C}_{\mathcal{R},\succeq} \cup [\bigcup \mathcal{F}^{i\geq 1}(\mathcal{C}_{\mathcal{R},\succeq})]$$

La structure de coalitions contient alors les coalitions qui se défendent seules contre leurs contrariants ($\mathcal{C}_{\mathcal{R},\succeq}$) ainsi que les coalitions qui sont défendues (directement ou indirectement) par les coalitions de $\mathcal{C}_{\mathcal{R},\succeq}$.

Définition 17 *Soient* $S \subseteq \mathcal{C}$ *et* $C_1, C_2, C_3 \in \mathcal{C}$. S défend strictement C_1 *si et seulement si pour tout* C_2 *tel que* C_2 *attaque* C_1*, il existe* $C_3 \in S$ *tel que* C_3 *disqualifie* C_2.

Théorème 1 $\forall\, C \in \underline{\mathcal{S}}_{\mathcal{R},\succeq}$, $\underline{\mathcal{S}}_{\mathcal{R},\succeq}$ *défend strictement* C.

Dans certains cas, l'ensemble $\underline{\mathcal{S}}_{\mathcal{R},\succeq}$ peut être vide. Illustrons-le sur l'exemple suivant.

Exemple 3 *Soit* $<\mathcal{C}, \mathcal{R}, \succeq>$ *un CGS tel que* $\mathcal{C} = \{C_1, C_2\}$, $\mathcal{R} = \{(C_1, C_2), (C_2, C_1)\}$, $C_1 \succeq C_2$ *et* $C_2 \succeq C_1$. *Dans ce cadre,* $\underline{\mathcal{S}}_{\mathcal{R},\succeq} = \emptyset$.

Dans l'exemple ci-dessus, aucune structure n'est retournée et par conséquent aucune coalition n'est formée. Cela n'est pas toujours souhaitable dans des applications multi-agents.

Supposons que dans l'exemple 3, les coalitions C_1 et C_2 permettent de réaliser respectivement les tâches t_1 et t_2. Supposons que l'application étudiée impose qu'un agent appartient à une seule coalition à la fois. Supposons qu'un agent a_i est à la fois dans C_1 et C_2. Selon la structure basique, $\underline{\mathcal{S}}_{\mathcal{R},\succeq} = \emptyset$ donc aucune des deux tâches ne sera réalisée. Or, rien n'empêche qu'au moins une des deux tâches soit réalisée. On pourra, par exemple, choisir la coalition possèdant la plus grande valeur.

Afin de pallier les limites de la structure basique, nous considérerons les sémantiques stables et préférées définies dans le cadre d'argumentation. À la différence de la sémantique ci-dessus qui renvoie seulement une seule structure de coalitions, ces nouvelles sémantiques peuvent générer plusieurs structures en même temps.

Avant de présenter ces sémantiques, commençons d'abord par adapter la notion de *sans-conflit* au contexte de la formation des coalitions :

Définition 18 (Sans-conflit) *Soit* $S \subseteq \mathcal{C}$. S *est* sans-conflit *si et seulement si* $\nexists\, C_1, C_2 \in S$ *telles que* C_1 *attaque* C_2.

Définition 19 (Structures stables) *Soient* $<\mathcal{C}, \mathcal{R}, \succeq>$ *un CGS, et* $S \subseteq \mathcal{C}$. S *est une* structure stable *si et seulement si*

1. S *est sans-conflit.*
2. S *attaque toute coalition qui n'est pas dans* S.

Notez qu'un cadre CGS peut posséder plusieurs structures stables. Ces structures stables correspondent aux différentes manières de réaliser l'ensemble des tâches.

Exemple 4 *Dans l'exemple 3, il y a deux structures stables* $S_1 = \{C_1\}$ *et* $S_2 = \{C_2\}$.

Définition 20 (Structures préférées) *Soient* $<\mathcal{C}, \mathcal{R}, \succeq>$ *un CGS et* $S \subseteq \mathcal{C}$. S *est une* structure préférée *si et seulement si*

1. S *est sans-conflit.*
2. S *défend tous ses éléments*
3. S *est maximal pour l'inclusion parmi les ensembles qui satisfont les deux conditions précédentes.*

Notez que chaque cadre CGS possède au moins une structure préférée.

Exemple 5 *Soit* $<\mathcal{C}, \mathcal{R}, \succeq>$ *un CGS tel que* $\mathcal{C} = \{C_1, C_2, C_3, C_4, C_5\}$, $\mathcal{R} = \{(C_2, C_1), (C_2, C_5), (C_1, C_5), (C_4, C_2), (C_3, C_2), (C_3, C_4), (C_4, C_3)\}$, $C_2 \succeq C_1$, $C_2 \succeq C_5$, $C_1 \succeq C_5$, $C_4 \succeq C_2$, $C_3 \succeq C_2$, $C_3 \succeq C_4$ *et* $C_4 \succeq C_3$.
Dans ce cadre $\underline{\mathcal{S}}_{\mathcal{R},\succeq} = \emptyset$ *alors qu'il y a deux structures préférées* $S_1 = \{C_1, C_3\}$ *et* $S_2 = \{C_1, C_4\}$.

Propriété 2 *Soit* $<\mathcal{C}, \mathcal{R}, \succeq>$ *un CGS :*

– *Chaque structure stable est aussi une structure préférée. Cependant, l'inverse n'est pas toujours vrai.*
– *La structure de coalitions* $\underline{\mathcal{S}}_{\mathcal{R},\succeq}$ *est incluse en chaque structure stable (resp. préférée).*

Soit $\{S_1 \ldots S_n\}$ l'ensemble de toutes les structures de coalitions selon une sémantique donnée. La valeur globale d'une structure de coalitions est la somme des valeurs des coalitions qu'elle contient.

Définition 21 (Valeur globale d'une structure de coalitions) *Soient* $<\mathcal{C}, \mathcal{R}, \succeq>$ *un CGS et* $\{S_1 \ldots S_n\}$ *ses structures de coalitions selon une sémantique donnée. La* valeur globale *de* S_i *est* $Valeur_g(S_i) = \sum_{C_j \in S_i} Valeur(C_j)$.

Il est à noter que les structures générées peuvent ne pas avoir les mêmes valeurs globales. Etant donné que le but pour des agents coopératifs est de trouver la solution qui maximise le gain du système parmi les structures générées, on retient celle qui a la plus grande valeur globale. On parle alors de solution optimale. Formellement :

Définition 22 (Solution optimale) *Soit* $<\mathcal{C},\mathcal{R},\succeq>$ *un CGS et* $\{S_1 \ldots S_n\}$ *ses structures de coalitions. La* solution optimale *est* $S^* \in \{S_1 \ldots S_n\}$ *telle que* $Valeur_g(S^*) \geq Valeur_g(S_i)$, $\forall\, S_i \neq S^*$.

3.3 Les propriétés du modèle

La première remarque importante concernant ce modèle est le fait qu'il ne maximise pas nécessairement le gain du système. Considérons l'exemple suivant :

Exemple 6 *Soit* $<\mathcal{C},\mathcal{R},\succeq>$ *un CGS avec* $\mathcal{C} = \{C_1,C_2,C_3\}$*,* $\mathcal{R} = \{(C_1,C_2),(C_1,C_3)\}$ *et* $C_1 \succeq C_2$*,* $C_1 \succeq C_3$*. Supposons que* $Valeur(C_1) = 40$*,* $Valeur(C_2) = 20$ *et* $Valeur(C_3) = 30$*.*
Il est clair que le système retourne une seule structure $\underline{S} = \{C_1\}$ *qui est à la fois basique, stable et préférée. La valeur globale de cette structure est exactement la valeur de* C_1*, soit* 40*. Or, on peut remarquer que les coalitions qui n'ont pas été retenues ont une valeur conjointe (*$Valeur(C_2) + Valeur(C_3) = 50$*) plus importante. De plus,* $\{C_2,C_3\}$ *est sans conflit.*

L'exemple ci-dessus montre bien que les sémantiques d'acceptabilité définies par Dung dans le cadre d'argumentation ne sont pas toujours bien adaptées dans le contexte de la formation des coalitions. Reprenons l'exemple où C_1 contrarie fortement C_2 et C_3. Si on était dans un contexte d'argumentation, c'est-à-dire C_1, C_2 et C_3 sont des arguments, notre but aurait été de chercher parmi ces trois arguments celui ou ceux qui justifient leurs conclusions. On ne pourra donc pas tolérer C_2 et C_3 puisqu'ils sont fortement contrariés par C_1. Cette forte contrariété les rend vulnérables.

Cependant, le but est différent dans le contexte de la formation des coalitions où on aimerait exécuter le maximum de tâches. Le besoin de réaliser les tâches est aussi accompagné du besoin de maximiser le gain du système. Ainsi dans l'exemple ci-dessus, même si C_2 et C_3 sont fortement contrariés par C_1 rien ne nous empêche de considérer $\{C_2,C_3\}$ comme solution du problème, c'est-à-dire comme la meilleure structure de coalitions, celle qui maximise le profit du système.

Afin de capturer ce résultat, nous considérerons une quatrième sémantique pour les structures de coalitions. Cette nouvelle sémantique consiste tout simplement à chercher les ensembles maximaux pour l'inclusion et sans conflit de coalitions.

Définition 23 (Les extensions) *Soit* $<\mathcal{C},\mathcal{R},\succeq>$ *un CGS. Un ensemble* $S \subseteq \mathcal{C}$ *est une* extension *(structure de coalitions) si et seulement si :*

1. $\nexists\, C_1, C_2 \in S$ *telles que* $C_1\ \mathcal{R}\ C_2$*.*
2. S *est maximal pour l'inclusion parmi les ensembles qui vérifient la propriété ci-dessus.*

Notons que dans cettte sémantique, la relation de préférence n'est pas prise en compte. En fait, cette dernière est utilisée pour classer les extensions obtenues et, par conséquence, pour déterminer la solution optimale (Définition 22). Cette dernière est alors maximale.

3.4 Théorie de la preuve

Jusqu'ici, nous avons fourni la sémantique d'une structure de coalitions en définissant l'ensemble des coalitions qu'elle doit contenir. Dans le cas de la sémantique basique, par exemple, une seule structure $\underline{\mathcal{S}}_{\mathcal{R},\succeq}$ est calculée. Cependant, dans la pratique nous n'avons pas besoin de calculer la totalité de l'ensemble $\underline{\mathcal{S}}_{\mathcal{R},\succeq}$ afin de connaître le statut d'une coalition donnée, c'est-à-dire si cette coalition sera formée ou pas. Dans cette section nous proposons une méthode de vérification du statut d'une coalition C, c'est-à-dire nous proposons une théorie de la preuve pour vérifier si C est dans $\underline{\mathcal{S}}_{\mathcal{R},\succeq}$ ou pas.

Dans ce but, nous sommes inspirés par le travail effectué dans [4] dans le contexte de la théorie d'argumentation.

L'idée fondamentale de cette théorie de la preuve est de parcourir la séquence $\mathcal{F}^1, \ldots, \mathcal{F}^n$ à l'envers. Considérons que C apparait pour la première fois dans $\mathcal{F}^n$. Nous partons par C, et puis pour n'importe quelle coalition B_i qui attaque C, nous trouvons une coalition C_i dans $\mathcal{F}^{n-1}$ qui défend C. Maintenant, en raison du Théorème 1, nous sommes seulement intéressés par les défenseurs stricts d'une coalition, et les défenseurs stricts de C élimineront B_i. Le même processus est répété pour chaque défenseur strict jusqu'à ce qu'il n'y ait plus aucun défenseur ou contrariant stricts.

Nous pouvons représenter ce processus en termes de jeu de dialogue entre deux joueurs P et O. P produit la coalition qui nous intéresse ainsi que ses défenseurs et O produit les contre-coalitions ou les contrariants.

Définition 24 *Un* dialogue *est une séquence non vide de coups,* $coup_i = (Joueur_i, Coal_i)$ *pour* $i \geq 0$ *tel que :*

1. $Joueur_i = P$ *si et seulement* i *est pair,* $Joueur_i = O$ *si et seulement si* i *est impair.*
2. $Joueur_0 = P$ *et* $Coal_0 = C$.
3. *Si* $Joueur_i = Joueur_j = P$ *et* $i \neq j$ *alors* $Coal_i \neq Coal_j$.
4. *Si* $Joueur_i = P$, $i > 1$, *alors* $Coal_i$ *disqualifie* $Coal_{i-1}$.
5. *Si* $Joueur_i = O$ *alors* $Coal_i$ *attaque* $Coal_{i-1}$.

Un arbre de dialogue *est un arbre fini dont chaque branche est un dialogue.*

Exemple 7 *Soit* $<\mathcal{C}, \mathcal{R}, \succeq>$ *un CGS tel que* $\mathcal{C} = \{a_0, a_{01}, a_{02}, a_{03}, a_{10}, a_{11}, a_{12}\}$, $\mathcal{R} = \{(a_{10}, a_0), (a_{01}, a_{10}), (a_{12}, a_{02}), (a_{02}, a_{10}), (a_{03}, a_{11}), (a_{11}, a_0)\}$. *Supposons que* $a_{03} \succeq a_{11} \succeq a_0$, $a_{01} \succeq a_{10} \succeq a_0$ *et* $a_{12} \succeq a_{02}$, $a_{02} \succeq a_{10}$. *Nous nous intéressons au statut de la coalition* a_0. *L'arbre de dialogue correspondant est représenté par la figure FIG. 3.2.*

L'arbre de dialogue peut être considéré comme un arbre ET/OU. Un nœud correspondant au joueur P est un nœud ET, et un nœud correspondant au joueur O est un nœud OU car une coalition est n'acceptable que si elle est défendue contre tous ses contrariants. Les fils d'un nœud contenant une coalition de P représentent des contrariants : ils doivent donc tous être contrariés. En revanche, les fils d'un nœud contenant

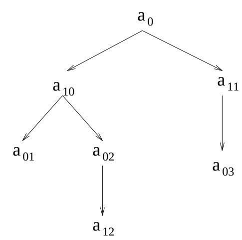

FIG. 3.2 – Un arbre de dialogue

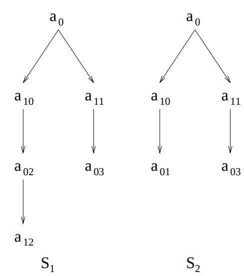

FIG. 3.3 – Sous-arbres candidats

une coalition de O représentent des défenseurs de P ainsi il suffit que l'un d'entre eux contrarie la coalition de O.

Définition 25 *Un joueur* gagne *un dialogue si et seulement il produit la dernière coalition du dialogue.*

Un joueur qui gagne un dialogue ne gagne pas nécessairement dans tous les sous-arbres de l'arbre de dialogue. Pour formaliser le gain d'un arbre de dialogue, le concept d'un sous-arbre solution est défini.

Définition 26 *Un* sous-arbre candidat *est un sous-arbre d'un arbre de dialogue contenant tous les fils de chaque nœud ET et exactement un fils de chaque nœud OU. Un* sous-arbre solution *est un sous-arbre candidat dont toutes les branches sont gagnées par P.*

Exemple 8 *Ainsi le dialogue représenté dans l'exemple 3 possède exactement deux sous-arbres candidats S_1 et S_2, (voir FIG. 3.3).*

Définition 27 P gagne *un dialogue si et seulement si l'arbre de dialogue correspondant possède un sous-arbre solution.*

Exemple 9 *Ainsi P gagne le dialogue présenté dans la figure FIG. 3.3 car S_2 est un sous-arbre solution.*

Définition 28 *Soit $C \in \mathcal{C}$. Une coalition C est* justifiée *si et seulement si il existe un arbre de dialogue dont la racine est C, et qui est gagné par le joueur P.*

Exemple 10 *Ainsi la coalition a_0 est justifiée parce que le joueur P a gagné l'arbre de dialogue.*

Le principal résultat de cette théorie de la preuve est :

Théorème 2 *Soit $<\mathcal{C}, \mathcal{R}, \succeq>$ un CGS.*

1. *$\forall\, C \in \mathcal{C}$, si C est justifiée alors chaque coalition de P appartenant au sous-arbre solution est dans $\underline{\mathcal{S}}_{\mathcal{R},\succeq}$, en particulier C.*
2. *$\forall\, C \in \underline{\mathcal{S}}_{\mathcal{R},\succeq}$, C est justifiée.*

En d'autres termes, le processus de dialogue construit toutes les coalitions acceptables, et seulement des coalitions acceptables. Il est donc robuste et complet.

Chapitre 4

Application du modèle général

4.1 Allocation de tâches pour coalitions disjointes

4.1.1 Position du problème

Afin d'illustrer notre modèle, considèrons le problème de la formation de coalitions décrit dans [14]. Ce problème est celui de l'allocation de tâches parmi des groupes d'agents autonomes et coopératifs. L'idée est qu'étant donné un ensemble $\mathcal{T}$ de tâches, le système doit satisfaire toutes les tâches, ou au moins en satisfaire le plus possible, tout en maximisant son profit global.

Dans ce travail, un système multi-agents est censé exécuter un *service*. Ce service exige plusieurs *critères* $<c_1, \ldots, c_r>$. Par exemple, pour un service de transport, les critères seront $poids$, $taille$ et $volume$.

Plusieurs *agents* $\mathcal{N} = \{a_1, \ldots, a_n\}$ sont intégrés dans ce système. Chaque agent a_i est supposé avoir un vecteur de *capacités* représentées par des valeurs réelles non négatives $B^i = <b_1^i, \ldots, b_r^i>$. Un élément b_j^i représente la capacité de l'agent a_i concernant le critère c_j. Dans le cas du transport, ceci signifie qu'un agent p possède un poids x, une taille y et un volume z transportable.

Le système possède également un ensemble $\mathcal{T} = \{t_1, \ldots, t_m\}$ de *tâches* à exécuter. Pour chaque tâche t, un vecteur $B^t = <b_1^t, \ldots, b_r^t>$ de capacités est associé. Un élément $b_k^{t_j}$ représente la quantité de c_k nécessaire pour la réalisation de la tâche t_j.

Définition 29 (Système multi-agents) *Un système multi-agents (SMA) est un triplet* $<\mathcal{S}, \mathcal{N}, \mathcal{T}>$ *tel que :*

- $\mathcal{S} = <c_1, \ldots, c_r>$.
- $\mathcal{N} = \{a_1, \ldots, a_n\}$ *est un ensemble d'*agents *tel que* $\forall\, a_i \in \mathcal{N}$ *on a :*
  1. *un vecteur* $B^i = <b_1^i, \ldots, b_r^i>$ *de ses capacités.*
  2. *une fonction* $Valeur$ *qui retourne la valeur d'une coalition donnée.*
- $\mathcal{T} = \{t_1, \ldots, t_m\}$ *est un ensemble de* tâches *à réaliser tel que :* $\forall\, t \in \mathcal{T}$ *on a un vecteur* $B^t = <b_1^t, \ldots, b_r^t>$ *de capacités nécessaires à sa réalisation.*

Notons que ces informations sont connues de tous les agents du système. Dans [14], les hypothèses suivantes sont posées :

– Les tâches sont indépendantes.
– Un agent ne peut pas appartenir à plus d'une coalition à la fois. Les coalitions formées avec une telle contrainte sont appelées *coalitions disjointes*.
– Une coalition ne peut réaliser qu'une unique tâche à la fois.

Posons formellement le problème :

Définition 30 (Problème) *Soit* $<\mathcal{S}, \mathcal{N}, \mathcal{T}>$ *un système multi-agents. On doit assigner à chaque tâche* $t_i \in \mathcal{T}$ *un groupe d'agent* $C_i \subseteq \mathcal{N}$ *tel que* C_i *réalise* t_i, $\bigcup C_i = \mathcal{N}$ *et* $\forall i \neq j$, $C_i \cap C_j = \emptyset$ *et que la valeur globale de la structure générée soit maximale.*

4.1.2 Instanciation du CGS

A partir des hypothèses posées par Shehory et Kraus dans [14], nous instancions dans cette section le CGS pour calculer les structures de coalitions de ce problème et cela, bien sûr, au niveau de chaque agent. Il nous faut donc définir ce qu'est une coalition, sa valeur et les contraintes sur les structures.

La notion de coalition

Une coalition est un groupe d'agents qui coopèrent afin de réaliser une tâche commune. En fait, une coalition devrait être *minimale* puisque dans cette application chaque coalition a un coût. Donc, plus la coalition est de grande taille, plus elle est coûteuse. Par ailleurs, un agent ne devrait pas faire partie d'une coalition s'il n'est pas utile et ne peut pas aider à la réalisation de la tâche. Mais avant de donner la définition formelle d'une coalition, définissons tout d'abord formellement ce qu'est réaliser une tâche.

Définition 31 (Tâche réalisable) *Soient* $<\mathcal{S}, \mathcal{N}, \mathcal{T}>$ *un système multi-agents,* $C \subseteq \mathcal{N}$ *et* $t \in T$.
Une tâche t *est* réalisable *par* C, *noté* $C \Vdash t$, *si et seulement si* $\forall\ 1 \leq j \leq r$, $\sum_{a_i \in C} b_j^i \geq b_j^t$.

La définition ci-dessus signifie qu'une tâche est réalisable par un groupe d'agents si les capacités des agents pris ensembles sont suffisantes à ce qui est exigé par la tâche. Nous sommes donc maintenant prêts à définir formellement une coalition.

Définition 32 (Coalition) *Soit* $<\mathcal{S}, \mathcal{N}, \mathcal{T}>$ *un système multi-agents. Une* coalition *est une paire* $<C, t>$ *telle que :*

1. $C \subseteq \mathcal{N}$
2. $t \in T$
3. $C \Vdash t$
4. C *est minimal pour l'inclusion parmi les ensembles qui satisfont les conditions précédentes.*

Nous appellerons C *le* support *de la coalition, et* t *sa* tâche. *Dans ce qui suit,* $\mathcal{C}_{disj}$ *dénotera l'ensemble de toutes les coalitions pouvant être générées à partir de* $<\mathcal{S}, \mathcal{N}, \mathcal{T}>$.

La force d'une coalition

Chaque agent est supposé être équipé d'une fonction $Valeur$ qui renvoie la valeur d'une coalition selon le point de vue de cet agent. La valeur d'une coalition peut être égale au gain obtenu de la coalition moins le coût de cette coalition. Cependant, la valeur peut être définie de différentes manières. Comme dans [14], nous supposons que cette valeur est indiquée et qu'elle est numérique.

Les valeurs des coalitions permettent de les comparer. En effet, la coalition avec une plus grande valeur est plus forte que celle avec une petite valeur. Formellement :

Définition 33 *Soient* $<C_1,t_1>$, $<C_2,t_2> \in \mathcal{C}_{disj}$. $<C_1,t_1>$ *est plus* profitable *que* $<C_2,t_2>$*, noté* $<C_1,t_1> \succeq <C_2,t_2>$*, si et seulement si* $Valeur(<C_1,t_1>)\ leq\ Valeur(<C_2,t_2>)$.

Conflits entre les coalitions

Les *structures de coalitions* devraient satisfaire les hypothèses déjà fixées en définissant le problème. La première contrainte est qu'un agent ne peut pas appartenir à plus d'une coalition en même temps. En effet, deux coalitions, définies selon la Définition 32 et contenant au moins un agent en commun, ne peuvent pas être ensemble dans la même strucutre de coalitions. De telles coalitions seraient conflictuelles. Ce genre de conflit s'appellera ici *Interférer*. Formellement :

Définition 34 (Coalitions interférentes) *Soient* $<C_1, t_1>$, $<C_2, t_2> \in \mathcal{C}_{disj}$.
$<C_1, t_1>$ interfère *avec* $<C_2, t_2>$ *si et seulement si* $C_1 \cap C_2 \neq \emptyset$.

La deuxième contrainte du problème étudié est qu'une même tâche ne peut pas être affectée à plus d'une coalition en même temps. Dans la structure de coalitions, il n'est donc pas possible que deux coalitions réalisent une tâche identique. Cette condition donne naissance à un autre type de conflit entre les coalitions. Dans ce qui suit, nous appellerons ce conflit *Concurrencer*. Formellement :

Définition 35 (Coalitions concurrentes) *Soient* $<C_1, t_1>$, $<C_2, t_2> \in \mathcal{C}_{disj}$.
$<C_1, t_1>$ concurrence $<C_2, t_2>$ *si et seulement si* $t_1 = t_2$.

Les deux relations ci-dessus peuvent être regroupées dans une définition unique de la contrariété comme suit :

Définition 36 (Contrarier) *Soient* $<C_1, t_1>$, $<C_2, t_2> \in \mathcal{C}_{disj}$.
$<C_1, t_1>$ contrarie $<C_2, t_2>$ *si et seulement si :*

- $<C_1, t_1>$ interfère *avec* $<C_2, t_2>$ *ou*
- $<C_1, t_1>$ concurrence$<C_2, t_2>$.

Notons que les trois relations ci-dessus sont symétriques.

Structures de coalitions

Une fois les notions de coalition et de contrariété définies, nous pouvons maintenant présenter le système qui sera employé pour générer des structures de coalitions.

Définition 37 *Un* cadre de génération de structures de coalitions *est un triplet* $<\mathcal{C}_{disj}$, $Contrarier$, $\succeq>$ *où* $\mathcal{C}_{disj}$ *est l'ensemble des coalitions pouvant être construites à partir du système d'agents* $<\mathcal{S}, \mathcal{N}, \mathcal{T}>$ *en utilisant la Définition 32,* $Contrarie$ *est la relation donnée par la Définition 36, et* $\succeq$ *est un pré-ordre (partiel ou total) sur* $\mathcal{C}_{disj}$ *suivant la Définition 33.*

La structure de coalitions de ce système selon la sémantique basique est :

$$\underline{\mathcal{S}}_{Contrarie,\succeq} = \bigcup \mathcal{F}^{i>0}(\emptyset)$$

$$= \mathcal{C}_{Contrarie,\succeq} \cup [\bigcup \mathcal{F}^{i\geq 1}(\mathcal{C}_{Contrarie,\succeq})]$$

Définissons deux fonctions : Supp et Tâche. La fonction Supp retourne, pour un ensemble donné de coalitions, l'ensemble de tous les agents impliqués dans ces coalitions. La fonction Tâche retourne, pour un ensemble donné de coalitions, l'ensemble des tâches réalisées par ces coalitions. Formellement :

Définition 38 *Soit* $<\mathcal{C}_{disj}, Contrarie, \succeq>$ *un CGS. Une structure de coalitions* S *est* complète *si et seulement si :*

1. *Supp(*S*)* $= \mathcal{N}$*, et*
2. *Tâche(*S*)* $= \mathcal{T}$.

Propriété 3 *La structure de coalitions* $\underline{\mathcal{S}}_{Contrarie,\succeq}$ *n'est pas toujours complète.*

Considérons l'exemple suivant :

Exemple 11 *Soit* $\mathcal{N} = \{a_1, a_2, a_3\}$ *et* $\mathcal{T} = \{t_1, t_2\}$*. Supposons que les deux coalitions suivantes sont construites :* $C_1 = <\{a_1, a_2\}, t_1>$ *et* $C_2 = <\{a_1, a_3\}, t_2>$*. Supposons également que* $C_1 \succeq C_2$*. La structure basique (extension, stable, préférée) de coalitions contient alors seulement la coalition* C_1*. Ainsi, seuls les agents* a_1 *et* a_2 *participeront à la réalisation d'une tâche. Par ailleurs, seule la tâche* t_1 *sera réalisée.*

Le résultat suivant peut être démontré :

Théorème 3 *Si les agents ont tous les mêmes valeurs pour les différentes coalitions, alors leurs cadres respectifs renverront tous la même structure de coalitions.*

Ce résultat est de grande importance puisqu'il montre qu'avec un tel cadre, plus de travail est effectué par les agents eux-mêmes, et par conséquent ceci peut réduire considérablement la communication qui est très coûteuse. Il n'y a donc pas besoin de l'étape de négociation.

4.1.3 L'algorithme de Shehory et Kraus

Shehory et Kraus associent à chaque coalition C_i, en plus de sa valeur, son coût qui en est l'inverse : $c(C_i) = \frac{1}{Valeur(C_i)}$. Il proposent alors un algorithme qui fournit une solution optimale au problème.

Dans un premier temps, les valeurs de coalitions sont calculées puis les agents décident de leurs coalitions préférées et les forment. Enfin, les bénéfices sont distribués entre les agents. Comme cet algorithme est très gourmand, les agents doivent chercher à minimiser leurs coûts de calcul en évitant les calculs et les communications inutiles. De la même façon, les coalitions de petite taille sont préférées à celles de grande taille. Ces heuristiques sont implémentées en définisant un entier k qui donne la taille de la plus grande coalition autorisée. De plus, un agent qui rejoint une coalition quitte le processus de formation pour ne laisser que des agents singletons.

1ère étape : calculer les valeurs de coalition de façon distribuée

Chaque agent a_i doit réaliser ce qui suit :

1. Calculer toutes les permutations incluant jusqu'à k agents dont a_i et les placer dans P_i, l'ensemble des coalitions potentielles pour a_i.
2. Tant que $P_i \neq a_i$ faire :
   (a) Contacter un agent a_j qui est membre d'une coalition de P_i.
   (b) Si c'est le premier contact avec a_j, s'informer sur ses compétences : c'est à dire retrouver B_j.
   (c) Promettre de calculer la valeur d'un sous-ensemble S_{ij} : un sous-ensemble de coalitions de P_i dont a_i et a_j sont membres.
   (d) Enlever S_{ij} de P_i et ajouter S_{ij} à la liste L_i de promesses.
   (e) Calculer les valeurs de coalitions des S_{ij} et contacter les agents dont a_i ne possède pas les compétences.
   (f) Pour chaque agent a_k qui contacte a_i, retirer de P_i l'ensemble S_{ki}.

A ce stade, chaque agent a_i possède une liste L_i de coalitions pour lesquelles il a promis de calculer la valeur ainsi que leurs valeurs préliminaires. De plus, il se doit maintenir une liste L_i^{cr} des coalitions pour lesquelles il doit calculer la valeur. Pour ceci, l'agent a_i doit faire pour chaque coalition C de L_i^{cr} :

1. Calculer le vecteur de compétences B_C tel que $B_C = \sum_{a_i \in C} B_i$.
2. Faire une liste E_C des gains espérés lorsque la coalition C réalise une tâche et pour ce faire, exécuter pour chaque $t_j \in T$:
   (a) Vérifier quelles compétences B_{t_j} sont nécessaires à la réalisation de t_j.
   (b) Comparer B_{t_j} à B_C pour trouver quelles tâches peuvent être réalisées par C.
   (c) Si t_j peut être réalisée, calculer le gain espéré net e_{t_j} qui est le gain espéré brut moins le coût de coordination et d'utilisation des compétences. Mettre e_{t_j} dans E_C.

3. Prendre la valeur maximale dans E_C qui devient la valeur de coalition V_C.
4. Calculer le coût de coalition qui est $c_C = \frac{1}{V_C}$.

Si un agent a_i a terminé sa part de calcul avant les autres, il peut contacter un agent a_j et le décharger d'une partie de son travail en relançant l'algorithme ci-dessus.

2$^{\text{ème}}$ étape : choisir les coalitions

Chaque agent possède désormais une liste de coalitions L_i ainsi que leurs valeurs. On note w_i le *poids de la coalition*, c'est-à-dire le ratio entre le coût de la coalition et sa taille tel que $w_i = \frac{c(C_i)}{|Ci|}$, et chaque agent a_i fait :

1. Sélectionner dans L_i la coalition C_j qui possède le plus petit w_j.
2. Annoncer publiquement la valeur de coalition w_j.
3. Choisir parmi toutes les annonces la coalition C_{min} qui a le poid w_{min} le plus faible ainsi que la tâche t_{min} qu'elle réalise.
4. Supprimer les membres de C_{min} de la liste des candidats pour une autre coalition.
5. Si $a_i \in C_{min}$ alors rejoindre les autres membres et former la coalition.
6. Supprimer L_i de toutes les coalitions possibles qui incluent les agents de C_{min}.
7. Supprimer de T la tâche t_{min}.
8. Assigner L_i^{cr} à les coalitions de L_i dont la valeur doit être recalculée.

Cette procédure est répétée jusqu'à ce qu'il n'y ai plus d'agents, c'est à dire qu'ils fassent tous partie d'une coalition, qu'il n'y ai plus de tâche à allouer ou que plus aucune coalition ne procure de bénéfice. Les valeurs de coalitions doivent être recalculées car elles dépendent de la configuration de coalition choisie : chacune est fonction des tâches à réaliser. Tout changement au sein de la configuration de coalition signifie qu'une tâche a été assignée et n'entre désormais plus en compte dans le calcul des valeurs de coalition.

4.2 Allocation de tâches pour coalitions recouvrantes

4.2.1 Position du problème

Shehory et Kraus proposent dans [15] une généralisation du modèle précédent. Ils reprennent le cadre formel en y adjoignant quelques modifications : suppression de l'hypothèse de disjonction entre les coalitions, permettant ainsi à un agent de se trouver dans plusieurs coalitions à la fois, et ajout d'un ordonnancement partiel sur les tâches. Reformulons donc le problème :

Définition 39 (Problème) *Soit $<\mathcal{S},\mathcal{N},\mathcal{T}>$ un système multi-agents coopératifs où $\mathcal{N} = \{a_1 \ldots a_n\}$ est un ensemble d'agents et $\mathcal{T} = \{t_1 \ldots t_m\}$ un ensemble pouvant être ordonné de tâches indépendantes. On doit assigner à chaque tâche $t_i \in \mathcal{T}$ un groupe d'agents $C_i \subseteq \mathcal{N}$ tel que la valeur globable de la structure générée soit maximale et que l'ordonnancement sur les tâches soit respecté.*

Dans toute la suite, nous dénotons par $\triangleright$ un pré-ordre (partiel ou total) sur $\mathcal{T}$, c'est-à-dire $\triangleright \subseteq \mathcal{T} \times \mathcal{T}$. Soient t_1, $t_2 \in \mathcal{T}$, $t_1 \triangleright t_2$ veut dire que t_1 doit obligatoirement être réalisée avant t_2. Ou encore, la réalisation de t_1 est obligatoire pour la réalisation de t_2.

4.2.2 Instanciation du CGS

Dans l'application présentée précédemment, deux relations de conflits ont été définies. La première capture l'idée de coalitions disjointes et la seconde capture le fait qu'une tâche est réalisable par une seule coalition à la fois. Dans cette nouvelle application, étant donné que nous acceptons les coalitions recouvrantes, la relation de contrariété *Interfére* n'est plus utile.

La question que l'on se pose est la suivante : peut-on appliquer le système défini dans la section 4.1.2 à notre nouvelle application en gardant uniquement la relation *Concurrence*. En d'autres termes, peut-on utiliser le système $<\mathcal{C}_{disj}$, $Concurrence$, $\succeq>$?

La réponse est non. Avec un tel système, les structures de coalitions calculées (basiques, préférées, stables, extensions) peuvent violer l'ordonnancement des tâches. Illustrons cela sur l'exemple suivant :

Exemple 12 *Soient $\mathcal{T} = \{t_1,t_2,t_3\}$ avec $t_1 \triangleright t_2 \triangleright t_3$, $\mathcal{C}_{disj} = \{<C_1,t_1>, <C_2,t_3>, <C_3,t_3>\}$ et, enfin, $<C_2,t_3> \succeq <C_3,t_3>$.*
Il est clair que $<C_2,t_3>$ concurrence $<C_3,t_3>$ et vise versa. Selon le système $<\mathcal{C}_{disj}$, $Concurrence$, $\succeq>$, on a une seule structure de coalitions qui est à la fois basique, stable, préférée et une extension. Il s'agit de $\underline{\mathcal{S}} = \{<C_1,t_1>, <C_2,t_3>\}$.
Dans cet exemple, la tâche t_3 est affectée au groupe C_2 alors qu'elle ne peut pas être réalisée étant donné que la tâche qui la précède t_2 n'est pas réalisable.

Pour prendre en compte l'ordonnancement des tâches, il est important de redéfinir la notion de tâche réalisable. L'idée est que pourqu'une tâche soit réalisable, il faut qu'il existe un groupe d'agent ayant les capacités nécessaires pour la réalisation de cette tâche et il faudra aussi que toutes les tâches qui la précèdent soient à leur tour réalisables. Pour cela, nous définissions une nouvelle notion de *fortement réalisable* qui assure l'ordonnancement des tâches.

Définition 40 (Tâche fortement réalisable) *Soient $<\mathcal{S}, \mathcal{N}, \mathcal{T}>$ un système multi-agents, $t \in \mathcal{T}$ et $C \subseteq \mathcal{N}$. t est* fortement réalisable *par C, dénoté $C \Vvdash_r t$, si et seulement si :*

1. *$C \Vvdash t$, et*
2. *$\forall\, t' \triangleright t$, $\exists\, C' \subseteq \mathcal{N}$ telle que $C' \Vvdash t'$.*

Ainsi, la nouvelle définition de la coalition et la suivante :

Définition 41 (Coalition) *Soit $<\mathcal{S}, \mathcal{N}, \mathcal{T}>$ un système multi-agents. Une* coalition *est une paire $<C, t>$ telle que :*

1. $C \subseteq \mathcal{N}$

2. $t \in T$

3. $C \Vdash_r t$

4. C *est minimal pour l'inclusion parmi les ensembles qui satisfont les conditions précédentes.*

Soit $\mathcal{C}_r$ l'ensemble de toutes les coalitions ainsi définies.

Le cadre qui sera donc utilisé pour calculer les structures de coalitions dans la nouvelle application est $<\mathcal{C}_r, Concurrence, \succeq>$. Ainsi, la structure basique, par exemple, de ce système est :

$$\underline{\mathcal{S}}_{Concurrence,\succeq} = \bigcup \mathcal{F}^{i>0}(\emptyset)$$

$$= \mathcal{C}_{Concurrence,\succeq} \cup [\bigcup \mathcal{F}^{i\geq 1}(\mathcal{C}_{Concurrence,\succeq})]$$

Théorème 4 *Soient* $<\mathcal{C}_r, Concurrence, \succeq>$ *un CGS et* $\{S_1 \ldots S_r\}$ *l'ensemble des structures de coalitions selon la sémantique* extension.
$\forall\ S_i$*,* $Taches(S_i)$ *est ordonné selon* $\triangleright$*.*

Preuve 1 *Soit* S *une extension représentant une structure de coalitions. Dénotons par* $Taches(S)$ *l'ensemble des tâches réalisées par* S*. Supposons que* $t \in Taches(S)$ *avec* $<C,t> \in S$*, et* $\exists\ t' \notin Taches(S)$ *telle que* $t' \triangleright t$*. Ceci implique* $\nexists <C',t'> \in S$*. Deux cas peuvent exister :*

Cas 1 : $\nexists <C',t'> \in \mathcal{C}_r$*, c'est-à-dire qu'il n'existe pas de groupe d'agents pouvant réaliser la tâche* t'*, ou encore une autre tâche indispensable pour* t'*. Ceci est impossible car cela contredit le fait que* $<C,t>$ *soit une coalition (d'après la définition d'une coalition). Donc, il y a contradiction.*

Cas 2 : $\exists <C',t'> \in \mathcal{C}_r$*. Par définition, une extension* S *est un ensemble maximal (pour* $\subseteq$*) et sans-conflit de coalitions. Donc,* $\exists <C'',t''> \in S$ *telle que* $<C'',t''>$ *concurrence* $<C',t'>$*. Or par définition de* Concurrence*,* $t'' = t'$ *et, par conséquent,* $t' \in Taches(S)$*. Il y a contradiction.*

Théorème 5 *Soient* $<\mathcal{C}_r, Concurrence, \succeq>$ *un CGS et* $\{S_1 \ldots S_r\}$ *l'ensemble des structures retournées.*
$\forall\ S_i$*, si* S_i *est une structure stable alors elle est aussi une extension.*

Preuve 2 *Soit* S *une structure stable alors :*

1. $\forall\ C_i \notin S$*,* $\exists\ C_j \in S$ *telle que* C_j *concurrence* C_i *et* $C_j \succeq C_i$*.*

Supposons que S *n'est pas une extension. Deux cas se présentent :*

Cas 1 : S *n'est pas sans-conflit. Ceci est impossible par définition car* S *est une structure stable.*

Cas 2 : S *n'est pas maximal pour l'inclusion. Dans ce cas,* $\exists\ C \notin S$ *telle que* $S \cup \{C\}$ *est sans-conflit. Ceci implique que* $\nexists\ C' \in S$ *telle que* C' *concurrence* C *et donc* $\nexists\ C' \in S$ *telle que* C' *concurrence* C *et* $C' \succeq C$*. Ceci contredit 1.*

Par conséquent, S est une extension.

Notons que les résultats précédents ne sont pas vérifiés par les sémantiques basiques comme le montre bien l'exemple suivant :

Exemple 13 *Soit $\mathcal{T} = \{t_1,t_2\}$ avec $t_1 \triangleright t_2$. Supposons que l'on a les coalitions suivantes : $\mathcal{C}_r = \{<C_1,t_1>,<C_2,t_1>,<C_3,t_2>\}$ avec $<C_1,t_1> \succeq <C_2,t_1>$ et $<C_2,t_1> \succeq <C_1,t_1>$, c'est-à-dire que les deux coalitions ont la même préférence. La structure basique contient donc une seule coalition $\underline{\mathcal{S}}_{Concurrence,\succeq} = \{<C_3,t_2>\}$. Or, t_2 ne peut être réalisée dans ce cas.*

4.2.3 L'algorithme de Shehory et Kraus

Shehory et Kraus proposent un algorithme où la génération des structures de coalitions ainsi que le calcul des valeurs de coalitions restent identiques au modèle précédent. La 1ère étape de l'algorithme, tout comme la 2ème si l'on travaille sur des coalitions disjointes, sont donc à réutiliser. Cependant, accepter des coalitions recouvrantes nécessite de redéfinir l'algorithme de la 2ème étape. Chaque agent possède toujours une liste de coalitions L_i ainsi que leurs valeurs. Dans le cas de coalitions recouvrantes, chaque agent a_i exécute :

1. Sélectionner L_i dans la coalition C_j qui possède le plus petit coût c_{C_j}.
2. Annoncer publiquement le coût de coalition c_{C_j}.
3. Choisir parmi toutes les annonces la coalition C_{min} qui a le coût c_{min} le plus faible ainsi que la tâche t_{min} qu'elle réalise.
4. Si $a_i \in C_{min}$ alors rejoindre les autres membres et former la coalition.
5. Supprimer de T la tâche t_{min}.
6. Mettre à jour les vecteurs de compétence des membres de C_{min} en fonction de leur contribution à la réalisation de t_{min}.
7. Assigner L_i^{cr} à les coalitions de L_i dont la valeur doit être recalculée.

Ordonnancement des tâches

Si les tâches sont ordonnées, il est nécessaire de modifier l'algorithme. On note P_t l'ensemble des prédécesseurs de la tâche t en incluant elle-même et pV_t la somme de toutes les valeurs des tâches de P_t. Tous les agents doivent exécuter itérativement ce qui suit jusqu'à ce qu'il n'y ait plus d'agents ou de tâches à réaliser :

1. Pour chaque tâche $t \in T$ faire :
   (a) Calculer les valeurs de toutes les tâches de P_t comme dans la 1ère étape et substituant T par P_t.
   (b) Si toutes les tâches dans P_t sont assignées à une coalition alors calculer pV_t.
   (c) Sinon supprimer t de T car elle ne peut pas être réalisée.

2. Sélectionner la tâche t^* ayant la pV_t maximale pour être réalisée ainsi que tous ses prédécesseurs. Former les coalitions nécessaires comme dans la 2$^{\text{ème}}$ étape.
3. Supprimer t^* et tous ses prédécesseurs de T.
4. Si l'on utilisant des coalitions recouvrantes, mettre à jour les vecteurs de compétence des agents, sinon effacer les agents ayant formé des coalitions de la liste des candidats disponibles.

Chapitre 5

Etat de l'art

5.1 Formation de coalitions pour des agents compétitifs

Shehory et Kraus ont aussi proposé dans [16] un modèle pour agents *comp étitifs*, c'est à dire des agents rationnels qui tentent de maximiser une fonction d'utilité personnelle sans se soucier d'une utilité globale. Cependant pour trouver une solution optimale, il faut garantir la stabilité des coalitions : aucun agent ne désirera changer de coalition une fois qu'elles auront été calculées.

Les hypothèses suivantes sont posées :
- Les agents ont une information complète dont l'accès peut être coûteux.
- Les communications sont coûteuses.
- Un système monétaire évalue les ressources et les productions.
- Les ressources et la monnaie sont transférables entre agents.
- L'environnement n'est ni superadditif, ni subadditif.
- Les coalitions sont disjointes.

Le modèle de Shehory et Kraus

Shehory et Kraus proposent la définition d'un équilibre limité dans le temps : un profil de stratégie p est un équilibre limité dans le temps si chaque entité croit, en fonction des informations dont elle dispose après un temps prédéfini ou une profondeur de recherche équivalente, qu'elle maximise son utilité en respectant sa stratégie pour le temps de calcul donné et que les autres entités font de même. Il s'agit donc d'une approximation dans le temps d'un équilibre de Nash.

Le concept rationnel qui se trouve derrière l' *équilibre limité en temps* est le suivant : comme le calcul est coûteux et que la valeur des gains d'une coalition est finie, les bénéfices qu'une entité peut attendre en formant une coalition décroissent au fur et à mesure que le temps passé à la négociation et à la formation augmente. A un moment donné, le gain sera nul. En fixant le temps de calcul ou la profondeur de recherche dans l'espace des stratégies, le risque d'avoir une coalition sans bénéfice (dont les gains sont inférieur au coût de la recherche) est réduit.

O. Shehory et S. Kraus ont développé dans un algorithme polynomial, le DNPK-CFM, qui calcule un équilibre limité dans le temps. Cependant, cette faible complexité n'est due qu'à une restriction sur la taille des coalitions.

Définition 42 (Valeur d'une coalition) *La* valeur *d'une coalition C, noté $V(C)$, est une évaluation de C si les membres de l'ensemble N_C peuvent arriver à une fonction de gain conjointe V. $V(C) = \sum_{a_i \in N_C} U^i(\overline{\Theta})$ où U^i est la fonction de gain de l'agent a_i et $\overline{\Theta}_i$ est son vecteur de ressource après redistribution dans C.*

Soit $U = < u^1 \dots u^n >$ un vecteur de gain où u^i est le gain de l'agent a_i. A chaque étape du processus de formation des coalitions, les agents se trouvent dans une structure de coalitions disjointe. Le couple vecteur de gain et structure de coalitions est noté $PC(U, \mathcal{C})$ ou juste PC pour *Payment Configuration*. L'espace de structures de coalitions, CSS, est l'ensemble de toutes les structures $\mathcal{C}$ telles que $\forall C_i \in \mathcal{C}$: $V_{C_i} \geq \sum_{a_j \in C_i} V_{\{a_j\}}$. L'espace de structure de gain (PCS) est un couple $(U, \mathcal{C})$ où U est un vecteur de gain rationnel individuel et $\mathcal{C} \in CSS$. L'amplitude de CSS est de $\mathcal{O}(n^n)$.

Le Noyau K

Le noyau est une notion de stabilité pour une structure de coalitions. C'est un PCS dans lequel chaque structure de coalitions PC est stable dans le sens où tout agent a_i et a_j, membres de la même coalition C, est en équilibre l'un par rapport à l'autre.

Définition 43 (Surplus maximum) *Le* surplus maximum, *noté S_{ij}, d'un agent a_i vis-à-vis de a_j au sein du même PC est défini par $S_{ij} = \max_{\{C \mid a_i \in C, a_j \notin C\}} e_C$ où e_C, l'*excès *de la coalition C, est $e_C = V(C) - \sum_{a_i \in C} u^i$. L'agent a_i* écrase *a_j si $S_{ij} > S_{ji}$ et $u^j > V(\{a_j\})$ où $V(\{a_j\})$ est la valeur de coalition de l'agent a_j en tant que coalition singleton.*

Définition 44 (Fonction de demande) *Soient un PC et deux agents a_i et a_j de la même coalition. La* fonction de demande *d_{ij} de l'agent a_i sur l'agent a_j est définie de la manière suivante : si $S_{ij} \geq S_{ji}$ alors $d_{ij} = \min[\frac{S_{ij} - S_{ji}}{2}, X_j]$ sinon $d_{ij} = 0$ où X_j est le gain de l'agent a_j.*

Définition 45 (K-stabilité) *Un PC est* K-stable *si $\forall i \neq j$, les agents a_i et a_j au sein de la même coalition $C \in PC$ se trouvent à l'équilibre. Un PC appartient au noyau si et seulement si il est K-stable. Deux agents sont à l'équilibre si et seulement si une des conditions suivantes est satisfaite :*
- $S_{ij} = S_{ji}$
- $S_{ij} > S_{ji}$ *et* $u^j = V(\{a_j\})$
- $S_{ij} < S_{ji}$ *et* $u^i = V(\{a_i\})$

Définition 46 (PC-erreur relative) *Soient une structure de coalitions $\mathcal{C}$, un vecteur de gain U et a_i, a_j des agents d'une même coalition de $\mathcal{C}$. On définit la PC-*erreur relative *e_r comme étant le ratio entre la PC-erreur $e = \max_{i,j}(S_{ij} - S_{ji})$ et la somme de tous les gains de tous les agents du PC donné.*

Proposition 1 (Optimalité) *Soient une structure de coalitions $\mathcal{C}$ et $PC_{\mathcal{C}} = \{PC_i(\mathcal{C}, U_i)\}$ où $PC_i \in K$ (le noyau) est l'ensemble de tous les PC qui sont constitués du vecteur de gain U_i et de la structure de coalitions $\mathcal{C}$. $\forall\ j$, si $PC_j \in PC_{\mathcal{C}}$ alors PC_j est un optimum de Pareto local de $PC_{\mathcal{C}}$.*

Le concept de noyau donne une distribution des gains stable pour n'importe quelle configuration de coalition. Une des propriétés du noyau est que des agents symétriques recoivent des gains équivalents. De plus, le noyau ainsi que l'ensemble de négociation est un sous-ensemble des PCS de tout PC rationnel. Cependant, le noyau est un sous-ensemble de l'ensemble de négociation, donc plus restreint. Enfin, le formalisme du noyau permet de diviser son calcul en petits processus polynomiaux, simplifiant ainsi même les cas exponentiels. Un PC K-stable calculé avec une précision $\epsilon > 0$ est appelée un PC K-ϵ-stable.

Modèle de formation des coalitions par DNPK-CFM

Le DNPK-CFM signifie Modèle de Formation de Coalition orienté-Noyau, Polynomial, orienté-Négociation et Distribué. Tous les agents sont des coalitions singletons et forment leurs coalitions par une sucession de propositions acceptées ou rejetées par les autres. Ceci continue jusqu'à ce qu'il y ait un point fixe ou qu'un temps imparti soit écoulé. Un point fixe de l'environnement est un état dans lequel les agents sont organisés en un PC tel que les agents ont trouvé un PC optimal selon Pareto et K-ϵ-stable, ou qu'il n'y a plus aucune proposition accordant un meilleur bénéfice pour eux. Pour réduire la complexité du problème à une valeur polynomiale, on définit deux entiers K_1 et K_2 tels que $K_1 \leq K_2$ pour ne considérer que les coalitions de taille $[K_1, K_2]$ afin de calculer les excès.

Définition 47 (Surplus maximal polynomial) *Le* surplus maximal polynomial, *noté SP, est le surplus maximal calculé à partir d'un ensemble polynomial d'excès. SP peut être utilisé pour examiner le PC à la recherche d'une K-ϵ-stabilité polynomiale. Une coalition $C \in \mathcal{C}$ est K-ϵ-stable polynomiale si $\forall\ a_i$, $a_j \in C$ et $i \neq j$ alors soit $u^i = 0$, soit $u^j = 0$ ou $|SP_{ij} - SP_{ji}| \leq \epsilon$.*

Protocole

On note par P_{xy} une proposition de la coalition C_x à la coalition C_y. Avant la négociation, les agents doivent calculer les valeurs des coalitions de taille bornée $[K_1, K_2]$ avec le DEK-CFM. De plus, un membre d'une coalition ne peut pas recevoir de proposition individuelle : les coalitions ne peuvent que s'étendre et il n'y a pas de destruction possible. Dans un premier temps, chaque coalition exécute :

1. Chaque coalition coordonne ses actions par un représentant et/ou le vote.

2. Chaque coalition C_p exécute itérativement :
   (a) Transmettre une proposition P_{pr} à la coalition C_r et attendre la réponse. P_{pr} est un cahier des charches de la coalition jointe C_{p+r} et de la configuration PC_{new} : redistribution des ressources et des tâches, distribution des gains, K-ϵ-stable polynomiale en fonction de K_1 et K_2. Le vecteur de gain U_{p+r} de C_{p+r} doit accroître les gains de chaque membre. PC_{new} doit inclure C_{p+r} sans modifier les autres coalitions.
   (b) Accepter P_{rp} si et seulement si $P_{pr} = P_{rp}$ et envoyer une confirmation à C_r. Si plusieurs propositions sont identiques, celle qui a été reçue la première est acceptée.
   (c) Si P_{pr} est acceptée et confirmée par C_p et C_r, construire C_{p+r} en suivant le cahier des charges de P_{pr}. Si nécessaire, C_{p+r} doit choisir un représentant.
   (d) Informer C_a, $\forall a \neq r$, de l'acceptation de P_{rp} et rejeter toutes P_{ap}.
   (e) Réitérer cette séquence jusqu'à un point fixe ou la fin du temps imparti.

Pour permettre une perception du point fixe, chaque coalition doit annoncer son statut (qu'il n'y a plus de proposition à transmettre). Cette annonce n'est valide que pour l'itération courante. Si le temps est écoulé, c'est le dernier PC qui devient valide.

Une seconde étape optionnelle peut se greffer à la première. Elle permet de travailler en autorisant les destructions de coalitions et, par conséquent, permettre de faire des propositions ne concernant qu'un unique agent. Lorsqu'un nouveau PC modifie les gains d'un agent, cela peut le conduire à quitter sa coalition qui est alors détruite. Comme la première, la seconde étape se termine lorsqu'un point fixe est trouvé ou que le temps imparti est écoulé. Si toutes les configurations de coalition possibles ont été calculées durant cette étape, le processus est arrêté. De même, pour permettre la perception du point fixe, chaque coalition doit annoncer son statut lorsqu'elle n'a plus de proposition à transmettre. Cette annonce ne dépend pas de celle qui a été faite dans la première étape.

Stratégie

Il est nécessaire de définir une stratégie pour qu'une coalition C_i fasse une proposition à une coalition C_j. Pour avoir une représentation compacte de ces stratégies, définissons $E^j_{P_i}$ le gain espéré de la coalition C_j d'une proposition P_{ij} pour le temps de calcul donné, q_{ij} la probabilité que C_i fasse une proposition à C_j, T_{pc} le temps utilisé pour calculer la proposition.

Soient une structure de coalitions $\mathcal{C}$ et C_i, $C_j \in \mathcal{C}$, on définit Q^k_{ij} la probabilité qu'il puisse y avoir k autres paires de coalition C_x, $C_y \in \mathcal{C}$ où $x \neq y \neq i \neq j$ telle que pour chacune de ces paires, C_x et C_y se fassent mutuellement une offre identique.

Le but est de hiérarchier et sélectionner les propositions. En entrée, on utilise toutes les valeurs de coalitions dans les bornes K ainsi que le seuil de temps limite T. En sortie, cette stratégie doit retourner une proposition P_{px} à faire à une coalition C_x. C_p doit alors faire :

Si T ne permet pas de multiples calculs de proposition :

1. Ordonner chaque coalition $C_r \neq C_p$ en fonction du gain espéré de C_p à former C_{p+r}. Selectionner une des méthodes ci-dessous en fonction des ressources computationnelles disponibles :
   (a) Calculer explicitement la valeur de la coalition jointe et le vecteur de gain polynomial-K-ϵ-stable correspondant.
   (b) Estimer les gains espérés en observant, par exemple, les quantités de ressources disponibles des autres coalitions et en examinant les gains précédents que leurs membres ont pu recevoir.
2. Calculer une proposition pour la coalition de plus haut rang et l'assigner à P_{px}.

Sinon :

1. $\mathcal{P}_{pr} = \emptyset$ où $\mathcal{P}_{pr}$ est l'ensemble des proposition de C_p pour C_r, $r \neq p$.
2. Tant que $T_{pc} < T$ faire en fonction du temps écoulé T_{pc} :
   (a) $\forall C_r$, calculer P_{pr} et $E^p_{P_{pr}}$. Si $E^p_{P_{pr}} > 0$ alors ajouter P_{pr} à $\mathcal{P}_{pr}$.
   (b) Selectionner la meilleure proposition pour C_p, $P_{pr'} \in \mathcal{P}_{pr}$, et calculer les $P_{ar'} \in \mathcal{P}_{ar'}$ que $C_{r'}$ devrait recevoir des autres coalitions $C_a \in \mathcal{C}$, $a \neq r'$. Pour chaque $P_{ar'}$, calculer $E^{r'}_{P_{ar'}}$ en fonction de T_{pc}.
   (c) Si P_{pr} est meilleure que les autres propositions dans $P_{ar'}$ (c'est-à-dire avec une valeur de gain attendue supérieure) assigner P_{pr} à P_{px} et sortir, sinon continuer.
   (d) S'il y a des propositions dans P_{pr} :
      i. $\forall C_i$, décider à quelles coalitions C_j, $i \neq j$, elle est suceptible de faire une proposition. Calculer les propositions P_{ij} et les ajouter à P_i.
      ii. Pour chaque C_i et pour chaque $P_{ij} \in \mathcal{P}_i$, calculer $E^j_{P_{ij}}$.
      iii. Tant que T le permet et qu'il n'y a pas eu de solution trouvée, sélectionner le plus petit $\mathcal{P'}_i \subset \mathcal{P}_i$ et assigner une probabilité q_{ij} à chaque proposition $P_{ij} \in \mathcal{P'}_i$ en résolvant les équations pour chaque paire de propositions P_{ij}, $P_{ik} \in \mathcal{P'}_i$, $i \neq j \neq k$, $q_{ij}(E^i_{P_{ij}} \sum_{l=0}^{\lfloor \frac{nc}{2} \rfloor -1} \frac{1}{l} Q^l_{ij}) = q_{ik}(E^i_{P_{ik}} \sum_{l=0}^{\lfloor \frac{nc}{2} \rfloor -1} \frac{1}{l} Q^l_{ik})$ et $\sum_l q_{il} = 1$.
      iv. Associer une probabilité q_{pr} à chaque proposition de $\mathcal{P'}_{pr}$.
      v. Choisir aléatoirement une proposition de $\mathcal{P'}_{pr}$ en fonction de leur probabilité et l'assigner à P_{px}.
3. Si $P_{px} \neq \emptyset$, il s'agit de la proposition sélectionnée pour C_p, sinon il n'y a pas de proposition bénéfique pour C_p et il faut annoncer un point fixe.

Bien que le nombre d'équation pour le calcul des probabilités puisse être plus grand que le nombre de probabilité lui-même et impliquer l'existance de $\mathcal{P}_i S$ où il n'y a pas de solution, l'environnement est un jeu fini à n personne qui, en théorie des jeux, possède toujours au moins un équilibre en stratégie mixte.

Election de représentants

Dans le but d'économiser du temps et d'éviter à ce que tous les membres d'une coalition soient obligé de calculer les propositions, une coalition peut désigner un émissaire chargé de concevoir, envoyer et recevoir les propositions ou bien de procéder par vote. Voter est avantageux pour des agents possédant de fortes compétences de calcul. Cependant, il s'agit d'une méthode beaucoup plus coûteuse. Désigner un agent, qui peut être remplacé à chaque itération, pour représenter la coalition minimisera les besoins de calcul. On peut noter que des agents compétitifs n'accepteront pas cette charge si elle n'est pas compensée, à moins d'être forcés par le protocole.

Construire une proposition

Le but est de construire une proposition en faveur de la coalition jointe C_{p+r}. On utilise en entrée le PC courant et les ressources, tâches et fonctions de gains de la coalition. En sortie, on reçoit une proposition, un nouveau PC, noté PC_{new}. La coalition C_p doit faire :

1. Calculer la valeur de coalition $V(C_{p+r})$ de C_{p+r}.
2. Si $V(C_p) + V(C_r) \geq V(C_{p+r})$, abandonner cette proposition pour C_r.
3. Sinon, $\forall C_a / |C_a|$ dans $[K_1, K_2]$ retrouver ou calculer $V(C_a)$.
4. Calculer U_{new}, le vecteur de gain de PC_{new} dans lequel C_p et C_r forment C_{p+r} et tout les C_a restent non modifiés en utilisant les méthodes de calcul précédentes.
5. Comparer U_{new} au U courant. Si $\forall a_i \in C_{p+r}$, $u^i_{new} \geq u^i$ et $\exists a_i \in C_{p+r}$, $u^i_{new} > u^i$ alors PC_{new} est la proposition pour C_r sinon abandonner cette proposition-ci.

Soit une période de temps donnée t_{wait}. Après avoir fait une proposition P_{pr} à C_r, C_p doit attendre t_{wait} une réponse. Durant cette période, C_p est assigné à sa proposition. S'il ne reçoit pas de réponse dans le délai accordé, cette obligation est levée.

Capture par un CGS

Dans cette application, les agents ne cherchent pas à réaliser des tâches données mais simplement former des groupes. Ici, il n'y a plus de notions de tâches pour la coalition et cette dernière est un simple ensemble d'agents. Ainsi, la relation de contrariété *Concurrencer* n'a plus lieu d'être.

Capturer les spécificités de cette application avec un CGS nécessiterait donc une nouvelle définition de la notion de coalition et de contrariété. Ici, la coalition serait donc n'importe quel sous-ensemble de $\mathcal{N}$ et les conflits entre coalitions seraient modélisés par la relation *Interférer*.

5.2 Méthodes et heuristiques

5.2.1 Recherche d'un résultat minimum

Les modèles précédents [14, 15, 16] recherchent une solution optimale en limitant la taille des coalitions pour assurer une faible complexité à leur algorithme. Cependant, cette heuristique ne garantit pas l'optimalité des solutions. En effet, ces dernières peuvent se trouver hors des bornes fixées. De plus il est toujours nécessaire de calculer les structures des coalitions au sein de ces bornes, ce qui est d'une haute complexité lorsque les agents sont en grand nombre. C'est pour cette raison que des heuristiques plus performantes ont été développées.

L'algorithme de Sandholm et *al.*

Sandholm, Larson, Andersson, Shehory et Tohmé proposent dans [12] un algorithme *anytime* de génération des structures de coalitions permettant d'obtenir un résultat final au moins égal à une certaine proportion du résultat optimal, ce qui permet de diminuer la taille de l'espace de recherche. Cet algorithme s'applique sous les hypothèses suivantes :

- Le système est un jeu à fonction caractéristique.
- L'environnement n'est pas nécessairement superadditif ou subadditif.
- Les coalitions sont disjointes.
- Les agents sont coopératifs.

Voici les notations employées par la suite dans cette section :

$\mathcal{N}$: l'ensemble des agents du système, $\mathcal{N} = \{a_1 \dots a_n\}$.

n : le nombre d'agents du système, soit $|\mathcal{N}|$.

C : une coalition.

$Valeur(C)$: la valeur de la coalition C.

S : une structure de coalitions disjointes.

$Valeur_g(S)$: la valeur globale de la structure S.

S^* : la structure de coalitions optimale : $S^* = \arg\max_{S \in \mathcal{S}} Valeur_g(S)$.

$\mathcal{S}$: l'ensemble de toutes les structures de coalitions possibles.

$\mathcal{S}_{\Updownarrow}$: les dernières structures de coalitions calculées lors de la recherche.

m : leur nombre, soit $|\mathbf{S_m}|$.

m_{min} : m minimum qui garantit un ratio borné sur l'optimum.

S^*_m : la structure de coalitions avec la plus grande utilité globale parmi celles calculées.

k : ratio borné dans le pire cas sur la valeur de la structure de coalitions.

Un treillis de structures de coalitions

L'idée est de raisonner à partir du niveau des partitions de coalitions possibles représentées sous forme de treillis. Au premier niveau se trouve la grande coalition, c'est à dire la structure de coalitions qui ne comporte qu'une unique coalition composée de l'ensemble des agents. Les nœuds du deuxième niveau correspondent aux structures de coalitions qui comprennent deux coalitions. Et ainsi de suite jusqu'au dernier niveau qui comprend une unique structure de coalitions composée de toutes les coalitions singletons du système.

Calculer toutes les structures de coalitions est trop complexe. On désire donc n'en calculer qu'un sous-ensemble $\mathcal{S}_{\Updownarrow} \subseteq \mathcal{S}$ tel que en sélectionnant la meilleure structure trouvée, $S^*_m = \arg\max_{S \in \mathcal{S}_{\Updownarrow}} Valgeur_g(S)$, on puisse garantir qu'elle est au voisinage de l'optimum, c'est-à-dire qu'il existe un k petit et fini tel que : $k = \min \kappa$ où $\kappa \geq \frac{Valeur_g(S^*)}{Valeur_g(S^*_m)}$. On définit alors m_{min} comme étant la plus petite taille de $\mathcal{S}_{\Updownarrow}$ qui permet d'établir une telle borne k.

En posant $Valeur_g(S^*_m) \leq n.\max_C Valeur(C) \leq n.\arg\max_{S \in \mathcal{S}_{\Updownarrow}} Valeur_g(S) = n.Valeur_g(S^*_m)$, on peut prouver qu'il suffit d'une recherche sur les deux premiers niveaux pour borner k et on obtient $k = n$ et $m = (2^{n-1})$. Aucun autre algorithme ne permet d'établir une borne sur k en n'explorant que (2^{n-1}) nœuds ou moins. Ainsi $m_min = (2^{n-1})$.

Cela signifie que l'on peut garantir une approximation de l'optimum sans explorer toutes les structures de coalitions et que plus il y a d'agents dans le système, plus une petite partie du graphe doit être explorée. Cependant, il reste toujours un nombre exponentiel de nœuds et aucun algorithme de génération de structures de coalitions pour jeu à fonction caractéristique ne peut trouver une borne sans évaluer au moins (2^{n-1}) structures.

Algorithme

1. Explorer les deux premiers niveaux du graphe de structures de coalitions.
2. Explorater en largeur d'abord à partir du dernier niveau du graphe jusqu'à ce que le temps impartit soit écoulé ou que le graphe ait été complétement exploré.
3. Retourner la structure de coalitions ayant la plus grande utilité globale.

Après avoir exploré les deux premiers niveaux, continuer la recherche permet de diminuer la borne. Ainsi, si l'on suppose avoir atteint le niveau l alors la borne $k(m)$ est $\lceil \frac{n}{h} \rceil$ si et seulement si $n \equiv h - 1 \pmod h$ et $n \equiv l \pmod 2$, sinon $\lfloor \frac{n}{h} \rfloor$ où $h = \lfloor \frac{n-l}{2} \rfloor + 2$. La borne est $k = n$ lors que $m = (2^{n-1})$ mais en incrémentant m de 1, on obtient $k = \frac{n}{2}$, puis $k = \frac{n}{3}$ en explorant encore deux niveaux supplémentaires. Plus simplement, le diviseur augmente de un tous les deux niveaux supplémentaires explorés.

On peut toutefois remarquer que lorsque le nombre d'agents devient un peu important, le minimum assuré est très bas par rapport à l'optimum, puisque n est le numérateur de la borne, alors que le nombre de nœuds à calculer et la complexité du calcul augmente rapidement.

Optimisation de Dung Dang et Jenning

Dung Dang et Jenning ont alors proposé une optimisation de l'algorithme de Sandholm et *al.* dans [6]. Après avoir exploré les deux premiers niveaux, l'algorithme recherche des structures de coalitions à partir du centre du graphe. Ainsi, on définit $\mathcal{S}(n, c)$ comme étant l'ensemble de toutes les structures de coalitions de cardinalité entre 3 et $n-1$ qui possède au moins une coalition dont la cardinalité est supérieure ou égale à c. L'algorithme se réécrit ainsi :

1. Explorer les deux premiers niveaux du graphe de structures de coalitions.
2. Explorer $\mathcal{S}(n, \left\lceil \frac{n(q-1)}{q} \right\rceil)$ avec $q = \lfloor \frac{n+1}{4} \rceil$.
3. Pour tout q de $\lfloor \frac{n+1}{4} \rceil - 1$ à 2, explorer $\mathcal{S}(n, \left\lceil \frac{n(q-1)}{q} \right\rceil) \setminus \mathcal{S}(n, \left\lceil \frac{nq}{q+1} \right\rceil)$
4. Retourner à chaque étape la structure de coalitions possèdant la plus grande valeur.

Application au CGS

Cette heuristique est utilisable optimiser la génération des structures de coalitions dans un cadre CGS. Le treillis des structures de coalitions est parcouru pour déterminer si une structure est acceptable selon une sémantique donnée ou non. En utilisant l'algorithme de Sandholm ou son optimisation, on peut garantir que les structures trouvées dans les bornes spécifiées seront une approximation de l'optimum si le treillis complet ne peut pas être parcouru.

5.2.2 Stratégies en information incomplète

Kraus, Shehory et Taase dans [9] relâchent certaines hypothèses restrictives du domaine dont notamment celle d'information complète. Dans un environnement *Request for Proposal*, les agents sont des contractants rationels qui se regroupent en coalitions disjointes pour effectuer des requêtes complexes composées de plusieurs sous-tâches sans pour autant connaître les coûts de réalisation des autres agents.

Définition 48 (Système multi-agents RFP) *Un système multi-agent RFP est un triplet $< T, N, \phi >$ tel que :*

– $T = \{t_1 \dots t_n\}$ *est un ensemble de tâches RFP où* $t_i \in T$ *est un ensemble de sous-tâches* $\{t_{il} \dots t_{im}\}$.
– $N = \{a_1 \dots a_n\}$ *est un ensemble d'agents compétitifs et rationels où chaque* a_j *posséde un coût de réalisation d'une sous-tâche* t_{ik} *noté* $b_{a_j}(t_{ik})$*. Ce coût est une connaissance privée de l'agent.*
– $\phi : T \times N \longmapsto \{\top, \bot\}$ *est une fonction booléenne qui associe à un agent* a_j *et à une sous-tâche* t_{ik} *la valeur* $\top$ *si* a_j *peut réaliser* t_{ik} *ou la valeur* $\bot$ *dans le cas contraire. Cette fonction est une connaissance commune des agents.*

Définition 49 (Coalition) *Une coalition* C_t *formée pour réaliser une tâche* t *est un triplet* $< N_t, \Phi_t, U_t >$ *où* N_t *est un ensemble d'agents,* Φ_t *une fonction d'allocation qui associe à chaque sous-tâche de* t *un membre de* N_t *tel que* $\Phi_t(t_{ik}) = a_j$ *si et* *seulement si $\phi(t_{ik}, a_j) = \top$, et $U_t =< u_1 \dots u_{|N_t|} >$ un vecteur de distribution des gains où u_j est le bénéfice de l'agent a_j.*

Protocole

L'idée est que lors de la soumission de la tâche, une valeur lui est attribuée en fonction de son besoin. Plus le temps passe, plus la tâche devient urgente et sa valeur diminue. Le rôle du commissaire-priseur est de révéler publiquement ces RFP, rassembler les différentes propositions et élire une coalition pour chacune. Il est aussi responsable de la distribution des gains entre les agents d'une coalition. Le processus de formation des coalitions se déroule en tour de $r_1 \dots r_f$ et à chaque tour r :

1. $\forall t_i \in T$, le commissaire-priseur fixe le gain $P(t_i)\delta^r$.
2. $\forall a_j \in N$, l'agent a_j peut faire une des actions suivantes :
   (a) Spécifier une coalition $C_{t_i} =< N_t, \Phi_t, U_t >$ où $a_j \in N_t$ et la proposer à tous les agents a_k tels que $a_k \in N_t$
   (b) Accepter une proposition de coalition C_{t_i} faite par un agent a_k.
3. Chaque coalition formée envoie sa requête au commissaire-priseur qui la vérifie.
4. Le commissaire-priseur accepte $C^*_{t_i}$ la première coalition proposée ou tirée aléatoirement s'il y en a plusieurs simultanées.
5. Les agents de $C^*_{t_i}$ forment la coalition et se retirent du processus.
6. L'itération est relancée avec toutes les tâches non allouées.

Définition 50 (Coût, profits et gains) *Le* coût *B_{C_t} d'une coalition C_t est la somme des coûts des agents qui la compose pour l'accomplissement de leurs sous-tâches à l'intérieur de C_t, soit $B_{C_t} = \sum_{a_j \in N_t} b_{a_j}$. Le* profit *d'une coalition C_t pour réaliser la tâche t au tour r est $\pi = P(t)\delta^r - B_{C_t}$. Chaque agent $a_j \in N_t$ reçoit un* gain *$\pi_{a_j} = b_{a_j} + \frac{\pi}{|N_t|}$.*

Stratégies et heuristiques

Les agents doivent décider quelles coalitions ils vont proposer et lesquelles ils vont accepter. L'agent doit dans un premier temps construire des groupes d'agents, connaissant leurs capacités, qui peuvent réaliser conjointement chaque tâche. Ceci est cependant d'une complexité exponentielle. A chaque tour, l'agent inspecte et classe les coalitions possible. S'il n'a pas reçu de proposition, il en émet une. Un agent qui reçoit des propositions les compare à la meilleure coalition possible à son sens. Il accepte la meilleure proposition qu'on lui fait, sinon il émet la meilleure proposition qu'il a estimé.

Pour classer les coalitions, un agent, ne connaissant pas le coût des autres et par conséquent celui de la coalition, doit les estimer. Il peut arbitrairement les fixer au même que le sien et les réviser en fonction du comportement de chaque agent. Cependant, la part de gain exigé par un agent ne doit pas servir de seul estimateur car il est possible de trop demander pour espèrer en avoir plus. Une fois les coalitions construites et les coûts estimés d'une quelconque manière, l'agent peut appliquer des heuristiques pour sélectionner la meilleure coalition.

Heuristique marginale : elle se base sur le profit marginal de toute la coalition. L'agent somme les coûts estimés de chaque agent de la coalition et soustrait la valeur courante de la coalition. La différence est la valeur nette de la tâche. On normale ceci en la divisant par la valeur maximale possible pour une coalition. Ceci pousse les agents à choisir une coalition possèdant une haute valeur normalisée.

Heuristique experte : elle se base sur une réduction de la concurrence par capitalisation des compétences. Un agent peut se considérer expert dans un domaine si peu d'agents peuvent accomplir les mêmes sous-tâches que lui. Cette heuristique conduit les agents à rechercher des tâches comprenant un faible de nombre de concurrents.

Si l'heuristique marginale tend à favoriser la maximisation de l'utilité (sans toutefois la garantir puisque les valeurs marginales ne sont que des estimations), elle peut poser problème lorsque de nombreuses coalitions sont en concurrence, réduisant grandement l'espèrence de gain. Une heuristique experte ne se base pas sur les coûts (donc une estimation), réduit les collusions entre coalitions et permet de réaliser globalement plus de tâches.

Un problème majeur des deux heuristiques est la récurrence des propositions. Une proposition peut être faite durant un tour, être refusée, puis être reproposée, etc. Une première adaptation pour pallier ceci est d'exclure les agents qui ont refusé de toute prochaine proposition. Cependant, cela peut conduire à empêcher la formation de coalitions profitables. Une seconde adaptation consiste plutôt à ne jamais reproposé une même allocation de sous-tâches à des agents qui l'ont déjà refusée.

Diverses simulations ont été faites avec une heuristique marginale, experte, presque marginale, presque experte et mixte. Dans leur globalité, il en ressort un résultat important. En situation d'information complète, plus l'heuristique est marginale, plus les profits sont importants et inversement pour l'heuristique experte. En situation d'information incomplète, plus l'heuristique est experte, plus les profits sont importants et inversement pour l'heuristique marginale.

Capture par un CGS

Le système multi-agents ayant été redéfinit, il est nécessaire de faire de même pour la notion de *tâche réalisable* de manière à ne plus prendre en compte des ressources ou compétences à valeurs réelles mais binaires.

Définition 51 (Tâche réalisable) *Une coalition* C réalise *la tâche* t_i *si pour toute sous-tâche* t_{ij} *de* t_i*, il existe un agent* $a_i \in C$ *tel que* $\phi(a_i, t_{ij}) = \top$.

La relation de contrariété reste *Contrarier* (Définition 36) car les coalitions doivent être disjointes et chaque tâche ne peut être allouée qu'à une unique coalition.

Enfin, vient le problème de la relation de préférence. Pour une heuristique marginale, il s'agit d'une comparaison des valeurs des coalitions où ces dernières correspondent au profit marginal de la coalition. Nous revenons donc à nos modèles précédents. Dans le cas d'une heuristique experte (et donc celle la plus appropriée pour une information incomplète), la notion de valeur doit être redéfinie.

Définition 52 (Valeur experte de coalition) *Soit un coalition $C_1 \in \mathcal{C}$ qui réalise la tâche t, la valeur de coalition de C_1 pour un agent a_i, notée $Valeur(C_1)$ selon une heuristique experte est le nombre de sous-tâches de t que a_i est le seul à pouvoir exécuter.*

5.3 De la génération à la négociation

Le cadre CGS est un système interne à chaque agent impliqué dans un processus de formation de coalitions. Les agents génèrent des structures de coalitions en fonction de leurs préférences et de leurs croyances. Dans un cas où l'information est incomplète ou que les agents ont des préférences différentes, l'étape de négociation devient cruciale. Dans cette dernière, les agents comparent les structures qu'ils ont trouvés et cherchent un compromis pour n'en sélectionner qu'une seule.

5.3.1 Processus de communication

Tohmé et Sandholm proposent dans [17] une analyse portant non pas sur la génération des structures de coalitions mais sur le rôle des stratégies et du processus en lui-même concernant le choix des coalitions solutions. Ils tentent de relâcher certaines hypothèses habituellement formulées en formation des coalitions et définissent leur environnement par les faits suivants :

1. Jeu à fonction caractéristique.
2. Agents rationels et compétitifs.
3. Pas de transfert d'utilité entre les agents.
4. Information incomplète quant aux ressources et utilité des autres agents.

Définition 53 (Jeu sous forme normale) *Un jeu $G = (\{Si\}, U)$ est défini par un ensemble d'agents $\mathcal{N} = \{a_1 \dots a_n\}$ où S_i est l'ensemble des stratégies possibles de l'agent a_i et le profil d'utilité résultant U : $\prod_{i=1}^n S_i \longmapsto \Re^n$ tel que pour chaque stratégie $(s_1 \dots s_n) \in \prod_{i=1}^n S_i$ on a $U(s_1 \dots s_n) = (u_1(s_1 \dots s_n) \dots u_n(s_1 \dots s_n))$ où u_i : $\prod_{i=1}^n S_i \longmapsto \Re$ est l'utilité de l'agent a_i.*
Un concept de solution en stratégie pure (c'est à dire sans hasard) est une correspondance γ : $G \longmapsto \prod_{i=1}^n S_i \cup \emptyset$ et chaque $s = (s_1 \dots s_n) \in \gamma(G)$ est appelé une solution de G.

A partir d'un jeu sous forme normale, pour avoir une fonction caractéristique du jeu il est nécessaire de faire des hypothèses sur les stratégies des autres agents. Tohmé et Sandholm proposent de faire l'α-hypothèse de Aumann qui dit que les agents qui ne sont pas membres d'une coalition donnée choisissent la stratégie la plus défavorable pour celle-ci. A partir de là, il est possible de définir :

Définition 54 (Fonction caractéristique) *Soit un jeu sous forme normale G pour lequel a été posé l'α-hypothèse de Aumann, alors il existe $v_\alpha : 2^I \longmapsto 2^{\mathbb{R}^I}$ la* fonction caractéristique *de G telle que pour chaque coalition $C \subset N$, $v_\alpha(C) \subset \mathbb{R}^I$ et $v_\alpha(C)$ est l'ensemble des utilités réalisables optimales pour C.*

α-coeur et superadditivité

L'α-hypothèse donne naissance à un concept de solution α-cœur qui définit un critère de stabilité pour la structure de coalitions. L'idée est que les profils de stratégies qui ne possèdent pas d'utilité résultante optimale ne sont pas candidats pour une solution. Un vecteur de stratégies jointes est bloqué par une coalition si ses membres ont intérêt à choisir un autre vecteur.

Définition 55 (Blocage) *Une coalition C* bloque *un profil de stratégie $s = (s_1 \ldots s_n)$ si pour chaque $a_i \in C$, $\exists\, s' \in \prod_{i=1}^{n} S_i$ tel que :*
- $\forall\, a_i$, $u_{a_i}(s') \geq u_{a_i}(s)$ *et* $\exists\, a_j$, $u_{a_j}(s') > u_{a_j}(s)$
- $(\ldots, u_{a_1}(s'), \ldots, u_{a_2}(s'), \ldots, u_{a_{|T|}}(s'), \ldots) \in v_\alpha(C)$

Cette relation permet de définir un ensemble de stratégies stables (non mutuellement bloquantes) : le α-cœur. Un profil de stratégie $s = (s_1 \ldots s_n)$ appartient au α-cœur $\gamma_{\mathcal{C}_\alpha}$ s'il n'existe aucune coalition C qui bloque s. Pour certains jeux comme par exemple une instance du dilemme du prisonnier [11], $\gamma_{\mathcal{C}_\alpha}$ peut être vide.

En fait, le α-cœur est non vide si et seulement si le jeu est superadditif. Dans tous les autres jeux, il existe au moins une coalition qui bloque la solution. Si ceci peut permettre de critiquer le concept d'α-cœur, on peut remarquer qu'avec des agents compétitifs et des utilités non-transférables, tous les jeux de type marché sont superadditif. L'économie de marché est un exemple typique de ce type de jeu, tout comme l'économie électronique en est un des domaines d'application.

Une autre propriété importante de l'α-cœur est le fait que toute stratégie de celui-ci est un équilibre fort de Nash et par conséquent un équilibre de Nash.

Les auteurs définissent un processus de communication comme étant une séquence $\{a_i^0 \ldots a_i^{t_i}\}$ d'ensembles d'activités de négociations. Chacun de ces ensembles correspond aux activités de négociations menés par l'agent a_i pour choisir sa stratégie et à chacune de ces activités est associé un coût $C_i(a_i^{t_j})$. Un processus de communication converge vers une structure de coalitions stable si aucune des stratégies a^t ne peut être bloquée par une coalition formée avec un autre processus.

Tohmé et Sandholm prouvent alors que si un processus de communication se trouve dans l'α-cœur alors ce processus converge vers une structure de coalitions stable.

5.3.2 Pareto-optimalité sans agrégation de préférences

Jusqu'à présent les modèles de formation de coalitions font l'hypothèse que les fonctions d'utilité de chaque agent peuvent être comparées ou agrégées en une utilité commune. Cela est souvent valable pour les utilités basées sur le profit mais dans les autres cas, cela n'est pas aussi évident. De plus, le besoin de recalculer toutes les coalitions lorsqu'il se produit un changement de conditions rend ces protocoles complexes. Caillou, Aknine et Pinson dans [5] et plus tard dans [1] proposent un modèle qui ne fait pas l'hypothèse d'agrégation ou de comparaison des utilités et permet une restructuration dynamique des coalitions en cas de changement. Les hypothèses sont les suivantes :

- Les agents du système sont coopératifs.
- Chaque agent possède une fonction d'utilité ordinale connue de lui seul.

Situation de référence

Les agents doivent pouvoir comparer les structures de coalitions entre elles avec comme critère ce qu'ils sont certains d'obtenir durant la négociation. Ce minimum est la *situation de référence*, notée SR. Si aucune coalition n'est encore formée, cette situation est celle des coalitions singletons. S'il y a déjà des coalitions existantes, c'est la situation de référence. Ce référant doit être le même pour tous les agents.

Algorithme

L'idée est d'atteindre un optimum de Pareto sans agréger, ni comparer les utilités. N'importe quel agent peut débuter une négociation, ce qui arrive lorsque de nouvelles tâches apparaissent ou qu'un agent modifie ses préférences. L'initiateur fait une annonce publique du début d'une négociation mais tous doivent attendre que la négociation précédente termine. Pour éviter les conflits, on demande l'envoi d'une confirmation. L'initiateur demande à recevoir toutes les tâches pouvant être réalisées par les agents et construit les ensembles de coalitions possibles, les classe en groupes de préférences et les envoie à l'agent suivant.

Lorsqu'un agent reçoit un groupe, il trie les ensembles en nouveaux groupes homogènes dont les ensembles au sein d'un même groupes sont équivalent en terme d'utilité, éliminant les ensembles néfastes. S'il n'est pas le dernier de la chaîne de négociation, il envoie les groupes par ordre décroissant de préférence à l'agent suivant.

S'il est le dernier, il considère que tous les ensembles du meilleur groupe sont des optima de Pareto. La solution est tirée aléatoirement parmis les ensembles équivalents. Une fois que le dernier agent a identifié un optimum de Pareto, il le transmet à tous les autres agents qui l'accepte comme étant une solution à la négociation.

- Si a_i est l'agent initiateur de la négociation :
  1. a_i attend de recevoir les listes de tâches que peuvent exécuter les agents.
  2. a_i construit toutes les structures de coalitions possibles en une liste L.
  3. a_i supprime de L toute structure néfaste pour lui au sens de SR.
  4. a_i découpe L en sous-listes L_k par classe de préférences.
  5. a_i transmet un par un ces groupes à a_{i+1} par ordre décroissant de préférence.
- Si a_i est le dernier agent de la file de négociation :

1. a_i transmet à l'agent initiateur la liste des tâches qu'il peut exécuter.
2. a_i attend de recevoir une liste L.
3. a_i choisit la structure de L qu'il préfère et la retourne comme solution.

– Sinon a_i exécute :

1. a_i transmet à l'agent initiateur la liste des tâches qu'il peut exécuter.
2. a_i attend de recevoir une liste L.
3. a_i supprime de L toute structure néfaste pour lui au sens de SR.
4. Si $L = \emptyset$, la liste est supprimée et a_i attend un nouvelle liste L'.
5. Sinon a_i redécoupe L en groupes L_k par classe de préférence.
6. a_i transmet L_k à l'agent a_{i+1} une à une par ordre décroissant de préférence.

Comportement des agents

Le fait de laisser à l'agent initiateur le soin de calculer toutes les structures de coalitions est d'une complexité exponentielle. On peut réduire cette dernière en assignant des heuristiques aux agents négociateurs.

Coalitions non développées : on peut diminuer la complexité de l'algorithme en transmettant des coalitions non développée, c'est-à-dire uniquement la tâche. Si un agent reçoit un groupe dans lequel se trouve une telle coalition, il l'écarte s'il sait qu'il ne peut pas y participer, sinon il la calcule. Cela permet de réduire le nombre de calcul de chaque agent et de les répartir sur l'ensemble du système.

Acceptabilité intermédiaire : lorsqu'un agent reçoit un groupe avec une coalition non développée, il y a une probabilité p pour qu'elle ne soit pas préférée à la situation référente. Si p est trop élevé, l'agent rejette automatiquement la coalition et ne prend pas la peine de la calculer.

Extension du modèle

On remarquera cependant que la structure finale générée par ce modèle de négociation (CGS ou pas) diffère suivant l'agent qui l'initie. C'est pour cela que le modèle est rafiné dans [1]. Cette fois, chaque agent calcule son propre ensemble de structures de coalitions et les ordonne suivant ses propres préférences. L'agent initiateur envoie son premier groupe. S'il est néfaste pour le récepteur, ce dernier le détruit, préviens les autres et envoie à son tour son premier groupe. Lorsqu'un agent, reçoit un groupe non pas néfaste mais sous-optimal pour lui, il le met en attente et relance une négociation avec son groupe préféré. Lorsque tous les agents ont accepté une même structure, la négociation se termine.

Exemple 14 *Soient deux agents a_1 et a_2. a_1 initie la négociation avec l'ordonnancement de préférence suivant $\{S_1\} \succeq \{S_2, S_3\} \succeq \{S4\}$ et considère $\{S_5\}$ comme néfaste. L'agent a_2 lui considère $\{S_1\}$ comme néfaste et ordonne le reste en $\{S_4\} \succeq \{S_2, S_5\} \succeq \{S3\}$. a_1 transmet $\{S_1\}$ à a_2 qui le rejette car néfaste. a_2 relance alors la négociation et transmet $\{S_4\}$ à a_1. Comme cette dernière est acceptable mais sous-optimale pour*

a_1, elle est mise en attente et a_1 envoie $\{S_2, S_3\}$. a_2 découpe le groupe en deux structures et comme S_2 fait partie de son groupe le plus préféré (car $\{S_4\}$ a été rejeté), il l'accepte. Puisque tous les agents ont accepté, la négociation se termine et la structure choisie est S_2.

Adjonction au CGS

Caillou, Aknine et Pinson font l'hypothèse que les agents calculent l'ensemble de toutes les structures de coalitions avant de les ordonner par préférence. Un CGS utilisant la sémantique d'extension permet de générer toutes les structures sans-conflits et de ne pas en disqualifier. On peut par la suite y appliquer la relation de préférences pour construire les groupes. Les agents qui reçoivent un groupe de structures de coalitions peuvent par leur CGS vérifier leur validité et les ordonner par préférence.

5.3.3 Modèles de préférences

Aknine, Pinson et Shakun dans [2, 3] proposent de ne se travailler que sur les préférences de l'agent, considérant qu'un agent ne peut s'intégrer à une coalition que s'il est accepté par ses membres, c'est à dire qu'il le préfère aux autres. L'utilisation d'un fonction d'utilité globale étant assez malaisée et tendant vers une centralisation du problème, une intégrale de Choquet est alors utilisée pour agréger les préférences des membres d'une coalition. Les hypothèses suivantes sont données :

- Les tâches ont été décomposées en sous-tâches.
- Les agents ont une connaissance des autres agents du système.
- Les coalitions sont disjointes.
- De nouvelles tâches peuvent être ajoutées dynamiquement.
- Les tâches peuvent devenir inter-dépendantes durant leur exécution.
- Les connaissances d'un agent sont exactes.
- Les communications ne sont pas distribuées.

Modèles de préférences

Chaque agent du système possède un vecteur de préférence pour les autres agents, noté $\Pi_{a_i} = < x_{ij} >$ où x_{ij} est la préférence de l'agent a_i pour a_j. Parfois, utiliser un seul critère pour construire des préférences n'est pas souhaitable. On définit alors $\Pi_{a_i} = (y^k_{ij})$ la matrice des préférences de l'agent a_i pour les autres par rapport à un critère D^k où $k = 1 \ldots p$. Chaque colonne correspond à un agent, chaque ligne à un critère.

Chaque coalition C possède un modèle de préférence Π_C qui est l'agrégation des vecteurs de préférence des agents qui la composent.

Définition 56 (Préférences) *Il y a deux types de préférences :*

- *La* préférence de la coalition C_i *pour un agent* $a_k \in N$ *est l'agrégation par intégrale de Choquet des préférences des* $a_j \in N_i$ *pour* a_k*. On la note* Π_{ik} *où* $\mu(N_i)$ *est le poids des sous-ensembles d'agents de la coalition.*

– *La* préférence de l'agent $a_k \in N$ *pour une coalition* C_i*, notée* $\Re(a_k, C_i)$*, est telle que* a_k *ordonne ses préférences pour chaque membre de* C_i *par ordre décroissant et calcule son intégrale de Choquet en utilisant* $\mu(N_i)$.

Formation des coalitions

Lorsque les agents vont essayer de trouver des coalitions, ils vont rechercher dans l'ensemble des partenaires possibles ceux qu'ils préfèrent aux autres mais aussi qui les préfèrent. Plusieurs techniques peuvent accélérer la recherche notamment celle de définir des classes d'équivalence pour les modèles de préférence. Les agents qui possèdent des modèles équivalent vont mutuellement se proposer une coalition.

Définition 57 (Equivalence de préférences) *Deux agents* a_i *et* a_j *ont des modèles de préférences* équivalents *si pour tout agent* a_k *tel que* $i \neq j \neq k$*,* a_i *et* a_j *ont le même degré de préférence pour* a_k *et* a_i *et* a_j *ont un degré de préférence identique l'un pour l'autre. On peut définir cette propriété d'une manière identique pour les coalitions. On note respectivement* $\Pi(a_i) \leftrightarrow \Pi(a_k)$ *et* $\Pi_i \leftrightarrow \Pi_j$.

Systèmes multi-agents coopératifs

1. Tous les agents débutent au sein de leur coalition singleton C.
2. Les agents échangent leurs modèles de préférence.
3. $\forall a_i \in N$ par ordre décroissant d'attraction unilatérale, a_i fait :
   (a) Si a_i a reçu une proposition, il l'accepte.
   (b) Si C ne permet pas de réaliser la tâche ou n'a pas la bonne cardinalité :
      i. Si $\exists\, \Pi(a_i) \leftrightarrow \Pi(a_k)$ alors a_i contacte les agents a_k.
      ii. Sinon a_i contacte a_k tel que $\arg\max_{a_k \in N} BAttraction(a_k, C)$.
      iii. Attend confirmation des agents a_k.
   (c) Lorsque tous les agents contactés ont répondu, a_i quitte le processus.
4. Tous les agents directement ou indirectement contacté par une coalition C la rejoignent.

Chapitre 6

Conclusion

L'idée derrière le problème de formation des coalitions est de former des groupes d'agents capables d'effectuer plus efficacement différents services ou tâches. Le processus de formation de coalition suit trois étapes :

1. Chaque agent construit ses structures de coalitions, c'est-à-dire l'affectation des tâches aux groupes d'agents.
2. Il les discute avec les autres agents afin de conclure un accord sur la structure qui sera adoptée.
3. Les agents de chaque coalition partagent les gains entre eux.

Plusieurs propositions [3, 8, 12, 9, 10, 14, 15] ont été présentées dans la littérature. Le but de ces travaux est de proposer des algorithmes efficaces pour calculer les structures de coalitions de chaque agent ainsi que des façons pour les agents d'atteindre une solution optimale lors d'une négociation courte.

Tous ces travaux ont en commun le problème de la formation de coalition, cependant, chacun d'eux étudie généralement une application particulière.

Inspiré des travaux sur la théorie d'argumentation, nous proposons un cadre *unifi é*, *général* et *abstrait* qui permet de générer élégamment les structures de coalitions. Le cadre formel possède trois composants : un ensemble de coalitions, une relation de contrariété entre les coalitions, et une relation de préférence entre les coalitions. Dans ce cadre, la notion de coalition demeure une entité abstraite . La définition exacte d'une coalition dépend largement de l'application étudiée. Ce peut être n'importe quel sous-ensemble de $\mathcal{N}$ (l'ensemble des agents), et dans ce cas-ci l'ensemble des coalitions sera exactement $2^{\mathcal{N}}$. Cependant, dans une autre application la définition de la coalition est plus précise. Dans le cas de l'allocation de tâches, une coalition est exactement un ensemble d'agents capables de réaliser ensemble une tâche donnée (ou un ensemble de tâches).

La notion de contrariété, elle, est induite et définie à partir des contraintes de l'application. En dernier lieu, la relation de préférence vient des valeurs que les agents peuvent assigner à chaque coalition. La valeur d'une coalition dépend également de l'application. Elle peut représenter les avantages, ou le coût d'une coalition.

Nous avons proposé trois sémantiques pour les structures de coalitions : la structure *basique* qui renvoie seulement une structure de coalitions, les structures *stables* et les structures *préferrées* qui peuvent, quant à elles, renvoyer plusieurs solutions à la fois. Chacune d'elles correspond à un point de vue particulier de l'agent. Cependant, ces différentes structures de coalitions peuvent ne pas être également préférées par l'agent. Pour chaque structure de coalitions, un agent peut avoir une valeur qui est la somme des valeurs des différentes coalitions dans la structure.

Nous avons montré que les solutions fournies par les trois sémantiques peuvent ne pas être optimales. Pour pallier une telle limite, nous avons défini une quatrième sémantique qui retourne les ensembles de coalitions sans-conflits maximaux pour l'inclusion.

Une autre contribution importante de ce mémoire est la théorie de la preuve proposée dans le cas d'une structure basique. Cette théorie de la preuve vise à vérifier si une coalition donnée se trouve dans la structure de coalitions ou non. C'est très important puisque les agents ne sont pas obligés de calculer toute la structure de coalitions pour savoir si une coalition est bonne pour eux ou pas. C'est particulièrement utile lors de l'étape de négociation. Quand un agent propose une coalition donnée, les autres agents peuvent facilement vérifier son acceptabilité sans calculer la structure entière.

Le cadre a été instancié dans deux applications particulières d'allocation de tâches. Une extension de ce travail devrait étudier plus profondément la manière dont les autres propositions existantes peuvent s'adapter à notre cadre, et comparer leurs résultats à ceux qui seront donnés avec lui.

Une autre extension importante de ce travail consiste à étudier la notion de relation de préférence. Dans ce mémoire, cette relation est très générale et, comme dit précédemment, elle reflète l'importance d'une coalition pour un agent. Cependant, dans les applications à agents, ces derniers sont autonomes et peuvent avoir des informations incomplètes au sujet de l'environnement et sur les autres agents du système. Ainsi, nous pouvons facilement imaginer qu'un agent puisse construire des coalitions qui sont plus ou moins «sûres» pour lui. Par ailleurs, une coalition peut satisfaire plus ou moins de buts prioritaires d'un agent. Il est alors important de tenir compte de ces facteurs en comparant des coalitions.

Comme il a déjà été dit, ce travail s'inspire des travaux en théorie de l'argumentation. Dans ce contexte particulier, plusieurs algorithmes ont été définis pour calculer les différentes sémantiques. Une extension de notre travail consisterait à adapter ces algorithmes au problème de formation de coalitions et comparer leur complexité à celle des algorithmes déjà existants.

En conclusion, il est très important d'étudier l'impact du cadre proposé sur la deuxième étape d'un problème de formation de coalitions.

Bibliographie

[1] S. Aknine and P. Caillou. Agreements without disagreements. In *Proceedings of the European Conference of Artificial Intelligence*, 2004.

[2] S. Aknine, S. Pinson, and M. Shakun. Coalition formation methods for multi-agent coordination problems. *In Group Decision and Negotiation*, 2000.

[3] S. Aknine, S. Pinson, and M. Shakun. Coalition formation problem : New multi-agent methods with preference models. In *Proceedings of the AAAI workshop on Coalition formation in dynamic multi-agent environments*, 2002.

[4] L. Amgoud and C. Cayrol. A reasoning model based on the production of acceptable arguments. *In Annals of Mathematics and Artificial Intelligence*, Vol. 34 :197–216, 2002.

[5] P. Caillou, S. Aknine, and S. Pinson. A multi-agent method for forming and dynamic restructuring of pareto optimal coalition. In *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multi-Agents Systems*, pages 1074–1061, 2002.

[6] V. D. Dang and N. Jennings. Generating coalition structures with finite bound from the optimal guarantees. In *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multi-Agents Systems*, pages 564–571, 2004.

[7] P. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *In Artificial Intelligence*, Vol. 77 :321–357, 1995.

[8] M. Klush and A. Gerber. Dynamic coalition formation among rational agents. *In IEEE Intelligent Systems*, pages 42–47, 2002.

[9] S. Kraus, O. Shehory, and G. Taase. Coalition formation with uncertain heterogeneous information. In *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multi-Agents Systems*, pages 1–8, 2003.

[10] S. Kraus, O. Shehory, and G. Taase. The advantages of compromising in coalition formation with incomplete information. In *Proceedings of the 3nd International Joint Conference on Autonomous Agents and Multi-Agents Systems*, pages 588–595, 2004.

[11] D. Kreps. *Game theory and economic modelling*. Oxford University Press, 1990.

[12] T. Sandholm, K. Larson, M. Andersson, O. Shehory, and F. Tohmé. Coalition structure generation with worst case guarantees. *In Artificial Intelligence Journal*, Vol. 111(1-2) :209–238, 1999.

[13] T. Sandholm and V. Lesser. Coalitions among computationally bounded agents. *In Artificial Intelligence*, pages 99–137, 1997.

[14] O. Shehory and S. Kraus. Task allocation via coalition formation among autonomous agents. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 655–661, 1995.

[15] O. Shehory and S. Kraus. Methods for task allocation via agent coalition formation. *In Artificial Intelligence*, Vol. 101(1-2) :165–200, 1998.

[16] O. Shehory and S. Kraus. Feasible formation of coalitions among autonomous agents in non-super-additive environments. *In Computational Intelligence*, Vol. 15(3) :218–251, 1999.

[17] F. Tohmé and T. Sandholm. Coalition formation processes with belief revision among bounded-rational self-interested agents. *In Journal of Logic and Computation*, Vol. 9(6) :793–815, 1999.