Formation de coalitions pour des agents libres

Guillaume Vauvert — Amal El Fallah Seghrouchni

Laboratoire d'Informatique de Paris Nord
Université Paris XIII
Avenue Jean-Baptiste Clément
93430 Villetaneuse

{guillaume.vauvert,amal.elfallah}@lipn.univ-paris13.fr

RÉSUMÉ. *En tant qu'organisation dynamique, la formation de coalitions apporte souplesse et dynamique pour appréhender des situations de coopération ou de compétitivité dans des environnements ouverts. En effet, dans un contexte économique distribué, les agents considérés sont nécessairement hétérogènes, égoïstes et libres en comportement et en raisonnement, ce qui nous a amené à proposer un protocole qui privilégie la liberté des agents. Appliqué à la répartition de tâches entre agents, ce protocole est de plus ouvert, distribué et égalitaire. Il repose sur un échange de préférences calculées sur la base de critères qualitatifs et quantitatifs. Tout en permettant diverses rationalités et stratégies, nous montrons qu'il termine et garantit l'atteinte d'un consensus.*

ABSTRACT.

MOTS-CLÉS : *Alliance, Coalition, Consensus, Diffusion, Préférence*

KEYWORDS:

1re soumission à *JFIADSMA*, le 28 avril 2000.

1. Introduction

L'intérêt essentiel des systèmes multi-agents (S.M.A.) réside dans la collaboration d'agents. Plusieurs approches ont été élaborées pour favoriser cette collaboration : la résolution distribuée de problèmes, les réseaux contractuels, les approches basées sur les organisations, les protocoles de négociation, la formation de coalitions, etc.
Les coalitions permettent aux agents qui les forment de satisfaire des besoins nécessitant la synergie des compétences de plusieurs agents comme par exemple dans le cadre de la résolution de tâches complexes pour lesquelles des agents agissant seuls seraient incapables ou beaucoup moins efficaces.
Une coalition peut être définie comme une organisation à court terme reposant sur des engagements ponctuels et contextuels permettant ainsi aux agents de tirer un bénéfice maximal de leurs compétences respectives (dans le contexte économique par exemple, plusieurs entreprises se regroupent "virtuellement" pour répondre à un appel d'offre nécessitant des compétences diverses). La formation des coalitions offre plusieurs avantages : 1) la notion d'engagement ponctuel laisse aux agents la possibilité de réagir de manière opportuniste et leur permet de réviser dynamiquement leurs intérêts et par conséquent leurs objectifs ; 2) la formation d'une coalition ainsi que sa dissolution sont contextes-dépendantes permettant ainsi aux agents d'adapter leurs accointances dynamiquement ; 3) contrairement à une organisation statique (i.e. prédéfinie), la formation de coalitions permet d'appréhender de façon plus souple des environnements ouverts et dynamiques.
D'un point de vue S.M.A., les recherches dans le domaine de la formation des coalitions tentent de proposer des protocoles d'automatisation qui soient les plus "réalistes" possibles. En effet, la recherche de solutions optimales (généralement mesurées en terme de gain global maximal) requièrent des algorithmes NP-complets et du fait de cette complexité, plusieurs hypothèses simplificatrices sont introduites afin de générer des solutions tractables. Une des hypothèses les plus fortes consiste à supposer que les agents sont coopératifs voire altruistes. Dans le cadre de ce papier, contrairement aux approches existantes (cf. section 2) et dans un souci de généralité, nous ne faisons aucune hypothèse sur les prédispositions des agents à collaborer, nous nous contentons de les y inciter (*e*.g. les agents seront amenés à coopérer s'ils y trouvent un certain intérêt). Ainsi, chaque agent est libre de son comportement et est susceptible d'avoir des objectifs égoïstes (*e*.g. dans le cadre du commerce électronique). Or un ensemble d'agents égoïstes mène généralement au blocage, ce qui finalement ne satisfait personne. Pour éviter ce type de situations, nous proposons un protocole pour la formation de coalitions qui respecte la liberté des agents tout en garantissant l'atteinte d'un consensus.
Cet article est organisé comme suit : la section 2 présente un bref état de l'art sur la formation de coalitions et positionne notre approche, en particulier, par rapport à celles qui traitent des agents égoïstes. La section 3 motive notre démarche qui privilégie la liberté des agents et en définit les contours. La section 4 présente notre approche de résolution du problème de consensus dans le cadre d'agents libres et décrit informellement les grandes lignes du protocole que nous proposons. La section 5 formalise les concepts nécessaires à la définition du protocole de consensus et en donne l'algo-

rithme. La section 6 discute les aspects algorithmiques de notre protocoles à savoir sa terminaison, sa complexité et quelques aspects d'implantation. Enfin, la section 7 conclut cet article et en présente les perspectives.

2. Positionnement

Plusieurs travaux dans le domaine des S.M.A. se sont intéressés aux organisations dynamiques et notamment à la formation de coalitions. La plupart d'entre eux supposent *a priori* que tous les agents ont la même rationalité, ce qui permet de réduire considérablement l'espace de recherche. L'hypothèse la plus forte consiste à supposer que les agents sont altruistes, ce qui revient, comme dans [SHEHORY 95a], à calculer de façon distribuée la solution qui maximise l'utilité commune.

Les travaux avec lesquels nous comparons ce papier ne concernent que des agents égoïstes : considérer des agents altruistes change profondément le problème et de ce fait les solutions. Ainsi, comme dans [SHEHORY 95a], le problème est réduit à faire calculer de façon distribuée la meilleure solution pour le système dans son ensemble, ce qui est un problème difficile, car il n'est pas évident d'avoir une vue globale du système pour un de ses composants. Pour nous, il s'agit d'arriver à concilier les intérêts individuels forcément conflictuels (sinon il n'y aurait pas de problème) pour parvenir, malgré tout, à une solution qui soit acceptée par tous, mais au sens de la légitimité de sa construction.

Puisqu'à toute règle il faut son exception, nous citerons [SHEHORY 95b] qui met en lumière l'intérêt des coalitions comme recouvrement et non pas partionnement des agents pour obtenir une solution. C'est pour cette raison que nous avons opté pour l'appartenance d'un agent à une ou plusieurs coalitions. En effet, augmenter la taille de l'ensemble des solutions facilite le consensus et permet des solutions plus intéressantes qualitativement.

La problématique de la formation de coalitions consiste à chercher une solution qui soit la plus satisfaisante pour l'ensemble des agents. C'est typiquement le cas lorsqu'on cherche à calculer la valeur de Shapley (qui correspond grosso-modo à l'espérance de l'utilité) : on cherche à trouver la configuration qui satisfasse globalement le plus les agents. Certes, l'intérêt global du S.M.A. est évident mais la méthode imposée laisse peu de place à la liberté individuelle des agents. En effet, soit la configuration est calculée de façon externe, soit elle est calculée par les agents eux-mêmes, mais en leur attribuant un rôle défini qu'ils doivent suivre. C'est le cas par exemple dans [ZLOTKIN 94] qui propose une méthode de calcul distribuée de la valeur de Shapley. Les auteurs imposent donc aux agents que la meilleure solution soit celle qui maximise la valeur de Shapley. Certes, pour garantir une relative satisfaction de l'agent, ils prennent soin de ne pas le forcer à former une coalition qui ne lui apporte pas plus que ce qu'il aurait eu avant. Mais tout ce déterminisme ne permet pas aux agents de développer leur stratégie propre : si un agent accepte de prendre le risque de créer une coalition à faible rendement dans l'espoir de former plus tard une coalition très productive, cela peut lui être très bénéfique.

Dans [KETCHPEL 94], Steven Ketchpel a redonné un peu de liberté aux agents en leur permettant d'avoir des estimations différentes du revenu d'une coalition. Le problème essentiel qui se pose est la répartition des bénéfices qui ne sont pas connus avec précision au moment de la formation. L'algorithme proposé consiste en une suite d'agrégations d'agents et de coalitions. Chacun négocie son prix de participation en fonction de ses croyances sur la valeur de la coalition. Dans un deuxième temps, les agents décident comment diviser les revenus futurs de la coalition. Pour chaque coalition, les agents élisent un manager qui négocie pour eux. Cette élection consiste en fait en une vente aux enchères, celui qui promet le plus est élu. Ce qui est intéressant ici, c'est que les agents peuvent avoir des conceptions différentes des revenus et parvenir tout de même à un accord. Mais, comme beaucoup d'autres, cet algorithme suppose que tout est monnayable, que tout peut être estimé par une valeur réelle, ce qui a des conséquences sur l'algorithme lui-même, le rendant ainsi très dépendant de ce présupposé. C'est pour cela que nous avons fait le choix des préférences qui sont indépendantes des critères qui les ont produites laissant aux agents toute la liberté pour élaborer leurs propres stratégies.

La deuxième critique concerne les dépendances entre les coalitions telles que définies dans [KETCHPEL 94] : dans une formation itérative, un agent ne peut pas prendre en compte la participation future éventuelle de tel ou tel agent ; en effet, l'intérêt pour une coalition dépend de l'ensemble de ses membres et pas seulement de l'augmentation de sa valeur à l'étape en cours. Dans notre approche, au contraire, toutes les solutions étant estimées globalement et en parallèle, les agents peuvent pleinement estimer leur intérêt pour chaque solution.

Les agents ont beaucoup de liberté, ce qui nous permet de prendre en compte naturellement ce qui apparait souvent comme des contraintes : la dépendance de tâches et celle des coalitions. Le protocole ne s'appuie que sur les préférences, il ne tient pas compte du raisonnement qui les a engendrées et par conséquent, chaque agent est libre d'avoir sa rationalité propre (cf. section 3.1). En effet, une rationalité bonne globalement (de coalition, altruisme) ne l'est pas individuellement et peut inciter les agents (tous ou certains) à ne pas l'appliquer. Ainsi, utiliser ces presupposés pour calculer la meilleure solution dans un contexte d'agents égoïstes n'est plus possible. De même, une stratégie individuelle ne peut être toujours optimale car elle dépend de celles des autres.

3. Notre démarche : privilégier la liberté des agents

3.1. *Liberté et rationalité*

La difficulté de gestion des coalitions augmente considérablement lorsqu'on considère des agents libres de raisonner et d'agir. La liberté de raisonnement nous empêche de faire des hypothèses sur les connaissances, les motivations, les mécanismes d'inférence ou de décision des agents. La liberté d'agir implique qu'il n'existe pas d'action interdite *a priori*.

Cependant, la liberté est légitimée voire indispensable pour une large classe d'applications. Par exemple, on ne peut pas supposer que tous les agents qui interviennent dans une négociation commerciale soient prêts à suivre à la lettre le protocole imposé, ni qu'ils seraient prêts à dévoiler leurs propres stratégies. Cette liberté satisfait également un des principaux fondements des agents, à savoir le principe d'autonomie tant recherché.

Cette liberté amène naturellement à supposer que les agents ont un comportement égoïste. En effet, l'altruisme n'est pas «naturel» (surtout en milieu économique), car l'agent n'a pas *a priori* intérêt à perdre pour donner aux autres. Il peut certes le faire, par choix stratégique, parce que forcé ou pour toute autre raison, mais il serait artificiel de le supposer.

Un tel relâchement de contraintes sur les agents a évidemment de nombreuses conséquences. Par exemple, on ne peut plus s'assurer qu'un consensus sera atteint, car si tous les agents y sont hostiles, aucun protocole ne permettra d'atteindre un état d'équilibre. De même, la possibilité de tricher, *i*.e. de ne pas suivre le protocole, doit également être prise en compte.

Cependant, cette liberté conférée aux agents est compensée par une forme de rationalité qui permet d'espérer un fonctionnement global cohérent. Contrairement aux travaux dans le domaine où la seule rationalité considérée est la rationalité économique, nous laissons aux agents la possibilité de définir leur propre rationalité. En effet, il peut y avoir des formes de rationalité différentes qui, toutes, produisent des comportements rentables. Ainsi, si un agent cherche à gagner le plus possible en moyenne, il a intérêt à se montrer conciliant pour s'assurer un revenu régulier. Cependant, rien n'empêche un autre agent d'avoir d'autres stratégies comme de ne faire aucun compromis pour gagner beaucoup mais rarement, ou d'autres objectifs comme obtenir le plus gros revenu en une seule fois. Cela donnerait alors des comportements peu sociables qui ne favoriseraient pas le consensus, mais qui ne seraient pas non plus très lucratifs. Or, dans un contexte économique, nous supposons que de tels agents – non rentables – seront rapidement éliminés. Il est donc profitable à un agent d'être conciliant et d'adopter une position moyenne plutôt que d'être intransigeant et de conserver les préférence initiales ; mieux vaut favoriser les solutions les plus plausibles, pas seulement les plus favorables.

3.2. *Conséquences sur les protocoles et les organisations*

La liberté des agents a des incidences directes sur la conception des protocoles et l'organisation des agents en coalitions. Pour obtenir des comportements collectifs cohérents, il est nécessaire d'établir des règles. Pour respecter les libertés individuelles, ces règles ne doivent contraidre que des données *observables*, et donc que des comportements dirigés vers les autres. En effet, seuls les comportements sont vérifiables et toute tentative d'inférer les raisonnements à partir d'actes perçus donne forcément un résultat très approximatif puisque interprété. Le protocole proposé repose uniquement sur les actes observables, car si l'on veut que les agents le respectent, il faut qu'en cas de fraude, ils soient sanctionnés. La sanction est alors le moteur d'incitation à suivre les règles données.

Dans un tel contexte, un protocole doit être : 1) universel, afin notamment de ne rien

supposer sur les choix individuels, ni sur les rationalités individuelles, respectant ainsi la liberté des agents ; 2) égalitaire, pour ne pas favoriser des agents sur des aspects extérieurs à l'objet de la négociation, comme par exemple la vitesse de communication, bien qu'un agent disposant de plus de ressources (processeur, mémoire, information, processus cognitifs) sera forcément avantagé ; 3) distribué, car la centralisation amène les problèmes habituels (surcharge du réseau et de l'agent central, faiblesse aux pannes,. . .) en plus de la possibilité accrue de triche pour l'agent central alors en situation de pouvoir ; bien que l'on puisse faire la supposition d'un agent impartial, l'intégrité d'un agent pourrait toujours être mise en cause (influences externes, corruption), ce qui nuirait à la légitimité de la solution.
La liberté des agents modifie aussi la perception qu'on peut avoir des organisations. Celles-ci sont en effet parfois considérées comme des entités à part entière, dépendant certes de leurs membres, mais possédant néanmoins des attributs propres pouvant aller jusqu'à une certaine autonomie. Dans le contexte de cet article, une organisation est un ensemble d'agents liés (contractuellement, ponctuellement), mais qui disposent toujours de toute leur liberté. Chaque agent y joue un certain rôle, mais pour les mêmes raisons, on n'identifiera pas rôle et comportement. Un rôle est un ensemble de droits et de devoirs, codifiés dans un règlement, dans l'espoir d'obtenir un comportement d'ensemble cohérent. Les droits sont représentés par l'ensemble des actions possibles à chaque étape et un devoir par l'ensemble des actions attendues par les autres agents. Mais tout agent garde sa liberté et peut donc respecter ou non le règlement, et il est donc nécessaire de prendre en compte ces possibilités.
Généralement, un protocole décrit à la lettre ce que doivent et ce que peuvent faire les agents, mais il le fait en suivant un certain esprit. Cependant, une faille dans un protocole peut permettre à des agents d'en respecter la lettre, mais pas l'esprit. Ce genre de comportement isolé peut ne pas trop nuire à l'ensemble, mais s'il est bénéfique à l'agent, il se généralisera certainement. Dans la plupart des cas, le système sera désorganisé et ne fonctionnera plus. Il faut donc que le protocole soit très bien pensé pour ne pas autoriser ces comportements qui mènent au blocage. Ces agents auront de fait la possibilité de s'échanger des informations extérieurement au protocole ; il faut donc nécessairement que cela ne constitue pas une source de blocage.

3.3. *Objectif : résolution du problème de consensus pour la répartition des tâches*

Nous nous plaçons dans le cadre de la résolution de tâches par des agents : un système reçoit un ensemble de tâches subdivisées en sous-tâches et propose ces sous-tâches à des agents inscrits. Ceux-ci tentent alors de se répartir l'ensemble des sous-tâches, chacun ne pouvant en exécuter qu'une partie. Une tâche est exécutée quand toutes ses sous-tâches le sont. Pour simplifier, nous avons choisi de fixer la décomposition, car celle-ci a des conséquences sur les revenus des agents et pourrait aussi constituer un enjeu de négociation. Nous supposons de plus qu'il n'y a pas de contrainte sur l'ordre des sous-tâches.
L'objectif de l'algorithme proposé est de permettre à des agents libres ayant des intérêts potentiellement incompatibles de parvenir à trouver un consensus sur la répartition

des sous-tâches, mais du point de vue des agents. Il ne s'agit pas d'imposer aux agents une solution ni même une procédure de calcul de la solution qui serait justifiée par le concepteur, mais de laisser chacun agir selon sa propre stratégie cohérente avec sa rationalité.
La motivation des agents à la réalisation des sous-tâches repose sur les gains associés et les stratégies propres aux agents qui leur permettent d'établir des préférences sur l'ensemble des solutions possibles. Ainsi, le protocole que nous proposons ne tient compte que des préférences choisies.

3.4. *Concepts du problème*

Nous définissons un problème de formation de coalitions par le $5-uplet$: $\langle \mathcal{A}, \mathcal{T}, \mathcal{S}, \mathcal{C}, \mathcal{G} \rangle$ avec :

- $\mathcal{A}$: l'ensemble des agents candidats à l'exécution de sous-tâches ;
- $\mathcal{T}$: l'ensemble des tâches qui devront être accomplies ;
- $\mathcal{S}$: l'ensemble des sous-tâches à exécuter ;
- $\mathcal{C}$: l'ensemble des compétences nécessaires aux sous-tâches ;
- $\mathcal{G}$: l'ensemble des gains.

Définition 1 **: agent** $a \in \mathcal{A}$ **:** possède un certain ensemble de compétences ; un agent $a \in \mathcal{A}$ est donc partiellement défini par :$a = \langle C, \ldots \rangle, C \subset \mathcal{C}$. Cette définition sera complétée utlérieurement par d'autres éléments comme sa stratégie (voir (5.2, p. 12)).
Définition 2 **: tâche** $t \in \mathcal{T}$ **:** ensemble de sous-tâches : $t = \langle S \rangle, S \subset \mathcal{S}$.
Définition 3 **: sous-tâche** $s \in \mathcal{S}$ **:** son exécution nécessite des compétences et peut rapporter un certain gain qui sera utilisé par l'agent pour calculer ses préférences ; ainsi, une sous-tâche est définie par $s = \langle C, g \rangle, C \subset \mathcal{C}, g \in \mathcal{G}$.
Définition 4 **: compétence** $c \in \mathcal{C}$ **:** item unique qui représente ce qui est nécessaire à un agent pour pouvoir réaliser une sous-tâche. Une sous-tâche peut nécessiter plus d'une compétence.
Définition 5 **: gain** $g \in \mathcal{G}$ **:** assimilé à un revenu pour simplifier les calculs internes des agents : $\mathcal{G} = [0, GainMax]$; cependant, n'importe quel ensemble pourrait être utilisé, l'algorithme est indépendant du type de gains.

4. Approche de résolution du problème de consensus

4.1. *Hypothèses pour le consensus*

Nous commencerons par décrire intuitivement la méthode de résolution proposée, puis nous formaliserons les concepts présentés.
Pour parvenir à un consensus, il faut que les agents modifient leurs opinions. Parmi toutes les possibilités imaginées (pression, corruption,. . .), nous avons retenu l'échange

de préférences, car elle nous parait plus rationnelle, et mène donc à une solution plus légitime.
Les agents échangent leurs préférences, ce qui modifie leurs connaissances et par conséquent leurs préférences personnelles. Cependant, des situations de blocage peuvent se produire : stagnation, bouclage, apparition de «clans» stables aux préférences proches et qui refusent de changer d'avis. Pour pallier de telles situations et favoriser la convergence, nous avons prévu une procédures de déblocage qui consite à former des *alliances*. Deux agents aux préférences proches peuvent décider de s'allier et de se comporter comme un agent de poids supérieur, c'est-à-dire en n'intervenant que comme une seule entité avec donc une seule préférence mais bien sûr pondérée par le nombre d'agents. Pour la suite du protocole, chaque alliance interviendra *via* son représentant, agent choisi (selon une procédure de vote) dans l'alliance pour jouer le rôle d'interface avec le reste du système. Notamment, des alliances peuvent devenir membres d'une nouvelle alliance.
Dans un premier temps, des agents (ou des alliances) peuvent s'allier, mais si aucune alliance ne se crée, une deuxième procédure d'alliance forcée est mise en place : l'ensemble des agents décide quelles sont les deux entités (agent ou alliance) qui doivent s'associer. Lorsqu'une alliance a été formée, le processus normal d'échange de préférences reprend.
Pour un agent, appartenir à une alliance augmente ses potentialités de gain, bien que ses préférences puissent se retrouver noyées dans la masse. Afin d'éviter des changements à l'infini et de favoriser la stabilité du système, la dissolution d'une alliance n'est possible qu'une fois le consensus atteint.

4.2. *Spécification des rôles des agents*

Un agent peut jouer plusieurs rôles au sein du système :

– Agent organisateur : il est chargé de la gestion des inscriptions, des tours et de l'envoi des données;

– Agent candidat : il recherche des sous-tâches à exécuter. Il reçoit et envoie ses préférences au sein de son alliance ou du système, propose/accepte le passage en mode déblocage/déblocage forcé, propose/accepte la formation d'alliance ;

– Agent surveillant : il est responsable de la vérification (aléatoire) du respect du protocole ;

– Agent représentant : il reçoit les préférences des autres membres du système (de même niveau que l'alliance) et les diffuse en interne. En externe, il diffuse les choix des membres de son alliance.

Pour inciter les agents à accepter certains rôles qui ne sont pas directement rentables mais qui nécessite des ressources, nous proposons de les lier à une rémunération, soit par acceptation du rôle (organisateur), soit par les résultats obtenus (surveillant), suivant notamment s'il est possible ou non de vérifier qu'ils agissent bien en fonction de leur rôle. Pour ces deux cas, il faut alors décider du revenu de chaque rôle. A titre d'exemple, pour le premier cas, il suffit par exemple de «vendre» le rôle aux enchères et pour le rôle de surveillant, nous optons pour une vérification aléatoire où la pénalité de la fraude est payée par l'agent coupable au surveillant.

4.3. *Description informelle de l'algorithme de consensus*

Initialement, l'agent organisateur possède les données du problème, c'est-à-dire l'ensemble des tâches, la décomposition en sous-tâches et pour chacune d'entre elles les compétences nécessaires et les revenus. Les agents qui désirent obtenir des sous-tâches s'adressent à l'organisateur qui leur transmet alors les données en sa possession, puis ceux qui sont intéressés s'inscrivent. Une date limite clôt les inscriptions et l'organisateur diffuse aux candidats la liste des agents candidats.

Les agents s'informent sur leurs capacités mutuelles afin de calculer l'ensemble des solutions possibles, ensemble sur lequel portera les préférences. Ce calcul peut être effectué en parallèle ou individuellement, car il est possible de vérifier que tous ont la même information et donc les mêmes solutions. En fait, un échange des sous-tâches dont on est capable suffit, car la répartition des tâches en sous-tâches et l'ensemble des compétences nécessaires à chacune d'entre elles est une connaissance commune. Les agents décident ensuite d'un ordre sur les solutions pour permettre de désigner une solution en faisant référence à son rang dans cet ordre. Cet ordre est une connaissance commune et n'est qu'un outil pour diminuer la taille des messages. Les agents procèdent ensuite à des échanges de préférences tour par tour. Ce système de tours permet d'obtenir des photographies globales des préférences au même état d'avancement, ce qui est nécessaire pour estimer s'il y a consensus et pour éviter que le système soit perturbé par des messages qui arriveraient en avance ou en retard d'un tour. De plus, un agent qui utiliserait des informations de tours suivants serait très avantagé, ce qui inciterait les agents à attendre et produirait à terme un blocage. Pour remédier à cette situation, nous proposons une diffusion parallèle (4.4, p. 11) qui améliore la diffusion cryptée de messages proposée dans [ZLOTKIN 94].

Pour le premier tour, les agents doivent estimer l'intérêt pour les sous-tâches sans connaître les préférences des autres ; leur principale base de critères sera donc les revenus promis. D'autres critères peuvent cependant intervenir s'ils possèdent des informations extérieures.

Aux tours suivants, les agents prennent en compte les préférences des autres pour calculer leurs nouvelles préférences.

Concernant le critère de consensus, on a le choix entre une estimation locale et un critère global. Dans le premier cas, comme il y a peu de situations où tous les agents sont d'accord, il faudrait un nouveau critère pour déterminer à partir de quel taux de satisfaction on considère qu'il y a effectivement consensus au niveau global (une majorité par exemple). Nous avons donc choisi de fixer un critère global connu des agents au début du processus, critère qui influencera leur stratégie. Les critères qui utilisent tous les agents (majorité, unanimité, $2/3$) sont très dépendants du nombre total d'agents, car plus celui-ci est grand, plus difficile est le consensus. Nous proposons donc d'utiliser un critère qui ne concerne que les agents inclus dans une solution. Ainsi, on peut

considérer qu'une solution est acceptable si tous les agents qui participent à la solution préfèrent cette dernière (classement en première position). Cette règle est plutôt rigide, mais elle incitera les agents à se presser de trouver une solution, et notamment à former des alliances.
À la fin de chaque tour, chaque agent doit proposer soit d'arrêter le processus s'il y a consensus, soit de passer au tour suivant, soit de former une alliance, soit de passer en mode déblocage.

– En cas de consensus, le critère étant global, tous les agents comprendront en même temps qu'un consensus a été atteint.

– Le passage au tour suivant est l'action par défaut, lorsqu'aucune des autres n'a été choisie.

– Un agent peut proposer de former une alliance en communiquant en privé avec ses éventuels partenaires ; si l'alliance est formée, les agents, *v*ia leur représentant, en informent le reste du système, puis passent au tour suivant ou en mode déblocage.

– Passage au mode déblocage : chaque agent a un critère pour estimer qu'il y a blocage, et, le cas échéant, il proposera aux autres de passer en mode déblocage. Si tous (ou une certaine proportion fixée) acceptent, le système passe en mode déblocage. Les agents peuvent alors former des alliances et si au moins une est formée, le système passe au tour suivant ; sinon, il passe en mode déblocage forcé.

En mode déblocage forcé, il faut imposer à un couple d'entités (agent ou alliance) de former une nouvelle alliance. Le choix peut se faire par vote ou sur critère connu. La deuxième méthode semble préférable, car elle produira des regroupements plus rationnels, moins soumis aux calculs stratégiques individuels. Par contre, les agents visés seront connus, ce qui risque de ne pas inciter les autres à chercher des partenaires. Le critère retenu consiste à mesurer une distance entre les préférences des entités et à obliger les deux plus proches à s'allier. En cas d'égalité, toutes sont forcées à s'allier. C'est l'agent organisateur qui choisit le couple d'entités (agent ou alliance) qui devra former une nouvelle alliance.
Nous avons conscience que la formation d'alliance peut engendrer des problèmes de dépendances bloquantes. En effet, si un agent a_1 fait une proposition d'alliance avec a_2, a_2 avec a_3 et a_3 avec a_1, et que tous préfèrent leur premier choix à celui qui leur est offert, ils risquent de tous attendre. Pour éviter des interblocages, nous proposons d'utiliser un délai qui soit accélérera soit empêchera la formation d'alliance, mais évitera dans tous les cas un blocage fatal. Ainsi, les alliances permettent-elles forcément d'obtenir une convergence.
Après formation d'une alliance, il faut une procédure d'unification des préférences de ses composantes qui soit rapide et rationnelle. Nous proposons d'effectuer une sorte de moyenne des préférences sur l'ensemble des membres de l'alliance en attribuant à chaque solution d'autant plus de points qu'elle apparaît bien classée dans une préférence, puis de ranger dans l'ordre décroissant ces solutions ; on obtient ainsi la nouvelle préférence. Nous supposons ici que les agents se sont engagés à respecter le choix du système et qu'ils acceptent donc les alliances désignées.

4.4. *La diffusion parallèle*

Dans [ZLOTKIN 94], les agents diffusaient leurs informations cryptées, puis lorsqu'ils avaient tous reçus l'information cryptées de tous les autres agents, ils diffusaient la clé. Mais cette technique n'empêche pas certaines situations de se produire : dans un système à trois agents a b et c, a envoie ses préférences à b et c, b à a, mais c uniquement à a ; comme a a reçu toutes les préférences, il envoie sa clé à b et c, ce qui permet à c d'envoyer à b une autre préférence. Les auteurs supposaient que les agents ne trichaient pas trop ; nous améliorons ce mécanisme pour empêcher toute fraude. Les agents envoient leurs préférences cryptées avec une clé privée, puis lorsque chacun à reçu toutes les préférences, il diffuse un accusé de réception. Ce n'est que lorsqu'un agent a reçu tous les accusés qu'il diffuse sa clé. Cette technique ne fait pas que déplacer le problème, elle le résout. Si on exécute le même scénario ici, l'agent qui a triché ne pourra plus changer son message crypté. Il pourrait bien sûr envoyer une fausse clé qui ne marche pas, mais la fraude serait visible et la suspicion pèserait alors sur lui.

5. Cadre conceptuel et mise en œuvre du consensus

5.1. *Définitions des concepts pour la résolution*

Pour résoudre le problème posé, les agents échangent leurs préférences sur les diverses solutions possibles. Ils ont donc besoin de représenter : les solutions, les préférences, les alliances et les coalitions.

Définition 6 **: solution :** Une solution est une affectation de chaque sous-tâche à un agent qui en est capable. Pour préciser ceci, on définit l'application $capacite : \mathcal{A} \rightarrow 2^{\mathcal{S}}$ qui à tout agent a associe l'ensemble des sous-tâches dont il est capable, un agent a étant capable d'une sous-tâche s si $s.C \subset a.C$. Une solution σ, c'est donc une application $\mathcal{S} \rightarrow \mathcal{A}$ telle que $\forall s \in \mathcal{S}, a = \sigma(s), s \in a.capacite$. Symétriquement, on peut définir pour tout $s \in \mathcal{S}$ l'ensemble $s.capable \subset \mathcal{A}$, l'ensemble des agents $a \in \mathcal{A}$ tels que $a.capable(s)$. On appelle Σ l'ensemble des solutions.

Définition 7 **: préférence :** Une préférence est une information sur l'intérêt de l'agent pour chaque solution. On pourrait utiliser comme représentation un ordre partiel avec distances entre préférences (un graphe orienté pondéré dont les sommets représenteraient les solutions et les arcs la relation « est préféré à », la pondération donnant alors une idée de la différence d'intérêt pour les solutions). Pour simplifier les calculs des agents, une préférence ne sera ici qu'un ordre total $\prec$ sur les solutions dont la sémantique est donnée par : $\forall(\sigma_1, \sigma_2) \in \Sigma \times \Sigma, \sigma_1 \prec \sigma_2 \Leftrightarrow$«la solution σ_1 est préférée à la solution σ_2». On appelle Π l'ensemble des préférences possibles des agents.

À chaque tour, on peut faire une photographie de toutes les préférences des agents : on appelle cela une vue, qui est donc une application de $\mathcal{A} \rightarrow \Pi$. La suite des vues $v_i, 1 \leq i \leq t$ du premier tour au tour t actuel s'appelle une histoire.

Nous avons dit qu'une organisation n'avait pas d'existence propre, mais seulement au travers des liens qui unissent les agents qui la composent. Chaque agent doit donc avoir une connaissance minimale des organisations auxquelles il appartient, afin de pouvoir au moins obéir au protocole. Nous nous limiterons ici à ce minimum, tout en ayant bien conscience qu'une augmentation d'information et son utilisation correcte rendrait un agent plus performant, notamment dans la gestion du pouvoir.

Définition 8 **: alliance :** Une alliance contient des agents ou d'autres alliances, avec la restriction qu'un agent ne peut pas appartenir à plusieurs alliances d'une même coalition. Une alliance a donc une structure hiérarchique, les feuilles représentant les agents et les nœuds les alliances.

Soit Λ l'ensemble des alliances ; une alliance $\lambda \in \Lambda$ est définie par $\lambda = \langle A, L, a_{rep} \rangle$, $A \subset \mathcal{A}$, $L \subset \Lambda$, et $a_{rep} \in \mathcal{A}$ un agent membre de l'alliance. Un agent ne peut appartenir qu'à une seule alliance. On dit qu'un agent a est membre d'une alliance λ si $a \in \lambda.A$ ou $\exists \lambda' \in L$, a est membre de λ'.

Il faut aussi définir l'opération de formation d'alliances. Soient A un ensemble d'agents et L un ensemble d'alliances qui désirent former une nouvelle alliance. On définit l'opération $fusion : 2^{\mathcal{A}} \times 2^{\Lambda} \to \Lambda$ par $fusion(A, L) = \{A, \bigcup_{\lambda \in L} \lambda\}$.

Nous définissons maintenant une fonction qui calcule la préférence d'une alliance. Nous proposons d'ordonner les solutions en nous basant sur un score obtenu d'après leur classement dans les préférences des membres. Plus une solution est préférée, plus elle obtient de points. Pour $\lambda \in \Lambda, \lambda = \{A, L\}, A \subset \mathcal{A}, L \subset \Lambda$, et pour tout $\sigma \in \Sigma$, on définit $score(\lambda = \{A, L\}, \sigma) = \sum_{a \in \mathcal{A}}(|\mathcal{S}| - a.ord(\sigma)) + \sum_{\lambda \in *} |\lambda| \times (|\mathcal{S}| - \lambda.ord(\sigma))$. On définit alors $PrefUnifiee(\lambda) = \prec / \forall(\sigma_1, \sigma_2) \in \Sigma^2, \sigma 1 \prec \sigma_2 \Leftrightarrow score(\sigma_1) \geq score(\sigma_2)$.

Définition 9 **: coalition :** ensemble des agents affectés *v*ia les sous-tâches à une tâche. Pour une solution σ, la coalition associée à la tâche $t \in \mathcal{T}$ est donc l'ensemble d'agents $\Xi(\sigma, t) \subset \mathcal{A}$ défini par $\Xi(\sigma, t) = \{a \in \mathcal{A} / \exists s \in \mathcal{S}, s \in t.S, \sigma(s) \ni a\} = \bigcup_{s \in t.S} \sigma(s)$. Une coalition contient exactement l'ensemble des agents ayant à effectuer la tâche ; la solution du problème contient donc exactement les coalitions auxquelles appartient l'agent et la description de toutes les coalitions.

5.2. *Les procédures*

Après avoir décrit formellement les concepts manipulés par les agents pour la résolution, nous allons décrire les outils *v*ia lesquels l'agent applique sa stratégie. Il s'agit de :

– $PrefAPriori$: préférences données au premier tour ;

– $PrefAPosteriori$: préférences aux tours suivants, qui dépendent de celles des autres ;

– $ProposeAlliance$: proposition de formation d'alliance ;

– $AccepteAlliance$: acceptation de formation d'alliance ;

– $ProposeDeblocage$: critère de proposition de passage en mode déblocage ;

– $AccepteDeblocage$: critère d'acceptation de passage en mode déblocage.

Un agent doit disposer d'une fonction lui permettant d'ordonner ses préférences. Cette fonction est définie par des critères personnels liés à un choix de stratégie. Elle dépend notamment de l'information dont il dispose et des préférences des autres.
Définition 10 **:** $PrefAPriori\ :\ \Sigma\ \mapsto \prec$ **:** donne au premier tour la préférence de l'agent, avant de connaître celles des autres. Pour la plupart des agents $a\ \in\ \mathcal{A}$, $PrefAPriori$ sera simplement définie par $\forall(\sigma_1, \sigma_2) \in \Sigma^2, \sigma_1 \prec \sigma_2$
ssi $revenu(a, \sigma_1) \geq revenu(a, \sigma_2)$.
Définition 11 **:** $PrefAPosteriori\ :\ \Sigma, h\ \mapsto \prec$ **:** renvoie la préférence connaissant toutes celles des autres aux tours précédents. Nous ne ferons pas d'étude exhaustive des stratégies, mais en voici tout de même quelques unes. Pour un agent «buté», $PrefAPosteriori(\Sigma, h) = PrefAPriori$. Pour un agent purement altruiste, $PrefAPosteriori(\Sigma, h) = PrefUnifiee(\mathcal{A})$. Par contre, pour un agent a égoïste mais rationnel, $PrefAPosteriori(\Sigma, h) = \prec / \forall(\sigma_1, \sigma_2) \in \Sigma^2, \sigma_1 \prec \sigma_2$
$\Leftrightarrow\ (ScoreAutre(\sigma_1))^{\varepsilon} \times revenu(a, \sigma_1) \geq (ScoreAutre(\sigma_2))^{\varepsilon} \times revenu(a, \sigma_2)$, avec $ScoreAutres(\sigma) = \sum_{a \in \mathcal{A}}(|\mathcal{S}| - a.ord(sigma))$. Si $\varepsilon = 1$, on retrouve les cas altruiste, et si $\varepsilon = 0$, on retrouve le cas égoïste. Une bonne stratégie consiste donc par exemple à choisir la valeur d'ε, et plus précisément sa variation au fil des tours. En effet, il semble intéressant de commencer avec un ε faible pour influencer sur l'ensemble des solutions qui vont être retenues, puis d'augmenter sa valeur pour prendre de plus en plus en compte les préférences des autres, un $\varepsilon \geq 1$ n'étant pas aberrant. Cette formule a en outre l'avantage de classer en dernier les solutions qui ne rapportent rien à l'agent (multiplication par 0).
Définition 12 **:** $ProposeAlliance\ :\ \mathcal{H} \rightarrow 2^{\mathcal{A}}$ **:** cherche quels agents sont suceptibles de former une alliance. Pour cela, nous proposons de mesurer une distance entre les préférences des agents et de choisir les plus proches.
Définition 13 **:** $AccepteAlliance\ :\ \mathcal{H}, \mathcal{A} \rightarrow \{Vrai, Faux\}$ **:** accepte ou refuse de former une alliance avec un groupe d'agents. Là aussi, une distance entre le préférences avec seuillage permet de trancher rapidement.
Définition 14 **:** $ProposeDeblocage\ :\ \mathcal{H} \rightarrow \{Vrai, Faux\}$ **:** décide s'il y a blocage afin de proposer un passage en mode déblocage. On peut estimer raisonnablement qu'il y a blocage lorsqu'il y a un cycle, c'est-à-dire que l'histoire se répète indéfiniment. Nous disons qu'il y a une boucle lorsque deux vues de l'histoire sont identiques. Mais une boucle ne produit pas forcément un cycle, car les critères des agents sont inconnus et puiqu'ils dépendent de l'histoire et en particulier du nombre de tours, deux vues identiques ne conduisent pas nécessairement à la même suite. Nous proposons néanmoins comme critère de détection de blocage la recherche de tels cycles : $ProposeDeblocage\ :\ h \mapsto ProposeDeblocage(h) =$«la dernière vue de h existait déjà dans h». Ce critère est cependant très coûteux en temps et en espace ; on peut cependant le simplifier – mais on y perd en qualité – en ne conservant que les 3 premières préférences de chaque agent. Une version encore moins coûteuse mais bien plus restrictive serait de mesurer uniquement la stabilisation : $h \mapsto ProposeDeblocage(h) =$«la dernière vue de h est la même que l'avant-dernière».
Définition 15 **:** $AccepteDeblocage$ **:** peut être identique à $ProposeDeblocage$ ou plus conciliant.

5.3. *Algorithme*

Chaque agent possède une liste d'interlocuteurs $ListeInter \subset \mathcal{A}$ initialisée à la liste des candidats. L'algorithme suivant est exécuté par chaque agent a_i de façon distribuée.

Algorithme 1

- 1 : Calcul de $PrefAPriori(a_i)$
- 2 : Diffusion/Réception parallèle aux/des membres de $ListeInter$
- 3 : Faire

    Calcul de $PrefAPosteriori(a_i)$

    Diffusion/Réception parallèle aux/des $ListeInter$

    Si consensus,

    return();
- 4 : Tant que $ProposeDeblocage = Faux$ pour tous les agents
- 5 : Si pas $AccordDeblocage$ de tous les agents

    Aller en 3
- 6 : Si $ProposeAlliance$ et $AccepteAlliance$

    Formation de l'alliance λ

    λ choisit $PrefUnifiee$

    Si $a_i \in \lambda$, $ListeInter \leftarrow \{\lambda.Representant\}$

    Si non, $ListeInter \leftarrow ListeInter \setminus (\lambda \setminus \lambda.Representant)$

    Aller en 3
- 7 : Aucune alliance formée $\Rightarrow$ Déblocage forcé
- 8 : Formation de l'alliance choisie par un critère donné
- 9 : Aller en 3

6. Discussion de l'algorithme

6.1. *Terminaison*

Nous n'avons pas fait de supposition sur les critères de passage en mode déblocage, ce qui nous empêche de conclure quant à la terminaison du processus. Cependant, si on fait l'hypothèse que le critère consiste à vérifier qu'il n'y a pas de bouclage, alors on peut conclure à une fin du processus. En effet, le nombre n d'agents et le nombre k de solutions possibles étant finis, il n'existe que $n.k!$ vues possibles. Après $n.k! + 1$ tour, l'histoire contiendra donc nécessairement deux vues identiques : il y aura donc eu une boucle. Or un passage en mode déblocage conduit directement ou indirectement (*via* le déblocage forcé) à une formation d'alliance. Quel que soit le critère de consensus, on peut estimer que si tous les agents sont d'accord, il y a consensus. Or la formation d'alliances conduit au pire à une grande alliance (qui contient tous les agents) à structure d'arbre binaire, ce qui aura nécessité $n-1$ étapes (le nombre de nœuds). Finalement, après $(n.k!+1)(n-1)$ tours, il y a forcément consensus.

6.2. *Complexité*

La complexité dépend notamment du nombre de solutions possibles qui est directement lié aux données du problème. Supposons que notre système contienne n agents. Calculer le nombre moyen d'agents capables de résoudre une sous-tâche n'a pas beaucoup de sens, car cela ne tient compte ni des compétences très répandues, ni surtout des compétences qui vont par lot. Supposons donc que chaque agent puisse traiter $1/m$ des tâches, une tâche a donc en moyenne n/m agents capables de la résoudre, ce qui donne alors $k=(n/m)^s$ solutions. Lorsque le nombre d'agents croît, la quantité de solutions à traiter devient rédibitoire ; il est alors nécessaire de limiter l'espace à considérer. On peut par exemple restreindre l'ensemble des solutions aux trois solutions préférées de chaque agent, avec éventuellement des retraits et des ajouts en cours d'avancement. Mais cela amène de nouveaux problèmes : une solution mal classée par tous peut s'avérer être une bonne candidate pour un consensus. Dans le cas le plus général, notre algorithme ne permet pas de changer de classe de complexité, mais un calcul en moyenne montre que plus les agents ont des compétences qui se recouvrent, plus il y a compétition et plus le consensus est difficile à atteindre. Cependant, il est raisonnable de supposer qu'un système bien conçu répartit les compétences de façon uniforme. Pour une bonne complexité, il faut que le rapport n/m soit le plus petit possible.

6.3. *Éléments d'implantation*

L'objectif est de vérifier l'implantabilité de l'algorithme et de tester les différents paramètres intervenant dans la formation de coalition : stratégies individuelles des agents (calcul des préférences, critère de déblocage, formation d'alliances) et des paramètres globaux (critère de consensus, choix des alliances en déblocage forcé).
Nous avons choisi le langage Java car il permettait *v*ia les threads de simuler naturellement le caractère distribué des S.M.A. Ainsi, chaque envoi de message génère un thread, ce qui permet de simuler des communications asynchrones.
Chaque agent n'a accès qu'à ses données locales et ne peut donc échanger de l'information que *v*ia les messages, ce qui garantit son autonomie. Chaque rôle est implanté sous la forme d'un thread chargé de réagir aux messages reçus ou d'agir de son propre chef. Lorsqu'un agent reçoit un message, il le dirige vers le thread chargé de gérer les messages destinés au rôle.
Pour plus de détails, voir notre démonstration si acceptée.

7. Conclusion et perspectives

Dans une perspective de déploiement de S.M.A. dans un contexte économique, il est nécessaire de considérer des agents hétérogènes, égoïstes et libres dans leur raisonnement et leur comportement. Pour parvenir, dans ce cadre, à former des coalitions nous avons proposé un protocole reposant sur un échange de préférences basées sur des critères quantitatifs et qualitatifs puisque dépendantes des stratégies propres aux agents. De plus, nous avons mis en place des procédures de déblocage grâce au concept souple d'alliance pour éviter la paralysie du système.
Nous avons montré qu'un tel protocole est ouvert, distribué et égalitaire. De ce fait, il ouvre de nombreuses perspectives. Dans un premier temps, on cherchera à améliorer la complexité de calcul en introduisant des heuristiques. Nous pourrons ensuite tester logiciellement les différents paramètres et choisir ainsi les critères et stratégies les plus intéressants.
Comme dans [LOUÇÃ 96], il serait également intéressant de permettre aux agents de garder une trace des alliances qui ont été bénéfiques, afin d'accélerer la convergence vers un consensus. Construire un modèle des autres permettra d'intégrer la notion de confiance et limitera la suspicion.

8. Bibliographie

[KETCHPEL 94] KETCHPEL S., « Forming Coalition in the Face of Uncertain Rewards », *Proceedings of the 12th National Conference on Artificial Intelligence. Volume 1*, Menlo Park, CA, USA, July 31–August 4 1994, American Association for Artificial Intelligence, AAAI Press, p. 414–419.

[LOUÇÃ 96] LOUÇÃ J. A., COELHO H., PINSON S., « Forming Coalition in Task Oriented Multi-Agent Systems », *Proceedings Of IBERAMIA 1996*, IBERAMIA, October 1996.

[SHEHORY 95a] SHEHORY O., KRAUS S., « Task Allocation via Coalition Formation among Autonomous Agent », MELLISH C. S., Ed., *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, San Mateo, August 20–25 1995, The International Joint Conferences on Artificial Intelligence, Morgan Kaufmann, p. 655–661.

[SHEHORY 95b] SHEHORY O., KRAUS S., « Formation of Overlapping Coalitions for Precedence-Ordered Task-Execution among Autonmous Agents », LESSER V., Ed., *Proceedings of the First International Conference on Multi–Agent Systems*, International Conference on Multi-Agent Systems, MIT Press, 1995, p. 330–337.

[ZLOTKIN 94] ZLOTKIN G., ROSENSCHEIN J. S., « Coalition, Cryptography, and Stability : Mechanisms for Coalition Formation in Task Oriented Domains », *Proceedings of the 12th National Conference on Artificial Intelligence. Volume 1*, Distributed AI, Menlo Park, CA, USA, July 31–August 4 1994, American Association for Artificial Intelligence, AAAI Press, p. 432–437.