Formation de coalitions A-core : S-NRB

Pascal François FAYE[1,2], Mbaye SENE[1], Samir AKNINE[2]

[1] LID UCAD, SENEGAL, [2] LIRIS UCBL, FRANCE
pfmfaye@hotmail.com, ngalagne@yahoo.com, samir.aknine@univ-lyon1.fr

RÉSUMÉ. Nous proposons un mécanisme décentralisé de formation de coalitions stables dans un contexte de tâches stochastiques qui tient compte des incertitudes sur, les dépendances, les conflits de préférences et la disponibilité imprévisible des agents. Nous ne supposons aucune connaissance a priori sur les coalitions stables à former et qu'il n'est pas possible de calculer de façon centralisée ces coalitions avant le début de l'exécution des tâches en raison de ces incertitudes sur les agents et des contraintes de temps des tâches stochastiques. Nous appelons *S-NRB (Sequential Non-Return Broadcast)*, ce mécanisme qui permet de former des coalitions dites *A-core*, c'est-à-dire, *Core-stable* et Auto-stables. *S-NRB* combine les méthodes de la théorie des jeux avec les lois de probabilités pour atteindre les coalitions stables requises. L'analyse théorique et les expérimentations montrent que *S-NRB* surmonte dynamiquement les incertitudes des agents et des tâches et favorise l'établissement des coalitions *A-core*.

ABSTRACT. We focus on devising an efficient parallel and decentralized coalition formation mechanism dealing with uncertainties on agent's dependencies, on agent's conflicts, on agents' preferences and on agents' availability when stable agents' teamwork (coalitions) are required to achieve stochastic tasks. We assume no prior knowledge on stable coalitions to form and we consider it is not possible to compute in a centralized manner these coalitions before tasks' achievement due to agents' uncertainties and the time constraint of stochastic tasks. We propose a coalition formation mechanism called S-NRB (Sequential Non-Return Broadcast). The main propertY we emphasize is the A-Core (*Core-stable* and Auto-stables coalitions). So, S-NRB combines methods of game theory and the laws of probability to reach the required stable coalitions. The theoretical analysis and the experiments show how S-NRB dynamically overcomes uncertainties on agents and on tasks.

MOTS-CLÉS : Agents, Coalition, Stabilité

KEYWORDS : Agents, Coalition, Stability

1. Introduction

Les réseaux de Smart-phones, de robots et de capteurs qui améliorent la sécurité et les prises de décision des humains, sont de plus en plus étudiés. De nombreux mécanismes de formation de coalitions ont été étudiés dans le domaine des agents [2], cependant, un ensemble d'aspects intéressants qui sont sous-jacents à l'utilisation des réseaux ad-hoc pour le déploiement des agents, restent faiblement explorés [7] [8]. [1] propose une application utilisant les réseaux de capteurs pour la surveillance sans tenir compte de la possibilité d'indisponibilité de certains capteurs et lorsque les informations sont complètes. [9] considère la dynamique des tâches mais avec des agents homogènes et coopératifs. Leur méthode de formation de coalitions est basée sur les MDP mais sans tenir compte de la possibilité d'indisponibilité des agents. Il utilise des connaissances a priori sur les préférences des agents et sur la dynamique des tâches pour guider le choix des coalitions à former sans garantir leurs stabilités. Notre contribution principale est la proposition d'un mécanisme de formation de coalitions *core stables* qui est adaptatif, décentralisé et asynchrone. Plus en détail, nous proposons un mécanisme de formation de coalitions appelé *S-NRB* qui se fonde sur des négociations multilatérales et qui tient compte - des incertitudes sur les dépendances qui existent entre les agents, - des préférences des agents et de leur disponibilité afin que les coalitions requises soient stables pour l'exécution des tâches stochastiques. En distribuant le contrôle et les prises de décisions des agents, *S-NRB* garantit une convergence vers des coalitions *core stables* malgré les changements dynamiques du contexte des agents (tâches et voisinages). *S-NRB* associe : - les lois de probabilités pour prédire les changements sur les tâches et sur la disponibilité des agents, et - les méthodes de la théorie des jeux pour trouver l'ensemble des agents qui peuvent former des coalitions *core stables* en profitant des dépendances et de la structure en réseau des agents. Le contexte applicatif est un environnement de sinistre où les agents sont déployés dans des nœuds ad-hoc (e.g. PDAs, capteurs, Smart-phones) qui ont des disponibilités aléatoires, où il est impossible de prédire a priori les interdépendances des agents et les stratégies de collaboration à établir avant le début de l'exécution des tâches stochastiques. Nous considérons aussi que, le contexte est ouvert (un nœud ad-hoc peut aléatoirement quitter ou rejoindre l'environnement de déploiement). Les interdépendances peuvent provenir du fait que des intervenants de spécialités différentes (e.g. pompiers, organismes de secours, volontaires, etc.) qui hébergent des agents dans leur kit de secours, doivent prendre des décisions en fonction de leur groupe d'appartenance et des informations contextuelles. Il est clair que dans un tel contexte, ces composants ad-hoc intégrés dans les kits d'assistances ne peuvent pas avoir des ressources énergétiques et une puissance de calcul illimitée. De plus, ils doivent tenir compte de la portée limitée de leur signal et de leur indisponibilité temporaire ou définitive (e.g. pannes, dysfonctionnement électronique, etc.). Par conséquent, il n'est pas abusif de considérer que ces composants électroniques et ces intervenants n'ont pas de connaissances a priori sur les tâches à exécuter ou sur les coalitions (équipes) à former.

2. Concepts de notre modèle

Nous définissons une tâche stochastique comme un ensemble d'actions non-déterministes à exécuter pour résoudre un problème qui peut changer d'états d'un instant à l'autre, par exemple, les actions à effectuer pour éteindre un feu qui peut s'amplifier, rester

stationnaire ou régresser. Le but d'un agent est d'exécuter un ensemble d'actions non-déterministes pour maximiser son utilité. Formellement, soit A=$\{a_1, ..., a_n\}$ un ensemble des agents de l'environnement multi-agents où chaque $a_i \in A : i \in [1, |A|]$ est un agent égoïste déployé dans un unique composant ad-hoc. Par C, nous définissons une coalition telle que C=$\{A_c, G_c, T_c, V_c\}$, où $A_c \subset A$=$\{a_1, a_2, ..., a_k\}$ est un ensemble d'agents qui partagent la coalition C et qui ont comme ensemble de buts $G_c \subseteq \{G_{a_i} : i \in N, a_i \in A_c\}$. T_c définit une tâche stochastique à exécuter par la coalition C et V_c est la récompense associée à la coalition C après l'exécution de T_c. En raison du contexte incertain et dynamique considéré, les paramètres essentiels que nous prenons en compte pour un agent a_i s'expriment à un instant t sous la forme : $\{R_{a_i}, Aut^t_{a_i}, Hs_{a_i}, \vartheta^t_{a_i}, U_{a_i}, L^{Net}_{a_i}\}$. Pour l'agent a_i, l'information R_{a_i} décrit sa ressource et $Aut^t_{a_i}$ son autonomie énergétique à l'instant t. Hs_{a_i} représente l'*historique* de ses interactions avec les autres agents. $\vartheta^t_{a_i}$ est la vue de l'agent a_i à l'instant t. U_{a_i} est la fonction d'utilité de a_i et $L^{Net}_{a_i}$ définit, à un instant t, le niveau de dépendance de l'agent a_i par rapport aux autres agents de son voisinage $\vartheta^t_{a_i}$ dans une structure réseau notée Net.

Définition 1

La fonction d'utilité U_{a_i} d'un agent a_i s'exprime sous la forme U_{a_i}=$\sum_{c=1}^{\infty} u^{a_i}_c$ où $u^{a_i}_c$ est son utilité suite à sa participation à la coalition C (cf. équation 1) . $U^t_{a_i}$ est la valeur de la fonction d'utilité de l'agent a_i à l'instant t.

La récompense d'un agent a_i dans une coalition C est notée v_{a_i}, celle de la coalition C s'exprime par V_c=$\sum_{a_i \in c} v_{a_i}$.

Définition 2

La fonction d'utilité qu'un agent a_i cherche à maximiser en participant dans la coalition C s'écrit :

$$u^{a_i}_c = v_{a_i} - Cost^{a_i}_c \tag{1}$$

Cette fonction d'utilité est une information privée vue que la fonction de coût $Cost^{a_i}_c$ est aussi privée. La seule connaissance partagée par les agents est la récompense V_c.

Définition 3

Soit $L^{Net}_{a_i}$ les dépendances de a_i avec les agents de sa vue $\vartheta^t_{a_i}$ qui forment une structure réseau Net telle que $\vartheta^t_{a_i} \subseteq Net$. $\forall a_i \in Net, \exists\, L^{Net}_{a_i}$ telle que $L^{Net}_{a_i} = \{\gamma^{Net}_{a_i}, H^{Net}_{a_i}, S^{Net}_{a_i}\}$. $\gamma^{Net}_{a_i}$ est l'ensemble des agents dont dépend a_i. $S^{Net}_{a_i}$ est l'ensemble des agents qui dépendent de a_i. $H^{Net}_{a_i}$=$(\vartheta^t_{a_i} \setminus \gamma^{Net}_{a_i}) \setminus S^{Net}_{a_i}$ est l'ensemble des agents qui ont la même importance que a_i dans l'environnement de déploiement.

Cette formulation des dépendances possibles pour un agent s'explique par le fait que, dans un environnement sinistré, des intervenants peuvent être sous les ordres d'un ensemble de leaders ou peuvent avoir des accords antérieurs auxquels ils doivent tenir compte.

3. Mécanisme de formation de coalitions : S-NRB

S-NRB fournit un ensemble de stratégies d'interactions que les agents utilisent afin de trouver le ou les meilleures coalitions A-cores dans un contexte incertain.

Définition 4

Soit une coalition C et E_C un ensemble de coalitions. Si $C \in E_C$ est une coalition A-core, alors :

- Il n'existe aucun agent ou groupe d'agents qui a la possibilité d'augmenter son utilité en changeant de coalition, c'est-à-dire, $\forall\, a_i \in C$ et $\forall\, C, C' \in E_C, u^{a_i}_c \geq u^{a_i}_{c'}$.

- Tout agent de C a une alliance non-dominée avec au moins un agent de C, c'est-à-dire, $\forall\, a_i \in C, \exists Al_{a_i,aj} : a_j \in C$.

- Les agents de C forment un ensemble connexe, ($\forall\ a_i \in C, \exists a_j \in \vartheta^t_{a_i} : a_j \in C$.)
À cause des incertitudes et de la dynamique du contexte considéré, *S-NRB* se fonde sur un principe de sondage parallèle et asynchrone et sur des offres composées.
Définition 5
Une offre composée pour la formation d'une coalition C d'un agent a_i est notée $O^{a_i}_c$. Formellement, $O^{a_i}_c$=$\{\{B_i, B_j, B_k, ...\}, \{pl_{a_i}\}\}$: B_i=$(Ri, \Delta i, Ui)$. B_i est un but à atteindre par l'offre qui spécifie, la ressource Ri requise pour T_c, les contraintes Δi de B_i et la récompense Ui associée à B_i. pl_{a_i}=$\{a_i, a_j, a_k, ...\}$: $a_i \in A$ désigne une liste de sondage qui permet de savoir dynamiquement, les agents qui acceptent de s'associer pour un but de l'offre de a_i.
Définition 6
Deux agents a_j et a_k sont en conflit pour une offre composée $O^{a_i}_c$, lorsque a_j et a_k ont la même contribution marginale (X_{a_j}=X_{a_k}), le même but $B_x \in O^{a_i}_c$ et veulent tous les deux participer à la même coalition C.
Notations
$\equiv$ L'équivalence entre deux paramètres. $\not\equiv$ La non équivalence entre deux paramètres.
$\succsim_P$ Dominance au sens de Pareto. $\succsim_L$ Dominance au sens de Lorenz.

3.1. Les étapes de notre mécanisme S-NRB

Le mécanisme *S-NRB* (cf. algorithme 1), se résume en ces quatre étapes :
Étape 1 : Stratégies de diffusion des offres de formation de coalition C.
Tout agent a_j qui reçoit une offre $O^{a_i}_c$ pour former C doit envoyer au maximum un message avec un TTL=1 (cf. équation 6) dans son voisinage $\vartheta^t_{a_j}$. Ce message doit indiquer sa réponse à l'agent par qui il a reçu cette offre. Nous imposons $TTL = 1$ pour permettre aux agents de privilégier les agents de leur vue et pour simplifier les négociations.
- Un agent a_j ne diffuse le message de sondage contenant l'offre $O^{a_i}_c$ dans $\vartheta^t_{a_j}$ que s'il accepte de participer à C (cf. ligne 1 - 4 algorithme 1). Cet envoi du message est effectué après avoir vérifié que son but $G_{a_j} \equiv B_i : B_i \in O^{a_i}_c$, que son utilité $U^t_{a_j}$ à cet instant t est supérieure à son utilité $U^{t-1}_{a_j}$ à l'instant $t-1$ et après avoir inscrit son identifiant dans la liste de sondage $pl_{a_i} \subset O^{a_i}_c$.
À noter que, si un agent accepte une offre, il supprime de cette offre l'information sur son but $G_{a_j} \equiv B_i : B_i \in O^{a_i}_c$ afin de n'envoyer dans son voisinage que les buts qui restent à atteindre et pour réduire le nombre de conflits de buts à satisfaire, par la coalition à former, qui peuvent apparaître entre les agents. De plus, cette suppression des informations sur les buts déjà acquis permet dans une certaine mesure, d'accélérer la convergence du mécanisme et de favoriser les agents proches de l'agent qui a initié l'offre de coalition.
- Un agent peut répondre à l'offre par un rejet ($ResetProbe(O^{a_i}_c)$, cf. ligne 26 - 28 algorithme 1) ou par une demande de modification de l'offre (*counter($O^{a_i}_c$)*, cf. ligne 23 - 25 algorithme 1). C'est seulement dans le cas d'une contre-proposition, qu'un agent peut entrer en négociation, si par exemple, il n'y a pas suffisamment d'agents qui ont accepté l'offre ou si la contre-proposition n'engendre pas d'incohérence sur l'offre $O^{a_i}_c$ proposée.
Étape 2 : Stratégies d'optimisation des messages envoyés.
$\forall\ a_j$ tel que, $\forall\ a_k \in \vartheta^t_{a_j}, a_k \in pl_{a_i}$ (c'est-à-dire, un agent avec des voisins déjà dans la liste de sondage) ou $\vartheta^t_{a_j} = \emptyset$ (c'est-à-dire, un agent sans voisin), a_k doit rendre sa décision (acceptation ou contre-proposition ou rejet) pour mettre fin au sondage sur le chemin de recherche auquel il appartient.
À cause des contraintes des équipements (e.g. portée du signal, débit des liens de communications) où les agents sont hébergés, parfois, un agent ne peut pas contacter au même

moment tous les autres agents de l'environnement de déploiement. Comme conséquence, plusieurs listes de sondage pl_{a_i}, pl'_{a_i}, $pl"_{a_i}$, ..., peuvent coexister, d'où la nécessité de la détection dynamique des listes de sondage redondantes.

Étape 3 : Assurer qu'une offre n'a qu'une liste de sondage.

Tout agent qui a reçu plus d'une liste de sondage et qui veut participer à C doit effectuer une fusion des différentes listes de sondage afin de limiter les conflits entre les agents et de diffuser une seule liste de sondage dans sa vue (cf. ligne 11 - 14 algorithme 1). Soit $pl^1_{a_i}, pl^2_{a_i}, pl^3_{a_i}$ un ensemble de 3 listes de sondage pour une même offre $O^{a_i}_c$. La fusion signifie la production d'une seule liste de sondage pl_{a_i} résultante qui ne contient pas de conflits entre les agents. Formellement, pl_{a_i}=$(pl^1_{a_i} \cup pl^2_{a_i} \cup pl^3_{a_i})$: $(pl^1_{a_i} \cap pl^2_{a_i} \cap pl^3_{a_i}) = \emptyset$ et $\forall a_x, a_y \in pl_{a_i}$, $a_x \neq a_y$, alors $G_{a_x} \not\equiv G_{a_y}$.

Étape 4 : Prise d'une décision collective sur l'état du sondage.

Si $(\cup Bi \subset O^{a_i}_c) \neq (\underset{a_j \in pl_{a_i}}{\cup} G_{a_j})$, alors les agents qui sont dans la liste cherchent à prédire s'ils sont capables de former une coalition stable et dans quelle mesure y parvenir (cf. ligne 20 - 22 algorithme 1). Pour ce faire, ils utilisent la loi de probabilité hypergéométrique [3], [4] . Cette loi de probabilité est la loi a priori pour la prédiction de l'issue probable d'un sondage. Ainsi, $\forall\ a_j \in pl_{a_i}$, *échantillon*=$n$=$Card(pl_{a_i})$, *population*=$N$=$Card(\cup B_i)$: $B_i \in O^{a_i}_c$. Soit $P_{counter}$ la probabilité d'avoir un agent qui accepte de rejoindre la liste de sondage après avoir renégocié l'offre. Soit P_{reject} la probabilité d'avoir un agent qui rejette l'offre. $\forall\ a_j \in pl_{a_i}$: $a_x \in \vartheta^t_{a_j}, a_x \notin pl_{a_i}$, il calcule $Q_{counter}$=$1 - (P_{counter} + P_{reject})$ c'est-à-dire la probabilité de contacter un agent qui accepte sans renégocier l'offre. Soit k le nombre de fois qu'il est encore nécessaire de contacter des agents pour former C, alors la probabilité de X=k est donnée par :

$$P[X = k] = \frac{C^k_{NQ_{counter}} C^{n-k}_{NP_{counter}}}{C^n_N} \tag{2}$$

L'espérance de cette probabilité est :

$$E(X) = nQ_{counter} \tag{3}$$

La décision de poursuivre leurs recherches dépend des valeurs de ces indicateurs. En effet, si $P[X = k] \neq 0$, l'agent responsable de la poursuite du sondage est celui de la liste de sondage qui domine au sens de Lorenz les autres agents de cette liste si on considère les paramètres $P[X = k]$ et $E(X)$. La dominance de Lorenz à été proposée par [5].

Considérons deux agents a_j et a_k, tel que :

pour a_j, notons la probabilité $P[X = k]$ par $P_j[X = k]$ et l'espérance $E(X)$ par $E_j(X)$.

pour a_k, notons la probabilité $P[X = k]$ par $P_k[X = k]$ et l'espérance $E(X)$ par $E_k(X)$.

La dominance de Lorenz pour faire un choix entre ces deux agents a_j et a_k, en utilisant $P[X = k]$ et $E(X)$, revient à poser comme vecteur de Lorenz généralisé associé :

pour a_j, $L(x)$=$(x_1, x_1 + x_2)$=$(P_j[X = k], P_j[X = k] + E_j(X))$ et

pour a_k, $L(y)$=$(y_1, y_1 + y_2)$=$(P_k[X = k], P_j[X = k] + E_k(X))$.

Si $\sum_{i=1}^{2} L_i(x) \geq \sum_{i=1}^{2} L_i(y)$, alors $x \succsim_L y \Leftrightarrow L(x) \succsim_P L(y)$. Cela, signifie que x domine y au sens de Lorenz et donc a_j est responsable de la poursuite du sondage.

Notons que le premier identifiant d'une liste de sondage (e.g. pl_{a_i}) est celui de l'agent a_i qui a initié l'offre $O^{a_i}_c$. Cela, permet de différencier les offres.

Mécanisme de résolution des conflits.

Ce mécanisme que nous notons S-NRB-CRP (Sequential Non-Return Broadcast Conflit

Resolution Protocol) est fondé sur :
Étape 1 : Matching(comparaison) et mesure de dissemblances.
L'indice et la distance de Jaccard [6] sont deux métriques utilisées pour évaluer la similarité et la diversité entre des ensembles. Soit n ensembles S_1, S_2, ..., S_n, l'indice de Jaccard est :

$$J(S_1, S_2, ..., S_n) = \frac{Card(S_1 \cap S_2 \cap ... \cap S_n)}{Card(S_1 \cup S_2 \cup ... \cup S_n)} \quad (4)$$

La distance de Jaccard mesure la dissemblance entre les ensembles. Elle consiste à soustraire l'indice de Jaccard à 1, c'est-à-dire, $J_\delta(S_1, S_2, ..., S_n) = 1 - J(S_1, S_2, ..., S_n)$.

$$J_\delta(S_1, S_2, ..., S_n) = \frac{Card(S_1 \cup S_2 \cup ... \cup S_n) - Card(S_1 \cap S_2 \cap ... \cap S_n)}{Card(S_1 \cup S_2 \cup ... \cup S_n)} \quad (5)$$

À la ligne 4 de l'algorithme 2, $Jacc(a_j)$ et $Jacc(a_k)$ signifie que a_j et a_k doivent donner respectivement leur liste $L_{a_j}^{Jacc}$ et $L_{a_k}^{Jacc}$ qui contiennent respectivement l'ensemble des agents des listes de sondage pl_{a_i} de a_j et pl'_{a_i} de a_k avec qui, ils ont des accords (alliances, dépendances) pour participer à C. Si nous notons par $AllianceSet^{a_x}$ l'ensemble des agents alliés d'un agent a_x, alors : $L_{a_j}^{Jacc}$=$(\gamma_{a_j}^{Net} \cup S_{a_j}^{Net} \cup AllianceSet^{a_j}) \cap pl_{a_i}$.
$L_{a_k}^{Jacc}$=$(\gamma_{a_k}^{Net} \cup S_{a_k}^{Net} \cup AllianceSet^{a_k}) \cap pl_{a_i}$.
$Jaccard(a_j, a_k)$ est l'indice de Jaccard entre les deux ensembles $L_{a_j}^{Jacc}$ et $L_{a_k}^{Jacc}$.
Étape 2 : Prise de décision en se fondant sur la dominance au sens de Lorenz.
Il utilise les ensembles $L_{a_j}^{Jacc}$ et $L_{a_k}^{Jacc}$, sur les fiabilités ρ_{a_j} et ρ_{a_k} et sur les probabilités de stabilité $P_s^{a_j}$ et $P_s^{a_j}$. Si $L(x)$=$(x_1, x_1+x_2, x_1+x_2+x_3)$=$(Card(L_{a_j}^{Jacc}), Card(L_{a_j}^{Jacc})+\rho_{a_j}, Card(L_{a_j}^{Jacc})+\rho_{a_j}+P_s^{a_j})$ et $L(y)$=$(y_1, y_1+y_2, y_1+y_2+y_3)$=$(Card(L_{a_k}^{Jacc}), Card(L_{a_k}^{Jacc})+\rho_{a_k}, Card(L_{a_k}^{Jacc}) + \rho_{a_k} + P_s^{a_k})$. Alors, si $\sum_{i=1}^{3} L_i(x) \geq \sum_{i=1}^{3} L_i(y)$, alors $x \succsim_L y \Leftrightarrow L(x) \succsim_P L(y)$ signifie que x domine y au sens de Lorenz. Alors, la participation de a_j est préférée à celle de a_k.

3.2. Analyse de S-NRB

Nous donnons un ensemble de propriétés de S-NRB qui mène à l'existence de coalitions *A-core*.

Lemme 1 *Qu'une possibilité de former une coalition core stable existe ou non, S-NRB termine toujours sans blocage.*

Lemme 2 *S'il est possible de former une coalition core stable, S-NRB permet aux agents de converger vers celle-ci.*

Théorème 1 *S-NRB permet toujours la converge vers une coalition core stable si celle-ci existe.*

Lemme 3 *Le processus de sondage de notre mécanisme S-NRB fait émerger une structure connexe d'agents pl_{a_i} où chaque agent a_j de cette structure a au moins un agent voisin a_k dans pl_{a_i} avec lequel il est accepte de partager la coalition C.*

Lemme 4 *Les agents les plus proches et les plus fiables du voisinage de l'agent qui a initié le mécanisme de formation de coalitions S-NRB ont toujours une plus grande probabilité d'être choisi pour former la coalition.*

Théorème 2 *S-NRB permet une auto-stabilisation des coalitions core stables.*

4. Évaluations de performances de notre mécanisme S-NRB

Nous effectuons ces simulations sur notre plateforme basée sur JADE (JAVA Agent DEvelopment Framework). Les indisponibilités, les préférences et les dépendances des agents et les états des tâches suivent des distributions de probabilités de loi Uniforme ou Normale. Ces deux lois sont les lois a priori à chaque fois qu'on ignore l'ensemble des situations qui peuvent justifier l'apparition de certains évènements. Aléatoirement, un agent autonome et égoïste avec des dépendances natives ou non peut initier un processus de formation de coalitions, être libre de toute dépendance ou devenir indisponible. La dynamique de la localisation, les propriétés et les états des tâches sont générées aléatoirement de même que pour les ressources nécessaires pour l'exécution d'une tâche à un instant t donné. Le nombre d'agents dans l'environnement est choisi entre 10 et 100 et le nombre de tâches simultanées est compris entre 3 et 8 tâches. Les simulations sont effectuées dans une machine Intel i7 (4 GHZ) avec 4GB de RAM. À cause de la dynamique des ressources disponibles ou requises pour une tâche, nous faisons nos simulations en considérant les *ratios*. Par exemple, s'il y a 12 agents de même caractéristiques. Si l'objectif est de former deux coalitions où la première requiert 9 agents et la seconde 6 agents, alors nous avons, *ratio*=$\frac{12}{9+6}$=$\frac{12}{15}$=0.8. Pour obtenir des valeurs moyennes, nous répétons 10 fois chaque simulation. Nous étudions ici, la capacité des coalitions *core stables* formées

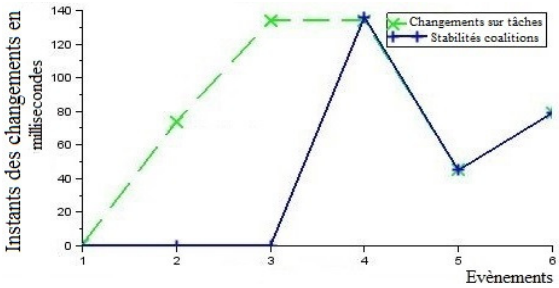

Figure 1. *Auto-stabilisation des coalitions si les changements d'états des tâches et la disponibilité des agents suivent une distribution de probabilités Uniforme* $U(a,b)$ *où* a=0 *et* b=10. *Évolutions des* $ratios$ *: 1.42, 1.66, 2, 3.33, 1.66 et 2.*

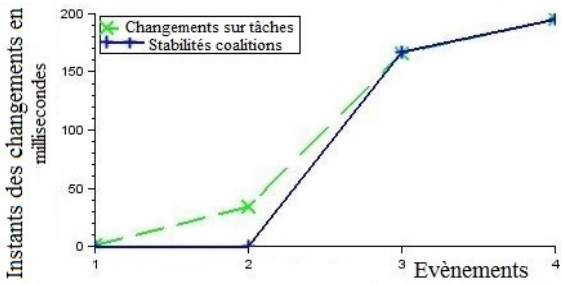

Figure 2. *Auto-stabilisation des coalitions si les changements d'états des tâches et la disponibilité des agents suivent une distribution de probabilités Normale* $N(\mu,\sigma^2)$ *où* μ=0.5 *et* σ=0.1. *Évolutions des* $ratios$ *: 0.90, 3.33, 3.33, 2.50.*

à s'auto-stabiliser. L'axe des abscisses (*Evènements*) répertorie l'ensemble des changements sur les $ratios$ et sur les tâches. Les résultats des simulations illustrés par ces figures montrent que, si une coalition est *core stable*, elle le reste même s'il y a des changements dynamiques sur les $ratios$ et les tâches.

5. Conclusion

Ce travail aborde la mise en œuvre d'un mécanisme de formation de coalitions *core stables* adaptatif, asynchrone et décentralisé qui permet une auto-stabilisation des coalitions formées d'agents égoïstes (coalitions *A-cores*). Pour ce faire, nous avons proposé le mécanisme S-NRB. Nous avons, principalement, prouvé de manière théorique et expérimentale, sa convergence vers des coalitions *A-cores* en cas de disparition d'un ensemble d'agents ou de changements sur les contraintes des tâches et en présence d'incertitudes sur les dépendances, les conflits, les préférences et la disponibilité des agents. À cause des incertitudes et des contraintes, nous ne supposons aucune connaissance a priori sur les coalitions stables à former pour les tâches stochastiques à exécuter et qu'il n'est pas possible de calculer de manière centralisée les coalitions à former et qu'il est impossible de prédire la stabilité des coalitions avant le début de l'exécution des tâches.

6. Bibliographie

[1] B. HORLING, R. VINCENT, R. MAILLER, J. SHEN, R. BECKER , K. RAWLINS , V. LESSER, « Distributed sensor network for real time tracking », *AGENTS'01 Autonomous Agents,ACM Press*, 417-424, 2001.

[2] O. SHEHORY, S. KRAUS, « Methods for Task Allocation via Agent Coalitions Formation », *AI Journal*,May, 165-200, 1998.

[3] R. D. YATES, D. J. GOODMAN, « Probability and Stochastic Processes : A Friendly Introduction for Electrical and Computer Engineers », *John Wiley and Sons, INC*, 2005.

[4] B. BAYNAT, « Théorie Des Files D'attente - Des Chaînes De Markov Aux Réseaux À Forme Produit », *Hermes Science Publications - Lavoisier*, June, 1-328, 2000.

[5] A. W. MARSHALL, I OLKIN, « Inequalities : Theory of Majorization and Its Applications », *The Mathematics in Science and Engineering Series*, n° 143, 1-569, 1979.

[6] K. JAHANBAKHSH, V. KING, G. C. SHOJA, « Predicting missing contacts in mobile social networks », *Pervasive and Mobile Computing, Elsevier*, n° 8, issue 5, 698-716, October, 2012.

[7] P.F. FAYE, S. AKNINE, O. SHEHORY, M. SÈNE, « Stabilizing Agent's Interactions in Dynamic Contexts », *AINA 2014*, May, 925-932, 2014.

[8] P.F. FAYE, S. AKNINE, M. SÈNE, O. SHEHORY, « Dynamic Coalitions formation in Dynamic Uncertain Environments », *IAT 2015*, December, 273-276, 2015.

[9] M. A. KHAN, D. TURGUTL. BöLöNI, « Optimizing coalition formation for tasks with dynamically evolving rewards and nondeterministic action effects », *Autonomous Agents and Multi-Agent Systems*, May, 415-438, 2011.

7. Annexe A

Preuve du lemme 1
Soit $U^t_{a_j}$ l'utilité de a_j à l'instant t. $\forall Probe(O^{a_i}_c)$ d'un agent a_i, un agent $a_j \neq a_i$ ne diffuse ce message de sondage que si $U^{t-1}_{a_j} \leq U^t_{a_j}$ et $\vartheta^t_{a_j} \neq \emptyset$, ce qui est une manière de gérer les coûts des communications et d'éviter les négociations en boucles. Tout conflit entre a_i et a_j est géré par les autres agents appartenant à leur liste de sondage grâce au calcul de la dominance de Lorenz entre a_i et a_j. Cette méthode évite le cas où a_i et a_j sont dans une impasse au moment où les agents de leur liste attendent la validation de la liste de sondage en une coalition. Ainsi, les blocages sont évités grâce à cette gestion des communications et la résolution décentralisée des conflits. Ce qui prouve notre lemme.□
Preuve du lemme 2
Soit $U^t_{a_i}$ l'utilité de a_i à l'instant t et $U^t_{a_j}$ l'utilité de a_j à l'instant t. $\forall B_i \in G_c$ et $B_i \in O^{a_i}_c$, si $\exists a_i, a_j \in A : R_{a_i} \in B_i, Aut^t_{ai} \neq 0, U^{t-1}_{a_i} \leq U^t_{a_i}$ et $R_{a_j} \in B_i, Aut^t_{aj} \neq 0, U^{t-1}_{a_j} \leq U^t_{a_j}$, alors, a_i (respectivement a_j) va atteindre un accord en utilisant ses dépendances $L^{Net}_{a_i}$ (respectivement $L^{Net}_{a_j}$). Cela se vérifie à la ligne 1 de l'algorithme 2. L'utilisation de l'index de Jaccard afin d'évaluer les différences entre les agents a_i et a_j, par rapport à leurs alliances et à leurs fiabilités, ainsi que la dominance de Lorenz permettent de faire un choix entre les deux agents. Si la liste de sondage pl_{a_i} est validée comme une coalition C, cela signifie que :
(1) $\forall a_i \in pl_{a_i}, \exists a_j \in pl_{a_i} : a_j \in \gamma^{Net}_{a_i}$ ou $a_j \in H^{Net}_{a_i}$ ou $a_j \in S^{Net}_{a_i}$;
(2) $\forall a_i \in pl_{a_i}, U^{t-1}_{a_i} \leq U^t_{a_i}$;
(3) $\forall a_i \in pl_{a_i}, \exists a_j \in pl_{a_i} : \exists Al_{a_i,a_j}$.
De là, C sera *core stable* car aucun agent $a_i \in pl_{a_i}$ ne sera motivé à quitter C pour une autre coalition C' et risquer ainsi de perdre ses alliance(s), de réduire son utilité et être considéré comme non fiable par les autres agents. Ce qui prouve notre lemme.□
Preuve du théorème 1
Le lemme 1 montre que S-NRB termine son exécution et le lemme 2 prouve que S-NRB mène à la formation de coalitions *core stables* si ces dernières peuvent exister entre les agents. Ainsi, nous pouvons affirmer que S-NRB cherche toujours à déterminer une coalition stable qui maximise le bien-être social des agents quels que soient les états initiaux des agents. Ce qui prouve notre théorème.□
Preuve du lemme 3
$\forall a_j \in A$, s'il reçoit une offre $O^{a_i}_c$, cela signifie que, $a_j \in \vartheta^t_{a_j}$ ou $\exists \{a_x, a_y, ...\} \in A : a_j \in L^{Net}_{a_i}$ et $a_i \in L^{Net}_{a_j}$. Ainsi, si a_i et $a_j \in pl_{a_i}$ alors, $a_j \in \vartheta^t_{a_i}$ ou $\exists \{a_x, a_y, ...\} \in pl_{a_i}$ car dans notre approche a_x ne diffuse pl_{a_i} que s'il appartient à pl_{a_i} (cf. lignes 3 - 5 algorithme 1) ou s'il existe un message de mise à jour de pl_{a_i} ($UpdateProbe(O^{a_i}_c)$). Dans ce dernier cas, si un agent a_j souhaite atteindre un but $B_i \in O^{a_i}_c$ il doit appartenir à $\vartheta^t_{a_i}$ ou à $\vartheta^t_{a_x} : a_x \in pl_{a_i}$. Ce qui prouve notre lemme.□
Preuve du lemme 4
Considérons les agents a_i, a_j et a_k tels que $a_j \in \vartheta^t_{a_i}$ et $a_k \in \vartheta^t_{a_j}$. Si $G_{a_j} \equiv G_{a_k} \equiv B_i \in O^{a_i}_c$ et $U^{t-1}_{a_j} \leq U^t_{a_j}$ et $U^{t-1}_{a_k} \leq U^t_{a_k}$ alors, a_j et a_k ne seront pas en conflits car si a_j reçoit l'offre $O^{a_i}_c$, il ajoute son identifiant dans la liste de sondage pl_{a_i} et supprime l'information concernant $G_{a_j} \equiv B_i \in O^{a_i}_c$ avant d'envoyer l'offre $O^{a_i}_c$ à $a_k \in \vartheta^t_{aj}$.
Ainsi, si a_k reçoit $O^{a_i}_c$ alors l'information $G_{a_k} \equiv B_i \notin O^{a_i}_c$ et donc aucun conflit n'arrive entre a_j et a_k. Ce qui prouve qu'un agent plus proche est toujours privilégié s'il faut former une coalition. Maintenant supposons que, $a_j, a_k \in \vartheta^t_{a_i}$ alors, les deux agents vont recevoir l'offre $O^{a_i}_c$ en même temps.

Si $G_{a_j} \equiv G_{a_k} \equiv B_i \in O_c^{a_i}$ et $U_{a_j}^{t-1} \leq U_{a_j}^t$ et $U_{a_k}^{t-1} \leq U_{a_k}^t$ alors, a_i, a_j et a_k vont procéder à la résolution du conflit en utilisant l'algorithme 2. Cela mène à la détermination de l'agent le plus fiable entre a_j et a_k. Ce qui donne la preuve de notre lemme.□

Preuve du théorème 2

Le lemme 2 prouve la convergence de S-NRB vers des coalitions *core stables* où aucun agent n'est motivé à quitter sa coalition. Le lemme 3 montre que, chaque évènement qui impacte dynamiquement l'état d'une (des) tâche(s) ou la disponibilité d'un ensemble d'agents sera détecté par au moins un agent de la liste de sondage ou de la coalition. Les lemmes 1 montrent qu'après une instabilité d'une coalition, cette dernière redevient stable après un nombre fini d'étapes. De plus, pour tout agent ajouté dans une coalition *core stable*, les conflits éventuels sont gérés de manière décentralisée et la décision d'ajouter un ensemble d'agents dans une coalition se fait en respectant les préférences des agents qui sont déjà dans la coalition. Avec ces contraintes de participation dans une coalition, notre mécanisme S-NRB évite le cas où un ensemble d'agents est motivé à quitter sa coalition après une instabilité causée par un ensemble de conflits ou à cause d'une décroissance de leur utilité. Ce qui prouve notre théorème.□

8. Annexe B

Règle 1 : offre et participation valide. Un agent a_i peut proposer une offre $O_c^{a_i}$ si et seulement si $a_i \in C$ et chaque but $B_i \in O_c^{a_i}$ est requis pour la formation et la stabilité de C. a_i peut conseiller (cf. règle 2) un autre agent sur les modifications à apporter sur son offre pour avoir des participants potentiels à la coalition C. Un agent a_j peut accepter $B_i \equiv G_{a_j} \in O_c^{a_i}$ si et seulement si $Ri \in B_i : Ri \in R_{a_j}$. De plus, a_j ne peut participer à C que si $\forall a_i \in C, u_{c'}^{a_i} \geq u_c^{a_i} : C'$=$C \cup a_j$.■

Règle 2 : Conseiller un agent. Les conseils de modification des offres ont pour objectif de permettre aux agents d'améliorer leurs connaissances sur les autres agents et sur les offres pouvant les intéresser. a_j peut conseiller a_i ($Counsel(a_i)$) sur des modifications à apporter sur $O_c^{a_i}$, si Al_{a_i,a_j} existe, si $G_{a_i} \in O_c^{a_i}$ et si cela peut éventuellement intéresser un agent appartenant à $L_{a_j}^{Net} \neq \emptyset$. Ainsi :

(1) Un conseil doit assurer que la modification de l'offre pour intéresser un agent a_j à C ne va pas diminuer l'utilité d'un agent qui a déjà donné son accord pour participer à C, c'est-à-dire, $\forall a_i \in C\ u_{c'}^{a_i} \geq u_c^{a_i} : C'$=$C \cup a_j$,

(2) a_j peut conseiller a_i s'il existe une *alliance* Al_{a_i,a_j}, sinon a_i ignore le conseil de a_j.

(3) L'expression formelle d'un conseil est sous la forme : $Counsel(a_i)$=$(O_c^{a_i}, B_i)$, qui signifie que le conseil de modification de l'offre $O_c^{a_i}$ de a_i concerne le but $B_i \in O_c^{a_i}$.■

Règle 3 : Acceptation d'une offre. a_j peut accepter une offre si $B_i \equiv G_{a_j} \in O_c^{a_i}$ et $Ri \in B_i : Ri \in R_{a_j}$. a_j peut participer dans C si $\forall a_i \in C, u_{c'}^{a_i} \geq u_c^{a_i} : C'$=$C \cup a_j$.

Une fois que a_j accepte l'offre de a_i, $B_i \equiv G_{a_j} \in O_c^{a_i} : a_i \in C$ et $a_j \in A \setminus C$, alors les deux agents mettent à jour leurs connaissances sur la(les) tâche(s) (état, localisation, etc.) et la stabilité de leur coalition C si elle est validé.

Si a_j formule une contre-proposition de l'offre $O_c^{a_i}$ telle que $B_i \equiv G_{a_j} \in O_c^{a_i} : a_i \in C$, alors a_j est automatiquement désengagé de tout accord précédent et doit refaire une négociation avec les autres agents ou avec a_i. Cependant, si la coalition a été validée, la fiabilité de a_j est d'abord réduite par tous les autres agents avant de vérifier si une autre négociation peut être entreprise avec a_j. Un agent peut se désengager d'un accord s'il doit pénaliser un autre agent de C ou s'il devient indisponible.■

Algorithme 1 : : S-NRB

Require: $Probe(O_c^{a_i}) : \vartheta_{a_i}^t \neq \emptyset, O_c^{a_i}$=$\{\{B_i : i \in [1, |T_c|]\}, \{pl_{a_i} : i \in [1, |A|]\}\}$
RESULT : Coalition C
if $G_{a_j} \equiv B_i \in O_c^{a_i}$ et $Aut_{aj}^t \neq 0$ et $U_{aj}^{t-1} \leq U_{aj}^t$ **then**
  if $\vartheta_{a_j}^t \neq \emptyset$ et $a_j \in pl_{a_i}$ **then**
    Envoyer $Probe(O_c^{a_i}) : pl_{a_i}$=$\{a_i, a_j\}$ $\forall$ $a_k \in \vartheta_{a_j}^t$
  end if
  if $\vartheta_{a_j}^t$==$\emptyset$ et $a_j \in pl_{a_i}$ **then**
    Envoyer $CommitProbe(O_c^{a_i}) : pl_{a_i}$=$\{a_i, a_j\}$ $\forall$ $a_k \in \vartheta_{a_j}^t$
  end if
  if $\vartheta_{a_j}^t \neq \emptyset$ et $a_j \in pl_{a_i}$ et $\exists$ $CommitProbe(O_c^{a_i})$ **then**
    Envoyer $Commit(pl_{a_i})$
  end if
  if ($\exists$ $a_j \in pl_{a_i}$) ou ($\exists$ $a_k \in pl'_{a_i}$) et ($a_k \notin pl_{a_i}$) **then**
    $UpdateProbe(O_c^{a_i})$ *//pour mettre à jour* pl_{a_i} *après fusion de* pl_{a_i} *et* pl'_{a_i}
  end if
  if $\exists$ conflit entre a_j et a_k **then**
    $Matching(pl_{a_i}, pl'_{a_i})$ *//pour initier l'exécution de S-NRB-CRP() (algorithme 2)*
    Envoyer $UpdateProbe(O_c^{a_i})$ *//pour mettre à jour* pl_{a_i}
  end if
  if $\cup Bi \subset O_c^{a_i}$==$\cup_{a_j \in pl_{a_i}} G_{a_j}$ **then**
    Valider C=$\{\forall$ $a_j \in pl_{a_i}\}$ et G_c=$\cup_{a_j \in pl_{a_i}} G_{a_j}$
  else
    $\forall$ $a_j \in pl_{a_i}$ trouver plus de ressources en utilisant $Al_{a_j,a_k} : a_k \in A$, L_{aj}^{Net} et $Counsel(a_i)$ *//Le* $Counsel(a_i)$ *est pour guider les recherches de* a_i *(cf. règle 2 section règles générales).*
  end if
else
  if $G_{a_j} \equiv B_i \in O_c^{a_i}$ et $Aut_{aj}^t \neq 0$ et $U_{aj}^{t-1} > U_{aj}^t$ **then**
    Envoyer $Counter(O_c^{a_i})$
  else
    Envoyer $ResetProbe(O_c^{a_i})$
  end if
end if

Règle 4 : Schéma de communication. Chaque agent peut formuler une offre, faire une contre-proposition ou conseiller la modification d'une offre à tout moment car il n'y a pas d'ordonnanceur. Cependant, pour : - assurer une convergence des négociations, - gérer la consommation de ressources, et - les offres en boucle, (1) toutes les communications se font en mode *non-return broadcast* et (2) chaque message à un *Time To Live (TTL)*. Le mode *non-return broadcast* signifie que lorsqu'une information de mise à jour (disparition d'agents de C, évolution sur une tâche, nouvelle offre, ...) qui arrive d'un agent a_k, ses agents voisins $\vartheta_{a_j}^t$ ne peuvent pas lui retourner la même information de mise à jour. Le *TTL* permet de définir le nombre de sauts autorisés pour un message.

$$1 \leq TTL \leq |\frac{Sz}{2 * \Upsilon}| \tag{6}$$

où Sz est la taille de l'environnement couvert par la tâche, Υ la portée du signal du composant qui héberge l'agent qui a initié le message (e.g. Bluetooth 100 mWatt, $\Upsilon \leq$100 mètres). Ainsi, le *TTL* permet de confiner les messages car un message n'est envoyé que si son *TTL* n'est pas épuisé. Cependant, même si un agent reçoit un message, il ne répond

ou ne le diffuse en mode *non-return broadcast* que selon ses propres stratégies qui visent à augmenter son efficacité et à ménager ses ressources.■

Algorithme 2 : : S-NRB-CRP()

Require: $\exists\,(a_j \in pl_{a_i})$ et $(a_k \notin pl'_{a_i})$ et G_{a_j}==G_{a_k} et X_{a_j}==X_{a_k}.
RESULT : $pl_{a_i} \cup pl'_{a_i}$: $pl_{a_i} \cap pl'_{a_i} = \emptyset$
1: **if** $a_j \in S^{Net}_{a_k}$ **then**
2: Notifier à a_j que sa demande de participation est annulée
3: **else**
4: $L^{Jacc}_{a_j}$=$Jacc(a_j)$ et $L^{Jacc}_{a_k}$=$Jacc(a_k)$
5: $Jaccard(a_j, a_k)$=$|\frac{L^{jac}_{a_j} \cap L^{jac}_{a_k}}{L^{jac}_{a_j} \cup L^{jac}_{a_k}}|$
6: **if** $1 - Jaccard(a_j, a_k) \neq 0$ **then**
7: **if** $Card(L^{Jacc}_{a_j}) < Card(L^{Jacc}_{a_k})$ **then**
8: Notifier à a_j que sa demande de participation est annulée
9: **else**
10: Notifier à a_k que sa demande de participation est annulée
11: **end if**
12: **else**
13: $L(x)$=$(x_1, x_1 + x_2, x_1 + x_2 + x_3)$=$(Card(L^{Jacc}_{a_j}), Card(L^{Jacc}_{a_j}) + \rho_{a_j}, Card(L^{Jacc}_{a_j}) + \rho_{a_j} + P^{a_j}_s)$
14: $L(y)$=$(y_1, y_1 + y_2, y_1 + y_2 + y_3)$=$(Card(L^{Jacc}_{a_k}), Card(L^{Jacc}_{a_k}) + \rho_{a_k}, Card(L^{Jacc}_{a_k}) + \rho_{a_k} + P^{a_k}_s)$
15: **if** $x \succsim_L y$ **then**
16: Notifier à a_k que sa demande de participation est annulée
17: **end if**
18: **if** $y \succsim_L x$ **then**
19: Notifier à a_j que sa demande de participation est annulée
20: **end if**
21: **end if**
22: **if** $\exists\,(a_j \in pl_{a_i})$ et $(a_k \notin pl'_{a_i})$ et G_{a_j}==G_{a_k} et X_{a_j}==X_{a_k} **then**
23: Sélectionner un agent tel que, $Max(\rho_{a_j}, \rho_{a_k})$, $Max(Aut^t_{a_j}, Aut^t_{a_k})$ et $Max(P^{a_j}_s, P^{a_k}_s)$.
24: **end if**
25: **end if**

Règle 5 : Terminaison d'une négociation. Une fois qu'un agent accepte ou reçoit une acceptation d'une offre, il doit interrompre la négociation en envoyant un message d'information à son voisinage pour faire part de sa décision ou du résultat de la négociation. Tout agent qui voit son offre refusée par un autre doit interrompre sa négociation avec cet agent et se tourner vers ceux susceptibles de l'accepter. Si un ensemble d'agents rivalisent pour la même offre, les agents retenus pour la coalition sont confirmés par un message de validation de participation tandis que les autres reçoivent un message de rejet. Ces messages mettent fin aux négociations. Ainsi, pour un agent dont la participation est rejetée, il doit interrompre la négociation afin d'avoir la possibilité de négocier une autre offre. Toute négociation a une durée au-delà de laquelle tout agent qui n'a pas un accord de participation pour son offre doit la modifier ou utiliser ses dépendances pour améliorer son offre. Cette offre est supprimée dans l'impossibilité de trouver des accords de participation à une coalition pour l'offre.■