

Concepts et modèles de recherche de solutions optimales au problème de formation et de restructuration dynamique de coalitions d'agents

Philippe Caillou^{*}, Samir Aknine[§], Suzanne Pinson^{*}

Résumé

Cet article présente un protocole de formation de coalitions pour des systèmes multi-agents qui permet d'obtenir une solution optimale au sens de Pareto et ce sans nécessiter d'agrégation de préférences entre les agents. Il propose également une extension de ce protocole permettant une restructuration dynamique des coalitions déjà formées en fonction de changements dans l'état du monde. Il donne enfin des modèles de comportements d'agents bien adaptés à ce protocole. Une application pour la formation d'emplois du temps en fonction des préférences des étudiants et des professeurs dans un milieu universitaire a été réalisée pour le tester. Les résultats sont présentés et montrent le bon fonctionnement du protocole.

Mots-clefs : système multi-agents, coalitions, agrégation, préférences, restructuration.

Abstract

This article presents a coalition formation method for multi-agent systems that finds a Pareto optimal solution without aggregating the preferences of the agents. This protocol is adapted to problems requiring coordination by coalition formation, where it is undesirable, or not possible, to aggregate the preferences of the agents. This method has been extended to enable dynamic restructuring of coalitions when changes occur in the system. Then it presents the behaviours of the agents

^{*}LAMSADE, Université Paris-Dauphine, {caillou,pinson}@lamsade.dauphine.fr

[§]LIP6, aknine@lip6.fr

teaching schedule at university using the utility function of the professors and the students. To evaluate these methods, we have implemented this application and presented the obtained results.

Key words: multi-agent system, coalitions, preferences, aggregation, reorganization

1 Introduction

La programmation orientée agent et les systèmes multi-agents sont devenus une solution à des problèmes réels et difficiles. Les programmes sont de plus en plus complexes et disposent de fonctions multiples qui doivent parfois être mises à jour ou complétées. Utiliser un ensemble d'agents spécialisés qui se coordonnent pour réaliser des tâches complexes permet d'obtenir la flexibilité, l'efficacité et l'évolutivité nécessaires aux programmes actuels.

Un agent possède des compétences particulières. Il est programmé pour effectuer au mieux certaines tâches. La recherche de compétences plus pointues conduit à un nombre d'agents nécessaires plus important. Ces agents étant de plus en plus spécialisés, ils ne sont plus capables d'effectuer des tâches seuls et doivent obligatoirement se coordonner entre eux. Plus généralement, un système multi-agents est composé de plusieurs agents identiques ou non qui communiquent entre eux. Pour réaliser des tâches complexes, les agents ont besoin de se coordonner, soit parce que la tâche demande trop de ressources pour qu'un agent l'effectue seul, soit parce que certaines sous-tâches seront effectuées plus efficacement par un agent spécialisé.

Comment coordonner les agents de façon efficace ? Une solution est de rechercher les agents qui forment les groupes les plus appropriés pour réaliser les tâches voulues. Autrement dit, on peut chercher à former des coalitions d'agents. Une coalition se définit comme une alliance temporaire entre des agents pour réaliser un projet commun. Le but ici est bien de mettre en commun des compétences pour réaliser un projet complexe. Ce n'est pas la seule solution de coordination. Celle-ci peut être imposée par une hiérarchie, réalisée par des contrats bilatéraux, ...

Le choix des coalitions dépend en fait du type de problèmes. Les coalitions sont bien adaptées lorsqu'il y a de fortes externalités entre les sous-tâches (c'est-à-dire lorsque l'accomplissement ou non d'une sous-tâche modifie l'utilité liée à l'accomplissement d'une autre sous-tâche) et/ou lorsque les nombreuses interactions entre agents font que l'apport d'un agent à une coalition dépend de ceux qui s'y trouvent déjà. Un contrat bilatéral serait alors difficile à négocier. Par exemple, pour une rémunération de l'agent à l'utilité marginale qu'il apporte au projet (c'est-à-dire une rémunération égale au supplément d'utilité induit par la participation de l'agent au projet), la rémunération prévue varierait au cours des négociations en fonction de l'entrée ou de la sortie de

membres de la coalition. Une fois la formation de coalitions choisie comme méthode de coordination, le choix et la définition du protocole correspondant restent problématiques. Celui-ci est, en effet, fortement dépendant du type de problèmes étudié. Le fait que les agents aient ou non le même objectif, qu'ils se fassent ou non confiance sont des exemples de paramètres qui peuvent engendrer des protocoles totalement différents. Les protocoles existants reposent sur une hypothèse pour pouvoir former les coalitions : les fonctions d'utilité des agents, qui mesurent leur degré de satisfaction pour chaque solution proposée, doivent être agrégeables ou identiques. C'est-à-dire que les agents doivent être capables de se mettre d'accord sur une fonction d'utilité commune, soit à tous (comme dans [SHE 98]), soit à leur coalition ou alliance (comme dans [AKN 99] et [VAU 01]). Cette hypothèse peut sembler acceptable pour la plupart des systèmes, notamment pour des projets à but productif où les utilités peuvent souvent toutes se ramener à l'unité monétaire.

Toutefois, de nombreux cas existent où la comparaison des utilités des agents et, à fortiori, leur agrégation semblent délicates. La mesure numérique de l'utilité d'un agent est déjà une hypothèse forte par rapport au simple classement des choix disponibles (Pareto avait déjà montré les nombreux avantages de l'utilité ordinale sur l'utilité cardinale en économie au XIXe siècle). La comparaison de l'utilité de deux individus l'est encore plus. Pourquoi une solution, notée 8 par un agent et 6 par un autre, serait-elle préférée à celle notée respectivement 4 et 7 ? Le modèle proposé ici ne suppose pas que les utilités des agents soient agrégées, voire même comparées. Un deuxième inconvénient des protocoles actuels est qu'ils supposent que tous les traitements ayant conduit à la formation des coalitions sont refaits dès qu'une condition change (un agent entre ou sort d'une coalition, une tâche est ajoutée ou retirée, ...). Or ces protocoles sont très complexes et ces changements peuvent être très fréquents. Une utilisation des informations obtenues lors de l'application précédente du protocole, autrement dit une restructuration dynamique des coalitions déjà formées, pourrait permettre d'économiser de nombreux calculs ; c'est le second objectif poursuivi par notre modèle.

Cet article se décompose comme suit : tout d'abord, une description du cadre d'application du protocole adapté aux deux objectifs présentés est décrite en section 2. Avant de présenter le modèle, quelques définitions sont données en section 3. La méthode de formation et de restructuration dynamique de coalitions proposée est décrite en section 4. Le modèle laissant une flexibilité dans le choix du comportement des agents, des modèles de comportement permettant d'accélérer la résolution du problème sont ensuite présentés en section 5. Un exemple d'application du protocole proposé, un programme de création et de modification d'emplois du temps à partir des fonctions d'utilités des professeurs et des élèves a été conçu et est présenté en section 6. Cette application a également permis de réaliser des expérimentations pour pouvoir comparer les différents modèles et montrer leurs intérêts. Les résultats obtenus sont présentés en section 7. L'étude des nombreux types de problèmes possibles et des protocoles correspondants qui ont été proposés dans la littérature est nécessaire pour bien comprendre l'intérêt et l'apport du modèle proposé ici. La section 8 porte sur l'état de

l'art en décrivant les différents protocoles disponibles et en les situant par rapport à celui qui est proposé ici. Enfin, la section 9 permet de donner quelques perspectives de recherche.

2 Cadre d'application

Le protocole proposé ici s'applique particulièrement bien dans le cas de problèmes avec tâches complexes (pour lesquelles il y a un besoin d'avoir plusieurs agents et donc d'une coalition), dynamiques (des tâches peuvent être ajoutées, d'autres retirées ou modifiées à tout moment) avec des fonctions d'utilité des agents totalement libres. Les agents considérés ici sont coopératifs, c'est-à-dire qu'ils se font uniquement confiance dans la recherche et dans l'application des solutions. Cela ne signifie pas qu'ils sont altruistes ; ils cherchent bien à obtenir la meilleure solution pour eux, mais, pour cela, ils sont dans l'obligation de ne communiquer que des informations vraies, d'appliquer le protocole sans le modifier et de respecter les engagements qu'ils prennent. Leur fonction d'utilité est totalement libre et inconnue des autres agents. La seule contrainte est qu'elle soit fixe tout au long de la négociation. Elle n'a pas besoin d'être cardinale, une utilité ordinale suffit. Les agents doivent donc juste être capables de choisir entre deux situations (ou être indifférents).

Un bon exemple de ce type de problèmes est la gestion d'emplois du temps en milieu universitaire. Ce genre d'applications a attiré l'attention de la communauté scientifique de plusieurs disciplines, incluant la recherche opérationnelle et l'intelligence artificielle. Ce problème peut être en effet résolu de multiples façons : algorithmes génétiques, approches multicritère, recherche taboue, etc. [BRE 02] [MEI 01][STI 96][KAN 01]. Les solutions proposées pour résoudre ce problème sont centralisées. Notre approche vise à apporter une solution distribuée et à illustrer l'évolution dynamique des coalitions puisqu'il est habituel qu'un cours soit ajouté ou supprimé, ou même qu'un professeur ou un groupe d'élèves apparaisse. Les agents considérés dans cet exemple sont les professeurs et les élèves. Les agents élèves représentent en fait des groupes d'élèves homogènes qui ont une fonction d'utilité commune et suivent les mêmes cours. Il est tout à fait possible d'attribuer un agent à chaque élève et ainsi de lui permettre d'avoir sa propre fonction d'utilité mais la complexité des calculs sera alors fortement augmentée. Les cours constituent les tâches à effectuer et donc une coalition est formée pour chaque cours. La plupart des coalitions se composent ainsi de deux agents : un agent professeur et un agent élève (la présence d'un nombre plus élevé d'agents est possible, par exemple pour des cours magistraux avec plusieurs groupes d'élèves). Chaque agent (élève ou professeur) définit librement l'utilité qu'il attribue à chaque emploi du temps. Sa fonction d'utilité pouvant être ordinale, il doit juste être capable de comparer deux emplois du temps et dire lequel il préfère ou s'il est indifférent. Les paramètres qu'il prend en compte sont libres. Un professeur peut ainsi préférer le matin, refuser le lundi, préférer certaines classes, aimer la stabilité dans son emploi du temps, etc.

De façon générale, le choix d'un agent peut dépendre des membres des coalitions auxquelles il participe, mais son appréciation d'une coalition peut également varier en fonction des autres coalitions, introduisant ainsi des externalités ou un ordonnancement entre les tâches. Ainsi, si la tâche A doit être effectuée avant B, l'utilité que l'agent attribuera à B sera nulle si aucun agent ne participe à la coalition qui s'occupe de A (la tâche A n'est alors pas réalisée et B n'a aucun intérêt). Le choix d'un agent peut également dépendre de paramètres qui lui sont propres et qui varient dans le temps. Ainsi, il peut être contre le changement. Un professeur peut préférer un emploi du temps à un autre parce qu'il est plus proche de la situation actuelle. La seule contrainte est que ces paramètres extérieurs soient stables pendant la durée d'une négociation.

3 Définitions

Dans cette section, nous présentons quelques définitions nécessaires à la bonne compréhension de notre modèle et que nous réutilisons dans la suite de cet article.

Coalition. Une coalition est formée pour chaque tâche. Elle regroupe aucun, un ou plusieurs agents qui vont effectuer des actions pour réaliser la tâche. Pour chaque agent, une action et des paramètres sont définis (par exemple, action de donner un cours avec comme paramètres la semaine, le jour et l'heure).

Ensemble de coalitions. Un ensemble représente une solution au problème de formation de coalitions. Il est composé d'autant de coalitions qu'il y a de tâches à réaliser dans le système à un instant donné (dans l'exemple proposé, il s'agit d'un emploi du temps).

Groupe d'ensembles de coalitions. Un groupe d'ensembles de coalitions correspond à plusieurs ensembles de coalitions regroupés pour pouvoir être traités ou transmis collectivement (par exemple, plusieurs emplois du temps possibles). Dans la suite de cet article, nous parlons de groupe d'ensembles ou tout simplement de groupe. Lorsqu'un agent constitue un groupe d'ensembles équivalents, cela signifie qu'il est indifférent entre tous les ensembles de coalitions qu'il met dans ce groupe (par exemple, il constitue un groupe avec tous les emplois du temps qu'il préfère aux autres mais qu'il ne peut pas départager entre eux).

Contexte. Ensemble de paramètres quelconques qui décrivent l'état du système dans lequel la formation de coalitions a été initiée. Ce contexte est important puisque, pour tout changement de ce dernier, le résultat de la négociation ne serait plus le même. C'est pour cette raison qu'on impose que ses paramètres soient stables au cours de la négociation. Il peut par exemple s'agir de la date ou d'un autre paramètre extérieur.

Fonction d'utilité. La fonction d'utilité sert à représenter les préférences de l'agent. Elle peut être ordinale ou cardinale. Si elle est cardinale, elle associe une utilité à un ensemble de coalitions et à un contexte donné. Si elle est ordinale, elle permet de comparer deux ensembles dans un contexte donné. Dans ce cas, mesurer l'utilité d'un

ensemble signifie le comparer à une situation de référence qui sera la même tout au long de la négociation.

Situation de référence. Pour que les agents sachent s'ils acceptent un ensemble de coalitions comme solution, il faut qu'ils puissent le comparer à ce qu'ils sont certains de pouvoir obtenir lors de la négociation. Ce minimum est la situation de référence. Si aucune coalition n'est encore formée, c'est la situation où personne ne fait rien. S'il y a déjà des coalitions, c'est la situation actuelle, avec éventuellement quelques changements pour prendre en compte de nouvelles informations (voir § 4.3.2). Pour que les agents obtiennent une solution négociation, la situation de référence doit être réalisable et la même pour tous les agents.

Ensemble acceptable. Un ensemble sera dit acceptable pour un agent s'il est préféré ou équivalent à la situation de référence actuelle.

Optimum de Pareto. Un optimum de Pareto est une situation où il n'est pas possible d'améliorer la situation d'un individu sans détériorer celle d'au moins un autre. Graphiquement, pour deux agents, une situation est optimale si aucune autre situation ne se situe dans le cadre en haut à droite de la situation considérée (cf. figure 1) :

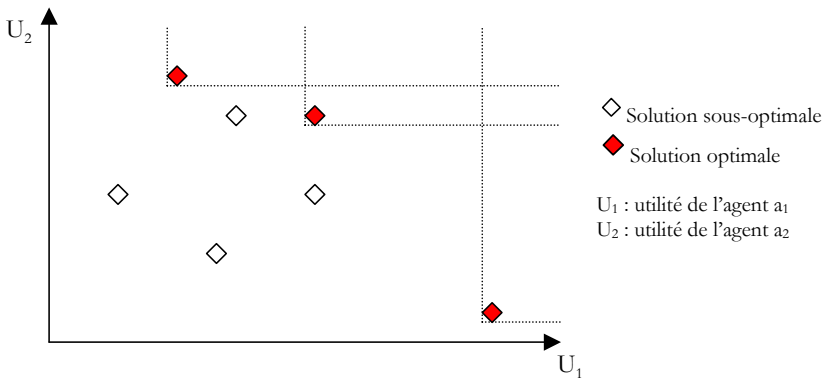


Figure 1. Exemple de situations optimales au sens de Pareto

4 Méthode de coordination

L'objectif suivi dans la définition de ce protocole a été de permettre la résolution de problèmes de formation de coalitions d'agents sans avoir à agréger leurs préférences [CAI 01]. On a cherché également à étendre ce protocole en permettant une restructuration dynamique et rapide de ces coalitions en fonction de nouvelles informations.

4.1 Présentation

Comme on ne souhaite pas agréger les utilités des agents, on va chercher une solution qui soit « objectivement bonne », c'est-à-dire qui ne puisse être contestée par aucun agent. Un critère objectif qui peut être logiquement accepté par tous est qu'on ne puisse augmenter l'utilité d'un individu sans détériorer celle d'au moins un autre. Si ce n'était pas le cas, c'est-à-dire s'il existait une situation telle qu'on puisse augmenter l'utilité d'un individu sans détériorer celle d'un autre, il n'y aurait aucune raison de ne pas préférer cette situation. La solution doit donc être un optimum de Pareto.

Quel optimum de Pareto choisir ? Le problème revient à comparer les utilités de différents agents. Comment choisir entre un emploi du temps qui est le premier choix d'un professeur et un autre qui est le premier choix d'un élève ? Une solution est de ne pas choisir et de chercher un optimum de Pareto quelconque. Cela présente l'avantage de limiter les calculs puisqu'on n'est pas obligé de les calculer tous. La seule contrainte est que chaque agent a intérêt à accepter cette solution, donc à préférer cette solution à la situation initiale. Le premier objectif de ce protocole est donc de trouver un optimum de Pareto qui soit accepté par tous les agents le plus vite possible.

4.1.1 Principe

Comment obtenir un optimum de Pareto ?

L'agent qui initialise la négociation recherche le ou les ensembles de coalitions qu'il préfère (cf. section 5). Il choisit l'agent suivant et les lui envoie (cf. § 4.2.1), puis il cherche les ensembles qu'il choisirait en deuxième choix et les lui envoie et ainsi de suite jusqu'à ce qu'il n'y ait plus d'ensemble qu'il trouve au moins aussi bon que la situation actuelle. Quand un agent reçoit un groupe d'ensembles, il les évalue et les envoie au suivant par groupes de préférences décroissantes. Lorsqu'un agent reçoit un groupe d'ensembles, qu'il y en a au moins un qui est aussi bon ou préférable à la situation actuelle et que tous les agents ont déjà participé à la négociation, l'ensemble qu'il préfère de ce groupe est un optimum de Pareto et peut être utilisé comme ensemble solution de la négociation.

Par exemple, si on considère deux agents et sept ensembles de coalitions possibles. On note $E(U_1; U_2)$ les utilités relatives des agents a_1 et a_2 pour l'ensemble E . Les sept ensembles possibles, avec E_0 qui représente la situation initiale sont: $E_0(0; 0)$; $E_1(0; 10)$; $E_2(2; 8)$; $E_3(4; 8)$; $E_4(4; 5)$; $E_5(-2; 2)$; $E_6(10; -1)$. Parmi ces sept ensembles, trois sont des optima de Pareto possibles (E_1 , E_3 et E_6). On représente ces solutions sur un plan en fonction de l'utilité qu'elles apportent à chacun des agents (cf. figure 2).

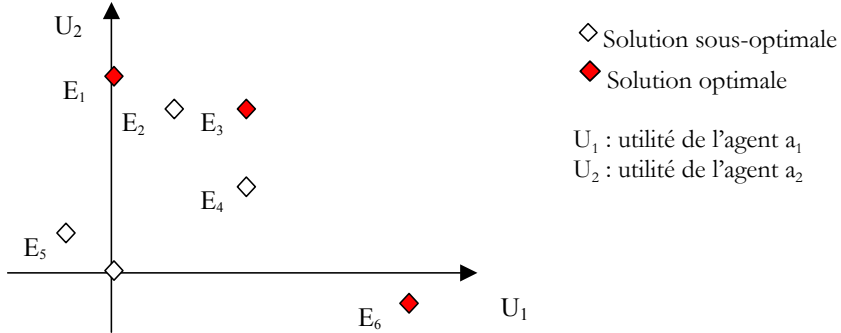


Figure 2. Représentation des solutions possibles dans l'espace des utilités des agents

L'agent a_1 initie la négociation. Il classe tous les ensembles acceptables pour lui en groupes d'ensembles équivalents (figure 3) : $G_1(E_6)$; $G_2(E_4;E_3)$; $G_3(E_2)$; $G_4(E_0;E_1)$. E_5 n'est pas classé car la situation de référence (E_0) lui est préférée :

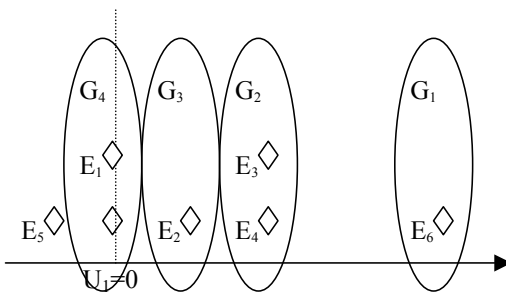


Figure 3. Classement des ensembles en groupes par l'agent a_1

Les groupes G_1 , G_2 , G_3 et G_4 sont acceptables pour l'agent a_1 car ils correspondent à une situation aussi ou plus satisfaisante que la situation initiale de référence. Il va donc envoyer dans l'ordre ces quatre groupes à l'agent suivant. Ainsi, l'agent a_2 commence par recevoir G_1 et l'évalue (cf. figure 4). Le seul ensemble de $G_1(E_6)$ est inacceptable pour l'agent car il amènerait une situation moins satisfaisante que la situation initiale. L'agent ne cherche pas à transmettre cet ensemble et attend la suite.

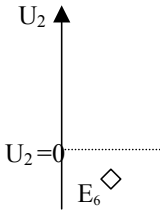


Figure 4. Classement du premier groupe reçu par l'agent a_2

L'agent reçoit ensuite G_2 et classe ses deux ensembles (figure 5) en deux groupes $G'_1(E_3)$ et $G'_2(E_4)$. G'_1 est acceptable. Comme tous les autres agents ont déjà participé à la négociation, l'agent a_2 ne peut pas le transmettre au suivant. Tous les ensembles de G'_1 peuvent donc être solution. L'agent n'a alors d'autre choix que E_3 qui est effectivement un optimum de Pareto et il transmet cet ensemble à l'agent a_1 pour lui indiquer le résultat de la négociation.

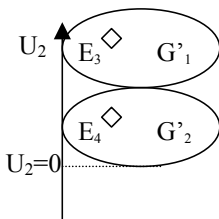


Figure 5. Classement du deuxième groupe reçu par l'agent a_2

4.1.2 L'algorithme

Chaque négociation se déroule en trois phases : initialisation de la négociation et transfert des tâches, négociation, transmission de la solution. On peut distinguer le comportement de l'agent qui initie la négociation, des agents intermédiaires et du dernier agent à participer. Il faut noter que l'ordre des agents n'est pas obligatoirement le même à chaque négociation. En effet, chaque agent peut se situer à n'importe quel

rang. En revanche l'ordre est stable durant une négociation. L'importance et l'influence de l'ordre sont discutées au § 4.2.1.

- Phase 1 : Initialisation de la négociation et transfert des tâches

Un agent doit initier la négociation. Cette initiative peut être provoquée par l'utilisateur ou par l'agent lui-même, par exemple en cas d'apparition d'une nouvelle tâche ou après une modification de ses préférences.

L'agent initiateur informe tous les autres agents qu'il débute une nouvelle négociation. Tout agent qui voudrait en débiter une autre devra attendre la fin de la négociation. Pour éviter des conflits entre deux demandes simultanées, chaque agent envoie une confirmation.

Chaque agent demande à chacun des autres agents de lui transmettre ses tâches. Il dispose ainsi de l'ensemble des tâches à effectuer et pourra associer à chacune d'elles une coalition dans chaque ensemble proposé.

L'agent initiateur calcule tous les ensembles de coalitions possibles¹. Il les rassemble dans un groupe d'ensembles et s'envoie ce groupe à lui-même (ou à l'agent qui doit débiter la phase de construction de la solution s'il existe un ordre fixe) pour débiter la négociation.

- Phase 2 : Négociation

Lorsqu'un agent reçoit un groupe d'ensembles (y compris lorsque l'agent initiateur s'envoie tous les ensembles possibles au début de la négociation), l'agent classe par ordre de préférence les ensembles reçus dans des groupes d'ensembles homogènes. Pour chaque groupe, l'agent est ainsi indifférent entre tous les ensembles du groupe. L'agent ne classe que les ensembles qu'il considère au moins équivalents à la situation de référence. Les autres ne sont pas classés dans un groupe, ils sont abandonnés. Les groupes sont envoyés à l'agent suivant par ordre décroissant (le meilleur groupe pour l'agent qui a effectué l'envoi, puis le suivant et ainsi de suite jusqu'au dernier). Si tous les autres agents ont déjà participé à la négociation (l'agent est donc le dernier), l'agent classe de la même façon les ensembles reçus en groupes. Si au moins un des ensembles reçu est acceptable, il considère le meilleur groupe. Tous les ensembles de ce groupe sont des optima de Pareto. Il peut donc en choisir un au hasard et ce sera la solution de la négociation.

¹ Pour l'amélioration de la complexité du modèle cf. § 4.1.3. et section 5.

- *Phase 3 : Transmission de la solution*

Une fois que le dernier agent a identifié un optimum de Pareto, il transmet cet ensemble à tous les autres agents qui l'acceptent comme solution de la négociation.

4.1.3 Amélioration de la complexité par utilisation de coalitions non développées

Le fait que le premier agent commence par calculer toutes les combinaisons possibles pour toutes les tâches a deux inconvénients : la complexité et le volume des données transmises aux agents suivants. Une solution pour simplifier les traitements sans changer la nature de l'information transmise, et donc le résultat et les propriétés de l'algorithme, est d'utiliser et de transmettre des coalitions « non développées », c'est-à-dire des tâches pour lesquelles toutes les coalitions possibles n'ont pas encore été calculées. Si un agent reçoit une coalition non développée dans un ensemble et que cette coalition ne peut affecter son utilité (qu'il y participe ou non), il la laisse telle quelle et ne s'en occupe pas. Si elle peut l'affecter, il calcule toutes les combinaisons possibles pour la tâche correspondante [CAI 02a]. Le résultat de ce calcul est le même quel que soit l'agent qui l'effectue.

Par exemple, pour l'application portant sur la formation d'emplois du temps, un professeur qui débute la négociation ne développera que les coalitions portant sur des cours qu'il est susceptible de donner car ce sont les seuls qui peuvent modifier son utilité.

Plus précisément, considérons le cas où l'établissement souhaite proposer un emploi du temps à un groupe d'élèves (e_1). Cet emploi du temps compte deux cours à assurer, c_1 et c_2 , pour lesquels nous avons deux professeurs disponibles, p_1 et p_2 . Les cours ne peuvent être assurés qu'à deux horaires possibles. On suppose que le cours c_1 ne peut être donné que par le professeur p_1 et que c_2 ne peut être donné que par le professeur p_2 , l'utilité des professeurs ne dépendant que de leurs propres cours. Lors de l'application de ce modèle, si les professeurs utilisent uniquement le protocole de base, i.e., sans recours aux coalitions non développées et que le professeur p_1 est le premier à commencer la négociation, il doit calculer et évaluer 9 ensembles possibles.

Ces ensembles résultent du fait qu'il y a 3 coalitions possibles pour le cours c_1 qu'il doit alors soit assurer dans le premier créneau horaire, soit dans le deuxième créneau, soit ne pas l'assurer (qui est la troisième possibilité). Chacune de ces trois configurations doit être associée aux trois autres configurations qui sont aussi envisageables pour le professeur p_2 . De la même façon, le professeur p_2 peut également assurer le cours c_2 avec le groupe e_1 soit dans le premier créneau horaire, soit dans le deuxième, soit ne pas l'assurer. Il découle de ces différentes possibilités 9 configurations puisque chacune des ses trois propres possibilités doit être examinée avec les trois autres possibilités du professeur p_2 . Le professeur p_1 doit alors les classer en groupes. Une fois ce traitement fait, il les transmet à l'agent suivant, i.e., le

professeur p_2 . Dans le cas où le premier professeur avait utilisé les coalitions non développées, il n'aurait eu que 3 ensembles à évaluer, constitués à chaque fois d'une des trois possibilités pour c_1 et ne précisant rien pour c_2 . Il les aurait aussi classés en groupes et transmis à l'agent p_2 qui se serait chargé de développer les coalitions possibles pour le cours c_2 .

Cette méthode apporte de très bons résultats si les agents ne sont affectés que par un petit nombre de tâches (c'est-à-dire s'il n'y a que quelques tâches qui peuvent modifier leur utilité). Au contraire, si un agent peut être affecté par n'importe quelle tâche, il sera obligé de développer tous les ensembles possibles lorsqu'il participera à la négociation. S'il débute la négociation, l'apport de la modification du protocole est alors nul puisque tous les ensembles possibles sont calculés dès le premier agent.

Seuls quelques changements sont nécessaires pour utiliser des coalitions non développées dans le protocole proposé. A la fin de la première phase, l'agent initiateur envoie à l'agent qui doit débiter la négociation (s'il existe un ordre fixe) un groupe d'ensembles composé d'un unique ensemble ne contenant que des coalitions non développées, ce qui lui permet de débiter la négociation. Lors de la deuxième phase, lorsqu'un agent reçoit un groupe, pour chaque ensemble de ce groupe et pour chaque coalition non développée de cet ensemble, l'agent vérifie si la tâche correspondante peut affecter son utilité. Si c'est le cas, il calcule toutes les coalitions possibles correspondant à cette tâche et ajoute autant de nouveaux ensembles de coalitions aux ensembles qu'il doit traiter et retire l'ensemble où se trouvait la coalition non développée.

4.2 Propriétés

4.2.1 Importance de l'ordre et choix de l'agent suivant

L'ordre dans lequel les agents négocient a une influence sur le résultat. Le premier agent est, en effet, favorisé pour le choix de la solution puisqu'il peut retenir en premier les ensembles qu'il préfère, parmi tous les ensembles possibles, pour les envoyer en premier au suivant. Le choix de l'agent suivant, lors de la phase de négociation, a donc une grande importance.

Une solution est de choisir au hasard parmi ceux qui n'ont pas encore participé à la négociation. Pour limiter la complexité du protocole, on peut toutefois préférer l'agent qui apparaît le plus dans les ensembles proposés. On peut supposer que, comme il participe à de nombreuses coalitions, cet agent sera plus intéressé par les alternatives qui lui sont proposées qu'un agent qui figure moins dans ces ensembles, qu'il classera donc les ensembles en plus de groupes (éventuellement il en rejetterait une bonne partie

qu'il jugerait inacceptable) et le suivant recevra alors des groupes plus petits qu'il pourra traiter plus rapidement.

Une deuxième solution consiste à choisir les agents selon un ordre d'importance préétabli. Le respect d'un ordre fonction de l'importance donnée aux agents peut ainsi permettre de favoriser ceux qui sont jugés prioritaires, ce qui peut être très pratique pour certaines applications réelles.

4.2.2 Pourquoi le résultat obtenu est-il un optimum de Pareto ?

Comment être certain que le premier ensemble satisfaisant reçu par le dernier agent participant à la négociation est un optimum de Pareto ? Tous les ensembles possibles sont loin d'avoir été évalués par tous les agents. Une démonstration est nécessaire.

Proposition 1.

Lorsqu'un agent reçoit un groupe d'ensembles, si :

- i. tous les autres agents ont déjà participé à la négociation,
- ii. un au moins des ensembles reçus est acceptable pour lui, c'est-à-dire qu'il est au moins équivalent à la situation de référence au niveau de la satisfaction,
- iii. aucun des ensembles reçus précédemment lors de la négociation ne respectait les deux conditions précédentes,

alors l'ensemble (ou les ensembles) S acceptable(s) qu'il préfère dans le groupe reçu est un optimum de Pareto et peut servir de solution à la négociation.

Démonstration.

Si cet ensemble (S) n'était pas un optimum de Pareto, cela voudrait dire qu'il existe un ensemble (S') qui est préférable pour un des agents (a_i) et au moins équivalent pour les autres. Si c'était le cas, tous les agents ayant précédé a_i dans la négociation auraient transmis S' soit dans le même groupe que S (s'ils sont indifférents), soit dans un groupe précédent (si au moins un agent préfère S' à S).

Si a_i avait reçu S' avant S , il l'aurait transmis (puisque S est solution, S est acceptable pour tous, donc S' , qui est au moins équivalent, est également acceptable pour tous). Les groupes reçus étant traités intégralement avant de passer au groupe suivant, a_i aurait renvoyé avant de traiter, et donc de renvoyer, S . Les agents suivant a_i auraient donc reçu S' avant S . Comme pour eux S' est également acceptable, ils l'auraient transmis au suivant, jusqu'au dernier agent qui l'aurait également trouvé acceptable et l'aurait donc choisi comme solution, ce qui est impossible puisque c'est S qui a été choisi.

Si a_i avait reçu S et S' dans le même groupe (tous les agents précédents les ayant jugés équivalents), il aurait transmis S' avant S puisqu'il préfère S' à S . Comme pour le

cas précédent, les agents suivant a_i auraient reçu S' avant S . Comme pour eux S' est acceptable, ils l'auraient transmis au suivant, jusqu'au dernier agent qui l'aurait également trouvé acceptable et l'aurait donc choisi comme solution, ce qui est impossible puisque c'est S qui a été choisi.

Il est donc impossible qu'il existe S' tel qu'un agent préfère S' à S et que tous les autres le trouvent au moins équivalent. Donc S est un optimum de Pareto.

4.2.3 Pourquoi est-on sûr de trouver une solution ?

Le premier optimum S trouvé est le premier ensemble qui arrive jusqu'au dernier agent et que celui-ci juge acceptable. Pourquoi y a-t-il toujours au moins un ensemble acceptable qui arrive jusqu'au dernier agent ? Une démonstration est nécessaire :

Proposition 2 :

Le protocole fournit toujours au moins une solution au problème.

Démonstration :

Le critère d'acceptabilité, pour chaque agent, est que l'ensemble soit au moins aussi satisfaisant que la situation de référence. Or, cette situation de référence est la même pour tous et fait partie des ensembles possibles. Donc tous les agents trouvent obligatoirement cette situation acceptable et vont la transmettre. Il y aura donc toujours au moins un ensemble acceptable qui arrivera jusqu'au dernier agent. Si la situation de référence est le premier ensemble à arriver, c'est un optimum de Pareto et c'est la solution au problème. Si ce n'est pas le premier, cela signifie qu'un autre ensemble acceptable est arrivé et qu'il y a donc une solution.

4.3 Restructuration dynamique des coalitions

4.3.1 Principe

Le protocole précédent permet d'obtenir une solution (un ensemble de coalitions) à partir des conditions initiales (fonctions d'utilités, ensemble de tâches et contexte). Que se passe-t-il si une de ces conditions change, par exemple si une tâche est ajoutée, enlevée ou si un agent désire modifier sa fonction d'utilité ? Dans les protocoles actuels ([SHE 98], [SAN 99], [AKN 00]), l'ensemble des calculs doit être refait pour trouver la nouvelle solution. Il peut sembler dommage de ne pas utiliser le fait qu'on ait déjà réalisé de nombreux calculs pour obtenir la solution actuelle. La nouvelle information s'ajoute aux conditions précédentes, elle ne les remplace pas complètement.

Un moyen simple d'utiliser les calculs précédents est de partir de la solution actuelle [CAI 02b]. Au lieu d'évaluer les différents ensembles par rapport à une situation initiale où chacun ne fait rien, les agents vont évaluer les nouvelles solutions par rapport à la solution actuelle. Comme elle est au moins équivalente à la situation initiale pour tous les agents (puisque c'était un optimum de Pareto), elle sera plus difficile à égaler ou à améliorer et moins d'ensembles et de groupes d'ensembles seront transmis et évalués, ce qui accélère la résolution. Le changement qui a provoqué la renégociation peut bien sûr affecter l'utilité des agents et c'est pourquoi ceux-ci doivent réévaluer les ensembles qu'ils traitent. Le gain de temps provient donc uniquement de la contrainte que constitue la nouvelle situation de référence à laquelle est associée une utilité élevée puisqu'il s'agissait de la solution précédente.

4.3.2 Précisions sur la nouvelle situation de référence

La situation de référence doit rester réalisable et identique pour tous malgré les nouvelles informations. Ce ne sera donc pas la situation actuelle qui servira de situation de référence mais la situation actuelle dont on aura retiré ce qui n'est plus possible ou ce qui a été modifié par les nouvelles informations. Par exemple, pour un agent qui se retire, la situation de référence sera l'ensemble des coalitions actuelles dont on aura retiré systématiquement l'agent en question partout où il apparaissait. Pour une tâche supprimée, ce sera l'ensemble des coalitions actuelles moins la coalition correspondant à la tâche.

4.3.3 Pourquoi la solution est-elle un optimum de Pareto ?

La démonstration de la proposition 1 (§ 4.2.2) reste valable : lorsque le dernier agent reçoit un groupe d'ensembles, qu'il n'a pas reçu auparavant d'ensemble acceptable et que le meilleur du groupe reçu est acceptable, cet ensemble S est une solution de la négociation. De plus, il ne peut exister d'ensemble qui soit au moins équivalent pour tous les agents et préférable pour au moins un d'entre eux car, sinon, le dernier agent l'aurait reçu avant de recevoir S et c'est cet ensemble qui aurait été choisi comme solution.

4.3.4 Pourquoi obtient-on toujours une solution ?

La démonstration de la proposition 2 (§ 4.2.3) reste également valable : la situation de référence est la même pour tous, acceptable par tous (puisque'elle est comparée à elle-même) et réalisable. Il existe donc au moins un ensemble (la situation de référence) qui sera transmis par tous les agents au suivant et qui sera donc la solution s'il arrive en premier au dernier agent.

5 Modèles de comportements des agents

Comment les agents font-ils pour trouver les ensembles de coalitions à envoyer à d'autres agents ? La réponse à cette question influence fortement la complexité et donc la durée d'exécution et les ressources nécessaires au bon fonctionnement du système. C'est pourquoi il convient d'analyser de façon détaillée les différentes méthodes possibles afin d'obtenir le résultat souhaité avec un minimum de calcul.

5.1 Objectif

L'objectif de chaque agent, lorsqu'il reçoit un groupe d'ensembles de l'agent précédent, est d'obtenir des groupes d'ensembles de coalitions homogènes par rapport à ses préférences, c'est-à-dire que l'agent doit être indifférent entre tous les ensembles du groupe, il doit préférer ces ensembles à tous les ensembles des groupes inférieurs et leur préférer tous les ensembles des groupes supérieurs. La méthode pour y arriver est libre et permet donc de trouver des heuristiques, éventuellement adaptées au contexte et à l'application, pour limiter la complexité de la recherche.

La solution la plus simple est que le premier agent calcule tous les ensembles possibles et qu'ensuite chaque agent se contente de classer tous les ensembles possibles de façon exhaustive. Cette solution présente l'avantage de la simplicité mais conduit à une complexité très importante, surtout pour le premier agent. D'autres méthodes de recherche peuvent à la fois limiter la complexité et la répartir entre les différents agents.

5.2 Comportements fondés sur des heuristiques pour limiter la complexité

5.2.1 Utilisation de coalitions non développées

L'utilisation de coalitions non développées (présentées au § 4.1.3) permet de limiter les calculs et le volume des informations transférées tout en gardant la simplicité des traitements pour les agents.

5.2.2 Tests intermédiaires d'acceptabilité

Une solution complémentaire pour limiter le nombre d'évaluations est de tester, avant de le développer, si un ensemble (non complètement développé) peut potentiellement être préféré à la situation de référence. Si ce n'est pas le cas, ce n'est pas nécessaire de le développer et cette branche de l'arbre du développement peut être oubliée. Ces tests sont surtout utiles lors des restructurations de coalitions. La situation de référence est alors la situation actuelle qui peut être très satisfaisante pour l'agent. Celui-ci peut écarter facilement un grand nombre d'ensembles qui n'auraient aucune chance d'aboutir après développement à une solution meilleure. C'est d'autant plus vrai

si l'agent est averse au changement, toutes les solutions qui commencent à s'éloigner de la solution actuelle sont alors rapidement écartées car l'agent leur préférera de façon certaine la situation de référence.

5.2.3 Recherche limitée au meilleur groupe

Le but d'un agent est d'envoyer au suivant les ensembles par groupes de satisfaction décroissante. Si la solution est dans le groupe G_i , tous les groupes G_j avec $j > i$ ont été évalués, classés et éventuellement développés inutilement. Il serait intéressant de n'évaluer que les ensembles de G_1 , puis ceux de G_2 et ainsi de suite. Le problème est qu'on ne sait pas à l'avance quel sera le degré de satisfaction associé au meilleur groupe. Or, pour n'évaluer que les membres de G_1 , il faut déjà connaître la satisfaction qui leur est associée, donc les avoir déjà évalués ! Même si se limiter aussi strictement aux seuls ensembles du groupe est impossible, on peut tenter de se limiter progressivement aux seuls ensembles intéressants. Pour cela, l'agent conserve une limite inférieure qui est le meilleur ensemble évalué jusqu'à présent et il ne développe que les ensembles qui peuvent au moins égaler cet ensemble limite. A chaque fois qu'un ensemble – même non complètement développé – est évalué et est supérieure à la situation limite (dans au moins un de ses développements futurs), il devient la situation limite. Lorsqu'un ensemble a été déjà évalué et ne peut atteindre cette situation mais qu'il peut quand même être acceptable dans un groupe plus faible, il est conservé et ajouté à un groupe qui servira de groupe de départ pour calculer les groupes suivants.

5.2.4 Recherche limitée avec évaluation intermédiaire

Dans le cas précédent, on remarque que l'ordre dans lequel les coalitions sont développées a une importance. En effet, plus le meilleur ensemble est atteint rapidement, plus il devient la situation de référence et moins les autres ensembles sont développés (car la situation de référence est plus rarement potentiellement atteignable). Il est donc intéressant de mettre en place une procédure d'évaluation intermédiaire des ensembles à développer pour traiter d'abord celui qui semble le plus susceptible de générer des ensembles apportant une grande satisfaction.

5.2.5 Recherche prospective

Pour aller encore plus loin dans l'utilisation de la fonction d'utilité, l'agent, au lieu de partir de l'ensemble vide et le développer, peut utiliser directement ses connaissances de sa fonction d'utilité et des tâches à accomplir pour en déduire les meilleurs ensembles. Si le nombre d'ensembles possibles est très important, cette solution peut être avantageuse dans la mesure où la complexité ne dépend pas ici du nombre d'ensembles possibles mais du type de fonction d'utilité de l'agent. Cette méthode peut donner des résultats beaucoup plus efficaces mais oblige à réécrire la procédure pour chaque type de fonction d'utilité.

5.2.6 Méthodes non exhaustives

Que ce soit à partir de la fonction d'utilité ou à partir du développement des coalitions possibles, l'agent peut faire des approximations pour avoir des résultats beaucoup plus rapides, même s'il n'est pas certain que ce soient les meilleurs possibles. Ainsi, pour la méthode du développement des coalitions, il peut choisir de ne pas explorer les ensembles – non développés – qui offrent des perspectives peu intéressantes, même s'il est théoriquement possible qu'un de leur développement donne la meilleure solution. La solution finale sera alors obtenue plus rapidement mais rien ne garantit que ce soit réellement un optimum de Pareto.

Dans les sections suivantes, nous présentons les résultats obtenus en appliquant les quatre premières heuristiques. Nous les avons présentées suivant l'ordre croissant de leur importance.

6 Implémentation et tests du modèle

Pour montrer le fonctionnement concret du modèle proposé, un système pour la construction d'emplois du temps à partir des fonctions d'utilité des professeurs et des élèves a été développé. Les fonctions d'utilité des agents étant totalement libres, de nombreux paramètres ont été introduits ici afin que professeurs et élèves puissent indiquer précisément leurs préférences. La fonction d'utilité renvoie un résultat entier. Ce résultat est toutefois relatif à la situation de référence, l'utilité de celle-ci (absolue) étant calculée au début de chaque négociation. Les différentes variables de la fonction d'utilité sont les suivantes : heure de début pour chaque journée, heure de fin, nombre d'heures par jour, nombre de cours non donnés, nombre de cours obligatoires pour l'agent non donnés, nombre de changements par rapport à l'emploi du temps actuel, nombre d'heures total au cours d'une semaine. Pour chaque agent, une série de paramètres modifiables est disponible : les formes des fonctions d'utilité partielle et les poids correspondants. Etant donné le nombre de paramètres, trois profils ont été définis pour simplifier les choix par défaut : matinal, après-midi et groupeur (préfère regrouper ses cours sur un minimum de journées). Ces profils correspondent à des valeurs de paramètres arbitraires utilisés pour les tests.

6.1 Description du système multi-agents

6.1.1 Structure du système

Le système est constitué de deux classes principales : l'agent et l'environnement. L'environnement a comme seul objectif d'identifier les nouveaux arrivants, de réaliser un suivi global du système et de permettre à l'utilisateur d'intervenir. Les agents, après avoir récupéré, lors de leur création, les adresses IP des autres agents, peuvent communiquer directement entre eux. Il s'agit d'un système multi-agents générique

permettant de redéfinir simplement un environnement et des agents adaptés au domaine auquel on veut l'appliquer, aussi bien au niveau de la définition des tâches, des actions ou du comportement des agents (maximisation de l'utilité collective, individuelle, ...). Le système est constitué de plusieurs programmes exécutables (ce sont des programmes indépendants et non des processus partageant des données). Le but est de faire que l'environnement, aussi bien que chaque agent, soit une entité à part entière.

6.1.2 Communications

Les agents et l'environnement communiquent par TCP/IP. Les différents programmes peuvent donc être exécutés sur des ordinateurs différents et communiquer en réseau local ou par Internet sans aucun changement de programmation. Cette division des ressources permet une plus grande rapidité ou un parallélisme réel entre les actions et les délibérations. Les agents s'inscrivent auprès de l'environnement dont ils doivent connaître l'adresse IP. Lors de leur inscription, l'environnement envoie aux agents les adresses des autres agents et prévient les autres de la présence des nouveaux arrivants. Tous les agents ont ainsi les adresses des autres et ils peuvent communiquer entre eux. Rappelons que l'environnement ne joue aucun rôle dans la prise des décisions des agents. Il n'est pour eux qu'un premier intermédiaire pour les faire connaître les uns aux autres au moment de leur arrivée dans le système multi-agent.

Pour communiquer, les agents et l'environnement utilisent des objets Message. Ces objets sont des vecteurs structurés de façon à être créés, envoyés et reçus. Tous les agents et l'environnement ont en permanence un processus en attente d'un nouveau message pour que celui-ci puisse être reçu immédiatement et traité plus tard.

6.1.3 Description de l'environnement

La présence d'un agent environnement, activé avant tout agent, a trois objectifs :

- (i) Permettre à l'utilisateur d'intervenir sur le système dans sa globalité
- (ii) Informer l'utilisateur sur l'état global du système.
- (iii) Enregistrer l'arrivée d'un nouvel agent, lui fournir les informations concernant l'environnement actuel (adresses des autres agents) et prévenir les autres agents de son arrivée en leur fournissant son adresse.

6.2 Description des agents

Un agent est un programme autonome qui agit dans son propre intérêt. Il est caractérisé principalement par les actions qu'il sait faire, le raisonnement qu'il est capable de construire et par sa structure de connaissances. L'agent dispose de connaissances sur les différents types de tâches qui existent dans le domaine considéré, par exemple pour construire un emploi du temps. Il connaît également différents types d'actions, par exemple celles qui lui permettent de calculer son utilité ou de répondre aux nouveaux changements de son environnement. Les types de tâches sont

indépendants de l'agent : tous les agents doivent connaître les mêmes types de tâches (même s'ils peuvent les traiter différemment) pour pouvoir communiquer. Par contre, les types d'actions ne dépendent que de l'agent. Un type d'action peut permettre de réaliser certains types de tâche. Ces connaissances sur son environnement et ses capacités permettent par exemple à l'agent de raisonner sur *d'éventuelles* actions pour répondre à une certaine tâche. En plus de ses connaissances sur lui-même, l'agent dispose de connaissances sur les autres agents. Ces connaissances lui sont transmises par les autres agents lors de leur entrée dans le système (ou lorsque l'agent y entre si les autres avaient rejoints le système avant son intégration) ou lorsqu'ils modifient une de leurs caractéristiques (comme ajouter ou retirer une matière dans la liste des compétences d'un professeur dans notre exemple d'application). Les actions qu'un agent peut accomplir sont totalement libres à son choix. Il n'est soumis à aucune influence, y compris de l'environnement. Tout comme l'environnement, l'agent dispose d'un processus qui lui permet d'attendre d'éventuelles communications des autres agents ou de l'environnement.

6.2.1 Agent négociateur

Bien que toujours réutilisable pour d'autres applications, cet agent connaît le protocole de formation de coalitions et peut donc négocier afin d'arriver à une situation optimale au sens de Pareto.

- Structure

L'agent se décompose en trois processus, tous activés lors de l'initialisation :

- (i) Un processus d'action, qui effectue les actions planifiées par l'agent.
- (ii) Un processus de négociation, qui s'active dès qu'une nouvelle négociation a lieu.
- (iii) Un processus de communication, qui attend de recevoir des messages des autres agents ou de l'environnement, et communique avec les autres processus en conséquence.

- Gestion des communications

L'agent négociateur sait gérer les négociations pour former des coalitions. Pour cela, il connaît un certain nombre de messages spécifiques : « Initialisation » et « mise à jour » sont des messages envoyés par l'utilisateur par le biais de l'environnement ou de l'agent, ou par l'agent lui même, qui provoquent l'initialisation d'une nouvelle formation de coalitions « initialisation » ou d'une nouvelle modification de coalitions « mise à jour ». L'agent commence par prévenir les autres agents de la nouvelle négociation (message « nouvelle négociation », auquel ils répondent par « confirmation nouvelle négociation »). Puis il commence la négociation et envoie progressivement les groupes d'ensembles de coalitions à l'agent suivant (message « négociation en cours » auquel l'agent répond à chaque fois qu'il a fini de traiter un groupe « fin évaluation »).

- Traitement d'une nouvelle négociation

Lorsqu'un agent reçoit un groupe d'ensembles d'un autre agent, il place ces ensembles dans un vecteur d'ensembles à analyser et active la procédure d'analyse de groupe. Lorsqu'un agent initialise une nouvelle négociation, il place dans le même vecteur un unique ensemble initial où aucune coalition n'est développée et il active la même procédure d'analyse. Initialiser une négociation revient à recevoir un seul ensemble vide depuis l'extérieur. Dans tous les cas, l'agent initialise ses groupes où il va classer les ensembles en fonction de la satisfaction qu'ils lui apportent.

- Recherche de coalitions

Pour classer les ensembles en groupes décroissants, l'agent dispose de plusieurs méthodes, décrites dans la section 5 : en premier lieu, la méthode de base, améliorée grâce à l'ajout de tests intermédiaires (cf. § 5.2.2) et en ne développant que les coalitions nécessaires (cf. § 5.2.1), puis la méthode de recherche limitée au meilleur groupe (cf. § 5.2.3) qui oblige à modifier profondément l'algorithme en passant d'une recherche en largeur à une recherche en profondeur. On doit en effet obtenir le plus vite possible des ensembles complètement développés pour pouvoir les évaluer. Au contraire, dans la méthode de base, de tels ensembles ne sont obtenus qu'à la fin, lorsqu'on développe, simultanément pour tous les ensembles intermédiaires, la dernière coalition. Enfin, la méthode se limitant au meilleur groupe avec test intermédiaire (cf. § 5.2.4) qui améliore la méthode précédente en réalisant des tests intermédiaires pour trouver plus vite un ensemble du meilleur groupe.

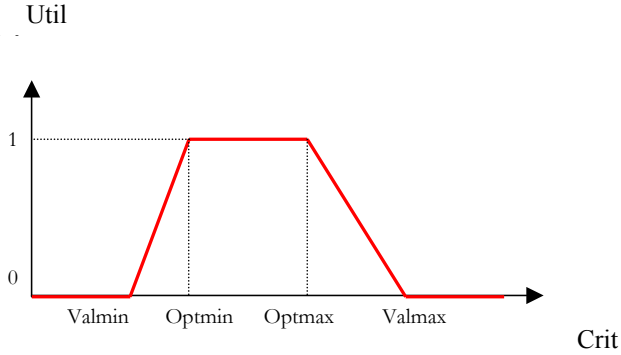
6.2.2 Les agents élèves et professeurs

Les fonctions d'utilité des agents étant totalement libres, de nombreux paramètres ont été introduits ici afin que professeurs et élèves puissent indiquer précisément leurs préférences. La fonction d'utilité renvoie un résultat entier. Ce résultat est toutefois relatif à la situation de référence, l'utilité de celle-ci (absolue, mesurée par la variable U_{comp}) étant calculée au début de chaque négociation.

Les différents paramètres de la fonction d'utilité sont les suivants :

- i. Heure de début de la journée : pour chaque jour, l'agent attribue une utilité à l'heure de début des cours $d(j,s)$.
- ii. Heure de fin de la journée : pour chaque jour, l'agent attribue une utilité à l'heure de fin des cours $f(j,s)$.
- iii. Nombre d'heures de la journée : pour chaque jour, l'agent attribue une utilité au nombre d'heures de cours $n(j,s)$.

Chacune de ces fonctions d'utilité partielle est de la forme suivante :



Valmin : valeur au-dessous de laquelle l'utilité est nulle
Optmin et Optmax : bornes entre lesquelles l'utilité est maximale
Valmax : valeur au-dessus de laquelle l'utilité est nulle
Critère : heure de début, de fin ou nombre d'heures dans la journée

Figure 6. Utilité partielle en fonction de la valeur du critère choisi

Pour chaque jour, l'agent calcule une pondération de ces trois fonctions ($d(j,s)$, $f(j,s)$, $n(j,s)$) par rapport à des coefficients qui lui sont propres (p_d , p_f , p_n) en fonction des critères auxquels il accorde le plus d'importance. L'agent fait la moyenne de ces utilités quotidiennes. Il recherche ensuite le nombre de cours nbtt qui ne sont pas donnés dans l'ensemble évalué. Il dispose d'un coefficient d'aversion c_t pour les cours non donnés. Il multiplie l'utilité intermédiaire par ce coefficient à la puissance nbtt. De même, il recherche le nombre de cours auquel il doit assister (tâches qui lui sont attribuées au départ) qui ne sont pas donnés nbtp. Il dispose d'un deuxième coefficient c_p qu'il met à la puissance nbtp et qu'il multiplie par le résultat précédent. Enfin, l'agent recherche le nombre de cours nbch qui ont été déplacés par rapport à la situation initiale dans son emploi du temps. Il dispose d'un troisième coefficient c_c qu'il met à la puissance nbch et qu'il multiplie par le résultat précédent rapporté aux nombres de jours nbj et de semaines nbs. Il dispose alors d'un résultat absolu auquel il soustrait l'utilité de la situation de référence actuelle U_{comp} pour obtenir un résultat relatif.

L'équation correspondant à ces calculs est, pour l'élève :

$$U(Elève) = \frac{c_i^{nbtu} c_p^{nbtp} c_c^{nbch} \sum_{s=1}^{nbs} \sum_{j=1}^{nbj} (pad(j,s) + p_f f(j,s) + p_{nt}(j,s))}{nbj \times nbs} - U_{comp}$$

Equation 1. Calcul de l'utilité relative d'un agent élève associé à un ensemble de coalitions

L'agent professeur dispose d'un quatrième critère : le nombre d'heures total au cours d'une semaine. Il fait la moyenne hebdomadaire de ce critère (t(s)), et il est ensuite pondéré (p_i) avec la moyenne quotidienne précédente. Son utilité est donc définie par :

$$U(Ens) = \frac{c_i^{nbtu} c_p^{nbtp} c_c^{nbch} \sum_{s=1}^{nbs} \sum_{j=1}^{nbj} (pad(j,s) + p_f f(j,s) + p_{nt}(j,s) + p_{it}(s))}{nbj \times nbs} - U_{comp}$$

Equation 2. Calcul de l'utilité relative d'un agent professeur associé à un ensemble de coalitions

Pour finir, l'agent vérifie s'il y a des impossibilités dans l'emploi du temps (deux cours à la même heure) ou des contraintes violées (heure de début avant la valeur minimum ou après la valeur maximum, ...). Si tel est le cas, il retourne la valeur -1. Sinon, il retourne l'utilité calculée. Tous ces paramètres sont modifiables individuellement pour chaque agent. Etant donné leur nombre, trois profils ont toutefois été définis pour simplifier les choix par défauts : matinal, après-midi et groupeur (préfère regrouper ses cours sur un minimum de journée).

6.3 Fonctionnement

6.3.1 Description

La première étape consiste à activer un environnement. Des agents (professeurs ou élèves) peuvent ensuite être ajoutés, soit directement à l'aide du menu correspondant à l'environnement, soit par un programme séparé, éventuellement sur un autre ordinateur sur le réseau.

Les différentes fonctions disponibles pour chaque agent sont :

- i. Redéfinition des compétences et de la fonction d'utilité. Pour les professeurs comme pour les élèves, on peut modifier tous les paramètres de la fonction d'utilité. Pour les professeurs, on peut de plus définir quelles sont les matières dans lesquelles ils sont compétents pour enseigner.
- ii. Modification de la méthode de recherche. Pour chaque agent, on peut choisir le modèle d'agent en définissant la méthode qu'il utilisera pour chercher les ensembles de coalitions.

- iii. Ajout ou suppression de cours. Pour les élèves, des cours peuvent être ajoutés dans une matière quelconque et ceux qui existent peuvent être supprimés.

Une fois tous les agents initiaux activés et leur fonction d'utilité définie, on peut provoquer l'initialisation qui permet d'obtenir un ensemble d'emplois du temps solution. Par la suite, on peut ajouter d'autres agents, d'autres tâches, modifier les fonctions d'utilité des agents. On peut ensuite lancer une mise à jour qui cherchera une nouvelle solution à partir de la situation actuelle.

7 Résultats

La programmation de l'application a permis de réaliser des tests sur les différentes solutions proposées. Pour comprendre ces tests, il convient de se rappeler les différents objectifs fixés au départ :

- i. Proposer un protocole pour former des coalitions optimales au sens de Pareto.
- ii. Proposer un protocole donnant une restructuration dynamique des coalitions.
- iii. Proposer des heuristiques de recherche des groupes de coalitions afin d'accélérer la négociation.

7.1 Formation de coalitions et optimalité du résultat

Comment évaluer un tel protocole ? On ne peut pas chercher à vérifier si la fonction d'utilité est maximale puisqu'il y a plusieurs fonctions d'utilité qui sont – par hypothèse – non comparables. On analyse les résultats du protocole en retenant plusieurs facteurs : le nombre de messages échangés, la taille de ces messages (c'est-à-dire le nombre d'ensembles de coalitions qu'il contient) et le nombre d'ensembles de coalitions évalués. Le nombre de messages échangés entre les agents est indépendant de la stratégie de recherche. Leur taille, par contre, dépend de l'utilisation ou non de coalitions non développées. Le nombre d'ensembles évalués dépend, lui, fortement de la méthode de recherche employée, la méthode de base se contentant d'évaluer systématiquement tous les ensembles possibles alors que les heuristiques proposées cherchent à limiter au maximum le nombre de ces ensembles pour obtenir le résultat plus rapidement. Quatre de ces heuristiques ont été implémentées (les quatre premières présentées § 5.2.1, 5.2.2, 5.2.3 et 5.2.4). Celle qui a donné les meilleurs résultats et qui a été utilisée dans les expérimentations décrites consiste à rechercher uniquement le meilleur groupe en réalisant des tests intermédiaires pour identifier le plus rapidement possible la valeur du meilleur groupe (cf. § 5.2.4).

Ces facteurs vont être analysés sur un exemple simple portant sur 4 agents (2 professeurs sur deux matières et 2 groupes d'élèves, chacun ayant un cours dans chaque matière, soit quatre tâches pour le système). On considère l'emploi du temps sur deux jours avec huit heures possibles par jour. On fait varier le profil de chaque agent (matinal, après-midi, groupeur) pour obtenir des résultats moyen, maximum et minimum. Les élèves disposent de 2 cours chacun à placer dans 16 créneaux horaires

(ou à ne pas placer s'il n'est pas donné, soit 16+1 cas) et avec un seul professeur possible pour chaque cours. Il y a donc 17^4 , soit 83 521 emplois du temps possibles (on considère un emploi du temps où deux cours sont à la même heure comme possible, c'est l'agent qui le jugera inacceptable). L'ordre dans lequel se déroulent les négociations est : groupe d'élèves e_1 , groupe d'élèves e_2 , professeur p_1 et professeur p_2 .

Pour compter les messages envoyés, on ne considère que les messages d'envoi de groupes (qui sont les plus nombreux et, surtout, les plus volumineux puisqu'ils contiennent les ensembles à évaluer). Le dernier agent n'envoie aucun message de ce type puisqu'il ne fait qu'attendre de recevoir un ensemble qui lui convienne pour ensuite l'envoyer aux autres agents comme solution au processus (c'est pourquoi il n'apparaît pas dans les figures 7 et 8). Le nombre de messages envoyés dépend fortement de la précision de la fonction d'utilité des agents. S'ils ont des préférences très précises, ils vont répartir les ensembles entre beaucoup de groupes de petite taille et envoyer beaucoup de messages. Les utilités des agents considérés ici ont cent niveaux. Il y a donc au maximum 100 messages envoyés par le premier agent (le deuxième peut donc en envoyer au maximum 10 000 car, pour chaque groupe reçu, il peut le diviser en 100 autres groupes). Le nombre de messages envoyés pour cet exemple est résumé figure 7.

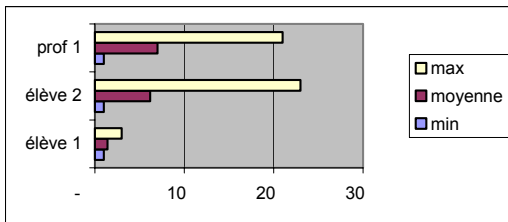


Figure 7. Nombre de messages envoyés lors d'une négociation avec 4 agents et 4 tâches

Le nombre de messages envoyés varie fortement en fonction des préférences incompatibles entre les agents. Par exemple, un élève de profil matinal va chercher à placer les cours le matin en priorité. Si le professeur a un profil après-midi, il va juger ces horaires inacceptables et l'élève devra envoyer d'autres propositions qui lui conviennent moins. Au contraire, si les premières propositions sont acceptées par tous, un seul message par agent est nécessaire (cas minimum). La taille totale des messages envoyés permet de mesurer l'encombrement du réseau. Cette taille totale doit être comparée aux 83 521 ensembles possibles.

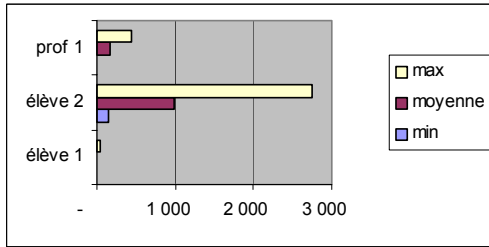


Figure 8. Taille totale des messages envoyés (mesurée en nombre d'ensembles) lors d'une négociation avec 4 agents et 4 tâches

L'agent qui envoie le plus d'ensembles et de messages est l'agent élève 2 pour deux raisons : l'agent élève 1 lui envoie les ensembles qu'il préfère en fonction de ses deux cours. Pour chacun de ces ensembles (qui n'affectent pas son utilité), l'agent élève 2 calcule toutes les combinaisons possibles de ses propres cours. Il envoie ensuite ces combinaisons par préférences décroissantes au professeur a_1 jusqu'à ce qu'il n'ait plus d'ensembles acceptables à envoyer. Ce n'est qu'alors que l'agent élève 1 lui envoie un deuxième message et que la négociation continue. Le nombre d'ensembles évalués permet de mesurer l'efficacité de l'heuristique de recherche du meilleur groupe. Si la méthode de base était employée, le premier agent évaluerait simplement les 83 521 ensembles possibles et les enverrait classés au suivant.

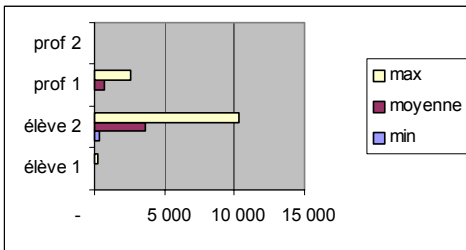


Figure 9. Nombre d'ensembles de coalitions évalués par les agents lors d'une négociation entre 4 agents avec 4 tâches

7.2 Restructuration dynamique des coalitions

La restructuration dynamique des coalitions a pour but de donner un résultat aussi satisfaisant que le protocole de base mais de façon plus rapide car on utilise de l'information tirée de la négociation précédente en prenant la solution précédente comme nouvelle situation de référence. Le résultat ne sera pas obligatoirement (et même probablement pas) le même que si le protocole initial était appliqué mais le

résultat est toujours un optimum de Pareto. On peut étudier l'effet de l'ajout de nouveaux cours à la situation précédente en termes de nombre d'ensembles évalués et transmis. On ajoute progressivement 4 cours aux agents élèves 1 et 2 (en commençant par le premier). La taille moyenne des messages envoyés lors de ces ajouts est indiquée figure 10.

La première négociation (4 cours) utilise le protocole d'initialisation alors que les quatre autres sont des restructurations à partir de la situation précédente. Le nombre d'ensembles envoyés doit être comparé au nombre total d'ensembles possibles, variant de 80 000 pour 4 cours à près de 7 milliards pour 8 cours. En pourcentage, on obtient les résultats de la figure 12 (on considère les ensembles évalués et non les ensembles envoyés).

Le nombre d'ensembles envoyés, de même que le nombre d'ensembles évalués, n'est ici plus fonction du nombre total de tâches à effectuer mais de deux facteurs : le nombre de tâches que les agents n'arrivent pas à réaliser (du fait de préférences incompatibles) et le nombre de nouvelles tâches. L'effet est cumulatif, ce qui explique que le nombre d'ensembles envoyés augmente progressivement. Par exemple, si le 6^{ème} cours n'a pas pu être affecté, cela affecterait le nombre d'ensembles envoyés après l'ajout des 7^{ème} et 8^{ème} cours car l'agent 1 essaierait à nouveau à chaque fois de placer ce 6^{ème} cours (ce qui n'est pas absurde puisqu'il pourrait y avoir une chance qu'il y arrive plus lors des négociations suivantes où de nouveaux cours peuvent modifier les utilités des agents).

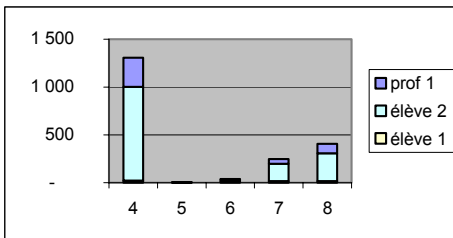


Figure 10. Taille totale moyenne des messages envoyés (mesuré en nombre d'ensembles) avec 4 agents et un nombre de cours variant de 4 à 8

7.3 Comparaison des modèles d'agent

Le nombre d'ensembles évalués est ici très faible par rapport au nombre d'ensembles possibles (en moyenne 1% pour l'initialisation précédente et entre $4.10^{-4}\%$ et $6.10^{-6}\%$ pour les restructurations). Toutefois, l'utilisation du modèle d'agent de base aurait conduit au calcul et à l'évaluation de la totalité des ensembles possibles avant d'envoyer les groupes d'ensembles acceptables. Le choix d'une bonne heuristique de recherche du meilleur groupe est donc fondamental pour que le temps de recherche soit acceptable.

Quatre des heuristiques proposées ont été implémentées ; celle qui a donné les meilleurs résultats est décrite § 5.2.4 et consiste à rechercher uniquement le meilleur groupe en réalisant des tests intermédiaires pour identifier le plus rapidement possible la valeur du meilleur groupe. L'utilisation de ces heuristiques a permis d'obtenir le nombre d'évaluations suivant (cf. figure 11) lors des restructurations effectuées précédemment.

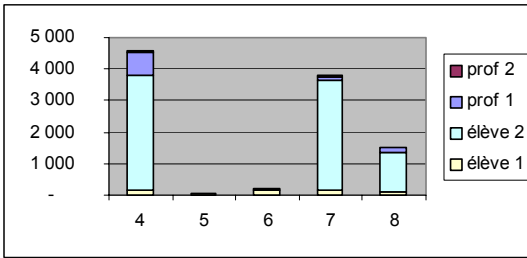


Figure 11. Nombre d'ensembles de coalitions évalués par 4 agents pour un nombre de cours variant de 4 à 8

Rapporté au nombre total d'ensembles possibles, on obtient les résultats présentés figure 12.

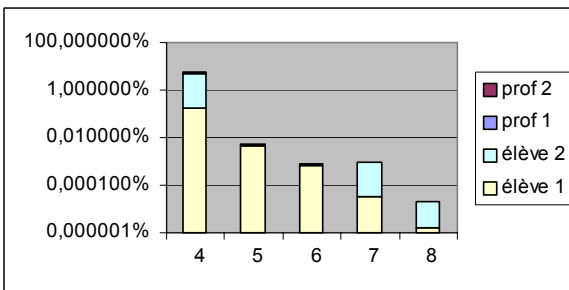


Figure 12. Proportion d'ensembles de coalitions évalués par 4 agents pour un nombre de cours variant de 4 à 8 (échelle logarithmique)

La proportion d'ensembles évalués par rapport au nombre total est ainsi d'un peu plus de 1% (soit 4 500 ensembles) pour la formation de coalitions, puis diminuée à moins de $10^{-2}\%$ (à peu près 50) pour la première restructuration jusqu'à atteindre $10^{-5}\%$ (à peu près 1 500 ensembles) pour la dernière.

8 Travaux sur le problème de formation de coalitions

Les protocoles de formation de coalitions se sont très largement inspirés des travaux réalisés en théorie des jeux (voir notamment [KAH 84]). Ce domaine a fourni la majorité du vocabulaire et des concepts utilisés en SMA pour l'analyse de ce problème (typologie des problèmes, des solutions, des équilibres, des fonctions, ...). A travers les indices de pouvoir, elle permet de calculer l'influence réelle d'un agent dans une coalition ([GAM 94]). Elle fournit des méthodes de calcul pour définir les meilleures coalitions dans les différents types de problèmes. Son application aux systèmes multi-agents a été en particulier étudiée par Sandholm [SAN 96]. La limite de l'utilisation qui peut en être faite provient des hypothèses sous-jacentes (les agents sont le plus souvent considérés comme parfaitement rationnels) et de l'objectif poursuivi (la théorie des jeux se préoccupe le plus souvent uniquement de la valeur de la solution optimale et non de la méthode la plus efficace pour y parvenir, encore moins de la méthode distribuée la plus efficace).

Les protocoles disponibles actuellement en système multi-agents se fondent tous sur la décomposition du problème suivante : génération des coalitions, résolution du problème d'optimisation dans chaque coalition et répartition de la valeur ainsi créée entre les agents. Ils s'intéressent au premier de ces trois problèmes. [SHE 98] propose un protocole simple et efficace s'appliquant dans des cas très généraux (recouvrement de coalitions, ordonnancement) et permettant de trouver la meilleure solution. Il suppose toutefois qu'une fonction d'utilité commune soit disponible et le protocole implique que la valeur de l'ensemble des coalitions possibles soit calculée au moins une fois, d'où une importante complexité pour arriver au meilleur résultat. [SAN 99] traite ce problème en proposant une méthode à la complexité limitée tout en assurant un résultat minimum (en proportion du résultat optimal). [SAN 97] présente une analyse des problèmes de limite en capacité de calcul et propose une terminologie adaptée à ce type de problème.

[AKN 00] et [VAU 01] s'intéressent à des problèmes où les agents ont leur propre fonction d'utilité et où une agrégation n'est nécessaire qu'à l'intérieur d'une coalition ([AKN 00]) ou d'une alliance([VAU 01]). [AKN 00] utilise l'intégrale de Choquet pour réaliser des agrégations multicritères entre des agents qui peuvent être coopératifs ou compétitifs. De plus, le protocole ne suppose pas que tous les agents se connaissent. Le protocole se limite toutefois au cas où les coalitions sont disjointes. [VAU 01] étudie le cas de coalitions non disjointes formées progressivement par formation d'alliances et adaptation progressive des préférences des agents (qui ont intérêt à s'adapter pour ne pas risquer d'être exclus de la solution, et donc du partage final). Néanmoins, ce processus implique des traitements coûteux puisque les agents commencent par calculer, chacun de son côté, toutes les solutions possibles et échangent leurs préférences. Ces préférences sont ensuite remaniées en fonction de celles des autres. Le processus s'arrête dès que tous les agents convergent vers une solution commune. S'ils se heurtent à un bouclage, ils recourent aux alliances qui représentent des groupes de faibles tailles.

Le protocole proposé ici peut difficilement être comparé à ceux qui ont été présentés car il n'a pas le même objectif. Dans les protocoles existants, les fonctions d'utilité sont systématiquement agrégées ou adaptées. Au contraire, les utilités ne sont ici ni agrégées ni même transmises. Les résultats ne peuvent donc être comparés car ils portent sur des problèmes différents. Cependant, si tous les agents ont la même fonction d'utilité au départ, le protocole proposé devra obtenir le même résultat que celui de [SHE 98] ou [AKN 00].

9 Conclusion et perspectives de recherche

Le protocole proposé est adapté à des problèmes nécessitant une coordination par formation de coalitions et où il n'est pas souhaitable, ou possible, d'agréger les préférences des agents. Le protocole permet alors d'obtenir une solution optimale au sens de Pareto. Dans le cas où il y a un changement dans les données du problème, il calcule rapidement une nouvelle solution, toujours optimale, en partant de la solution actuelle.

Pour le type de problèmes considérés (formation et restructuration de coalitions sans agrégation des préférences), il a été démontré que le protocole proposé permet d'obtenir une solution qui est optimale au sens de Pareto. Les tests ont de plus montré que la complexité moyenne restait faible comparativement au nombre total de cas possibles.

Malgré ces résultats encourageants, de nombreuses améliorations sont encore possibles. Au niveau du protocole de coopération, une extension logique serait de transmettre des ensembles avec des contraintes sur les coalitions au lieu de plusieurs ensembles de coalitions indépendants. En se basant sur l'application concernant les emplois du temps, un agent pourrait par exemple, au lieu de transmettre trois ensembles de coalitions avec les trois alternatives heure 1, heure 2, heure 3, envoyer : « heure comprise entre 1 et 3 ». Cela limiterait le nombre d'ensembles de coalitions traités et permettrait à l'agent qui les reçoit de faire des recherches intelligentes au lieu de devoir évaluer tous les ensembles sans chercher de liens entre eux.

Références

- [AKN 99] AKNINE, S. « A Reliable Algorithm for Multi-agent Coalition Formation », *14th IEEE International Symposium on Intelligent Control, Intelligent Systems and Semiotics*, KOKAR, M. (ed.), Cambridge, Massachusetts, USA, pp. 142-147, 1999.
- [AKN 00] AKNINE S., PINSON S., SHAKUN M.F. « Méthodes de Coordination Multi-agent par Formation de Coalitions », *Systèmes Multi-agents, Ingénierie, Technologie, Expérimentations*, Hermès, 149-161, 2000.

- [BUR 01] BURKE, E.K. MACCARTHY, B. PETROVIC, S., QU, R. « Case-Based Reasoning in Course Timetabling: An Attribute Graph Approach », *ICCBR*, 90-104, 2001.
- [CAI 01] CAILLOU, P., AKNINE, S., PINSON, S. « Méthode de Formation et de Restructuration Dynamique de Coalitions d'Agents Fondée sur l'Optimum de Pareto », AMAL EL-FALLAH SEGHROUCHNI (ed.), *Fondements des Systèmes Multi-agents : Modèles, Spécifications Formelles et Vérification*, pp. 55 -70, Hermès, 2001.
- [CAI 02a] CAILLOU, P., AKNINE, S., PINSON, S. « A Multi-Agent Method for Forming and Dynamic Restructuring of Pareto Optimal Coalitions », *AAMAS, International Joint Conference on Autonomous Agents and Multi-agent Systems*, Bologna, Italy, 1074-1081, July 2002.
- [CAI 02b] CAILLOU, P., AKNINE, S., PINSON, S. « Multi-Agent Models for Searching Pareto Optimal Solutions to the Problem of Forming and Dynamic Restructuring of Coalitions », *ECAI*, Lyon, July, 13-17, 2002.
- [KAH 84] KAHAN J.P., RAPOPORT A. *Theories of coalition formation*, Hillsdale, LEA, 1984.
- [KAN 01] KANO, H., KONDO, M., SUGIMOTO, M. « Solving Timetabling Problems using Genetic Algorithms based on Minimizing Conflict Heuristics », *EUROGEN*, 19-21, September 2001.
- [MEI 01] MEISELS A., SCHAERF, A. « Modelling and Solving Employee Timetabling Problems », *Applied Intelligence*, 2001.
- [SAN 96] SANDHOLM T.W. *Negotiation among Self-Interested Computationally Limited Agents*, *PhD. Thesis*, University of Massachusetts, Amherst, 1996.
- [SAN 97] SANDHOLM T.W., LESSER V.R. « Coalitions among computationally bounded agents », *Artificial Intelligence*, 94, 99-137, 1997.
- [SAN 99] SANDHOLM T.W., LARSON K., ANDERSSON M., SHEHORY O., TOHMÉ F. « Coalition Structure Generation with Worst Case Guarantees », *Artificial Intelligence*, 111, 209-238, 1999.
- [SHE 98] SHEHORY O, KRAUS S. « Methods for task allocation via agent coalition formation », *Artificial Intelligence* 101, 165-200, 1998.
- [STI 96] STIDSEN, T. KRAGELUND, L. V. MATEESCU, O., Jobshop « Scheduling in a Shipyard », 639-646, *ECAI*, 1996.
- [VAU 01] VAUVERT G., EL FALLAH – SEGHROUCHNI A. « Coalition Formation among Strong Autonomous and Weak Rational Agents », *MAAMAW*, Annecy, 2001

