



Modèles de formation de coalitions stables dans un contexte ad-hoc et stochastique

Pascal François Mbissane Faye

► To cite this version:

Pascal François Mbissane Faye. Modèles de formation de coalitions stables dans un contexte ad-hoc et stochastique. Autre [cs.OH]. Université Claude Bernard - Lyon I; Université Cheikh Anta Diop (Dakar), 2015. Français. NNT: 2015LYO10062 . tel-01167110

HAL Id: tel-01167110

<https://theses.hal.science/tel-01167110>

Submitted on 23 Jun 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

NUMÉRO D'ORDRE : 62-2015

UNIVERSITÉ CLAUDE BERNARD (UCBL), LYON, FRANCE

LIRIS

UNIVERSITÉ CHEIKH ANTA DIOP DE DAKAR (UCAD), SÉNÉGAL

LID

THÈSE EN COTUTELLE

Présenté en vue d'obtenir le grade de Docteur,
spécialité Informatique

par

Pascal François Mbissane FAYE

MODÈLES DE FORMATION DE COALITIONS STABLES DANS UN CONTEXTE AD-HOC ET STOCHASTIQUE.

Thèse soutenue le 27/05/2015 devant le jury composé de :

Pr.	PHILIPPE MATHIEU	Université de Lille 1	(Rapporteur)
Pr.	RENÉ MANDIAU	Université de Valenciennes	(Rapporteur)
Pr.	HONIDEN SHINICHI	National Institute of Informatics, Japon	(Examineur)
Pr.	HAMAMACHE KHEDDOUCI	Université de Lyon 1	(Invité)
Dr.	ONN SHEHORY	Technion Haifa IBM lab., Israël	(Invité)
Pr.	SAMIR AKNINE	Université de Lyon 1	(Directeur)
MC. HDR.	MBAYE SENE	Université Cheikh Anta Diop	(Co-directeur)

À qui veut croire aux positives attitudes ...

DÉDICACE

À mes très chers parents Raymond FAYE et Catherine Tine ; à ma très chère tante Jeanne TINE qui a grandement participé à mon éducation sociale ; à ma grand-mère Rose DIOP pour toutes ses prières ; à mon oncle Michel Ngagne TINE qui m'a toujours encouragé et soutenu dans mes études et dans ma vie ; à mes frères et sœurs avec qui j'ai grandi et qui partagent mes inquiétudes, mes peines, mes joies et mes espérances ; à ma nièce Thérèse Marie FAYE.

REMERCIEMENTS

JE commencerais par remercier Dieu le tout-puissant qui m’a permis de réaliser ce travail. Je tiens à exprimer toute ma reconnaissance à mon directeur de thèse M. Samir AKNINE et à mon co-directeur de thèse M. Mbaye SENE, pour leurs encouragements, pour leur confiance en mes capacités intellectuelles, pour leurs conseils et soutiens pendant ces années de thèse. Je les remercie également pour leur patience et leur disponibilité (lectures et relectures de mes travaux).

Je voudrais également exprimer toute ma gratitude envers les membres de mon jury : les professeurs Philippe MATHIEU et Renée MANDIAU d’avoir accepté d’être rapporteurs de cette thèse et d’avoir examiné minutieusement ce travail. Le professeur Honiden SHINICHI pour l’intérêt qu’il a bien voulu porter à mon travail en acceptant de prendre part à ce jury. Le Dr. Onn SHEHORY qui a été un collaborateur externe très pertinent en ce qui concerne le review de mes travaux de thèse. Je souhaite remercier particulièrement le professeur Hamamache KHEDDOUCI, responsable de l’équipe GOAL (Graphes, AlgOrithmes et AppLications) qui a été responsable du laboratoire GAMA (Graphes, Algorithmes et Multi-Agents) dans laquelle j’ai commencé ce travail, pour l’accueil et les conseils qu’il a pu m’apporter pendant ces années.

Ma reconnaissance va aussi à l’endroit de ceux qui m’ont soutenu et qui m’ont aidé tout au long de ma scolarité. Je pense à mes chers parents : Raymond et Katrine TINE, à ma grand-mère Rose DIOP, à mon grand-père Pierre TINE, à mes tantes Jeanne TINE, Hélène TINE, Joséphine TINE, Elisabeth TINE, Jeannette TINE, à mes oncles Michel TINE, Léon TINE, Raphaël TINE, Jean TINE, à mes sœurs Rosalie Noëlie et sa fille Thérèse Marie, Sabine Bernadette, Jeanne Anna Awa, à mes frères Jean Jacques (Socio), Pierre Christophe (Piosly), Georges Gabriel (Duc Doggers).

Merci à tous les doctorants de notre équipe SMA, année scolaire 2014-2015, spécialement, Ndèye Arame DIAGO doctorante de deuxième année, pour sa gentillesse, sa disponibilité, sa collaboration scientifique et sociale depuis toutes ces années que nous avons partagé depuis la classe de Licence. Merci à Fairouz BEGGAS pour sa gentillesse et sa disponibilité depuis que mon collègue de bureau Brahim NEGGAZI nous a permis de faire connaissance. Merci aussi à Isabelle BUISSON et Brigitte

GUYADER pour leur gentillesse, pour leur disponibilité et pour ma prise en charge administrative au sein du laboratoire LIRIS.

Je remercie le Père Nicolas de BOCCARD pour son amitié et son aide sans faille depuis que j'ai commencé mes séjours de recherche à Lyon. Merci à Alix RUDIGOZ et à la famille RUDIGOZ qui ont aussi rendu mes séjours à Lyon plus conviviales.

Merci à la coopération française et d'action culturelle de l'ambassade de France au Sénégal pour la bourse d'appui SCAC qu'elle m'a offert au cours de mes deux dernières années de thèse sans laquelle il me serait difficile d'effectuer mes derniers séjours de recherche en France. En particulier, Mme. Fabienne DESERT pour sa gentillesse et pour avoir toujours géré avec professionnalisme, la planification et le bon déroulement de mes séjours en France.

Merci aux professeurs Mamadou SARR et Meissa FALL et au Maître de conférences Mapathé NDIAYE de l'université de THIES pour leur soutien et la confiance sans conditions en mon égard depuis 2009. Je les remercie aussi pour les activités pédagogiques qu'ils m'ont confiées à ladite université et pour leur appui pour le bon déroulement de mes travaux de thèse.

Mention spéciale à Brahim NEGGAZI (Amigo) mon collègue de bureau avec qui j'ai commencé la thèse qu'il a soutenue le 15 Avril 2015. En effet, Amigo n'a ménagé aucun effort pour m'aider dans mes démarches administratives à Lyon 1, pour faciliter mon intégration sociale au laboratoire, pour rendre mes séjours au laboratoire plus facile notamment pour les balades et pour les informations utiles qu'on ne trouve dans aucun manuel pour étranger arrivant en France.

TABLE DES MATIÈRES

TABLE DES MATIÈRES	ix
LISTE DES FIGURES	xi
INTRODUCTION GÉNÉRALE	1
1 FORMATION DE COALITIONS MULTI-AGENTS	7
1.1 LA FORMATION DE COALITIONS	9
1.1.1 Formation de coalitions avec des agents coopératifs	9
1.1.2 Formation de coalitions avec des agents égoïstes	15
1.2 PRINCIPES DES MÉTHODES DE FORMATION DE COALITIONS STABLES	28
1.2.1 Formation de coalitions stables sans inférence dynamique	29
1.2.2 Formation de coalitions stables avec inférence dynamique	36
CONCLUSION	39
2 FORMATION DE COALITIONS NASH-STABLES ET PERSISTANTES	41
2.1 OBJECTIFS ET APPROCHE GÉNÉRALE	43
2.2 CONCEPTS DU MODÈLE	45
2.2.1 Propriétés d'un agent	45
2.2.2 Concepts de formation de coalitions	49
2.2.3 Règles générales de notre mécanisme de formation de coalitions	51
2.3 MODÈLE DE RECOMMANDATION	53
2.4 MÉCANISME DE FORMATION DE COALITIONS	55
2.4.1 Le vecteur invariant d'une tâche	55
2.4.2 Négociation multilatérale et formation de coalitions	57
2.5 DÉROULEMENT DE NOTRE MÉCANISME SUR UN EXEMPLE	60
2.6 RECHERCHE DE LA PERSISTANCE DES COALITIONS	66
2.6.1 Exploration localisée : DEP (Depth Exploration Protocol)	67
2.6.2 Exploration étendue : WEP (Width Exploration Protocol)	72
2.6.3 Seuil du risque d'échec d'une coalition	76

2.7	ANALYSE DU MÉCANISME DE FORMATION DE COALITIONS	77
2.8	ÉVALUATIONS DE PERFORMANCES	84
2.8.1	Évaluation analytique de notre mécanisme	84
2.8.2	Évaluation expérimentation de notre mécanisme	87
2.8.3	Évaluation du coût des messages	95
	CONCLUSION	96
3	FORMATION DE COALITIONS A-CORE : S-NRB	99
3.1	OBJECTIFS ET APPROCHE GÉNÉRALE	101
3.2	CONCEPTS DE NOTRE MODÈLE	102
3.3	MÉCANISME DE FORMATION DE COALITIONS	104
3.3.1	Notre mécanisme S-NRB	105
3.3.2	Analyse de S-NRB	120
3.3.3	Règles générales	123
3.4	ÉVALUATIONS DE PERFORMANCES DE NOTRE MÉCANISME S-NRB	125
3.4.1	Évaluations analytiques	126
3.4.2	Expériences	127
	CONCLUSION	130
4	EXTENSION DE LA FORMATION DE COALITIONS A-CORE : P-NRB	131
4.1	OBJECTIFS ET APPROCHE GÉNÉRALE	133
4.2	CONCEPTS DE NOTRE MODÈLE	133
4.3	MÉCANISME DE FORMATION DE COALITIONS	134
4.3.1	Notre mécanisme P-NRB	135
4.3.2	Analyse de notre mécanisme P-NRB	144
4.3.3	Règles générales	149
4.4	ÉVALUATIONS DE NOTRE MÉCANISME P-NRB	151
4.4.1	Évaluations analytiques	151
4.4.2	Expériences	153
4.5	ÉTUDE COMPARATIVE ENTRE S-NRB ET P-NRB	156
4.5.1	Comparaisons analytiques entre S-NRB et P-NRB	156
4.5.2	Comparaison fonctionnelle entre S-NRB et P-NRB	159
	CONCLUSION	161
	CONCLUSION GÉNÉRALE	163
A	ANNEXES	167
A.1	RAPPELS SUR LOIS DE PROBABILITÉ USUELLES	169
A.1.1	Reconnaissance d'une loi de Bernoulli	169

A.1.2	Reconnaissance d'une loi Binômiale	169
A.1.3	Reconnaissance d'une loi Hypergéométrique	170
A.1.4	Reconnaissance d'une loi géométrique	171
A.1.5	Reconnaissance d'une loi de Poisson	172
A.2	RAPPELS SUR LES PROCESSUS DE MARKOV	172
A.2.1	Probabilité de transition et matrice de transition	173
A.2.2	Probabilités de transition en m étapes	173
A.2.3	Graphes représentatifs et classification des états	174
A.2.4	Périodicité des chaîne de Markov	176
A.2.5	Distribution initiale et comportement transitoire	176
A.2.6	comportement asymptotique des chaînes irréductibles	177
BIBLIOGRAPHIE		179

LISTE DES FIGURES

1.1	Le graphe représentant les structures de coalitions.	13
1.2	Le graphe des Integer-partitions, c'est-à-dire, qui regroupe les structures de coalitions en fonction de la taille de toutes les coalitions qu'elles contiennent.	14
1.3	Méthodes de formation de coalitions stables.	30
2.1	Dispositions schématique des agents primaires et des agents subordonnés dans une structure de coalition.	51
2.2	Représentation hiérarchique de l'historique	53
2.3	Historique de l'agent a_1	55
2.4	Topologie de cinq composants ad-hoc mobiles (Smartphones, robots, capteurs, ...) où chacun de ces composants utilise la technologie Bluetooth et implémente un agent a_i	61
2.5	a_1 et a_4 déclarent leur intention de former une coalition pour leurs tâches.	62
2.6	Les agents a_2 et a_3 choisissent de rejoindre la coalition de l'agent a_4 . Cependant, a_5 répond favorablement à a_4 , mais à a_1 , il répond pour demander l'établissement d'une nouvelle <i>alliance</i> $Al_{a_1}^p$ qui tient compte de sa demande de modification des contraintes $\Delta_{T_{c_1}}$	63
2.7	Formation de la coalition C_4 pour la tâche de l'agent a_4 où a_4 préfère a_5 et où a_5 est d'accord avec a_4 pour rejoindre le <i>backbone</i> de la coalition.	64

2.8	Envoi d'un nouveau message de demande formation d'une coalition par a_1 en changeant $TTL=2$ en $TTL=3$ après le calcul du nouveau vecteur invariant de sa tâche. . .	65
2.9	Grâce à l'augmentation du TTL , a_2 reçoit et soumet une réponse à la demande de coalition de l'agent a_1 . De plus, a_3 reçoit le message de demande et envoie une réponse à a_1 car il n'est pas dans le <i>backbone</i> de la coalition C_4 formée de l'agent a_4 et a_5	65
2.10	a_1 accepte de partager la coalition avec a_2 comme agent <i>primaire</i> . Les deux agents synchronisent leurs connaissances sur la tâche, ce qui amène a_2 et a_1 à être dans le même voisinage. De plus, ils se synchronisent à propos de la stabilité du <i>backbone</i> et de la validation de leur <i>alliance</i> Al_{a_1,a_2}	66
2.11	Schéma montrant les agents <i>primaires</i> et les agents <i>subordonnés</i> de chaque coalition $C_i : i=1$ ou 4	66
2.12	L'agent a_1 détecte la disparition d'un agent <i>primaire</i> de sa coalition C et la notifie aux autres agents <i>primaires</i> de sa coalition. Après le calcul du vecteur invariant de la ou des tâches de la coalition, a_4 détecte une insuffisance des actions entreprises par la coalition C face à la ou aux tâches et propage ainsi cette information dans le <i>backbone</i> de sa coalition C	69
2.13	Chaque agent <i>primaire</i> $a_i / i \in [1,5]$ de la coalition essaie d'identifier et d'ajouter les éléments requis pour la coalition en matière de ressources ou de contribution marginale pour assurer la stabilisation de la coalition et pour obtenir une récompense supplémentaire au regard des ressources ajoutées.	70
2.14	Chaque agent <i>primaire</i> $a_i i \in [1,5]$ prévient son agent <i>subordonné</i> $a_{s_i} i \in [1,4]$ afin de rechercher un autre état de stabilité de leur coalition. Si un agent n'a pas de subordonné (par exemple, a_5) ou des ressources à ajouter, il n'attend que les messages de synchronisation pour prendre connaissance de l'état de stabilité de la coalition et des décisions prises par le <i>backbone</i>	70
2.15	À cette étape, les agents <i>subordonnés</i> a_{s1} et a_{s2} des agents a_1 et a_2 ont les ressources requises. Suite à la réponse favorable à l'intégration de ces <i>subordonnés</i> dans le <i>backbone</i> , a_1 et a_2 se synchronisent avec les autres agents <i>primaires</i> de leur coalition. Cette synchronisation informe des ressources et des agents ajoutés pour éviter qu'un des agents ne recherche les mêmes ressources.	71
2.16	Les agents <i>primaires</i> a_1 et a_2 confirment respectivement l'intégration des agents <i>subordonnés</i> a_{s1} et a_{s2} dans la coalition et calculent de nouveau le vecteur invariant de la ou des tâches pour évaluer la stabilité de leur coalition. Les agents a_1 et a_2 synchronisent leurs connaissances sur la ou les tâches et sur la stabilité de leur coalition avec les nouveaux agents intégrés et ceux du <i>backbone</i>	72
2.17	Les agents du <i>backbone</i> échangent leurs connaissances sur la ou les tâches.	74

2.18	Utilisation du mécanisme de <i>recommandation</i> par les agents <i>primaires</i> pour prendre contact avec leurs agents <i>subordonnés</i> et avec les autres agents alliés de leurs voisinages. Seul a_1 envoie une demande de <i>recommandation</i> à un agent allié qui ne participe pas à sa coalition.	75
2.19	Initiation des négociations par chaque agent <i>primaire</i> avec leurs agents voisins qu'ils soient <i>subordonnés</i> , alliés ou non. Ainsi, les agents <i>primaires</i> a_1, a_2, a_3, a_4 et a_5 envoient des messages à tout agent de leur voisinage.	75
2.20	Fiabilité d'exécution des tâches en fonction du taux de défaillance des composants hébergeant les agents.	85
2.21	Comparaison entre R1 et R2 lorsque au fil du temps l'indisponibilité des agents (panne) suit une distribution de probabilité Normale.	86
2.22	Comparaison entre R1 et R2 lorsque au fil du temps l'indisponibilité des agents (panne) suit une distribution de probabilité de Poisson.	86
2.23	Comparaison entre R1 et R2 lorsque au fil du temps l'indisponibilité des agents (panne) suit une distribution de probabilité Uniforme.	86
2.24	Comparaison entre R1 et R2 lorsque au fil du temps l'indisponibilité des agents (panne) suit une distribution de probabilité Brownienne.	87
2.25	Pourcentage de stabilisation des coalitions sachant que l'indisponibilité aléatoire des agents suit une distribution de Poisson $P(0.5)$	89
2.26	Niveau de stabilité garanti suivant les ratios lorsque les indisponibilités des agents suivent une distribution Normale $panne \sim N(0.5, (0.1)^2)$	90
2.27	Le temps moyen nécessaire pour former $\Phi_i/i \in [1, 3]$ coalitions stables suivant les <i>ratios</i> et quand les indisponibilités des agents suivent une distribution Normale $panne \sim N(0.5, (0.1)^2)$	90
2.28	Comparaisons des temps nécessaire pour former les coalitions.	91
2.29	Probabilité d'obtenir une coalition stable sachant les probabilités $P1, P2$ et $P3$	92
2.30	Probabilité d'avoir deux coalitions stables sachant les probabilités $P1, P2$ et $P3$	92
2.31	Probabilité d'avoir trois coalitions stables sachant les probabilités $P1, P2$ et $P3$	92
2.32	Temps nécessaires pour former cinq coalitions stables lorsque les protocoles d'exploration <i>DEP</i> (<i>Depth Exploration Protocol</i>) et <i>WEP</i> (<i>Width Exploration Protocol</i>) sont utilisés.	93
2.33	Temps nécessaires pour former huit coalitions stables lorsque les protocoles d'exploration <i>DEP</i> (<i>Depth Exploration Protocol</i>) et <i>WEP</i> (<i>Width Exploration Protocol</i>) sont utilisés.	94

- 3.1 Répartition (Net) des agents dans l'environnement de déploiement. La figure suggère que, $\vartheta_{a_1}^t = \{a_5, a_6\}$, $\vartheta_{a_2}^t = \{a_6, a_3\}$, $\vartheta_{a_3}^t = \{a_2, a_6\}$, $\vartheta_{a_4}^t = \{a_5\}$, $\vartheta_{a_5}^t = \{a_1, a_4\}$ et $\vartheta_{a_6}^t = \{a_1, a_2, a_3\}$. Le trait entre les agents a_6 et a_2 et celui entre les agents a_6 et a_3 matérialisent le fait que c'est a_6 qui gère la participation des agents a_2 et a_3 dans l'environnement de déploiement. 104
- 3.2 Diffusions séquentielles du message de sondage pour l'offre $O_c^{a_5} = \{\{B_1, B_2, B_4, B_5\}, \{\emptyset\}\}$, c'est-à-dire, une offre de formation d'une coalition C assignée à a_5 . Seuls les agents de la vue de a_5 reçoivent l'offre de a_5 , car $TTL = 1$ dans le message de sondage. De plus, a_4 et a_1 doivent prendre une décision locale avant de contacter à leur tour, les agents de leur vue. 106
- 3.3 Prises de décision individuelles avant de propager le message de sondage pour l'offre. Le voisinage de a_4 est vide donc il donne sa réponse, mais a_1 poursuit le sondage pour l'offre proposée par a_5 . Nous voyons une différence des listes de sondage, car $pl_{a_5} = \{\{a_5, a_4\}\}$ dans la réponse de a_4 alors que $pl_{a_5} = \{\{a_5, a_1\}\}$ dans le message de a_1 . Cela est dû au fait que, a_4 et a_1 ne sont pas dans le même voisinage et chacun d'eux ignore la réponse de l'autre. 108
- 3.4 Prises de décisions individuelles avant de répondre au message de sondage pour l'offre. Dans le cas du refus de a_6 , nous voyons qu'il n'y a pas de changement dans la liste de sondage par rapport à la figure 3.3 et puis a_6 ne contacte pas les autres agents de sa vue (a_2 et a_3). 108
- 3.5 Réponse d'un agent après la fin du sondage de son voisinage. Suite au refus de a_6 , a_1 donne sa réponse à a_5 pour indiquer sa décision et celle de son voisinage. 109
- 3.6 Prise de décision collective entre les agents a_1 , a_4 et a_5 qui ont accepté l'offre pour décider s'il est nécessaire de poursuivre le sondage pour l'offre. Cette décision dépend de l'état d'acceptation des buts de l'offre et de la disponibilité des agents de la liste de sondage. 109
- 3.7 Message de confirmation de la continuité du sondage pour indiquer la nécessité d'avoir plus de réponses favorables à la formation de la coalition pour l'offre. Ici, seulement a_1 envoie ce message de confirmation, car tous les voisins de a_5 sont déjà dans la liste de sondage pl_{a_5} et a_4 n'a que a_5 comme voisin. Nous n'avons pas séparé les étapes de réception du message par a_6 et l'étape où a_6 envoie le message, car a_6 ne fait que relayer le message. Cela, du fait que a_6 a déjà rejeté l'offre de formation de C 111

3.8	Réponses supplémentaires d'agents après la confirmation de la continuité du sondage pour l'offre. Les agents peuvent avoir à se déplacer en cas de nécessité, pour former un ensemble connexe. Dans cette figure, nous supposons que c'est a_2 qui se déplace. Ici aussi, la réponse de a_2 est simplement relayée par les agents qui la reçoivent, car c'est un message de mise à jour d'informations.	112
3.9	Validation de la liste de sondage pour l'offre. La coalition pour l'offre par les agents de la liste de sondage $pl_{a_5}=\{a_1, a_2, a_4, a_5\}$ comme l'ensemble des buts de l'offre sont acceptés. Ainsi, $A_c=\{a_1, a_2, a_4, a_5\}$	112
3.10	Détection de conflits entre deux agents par l'agent a_1 suite à la réception de la réponse de a_3 . En supposons que a_3 et a_4 sont en conflits.	117
3.11	a_1 envoie une demande pour la recherche de garants des agents a_3 et a_4 qui sont des agents en conflits. À noter que les agents en conflits ne sont pas contactés.	118
3.12	Dans cette figure, nous supposons que a_5 se porte garant de l'agent a_4 , mais que a_3 n'a pas de garant.	119
3.13	a_1 propose une nouvelle liste suite à la réponse de a_5 . C'est finalement a_4 qui est inclus dans la liste de sondage pl_{a_5} en attendant des amendements sur cette nouvelle liste pl_{a_5}	119
3.14	Coût des messages jusqu'à la convergence de S-NRB si $conflits \sim U(a, b)$ où $a=0$ et $b=10$	126
3.15	Coût des messages jusqu'à la convergence de S-NRB si $conflits \sim N(\mu, \sigma^2)$ où $\mu=0.5$ et $\sigma=0.1$	126
3.16	Coût des messages jusqu'à la convergence de S-NRB si $conflits \sim P(\lambda)$ où $\lambda=0.5$	127
3.17	Auto-stabilisation des coalitions <i>core stables</i> formées avec notre mécanisme S-NRB 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 $b=10$. Pour ces six changements visibles sur la figure, les <i>ratios</i> sont de : 1.42, 1.66, 2, 3.33, 1.66 et 2.	129
3.18	Auto-stabilisation des coalitions <i>core stables</i> formées avec notre mécanisme S-NRB si les changements d'états des tâches et la disponibilité des agents suivent une distribution de probabilités Normale $N(\mu, \sigma^2)$ où $\mu=0.5$ et $\sigma=0.1$. Pour ces quatre changements visibles sur la figure, les <i>ratios</i> sont de : 0.90, 3.33, 3.33, 2.50.	129
3.19	Auto-stabilisation des coalitions <i>core stables</i> formées avec notre mécanisme S-NRB si les changements d'états des tâches et la disponibilité des agents suivent une distribution de probabilités de Poisson $P(\lambda)$ où $\lambda=0.5$. Pour ces cinq changements visibles sur la figure, les <i>ratios</i> sont de : 3.33, 1, 2.5, 2, 1.66.	129

4.1	Diffusion de l'offre. Dans ce cas, tout agent a_i qui reçoit d'un agent a_j un message de sondage pour une offre composée, il le diffuse dans son voisinage $\vartheta_{a_i}^t \setminus \{a_j\}$, c'est-à-dire, en mode non-return broadcast. Dans cette figure, l'agent a_1 est l'initiateur de l'offre composée et nous avons supposé des problèmes de transmission de messages entre les agents a_5 et a_8 au moment de la diffusion de cette offre.	135
4.2	Définition des rangs des agents, c'est-à-dire, chaque agent calcul son rang en utilisant l'équation 4.1.	136
4.3	Formation des niveau-discussions en se fondant sur les rangs des agents.	136
4.4	Prise de décision entre les <i>niveau-discussions</i>	140
4.5	Coûts des messages de P-NRB lorsque $conflits \sim U(a, b)$ où $a=0$ et $b=10$	152
4.6	Coûts des messages de P-NRB lorsque $conflict \sim N(\mu, \sigma^2)$ où $\mu=0.5$ et $\sigma=0.1$	152
4.7	Coûts des messages de P-NRB lorsque $conflict \sim P(\lambda)$ où $\lambda=0.5$	153
4.8	Stabilité des coalitions dans le temps lorsque les <i>Ratios</i> suivent une distribution de probabilité Uniforme $U(a, b) : a=0$ et $b=10$	154
4.9	Stabilité des coalitions dans le temps lorsque les <i>Ratios</i> suivent une distribution de probabilité Normale $N(\mu, \sigma^2) : \mu=0.5$ et $\sigma=0.1$	155
4.10	Stabilité des coalitions dans le temps lorsque les <i>Ratios</i> suivent une distribution de probabilité de Poisson $P(\lambda) : \lambda=0.5$	155
4.11	Coûts en matière de messages jusqu'à la convergence de S-NRB et P-NRB, lorsque $conflits \sim U(a, b)$ où $a=0$ et $b=10$	156
4.12	Coûts en matière de messages jusqu'à la convergence de S-NRB et P-NRB, lorsque $conflits \sim N(\mu, \sigma^2)$ où $\mu=0.5$ et $\sigma=0.1$	156
4.13	Coûts en matière de messages jusqu'à la convergence de S-NRB et P-NRB, lorsque $conflits \sim P(\lambda)$ où $\lambda=0.5$	157
4.14	Temps moyen requis pour la convergence de S-NRB et de P-NRB, si $conflits \sim U(a, b)$ où $a=0$ et $b=10$).	157
4.15	Temps moyen requis pour la convergence de S-NRB et de P-NRB, si $conflits \sim N(\mu, \sigma^2)$ où $\mu=0.5$ et $\sigma=0.1$).	158
4.16	Temps moyen requis pour la convergence de S-NRB et de P-NRB, si $conflits \sim P(\lambda)$ où $\lambda=0.5$).	158
4.17	Comparaisons fonctionnelles entre S-NRB et P-NRB.	160
A.1	Graphe de transition de la chaîne de Markov.	175

INTRODUCTION GÉNÉRALE

L'informatique en général et l'intelligence artificielle en particulier considère les programmes comme des entités individualisées capables de rivaliser avec l'être humain dans des domaines précis. L'intelligence artificielle est une branche de l'informatique qui vise la recherche des moyens susceptibles de doter les systèmes informatiques de capacités de raisonnement comparables à celles des êtres humains. Ainsi, les systèmes informatiques sont maintenant consacrés à des tâches de domaines de plus en plus complexes telles que l'aide à la prise de décision dans une entreprise, la surveillance et le contrôle des processus industriels, l'aide au diagnostic médical, etc.

Les systèmes multi-agents se situent à l'intersection de l'intelligence artificielle et des systèmes distribués qui sont des systèmes informatiques composés de nœuds dotés de capacités de calcul, connectés à l'aide d'un réseau de communication.

Le paradigme multi-agents fournit un cadre approprié pour modéliser des problèmes et pour les résoudre. Les systèmes multi-agents sont issus d'une volonté de distribuer l'intelligence d'un système dans le but de faire résoudre plus efficacement par plusieurs entités appelées les agents, des problèmes qui, par leur nature, ne sont pas adaptés à une résolution centralisée, par exemple, problèmes trop complexes, physiquement distribués, etc. Les agents évoluent dans un environnement, guidés par des objectifs à atteindre et coopèrent au sein de communautés selon des modes parfois complexes d'interactions, de conflits et de concurrences. De ces interactions émergent des structures organisées qui, en retour, contraignent et modifient les comportements des agents. D'une manière générale, chaque agent a ses propres objectifs et ne s'intéresse pas a priori aux objectifs des autres agents. Par exemple, sur une plateforme d'E-marché, des agents achètent et vendent des actifs financiers avec pour objectif de gagner de l'argent.

Un ensemble de définitions valables suivant le domaine peuvent se rapporter à la notion d'agent. Cependant, nous considérons ici un agent comme une entité physique (par exemple, robot, voiture, etc.) ou virtuelle (par exemple, logiciel) (cf. (Ferber 1997)(Chalkiadakis et al. 2011)(Elkind et al. 2013)) qui :

- est autonome, capable de percevoir et d'agir dans un environnement,
- peut communiquer directement avec d'autres entités,
- est motivé par un ensemble de tendances sous la forme d'objectifs individuels ou d'une fonction de satisfaction, qu'il cherche à optimiser,
- dispose d'une représentation entière ou partielle ou éventuellement aucune de cet environnement,
- a tendance à satisfaire ses objectifs, en ne tenant compte que des ressources et des compétences dont elle dispose, et en fonction de sa perception et des informations qu'elle reçoit.

De là nous considérons un système multi-agents comme étant la combinaison des éléments suivants (cf. (Ferber 1997)(Chalkiadakis et al. 2011)(Elkind et al. 2013)) :

- Un environnement E , c'est-à-dire, un espace disposant d'une métrique utilisable par les agents pour affiner leur perception de cet environnement.
- Un ensemble d'objets O situés dans cet environnement où pour tout objet, il est possible, à un instant t , de lui associer une position dans E . Il existe des objets qui sont passifs dans E , c'est-à-dire, qui peuvent être perçus, créés, détruits ou modifiés par les agents.
- Un ensemble A d'agents, qui sont des objets particuliers ($A \subseteq O$), lesquels représentent les entités actives du système.
- Un ensemble de relations R qui lient ces objets entre eux.
- Un ensemble d'opérations permettant aux agents de percevoir, consommer, transformer, manipuler des objets et produire des effets qui affectent E .

Dans un système multi-agents, les agents peuvent avoir recours à différentes techniques pour atteindre leurs objectifs. Cela peut les amener à se regrouper ou à se coordonner sans une organisation particulière.

Contexte de la thèse

Dans le cadre de cette thèse, nous nous intéressons à la détermination d'un ensemble de modèles de formation de coalitions multi-agents dans un contexte où des agents autonomes et égoïstes sont déployés dans des composants ad-hoc comme des robots, PDA, etc. et où les tâches sont stochastiques.

La formation de coalitions est un moyen incitatif pour la coopération. Elle permet à des individus de se regrouper temporairement afin d'atteindre conjointement des objectifs communs. Adapter le comportement des nouveaux systèmes aux contextes humains pousse plusieurs recherches actuelles à repenser le problème de la formation de coalitions. Ce problème apparaît sous

différentes formulations qui ne se focalisent pas toujours sur les mêmes questionnements et ne nécessitent pas toujours les mêmes analyses et méthodes de résolution. Lorsque les individus ont les mêmes objectifs, les faire coopérer relève plus du rapprochement naturel que de l'exploit. Dans le cas d'individus à caractère égoïste, ce comportement coopératif est motivé le plus souvent par l'obtention d'une récompense individuelle.

Un réseau ad-hoc est formé d'un ensemble de nœuds autonomes et mobiles reliés par des liens sans fils. Les nœuds peuvent se déplacer aléatoirement et le nombre de connexions et déconnexions peut être important. Par conséquent, la topologie du réseau change rapidement, ce qui rend l'environnement fortement dynamique et rend incertaine l'exécution des tâches. Nous considérons dans le cadre de ce travail que les tâches stochastiques requièrent un ensemble d'agents pour leur exécution et que même si une tâche est prise en charge, l'environnement continue à évoluer dynamiquement et l'effet des actions des agents est non-déterministe.

Par ailleurs, les agents ne connaissent pas l'ensemble des participants dans le système et ne peuvent communiquer directement qu'avec leurs voisins. Cependant, ce voisinage change en fonction de la dynamique du réseau et le plus souvent de manière imprévisible. Une des difficultés majeures consiste alors à définir pour les systèmes multi-agents, des mécanismes de coordination efficaces qui tiennent compte des contraintes de déploiement des agents en se fondant sur la formation de coalitions ainsi que de garantir leur exécution dans un contexte ad-hoc.

Problèmes et objectifs de la thèse

Une des hypothèses fortes des systèmes multi-agents est que l'infrastructure réseau sous-jacente est stable et que les tâches n'évoluent plus une fois que les coalitions sont formées, c'est-à-dire que, les nœuds du réseau sur lesquels les agents sont déployés sont connus, et qu'il n'y a pas de rupture dans l'exécution des protocoles des agents.

Bien au contraire, les réseaux ad-hoc définissent un cadre d'étude caractérisé particulièrement par le fait que les nœuds dédiés au déploiement des agents peuvent apparaître ou disparaître dynamiquement dans le réseau ou même devenir inaccessibles, en raison, par exemple, de la mobilité de l'utilisateur dans son environnement ou de sa déconnexion. Ces changements de la configuration du système créent des situations exceptionnelles qui rendent la plupart des mécanismes actuels de raisonnement des agents inappropriés.

Par exemple, comment gérer les interruptions de protocoles, la disparition d'un ensemble d'agents (et donc de compétences), la migration des données selon le contexte, ... sans avoir à renégocier les accords des agents.

Faire l'hypothèse que les tâches n'évoluent plus une fois que le processus de coordination des

agents est en marche peut s'avérer être en déphasage avec certains contextes d'utilisation des applications multi-agents, par exemple, dans le cas de l'extinction d'un feu, de la gestion du trafic routier, etc.

Plusieurs difficultés existent, nous pouvons citer entre autres :

- la dynamique caractéristique de l'environnement, la négociation entre les agents devrait être effectuée de proche en proche en matière de voisinage direct et non pas de façon globalisée, car, les agents ne peuvent pas connaître l'ensemble des composants du système multi-agents,
- l'inaccessibilité de certains agents suite à la mobilité fortuite de certains nœuds peut empêcher les processus de négociation ou de coordination de se poursuivre, ce qui peut créer des incohérences dans les données des agents,
- la dynamique des tâches qui ont une évolution non-déterministe.

Notre objectif est en se fondant sur nos modèles de formation de coalitions, d'adapter les mécanismes de coordination et d'interaction tels qu'ils sont connus dans les systèmes multi-agents pour le contexte spécifique des réseaux ad-hoc où évoluent des tâches stochastiques. Cette problématique est caractérisée à la fois par l'inexistence de vue globale au niveau réseau reflétant une image complète du système multi-agent, par la dynamique des nœuds inhérents aux actions des agents et par des incertitudes en ce qui concernent l'issue de l'exécution des tâches.

Plan de la thèse

Nos contributions portent sur la proposition de différents modèles de formation de coalitions pour une coordination efficace des agents qui doivent exécuter des tâches stochastiques.

Le *premier chapitre* introduit le domaine des systèmes multi-agents et un survol du problème de la formation de coalitions. Nous y présentons d'abord les travaux afférents à la formation de coalitions d'agents coopératifs puis aux travaux relatifs aux agents égoïstes. Nous y détaillons les travaux fondamentaux, leurs problématiques et leurs solutions. Nous y définissons les concepts essentiels et nous y exposons certains résultats que nous estimons indispensables à la compréhension des différents travaux du domaine d'une manière générale et des travaux effectués dans cette thèse en particulier.

Le *deuxième chapitre* présente un nouveau mécanisme de formation de coalitions stables dans un contexte de tâches stochastiques et d'agents autonomes égoïstes avec des disponibilités imprévisibles. Ces agents ne peuvent en aucun cas profiter de connaissances a priori ou des avantages

d'une quelconque centralisation des calculs. Notre mécanisme de formation de coalitions détaillé dans ce chapitre, est fondé sur - un principe d'*alliances* et de recommandations pour faciliter la formation de coalitions stables, - les lois de probabilité et les MDPs (Chaîne de Markov) pour suivre et prédire la stabilité des coalitions. Cette prédiction de la stabilité des coalitions nous permet d'introduire la notion de coalitions persistantes (des coalitions qui restent stables dans le temps malgré la dynamique et les incertitudes sur les agents, sur les composants et sur leurs tâches). Suite à l'analyse de notre modèle, pour fournir les preuves expérimentales des propriétés intrinsèques que nous avons mises en exergue, nous avons développé un simulateur du fait de l'absence de simulateur adaptable à notre contexte d'étude.

Le *troisième chapitre* présente en détail la mise en œuvre d'un mécanisme décentralisé de formation de coalitions stables dans un contexte de tâches stochastiques qui tient compte des incertitudes liées d'une part aux dépendances, aux préférences, aux disponibilités imprévisibles et aux conflits qui peuvent arriver entre les agents et, d'autre part, au caractère stochastique des tâches à exécuter. Dans ce contexte incertain, nous ne considérons aussi aucune connaissance qui peut nous permettre de déterminer a priori les coalitions stables à former pour l'exécution de ces tâches. Par ailleurs, nous considérons qu'il n'est pas possible de calculer de façon centralisée ces coalitions avant le début de l'exécution des tâches en raison des incertitudes des agents et la contrainte de temps des tâches stochastiques. Les différences principales par rapport au chapitre précédent sont qu'ici, nous recherchons des coalitions stables au sens du Core et nous tenons compte des dépendances et des conflits entre les agents. Le mécanisme de formation de coalitions stables que nous proposons est dit *S-NRB (Sequential Non-Return Broadcast)* et permet de former des coalitions dites *A-core*, c'est-à-dire, *Core-stable* et *Auto-stables*. Notre mécanisme *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*. Pour faire cette analyse expérimentale, nous avons eu à ajouter des fonctionnalités supplémentaire pour obtenir la version 2 du simulateur qui nous a servi pour l'évaluation de performances du mécanisme détaillé dans le second chapitre.

Le *quatrième chapitre* se positionne dans un contexte multi-agents hétérogène et distribué qui est caractérisé par la dynamique et les incertitudes de l'environnement de déploiement comme pour le troisième chapitre. Spécifiquement, nous cherchons à étendre nos travaux qui ont conduit à la proposition du mécanisme de formation de coalitions *S-NRB*. Notre nouveau mécanisme de formation de coalitions est parallèle, asynchrone, décentralisé et *anytime* (trouvent des coalitions stables en cas d'interruption prématurée du processus de formation de coalitions). Nous appelons *P-NRB (Parallel Non-Return Broadcast)* ce nouveau mécanisme de formation de coalitions *A-core*. *P-NRB* combine les principes de la théorie des jeux, avec le formalisme des chaînes de Markov (*MDP*)

et les lois de probabilités. Le mécanisme converge graduellement vers les coalitions stables qui maximisent le bien-être social des agents. Les analyses et les simulations que nous avons menées, montrent l'efficacité du mécanisme *P-NRB*. De plus, il améliore ou étend dans la plupart des cas, notre mécanisme *S-NRB*. Par étend, nous comprenons que, *P-NRB* fournit des propriétés qu'on ne peut pas obtenir avec *S-NRB*. Cela, a mené aussi à la troisième version de notre simulateur.

1

FORMATION DE COALITIONS MULTI-AGENTS

La formation de coalitions est un moyen incitatif pour la coopération. Elle permet à des individus de se regrouper temporairement afin d'atteindre conjointement des objectifs communs. Adapter le comportement des nouveaux systèmes aux contextes humains pousse plusieurs recherches actuelles à repenser le problème de la formation de coalitions. Ce problème apparaît sous différentes formulations qui ne se focalisent pas toujours sur les mêmes questionnements et ne nécessitent pas toujours les mêmes analyses et méthodes de résolution. Lorsque les individus ont les mêmes objectifs, les faire coopérer relève plus du rapprochement naturel que de l'exploit. Dans le cas d'individus à caractère égoïstes, ce comportement coopératif est motivé le plus souvent par l'obtention d'une récompense individuelle. Nous présentons, dans ce chapitre, les différentes approches du problème de la formation de coalitions en nous intéressant d'abord aux travaux afférents à la formation de coalitions d'agents coopératifs puis aux travaux relatifs aux agents égoïstes. Nous y présentons les travaux fondamentaux et les notions clés nécessaires à la compréhension des modèles existants.

1.1 LA FORMATION DE COALITIONS

Les coalitions sont largement utilisées dans les systèmes multi-agents pour exécuter des tâches collectives (Kahan et Rapoport 1984)(Sandholm et al. 1999). Les coalitions sont généralement formées pour une courte période pour un ou plusieurs objectifs et elles sont dissoutes quand ces objectifs n'existent plus, ou ne peuvent plus se réaliser.

Dans une coalition, les agents coordonnent leurs activités afin d'atteindre cet ensemble d'objectifs, sans imposer une coordination entre les agents appartenant à différentes coalitions sauf s'il y a un ensemble d'objectifs interdépendants entre les coalitions coexistantes.

Les principaux problèmes inhérents à la formation de coalitions sont généralement :

Le calcul de la valeur de chaque coalition possible ; la recherche de la structure de coalitions (ensemble des coalitions disjointes) optimale qui maximise la valeur de chaque coalition ; l'étude des dépendances entre les coalitions qui influent sur la valeur de chaque coalition ; les mécanismes de formation de coalitions dictées par le type et l'urgence des tâches à réaliser ; le schéma de partage de la valeur de chaque coalition entre ses membres (récompense obtenue par chaque agent).

Pour former les coalitions, on a recours - soit à un ajustement dynamique ou non des comportements des agents par planification ou par négociation, - soit par calcul supervisé ou non des coalitions à former.

Dans la suite de ce chapitre, nous présenterons les concepts de base nécessaires à la compréhension de nos travaux. Ces concepts sont tirés de travaux de différents auteurs comme (Chalkiadakis et al. 2011), (Elkind et al. 2013).

1.1.1 Formation de coalitions avec des agents coopératifs

Nous nous intéressons ici aux jeux coalitionnels dans lesquels un ensemble d'agents coopératifs se partagent la recherche d'une structure de coalitions optimale et aux problèmes de l'accomplissement collectif de tâches.

De nombreux travaux ont été réalisés dans ce sens. Il s'agit pour un ensemble d'agents coopératifs de s'organiser en plusieurs groupes d'agents. À chaque groupe d'agents formé est associée une utilité. L'objectif commun de tous les agents du système est de maximiser la somme des utilités des différents groupes qu'ils forment.

Ci-dessous, nous allons décrire ce problème plus formellement et détailler quelques méthodes proposées dans la littérature pour le résoudre.

1.1.1.1 Formalisation des jeux coalitionnels

Soit $A = \{a_1, a_2, \dots, a_n\}$ un ensemble de n agents coopératifs. Ces n agents peuvent former différentes coalitions comme le montrent les définitions ci-dessous.

Définition 1.1 Une coalition $C \subseteq A$ est formée d'un ensemble non vide d'agents. C est la grande coalition, si $C = A$. Si C est la coalition singleton, alors $C = \{a_i\} : a_i \in A$.

Définition 1.2 Une structure de coalitions est une partition de A , c'est-à-dire, un ensemble de coalitions $SC = \{c_1, c_2, \dots, c_m\} : \forall c_i \in SC \ c_i \neq \emptyset, \forall i \neq j \ (c_i, c_j) \in SC \ c_i \cap c_j = \emptyset$ et $\bigcup_i^m c_i = A$.

Propriété 1.1 L'ensemble de toutes les coalitions E_C est égal à 2^N où le nombre de partitions des agents est un N -ième nombre de Bell (Bell 1934) (Levine et Dalton 1962).

Exemple 1.1 Soit $A = \{a_1, a_2, a_3\}$, il y a sept $(2^3 - 1)$ manières de former une coalition plus la coalition vide, d'où les 2^3 manières de former une coalition.

$$E_C = \{\{\emptyset\}, \{a_1\}, \{a_2\}, \{a_3\}, \{a_1, a_2\}, \{a_1, a_3\}, \{a_2, a_3\}, \{a_1, a_2, a_3\}\}.$$

Cinq structures de coalitions sont possibles, c'est-à-dire :

$$\begin{aligned} SC = (&\{\{a_1\}, \{a_2\}, \{a_3\}\}, \\ &\{\{a_1, a_2\}, \{a_3\}\}, \\ &\{\{a_1\}, \{a_2, a_3\}\}, \\ &\{\{a_1, a_3\}, \{a_2\}\}, \\ &\{\{a_1, a_2, a_3\}\}). \end{aligned}$$

Une fonction caractéristique v , appelée également fonction de valuation, associe à chaque coalition une valeur numérique réelle et positive, correspondant à la valeur ou l'utilité de la coalition.

$$\begin{aligned} v : E_C &\mapsto \mathbb{R}^+ \\ C &\mapsto v(C) \end{aligned}$$

Définition 1.3 La valeur d'une structure de coalitions $SC \in E_{SC}$ est définie comme la somme des valeurs des coalitions qui la composent, c'est-à-dire, $v(SC) = \sum_{C \in SC} v(C)$. E_{SC} est l'ensemble des structures de coalitions.

Définition 1.4 Un jeu coalitionnel est un couple (A, v) où A est un ensemble d'agents et v est une fonction caractéristique. Il est question de trouver la structure de coalitions SC parmi l'ensemble des structures de coalitions E_{SC} qui a la valeur la plus élevée possible, c'est-à-dire de trouver la structure de coalitions SC^* telle que :

$$SC^* = \underset{SC \in E_{SC}}{\operatorname{argmax}} \ v(SC) \quad (1.1)$$

Les valeurs des coalitions peuvent être connues a priori ou pas.

Généralement les algorithmes proposés se fondent sur le fait que les agents doivent se partager la recherche de la structure de coalitions optimale SC^* et dans le cas où l'algorithme dispose d'un temps limité, les agents doivent garantir que la qualité de la solution trouvée est proche de celle de la solution optimale.

Dans un contexte coopératif, il n'y a pas de contraintes en ce qui concerne les échanges des informations et des résultats de calculs.

1.1.1.2 Calcul des valeurs des coalitions

Dans (Shehory et Kraus 1998), les auteurs proposent un algorithme de négociation qui permet aux agents coopératifs de se partager le calcul des valeurs des coalitions. Leur algorithme a des limites en matière de critère de performances, notamment, à cause du nombre élevé de messages envoyés, de la redondance des calculs des valeurs des coalitions et de l'occupation mémoire qui augmentent de manière exponentielle avec le nombre d'agents. À cela s'ajoute, un temps de calcul élevé et un problème d'inégalité de partage des calculs entre les agents, car cela dépend de l'ordre dans lequel les agents se sont contactés, c'est-à-dire, plus un agent est contacté, plus il est déchargé de calculs.

Un algorithme qui évite les calculs redondants et équilibre, la répartition des calculs entre les agents est proposé dans (Rahwan et Jennings 2007). Les auteurs fondent leur algorithme sur l'hypothèse que le calcul des valeurs des coalitions est d'autant plus élevé que le nombre d'agents qui les composent est élevé. L'idée est de répartir les calculs entre les agents en fonction de la taille des coalitions. Cet algorithme permet aux agents de connaître exactement les coalitions dont ils ont la charge de calcul, en n'ayant pour information, que le nombre total d'agents A et leur numéro d'identification unique. Contrairement à l'algorithme de (Shehory et Kraus 1998), cet algorithme ne requiert aucune communication entre les agents, puisque ceux-ci connaissent a priori les coalitions dont ils ont la charge et nécessite très peu de mémoire. Cet algorithme a également l'avantage de pouvoir s'adapter aux cas où les agents ont des capacités de calculs différents. Néanmoins, cet algorithme se fonde sur une répartition automatique grâce à une fonction de répartition pour permettre aux agents de connaître les coalitions pour lesquelles calculer les valeurs et non sur un choix d'affectation que feraient les agents eux-mêmes.

Enfin, tous ces travaux supposent que les agents ont les mêmes fonctions d'utilité, qu'ils se font confiance et se partagent les calculs.

1.1.1.3 Recherche de la structure de coalitions optimale

Comme le prouve (Sandholm et al. 1999), cette recherche est coûteuse en calcul si elle est effectuée de manière naïve. De ce fait, plusieurs algorithmes de recherche sont proposés pour raccourcir les délais nécessaires pour la génération des structures de coalitions optimales.

(Yeh 1986) et (Rothkopf et al. 1998) proposent deux algorithmes fondés sur la programmation dynamique pour deux problèmes proches de la recherche de la structure de coalitions optimale (problème de partitionnement et problème d'enchères combinatoires) et donc aisément adaptables. L'idée de base de ces travaux est de subdiviser le problème d'optimisation en sous-problèmes qui peuvent être résolus de façon récursive et combiner ensuite les résultats de ces sous-problèmes pour produire une solution au problème original.

Ces algorithmes ont une complexité en $O(3^n)$, où n est le nombre d'agents. Ils ne produisent de résultat en sortie que si une structure de coalitions optimale est trouvée, ce qui peut s'avérer long et délicat dans certains contextes dynamiques. Ils nécessitent un espace mémoire non négligeable pour les calculs (Trois tables dont l'une contient les $(2^n - 1)$ entrées). De plus, ces algorithmes n'évaluent pas les décompositions des coalitions de taille s jusqu'à ce qu'ils aient fini les calculs pour les coalitions de tailles $1, \dots, S - 1$. On note aussi une redondance des calculs avec ces algorithmes.

(Sandholm et al. 1999) proposent un algorithme de recherche de la structure de coalitions optimale fondée sur la représentation de l'espace de recherche sous la forme d'un graphe. Le graphe représentant les structures de coalitions est divisé en niveaux hiérarchiques superposés et numérotés de 1 à n . Chaque nœud du graphe correspond à une structure de coalitions et chaque nœud appartient à un niveau hiérarchique : les structures de coalitions composées de i coalitions appartiennent au même niveau i . Deux structures de coalitions appartiennent à deux niveaux successifs, s'il est possible d'obtenir l'une des structures à partir de l'union de deux coalitions de l'autre structure.

Exemple 1.2 *Pour un ensemble $A = \{a_1, a_2, a_3, a_4\}$ de 4 agents, nous avons la figure 1.1, qui propose une représentation du graphe des structures de coalitions possibles avec ces 4 agents.*

Pour identifier SC^* , on commence par les deux premiers niveaux du graphe. On remarque que la totalité des coalitions sont représentées dans les niveaux 1 et 2. Après cela, on commence par le haut du graphe en faisant un parcours en largeur, tant qu'il reste du temps de calcul. Les deux premiers niveaux nécessitent le calcul de la valeur de 2^{n-1} structures de coalitions.

Une première amélioration avant celle de (Rahwan et al. 2009b) est proposée par (Dang et Jennings 2004) qui utilise une autre méthode pour parcourir le graphe, ce qui permet ainsi d'améliorer

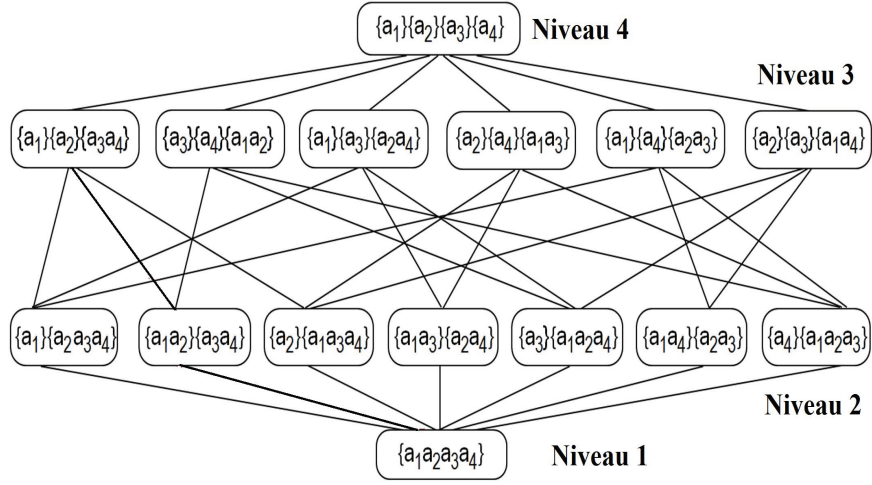


FIGURE 1.1 – Le graphe représentant les structures de coalitions.

les résultats de manière significative. L'algorithme de (Dang et Jennings 2004) ne fait plus des parcours par niveau comme dans (Sandholm et al. 1999), mais en suivant des sous-ensembles de structures de coalitions dépendant à la fois du nombre de coalitions ainsi que du nombre d'agents par coalition. Cette méthode permet de diminuer rapidement la borne de la meilleure solution trouvée. Cependant, il ne réduit pas la complexité de recherche de la solution optimale qui nécessite toujours un parcours de la totalité des structures de coalitions possibles.

(Rahwan et al. 2007) proposent un algorithme utilisant une nouvelle représentation de l'espace de recherche, regroupant les structures de coalitions en fonction de la taille de toutes les coalitions qu'elles contiennent comme le montre la figure 1.2 pour un ensemble de 4 agents. Ils appellent cette représentation un *Integer-Partition*. Les auteurs ont ensuite proposé de nouvelles améliorations dans (Rahwan et Jennings 2008a), (Rahwan et Jennings 2008b), (Rahwan et al. 2009b), (Rahwan et al. 2012a).

Par exemple, pour éviter la redondance des calculs, l'algorithme calcule, la borne supérieure et inférieure d'une première structure de coalitions. Une borne est une valeur d'une structure de coalitions dans une configuration Π_I^A . Par exemple, dans la figure 1.2, nous avons les configurations $\Pi_{\{1,1,1,1\}}^A$, $\Pi_{\{1,1,2\}}^A$, $\Pi_{\{2,2\}}^A$, $\Pi_{\{1,3\}}^A$ et $\Pi_{\{4\}}^A$. L'algorithme améliore progressivement les bornes en éliminant les configurations n'ayant pas de meilleure borne que la dernière meilleure borne trouvée. L'algorithme s'arrête à la fin de la liste des configurations. La notion de configuration est exemplifiée ci-dessous.

Exemple 1.3 Soit trois agents a_1, a_2 et a_3 . Les structures de coalitions $\{\{a_1\}, \{a_2, a_3\}\}$ et $\{\{a_3\}, \{a_1, a_2\}\}$ ont la même configuration notée $\Pi_{\{1,2\}}^A$, c'est-à-dire, leur première coalition est formée de 1 agent et leur seconde coalition de 2 agents.

L'algorithme a une complexité de $O(n^n)$, où n est le nombre d'agents. Il peut retourner un résultat même si tout l'espace des coalitions possibles n'est pas totalement parcouru. En cas d'interruption prématurée, il retourne la meilleure réponse dernièrement trouvée. Cela lui procure la propriété *anytime*.

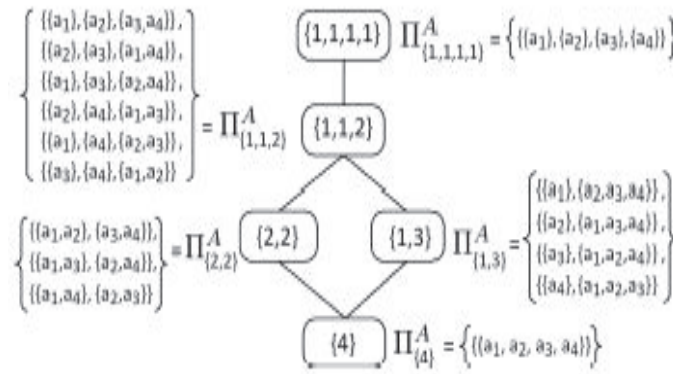


FIGURE 1.2 – Le graphe des Integer-partitions, c'est-à-dire, qui regroupe les structures de coalitions en fonction de la taille de toutes les coalitions qu'elles contiennent.

Les auteurs dans (Modi 2003) introduisent la notion de génération de coalitions en utilisant les méthodes d'optimisation de contraintes distribuées (*Distributed Constraint Optimization Problem (DCOP)*). Dans cette approche, - chaque agent a un choix d'actions à réaliser ; - le gain d'un agent est déterminé à partir de la combinaison des actions et - le but de chaque agent est de choisir l'action qui maximise la somme de ses gains.

(Ueda et al. 2010) considèrent le problème de génération des structures de coalitions où le système multi-agent est représenté comme un seul *DCOP* qu'ils appellent *DCOP modifié*. Cela signifie qu'ils proposent de trouver la structure de coalitions optimale en résolvant un seul *DCOP*. Dans ce cas, la valeur de chaque coalition est calculée comme si la solution optimale du *DCOP* est représentée par les agents de cette coalition. Pour éviter de calculer les valeurs de 2^n coalitions possibles pour n agents, ils combinent les étapes de calcul des valeurs des coalitions avec la recherche de la combinaison optimale des coalitions disjointes grâce à un paramètre σ . Le paramètre σ spécifie, suivant le domaine applicatif, le nombre maximum de coalitions qui peuvent avoir plus d'un agent et l'ensemble des actions possibles pour un agent.

(Rahwan et al. 2011) proposent la modélisation du problème de recherche de structure de coalitions optimales comme un ensemble de contraintes distribuées (*Constrained Coalition Formation (CCF)*). Ils considèrent qu'il y a des contraintes (de préférences des agents, compétences des agents, etc.) inhérentes au contexte de déploiement des agents qui empêchent certains agents d'être ou de ne pas être dans la même coalition. Ce qui leur permet d'imposer des structures de coalitions

à former et de restreindre ainsi l'espace de recherche des coalitions optimales sans évaluer les 2^n coalitions possibles pour n agents. Formellement, le modèle de CCF s'exprime sous la forme, $\{A, E_S C, v\}$ où A est un ensemble d'agents, $E_S C$ un ensemble de structure de coalitions réalisables et v une fonction caractéristique qui assigne une valeur à chaque coalition qui apparaît dans une structure de coalitions réalisables. Ils proposent une représentation des contraintes qui permet de transformer celles-ci en entrées valides pour leur algorithme afin de générer les coalitions réalisables.

Dépendance aux coalitions externes.

Certains travaux étendent le cadre présenté, aux cas où la valeur d'une coalition ne dépend pas seulement de la coalition elle-même, mais aussi des autres coalitions formées.

Dans ce cadre, nous avons les travaux de (Rahwan et al. 2009a) et (Rahwan et al. 2012b). D'autres travaux proposent différentes méthodes de calcul à base d'heuristiques utilisant des techniques telles que les algorithmes génétiques (Sen et al. 2000), (Keinänen et Keinänen 2008) et (Scully et al. 2004a).

1.1.2 Formation de coalitions avec des agents égoïstes

Chaque agent agit uniquement pour satisfaire ses propres objectifs pour maximiser son utilité, sans considérer l'utilité des autres agents avec qui il partage le système multi-agent. Dans ce cas, on distingue deux catégories de jeux différents, à savoir, les jeux à utilité transférable et les jeux à utilité non transférable.

Dans une phase coalitionnelle, les agents forment des coalitions et chaque coalition choisit ses actions en fonction des objectifs ayant conduit à sa formation.

Si une coalition est formée dans un contexte d'un jeu à utilité transférable, alors la récompense est assignée à la coalition. Donc, les agents reçoivent une utilité commune qu'ils doivent ensuite se partager.

Exemple 1.4 Soit 3 enfants (e_1, e_2, e_3) où chacun a une certaine somme d'argent de poche, par exemple, e_1 a 4 euros, e_2 a 3 euros et e_3 a 5 euros.

L'objectif de chaque enfant est de maximiser la quantité de crème glacée qu'il peut manger avec son argent de poche.

Si le pot de 500g coûte 7 euro, celui de 750g coûte 9 euro et celui de 1kg coûte 11 euro, alors aucun des enfants n'est capable d'acheter de la crème glacée sans coopérer avec au moins un autre enfant. De plus suivant les coopérations qu'ils établissent, ils vont augmenter ou diminuer leur pouvoir d'achat.

Dans cette situation, la récompense de chaque groupe d'enfants est la quantité de crème glacée qu'il a achetée

et le partage de la crème glacée d'un groupe peut être fait de manière aléatoire ou suivant la contribution de chaque enfant du groupe.

Dans le contexte d'un jeu à utilité non transférable, chaque agent a sa propre fonction d'utilité ou sa propre relation de préférence sur les coalitions auxquelles il peut appartenir. Donc, la récompense d'un agent dépend de ses actions dans sa coalition.

Exemple 1.5 Soit n chercheurs travaillant dans m universités différentes qui peuvent former x associations temporairement pour rédiger un article scientifique.

Si chacun de ces n chercheurs peut travailler avec les autres, alors la qualité d'un papier dépend du groupe de chercheurs qui l'a rédigé (les auteurs).

Dans le cas où un papier est accepté, chaque auteur reçoit une récompense de sa propre université (promotion, bonus, etc.). Il est clair que ces récompenses sont non transférables entre les chercheurs.

1.1.2.1 Jeux à utilité transférable

Dans un jeu à utilité transférable, si la récompense d'une coalition dépend des actions des autres coalitions, alors on l'appelle, jeu à fonction de partition. Sinon, si la récompense de chaque coalition dépend uniquement des actions de ses membres, alors on l'appelle, jeu à fonction caractéristique. Nous reprenons les mêmes définitions que celles qui sont exposées dans le cadre des jeux coalitionnels pour des agents coopératifs en plus de celles qui sont inhérentes aux jeux à utilités transférables. Nous considérons un ensemble $A = \{a_1, a_2, \dots, a_n\}$ de n agents. Une coalition C est un sous-ensemble non vide de A et une structure de coalitions SC est une partition de A . La coalition équivalente à A , qui est formée par tous les agents, est appelée la grande coalition. À chaque coalition, on attribue une utilité définie par une fonction $v : 2^A \mapsto \mathbb{R}^+$, appelée fonction caractéristique.

Définition 1.5 Un jeu à utilité transférable est défini par un couple (A, v) où A est un ensemble de n agents et v une fonction caractéristique. Les agents d'une coalition C doivent se partager une récompense $v(C)$. Chaque agent a_i reçoit un paiement x_i .

Définition 1.6 Pour une coalition $C = \{a_1, a_2, \dots, a_m\}$ de m agents, la distribution de paiement de cette coalition $C \subseteq A$ est définie par le vecteur $\vec{x}_C = (x_1, x_2, \dots, x_m) \in (\mathbb{R}_+)^m$, qui est la distribution d'utilité que reçoivent les agents qui sont membres de C .

Définition 1.7 Soit un jeu coalitionnel (A, v) . Soit une coalition C de m agents et un vecteur $\vec{x}_C = (x_1, x_2, \dots, x_m) \in (\mathbb{R}_+)^m$ qui est la distribution de paiements que reçoivent les agents de C . $\vec{x}_C$ est faisable lorsque $\sum_{i=1}^m x_i \leq v(C)$. $\vec{x}_C$ est efficace si $\sum_{i=1}^m x_i = v(C)$.

Il peut y avoir plusieurs distributions de paiement pour une coalition ou une structure de coalitions. Certaines distributions peuvent être meilleures que d'autres. Ce qui amène la définition suivante.

Définition 1.8 Soient $(SC, \vec{x})$ et $(SC', \vec{x}')$, deux configurations de paiement de deux structures de coalitions identiques ou différentes. $(SC, \vec{x})$ domine $(SC', \vec{x}')$ au sens de Pareto, lorsque tous les agents reçoivent dans $(SC, \vec{x})$, un meilleur paiement ou un paiement équivalent à celui qu'ils reçoivent dans $(SC', \vec{x}')$, c'est-à-dire, $\forall i : x_i \in \vec{x}, y_i \in \vec{x}', x_i \geq y_i$.

Quelques classes importantes de jeux coalitionnels.

Définition 1.9 Un jeu coalitionnel (A, v) est monotone si $\forall (C, C') \in E_C^2 : C \subseteq C'$ on a $v(C) \leq v(C')$. Ainsi, la grande coalition admet la plus grande valeur caractéristique, mais elle n'est pas forcément la plus intéressante à former pour les agents.

Définition 1.10 Un jeu coalitionnel (A, v) est super-additif, si $\forall (C, C') \in E_C^2 : C \cap C' = \emptyset$ on a $v(C \cup C') \geq v(C) + v(C')$.

Cette définition signifie que, si deux coalitions d'un jeu super-additif fusionnent, l'utilité de la coalition résultante est supérieure à la somme de ces deux coalitions prises séparément.

Sous l'hypothèse de super-additivité, la grande coalition $C = A$ obtient la plus grande valeur de paiement. Elle pourrait sembler être, par conséquent, la structure de coalitions la plus intéressante pour tous les agents, puisqu'elle permet aux agents d'acquérir un paiement total supérieur à n'importe quelle autre structure de coalitions SC . En effet, puisque $V(A) \geq \sum_{C \in SC} v(C)$, toute distribution de paiement réalisable pour une quelconque structure de coalitions, est réalisable pour $C = A$ avec un supplément d'utilité $v(A) - \sum_{C \in SC} v(C)$ à partager.

Définition 1.11 Un jeu coalitionnel (A, v) est convexe si $\forall (C, C') \in E_C^2$ on a $v(C \cup C') \geq v(C) + v(C') - v(C \cap C')$.

Propriété 1.2 Un jeu convexe est super-additif. Cependant, un jeu super-additif n'est pas nécessairement convexe.

Démonstration. Si (A, v) est un jeu convexe, alors $\forall (C, C') \in E_C^2$, on a $v(C \cup C') \geq v(C) + v(C') - v(C \cap C')$. Puisque $v(C \cap C') \geq 0$, alors que, $v(C \cup C') \geq v(C) + v(C')$ est la définition de la super-additivité, nous déduisons qu'un jeu convexe est aussi super-additif. Cependant, un jeu super-additif ne peut être convexe que si $v(C \cap C') = 0$. $\square$

Définition 1.12 Un jeu coalitionnel (A, v) est simple s'il est monotone et que sa fonction caractéristique

prend la valeur 0 ou 1, c'est-à-dire, $v(C) \in \{0, 1\}$ pour chaque coalition $C \subseteq A$. La monotonie signifie que, si $v(C) = 1 : C \subseteq D$, alors $v(D) = 1, \forall D \subseteq A$.

Définition 1.13 Dans un jeu coalitionnel (A, v) simple, un agent a_i est un agent véto, si $v(C) = 0 \forall C \subseteq A \setminus \{i\}$ et $v(A \setminus \{i\}) = 0$ à cause de la monotonie de tout jeu coalitionnel simple.

Dans la suite de cette section, nous nous limitons aux jeux coalitionnels super-additifs. Dans ce cas, on se pose moins la question de la recherche de la meilleure structure de coalitions puisqu'il s'agit de la grande coalition. Mais les agents ayant formé la grande coalition doivent définir, comment se partager l'utilité $v(A)$, qu'ils ont reçu de la manière la plus équitable possible pour tous les agents.

La distribution des paiements.

Chaque distribution des paiements pour les agents d'une coalition a des propriétés qui le caractérisent comme nous le montrons dans ce qui suit.

Propriété 1.3 Un agent égoïste a_i n'accepte la formation d'une coalition C que s'il y a une rationalité individuelle, c'est-à-dire, $\forall a_i \in C, x_i \in \bar{x}_C, x_i \geq v(\{a_i\})$, ce qui signifie que, le paiement qu'il obtient est supérieur au paiement qu'il peut obtenir en formant la coalition singleton.

La rationalité individuelle est une condition que doit vérifier une configuration de paiement raisonnable, mais elle ne garantit pas une distribution équitable du paiement.

Exemple 1.6 Soit un jeu coalitionnel (A, v) tel que $A = \{a_1, a_2, a_3\}$, $v(\{a_1\})=v(\{a_2\})=v(\{a_3\})=5$, $v(\{a_1, a_2\})=6$, $v(\{a_1, a_3\})=v(\{a_2, a_3\})=10$ et $v(\{a_1, a_2, a_3\})=30$.

Par exemple, la configuration de paiement $(\{A\}, (4, 4, 22))$ n'est pas rationnelle pour l'agent a_1 qui préférera former une coalition singleton ou une coalition avec a_3 pour obtenir $x_1 = 5$ au lieu de $x_1 = 4$ dans une grande coalition. Idem pour l'agent a_2 . L'agent a_3 serait alors contraint de former la coalition singleton $\{a_3\}$ et d'obtenir $x_3 = 5$. Une configuration de paiement qui est rationnelle pour chacun des agents serait, par exemple, $(\{A\}, (7, 16, 7))$.

Cependant, la rationalité individuelle ne garantit pas une configuration de paiement équitable. En effet, la configuration de paiement $(\{A\}, (7, 16, 7))$ est rationnelle pour chaque agent, mais semble plus avantageuse pour a_2 qui récupère presque entièrement le surplus gagné par la formation de la grande coalition $\{a_1, a_2, a_3\}$.

Nous pouvons remarquer dans l'exemple ci-dessus que la présence de a_3 dans une coalition augmente la valeur de celle-ci.

De manière intuitive, nous pouvons rendre équitable la configuration de paiement en favorisant, par exemple, l'agent qui apporte plus de valeur ajoutée aux coalitions. Comme le jeu est super-additif, donc si les trois agents forment la grande coalition, il paraîtrait plus juste d'attribuer à a_3 un paiement x_3 supérieur aux paiements de a_1 et de a_2 .

Il existe différentes méthodes permettant une distribution équitable des paiements. Dans le cadre des jeux super-additifs où les agents forment la grande coalition, une méthode très utilisée est la valeur de Shapley (Shapley 1944) qui tient compte de la valeur ajoutée par la présence d'un agent dans une coalition (contribution marginale).

Définition 1.14 La contribution marginale $\mu_{a_i}(C)$ d'un agent a_i dans une coalition $C : a_i \notin C$, est égale à l'utilité qu'apporte l'agent a_i s'il rejoint la coalition $C \cup \{a_i\}$. Cette contribution s'écrit sous la forme : $\mu_{a_i}(C) = v(C \cup \{a_i\}) - v(C)$.

Définition 1.15 Deux agents a_i et a_j sont dits interchangeables s'ils contribuent toujours de la même manière à toutes les coalitions formées avec les autres agents, c'est-à-dire, $\mu_{a_i}(C) = \mu_{a_j}(C)$, $\forall C \subseteq A$.

Définition 1.16 Un agent est dit inutile s'il n'apporte aucune utilité supplémentaire aux coalitions des autres agents, c'est-à-dire, $v(C) = v(C \cup \{a_i\})$, $\forall C \subseteq A$.

Valeur de Shapley.

Le calcul de la valeur de Shapley se fonde sur les trois propriétés ci-dessous, pour garantir une distribution de paiement équitable (Shapley 1944).

Propriété 1.4 Si deux agents a_i et a_j sont interchangeables, ils doivent recevoir le même paiement, c'est-à-dire, $x_i = x_j$ (axiome de symétrie des paiements).

Propriété 1.5 Un agent a_i inutile dans une coalition C , doit avoir un paiement égal à 0 dans C , c'est-à-dire, $x_i = 0$ (axiome des agents inutiles).

Propriété 1.6 Pour deux jeux coalitionnels (A, v_1) et (A, v_2) , supposons que x_i est le paiement de a_i dans (A, v_1) et x'_i est le paiement de a_i dans (A, v_2) . Si le jeu coalitionnel (A, v_3) est défini de telle manière que $\forall C \in E_C$, $v_3(C) = v_1(C) + v_2(C)$, alors le paiement x_i'' de a_i dans (A, v_3) doit être égal à la somme des paiements reçus dans (A, v_1) et (A, v_2) , c'est-à-dire, $x_i'' = x_i + x'_i$ (axiome d'additivité des paiements).

D'après (Osborne et Rubinstein 1994), nous avons le théorème ci-dessous.

Théorème 1 Soit un jeu coalitionnel (A, v) , il existe une unique distribution de paiement efficace de la grande coalition vérifiant les trois axiomes concernant la symétrie, l'agent inutile et l'additivité des paiements. Cette unique distribution de paiement est la valeur de Shapley.

Calcul de la valeur de Shapley.

Soit la grande coalition A et un agent a_i . Pour comprendre comment se calcule la valeur de Shapley, il faut considérer le fait que, la grande coalition est formée suivant un schéma de construction séquentielle.

Définition 1.17 *Un schéma de construction d'une coalition C est une suite de coalitions $[\emptyset, C_1, C_2, \dots, C_{n-1}, C]$, où $|C_i| = i$ et C_{i+1} telle qu'il existe un agent $a_k \in A : a_k \notin C_i$ et $C_{i+1} = C_i \cup \{a_k\}$.*

Exemple 1.7 *Pour 4 agents qui forment une coalition $C = \{a_1, a_2, a_3, a_4\}$, un schéma de construction de C peut être, $[\emptyset, \{a_4\}, \{a_2, a_4\}, \{a_2, a_3, a_4\}, \{a_1, a_2, a_3, a_4\}]$.*

À chaque schéma de construction de C s'associe une permutation σ des agents de C qui représente l'ordre d'apparition des agents dans les coalitions successives.

Exemple 1.8 *La permutation correspondant au schéma de construction $[\emptyset, \{a_4\}, \{a_2, a_4\}, \{a_2, a_3, a_4\}, \{a_1, a_2, a_3, a_4\}]$ est $\sigma = (4, 2, 3, 1)$, c'est-à-dire, il y a d'abord la présence de l'agent a_4 , puis de l'agent a_2 , puis de l'agent a_3 et enfin celle de l'agent a_1 dans les coalitions successives.*

Ainsi, il existe autant de schémas de construction que de permutations du nombre d'agents de C . L'ensemble des permutations des entiers de 1 à k est noté $\Gamma(k)$. Pour une permutation $\sigma \in \Gamma(k)$ donnée, la dernière coalition ne contenant pas l'agent a_i , dans le schéma de construction associé à la permutation σ , est notée $C_i(\sigma)$.

Exemple 1.9 *Soit une coalition $C = \{a_1, a_2, a_3\}$ formée de 3 agents. L'ensemble des permutations des entiers de 1 à 3 est $\Gamma(3) = \{(a_1, a_2, a_3), (a_1, a_3, a_2), (a_2, a_1, a_3), (a_2, a_3, a_1), (a_3, a_1, a_2), (a_3, a_2, a_1)\}$. En considérant le schéma de construction $[\emptyset, \{a_1\}, \{a_1, a_3\}, \{a_1, a_2, a_3\}]$, alors on a la permutation $\sigma = (1, 3, 2)$. Ainsi, $C_1(\sigma) = \emptyset$, $C_2(\sigma) = \{a_1, a_3\}$ et $C_3(\sigma) = \{a_1\}$.*

Définition 1.18 *La valeur de Shapley d'un agent a_i , que nous notons $sh(a_i)$ est calculée en faisant la moyenne de toutes les contributions marginales de l'agent a_i lors du passage de $C_i(\sigma)$ à $C_1(\sigma) \cup \{a_i\}$, pour tous les schémas de construction possibles menant à la grande coalition, c'est-à-dire pour toutes les permutations $\sigma \in \Gamma(n)$.*

$$sh(a_i) = \frac{1}{n!} \sum_{\sigma \in \Gamma(n)} \mu_{a_i}(C_i(\sigma)) \quad (1.2)$$

Exemple 1.10 *Soit un jeu coalitionnel (A, v) tel que, $A = \{a_1, a_2, a_3\}$. Dans ce cas, l'ensemble des permutations $\Gamma(3) = \{(a_1, a_2, a_3), (a_1, a_3, a_2), (a_2, a_1, a_3), (a_2, a_3, a_1), (a_3, a_1, a_2), (a_3, a_2, a_1)\}$.*

Supposons que, $v(\{a_1\}) = v(\{a_2\}) = v(\{a_3\}) = 5$, $v(\{a_1, a_2\}) = 6$, $v(\{a_1, a_3\}) = v(\{a_2, a_3\}) = 9$ et $v(\{a_1, a_2, a_3\}) = 30$.

Permutations $\sigma \in \Gamma(6)$	$\mu_{a_1}(C_1(\sigma))$	$\mu_{a_2}(C_2(\sigma))$	$\mu_{a_3}(C_3(\sigma))$
(a_1, a_2, a_3)	5	1	24
(a_1, a_3, a_2)	5	21	4
(a_2, a_1, a_3)	1	5	24
(a_2, a_3, a_1)	21	5	4
(a_3, a_1, a_2)	4	21	5
(a_3, a_2, a_1)	21	4	5
$\sum_{\sigma \in \Gamma(6)} \mu_{a_i}(C_i(\sigma))_{i \in [1,3]}$	57	57	66
$sh(a_i) = \frac{1}{6} \sum_{\sigma \in \Gamma(6)} \mu_{a_i}(C_i(\sigma))_{i \in [1,3]}$	9.5	9.5	11

Avec ces résultats, nous voyons que les agents a_1 et a_2 , qui sont interchangeables, reçoivent bien le même paiement. L'agent a_3 reçoit un paiement plus élevé, car il augmente la valeur des coalitions auxquelles il appartient, c'est-à-dire, il a une contribution marginale plus élevée que celle de a_1 et a_2 .

La méthode de Shapley distribue la valeur de la grande coalition sur l'ensemble des agents de la coalition, pour cette raison, on dit que c'est une méthode efficace.

Propriété 1.7 Avec la méthode de Shapley, nous avons $\sum_{i=1}^n sh(a_i) = v(A)$ pour tout jeu coalitionnel (A, v) .

La méthode de Shapley n'alloue aucun paiement à un agent inutile d'une coalition.

Propriété 1.8 $\forall a_i \in A : v(C) = v(C \cup a_i), C \subseteq A, sh(a_i) = 0$ pour tout jeu coalitionnel (A, v) .

Si deux agents contribuent de la même façon dans un jeu coalitionnel, alors ils ont la même valeur de Shapley.

Propriété 1.9 $\forall a_i, a_j \in A : v(C \cup a_i) = v(C \cup a_j), C \subseteq A, sh(a_i) = sh(a_j)$ pour tout jeu coalitionnel (A, v) .

La méthode de Shapley a la propriété d'additivité.

Propriété 1.10 $\forall a_i \in A$ si sa valeur de Shapley est $sh(a_i)$ pour un jeu coalitionnel (A, v_1) et $sh'(a_i)$ pour un jeu coalitionnel (A, v_2) , alors $sh''(a_i) = sh(a_i) + sh'(a_i)$ dans (A, v_3) . Sachant que le jeu coalitionnel (A, v_3) est défini de telle manière que $\forall C \in E_C, v_3(C) = v_1(C) + v_2(C)$.

D'après (Kahan et Rapoport 1984), le calcul de la valeur de Shapley est d'une complexité en $\Theta(2^n)$, ce qui rend son utilisation difficile en pratique lorsque le nombre d'agents est important.

Un autre concept de partage est l'indexe de Banzhaf (Banzhaf 1965). Au lieu de faire une moyenne des valeurs de toutes les permutations possibles avec l'ensemble des agents du jeu comme pour la valeur de Shapley, (Banzhaf 1965) propose une moyenne des valeurs de l'ensemble des coalitions possibles avec les agents du jeu.

Définition 1.19 Soit un jeu coalitionnel $G = (A, v)$ tel que $A = \{a_1, a_2, \dots, a_n\}$. L'indexe de Banzhaf d'un agent $a_i \in A$ est,

$$\beta_{a_i}(G) = \frac{1}{2^{n-1}} \sum_{C \subseteq A \setminus \{a_i\}} \delta_{a_i}(C),$$

où $\delta_{a_i}(C) = v(C \cup \{a_i\}) - v(C) : C \subseteq A$

La méthode de Banzhaf n'alloue aucun paiement à un agent inutile de la coalition.

Propriété 1.11 $\forall a_i \in A : v(C) = v(C \cup a_i), C \subseteq A, \beta_{a_i}(G) = 0$ pour tout jeu coalitionnel $G = (A, v)$.

Si deux agents contribuent de la même façon dans un jeu coalitionnel, alors ils ont le même indexe de Banzhaf.

Propriété 1.12 $\forall a_i, a_j \in A : v(C \cup a_i) = v(C \cup a_j), C \subseteq A, \beta_{a_i}(G) = \beta_{a_j}(G)$ pour tout jeu coalitionnel $G = (A, v)$.

La méthode de Banzhaf a la propriété d'additivité.

Propriété 1.13 $\forall a_i \in A$, si son indexe de Banzhaf est $\beta_{a_i}(G_1)$ pour un jeu coalitionnel $G_1 = (A, v_1)$ et $\beta_{a_i}(G_2)$ pour un jeu coalitionnel $G_2 = (A, v_2)$, alors $\beta_{a_i}(G_3) = \beta_{a_i}(G_1) + \beta_{a_i}(G_2)$ dans $G_3(A, v_3)$, sachant que le jeu coalitionnel $G_3 = (A, v_3)$ est défini de telle manière que $\forall C \in E_C, v_3(C) = v_1(C) + v_2(C)$.

Cependant, la méthode de Banzhaf n'est pas toujours une méthode efficace. Cela est d'ailleurs une ascendance qu'a la méthode de Shapley sur celle de Banzhaf.

Propriété 1.14 Avec la méthode de Banzhaf nous pouvons avoir $\sum_{i=1}^n \beta_{a_i}(G) \neq v(A)$ pour tout jeu coalitionnel $G = (A, v)$.

Dans le reste, nous n'utiliserons que la valeur de Shapley, car comme nous venons de le voir, la méthode de Banzhaf n'est pas toujours efficace.

En tenant compte des contributions marginales de chaque agent, la valeur de Shapley et l'indexe de Banzhaf garantissent la propriété de la symétrie, de l'additivité et des agents inutiles

lorsque le jeu est super-additif. Cependant, il peut exister un sous-groupe d'agents pour lequel il serait plus intéressant de se scinder de la grande coalition, pour former une coalition plus petite et se partager ensuite les gains de cette plus petite coalition.

Exemple 1.11 Soit un jeu coalitionnel $(A, v) : A = \{a_i, a_j, a_k\}$, $v(a_i) = v(a_j) = v(a_k) = 10$, $v(a_i, a_j) = v(a_i, a_k) = 100$, $v(a_j, a_k) = 20$ et $v(a_i, a_j, a_k) = 110$.

La valeur de Shapley de chaque agent dans (A, v) est $sh(a_i) = sh(a_j) = 23.33$ et $sh(a_k) = 63.34$. Donc, il est préférable pour a_i de former une coalition de deux agents, car $v(a_i, a_j) = v(a_i, a_k) = 100$ et chaque couple d'agents peut se partager le gain de la coalition formée par leur couple de telle sorte que chacun reçoive un paiement supérieur à sa valeur de Shapley. Dans ce cas, la distribution de paiement est dite instable.

Cela a entraîné la définition du concept de la stabilité dans les jeux à utilité transférable.

Stabilité au sens du Core dans un jeu à utilité transférable.

Définition 1.20 Le vecteur de paiement $\vec{x}$ d'une configuration de paiement $(E_{SC}, \vec{x})$ est dans le Core d'un jeu coalitionnel à utilité transférable (A, v) s'il n'existe aucun agent ou groupe d'agents qui a la possibilité d'obtenir plus que le paiement qui lui est déjà affecté en changeant de coalition. Donc, $\forall C \in E_C$ et $x_i \in \vec{x}$ $\sum_{i \in C} x_i \geq v(C)$

Néanmoins, une distribution de paiement stable peut ne pas exister dans un jeu à utilité transférable, c'est-à-dire, le Core peut être vide. De plus, plusieurs distributions de paiement stables au sens du Core peuvent exister dans un même jeu à utilité transférable.

(Conitzer et Sandholm 2004) ont prouvé que vérifier qu'une distribution de paiement appartient au Core d'un jeu super-additif est de complexité NP-complet. Dans (Conitzer et Sandholm 2003), les auteurs montrent qu'on a une complexité NP-complet, s'il faut vérifier que le Core d'un jeu super-additif est non vide.

Il existe d'autres concepts de stabilité pour les jeux à utilité transférable. Ces concepts sont soit des variantes du Core comme le ϵ -Core et le least-Core, soit d'autres approches de stabilité comme le Kernel et le nucleolus.

Le ϵ -Core d'un jeu à utilité transférable.

Si le Core est vide, nous pouvons espérer trouver une approximation stable des résultats. Pour ce

faire, l'idée du ϵ -Core est de faire une relaxation de certaines notions du Core.

Définition 1.21 *Le vecteur de paiement $\vec{x}$ d'une configuration de paiement $(E_{SC}, \vec{x})$ est dans le ϵ -Core d'un jeu coalitionnel à utilité transférable (A, v) et super-additif, si $\forall C \in E_C$ et $x_i \in \vec{x} \sum_{i \in C} x_i \geq v(C) - \epsilon$, sachant que $\forall C \in E_C$ et $x_i \in \vec{x} \sum_{i \in C} x_i \geq v(C)$.*

D'habitude, le ϵ -Core ne se définit que dans un contexte de jeux super-additifs.

Exemple 1.12 *Soit un jeu coalitionnel $(A, v) : A = \{a_1, a_2, a_3\}$ tel que $v(C) = 1$ si $|C| > 1$, sinon $v(C) = 0$. Si nous choisissons $\epsilon = \frac{1}{3}$, alors $\frac{1}{3}$ -Core est non vide, car, $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}) \in \frac{1}{3}$ -Core. Cependant, ϵ -Core est vide pour tout $\epsilon < \frac{1}{3}$, car $\forall a_i \in A$ si $x_i \geq \frac{1}{3}$, alors $\forall a_j \in A : j \neq i, x_j \leq \frac{2}{3}$ et $v(C \setminus \{a_i\}) = 1$.*

Si une configuration de paiement $(E_{SC}, \vec{x})$ est dans le ϵ -Core d'un jeu coalitionnel à utilité transférable (A, v) et super-additif, le déficit pour toute coalition C est égal au plus à ϵ .

Si pour une coalition C , $x(C) = \sum_{a_i \in C} x_i$ et si on s'intéresse aux résultats qui minimisent les déficits $v(C) - x(C)$ dans le pire des cas, cela correspond à la définition du least-Core.

Le least-Core d'un jeu à utilité transférable.

Définition 1.22 *Soit un jeu coalitionnel $G = (A, v)$, la valeur du least-Core est $\epsilon^*(G) = \inf \{\epsilon | \epsilon\text{-Core de } G \text{ est non vide}\}$, si $\epsilon^*(G)$ -Core est non vide alors on l'appelle le least-Core de G .*

Exemple 1.13 *Soit un jeu coalitionnel $(A, v) : A = \{a_1, a_2, a_3\}$ tel que $v(C) = 1$ si $|C| > 1$, sinon $v(C) = 0$. Comme nous l'avons indiqué dans l'exemple précédent, $\frac{1}{3}$ -Core est non vide car $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}) \in \frac{1}{3}$ -Core. Cependant, ϵ -Core est vide pour tout $\epsilon < \frac{1}{3}$. Ainsi, le least-Core est égal au $\frac{1}{3}$ -Core et la valeur du least-Core est égale à $\frac{1}{3}$.*

Le least-Core est donc toujours non vide. Cependant, il peut y avoir plusieurs valeurs de least-Core. Il est alors nécessaire d'identifier celle qui donne une plus grande stabilité.

Le nucleolus d'un jeu à utilité transférable.

Définition 1.23 Une configuration de paiement $(E_{SC}, \vec{x})$ d'un jeu (A, v) est une imputation s'il satisfait la rationalité individuelle de chaque agent, c'est-à-dire, $\forall a_i \in A, x_i \geq v(\{a_i\})$.

Définition 1.24 Le vecteur des déficits d'une distribution de paiement $\vec{x}$ pour un ensemble de coalitions E_C est une liste des déficits de l'ensemble E_C . Cette liste est ordonnée du plus grand déficit au plus petit déficit.

Définition 1.25 Le nucleolus d'un jeu $G = (A, v)$ est une imputation qui minimise le déficit le plus grand, puis le second déficit le plus grand, etc. Autrement dit, le nucleolus est une imputation qui minimise le vecteur des déficits d'une configuration de paiement $(E_{SC}, \vec{x})$.

En comparaison, nous voyons que le least-Core minimise le déficit maximal alors que le nucleolus est une imputation qui minimise le vecteur des déficits.

Stabilité au sens du Kernel.

Soit un jeu coalitionnel (A, v) tel que $I(a_i, a_j) = \{C \subseteq A : a_i \in C, a_j \notin C\}$.

Définition 1.26 Si $\vec{x}$ est une imputation, le surplus $S(a_i, a_j)$ d'un agent a_i par rapport à un agent a_j est égal à : $S(a_i, a_j) = \max_{C \in I(a_i, a_j)} d(C)$ où $d(C) = v(C) - x(C)$ et $x(C) = \sum_{a_i \in C} x_i$.

Définition 1.27 Un agent a_i dépasse un agent a_j pour une imputation $\vec{x}$, si $S(a_i, a_j) > S(a_j, a_i)$.

Si un agent a_i dépasse un agent a_j pour une imputation $\vec{x}$, alors a_i peut réclamer une partie de la récompense de a_j . Cependant, le montant que a_i peut réclamer est conditionné par la rationalité individuelle de a_j , c'est-à-dire, $x_j \geq v(\{a_j\}) : x_j \in \vec{x}$.

Définition 1.28 Deux agents a_i et a_j sont en équilibre par rapport à une imputation $\vec{x}$ si l'une des conditions suivantes est remplie :

- $S(a_i, a_j) = S(a_j, a_i)$.
- $S(a_i, a_j) > S(a_j, a_i)$ et $x_j = v(\{a_j\}) : x_j \in \vec{x}$.
- $S(a_i, a_j) < S(a_j, a_i)$ et $x_i = v(\{a_i\}) : x_i \in \vec{x}$.

Définition 1.29 Une imputation $\vec{x}$ est dans le Kernel, si pour tout couple d'agents a_i et a_j , ils sont en équilibre par rapport à $\vec{x}$.

En comparaison, nous voyons que le nucleolus est inclus ou égal au kernel.

1.1.2.2 Les jeux à utilité non transférable

Dans les jeux à utilité transférable, on observe une échelle commune de mesure de satisfaction des agents permettant de diviser la valeur de la coalition en une distribution de paiements pour ses agents.

Dans les jeux à utilité non transférable, chaque agent dispose de sa propre fonction d'utilité sur les coalitions dont il est membre. Les utilités ne sont pas comparables, elles induisent uniquement une relation de préférence sur les coalitions comme dans le cas des jeux hédoniques.

Ainsi, la plupart des concepts précédemment étudiés ne sont plus applicables. Pour les différents concepts de solutions étudiés, nous prenons comme exemple, les jeux hédoniques.

Définition 1.30 *Un jeu coalitionnel à utilité non transférable est un couple (A, u) où $A = \{a_1, a_2, \dots, a_{n-1}, a_n\}$ est un ensemble fini de n agents et $u = \{u_1, u_2, \dots, u_{n-1}, u_n\}$ est un ensemble fini de n fonctions d'utilité appartenant aux n agents sur l'ensemble des coalitions E_C . u_i est la fonction d'utilité de l'agent a_i sur l'ensemble des coalitions E_{C_i} auxquelles il participe. Ainsi,*

$$u_i : E_{C_i} \mapsto \mathbb{R}^+$$

$$C \mapsto u_i(C)$$

Définition 1.31 *Un jeu hédonique est un couple $(A, \succeq)$ où $A = \{a_1, a_2, \dots, a_{n-1}, a_n\}$ est un ensemble fini de n agents et $\succeq = \{\succeq_1, \succeq_2, \dots, \succeq_{n-1}, \succeq_n\}$ est un ensemble fini de n relations de préférence appartenant aux n agents sur l'ensemble des coalitions E_C .*

$\succeq_i$ est la préférence de l'agent a_i sur l'ensemble des coalitions E_{C_i} auxquelles il participe. La préférence stricte d'un agent a_i se note $\succ_i$.

Les agents sont égoïstes et ne cherchent à former une coalition que si leur utilité est satisfaisante. L'utilité d'un agent ne dépend que de la coalition à laquelle il appartient, ce qui fait que le problème du partage de gain ne se pose plus.

Dans les jeux coalitionnel à utilité non transférable, s'intéresser à la stabilité des coalitions formées, revient à vérifier pour chaque coalition C si réellement les agents qui forment C sont satisfaits de leur choix ou pas.

Concepts de stabilité les plus usités.

Définition 1.32 *Soit un jeu hédonique $(A, \succeq)$. Une structure de coalitions SC est rationnelle individuellement lorsque chaque agent préfère sa coalition qui est dans SC plutôt que sa coalition singleton, c'est-à-dire, $\forall a_i \in A, SC \succeq_i \{a_i\}$.*

Définition 1.33 Soit un jeu hédonique $(A, \succeq)$. Une structure de coalitions SC domine une autre structure de coalitions SC' au sens de Pareto, lorsque chaque agent préfère sa coalition qui est dans SC plutôt que, celle qui est dans SC' , c'est-à-dire, $\forall a_i \in A, SC \succeq_i SC'$.

Cette dominance de SC sur SC' devient stricte s'il existe au moins un agent qui préfère strictement SC c'est-à-dire, $\forall a_i \in A, SC \succeq_i SC'$ et $\exists a_j \in A : SC \succ_j SC'$.

Définition 1.34 Une structure de coalitions SC d'un ensemble E_{SC} de structure de coalitions est un optimum au sens de Pareto, si et seulement si, aucune autre structure de coalitions $SC' \in E_{SC}$ ne domine strictement SC au sens de Pareto. On note :

$$\neg(\exists SC' \in E_{SC}, SC' \neq SC : \forall a_i \in A, SC' \succeq_i SC \text{ et } \exists a_j \in A : SC' \succ_j SC)$$

Définition 1.35 Une structure de coalitions SC est stable au sens de Nash (Nash stable), si aucun agent de A est motivé à quitter unilatéralement sa coalition $C \in SC$ pour rejoindre une autre coalition $C' \in SC$, c'est-à-dire,

$$\neg(\exists a_i \in A : a_i \in C \text{ et } C \in SC, \exists C' \in SC : C' \cup \{a_i\} \succ_i C).$$

Définition 1.36 Une structure de coalitions SC est stable individuellement, si aucun agent $a_i \in A$ ne cherche à rejoindre une autre coalition C' existante où les membres de C' seraient intéressés par la venue de a_i dans leur coalition, c'est-à-dire,

$$\neg(\exists a_i \in A : a_i \in C \text{ et } C \in SC, \exists C' \in SC : [C' \cup \{a_i\} \succ_i C] \text{ et } [\forall a_j \in C', C' \cup \{a_i\} \succ_j C']).$$

Définition 1.37 Une structure de coalitions SC est dans un état de stabilité contractuelle individuelle, s'il n'existe pas un agent a_i de $C \in SC$ qui est intéressé par rejoindre une autre coalition $C' \in SC$, dont les membres seraient intéressés par l'arrivée de a_i et que les agents de $C \setminus \{a_i\}$ préféreraient également le départ de a_i . On note,

$$\neg(\exists a_i \in A : a_i \in C \text{ et } C \in SC, \exists C' \in SC : [C' \cup \{a_i\} \succ_i C] \text{ et } [\forall a_j \in C', C' \cup \{a_i\} \succ_j C'] \text{ et } [\forall a_k \in C, C \setminus \{a_i\} \succ_k C]).$$

De ces définitions de concepts de stabilité, nous pouvons identifier un ensemble de relations entre elles.

Remarque 1.1 La stabilité individuelle implique la rationalité individuelle du moment qu'aucun agent ne voudrait quitter sa coalition pour rejoindre une autre coalition ou pour être dans une coalition singleton.

Remarque 1.2 Si une structure de coalitions est stable au sens de Nash alors elle est individuellement stable et contractuellement individuellement stable. Mais le contraire n'est pas toujours vrai, car une structure de coalitions individuellement stable n'est pas toujours stable au sens de Nash.

Exemple 1.14 Soit $A = \{a_1, a_2, a_3\}$ un ensemble d'agents qui ont les préférences de coalitions ci-dessous : $a_1 : \{a_1, a_2\} \succ_1 \{a_1\} \succ_1 \{a_1, a_2, a_3\} \succ_1 \{a_1, a_3\}$.

$a_2 : \{a_1, a_2\} \succ_2 \{a_2\} \succ_2 \{a_1, a_2, a_3\} \succ_2 \{a_2, a_3\}.$

$a_3 : \{a_1, a_2, a_3\} \succ_3 \{a_2, a_3\} \succ_3 \{a_1, a_3\} \succ_3 \{a_3\}.$

La structure de coalitions $\{a_1, a_2\}, \{a_3\}$ est individuellement stable et contractuellement individuellement stable, mais elle n'est pas stable au sens de Nash.

Remarque 1.3 Si une structure de coalitions est individuellement stable, cela implique qu'elle est contractuellement individuellement stable.

Remarque 1.4 D'après (Bogomolnaia et Jackson 2002), une structure de coalitions optimale au sens de Pareto est contractuellement individuellement stable.

Exemple 1.15 Soit $A = \{a_1, a_2, a_3\}$ un ensemble d'agents qui ont les préférences sur les coalitions ci-dessous :

$a_1 : \{a_1, a_2, a_3\} \succ_1 \{a_1, a_2\} \succ_1 \{a_1\}.$

$a_2 : \{a_1, a_2, a_3\} \succ_2 \{a_1, a_2\} \succ_2 \{a_2\}.$

$a_3 : \{a_1, a_3\} \succ_3 \{a_1, a_2, a_3\} \succ_3 \{a_2, a_3\} \succ_3 \{a_3\}.$

La structure de coalitions $\{\{a_1, a_3\}, \{a_2\}\}$ n'est pas individuellement rationnelle car l'agent a_1 préfère la coalition singleton $\{a_1\}$ plutôt qu'une coalition avec a_3 .

La structure de coalitions $\{\{a_1, a_2\}, \{a_3\}\}$ est individuellement rationnelle, mais n'est pas un optimum de Pareto, car elle est dominée par la structure de coalitions $\{\{a_1, a_2, a_3\}\}.$

La structure de coalitions $\{\{a_1, a_2, a_3\}\}$ est un optimum de Pareto.

Définition 1.38 Une structure de coalitions SC est dite stable au sens du Core, si aucune coalition C' de l'ensemble des coalitions E_C n'est préférée par tous ses membres par rapport à leur coalition $C \in SC$. On note, $\nexists C' \in E_C : \forall a_i \in C, C' \succ_i C$.

Ainsi, le Core d'un jeu à utilité non transférable représente, l'ensemble des structures de coalitions stables au sens du Core.

Définition 1.39 Si une coalition C' empêche la stabilité d'une structure de coalitions SC , alors on dit que C' est une coalition bloquante pour SC et on l'exprime par :

$\exists C' \in E_C : \forall a_i \in C, C' \succ_i C$.

1.2 PRINCIPES DES MÉTHODES DE FORMATION DE COALITIONS STABLES

Comme nous pouvons le voir, la formation de coalitions a reçu beaucoup d'attention dans le domaine des systèmes multi-agents afin de proposer des solutions de coopération aux agents. Plusieurs de ces méthodes de formation de coalitions peuvent aborder la question de la stabilité

des coalitions ou non. Cependant, le défi de la stabilité dans les contextes incertains et dynamiques du type discuté dans notre thèse n'était pas clairement abordé jusqu'ici dans le domaine.

Généralement, les mécanismes de formation de coalitions considèrent la stabilité comme une propriété qui indique que les agents n'ont aucune intention de quitter leurs coalitions avant l'achèvement des tâches. Cependant, ils ne prennent pas en compte le contexte dynamique et stochastique dans lequel un agent peut devenir indisponible quelles que soient ses intentions au moment de la formation de sa coalition.

La complexité associée à la stabilité des coalitions qui se trouve être une barrière pour des solutions actuelles est que dans le cas de N agents, il est possible de former 2^N coalitions et N -Bell partitions d'agents (Bell 1934) (Levine et Dalton 1962). De plus, la validité des solutions (coalitions ou partitionnement) peut dépendre de la dynamique des tâches à exécuter par les agents et de la disponibilité ou des incertitudes des agents.

En se fondant sur le critère de la prise en charge de la dynamique de l'environnement et les stratégies des agents, nous avons classé en deux catégories les méthodes de formation de coalitions stables :

- 1- les méthodes ne supportant pas une inférence dynamique,
- 2- les méthodes faisant une inférence dynamique.

Nous comprenons par inférence dynamique la prise en charge dynamique des événements aléatoires (disponibilités des agents, dynamique des tâches, etc.) qui peuvent interférer avec le processus normal de formation des coalitions. Une classification des deux catégories est donnée par la figure 1.3 ci-dessous.

Nous détaillons dans la section suivante, les méthodes de formation de coalitions stables ne supportant pas une inférence dynamique.

1.2.1 Formation de coalitions stables sans inférence dynamique

Dans cette section, nous présentons les méthodes qui ne tiennent pas compte des changements dynamiques du contexte une fois que le processus de coalitions est entamé. Ces méthodes sont, généralement, soit fondées sur l'utilité soit sur la complémentarité (Conte et Sichman 1995)

Les méthodes fondées sur l'utilité cherchent d'abord à générer les structures de coalitions, en faisant un partitionnement de l'ensemble des agents en coalitions. Ce partitionnement vise à maximiser le

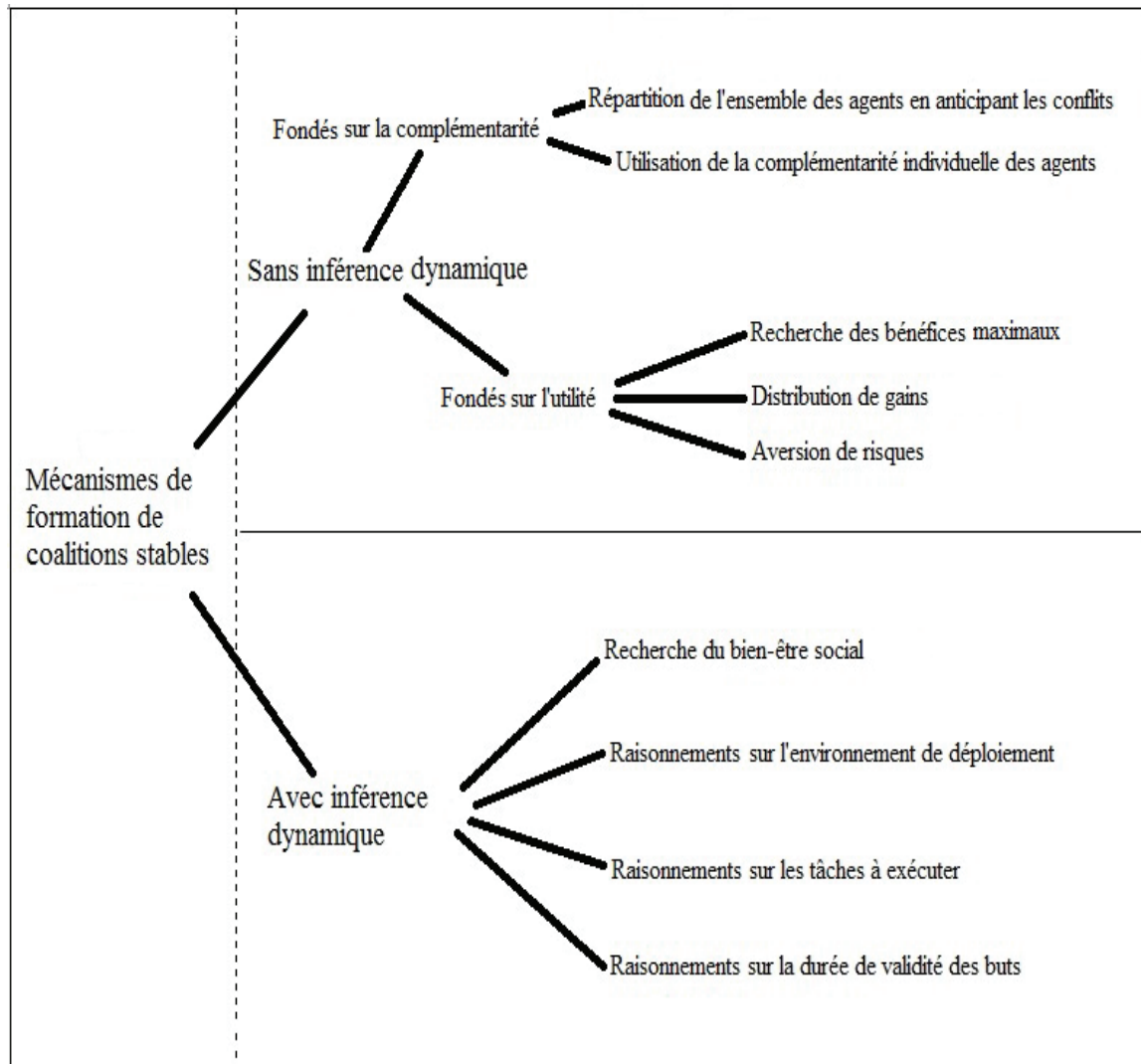


FIGURE 1.3 – Méthodes de formation de coalitions stables.

gain des agents lors de l'accomplissement des tâches tout en minimisant les ressources utilisées, le temps passé pour l'exécution des tâches et éventuellement distribuer le bénéfice parmi les participants de chaque coalition. Cela est le cas pour certaines méthodes comme (Rahwan et Jennings 2008b) (Rahwan et al. 2009b) et leur amélioration (Rahwan et al. 2012a). L'adaptation de ces algorithmes à un contexte incertain et stochastique de formation de coalitions stables est difficile du fait de la centralisation et de la complexité des calculs à effectuer du point de vue informatique. Cette complexité est accentuée si les tâches sont stochastiques et en présence des incertitudes en ce qui concerne la disponibilité, les préférences et les dépendances des agents. Même si une solution hybride de ces algorithmes qui fournit un résultat en temps réel est proposée, il est toujours nécessaire d'avoir des connaissances a priori et de recalculer les meilleures coalitions en réponse à chaque changement sur les tâches ou sur les agents.

(Abdallah et Lesser 2004) se sont intéressés au paiement reçu après la réalisation des tâches. Ils proposent un algorithme distribué d'approximation permettant aux agents de réaliser un sous-ensemble de tâches. L'algorithme est fondé sur une approche organisationnelle où il y a un partage de rôles des agents en managers et subordonnés (les subordonnés pouvant être à leur tour managers d'autres subordonnés, etc.) et dans lequel les agents managers connaissent les ressources de leurs subordonnés. L'idée générale de l'algorithme est, lorsqu'un agent identifie une tâche à réaliser, il prend soit la décision de la réaliser par son équipe, soit il confie la tâche à son supérieur hiérarchique, par exemple, en cas de manque de ressources. De plus, ces travaux se fondent sur un apprentissage par renforcement en modélisant chaque manager comme un MDP pour optimiser les décisions d'allocation locale des managers. L'algorithme d'apprentissage par renforcement utilisé est le Q-learning (Sutton et Barto 1998) où plus les managers gagnent en expériences de répartitions de tâches, plus leurs prises de décisions sont pertinentes. Cependant, pour éviter l'utilisation d'un organisateur central, ils supposent que le temps est divisé par épisodes et dans chacun de ces épisodes, chaque agent reçoit un ensemble de tâches. Une fois qu'une tâche est allouée à une coalition, les agents de celle-ci ne peuvent plus intégrer de nouvelles tâches jusqu'à la fin de l'épisode. De plus, les auteurs font l'hypothèse qu'il n'y a pas de nouveaux agents ou de nouvelles tâches qui arrivent dans le système durant un épisode et que les agents sont toujours disponibles. Par ailleurs, leurs travaux ne se focalisent pas nécessairement sur une recherche de la stabilité des coalitions.

(Scully et al. 2004b) considèrent une place de marché dynamique d'agents égoïstes avec des capacités différentes. Chaque agent tente de former une coalition pour effectuer une tâche. Avant de proposer une coalition, un agent doit déterminer l'ensemble optimal des agents avec qui partager sa coalition pour cette tâche. Leur méthode suppose que, pour déterminer la coalition optimale, chaque agent peut calculer de manière centralisée l'ensemble des valeurs des coalitions possibles avec les autres agents et que durant ce calcul aucune indisponibilité n'est envisagée pour l'agent qui effectue ce calcul. Chaque agent se fonde sur plusieurs critères (par exemple, le coût, le temps, etc.) pour déterminer la valeur d'une coalition. Toutefois, en raison de l'importance de certains critères et de la tendance qui peut faire varier un critère au fil du temps, il est difficile d'obtenir une véritable évaluation d'une coalition donnée. C'est pourquoi, ils proposent une solution fondée sur la dominance de Pareto couplée avec un algorithme de Pondération Inverse à la Distance (PID). La PID est une méthode d'interpolation spatiale, c'est-à-dire, un processus permettant d'assigner une valeur à un espace non connu à partir d'un semis de points connus.

(Génin et Aknine 2010b) proposent des stratégies d'acceptation et de proposition pour la formation de coalitions où les agents sont égoïstes et ont un ensemble de tâches à réaliser. Ils ont développé des stratégies adaptées à un domaine orienté tâches dans lequel les agents forment des coalitions dans le but de réaliser conjointement des lots de tâches. À cet effet, ils ont proposé une stratégie de construction des coalitions permettant de sélectionner un lot de tâches à réaliser ainsi qu'un groupe d'agents à qui proposer la réalisation de ce lot et une stratégie d'acceptation leur permettant de

décider s'ils acceptent ou non une proposition de formation d'une coalition envoyée par un agent du système.

(Zlotkin et Rosenschein 1994) proposent la formation de coalitions pour la réalisation d'une liste ordonnée de tâches sans prendre en compte les ressources des agents. Le but d'un agent étant de réaliser un ensemble de tâches parmi une liste ordonnée de tâches. Ils ont défini une fonction pour calculer le coût pour la réalisation des tâches par les agents et le coût encouru par un agent s'il exécute seul sa tâche. Ils considèrent qu'un sous-groupe d'agents peut passer des accords sans que cela ne concerne le groupe en entier. Il mesure la rationalité individuelle et la dominance d'un groupe au sens de Pareto pour évaluer la stabilité des coalitions.

(Génin et Aknine 2010a) ont abordé le problème de la formation de coalitions pour des agents égoïstes dans un contexte de jeux hédoniques. Ils se sont focalisés sur le niveau de satisfaction de chaque agent qui dépend des membres de sa coalition. Ils ont proposé deux types de stratégies, à savoir, (1) les stratégies d'acceptation de propositions qui permettent aux agents d'accepter ou de rejeter une demande de formation de coalitions et (2) les stratégies de sélection de propositions dont l'efficacité dépend de l'historique des négociations qui permet à un agent de sélectionner une coalition pertinente.

(Génin et Aknine 2008) suggèrent un protocole qui permet aux agents d'atteindre des objectifs communs et des stratégies individuelles pour construire des propositions efficaces à partir des informations collectées sur les autres agents. Les agents font face à deux niveaux de prise de décision pendant la formation des coalitions, c'est-à-dire, face à l'utilité seuil d'acceptation et l'utilité seuil de confirmation. L'utilité seuil d'acceptation est l'utilité minimale pour qu'une proposition soit acceptée et l'utilité seuil de confirmation est l'utilité minimale pour qu'une proposition soit confirmée.

(Aknine et al. 2004b) considèrent les préférences individuelles des agents et ne cherchent pas à améliorer une utilité globale du système. De ce fait, chaque agent crée sa matrice de préférences sur les autres agents par critère. Ils proposent deux modèles qui se fondent sur - l'intégrale de Choquet comme pour faire l'agrégation des préférences des agents et - la méthode ESD (Evolutionary System Design) de (Melvin 1996) pour la formation de coalitions. Le premier modèle est fondé sur l'ESD pour des agents compétitifs. Selon cette approche, s'il est impossible de former des coalitions avec les modèles de préférences des agents à cause de la divergence des préférences sur la base d'un seul critère, les agents annulent et reconstruisent leurs modèles de préférence en introduisant progressivement de nouveaux critères de négociation. Ils considèrent des poids sur les critères afin de définir un ordre de priorité sur ces derniers. Malgré une renégociation, on peut néanmoins ne pas obtenir de coalitions. Le second modèle est pour des agents coopératifs où les agents peuvent échanger leurs informations sur leurs préférences. Dans ce cas, ils garantissent la formation d'une coalition dans tous les cas. Ce modèle est fondé sur le principe de circulation de la procédure de formation de la coalition entre les différents agents. Cette méthode se fonde sur un indice d'attraction unilatérale qui évalue la préférence de tous les autres agents pour chaque agent. Le processus

de formation de coalitions passe de l'agent le plus préféré à ceux qui sont moins préférés tant qu'il y a des agents qui ne sont pas encore au sein d'une coalition.

Les modèles fondés sur la complémentarité cherchent à diviser l'ensemble des agents en évitant les préférences incompatibilités (Luce et Raiffa 1957)(Axelrod 1984) ou sur l'utilisation de manière collaborative des capacités individuelles qui se complètent pour améliorer les capacités de chaque agent à atteindre ses objectifs. Plusieurs de ces méthodes visent à satisfaire les propriétés du *Core-stable* (Sandholm 1999)(Chalkiadakis et al. 2012).

Pour former des coalitions stables dans (Sandholm 1999), les auteurs recherchent le bien-être collectif qui maximise le graphe des structures de coalitions (le *Core*). Seules les structures de coalitions qui maximisent le bien-être social sont considérées comme stables. L'objectif de (Chalkiadakis et al. 2012) est d'étudier la stabilité des structures de coalitions (la stabilité du *Core*) pour les jeux coopératifs dans des environnements structurés. Les auteurs représentent les fonctions caractéristiques des jeux modélisés par un graphe dans le but de déterminer les coalitions possibles. Dans cette méthode, une coalition est un ensemble connexe dans le graphe représentant l'environnement des agents. Pour déterminer les structures de coalitions possibles, ils supposent connaître les besoins, ou au moins les positions des agents. Toutefois, il est prouvé dans (Sandholm et al. 1998) que la recherche d'une structure de coalitions stables est un problème difficile voire impossible pour des composants ayant des ressources limitées. Une autre restriction de ces travaux est qu'ils ne proposent pas de méthodes de formation de coalitions dynamiques car ils obligent la restructuration des coalitions après l'indisponibilité d'un ou de plusieurs agents pour garantir la continuité de l'exécution des tâches. Cependant, la restructuration des coalitions n'est pas toujours optimale ou possible dans un contexte d'environnement dynamique.

(Aknine et al. 2004a) proposent une méthode de formation de coalitions en se basant sur des concessions graduelles entre les agents, pour trouver une solution Pareto optimale. Cela, dans un contexte de filtrage des documents racistes (analyse de contenu). Ils assimilent la combinaison des critères linguistiques à un processus de coopération entre des agents. Utilisation d'agents-critères, qu'ils associent à chacun des critères linguistiques. Le rôle d'un agent-critère est de générer un ensemble d'informations sur les caractéristiques de chaque document en utilisant un ensemble de traitements propres et en se fondant sur les informations produites par les autres agents-critères. Ils suggèrent un protocole qui permet une restructuration dynamique des coalitions en fonction des nouvelles informations qu'ils génèrent, sans agrégations des préférences. Les auteurs proposent une amélioration dans (Aknine et Caillou 2004) où il n'y a plus d'ordre imposé lors du processus de formation de coalition et où il n'y a plus de priorité entre agents. De plus, les agents ne sont pas obligés de se faire confiance.

(Shehory 2004) propose un système où les agents forment des coalitions pour acheter des biens, avec un certain prix qui dépend de la taille de la coalition. Les agents évoluent de manière aléatoire sur des sites qui proposent des marchandises. Chaque agent peut décider de se joindre aux agents

présents si la formation du groupe lui apporte un bénéfice et, comme il peut également décider de quitter la coalition formée. L'auteur modélise le problème sous la forme d'équations différentielles du premier ordre et étudie la stabilité d'un tel système.

(Yamamoto et Sycara 2001) proposent des mécanismes d'enchères pour la formation de coalitions d'acheteurs dans un contexte d'E-Market. Les acheteurs déclarent à un manager le prix maximal qu'ils sont prêts à payer pour obtenir les biens en se fondant sur des critères (items) et les vendeurs déclarent leur prix. L'agent chef de groupe divise le groupe en sous-groupes (coalitions). Il sélectionne un vendeur gagnant pour chaque coalition et divise l'excédent du gain entre les acheteurs. Cela, afin d'augmenter l'utilité des acheteurs, de garantir une stabilité dans la division de l'excédent au sein de chaque coalition et une stabilité au sens du Core. (Li et Sycara 2002) restent dans le même contexte E-Market. Le modèle suppose que chaque acheteur peut émettre une offre sur une combinaison d'items avec un coût de réservation et les vendeurs offrent des remises de prix pour chaque item en fonction des volumes. En divisant artificiellement le coût de réservation de chaque acheteur sur les items, ils construisent des coalitions optimales à l'égard de chaque item. Ces coalitions garantissent la complémentarité des items en matière de transfert de coûts de réservation et donc induisent la solution optimale. Les auteurs ne considèrent que les systèmes dotés de fonctions de prix linéaires, ce qui leur permet de présenter un algorithme polynomial pour trouver une solution semi-optimale et un système de division de gain pour une stabilité des coalitions au sens du Core.

Les algorithmes de formation de coalitions stables qui s'appuient sur le concept de la stabilité de la valeur de Shapley et ses extensions comme la valeur de Shapley bilatérale (Ketchpel 1994) appliquée aux jeux coopératifs de n -agents arbitraires, sont proposées dans (Klusch et Shehory 1996)(Contreras et al. 1997).

Un algorithme de formation de coalitions qui étudie la stabilité en déterminant le Kernel du jeu est proposé dans (Blankenburg et Klusch 2004). Une structure de coalitions S qui est le Kernel d'un jeu coopératif est l'ensemble des K configurations stables (S, u) dans lesquelles toutes les coalitions dans S sont en équilibre en présence de la fonction d'utilité u . Une coalition C est dans un tel équilibre si chaque paire d'agents dans C est en équilibre, c'est-à-dire, pour chaque paire (i, j) formée par les agents i et j , l'agent i ne peut pas obtenir un gain supérieur à celui qu'il avait en étant avec l'agent j dans (S, u) juste en menaçant de quitter sa coalition C actuelle pour obtenir une meilleure récompense dans une ou des coalitions qui ne sont pas dans S et excluant j .

Pour tenir compte des incertitudes qui peuvent produire des connaissances vagues ou incomplètes sur les profits attendus par les agents ou sur le partage de la récompense de leurs coalitions, certaines méthodes de formation de coalitions proposent l'utilisation du principe de coalitions floues (*fuzzy*) (Aubin 1981) (Mares 2001) (Chalkiadakis et al. 2011). Un jeu coopératif flou consiste en un ensemble d'agents, une fonction caractéristique floue v et une fonction d'adhésion m des quantités floues $v(C)$. $v(C)$ est interprété comme la vague espérance de profits de la coalition C qui doit être

distribuée parmi ses membres.

Une autre classe de jeux coopératifs résulte du problème de prise de décisions coopératives d'environnements stochastiques. La notion de jeux coopératifs stochastiques ou de jeux coopératifs avec des récompenses stochastiques, est présentée et étudiée dans (Suijs et al. 1999). Un jeu avec des récompenses stochastiques est défini par, un ensemble d'agents, un ensemble d'actions possibles et une fonction stochastique. Cette fonction assigne à chaque action d'une coalition, une variable stochastique à valeur réelle, qui représente la récompense attendue par la coalition quand une action particulière est prise. Par contre avec un jeu coopératif déterministe, les récompenses peuvent être des variables aléatoires mais les actions d'une coalition doivent être explicitement exprimées même si la fonction de distribution des récompenses n'est pas unique.

Notez que, les méthodes de formation de coalitions floues ou stochastiques même si elles considèrent des aspects dynamiques des agents lors du calcul des récompenses, ces méthodes ne permettent pas des inférences dynamiques et une stabilisation dynamique des coalitions.

Le concept de coalitions durables est proposé dans (Barberà et Gerber 2003) dans un contexte purement hédonique, c'est-à-dire, dans une coalition, il n'y a aucun problème distribué à résoudre. Les coalitions sont choisies sur cette base de connaissances a priori de l'aversion au risque maximal des agents et de l'ensemble des préférences des agents. Dans un contexte de planification de tâches par négociation où chaque agent est doté d'un système qui génère des plans pour chacun de ses objectifs recherchés, (Arib et Aknine 2011) considèrent que les agents évoluent dans un système totalement observable, c'est-à-dire que les agents ont une vue totale sur les plans des autres agents. Ils proposent un mécanisme de formation de coalitions fondé sur l'importance que peut avoir la position des actions dans les plans des agents. Leur mécanisme utilise les relations entre les actions pour guider la recherche des coalitions par les agents et l'importance de chacune d'elles dans leurs plans et facilite l'obtention d'accords dans le sens où, les agents trouvent un intérêt à infléchir leurs positions au regard des propositions formulées par les autres agents.

(Arib et al. 2014) considèrent que les agents évoluent dans un système où chacun a une connaissance partielle (ou inexistante) des plans des autres agents. Ils proposent un mécanisme qui ne tient pas compte de la disposition des actions dans les plans des agents. Dans ce cas, chaque agent calcule un degré de croyance pour chaque action qu'il ne peut effectuer seul par rapport à son plan et à la confiance qu'il a vis-à-vis d'autres agents avec lesquels il trouve nécessaire de coopérer. Cela, leur a permis de déterminer les contraintes à imposer aux agents pour former des coalitions stables.

(Arib et Aknine 2013) se sont intéressés au problème de la formation de coalitions avec des contraintes dynamiques et des préférences que les agents expriment explicitement et qu'ils acceptent de communiquer aux autres agents lors de leurs négociations. Ils ont proposé un langage de représentation des propositions des agents fondé sur leurs contraintes respectives et qui facilite leur négociation. Dans ce même contexte, ils ont proposé un mécanisme qui s'appuie sur la résolu-

tion des systèmes de contraintes pour orienter la négociation entre les agents tout en gérant leurs préférences.

1.2.2 Formation de coalitions stables avec inférence dynamique

Les agents doivent pouvoir dynamiquement construire des coalitions stables en tenant compte de leur propre comportement et de la dynamique de l'environnement et des tâches. Une des approches proposées est le principe de chevauchement des coalitions (*overlapping coalitions*) (Shehory et Kraus 1996) où les phases de négociation et de formation de coalitions peuvent être adaptées au contexte des agents et où les agents peuvent distribuer leurs ressources entre plusieurs tâches. Chaque tâche génère un profit qui peut être librement divisé entre les agents qui participent à l'exécution de la tâche (Chalkiadakis et al. 2010) (Zick et al. 2012). Leur algorithme distribué calcule le partitionnement approprié de l'ensemble des agents.

Une autre solution est l'utilisation de mécanismes de raisonnements sociaux basés sur la notion de réseau de dépendances sociales (Castelfranchi et al. 1992) (Sichman 1998), ou sur la gestion des réputations et des croyances. Pour acquérir et utiliser la connaissance des dépendances sur l'ensemble des agents considérés, chaque agent doit de manière dynamique représenter explicitement quelques préférences des autres agents (par exemple, leurs buts et leurs plans), exploiter cette connaissance afin d'optimiser son comportement selon l'évolution de la société (l'ensemble des agents). Pour un agent, le monitoring et la révision de la connaissance sur la société, lui permet d'éviter des incohérences sans préétablir un contrôle global (Klusch et Gerber 2002).

Par exemple dans (Milchtaich et Winter 2002), les modèles de formation de coalitions sont fondés sur la supposition que chaque agent préfère s'associer à des agents qui lui sont semblables. Ces modèles ont pour but de stabiliser la dynamique des comportements myopes individuels qui décrivent l'évolution de la formation des coalitions et qui déterminent une stabilité probable et une partition *Pareto* efficace.

Dans les mécanismes de gestion de réputations, le but est d'éviter l'interaction avec des participants indésirables. De tels mécanismes permettent aux agents de réagir efficacement face aux changements fréquents de réputation des partenaires potentiels en fonction des profits attendus et de la fiabilité de ces partenaires (Rubiera et al. 2001) (Yu et Singh 2000) (Manchala 2000). De même dans (Ye et al. 2012), les coalitions peuvent permettre aux agents d'adapter dynamiquement leur implication dans leur coalition et de quitter leur coalition courante pour d'autres coalitions sous le contrôle d'un superviseur centralisé. Cela entraîne la restructuration des coalitions.

Dans de nombreux cas, les contraintes de ressources et de temps nécessitent la conception d'un mécanisme de raisonnement sur le temps du mécanisme de formation de coalitions afin d'améliorer la stabilité des coalitions.

Dans (Jiang et al. 2006), les auteurs proposent un mécanisme incitatif pour la formation de coalitions dans les réseaux MANETs dans un contexte coopératif. Les auteurs modélisent le réseau comme un graphe dont les nœuds sont les agents et les arcs les liens de communication entre les agents. Ils définissent une coalition stable comme étant formée de tous les agents dont les liens peuvent être activés. Contrairement à notre approche, leur mécanisme nécessite la connaissance de tous les liens possibles et les positions des participants potentiels. En outre, il est indispensable de réévaluer le graphe à chaque fois qu'un agent change de voisinage ou qu'il devient indisponible. (Šišlák et al. 2005) traitent l'inaccessibilité dans les réseaux MANETs. Ils proposent de maintenir et de construire une distribution optimale d'agents spécialisés (*Mobile Middle Agents (MMAs)*) et dont le rôle est de connecter les agents. Toutefois, cette solution ne peut être efficace que si ces agents spécialisés sont disponibles, car aucune solution n'est proposée pour gérer leur indisponibilité.

(Klusch et Gerber 2002) proposent une méthode de formation de coalitions dynamiques (DCF-A) pour permettre aux agents cognitifs de réagir face aux événements imprévus. Dans le système DCF-A, chaque coalition construite est représentée par un agent appelé *leader* qui agit comme représentant de la coalition. Le *leader* simule des événements hypothétiques et ajuste la configuration de la coalition au fur et à mesure des simulations en fonction du risque encouru par la coalition en ajoutant ou en supprimant un agent de la coalition. Lorsque le *leader* de la coalition parvient à une amélioration significative de la valeur de la coalition par ses simulations, il en informe tous les membres de cette coalition sur les alternatives possibles. Cette approche est inadaptée au contexte dynamique en raison de la centralisation des décisions autour du *leader*. De plus, aucun mécanisme pour gérer l'indisponibilité des *leaders* n'est proposé.

Dans (Khan et al. 2011) les auteurs considèrent la dynamique des tâches mais avec des agents homogènes et coopératifs. Ils proposent une méthode de formation de coalitions basée sur les MDPs (Markov le Processus de Décision) mais sans tenir compte de la disponibilité aléatoire des agents lors de l'exécution des tâches et sans chercher à garantir la stabilité des coalitions formées. De plus, ils utilisent des connaissances a priori sur les préférences des agents et sur la dynamique des tâches pour guider le choix des coalitions à former.

(Caillou et al. 2002) proposent un protocole de formation de coalitions optimales au sens de Pareto dans un contexte d'agents égoïstes quand il est indésirable ou impossible d'agréger les préférences des agents. De plus, ils suggèrent une méthode de restructuration de la coalition si un changement intervient dans le système. Ils proposent d'appliquer leur méthode à la génération d'emplois du temps où une coalition est composée d'un professeur, d'un groupe d'élèves avec des paramètres tels que les horaires du cours, la disponibilité des salles, etc. Les agents sont préalablement ordonnés et s'envoient des coalitions suivant leurs préférences. Ils peuvent alors supprimer les coalitions qui ne sont pas satisfaisantes pour eux.

(Chalkiadakis et Boutilier 2004) proposent d'utiliser un processus décisionnel de Markov partiellement observable (POMDP) afin d'effectuer un apprentissage par renforcement permettant à un

agent d'estimer la valeur des capacités des autres agents du système et donc d'estimer par la suite la valeur des coalitions. La valeur d'une coalition est calculée à partir des actions qu'elle est capable de réaliser et des capacités individuelles de ses agents. Ces derniers ne sont pas capables de connaître la valeur exacte d'une coalition puisqu'ils ne connaissent pas les capacités des autres agents. Les auteurs s'intéressent aux situations où les agents sont dotés de capacités, mais où un agent ne connaît pas les capacités des autres agents du système. Une coalition d'agents est capable de réaliser des actions et les actions réalisées apportent un paiement dépendant des capacités des agents de la coalition. Les agents changent de coalitions suivant leurs valeurs estimées et forment une nouvelle structure de coalitions. Ils mettent alors leurs estimations à jour à partir des nouvelles coalitions formées. Ils réitèrent ensuite le processus de formation de coalitions, suivant l'existence de meilleures coalitions, en fonction de leurs nouvelles estimations. Le processus se réitère jusqu'à la convergence vers un état stable, c'est-à-dire, le Core bayésien.

(Bonnet et Tessier 2008) proposent une application pour la formation de coalitions dans le cadre d'une constellation de satellites ayant pour mission la prise de photographies de la Terre. Une tâche est définie comme une requête d'observations. Les auteurs proposent un protocole de communication ad-hoc permettant de mettre à jour les connaissances des satellites sur le système (nouvelles tâches à accomplir, coalitions formées, etc.). Aucun satellite n'étant capable de réaliser seul toutes les tâches, ils forment ainsi des coalitions afin de mettre en commun leurs compétences.

(Horling et al. 2001) proposent une application utilisant les réseaux de capteurs pour la surveillance de cibles. Lorsque l'information est complète. Dans le même cadre, (Dang et al. 2006) proposent deux algorithmes de formation de coalitions (un algorithme approché et polynomial et un algorithme optimal) permettant de résoudre ce problème. Une tâche correspond à la surveillance d'une cible et une coalition est un groupe d'agents dédié à la surveillance d'une cible. Les auteurs y définissent la valeur d'une coalition et la valeur d'une structure de coalitions, que les algorithmes doivent maximiser.

En utilisant la théorie des graphes, (Sless et al. 2014) cherchent à trouver l'ensemble des structures de coalitions Core-stable qui maximisent le bien-être social des agents dans le cas où la formation des coalitions et la détermination des valeurs de ces dernières se fondent sur la force des relations qui existent entre les agents déployés dans un réseau social. Ils proposent les concepts de k -coalitions-core et de groupe stable. Ils considèrent qu'une structure de coalitions CS est k -coalitions-core, si CS n'admet aucune coalition bloquante et si CS est individuellement rationnelle. Si CS est groupe stable, alors CS n'admet aucune coalition bloquante B telle que $B \subset C_i \cup C_j$ où $C_i, C_j \in CS$ et $i \neq j$. Cette notion de groupe stable est une généralisation de la notion de stabilité interne proposée par (Brânzei et Larson 2009). Cependant, ils supposent qu'un organisateur central a la capacité de construire les structures de coalitions en utilisant une fonction de mappage qui permet d'affecter chaque agent à une coalition afin de réaliser un ensemble de tâches. En outre, il est possible pour cet organisateur central de créer de nouvelles relations entre les agents pour adapter le jeu sans

qu'il soit nécessaire que les agents d'une coalition forment un ensemble connexe dans le réseau social. Ces suppositions sont des hypothèses fortes pour notre contexte dans la mesure où, nous devons avoir un mécanisme décentralisé qui doit tenir compte des incertitudes sur les tâches et sur les préférences des agents où chaque agent d'une coalition C doit avoir au moins un agent voisin qui est aussi dans C .

(Kranich et al. 2005) cherchent à définir le concept de Core dans un contexte de jeu à utilité transférable (TU-game) lorsque les coopérations des joueurs se déroulent dans un cadre dynamique dans lequel les joueurs sont confrontés à une séquence finie d'événements exogènes. Une solution pour leur modèle dynamique spécifie un ensemble de paiements pour chaque joueur, car, l'évolution du jeu dans le temps est divisée par étapes dans lesquelles chaque joueur reçoit une récompense pour chaque étape. En se fondant sur les interprétations des écarts rentables pour les coalitions, ce travail propose trois concepts de Core différents, c'est-à-dire, le Core classique, le fort Core séquentiel et le faible Core séquentiel. Toutefois, dans une étape, ils supposent qu'il n'y a pas de changements du nombre de joueurs. En outre, ce travail relâche les contraintes de ressources instantanées, il fait abstraction des influences des événements endogènes, considère comme connue la séquence des étapes du jeu et qu'une solution de chaque étape dépend de l'étape qui le précède comme dans (Filar et Petrosjan 2000).

Contrairement aux jeux classiques dans lesquels le même jeu est répété plusieurs fois au fil du temps, (Avrachenkov et al. 2013), ont trouvé une procédure de distribution de récompenses pour un MDP coopératif (MDP-CPDP) qui distribue au fil du temps, le gain que les joueurs gagnent dans le jeu à long terme (c'est-à-dire, lorsque l'interaction entre les joueurs est réitérée au fil du temps) dans l'hypothèse d'utilité transférable. Dans ce cas, plusieurs jeux statiques différents se succèdent. Les jeux statiques suivent une chaîne de Markov à temps discret, dont les probabilités de transition dépendent des actions des joueurs dans chaque état de la chaîne de Markov. Les joueurs peuvent décider de rejoindre la grande coalition ou de former des coalitions plus petites. En affinant le concept du Core pour un MDP coopératif, ils ont prouvé sous quelle condition le MDP-CPDP vérifie la propriété de cohérence temporelle et de renforcement de la cohésion de la coalition au fil du temps. Contrairement à notre travail, l'évolution du jeu dans le temps est divisée en étapes déterministes dans lesquelles chaque joueur reçoit un paiement en fonction de ses actions antérieures, les contraintes d'une coalition n'évoluent pas dans une étape et une fois qu'un groupe de joueurs se retire de la grande coalition, il ne peut plus le rejoindre.

CONCLUSION DU CHAPITRE

Les coalitions sont largement utilisées dans les systèmes multi-agents pour effectuer des tâches collectives. Les méthodes de formation de coalitions peuvent aborder la question de la stabilité

des coalitions ou non, cependant, elles n'abordent pas la stabilité dans des contextes incertains dynamiques du type mentionné dans nos travaux. Généralement, les mécanismes de formation de coalitions considèrent la stabilité comme une propriété des coalitions qui indique que les agents n'ont aucune intention de quitter leurs coalitions avant l'achèvement leurs tâches. Ces travaux ne prennent pas en compte le contexte dynamique et stochastique dans lequel les agents peuvent devenir indisponibles quelles que soient leurs intentions au moment de la formation de leur coalition. En revanche, dans le cadre de notre travail, nous considérons un autre aspect de la stabilité, à savoir, la nécessité d'assurer la stabilité des coalitions en évitant la renégociation fréquente malgré la dynamique et les incertitudes des tâches et des agents.

2

FORMATION DE COALITIONS NASH-STABLES ET PERSISTANTES

Dans ce chapitre, nous présentons un nouveau mécanisme de formation de coalitions stables dans un contexte de tâches à évolution non déterministe et d'agents autonomes égoïstes avec des disponibilités imprévisibles. Ces agents ne peuvent en aucun cas profiter de connaissances a priori ou des avantages d'une quelconque centralisation des calculs. Notre mécanisme de formation de coalitions détaillé dans ce chapitre, est fondé sur - un principe d'alliances et de recommandations pour faciliter la formation de coalitions stables, - les lois de probabilité et les MDPs (Chaîne de Markov) pour suivre et prédire la stabilité des coalitions. Cette prédiction de la stabilité des coalitions nous permet d'introduire la notion de coalitions persistantes (des coalitions qui restent stables dans le temps malgré la dynamique et les incertitudes sur les agents, sur les composants et sur leurs tâches). Une évaluation de performances théorique et expérimentale de notre mécanisme est également proposée pour dégager ses propriétés intrinsèques.

2.1 OBJECTIFS ET APPROCHE GÉNÉRALE

Les travaux détaillés dans ce chapitre ont fait l'objet des 4 publications suivantes (Faye et al. 2014a), (Faye et al. 2014b), (Faye et al. 2014c) et (Faye et al. 2014d).

Pour faire coopérer des agents hétérogènes et compétitifs dans un système multi-agent, on a le plus souvent recours aux méthodes de formation de coalitions. Les méthodes actuelles considèrent que : (1) l'ensemble des coalitions stables qui maximisent l'utilité des agents peut être calculé a priori de manière déterministe, (2) les coalitions sont stables durant l'exécution des tâches, (3) les tâches ne sont pas censées évoluer une fois les coalitions formées. Cependant, ces considérations sont des hypothèses fortes pour un certain nombre de contextes réels, par exemple, dans le cas d'un sinistre où des intervenants qui disposent de composants électroniques communicants (Smartphones, PDA, etc.), doivent s'organiser en groupes de manière décentralisée afin de réaliser des tâches complexes (extinction d'incendies, évacuation de survivants, ...). Il est clair que dans ce contexte, si chaque agent est déployé dans seulement un de ces composants électroniques, alors, un agent ne peut pas connaître a priori les préférences et la disponibilité des autres agents et ne peut communiquer qu'avec son voisinage direct. À ces contraintes s'ajoute la dynamique des tâches qui peut altérer l'efficacité des actions relatives aux interventions urgentes et l'intérêt des intervenants (par exemple, éviter les risques et maximiser leurs gains en temps ou monétaires). Certaines tâches exigent alors la collaboration entre les humains et les robots que les agents doivent faciliter au moyen de la formation de coalitions.

Cependant, les processus de formation de coalitions exigent des calculs et des communications importants qui peuvent être prohibitifs dans des scénarios où les ressources en calculs et communication sont limitées et où les délais d'exécution des tâches sont un critère important. Ainsi, notre mécanisme décentralisé se fonde sur les négociations multilatérales pour permettre la collaboration entre les différents intervenants et gère les communications, les calculs et les ressources énergétiques associées.

La contribution de ce chapitre est une nouvelle méthode de formation de coalitions stables qui tient compte de ces contraintes de ressources et de l'absence d'informations sur les préférences des agents, sur leur disponibilité et sur la dynamique des tâches. Plus en détails, nous proposons un mécanisme de construction progressive et dynamique de connaissances et d'affinités entre des agents autonomes et égoïstes afin de surmonter les informations absentes et de faciliter la formation de coalitions auto-stables (persistantes). Par coalitions auto-stables nous comprenons, une adaptation dynamique en un temps fini de la configuration des coalitions qui leur permet de retrouver un état stable sans interventions extérieures.

Dans un premier temps, nous définissons un mécanisme de formation de coalitions adaptatif et dynamique qui tient compte du temps et des contraintes de ressources des tâches. Le mécanisme a

pour but d'augmenter les performances collaboratives en matière d'exécution des tâches des agents autonomes et compétitifs. Pour ce faire, le mécanisme fournit aux agents la capacité de construire progressivement leurs connaissances, d'interagir de manière distribuée dans un contexte où les tâches changent dynamiquement (par exemple, l'extinction de feu, la recherche de survivants, etc.) et où il est difficile de prévoir de façon déterministe la stabilité des interactions engagées en raison des incertitudes sur les tâches et sur les agents. Ce mécanisme de formation de coalitions est fondé sur : - notre mécanisme de *recommandation* pour la construction progressive de connaissances fiables, - des lois de probabilité pour la prédiction dynamique des évolutions des agents et des tâches, et, - le Processus de Décision Markov (MDP) pour la prédiction dynamique de la stabilité des coalitions. Malgré l'absence de connaissances a priori sur les tâches et sur les autres agents, notre mécanisme de *recommandation* aide à surmonter les contraintes et les incertitudes des agents et des tâches, de faciliter l'acquisition de ressources et le partage des connaissances pour une construction dynamique des préférences d'un agent et des connaissances sur la fiabilité des agents. Les lois de probabilité et le MDP permettent de prendre en compte les événements stochastiques (disponibilité imprévisible des agents, tâches dynamiques) pour aider à la prise de décision des agents.

Dans un deuxième temps, nous proposons un mécanisme qui facilite l'auto-stabilisation dans le temps des coalitions en fonction des contraintes de ressources, d'évolution des tâches, de la dynamique des agents (disponibilité aléatoire) et de l'impossibilité de prédire les coalitions qui peuvent se stabiliser au fil du temps avant le début de l'exécution des tâches. Pour ce faire, nous proposons un mécanisme de recherche des coalitions *persistantes* par exploration localisée et par exploration étendue du voisinage des agents. Une coalition est *persistante* si elle est stable et si elle a la capacité de continuellement se stabiliser au fil du temps en dépit des incertitudes sur la disponibilité des agents, sur les préférences des agents et sur la dynamique des tâches. Notre mécanisme de formation de coalitions *persistantes* se fonde sur un mécanisme de *recommandation* pour permettre l'auto-stabilisation dynamique des coalitions. En raison des événements stochastiques qui peuvent dynamiquement modifier la stabilité des coalitions, nous utilisons des lois de probabilité, le MDP en évaluant le *risque de renégocier les coalitions* pour permettre la prise de décision et pour prévoir dynamiquement les événements critiques (par exemple, l'indisponibilité des agents, l'amplification de la tâche ...), qui peuvent mener à des coalitions instables.

Dans ce travail, par coalition stable, nous entendons une stabilité au sens de Nash (Nash 1950).

Pour évaluer ce travail, nous prenons comme exemple applicatif une situation où des agents doivent aider à la prise de décision et à l'établissement d'un réseau de communication ad-hoc pour un ensemble d'intervenants et de robots qui doivent intervenir dans une zone sinistrée. Les agents sont déployés dans les robots, les Smart-phones et les autres équipements communicants des intervenants, par exemple, les pompiers chargés de secourir la population (chaque robot ou équipement communicant requiert un seul agent). L'objectif recherché est que les agents puissent aider à la formation de groupes d'intervention stables (coalitions stables), à trouver les meilleures réparti-

tions des secours en évitant les zones à risques pour les intervenants (par exemple, explosions, éboulements). Ils doivent par ailleurs tenir compte des préférences, des fiabilités et contraintes des intervenants sans oublier les contraintes associées à l'exécution des tâches. Nous considérons que les agents utilisent les ressources des composants sur lesquels ils sont déployés afin de faire un monitoring de leur environnement, d'effectuer les calculs nécessaires à leur raisonnement et de trouver des agents voisins avec qui communiquer, en utilisant par exemple, la technologie Bluetooth. Nous supposons que les agents peuvent obtenir des données (par exemple, positions, niveaux d'urgences) sur les tâches en utilisant des capteurs. Nous considérons, par ailleurs, qu'un robot ou un intervenant n'a pas toutes les connaissances sur la distribution des ressources et sur la disponibilité des autres robots ou intervenants.

2.2 CONCEPTS DU MODÈLE

Nous considérons qu'une tâche est assignée automatiquement à un agent (*l'agent d'initiateur*) qui doit gérer l'exécution collective de la tâche, puisque les tâches ne peuvent être exécutées isolément par un seul agent. Cette exécution se fait de manière décentralisée. Par ailleurs, l'indisponibilité de l'agent initiateur durant le processus d'exécution de la tâche ne doit pas interrompre le processus. Nous définissons une tâche comme un ensemble d'actions à exécuter (par exemple, éteindre un feu), tandis qu'un but est un objectif qu'un agent veut atteindre (par exemple, la récompense accordée à l'intervenant suite à l'accomplissement de ses actions). Ainsi, les agents se servent de l'exécution des tâches comme un moyen pour atteindre leurs buts. Soit $A = \{a_1, \dots, a_n\}$ un ensemble d'agents autonomes et compétitifs où chaque agent $a_i \in A : i \in [1, |A|]$ a un ensemble de propriétés qui lui sont propres.

2.2.1 Propriétés d'un agent

Les contraintes d'un agent a_i appartenant à un ensemble d'agents $A = \{a_1, a_2, \dots, a_n\}$, sont données par les paramètres $\{R_{a_i}, Aut_{a_i}, Hs_{a_i}, \vartheta_{a_i}, B_{a_i}\}$ où R_{a_i} est l'ensemble de ses ressources et Aut_{a_i} son autonomie énergétique. Hs_{a_i} son *historique* des interactions qui contient l'ensemble des *alliances* ainsi que l'ensemble des fiabilités et l'ensemble des probabilités de stabilité des agents avec qui il a établie une *alliance*. ϑ_{a_i} est sa *vue* et B_{a_i} son but.

Nous faisons la distinction entre Aut_{a_i} et R_{a_i} car nous considérons que les agents ne partagent pas leur énergie et qu'une coalition a besoin que de ressources (par exemple, eau, extincteur, etc.) susceptibles d'exécuter une tâche (par exemple, éteindre un feu).

Pour atteindre ses buts, a_i doit déterminer dynamiquement les coalitions à former avec les agents

de son voisinage que nous appelons ici sa *vue*. Par exemple, le voisinage d'un agent va dépendre de la force du signal du dispositif qui héberge l'agent. Ainsi, la vue d'un agent peut contenir 0, 1, ... n agents voisins.

Définition 2.1 La vue ϑ_{a_i} d'un agent a_i référence l'ensemble des agents de son voisinage avec qui a_i peut directement communiquer.

Exemple 2.1 Si $\vartheta_{a_i} = (\{a_j, x_{i,j}\}, \{a_k, x_{i,k}\})$, alors les agents a_j et a_k sont des voisins directs de l'agent a_i . Les paramètres $x_{i,j}$ et $x_{i,k}$ sont des paramètres booléens qui spécifient respectivement s'il existe ou non une alliance entre a_i et ses voisins a_j et a_k . Si $x_{i,j} = \text{Vrai}$ cela signifie qu'il existe une alliance entre a_i et a_j sinon, $x_{i,j} = \text{Faux}$ signifie qu'il n'existe pas d'alliance entre a_i et a_j .

Le concept d'alliances se définit comme suit :

Définition 2.2 Une alliance Al_{a_i, a_j} est un pacte pouvant persister entre deux agents a_i, a_j , dans lequel ces deux agents établissent un accord de partage de ressources et d'informations en cas de besoin et suivant les termes de cette alliance.

Par exemple, une alliance entre a_i et a_j se traduit par la relation $Al_{a_i, a_j} = (R(Al_{a_i, a_j}), T_{help}(Al_{a_i, a_j}))$. Plus en détails, $R(Al_{a_i, a_j}) = \{R_{a_i}, R_{a_j}\}$ est l'ensemble des ressources de l'alliance et $T_{help}(Al_{a_i, a_j}) = \{T_{help}^{a_i}, T_{help}^{a_j}\}$ est l'ensemble des temps de sollicitation de l'alliance. Ainsi, Cette relation signifie que l'agent a_i met à la disposition de l'agent a_j sa ou ses ressources R_{a_i} durant un temps $T_{help}^{a_i}$ s'il est sollicité par a_j et que l'agent a_j fait de même à l'égard de l'agent a_i avec sa ou ses ressources R_{a_j} durant un temps $T_{help}^{a_j}$.

Notons que, aucune obligation de respect de cette alliance n'est imposée ni à l'un ni à l'autre. Maintenir des alliances permet aux agents de construire dynamiquement leurs connaissances sur les autres agents et de faciliter l'acquisition d'informations et de ressources.

Définition 2.3 L'ensemble des alliances valides, l'ensemble des fiabilités et l'ensemble des probabilités de stabilité qu'un agent détient comme informations sur d'autres agents constituent son historique. Pour un agent a_i son historique est défini sous la forme $Hs_{a_i} = (\cup_{j=1}^{|A'|} Al_{a_i, a_j}) \cup (\cup_{k=1}^{|A'|} \rho_{a_k}) \cup (\cup_{m=1}^{|A'|} P_s^{a_m}) : A' \subseteq A$. ρ_{a_k} est la fiabilité de l'agent allié a_k et $P_s^{a_m}$ est la probabilité de stabilité de l'agent allié a_m .

La probabilité de stabilité $P_s^{a_j}$ d'un agent a_j s'exprime de la manière suivante :

$$P_s^{a_j} = 1 - Q_s^{a_j} \quad (2.1)$$

où $Q_s^{a_j}$ est la probabilité d'indisponibilité de l'agent a_j . Pour déterminer $Q_s^{a_j}$, a_j utilise la loi géométrique modifiée (Yates et Goodman 2005), (Baynat 2000) qui est la loi de probabilité a priori quand on s'intéresse à l'arrivée d'une première indisponibilité.

$$Q_s^{a_j} = (q_s^{a_j})^k * (1 - q_s^{a_j}) \quad (2.2)$$

où $q_s^{a_j} = e^{-\tau_{a_j}} * \frac{(\tau_{a_j})^k}{k!}$. τ_{a_j} est le taux d'indisponibilité de l'agent a_j , k est le nombre de fois où l'agent a été indisponible. Ainsi pour a_i , si $P_s^{a_k} > P_s^{a_j}$ alors a_i va préférer en matière de stabilité, a_k au lieu de l'agent a_j .

Nous entendons par *alliance* valide une *alliance* maintenue dans le temps du fait que l'agent allié est considéré comme fiable. La fiabilité d'un allié a_j est calculée par un agent a_i en utilisant la loi de *Poisson* (Yates et Goodman 2005), (Baynat 2000) qui est la loi de probabilité, a priori, si nous nous intéressons au nombre d'événements aléatoires (nombre de fois (k) que l'*alliance* n'a pas été respectée) dans un intervalle de temps.

Ainsi, en présence de k événements aléatoires, la fiabilité de l'agent a_j est exprimée par : $\rho_{a_j} = e^{(-\lambda_{a_j})} * \left(\frac{\lambda_{a_j}^k}{k!}\right)$: λ_{a_j} le taux des événements aléatoires.

Comme les agents sont autonomes et égoïstes, alors l'agent a_j n'informe pas nécessairement l'agent a_i sur ses propres intentions. De ce fait, a_i suppose en calculant la première valeur de la fiabilité, que l'agent a_j respectera toujours cette *alliance* (probabilité d'avoir zéro non-respect de l'*alliance*).

Alors, dans l'expression de la fiabilité, a_i met la valeur de k à 0. Il obtient ainsi : $\rho_{a_j} = e^{(-\lambda_{a_j})} * \left(\frac{\lambda_{a_j}^0}{0!}\right)$, c'est-à-dire :

$$\rho_{a_j} = e^{(-\lambda_{a_j})} \quad (2.3)$$

où λ_{a_j} est le taux de désengagement de l'agent a_j en vers l'*alliance* Al_{a_i, a_j} avec a_i . Nous supposons que si a_i relève que le nombre de désengagements consécutifs NW_{a_j} de l'agent a_j vis-à-vis de leur *alliance* a atteint un certain seuil W_th , alors a_i supprime toutes les *alliances* qu'il a avec a_j ce qui signifie que pour a_i , l'agent allié a_j n'est plus fiable.

Ainsi : $\lambda_{a_j} = \frac{NW_{a_j}}{W_th}$.

Le seuil du nombre de désengagements autorisés W_th dépend du contexte et de l'application multi-agent. De plus, comme nous le montrons ci-dessous, un agent a_i n'a pas nécessairement besoin de mémoriser chaque valeur λ_{a_j} de son agent allié a_j .

Par exemple, si pour a_i , $\rho_{a_j} = 0.74$, alors, $e^{(-\lambda_{a_j})} = 0.74$

$$\Rightarrow \frac{NW_{a_j}}{W_th} = -\ln(0.74) \Rightarrow NW_{a_j} = -\ln(0.74) * W_th.$$

Si suite à cela, a_j ne respecte pas cette *alliance*, alors a_i met à jour ρ_{a_j} / $\rho_{a_j} = e^{-\left(\frac{(-\ln(0.74) * W_th) + 1}{W_th}\right)}$, sinon, si a_j accepte et coopère avec a_i alors la mise à jour de ρ_{a_j} donne : $\rho_{a_j} = e^{-\left(\frac{(-\ln(0.74) * W_th) - 1}{W_th}\right)}$.

Autrement dit, le calcul directe de la mise à jour de ρ_{a_j} en ρ'_{a_j} est donnée sous la forme suivante :

$$\rho'_{a_j} = \begin{cases} e^{\frac{-[(-\ln(\rho_{a_j}) * W_th) + 1]}{W_th}} & \text{si } NW_{a_j} = NW_{a_j} + 1 \\ e^{\frac{-[(-\ln(\rho_{a_j}) * W_th) - 1]}{W_th}} & \text{si } NW_{a_j} = NW_{a_j} - 1 \end{cases}$$

Ainsi, un agent a_j doit respecter ses *alliances* pour pouvoir améliorer ou conserver sa fiabilité vis-à-

vis des autres agents. Il en est de même s'il veut être accepté dans une coalition pour atteindre ses buts.

Notons qu'une *alliance* peut être *équitable*, *préférée* et *non-dominée*.

Définition 2.4 Une alliance Al_{a_i, a_j} est dite *équitable* si elle est réalisable, si elle n'est pas contraignante pour les deux alliés a_i et a_j , et si elle améliore la capacité de ces deux agents.

$\forall a_i$ et $a_j \in A$, si l'alliance Al_{a_i, a_j} est réalisable alors :

- R_{a_i} et $R_{a_j} \in R_{min}$.
- $R_{a_i} \neq \emptyset$ et $R_{a_j} \neq \emptyset$.

Si l'alliance Al_{a_i, a_j} n'est pas une contrainte alors :

- $T_{help}^{a_i} \leq D_c \leq Aut_{a_i}$ $T_{help}^{a_j} \leq D_c \leq Aut_{a_j}$ où D_c est le temps d'exécution stochastique de la tâche T_c (cf. équation 2.4).

- $R_{a_i} \cap R_{a_j} = \emptyset$ et $T_{help}^{a_i} \approx T_{help}^{a_j}$.

$\forall a_i, a_j$ et $a_k \in A$, si l'alliance Al_{a_i, a_j} améliore la capacité des agents alors :

- $\forall R_{a_j} \in R(Al_{a_i, a_j}) \subseteq \bigcup_{k=1}^{|A'|} R(Al_{a_i, a_k}) : A' \subseteq A$ alors $R_{a_i} \neq R_{a_j}$. Si $R(Al_{a_i, a_j})$ est l'ensemble des ressources concernées par l'alliance Al_{a_i, a_j}
- $R(Al_{a_i, a_j}) \cap R(Al_{a_i, a_k}) = \emptyset$.

Ainsi, comme une *alliance* de l'agent a_i dépend de R_{a_j} , de $T_{help}^{a_j}$ et de ρ_{a_j} de l'allié a_j , alors a_i peut obtenir une préférence entre ses alliés.

Définition 2.5 Pour un agent a_i , une alliance Al_{a_i, a_j} est *préférée* à Al_{a_i, a_k} si, a_i a plus confiance en la fiabilité de l'agent a_j qu'en celle de l'agent a_k , $\rho_{a_j} > \rho_{a_k}$, ou si a_j a une meilleure probabilité de stabilité $P_s^{a_j} > P_s^{a_k}$ (cf. équation 2.1). Cette relation de préférence est notée : $a_j \succ_{Al} a_k$.

Notons qu'il n'y a aucune connaissance a priori qui permet de savoir les futures *alliances préférées* d'un agent, car elles dépendent de leur fiabilité, des ressources et du temps de coopération en tant qu'allié des agents. Une *alliance* peut être *équitable* et *préférée* ou *équitable* mais non *préférée*. Pour faire la distinction entre ces deux cas, nous définissons le concept d'*alliance non-dominée*.

Définition 2.6 Une alliance est *non-dominée* si elle est *équitable* et *préférée*.

Il est clair qu'un agent va progressivement identifier les agents fiables avec lesquels coopérer. Il va donc chercher à établir des *alliances non-dominées* afin de maximiser sa capacité à former des coalitions stables pour atteindre ses buts malgré le contexte stochastique du problème.

Nous donnons plus loin les détails sur la manière dont les *alliances* sont formées.

Dans la section suivante, nous formalisons nos concepts de formation de coalitions qui permettent

aux agents de construire des coalitions stables du point de vue de l'équilibre de Nash (Nash 1950) et qui prennent en compte leur disponibilité imprévisible ainsi que la dynamique des tâches.

2.2.2 Concepts de formation de coalitions

Une coalition C est définie par $C = \{A_c, G_c, R_{min}, T_c, D_c\}$; où $A_c = \{a_1, a_2, \dots, a_k\}$ est un ensemble d'agents avec comme buts $G_c \subseteq \{B_{a_i} : i \in N, a_i \in A_c\}$. $R_{min} = \{R_1, R_2, \dots, R_m\}$ est un ensemble de ressources nécessaires à C , T_c est la tâche qui a induit la formation de C . D_c est le temps d'exécution stochastique de T_c (cf. équation 2.4).

Notre approche considère que toute tâche est dynamique. Une tâche dynamique est une tâche à évolution non prévisible qui peut s'amplifier, régresser ou être stationnaire.

Exemple 2.2 Soit une coalition C_1 formée par les agents a_1 , a_2 et a_3 qui ont comme buts d'éviter les dangers, d'établir de nouveaux partenariats et d'obtenir une rémunération sur la tâche T_{c_1} à effectuer.

Si l'ensemble des ressources nécessaires pour la réalisation de la tâche sont, par exemple, de l'eau et des extincteurs durant un temps approximatif de 2 heures alors cette coalition s'exprime formellement comme suit :

$$C_1 = \{\{a_1, a_2, a_3\}, \{\text{éviter dangers, partenariats, argent}\}, \{\text{eau, extincteurs}\}, T_{c_1}, 2\text{heures}\}.$$

La tâche d'une coalition C se formalise comme un tuple $T_c = \{\Theta_{T_c}, S_{T_c}, \varphi_{T_c}\}$, où $\Theta_{T_c} = \{t_1, t_2, \dots, t_n\}$ comprend une ou plusieurs sous-tâches t_i et où $S_{T_c} = \{S_{t1}, S_{t2}, \dots, S_{tn}\}$ identifie l'ensemble des états des sous-tâches. $\varphi_{T_c} = (\alpha, \beta)$ est la localisation de T_c dans l'environnement, par exemple, la latitude α et la longitude β .

Exemple 2.3 Soit une tâche T_{c_1} qui a comme sous-tâche éteindre un feu à l'état stationnaire, localisée à la latitude et longitude $\{45.75, 4.82\}$ et une autre sous-tâche urgente pour sauver des vies localisée à la latitude et longitude $\{45.77, 4.87\}$. Cette tâche s'écrit formellement comme suit : $T_{c_1} = \{\{\text{éteindre un feu, sauver des vies}\}, \{\text{stationnaire, urgente}\}, \{45.75, 4.82\}, \{45.77, 4.87\}\}$.

L'agent initiateur est seulement responsable du début du processus de coordination, car une fois que le processus de coordination est entamé, tous les agents qui y participent doivent, pour atteindre leurs buts, synchroniser leurs actions de manière décentralisée pour exécuter la tâche.

Le temps d'exécution stochastique D_c de T_c est une durée non prévisible a priori pour la réalisation de T_c . Il dépend de l'évolution aléatoire des tâches. Pour estimer ce temps D_c , un agent qui identifie une tâche à exécuter commence par estimer le nombre moyen $\bar{N}$ d'agents requis pour la tâche en supposant que chaque agent qui accepte de participer à la coalition y contribue avec au moins une

ressource. D_c est donné par :

$$D_c = \frac{\bar{N}}{\sum_{i=1}^{\bar{N}} \bar{X}_{a_i}} \quad (2.4)$$

où $\bar{X}_{a_i}$ est la contribution marginale de l'agent a_i qui mesure son niveau de contribution dans la réalisation de la tâche. Ainsi, $\sum_{i=1}^{\bar{N}} \bar{X}_{a_i}$ est une estimation de la somme des contributions marginales de tous les agents de la coalition. On considère que tout agent qui initie une demande de formation d'une coalition pour une tâche ne possède aucune connaissance sur la contribution marginale des agents avec lesquels il doit coopérer sauf s'ils ont précédemment eu à coopérer. Dans l'ignorance, il fait comme hypothèse que les agents ont au moins le même niveau de contribution que lui.

Les contraintes associées à l'exécution de T_c sont exprimées par $\Delta_{T_c} = (T_c, D_c, Cons_c, val_c)$ où D_c est le temps d'exécution stochastique de la tâche T_c de C , $Cons_c$ les contraintes concernant les ressources acceptables et requises pour T_c (par exemple, autonomie énergétique minimale, fiabilité minimale, etc.) et val_c la récompense associée à la réalisation de T_c .

La récompense de l'agent a_i qui participe à la coalition C est calculée comme suit :

$$reward_{a_i} = val_c * \frac{\delta t_{a_i}}{D_c} \quad (2.5)$$

où val_c est la récompense associée à la réalisation de T_c , δt_{a_i} la période durant laquelle a_i est resté dans C . Ainsi, $reward_{a_i}$ est maximale si $\delta t_{a_i} = D_c$. Cette formulation de la récompense a pour objectif de motiver les agents à rester dans leur coalition et à minimiser les désengagements volontaires car un agent peut quitter sa coalition à tout instant, même s'il est initiateur de celle-ci.

Exemple 2.4 Les contraintes d'une sous-tâche, par exemple, éteindre un feu, peuvent être : $\Delta_{T_{c_1}} = (\text{éteindre un feu, 2 heures, \{disponible, 1 expérience\}, 15 \$})$, dans ce cas 2 heures est le temps d'exécution stochastique, $\{disponible, 1 expérience\}$ sont imposés par l'environnement de déploiement et 15 \$ est la récompense associée à l'exécution de la tâche.

Pour tenir compte de la disponibilité aléatoire des agents, nous définissons deux classes d'agents pour une coalition : les agents dits *primaires* qui sont les membres de la coalition et les agents dits *subordonnés* qui sont les suppléants des agents *primaires*.

Définition 2.7 Un agent $a_i \in A_{c_k}$ est dit *primaire*, s'il est engagé dans une coalition C_k et a choisi un agent suppléant $a_{sc_k} \in A \setminus A_{c_k}$ appelé *subordonné* qui s'engage à l'assister au cas où il serait en manque de ressources pour se maintenir dans C_k ou s'il doit rechercher de nouvelles ressources pour stabiliser C_k .

Définition 2.8 L'ensemble des agents *primaires* d'une coalition C est appelé *backbone*.

Notons qu'un agent ne peut appartenir qu'au *backbone* d'une seule coalition. Cette contrainte permet d'éviter que l'indisponibilité d'un agent n'entraîne l'instabilité successive de plusieurs coalitions. Ainsi, un agent bien qu'il ait été à l'origine de la coalition, une fois que celle-ci est établie,

il a la même responsabilité dans la coalition que les autres agents du *backbone* car le contrôle de la coalition devient décentralisé. Donc, même l'agent initiateur doit choisir un agent subordonné qui peut le remplacer en fonction des contraintes de la coalition. Si un agent du *backbone* n'a pas de subordonné, les autres agents primaires (*backbone*) de sa coalition vont prendre en charge son indisponibilité. Nous reviendrons plus en détails sur la relation entre *primaire* et *subordonné*.

La figure 2.1 met en exergue notre approche. Un agent peut être dans le *backbone* de la coalition C_2

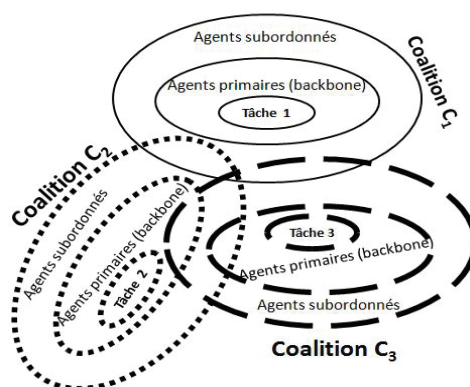


FIGURE 2.1 – Dispositions schématique des agents primaires et des agents subordonnés dans une structure de coalition.

et être un agent *subordonné* dans la coalition C_3 et vice versa. De plus, un agent peut être un agent *subordonné* dans deux coalitions C_1 et C_2 . Si un agent primaire de C_2 est aussi un agent *subordonné* dans C_3 , cela signifie que la ressource qu'il consent à fournir à son agent primaire, qui est dans C_3 , ne va pas exiger de même qu'il quitte sa coalition C_2 quand la coalition C_3 aura besoin de sa ressource. Cependant, suivant les stratégies dont un agent dispose pour maximiser ses *alliances* et son gain, il peut décider de prendre le risque que sa fiabilité soit réduite par les agents de C_2 ou par son agent *primaire* qui est dans C_3 respectivement s'il quitte sa coalition C_2 ou en cas de non-respect des engagements pris avec son agent primaire de C_3 .

Une restriction importante est qu'un agent doit être au plus dans un *backbone*. Cette contrainte évite le cas où des agents doivent renégocier leurs coalitions à cause de l'indisponibilité d'un seul agent *primaire*.

2.2.3 Règles générales de notre mécanisme de formation de coalitions

Cette section énumère des règles communes à toutes les étapes de la formation des coalitions des agents. Par souci de clarté, les règles spécifiques à chaque étape sont expliquées dans les sections concernées.

Règle 1. Schéma de communication. Chaque agent peut communiquer, à tout moment, avec

son voisinage pour transmettre un message (par exemple, pour initier une négociation). Cependant, pour assurer la convergence des négociations, pour gérer les coûts de communication et éviter les messages envoyés en boucles, nous définissons les concepts d'envoi de messages en mode *non-return broadcast* et de durée de validité *TTL* (Time To Live) des messages. Le mode *non-return broadcast* signifie que si un message, qui concerne une demande de coalition, une information sur une tâche, etc., est émis par un agent a_i , un agent voisin $a_j \in \vartheta_{a_i}$ ne doit pas lui retourner la même information. Tandis que, la durée *TTL* signifie qu'un message sera relayé par les agents jusqu'à ce que $TTL=0$ car à chaque fois qu'un agent doit transmettre un message, il décrémente de 1 la valeur du *TTL* du message reçu.

$$TTL = \left\lfloor \frac{Sz}{2 * Y} \right\rfloor \quad (2.6)$$

où Sz est la taille de la zone que couvre la tâche à exécuter, Y est la portée du signal de l'équipement où est déployé l'agent qui a envoyé le message de départ (par exemple, Bluetooth 100 mWatt, $Y \leq 100$ mètres). $TTL=1$ pour tout message de partage d'informations entre les agents *primaires* d'une même coalition C . De ce fait, un message qui ne concerne que C ne sera reçu que par les agents de C .

Règle 2 : Consistance d'une demande de formation d'une coalition. Un agent a_i ne peut faire une demande de formation d'une coalition C que si et seulement si $\vartheta_{a_i} \neq \emptyset$, $a_i \in A_c$, $R_{a_i} \in R_{min} \subset C$ et $G_c \subset C$ est requis pour former ou stabiliser la coalition C . Un agent a_j ne peut accepter une demande de formation d'une coalition que si $B_{a_j} \in G_c$ et si $R_{a_j} \in R_{min}$.

Règle 3 : Validation d'une participation. Une fois qu'un agent $a_j \in A \setminus A_c$ accepte la demande de formation d'une coalition C d'un agent a_i , l'agent $a_k \in A_c$ qui a reçu l'acceptation de l'agent a_j doit s'assurer que $R_{a_j} \subseteq R_{min} \subset C$ et qu'au moins une *alliance* Al_{a_k, a_j} est valide. Les deux agents a_k et a_j doivent partager leurs connaissances sur la ou les tâches T_c de C , sur la stabilité de C et se synchroniser avec les autres agents du *backbone* de C .

Règle 4 : Fin d'une négociation. La validation de la participation de l'agent a_j dans une coalition C par $a_k \in A_c$ et l'établissement de Al_{a_k, a_j} mettent fin à la négociation entre a_k et a_j . $\forall a_j \in A$ dont la participation est validée dans le *backbone* d'une coalition C , a_j doit suspendre grâce à un message de rejet toute autre négociation de formation d'une coalition C' si sa participation est requise dans le *backbone* de C' . Cela évite de laisser des négociations en suspend.

Pour assurer la fin des négociations, toute négociation avec un agent est suspendue automatiquement si la durée prévue de la négociation est atteinte sans trouver un accord. Cette durée n'est pas connue à l'avance, mais elle est bornée et déterminée par - les récompenses associées à l'exécution des tâches, - par les *alliances* proposées par les agents, et, - par le fait que la tâche T_c qui a induit la demande de formation de la coalition C est toujours contrôlable.

Dans notre modèle, cette durée de négociation est établie dynamiquement par chaque agent, car les

tâches et le contexte sont dynamiques. De ce fait, comme les agents sont autonomes et égoïstes, c'est à chaque agent d'allouer une durée à ses négociations en fonction de son autonomie énergétique et de ses stratégies pour maximiser ses récompenses et ses *alliances*.

Pour faciliter la stabilisation dynamique des coalitions qui favorise les accords et le partage d'informations entre les agents égoïstes et qui tient compte de l'instabilité imprévisible des agents, nous proposons un nouveau modèle de *recommandation*. Ce modèle a aussi pour objectif d'améliorer la capacité des agents à atteindre de manière décentralisée leurs buts.

2.3 MODÈLE DE RECOMMANDATION

Notre modèle de *recommandation* repose sur les *alliances* et les fiabilités des agents. Pour faciliter la recherche de l'agent le plus fiable pour une tâche, chaque agent utilise une représentation hiérarchique de ses alliés où chaque identification a_j d'un allié est couplée à sa fiabilité ρ_{a_j} avant de le lier : - à la ou les ressources R_{a_j} que l'allié a_j est prêt à fournir dans le cadre de son *alliance*, et, - à un intervalle de probabilité de stabilité qui inclut $P_s^{a_j}$ de l'allié a_j . Nous appelons ce modèle hiérarchique *l'historique* (cf. Figure 2.2), puisqu'il enregistre des données d'interactions de chaque agent qui peuvent être maintenues et mises à jour dynamiquement dans le temps. Les *alliances* et les *fiabilités* des agents utilisées par notre modèle de *recommandation* sont obtenues à partir de cet *historique*. La représentation hiérarchique de *l'historique* est pratique pour une mise à jour dyna-

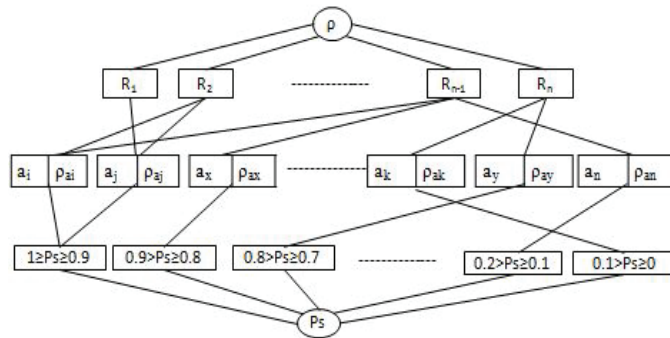


FIGURE 2.2 – Représentation hiérarchique de *l'historique*

mique sans nécessiter des restructurations ou des calculs supplémentaires. En effet, pour ajouter un nouvel allié, un agent lie son allié avec une plage de probabilité de stabilité qui inclut celui de son allié et une fiabilité d'initialisation ρ qui est égale à 1 en fonction de la ou des ressources fournies par l'agent allié. La suppression d'un allié devenu non fiable par un agent a_i n'affecte pas la cohérence de *l'historique* de l'agent a_i qui doit supprimer que l'identifiant de cet allié. La mise à jour de la fiabilité ou de la probabilité de stabilité d'un allié est aussi simplifiée, car la structure

hiérarchique de l'*historique* permet un accès direct à ces informations.

Mécanisme de recommandation.

Les agents utilisent le mécanisme de *recommandation* lorsqu'un agent demande la recherche d'un ensemble de ressources pour la formation ou la stabilisation d'une coalition.

Lorsqu'un agent a_i souhaite trouver une ou des ressources suite à la demande de *recommandation*, il utilise son *historique* Hs_{a_i} . Pour cela, l'agent a_i commence par rechercher l'ensemble des agents alliés $a_j \in Hs_{a_i}$ qui ont au moins une ou des ressources requises et telle que leur fiabilité ρ_{a_j} est maximale. Pour chaque ressource requise, s'il y a plus d'un choix d'allié, a_i trie ces alliés suivant leur probabilité de stabilité $P_s^{a_j} : a_j \in Hs_{a_i}$ avant de les contacter un à un suivant un ordre décroissant de valeur de probabilité de stabilité.

Pour envoyer une demande de *recommandation*, un agent a_i utilise le mécanisme de communication en mode *non-return broadcast* et une contrainte de *TTL* (cf. équation 2.6). L'agent a_i choisit un agent qui a une ressource requise si la fiabilité et la probabilité de stabilité de cet agent sont plus élevées pour cette ressource. Cependant, la fiabilité est regardée en premier lieu puisqu'un agent ne peut pas être jugé stable s'il n'est pas fiable.

Lorsqu'un agent a_i reçoit une demande de *recommandation* d'un agent a_j , pour une tâche donnée, deux cas se posent :

Cas 1 : a_i a une *alliance* avec a_j .

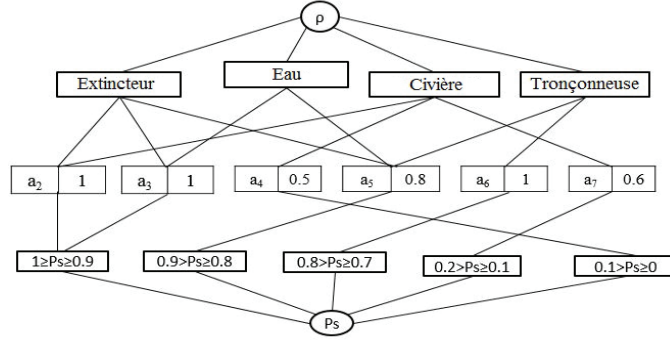
Dans ce cas, a_i examine l'*alliance* qu'il partage avec a_j . Si a_i peut honorer l'*alliance*, il propose sa ou ses ressources et attend la confirmation de sa participation dans la coalition de l'agent a_j . Si a_i ne peut pas honorer l'*alliance*, il recherche parmi les alliés de son *historique*, un ou des agents qui peuvent honorer l'*alliance*. Si a_i identifie un ensemble d'alliés, il répond à la demande de *recommandation* en proposant cet ensemble. Si a_i ne peut toujours pas répondre à la demande de *recommandation*, il l'ignore mais en sachant que sa fiabilité sera réduite au niveau de l'agent a_j .

Cas 2 : a_i n'a pas d'*alliance* avec a_j .

Dans ce cas, a_i envoie à a_j une demande d'établissement d'*alliance* avant d'attendre la réponse de ce dernier. Si a_j accepte une *alliance*, a_i enregistre a_j dans son *historique* et procède aux étapes du **cas 1** ci-dessus. Si a_j n'accepte pas l'*alliance*, a_i ignore la demande de *recommandation*.

Exemple 2.5 Soit a_1 un agent qui a l'*historique* de la Figure 2.3 ci-dessous et qui souhaite trouver plus de ressources en utilisant ses *alliances*.

Supposons que l'agent a_1 souhaite trouver un ensemble d'agents parmi ses alliés pouvant fournir la ressource nommée *extincteur*. La Figure 2.3 montre la fiabilité ρ et la probabilité de stabilité P_s de l'ensemble des agents de l'environnement avec qui a_1 a une *alliance*. Ses alliés sont $\{a_2, a_3, a_4, a_5, a_6, a_7\}$. Ainsi, le résultat peut être

FIGURE 2.3 – Historique de l'agent a_1 .

a_2 , a_3 et a_5 . En classant ses alliés suivant leur fiabilité et leur probabilité de stabilité, si les alliances préférées de l'agent a_1 sont telles que l'agent $a_2 \succ_{AI} a_3 \succ_{AI} a_5$ alors a_2 est préféré à a_3 qui est, à son tour, préféré à a_5 .

Dans la section suivante, nous décrivons notre méthode de formation de coalitions stables du point de vue de Nash (Nash 1950) et qui tient compte de la disponibilité aléatoire des agents et de la dynamique des tâches.

2.4 MÉCANISME DE FORMATION DE COALITIONS

Pour assurer la *persistance* de leurs coalitions, les agents *primaires* analysent la stabilité de ces coalitions après des événements d'ajout de ressource(s), de retrait d'un ou des agents *primaires*, etc. Cette analyse est effectuée en déterminant le vecteur invariant qui fournit la probabilité que la ou les tâches de leur coalition deviennent incontrôlables et évalue la probabilité que ces événements les amènent à renégocier leur coalition. Les détails sur cette analyse sont donnés ci-dessous.

2.4.1 Le vecteur invariant d'une tâche

Un agent crée le MDP qui permet de prendre les décisions concernant l'exécution d'une tâche en utilisant les données issues de son environnement qu'il a obtenues grâce à un ensemble de dispositifs tels que des capteurs et des satellites (dont le fonctionnement ne rentre pas dans le cadre de ce travail). Formellement, un MDP se définit comme $\{S, Act, \pi_{s^i, s^j}^T, Rew\}$, où $S = \{s^0, s^1, \dots, s^n\}$, $n \in \mathbb{N}$ est un ensemble fini d'états, Act un ensemble fini d'actions qu'un agent peut entreprendre (par exemple, envoyer une demande de formation d'une coalition, négocier,...), π_{s^i, s^j}^T une probabilité de transition d'un état s^i vers un état s^j après une action $act_k \in Act$ sur la tâche T . Rew définit la récompense obtenue après chaque action.

Pour calculer la probabilité de transition d'un état s^i vers un autre état s^j on ignore l'historique de la chaîne de Markov et on ne se fonde que sur l'état courant s^i car une chaîne de Markov possède la propriété sans mémoire : $P(X_n|X_0, \dots, X_{n-1}, act_k) = P(X_n|X_{n-1}, act_k)$, ainsi $\pi_{s^i, s^j}^T = P(X_n = s^j | X_{n-1} = s^i, act_k)$.

Comme les agents évoluent dans un contexte dynamique, ils doivent calculer dynamiquement les différentes probabilités de transition des états de la chaîne de Markov.

(a) Chaque agent estime la probabilité (π_{s^i, s^j}^T) de transition entre les différents états du MDP.

Soit $A_{\delta t}^n$ la probabilité d'avoir $card(\vartheta_{a_k})=n$ agents dans une *vue* ϑ_{a_k} d'un agent a_k durant un intervalle de temps δt . Alors, la variable n suit une distribution de *Poisson* car dans un intervalle de temps δt nous nous intéressons au taux d'arrivée des agents dans la *vue* de l'agent a_k . Ainsi, $A_{\delta t}^n = e^{-(\frac{n}{\delta t})} * \frac{(\frac{n}{\delta t})^n}{n!}$.

Soit $P_{\delta t}^x$ la probabilité d'avoir x agents dans ϑ_{a_k} qui acceptent de rejoindre la coalition après une action act_k (par exemple, une demande de formation d'une coalition, une demande de *recommandation*). Pour déterminer $P_{\delta t}^x$, nous utilisons la loi Binomiale qui est la loi de probabilité a priori si nous recherchons le nombre de succès dans une succession d'expériences identiques et où chaque expérience a deux issues possibles (succès ou échec) (Yates et Goodman 2005), (Baynat 2000). Alors $P_{\delta t}^x = C_n^x p^x q^{n-x}$ où p est la probabilité du succès et $q=1-p$ est la probabilité de l'échec.

Si nous considérons qu'un événement peut induire l'instabilité d'une coalition C si un agent a_k de C fait face à un événement, au moment où il n'a aucun agent voisin qui peut l'aider à surmonter les conséquences des événements, alors, la probabilité de transition d'un état à un autre de la chaîne de Markov peut s'exprimer par : $P_{s^i, s^j}^{i \rightarrow j} = 1 - A_{\delta t}^n$.

Ainsi, la probabilité de transition d'un état s^i vers un état s^j , $s^i \neq s^j$ sachant que $P_{\delta t}^x$ est donnée par :

$$\pi_{s^i, s^j}^T = P(X/Y) = \frac{P(X \cap Y)}{P(Y)} : X = P_{s^i, s^j}^{i \rightarrow j}, Y = P_{\delta t}^x. \quad (2.7)$$

Pour chaque tâche T , $\sum_{j=0}^n \pi_{s^i, s^j}^T = 1$, $i, j, n \in N$ car la matrice P de chaque chaîne de Markov est stochastique. P est telle que, $\forall i, j \in [0, 2]$ où s^0 est l'état *stationnaire*, s^1 l'état *régression* et s^2 l'état *amplification* :

$$\begin{bmatrix} \pi_{s^0, s^0}^T & \pi_{s^0, s^1}^T & \pi_{s^0, s^2}^T \\ \pi_{s^1, s^0}^T & \pi_{s^1, s^1}^T & \pi_{s^1, s^2}^T \\ \pi_{s^2, s^0}^T & \pi_{s^2, s^1}^T & \pi_{s^2, s^2}^T \end{bmatrix}$$

Cette formulation des probabilités de transition de la chaîne de Markov vise à adapter dynamiquement le calcul du vecteur invariant des tâches.

(b) Chaque agent détermine le vecteur invariant de la chaîne de Markov en calculant l'équa-

tion :

$$\Pi * Q = 0 \quad (2.8)$$

où $Q = I - P$, $\sum_{j=0}^2 \Pi_j = 1$, I est une matrice identité et P la matrice stochastique du MDP.

La résolution de l'équation 2.8 pour l'obtention du vecteur invariant du MDP est donnée par l'algorithme 1 suivant :

Algorithme 1 : Calcul du vecteur invariant.

Require: *nb_iteration* est le nombre d'itérations requises pour que la chaîne de Markov converge, P la matrice stochastique de la chaîne de Markov de la tâche.

- 1: $[li, col] = size(P)$ pour obtenir la taille de P .
 - 2: $\Pi_0 = rand(1, li)$ pour générer un vecteur de même dimension que P ayant des valeurs aléatoires.
 - 3: $sum_li = \sum_{i=0}^{li} \Pi_0[i]$
 - 4: **for** $j = 1 : li : j++$ **do**
 - 5: $\Pi_0[j] = \Pi_0[j] / sum_li$ pour assurer que Π_0 est stochastique.
 - 6: **end for**
 - 7: $X = \Pi_0$
 - 8: **for** $i = 1 : nb_iteration : i++$ **do**
 - 9: $X = X * P$
 - 10: **end for**
 - 11: $\Pi = X$
 - 12: **return** Π
-

Exemple 2.6 Considérons une tâche T et un ensemble d'état $S = \{s^0, s^1, s^2\}$ possible pour la tâche, où s^0 est l'état "stationnaire", s^1 l'état "régression" et s^2 l'état "amplification". Si par exemple, la matrice P égale :

$$\begin{bmatrix} \frac{1}{4} & 0 & \frac{3}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{bmatrix}$$

Le calcul de $\Pi * Q = 0$ pour obtenir $\Pi = (\Pi_0, \Pi_1, \Pi_2)$ donne $\Pi = (\frac{1}{4}, \frac{3}{10}, \frac{9}{20})$. Cela signifie que la probabilité que la tâche soit dans s^0 égale $\Pi_0 = \frac{1}{4}$, $\Pi_1 = \frac{3}{10}$ dans s^1 et $\Pi_2 = \frac{9}{20}$ dans s^2 .

2.4.2 Négociation multilatérale et formation de coalitions

Le mécanisme permettant à un agent a_i de gérer sa demande de formation d'une coalition C pour une tâche T_c est décrit par l'algorithme 2. Dans cet algorithme, a_i commence par calculer le temps

de réponse approximatif ϵt nécessaire pour que l'agent voisin qu'il considère comme le plus éloigné, ait le temps de répondre à sa demande. $\epsilon t = (2 * dt_1) + dt_2$ où dt_1 regroupe le temps de modulation du message à transmettre et de démodulation du message reçu tandis que dt_2 est le temps nécessaire pour envoyer et recevoir un message sur le réseau de communication.

À l'étape suivante, a_i diffuse un message *Declare*($\{\Pi_0, \Pi_1, \Pi_2\}, \Delta_{T_c}, R_{min}, Al_{a_i}^P$) qui contient : (1) le vecteur invariant $\Pi = \{\Pi_0, \Pi_1, \Pi_2\}$ de la tâche T_c , (2) les contraintes Δ_{T_c} associées à T_c , (3) l'information sur le minimum de ressources R_{min} requis pour T_c et (4) la proposition d'une *alliance* $Al_{a_i}^P$. La proposition $Al_{a_i}^P$ contient la ou les ressources R_{a_i} et le temps $T_{help}^{a_i}$ durant lequel R_{a_i} vont être mises à la disposition d'un agent qui accepte l'*alliance* et qui rejoint la coalition C de l'agent a_i . À noter que $T_{help}^{a_i} = D_c - \omega t$ où ωt est la durée de la négociation d'*alliance*. Cette expression de $T_{help}^{a_i}$ permet d'éviter les périodes de négociation d'*alliances* qui peuvent ralentir au lieu d'accélérer la négociation proprement dite des coalitions.

Si un agent a_j est disposé à rejoindre C et qu'il consent avec $Al_{a_i}^P$, il envoie à a_i un message *Help*($P_s^{a_j}, t_i, R_{a_j}$) qui contient la sous-tâche $t_i \in T_c$ qu'il souhaite gérer, sa probabilité de stabilité $P_s^{a_j}$ (cf. équation 2.1) et sa ou ses ressources R_{a_j} . Si l'agent a_i valide la participation de l'agent a_j avec un message *Def_Ack*(a_j) parce que $R_{a_j} \in R_{min}$, alors l'*alliance* Al_{a_i, a_j} est établie et chaque agent met à jour son *historique* et l'information R_{a_j} est supprimée de R_{min} .

À la prochaine étape, si nécessaire, a_i et a_j synchronisent leurs connaissances à propos des autres agents qui participent au processus de formation de la coalition pour l'exécution de la tâche T_c .

Notons que, chaque agent a_j notifié par un *Def_Ack*(a_j) à cette étape devient lui aussi un agent *primaire*, responsable de sa sous-tâche $t_i \in T_c$ et doit identifier un agent *subordonné* ou plus de ressources si c'est nécessaire pour la coalition. Cependant, si un agent *primaire* devient indisponible, sa sous-tâche et son *subordonné* sont gérés par les autres agents *primaires* avec lesquels il partage le même *backbone* pour ainsi pallier son départ.

Si un agent a_j veut demander à un autre agent a_i de modifier la ou les ressources R_{a_i} et ou le temps $T_{help}^{a_i}$ proposé(s) dans $Al_{a_i}^P$, alors a_j envoie un message *SendNewAl*(Al_{a_j, a_i}). L'agent a_i vérifie si cette modification engendrerait une *alliance* Al_{a_j, a_i} équitable. Si tel est le cas et que l'agent a_i accepte Al_{a_j, a_i} , alors a_i valide la participation de l'agent a_j par un message *AckSendNewAl*(Al_{a_j, a_i}) et les deux agents mettent à jour leur *historique*. Dans le cas contraire, a_i ignore la proposition de modification de l'agent a_j .

Dans le cas où a_i valide la participation de l'agent a_j , il supprime de la liste R_{min} préétablie pour la coalition, la ou les ressources R_{a_j} . À l'issue de cette étape, a_i et a_j synchronisent leurs connaissances à propos de la tâche et se coordonnent avec les autres membres de la coalition s'il en existe. Tout agent a_j accepté pour satisfaire R_{min} devient agent *primaire*, responsable de sa sous-tâche t_j et doit chercher un *subordonné* (*Find_substitute*(Δ_{t_i})).

Cette distribution de la gestion des sous-tâches t_i de T_c permet d'éviter une centralisation du contrôle de la coalition.

Algorithme 2 : Négociation et formation de la coalition.**Require:** Une tâche T_c .

- 1: Calculer la matrice stochastique P de la tâche T_c et l'équation $\Pi * Q=0$ pour trouver le vecteur invariant $\{\Pi_0, \Pi_1, \Pi_2\}$ cf. *algorithme 1*.
- 2: Calculer L qui est la distance maximale avec le voisin le plus éloigné.
- 3: Calculer $dt_1=(2 * \frac{Tmsg}{DebNet})$, $dt_2=(2 * \frac{L*Tmsg}{DebNet})$ et $\epsilon t=(2 * dt_1) + dt_2$ // $Tmsg$ est la taille du message et $DebNet$ est le débit du support de communication.
- 4: Collecter les données sur la taille S_z que couvre la zone sinistrée et les informations sur le minimum de ressources requises R_{min} .
- 5: Calculer le TTL (Time To Live) du message de déclaration de la tâche T_c à négocier. $TTL=|\frac{S_z}{L}|$.
- 6: $D_c=\frac{\bar{N}}{\sum_{i=1}^N \bar{X}_{a_i}}$ et $T_{help}^{a_i}=D_c$ et $Al_{a_i}^P=(R_{a_i}, T_{help}^{a_i})$.
- 7: Déterminer les contraintes $\Delta_{T_c}=(T_c, D_c, Cons_c, val_c)$ et envoyer le message de déclaration $Declare(\{\Pi_0, \Pi_1, \Pi_2\}, \Delta_{T_c}, R_{min}, Al_{a_i}^P)$ de la tâche T_c à négocier.
- 8: **repeat**
- 9: **if** ($Help(P_s^{a_j}, t_i, R_{a_j})$) et $t_i \in T_c$ et $R_{a_j} \in R_{min} : a_j \in A \setminus a_i$ **then**
- 10: Supprimer R_{a_j} de R_{min} .
- 11: $Def_Ack(a_j)$ et enregistrer a_j dans l'historique.
- 12: **else**
- 13: $D_c=\frac{\bar{N}}{\sum_{i=1}^N \bar{X}_{a_i}}$ et $T_{help}^{a_i}=(T_{help}^{a_i} - \epsilon t)$ et $Al_{a_i}^P=(R_{a_i}, T_{help}^{a_i})$.
- 14: $Declare(\{\Pi_0, \Pi_1, \Pi_2\}, \Delta_{T_c}, R_{min}, Al_{a_i}^P)$
- 15: **end if**
- 16: Attendre une réponse durant ϵt et incrémenter le nombre de tentatives ($k++$) qui ne doit pas dépasser 3. //Cela, pour rester cohérent par rapport au fonctionnement de la plupart des réseaux ad-hoc et pour ne pas rester dans une impasse en l'absence de ressources disponibles.
- 17: **until** ($k == 3$) || ($R_{min} == \emptyset$)
- 18: **if** ($R_{min} == \emptyset$) **then**
- 19: $Find_substitute(\Delta_{T_c})$.
- 20: **else**
- 21: $Recom(R_{min}, \Pi_0, \Pi_1, \Pi_2, Al_{a_i}^P)$.
- 22: **if** ($SendRecom(a_j, Al_{a_i, a_j})$) **then**
- 23: $AckSendRecom(a_j, Al_{a_i, a_j})$.
- 24: **if** $AckRecom(a_j)$ et $R_{a_j} \in R_{min}$ **then**
- 25: Supprimer R_{a_j} de R_{min} .
- 26: $Def_Ack(a_j)$ et enregistrer a_j dans l'historique.
- 27: **end if**
- 28: **end if**
- 29: **if** $SendNewAl(Al_{a_j, a_i})$ et Al_{a_j, a_i} est équitable **then**
- 30: $AckSendNewAl(Al_{a_j, a_i})$.
- 31: $Def_Ack(a_j)$ et enregistrer a_j dans l'historique.
- 32: **end if**
- 33: **end if**

Si les ressources minimales R_{min} pour la coalition ne sont pas trouvées, alors chaque agent a_i du *backbone* de C utilise son *historique* Hs_{a_i} pour trouver des alliés a_j de fiabilité ρ_{a_j} maximale qui ont les ressources recherchées. Pour chaque ressource, s'il y a plusieurs alliés qui peuvent la fournir, a_i les trie suivant leur probabilité de stabilité avant de les contacter ($Recom(R_{min}, \Pi_0, \Pi_1, \Pi_2, Al_{a_i}^P)$) un à un, par ordre décroissant de probabilité de stabilité. Ce message $Recom(R_{min}, \Pi_0, \Pi_1, \Pi_2, Al_{a_i}^P)$ est une demande de *recommandation* qui signifie qu'un allié a_j de l'agent a_i doit fournir $R_{a_j} \in R_{min}$ conformément à l'*alliance* sous peine de voir a_i réduire sa fiabilité ρ_{a_j} (cf. équation 2.3).

Si un agent a_i reçoit une réponse $SendRecom(a_j, Al_{a_i, a_j})$ de la part d'un agent a_j recommandé, l'agent a_i lui envoie un message $AckSendRecom(a_j, Al_{a_i, a_j})$. Après cela, si a_i reçoit le message d'acquiescement $AckRecom(a_j)$ de l'agent a_j , alors a_i et a_j mettent à jour leur *historique* et en informent le reste des agents de la coalition.

La recherche de *subordonné* ($Find_substitute(\Delta_{t_i})$) par $a_i \in C$ commence par l'envoi des contraintes Δ_{t_i} de la sous-tâche $t_i \in T_c$ qu'il gère et de l'envoi de l'*alliance* $Al_{a_i}^P$ qu'il propose à son voisinage pour établir un accord de subordination. Chaque agent $a_j \in A \setminus A_c$ intéressé, envoie sa probabilité de stabilité $P_s^{a_j}$ à a_i . Après une attente de durée égale à ϵt , a_i sélectionne la meilleure $P_s^{a_j} : a_j \in A \setminus A_c$ reçue avant de le notifier à l'intéressé. Cette notification permet aux agents non retenus d'être disponibles pour d'autres agents qui recherchent un agent *subordonné*. Si l'agent a_j choisi d'accepter, alors a_i l'enregistre comme son *subordonné*, sinon a_i notifie l'agent qui a la meilleure probabilité de stabilité qui suit. Lorsque l'agent a_i trouve un *subordonné* a_j , il partage cette information avec les autres agents *primaires* de sa coalition C qui pourront contacter a_j si a_i quitte C sans avoir eu le temps d'en notifier a_j . Si a_i n'a pas trouvé de *subordonné*, il fait une demande de *recommandation* aux agents de son *historique* Hs_{a_i} avant d'attendre les réponses.

L'utilisation du mécanisme de *recommandation*, à cet instant, dans le processus de recherche de *subordonné* se justifie par le fait que dans notre mécanisme où les agents sont égoïstes, nous cherchons à promouvoir une stratégie qui favorise l'établissement de nouvelles *alliances* dans le voisinage des agents.

Comme précisé dans les règles générales de notre mécanisme de formation de coalitions, durant les différentes négociations, tant que le *TTL* d'un message est toujours valide, les agents qui reçoivent ce message doivent y répondre en respectant le principe d'interaction en mode *non-return broadcast* pour la gestion de la bande passante et de la complexité des négociations.

2.5 DÉROULEMENT DE NOTRE MÉCANISME SUR UN EXEMPLE

Soit $A = \{a_1, a_2, a_3, a_4, a_5\}$ un ensemble d'agents qui doivent être divisés en paires stables pour l'exécution des tâches d'une zone sinistrée. Chaque agent est implémenté dans un unique composant ad-hoc mobile (Smart-phone, robot, capteur, ...). Ainsi, les propriétés (ressources, autonomie

énergétique, etc.) d'un agent dépendent du composant qui l'héberge et des propriétés du robot ou de l'humain qui utilise le composant.

Soit a_1 et a_4 deux agents *initiateurs* d'un processus de formation de coalitions pour un ensemble de tâches. Les agents utilisent la technologie Bluetooth (portée de signal de 100 mètres, taille de trame de communication de 2871 bits et un débit réseau de 3Mbps, c'est-à-dire, 3×10^6 bps) des composants pour leurs communications.

Soit la configuration multi-agent initiale de la Figure 2.4 suivante. Ainsi, a_3 et a_5 sont les agents voisins de l'agent a_1 ($\vartheta_{a_1} = (\{a_3, x_{1,3}\}, \{a_5, x_{1,5}\})$) et les agents voisins de l'agent a_4 sont a_2 , a_3 et a_5 ($\vartheta_{a_4} = (\{a_2, x_{4,2}\}, \{a_3, x_{4,3}\}, \{a_5, x_{4,5}\})$).

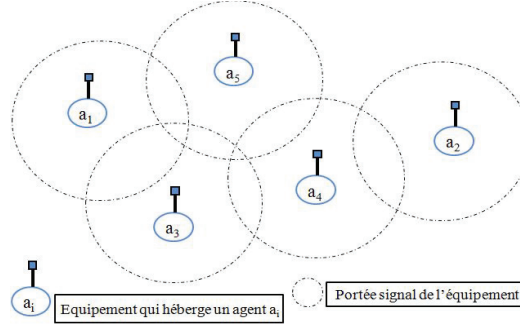


FIGURE 2.4 – Topologie de cinq composants ad-hoc mobiles (Smartphones, robots, capteurs, ...) où chacun de ces composants utilise la technologie Bluetooth et implémente un agent a_i .

À l'initialisation, le voisinage et la vue d'un agent sont identiques et l'historique de chaque agent est vide puisque les *alliances* ne sont pas encore établies ($x_{1,3}=x_{1,5}=x_{4,2}=x_{4,3}=x_{4,5}=Faux$). Du fait de l'absence de connaissances a priori sur les agents et les tâches à exécuter, aucun agent ne connaît les ressources, les intentions, la fiabilité et la probabilité de stabilité des autres agents ou les évolutions possibles des tâches de l'environnement de déploiement.

Étape 1 : Initialisation de la déclaration des tâches.

(a) a_1 et a_4 calculent le vecteur invariant (cf. algorithm 1) de la ou des tâches pour lesquelles ils cherchent à former une coalition.

(b) a_1 et a_4 calculent une estimation du temps de réponse $\epsilon t = (2 * dt_1) + dt_2$ durant lequel ils vont attendre des réponses à leur demande de coalitions.

$$dt_1 = (2 * \frac{2871}{3 * 10^6}).$$

$$dt_2 = 2 * \frac{L}{\frac{3 * 10^6}{2871}}.$$

où L est la distance la plus grande avec un voisin directe. Par conséquent, si L égale $2 * 100$ alors ϵt égal 0.386628 sec.

(c) a_1 et a_4 calculent leur *TTL* initial pour leur message de demande de formation de coalitions qui va induire une phase de négociation. $TTL = \lfloor \frac{Sz}{L} \rfloor$ où Sz est la taille de la zone de déploiement des agents (par exemple, zone sinistrée). Si nous considérons que $Sz=300$ mètres pour a_1 et $Sz=150$

mètres pour a_4 , alors le *TTL* initial du premier message de demande de formation d'une coalition de l'agent a_1 est égale à 2 et pour a_4 , *TTL*=1.

(d) a_1 et a_4 calculent encore, chacun de son côté, l'*alliance* qu'il va proposer à son voisinage.

- a_1 calcule $Al_{a_1}^p$ qui contient sa ressource R_{a_1} et le temps $T_{help}^{a_1}$ qui est initialement égal au temps d'exécution stochastique D_{c_1} de sa tâche T_{c_1} . Initialement, a_1 n'a aucune connaissance sur les autres agents $a_i \in A$, alors $D_{c_1} = \frac{2}{X_{a_1} + X_{a_i}} = \frac{1}{X_{a_1}}$.

X_{a_1} est la contribution marginale de l'agent a_1 si la coalition pour sa tâche est établie. Si X_{a_1} est égale à 2, alors $D_{c_1} = \frac{1}{2}$ heure.

- a_4 calcule aussi son $Al_{a_4}^p$ qui contient sa ressource R_{a_4} et le temps $T_{help}^{a_4}$ qui est initialement égal au temps d'exécution stochastique D_{c_4} de sa tâche T_{c_4} . De même, initialement a_4 n'a aucune connaissance a priori sur les autres agents $a_j \in A$, alors $D_{c_4} = \frac{2}{X_{a_4} + X_{a_i}} = \frac{1}{X_{a_4}}$.

X_{a_4} est la contribution marginale de l'agent a_4 si la coalition pour sa tâche est validée. Si nous supposons que $X_{a_4}=1$, alors $D_{c_4}=1$ heure.

(e) a_1 et a_4 déterminent les contraintes associées à leurs tâches.

- La tâche T_{c_1} de l'agent a_1 est une extinction d'un feu de temps d'exécution stochastique $D_{c_1} = \frac{1}{2}$ heure. a_1 détermine, par exemple, les contraintes $\Delta_{T_{c_1}} = (\text{Feux}, \frac{1}{2} \text{ heure}, \{\text{disponibilité}, 1 \text{ expérience}\}, 15 \$)$.
- La tâche T_{c_4} de l'agent a_4 , concerne aussi l'extinction d'un feu mais de temps d'exécution stochastique $D_{c_4}=1$ heure. a_4 détermine, par exemple, les contraintes $\Delta_{T_{c_4}} = (\text{Feux}, 1 \text{ heure}, \{\text{disponibilité}, P_s \geq 0.5\}, 20 \$)$.

Étape 2 : Déclaration des demandes de formation des coalitions pour les tâches.

Chacun des deux agents a_1 et a_4 envoie un message de déclaration pour annoncer son souhait de former une coalition pour sa tâche (cf. Figure 2.5). Leurs agents voisins respectifs propagent cette déclaration en respectant le mode de communication *non-return broadcast* tant que le *TTL* du message de déclaration n'est pas épuisé.

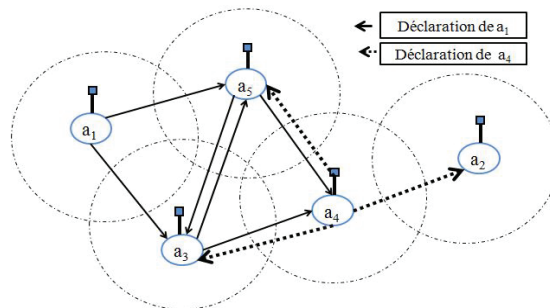


FIGURE 2.5 – a_1 et a_4 déclarent leur intention de former une coalition pour leurs tâches.

Étape 3 : Le choix des agents voisins.

Si les pr f rences de l'agent a_5 en fonction des *alliances* propos es par a_1 et a_4 sont telles que : $a_4 \succ_{AI} a_1$ ou $a_1 \succ_{AI} a_4$ si a_1 change les contraintes $\Delta_{T_{c_1}}$ en, par exemple, $\Delta_{T_{c_1}} = (Feux, \frac{1}{2} \text{ heure}, \{disponibilit \}, 15 \$)$. Ce qui signifie que l'agent a_5 pr f re a_4 sauf si la t che T_{c_1} de l'agent a_1 ne requiert pas une exp rience. Si la pr f rence de l'agent a_3 est telle que l'agent $a_4 \succ_{AI} a_1$ et celle de l'agent a_2 est telle que l'agent $a_4 \succ_{AI} a_1$, alors, a_2 , a_3 et a_5 vont r pondre   a_4 en envoyant leur probabilit  de stabilit  $P_s^{a_i}$ et leurs ressources R_{a_i} o  $a_i \in \{a_2, a_3, a_5\}$ (cf. Figure 2.6). De plus, a_5 envoie une requ te   l'intention de l'agent a_1 pour une demande d' tablissement d'une nouvelle *alliance* $Al_{a_1}^p$ et pour tenir compte de sa demande de modification des contraintes $\Delta_{T_{c_1}}$.

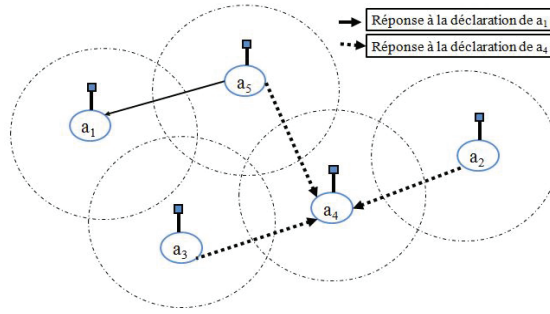


FIGURE 2.6 – Les agents a_2 et a_3 choisissent de rejoindre la coalition de l'agent a_4 . Cependant, a_5 r pond favorablement   a_4 , mais   a_1 , il r pond pour demander l' tablissement d'une nouvelle *alliance* $Al_{a_1}^p$ qui tient compte de sa demande de modification des contraintes $\Delta_{T_{c_1}}$.

 tape 4 : Validation des r ponses favorables par les agents initiateurs a_1 et a_4 .

Apr s le d lai d'attente ϵt des r ponses, si la pr f rence de l'agent a_4 est $a_5 \succ_{AI} a_2 \succ_{AI} a_3$, alors a_4 envoie un message d'acquiescement   a_5 pour lui sp cifier qu'il est retenu en tant qu'agent *primaire* de la coalition et que l'*alliance* Al_{a_4, a_5} est valid e. Apr s quoi, a_4 met   jour son *historique*. a_5 propose   a_1 de devenir son agent *subordonn *.

  partir de cet instant, quelle que soit la raison (panne, indisponibilit , ...), si un des deux agents a_4 et a_5 interrompt l'ex cution de la t che, l'agent qui est rest  dans la coalition, r duit la fiabilit  de celui qui a quitt  celle-ci. Par exemple, si a_5 ne rejoint pas le *backbone* de la coalition de l'agent a_4 , alors a_4 calcule la nouvelle fiabilit  de l'agent a_5 comme suit :

$\rho_{a_5} = e^{(-\lambda_{a_5})}$ o  λ_{a_5} est le taux de d fections de l'agent a_5 par rapport   une *alliance* avec a_4 . Ainsi, $\rho_{a_5} = e^{-(0.1)} = 0.9$.

Si   ce stade, a_1 trouve que la demande de modification de l'*alliance* par a_5 entra ne une alliance qui n'est pas  quitable, alors a_1 ignore la demande de l'agent a_5 . Apr s expiration du d lai d'attente ϵt , a_1 envoie de nouveau un message de demande de formation de la coalition pour sa t che.

Le calcul de la fiabilit  et sa mise   jour dynamique sont discut s dans la section 2 (cf.  quation 2.3). Les tableaux 2.1 et 2.2 montrent les  volutions de la fiabilit  d'un agent a_i en fonction de ses d fections et de ses participations au cours du temps.

Taux de défections de l'agent a_i	0	1	2	3	4	5	6	7
λ_{a_i}	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
ρ_{a_i}	1	0.9	0.8	0.7	0.67	0.6	0.5	Suppression <i>alliances</i>

TABLE 2.1 – Évolutions de la fiabilité de l'agent a_i s'il n'a jamais respecté ses engagements dans le temps.

Pour améliorer ou conserver sa fiabilité, l'agent a_i doit respecter ses engagements et participer à une coalition. Par exemple, si $\lambda_{a_i}=0.5$ pour un agent a_j , alors, si a_i forme au cours du temps un ensemble de coalitions avec a_j , l'évolution de la fiabilité de l'agent a_i est sous la forme présentée dans le tableau 2.2. Comme nous l'avons montré dans le tableau 2.1, si la fiabilité est de 0.5, alors $\lambda_{a_i}=0.6$.

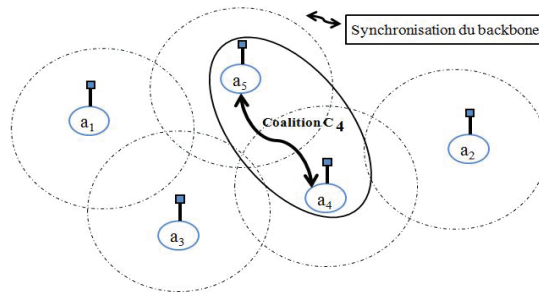
Nombre d'engagements respectés par a_i	0	1	2	3	4	5	6	7
λ_{a_i}	0.6	0.5	0.4	0.3	0.2	0.1	0	0
ρ_{a_i}	0.5	0.6	0.67	0.7	0.8	0.9	1	1

TABLE 2.2 – Évolutions de la fiabilité de l'agent a_i sachant que la fiabilité initiale est de 0.5

Étape 5 : Formation du *backbone* de la coalition.

Si a_5 est encore en accord avec a_4 , il met à jour son *historique* et rejoint le *backbone* avec a_4 (cf. Figure 2.7). Après cela, a_5 propose à l'un des agents a_1 et a_3 de devenir un agent *subordonné* sachant que c'est le premier des deux agents qui accepte qui devient *subordonné*.

Si a_5 décide de mettre fin à sa collaboration avec a_4 à cause d'une indisponibilité ou autres de ce dernier, a_5 envoie un message *non-ack* et propose à a_2 de devenir un agent *primaire* et à a_3 de devenir son agent *subordonné*.

FIGURE 2.7 – Formation de la coalition C_4 pour la tâche de l'agent a_4 où a_4 préfère a_5 et où a_5 est d'accord avec a_4 pour rejoindre le *backbone* de la coalition.

Lorsque le délai d'attente est égal à $\epsilon t * 3$ et que l'agent a_1 n'a toujours pas formé le *backbone* de sa coalition, a_1 calcule de nouveau le vecteur invariant Π de sa tâche pour vérifier et mettre à jour ses informations sur sa tâche. Si la probabilité que sa tâche devienne incontrôlable (Π_2) est supérieure

au Π_2 précédemment calculé, lors de la première demande de formation de la coalition, alors a_1 incrémente la valeur du TTL de son message de demande de formation de sa coalition telle que $TTL=TTL+1=3$ et $et=et * 3$ avant d'envoyer sa nouvelle demande (cf. Figure 2.8).

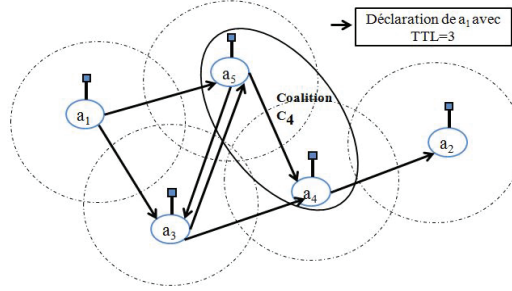


FIGURE 2.8 – Envoi d'un nouveau message de demande formation d'une coalition par a_1 en changeant $TTL=2$ en $TTL=3$ après le calcul du nouveau vecteur invariant de sa tâche.

Comme la préférence de l'agent a_2 , par rapport aux *alliances* proposées, est telle que l'agent $a_4 \succ_{Al} a_1$ et qu'à cette étape a_2 n'est pas un agent *primaire* d'une coalition, mais seulement un *subordonné* de l'agent a_4 , alors a_2 peut envoyer un message d'acceptation de la nouvelle *alliance* $Al_{a_1}^p$ de l'agent a_1 (cf. Figure 2.9). Si a_1 accepte l'accord avec a_2 , il met à jour son *historique* et notifie à a_2 qu'il est maintenant agent *primaire* de la coalition.

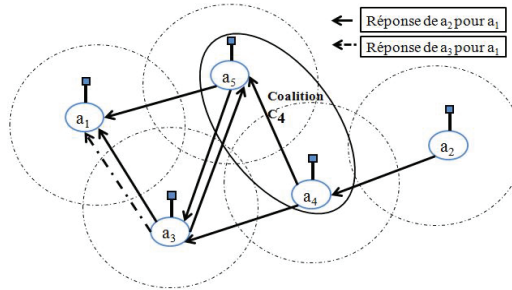


FIGURE 2.9 – Grâce à l'augmentation du TTL , a_2 reçoit et soumet une réponse à la demande de coalition de l'agent a_1 . De plus, a_3 reçoit le message de demande et envoie une réponse à a_1 car il n'est pas dans le *backbone* de la coalition C_4 formée de l'agent a_4 et a_5 .

Après quoi, a_1 évalue une proposition pour que l'agent a_3 devienne son *subordonné*. Comme dans les propriétés d'une tâche il y a la localisation géographique (cf. exemple 2), alors a_2 cherche à être dans le même voisinage de l'agent a_1 et de la tâche T_{c_1} (cf. Figure 2.10). Cependant, a_2 doit rester aussi dans le voisinage de l'agent a_4 car il est son agent *subordonné* sinon a_4 réduit sa fiabilité. La nouvelle configuration des agents représentant une structure de coalition est sous la forme exemplifiée par la Figure 2.11.

Les coalitions C_1 et C_4 sont dans un équilibre de Nash, car les agents *primaires* de chaque coalition ont la même responsabilité de gestion et d'exécution de la tâche de leur coalition. La récompense d'un agent ne peut être maximale que s'il reste dans sa coalition durant au moins le temps d'exé-

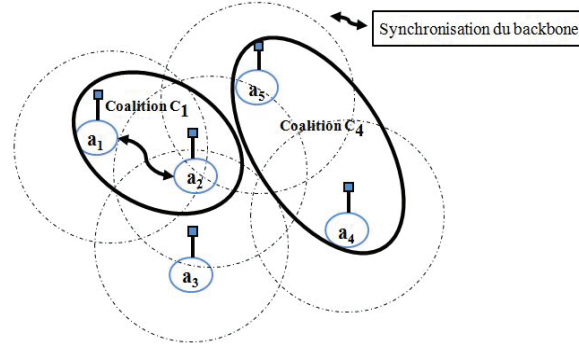


FIGURE 2.10 – a_1 accepte de partager la coalition avec a_2 comme agent primaire. Les deux agents synchronisent leurs connaissances sur la tâche, ce qui amène a_2 et a_1 à être dans le même voisinage. De plus, ils se synchronisent à propos de la stabilité du backbone et de la validation de leur alliance Al_{a_1,a_2} .

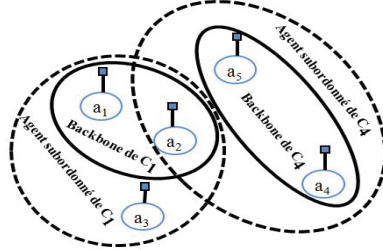


FIGURE 2.11 – Schéma montrant les agents primaires et les agents subordonnés de chaque coalition $C_i : i=1$ ou 4 .

cution stochastique de la tâche $D_{c_i} : i=1$ ou 4 de sa coalition (cf. équation 2.5). À cela s'ajoute le fait que, si un agent *primaire* quitte sa coalition, sa fiabilité est réduite par les autres agents de sa coalition et de manière transparente (sans négociation) son agent *subordonné* va le remplacer et recevoir à sa place la récompense associée à l'exécution de la tâche.

Si la stabilité d'une coalition n'est pas atteinte à cause d'une amplification de sa ou ses tâches ou d'une indisponibilité temporaire des agents impliqués, les agents *primaires* se fondent sur les protocoles de formation de coalitions *persistantes* décrit ci-dessous dont l'utilisation permet d'éviter une restructuration après une renégociation des engagements, d'éviter la perte de la récompense et des *alliances* ou une dissolution de leur coalition.

2.6 RECHERCHE DE LA PERSISTANCE DES COALITIONS

La dynamique qui peut amener les agents à quitter ou rejoindre l'environnement de déploiement, peut être volontaire ou résulter des effets sous-jacents aux dispositifs qui hébergent les agents (par exemple, épuisement des ressources, pannes, etc.). Cette section décrit la méthode utilisée par les agents pour assurer l'exécution des tâches en maintenant la persistance (la stabilité au fil du temps)

des coalitions malgré la dynamique des tâches et leurs disponibilités imprévisibles.

Pour ce faire, lorsqu'un agent *primaire* a_i identifie une mise à jour (par exemple, des ressources insuffisantes, une disparition d'agents, ...) dans sa *coalition* C , une amplification de la tâche ou un dépassement du temps d'exécution stochastique D_c (cf. équation 2.4), a_i doit se coordonner avec les autres agents *primaires* de sa coalition C pour effectuer les actions requises à la maîtrise de la ou des tâches ou à la stabilité de leur coalition C . Dès qu'un agent *primaire* quitte son *backbone* sans le notifier aux autres agents *primaires* de sa coalition ou bien coordonner son désengagement (par exemple, intégrer son agent *subordonné* de manière transparente pour les autres *primaires*), sa fiabilité sera réduite par tous les agents de sa coalition. La fiabilité d'un agent *primaire* peut être aussi abaissée lorsque cet agent refuse de participer à la stabilisation dynamique de sa coalition.

Ces contraintes de participation permettent d'éviter les désengagements et les abstentions volontaires des agents. Ainsi, pour éviter que des buts ne soient pas atteints et les pertes de gain, d'*alliances* ou de fiabilité, chaque agent *primaire* doit participer à la stabilisation de sa coalition surtout s'il est à l'origine de l'instabilité de celle-ci.

Nous détaillons dans la section suivante, notre mécanisme pour la recherche de la persistance des coalitions.

2.6.1 Exploration localisée : DEP (Depth Exploration Protocol)

Ce protocole est utilisé lorsque l'ajout de ressources et la gestion de la tâche peuvent être effectués par les agents du *backbone* ou par intégration d'un ou de plusieurs agents *subordonnés* sans renégocier les accords ou éclater la coalition. Le mécanisme repose sur les *alliances* des agents, leur fiabilité et les *subordonnés* afin de retrouver un état de stabilité des coalitions suite à l'occurrence d'un ou de plusieurs événements empêchant une ou des coalitions de retrouver un état d'équilibre de Nash. L'algorithme 3 décrit les étapes successives de notre protocole d'exploration localisée *DEP*. Ce protocole est dit localisé, car il ne fait intervenir que les agents d'une même coalition en évitant de faire une propagation des informations ne concernant pas les agents extérieurs à cette coalition. Ce modèle est efficace pour privilégier les agents locaux aux coalitions et surtout bénéfique dans le cas où les agents sont déployés dans des réseaux ad-hoc à ressources énergétiques limitées où le contrôle des informations, des communications et de la bande passante sont des critères significatifs.

Les données d'entrée de l'algorithme 3 sont la matrice stochastique P des états courants de la tâche (cf. équation 2.7) et le temps d'exécution stochastique D_c (cf. équation 2.4) calculé lors de la demande de formation de la coalition.

Soit C une coalition où un ensemble d'agents *primaires* détectent que la ou les tâches de leur coalition se sont amplifiées, ou identifient une instabilité de leur *backbone* dû à un désengagement d'un

ensemble d'agents *primaires*. Dans ces cas, les fiabilités des agents *primaires* qui se sont désengagés et qui sont mises en cause, sont réduites au niveau des agents *primaires* qui sont restés dans la coalition C avant de prendre les mesures suivantes :

(1) Calculer la matrice stochastique de leur tâche afin de déduire son vecteur invariant après le calcul de $\Pi * P = \Pi$ où $\sum_{j=0}^2 \Pi_j = 1$. Les étapes de calculs sont détaillées par l'algorithme 1.

Algorithme 3 : DEP (Depth Exploration Protocol).

Require: P et D_c

```

1: Calcul du nouveau vecteur invariant  $\Pi$  de la tâche (cf. algorithme 1)
2: if Compteur temporel  $cpt > D_c$ , ou bien, la nouvelle probabilité  $\Pi_2$  est supérieure à la
   précédente  $\Pi_2$  then
3:    $\forall a_i \in A_c$  chercher à ajouter une ressource si  $R_{a_i}$  est requise dans la coalition ou envoyer un
      message de synchronisation  $Syn\_backbone(R_{a_i}, T_c)$  avec un  $TTL=1$ .
4: end if
5: while  $a_{s_i} \in A_c$  do
6:    $join\_substitute(\Delta_{T_c}, \phi_{T_c}, R_{a_i}, S_{T_c})$ 
7:   repeat
8:     if  $a_{s_i}$  accepte then
9:        $Syn\_backbone(R_{a_i}, T_c)$  et faire la comparaison entre la nouvelle valeur de  $\Pi_2$  et la
         précédente valeur de  $\Pi_2$ 
10:      if ( $\Pi_2$  est inférieure à sa précédente valeur then
11:        arrêter le processus d'exploration par le DEP.
12:      end if
13:    end if
14:  until délai d'attente inférieur à  $\epsilon t$ 
15:   $ressources = R_{min} \setminus \{R_i, \dots, R_k\}$  et exécuter  $W\_Exploration(ressources)$ 
16:  return  $R_{min} \setminus \{R_i, \dots, R_k\}$ 
17: end while

```

(2) Si la nouvelle valeur de la probabilité que la tâche devienne incontrôlable (Π_2) est supérieure au résultat précédemment obtenu pour cette probabilité Π_2 , ou bien, si le temps d'exécution stochastique D_c est épuisé, alors en premier lieu, les agents *primaires* de C vont chercher à corriger eux-mêmes ces insuffisances :

The diagram illustrates the evolution of tasks in a multi-agent system. A central node labeled "Evolution des tâches" is connected to five primary agents (a_1, a_2, a_3, a_4, a_5) via red arrows. These primary agents are further connected to subordinate agents ($a_{s1}, a_{s2}, a_{s3}, a_{s4}, a_{s5}$) via green dotted arrows. The diagram also shows agents without engagement ($a_{s1}, a_{s2}, a_{s3}, a_{s4}, a_{s5}$) and a legend for agent types and signal range.

Legend:

- Agent sans engagement (Agent without engagement)
- Agent primaire (Primary agent)
- Agent subordonné (Subordinate agent)
- Mise à jour du backbone (Backbone update)
- Portée signal de l'équipement (Equipment signal range)

(2-b) Chaque agent *primaire* de C cherche d’abord à augmenter son implication dans C en matière de ressources proposées, s’il a une ou des ressources supplémentaires requises (cf. Figure 2.13).

Donc, si l’instabilité est due à la disparition d’un ensemble d’agents *primaires*, ce message `join_substitute`($\Delta_{T_c}, \varphi_{T_c}, R_{a_i}, S_{T_c}$) informe les agents *subordonnés* sur les ressources et la contribution marginale des agents *primaires* qui ont quitté le *backbone*. Dans ce cas aussi, les messages sont propagés en mode *non-return broadcast* avec un *TTL* égal à 1 pour éviter la profusion des messages en dehors de la coalition concernée. À cela s’ajoute, la nécessité de privilégier la participation des agents *subordonnés*, car leur intégration n’implique pas une nouvelle négociation du fait qu’elle a

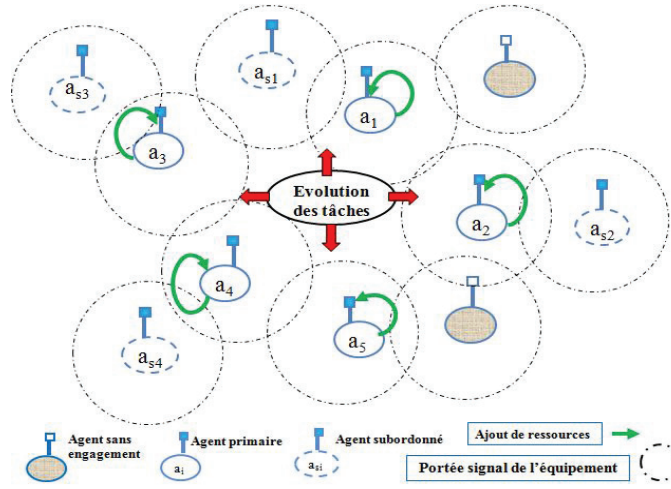


FIGURE 2.13 – Chaque agent primaire a_i , $i \in [1, 5]$ de la coalition essaie d'identifier et d'ajouter les éléments requis pour la coalition en matière de ressources ou de contribution marginale pour assurer la stabilisation de la coalition et pour obtenir une récompense supplémentaire au regard des ressources ajoutées.

été effectuée au moment de la formation de la coalition et qu'une alliance avec son agent *primaire* est déjà établie.

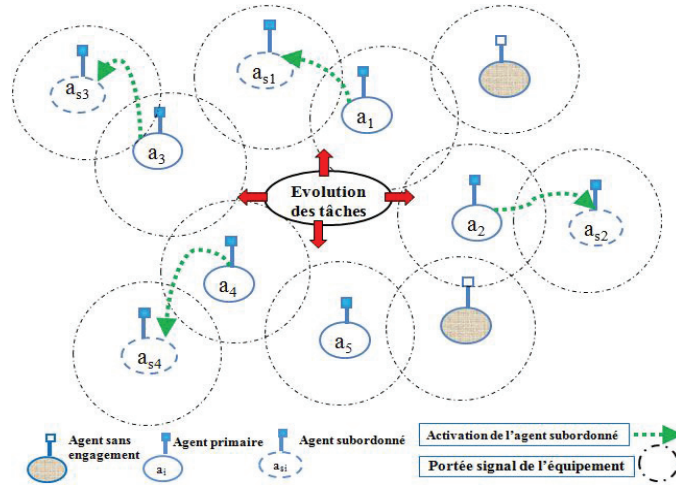


FIGURE 2.14 – Chaque agent primaire a_i , $i \in [1, 5]$ prévient son agent subordonné a_{s_i} , $i \in [1, 4]$ afin de rechercher un autre état de stabilité de leur coalition. Si un agent n'a pas de subordonné (par exemple, a_5) ou des ressources à ajouter, il n'attend que les messages de synchronisation pour prendre connaissance de l'état de stabilité de la coalition et des décisions prises par le backbone.

Un agent *primaire* qui a un agent *subordonné* qui peut participer à la coalition en intégrant le *backbone* avec la ou les ressources manquantes, il en informe les autres agents de son *backbone* en envoyant un message $Syn_backbone(R_{a_i}, T_c)$. Ce message permet une synchronisation des connaissances du *backbone* avant toute intégration de la ou des ressources R_{a_i} d'un agent dans la coalition. Ce

partage d'information maintient la cohérence des actions de la coalition et évite les cas où des ressources sont dupliquées ou bien des agents sont monopolisés inutilement par une coalition. À cela s'ajoute le fait que chaque agent peut déterminer dynamiquement les ressources disponibles et les ressources à rechercher (cf. Figure 2.15).

Grâce à la transitivité des *alliances*, même si un agent *subordonné* n'est pas jugé fiable par certains agents *primaires*, son intégration dans le *backbone* ne va pas entraîner de refus de la part des agents du *backbone*.

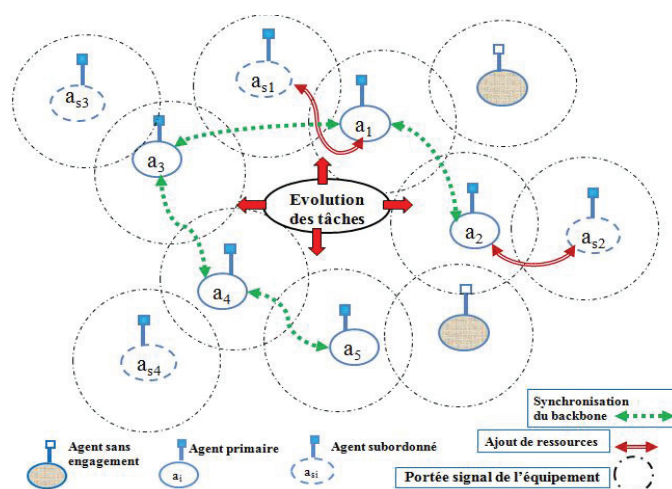


FIGURE 2.15 – À cette étape, les agents subordonnés a_{s1} et a_{s2} des agents a_1 et a_2 ont les ressources requises. Suite à la réponse favorable à l'intégration de ces subordonnés dans le backbone, a_1 et a_2 se synchronisent avec les autres agents primaires de leur coalition. Cette synchronisation informe des ressources et des agents ajoutés pour éviter qu'un des agents ne recherche les mêmes ressources.

Cependant, supposons que l'agent a_i est un agent *primaire* dans C_i et un agent *subordonné* dans C_j . Si avant la fin de l'exécution de la tâche dans sa coalition C_i , sa ou ses ressources sont requises dans la coalition C_j et que son agent *primaire* a_j dans C_j a sollicité sa participation, alors a_i doit satisfaire la demande de l'agent a_j sinon a_j diminuera sa fiabilité. Lorsque l'agent a_i veut maintenir sa fiabilité au prêt de l'agent a_j , l'agent a_i doit trouver un *subordonné* capable de le remplacer au prêt de l'agent a_j ou bien demander à son *subordonné* dans C_i de rejoindre la coalition C_j . L'agent a_i peut aussi quitter sa coalition C_i s'il juge qu'il est préférable de rejoindre la coalition C_j plutôt que de rester dans C_i quand son *subordonné* peut le remplacer dans C_i sans mettre en péril la stabilité de C_i .

Toutefois, l'agent a_i peut aussi quitter C_i sans aucune mesure préalable s'il juge que C_j est plus bénéfique que C_i malgré le fait que sa fiabilité sera réduite par tous les agents de sa coalition C_i et qu'il n'obtiendra pas de récompense suite à sa participation dans C_i .

(4) Chaque agent *primaire* de C calcule la stabilité de sa coalition en faisant une comparaison

entre la nouvelle valeur de la probabilité Π_2 avec son ancienne valeur.

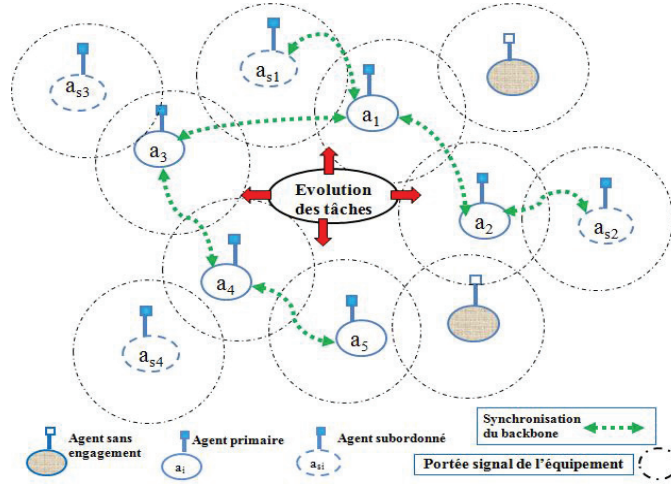


FIGURE 2.16 – Les agents primaires a_1 et a_2 confirment respectivement l'intégration des agents subordonnés a_{s1} et a_{s2} dans la coalition et calculent de nouveau le vecteur invariant de la ou des tâches pour évaluer la stabilité de leur coalition. Les agents a_1 et a_2 synchronisent leurs connaissances sur la ou les tâches et sur la stabilité de leur coalition avec les nouveaux agents intégrés et ceux du backbone.

Si la coalition n'est toujours pas stable à cause de l'absence de ressource(s) ou de *subordonné(s)* capable(s) d'intégrer la coalition, les agents *primaires* de C synchronisent leurs connaissances sur la ou les tâches et la stabilité de leur coalition afin de déduire de là, les ressources requises (cf. Figure 2.16). Après cela, les agents *primaires* initient l'exécution du protocole d'exploration qui va étendre les recherches en dehors de leur coalition. Nous appelons ce protocole d'exploration étendue : WEP (*Width Exploration Protocol* (cf. algorithm 4).

La valeur de sortie $R_{min} \setminus \{R_i, \dots, R_k\}$ de l'algorithme 3 est la valeur d'entrée de l'algorithme 4. R_{min} est l'ensemble minimum des ressources dont la coalition a besoin et $\{R_i, \dots, R_k\}$ est l'ensemble des ressources disponibles dans le *backbone* de la coalition. La fonction $W_Exploration(resources)$ fait appel à l'algorithme 4 du protocole WEP où *resources* a pour valeur $R_{min} \setminus \{R_i, \dots, R_k\}$.

2.6.2 Exploration étendue : WEP (Width Exploration Protocol)

Ce protocole, représenté par l'algorithme 4, est utilisé par les agents *primaires* lorsque la stabilité de leur coalition ne peut plus être garantie en se fondant sur une exploration localisée entre les membres de la coalition. Ils prennent alors contact avec les agents externes à leur coalition. L'exécution de ce protocole mène à des négociations entre les agents *primaires* et leur voisinage. Ces négociations ne concernent pas les engagements qui ont mené à la formation de la coalition initiale. Ce protocole WEP est principalement fondé sur le mécanisme de *recommandation*. En cas d'échec

du *WEP*, la coalition est interrompue si le risque d'échec de la coalition atteint un certain seuil (cf. section 2.7 et proposition 9).

Algorithme 4 : WEP (Width exploration protocol).

Require: $R_{min} \setminus \{R_i, \dots, R_k\}$

- 1: *notify_Backbone*(R_{find}, T_c)
- 2: $R_{find} \subseteq \{R_{min} \setminus \{R_i, \dots, R_k\}\}$
- 3: Calcul de $\Delta_{T_c} \in T_c, \Pi_0, \Pi_1, \Pi_2, Al_{a_i}^P$
- 4: *Declare*($\Pi_2, T_c, R_{find}, Al_{a_i}^P$)
- 5: *negotiate*() pour négocier avec le voisinage suivant le principe de l'algorithme 2.
- 6: **if** ($R_{find} == \emptyset$) **then**
- 7: *find_substitute*(Δ_{T_c}) pour trouver un agent *subordonné*
- 8: **else**
- 9: *Recom*($R_{find}, \Pi_0, \Pi_1, \Pi_2, Al_{a_i}^P$)
- 10: **if** (*SendRecom*(a_j, Al_{a_i, a_j})) **then**
- 11: *AckSendRecom*(a_j, Al_{a_i, a_j})
- 12: **if** *AckRecom*(a_j) et $R_{a_j} \in R_{find}$ **then**
- 13: supprimer R_{a_j} de R_{find} et enregistrer a_j dans l'*historique*
- 14: **end if**
- 15: **end if**
- 16: **if** *SendNewAl*(Al_{a_j, a_i}) et Al_{a_j, a_i} est équitable **then**
- 17: *AckSendNewAl*(Al_{a_j, a_i}) supprimer R_{a_j} de R_{find} et enregistrer a_j dans l'*historique*
- 18: **end if**
- 19: **end if**

(1) Les agents *primaires* se synchronisent grâce au message *notify_Backbone*(R_{find}, T_c) qui leur donne par la même occasion, les informations sur les ressources $R_{find} = R_{min} \setminus \{R_i, \dots, R_k\}$ à rechercher pour la ou les tâches T_c de leur coalition (cf. Figure 2.17).

(2) Chaque agent *primaire* a_i calcule le nouveau vecteur invariant $\Pi = (\Pi_0, \Pi_1, \Pi_2)$ de T_c (cf. algorithme 1). Ce calcul décentralisé du vecteur invariant permet d'éviter une centralisation de la prise de décision et les erreurs de calcul d'un agent qui peuvent parfois entraîner la poursuite de l'exécution du protocole *WEP* au moment où la coalition est déjà stable après l'exécution du protocole *DEP*.

(3) Si un agent a_i relève une grande probabilité que T_c devienne incontrôlable, c'est-à-dire, Π_2 supérieur à Π_0 et Π_1 et de plus que la nouvelle valeur de Π_2 est supérieure à sa précédente valeur, alors a_i poursuit le protocole *WEP*. Il est clair que, si un agent *primaire* poursuit ce protocole, les

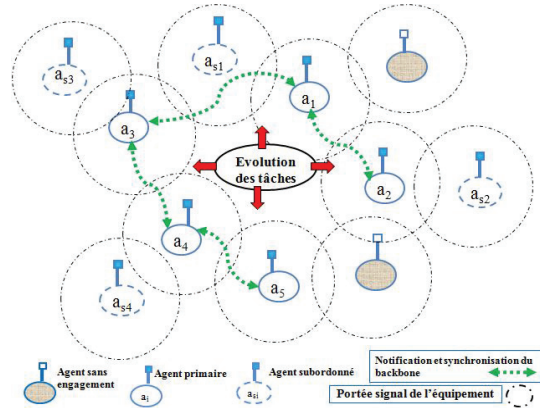


FIGURE 2.17 – Les agents du backbone échangent leurs connaissances sur la ou les tâches.

autres agents du *backbone* feront le même choix sans que ce soit nécessaire de faire une synchronisation, car comme nous le montrons dans la proposition 4, le vecteur invariant obtenu par tous les agents sera unique quelle que soit la distribution de probabilité initiale qu'ils ont utilisée pour la tâche (propriété d'ergodicité du MDP (Ortner 2007), (Baynat 2000)).

(3-a) Chaque agent *primaire* a_i calcule les nouvelles contraintes Δ_{T_c} de T_c .

(3-b) Chaque agent *primaire* a_i détermine $R_{find} \subseteq \{R_{min} \setminus \{R_i, \dots, R_k\}\}$ et l'alliance $Al_{a_i}^P$ qu'il va proposer à son voisinage ϑ_{a_i} , $\{R_i, \dots, R_k\}$ étant les ressources présentes dans la coalition de l'agent a_i .

(4) À cette étape, chaque agent *primaire* utilise le mécanisme de *recommandation* en envoyant dans son voisinage un message $Declare(\Pi_2, T_c, R_{find}, Al_{a_i}^P)$ (cf. Figure 2.18). Dans ce cas, un agent qui n'a aucun lien avec la coalition peut être contacté si c'est un allié d'un agent *primaire* de la coalition.

En privilégiant les alliés dans cette première demande de *recommandation*, nous cherchons à ce que les agents puissent grâce à leurs *alliances* trouver d'autres ressources sans nécessairement entrer dans de longues phases de négociation ou de recherche de ressources.

(5) Si la restructuration de la coalition est évitée, alors chaque agent *primaire* doit, s'il n'a plus d'agent *subordonné*, rechercher un ($find_substitute(\Delta_{T_c})$).

(6) Sinon s'il est encore nécessaire de rechercher un état de stabilité de la coalition, chaque agent *primaire* étend ses recherches en allant négocier avec des agents qui ne seraient pas des alliés. Cette phase de négociation est fondée sur la transitivité des *alliances* et du mécanisme de *recommandation* ($Recom(R_{find}, \Pi_0, \Pi_1, \Pi_2, Al_{a_i}^P)$) avec un $TTL=1$ d'abord. Dans ce cas, le $TTL=1$ évite

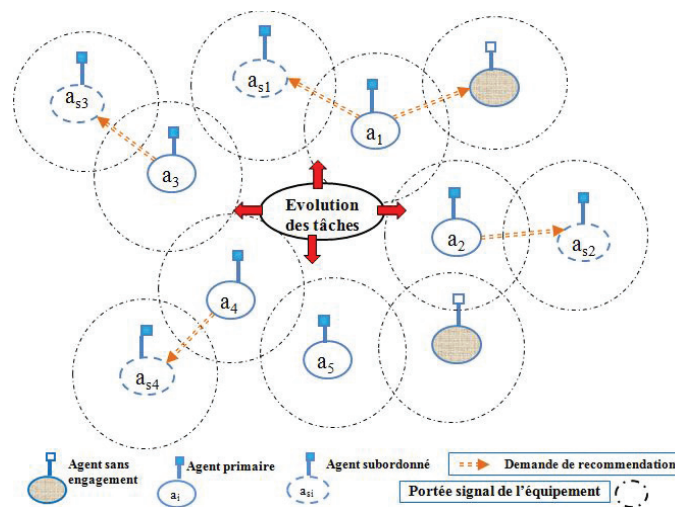


FIGURE 2.18 – Utilisation du mécanisme de recommandation par les agents primaires pour prendre contact avec leurs agents subordonnés et avec les autres agents alliés de leurs voisinages. Seul a_1 envoie une demande de recommandation à un agent allié qui ne participe pas à sa coalition.

de faire une négociation proprement dite où même des agents non voisins peuvent répondre (cf. Figure 2.19) tout en maximisant les chances de trouver un ou des agents susceptibles de rejoindre la coalition ou bien de repérer un ou des alliés qui peuvent utiliser leurs *alliances* pour recommander un ensemble d'agents ayant les ressources requises pour garantir la stabilité de la coalition.

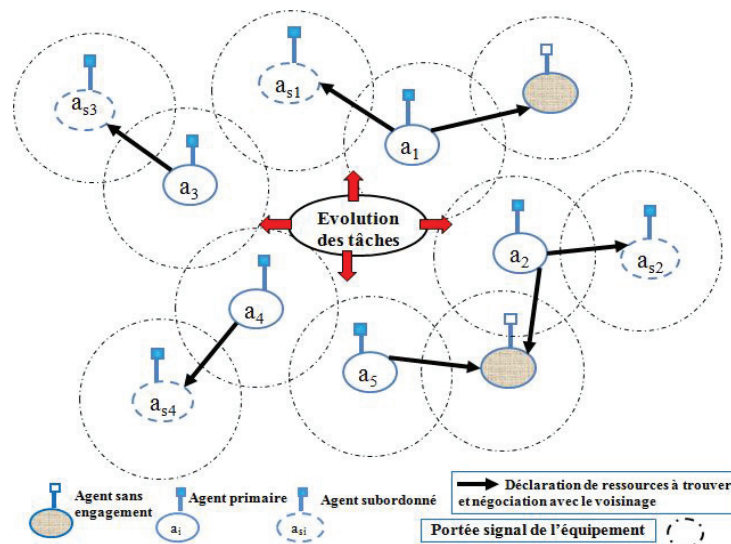


FIGURE 2.19 – Initiation des négociations par chaque agent primaire avec leurs agents voisins qu'ils soient subordonnés, alliés ou non. Ainsi, les agents primaires a_1 , a_2 , a_3 , a_4 et a_5 envoient des messages à tout agent de leur voisinage.

À la suite de cette étape, si un agent *primaire* identifie un ensemble d'agents capables de fournir ces

ressources, l'agent *primaire* se synchronise avec les agents *primaires* de son *backbone* avant de vérifier de nouveau la stabilité de leur coalition.

Si un agent *primaire* a_i n'a reçu qu'un acquittement $SendRecom(a_j, Al_{a_i,a_j})$ pour sa demande de *recommandation* de la part d'un agent a_j , alors l'agent a_i envoie une confirmation $AckSendRecom(a_j, Al_{a_i,a_j})$ à l'agent recommandé a_j si l'alliance Al_{a_i,a_j} est *équitable*. Après quoi, si a_i reçoit une acceptation $AckRecom(a_j)$ de la part de l'agent a_j , alors a_i et a_j mettent à jour leurs *historiques* avant de le notifier aux agents *primaires* de leur *backbone* et de calculer le nouveau vecteur invariant de la ou des tâches de leur coalition.

Si un agent a_j demande de modifier la ou les ressources R_{a_i} ou le temps $T_{help}^{a_i}$ que l'agent a_i propose dans l'alliance $Al_{a_i}^P$, alors a_j doit envoyer un message $SendNewAl(Al_{a_j,a_i})$ pour que l'agent a_i puisse décider d'accepter ou de refuser la nouvelle demande d'alliance Al_{a_j,a_i} de l'agent a_j . Si et seulement si l'alliance est *équitable*, a_i envoie une confirmation $AckSendNewAl(Al_{a_j,a_i})$ à a_j et les deux agents a_i , a_j mettent à jour leur *historique* et en notifient les agents *primaires* de leur *backbone* avant de calculer le nouveau vecteur invariant.

Si après toutes ces tentatives, la coalition n'est toujours pas stable et que $\Pi_0=0$, $\Pi_1 \approx 0$ et $\Pi_2 \approx 1$, et si le niveau seuil du risque d'échec de la coalition est atteint, alors les agents restructurent leur coalition après une nouvelle phase de négociation. Les détails sur le concept du niveau seuil du risque d'échec d'une coalition sont exposés dans la section suivante.

2.6.3 Seuil du risque d'échec d'une coalition

Pour évaluer la persistance d'une coalition après le désengagement d'un ensemble d'agents du *backbone* ou bien suite à une amplification de la(des) tâches de la coalition, les agents *primaires* présents dans cette coalition calculent et synchronisent leurs connaissances à propos du risque de devoir renégocier la coalition (risque d'échec de la coalition). Ce calcul du risque permet aux agents de déterminer s'il est utile de continuer à maintenir leur coalition et s'ils peuvent revenir à un état d'équilibre de Nash dans leur coalition en se fondant sur les protocoles d'exploration *DEP* ou *WEP* pour ainsi éviter la restructuration de leur coalition (cf. section 2.7 proposition 9).

Pour calculer ce risque, les agents utilisent une notion bien connue en théorie des jeux, à savoir la probabilité de ruine. Adopter cette notion de la probabilité de ruine pour la formation de coalitions donne la probabilité d'échec d'une coalition suite à un ensemble d'événements impactant la coalition.

Nous notons par ξ_{C^t} , cette probabilité d'échec d'une coalition C à l'instant t . Cet échec intervient lorsque les ressources R^t de la coalition à l'instant t sont insuffisantes pour accomplir la ou les tâches de la coalition et si la probabilité que la ou les tâches deviennent incontrôlables (Π_2)

est presque égale à 1 (cf. équation 2.8). Les agents continuent de maintenir leur coalition jusqu'à $\Pi_2 \approx 1$, car même si les ressources sont insuffisantes à l'instant t , cela peut ne plus être le cas à l'instant $t + dt$, i.e, des agents désengagés à cause d'une indisponibilité temporaire sont redevenus disponibles ou si la ou les tâches ont diminuées en intensité. D'où l'intérêt d'évaluer le niveau seuil du risque d'échec des coalitions.

Pour y arriver, nous modélisons le processus stochastique des coûts des participations dans une tâche d'un agent comme étant $X_k / k \geq 1$ et l'instant des participations comme $\tau_k / k \geq 1$ (τ_1 est l'instant de la première participation). Nous considérons que, τ_k suit une *distribution Exponentielle* $E(\mu)$ car si le processus qui modélise le nombre de participations dans un intervalle de temps, suit une distribution de *Poisson*, alors le processus qui modélise le temps entre deux participations, suit une distribution *Exponentielle*.

Soit N_t le nombre de participations accomplies par un agent durant un intervalle de temps $[0, t]$. Le processus qui modélise $N_t / t \geq 0$ est un processus de *Poisson* de paramètre λ . L'évolution des ressources d'une coalition peut être exprimée à l'instant t par la fonction suivante :

$$R^t = R^0 + gt - \sum_{k=1}^{N_t} X_k \quad (2.9)$$

où R^0 est le nombre initial de ressources, gt le nombre de ressources ajoutées depuis le début de l'exécution de la ou des tâches. La probabilité d'échec de la coalition s'exprime alors comme suit :

$$\xi_{C^t} = P(\inf_{t \geq 0} R^t \leq 0 / R^0) \quad (2.10)$$

Quand la persistance d'une coalition ne peut pas être assurée à cause du manque de ressources ou d'accords entre les agents ou à cause d'une amplification de la ou des tâches, les agents *primaires* qui doivent restructurer la coalition peuvent déterminer la priorité de leurs tâches afin de savoir s'ils peuvent former de manière séquentielle de nouvelles coalitions pour leurs différentes tâches. L'objectif est d'éviter l'absence de la prise en charge de certaines tâches que nous supposons ici critiques et dynamiques (amplification, régression ou stationnaire).

La priorité d'une tâche d'un agent a_i est : $\pi_{a_i} = 1 - \xi_{C^t}$. Ce qui signifie que plus π_{a_i} est petit, plus la tâche est prioritaire.

2.7 ANALYSE DU MÉCANISME DE FORMATION DE COALITIONS

La proposition 1 justifie l'utilisation de la loi de *Poisson* pour le calcul de la fiabilité des agents.

Proposition 2.1 *Le calcul de la fiabilité est cohérent même si de nouveaux événements aléatoires (pannes, hors de portée, ...) impactent la fiabilité des agents puisque la somme de tous ces événements aléatoires engendre un processus de Poisson de paramètre $\lambda = \sum_{i=1}^{\infty} \lambda_i$ où $\lambda_i : i \in [1, \infty[$ est le paramètre d'un événement aléatoire.*

Démonstration. Soit $Proc_1$ un processus de *Poisson* dont la probabilité que sa variable X prenne la valeur k est : $P[X=k]=e^{-\lambda_1} \frac{\lambda_1^k}{k!}$. Soit $Proc_2$ un processus de *Poisson* dont la probabilité que sa variable Y prenne la valeur k s'écrit : $P[Y=k]=e^{-\lambda_2} \frac{\lambda_2^k}{k!}$.

Si le processus $Proc$ résulte de la somme des deux processus $Proc_1$ et $Proc_2$, alors la probabilité que sa variable Z prenne la valeur k peut être exprimée par $P[Z=k]=P[X+Y=k]$.

$$P[Z=k]=\sum_{i=0}^k P[X=i] * P[Y=k-i].$$

$$P[Z=k]=\sum_{i=0}^k (e^{-\lambda_1} \frac{\lambda_1^i}{i!}) * (e^{-\lambda_2} \frac{\lambda_2^{(k-i)}}{(k-i)!}).$$

$$P[Z=k]=e^{-(\lambda_1+\lambda_2)} \sum_{i=0}^k \frac{\lambda_1^i}{i!} * \frac{\lambda_2^{(k-i)}}{(k-i)!}.$$

$$P[Z=k]=e^{-(\lambda_1+\lambda_2)} \sum_{i=0}^k \frac{\lambda_1^i}{i!} * \frac{\lambda_2^{(k-i)}}{(k-i)!} * \frac{k!}{k!}.$$

$$P[Z=k]=\frac{e^{-(\lambda_1+\lambda_2)}}{k!} \sum_{i=0}^k (\lambda_1^i * \lambda_2^{(k-i)}) \frac{k!}{(i!)*(k-i)!}.$$

Nous savons que $\frac{k!}{(i!)*(k-i)!}=C_k^i$, alors :

$$P[Z=k]=\frac{e^{-(\lambda_1+\lambda_2)}}{k!} \sum_{i=0}^k C_k^i \lambda_1^i * \lambda_2^{(k-i)}. \text{ D'après le théorème du Binôme de Newton : } (a+b)^n = \sum_{j=0}^n C_n^j * a^j * b^{(n-j)}.$$

Alors, $P[Z=k]=e^{-(\lambda_1+\lambda_2)} * \frac{(\lambda_1+\lambda_2)^k}{k!}$. Ainsi, $Proc$ est aussi un processus de *Poisson* de paramètre $\lambda_1+\lambda_2$.

Ce qui prouve notre proposition. $\square$

L'analyse de la représentation de l'*historique* des agents nous amène à la proposition suivante.

Proposition 2.2 *L'historique est borné pour chaque agent a_i qui a établi un ensemble d'alliances.*

Démonstration. Soit X un ensemble d'agents alliés et A un ensemble d'agents. Un agent peut établir au plus une *alliance* avec chaque agent de l'ensemble A . Ainsi, $1 \leq X \leq Card(A)$ où $Card(A)$ est la cardinalité de l'ensemble A . Soit r un ensemble de ressources qu'un allié peut suggérer et R l'ensemble des ressources disponibles dans l'environnement multi-agent. Ainsi, $1 \leq r \leq Card(R)$ où $Card(R)$ est la cardinalité des ressources R . Si $Card(A)$ et $Card(R)$ sont bornés, alors X et r sont aussi bornés. La profondeur du modèle hiérarchique de l'*historique* est aussi bornée et est toujours égale à 3. Ainsi, l'*historique* est toujours borné pour tout agent $a_i \in A$. Ce qui prouve notre proposition. $\square$

De par la multiplicité des *alliances* qui peuvent exister entre les agents, cela peut ne pas suffire si ces *alliances* ne sont pas réutilisables dans certain cas. Ce questionnement sur notre mécanisme nous a conduit à identifier une propriété importante sur les *alliances*. La proposition et le corollaire ci-dessous en font la présentation.

Proposition 2.3 *La transitivité des alliances est vérifiée grâce au mécanisme de recommandation et améliore la formation et la stabilisation des coalitions.*

Démonstration. Si $a_i \succ_{A1} a_j$ et $a_j \succ_{A1} a_k$ existent alors si a_i a besoin des ressources de l'agent a_k alors a_j va recommander a_k avec une fiabilité $\rho_{a_k} \geq 0.5$ à cause de Al_{a_i,a_j} et du fait que l'agent a_j , étant

égoïste, il cherchera à éviter que sa fiabilité soit décrémentée par a_i . Ainsi, la préférence $a_i \succ_{AI} a_k$ sera vérifiée. Ce qui prouve notre proposition. $\square$

Corollaire 2.1 Soit $A = \{a_1, a_2, \dots, a_k\}$ un ensemble d'agents et $\Omega = \{C_1, C_2, \dots, C_k\}$ un ensemble de coalitions jugées utiles $C_i : i \in [1, |\Omega|]$ formées au cours du temps. Tout agent a_i qui a eu à participer dans au moins une de ces coalitions s'est doté d'un ensemble d'alliés avec qui il peut coopérer pour l'exécution d'une tâche sans faire de négociations supplémentaires.

La proposition suivante, justifie pourquoi nous avons adapté les chaînes de Markov à notre mécanisme de formation de coalitions.

Proposition 2.4 La chaîne de Markov liée à une tâche T est toujours ergodique et sans nécessité de synchronisation, tous les agents impliqués dans le même processus de formation de coalitions pour T ont la même connaissance à propos de la stabilité de la coalition qu'ils peuvent espérer former.

Démonstration. D'après notre modélisation d'une tâche T ,

(1) $\forall s^i$ et $s^j : i, j \in [0, 2], i \neq j, \pi_{s^i, s^j}^T \geq 0$ et $\sum_{j=0}^2 \pi_{s^i, s^j}^T = 1$. Donc la matrice P de la chaîne de Markov sous-jacente est stochastique car les éléments de la matrice sont non négatifs et la somme de tous les éléments d'une ligne égale à 1.

(2) Une propriété importante des MDP dit que, si la matrice P est stochastique, alors toute puissance $P^{(m)}, m \geq 0$ de P est aussi stochastique.

(3) $\forall s^i$ et $s^j : i, j \in [0, 2], i \neq j, \exists n \geq 0, m \geq 0 : (\pi_{s^i, s^j}^T)^n > 0$ et $(\pi_{s^j, s^i}^T)^m > 0$, donc s^i et s^j communiquent. De plus, d'après (1) et (2) ci-dessus, quel que soit le nombre n ou m , la matrice reste stochastique, alors $\forall s^i, \pi_{s^i, s^i}^T \neq 1$ (pas d'état absorbant). De là, nous déduisons que, le graphe représentatif de la chaîne de Markov est fortement connexe et est formé d'une seule classe.

(4) Comme $\forall s^i, \pi_{s^i, s^i}^T > 0$, alors les états s^i sont apériodiques. Nous déduisons que la chaîne est alors apériodique car si un état est apériodique alors tous les états de la même classe de la chaîne de Markov sont apériodiques.

Des propriétés (3) et (4) ci-dessus, nous pouvons dire que la chaîne de Markov représentative de la tâche T est ergodique.

Une chaîne de Markov possède toujours un vecteur invariant Π . Si la chaîne de Markov est ergodique elle admet une distribution asymptotique (vecteur invariant), unique et indépendante de la distribution initiale, c'est-à-dire pour toute distribution initiale $\Pi^{(0)}$ des états, alors, $\lim_{n \rightarrow \infty} \Pi^{(0)} P^{(n)} = \Pi$.

Ainsi, quelle que soit la distribution de probabilité initiale des états de la tâche T qui est donnée par sa matrice P , le vecteur invariant est toujours le même. Donc le calcul effectué par deux agents distincts pour la même tâche T donne toujours le même vecteur invariant comme résultat. De là, nous déduisons que les agents vont avoir la même connaissance à propos de cette tâche T et sur

la coalition à former pour T sans avoir besoin de s'échanger des messages. Ce qui prouve notre théorème. $\square$

Du fait que la fiabilité est déterminée par des agents autonomes, égoïstes, de disponibilité aléatoire et sont déployés dans des environnements à ressources énergétiques limitées, des erreurs de calcul ou des stratégies pour réduire la fiabilité des agents peuvent néanmoins exister si notre mécanisme ne tient pas compte de cela. La proposition ci-dessous donne une réponse à ces interrogations.

Proposition 2.5 *Le mécanisme de recommandation permet à un agent de corriger un jugement erroné sur sa fiabilité.*

Démonstration. Considérons trois agents a_i, a_j, a_k et deux alliances Al_{a_i, a_j} et Al_{a_j, a_k} où a_i et a_j participent à la coalition C . Supposons que, a_k considère que l'agent a_i n'est pas fiable. Si a_j utilise son alliance avec a_k pour former ou stabiliser C alors il existera une alliance Al_{a_i, a_k} . Si a_i et a_k restent dans C jusqu'à la fin de l'exécution de la tâche, l'alliance, Al_{a_i, a_k} sera validée par a_i et a_k . Ainsi, a_k recalculera la fiabilité de l'agent a_i et enregistrera a_i dans son Hs_{a_k} . Cela prouve notre proposition. $\square$

Les propositions 6 et 7 ainsi que le corollaire 2, prouvent l'existence de la propriété de stabilité recherchée dans ce mécanisme.

Proposition 2.6 *Une coalition C est dans un équilibre de Nash, si chaque agent primaire de C a une alliance non-dominée avec au moins un agent de la coalition C .*

Démonstration. Si a_i de C a une alliance non-dominée avec au moins un a_j de C alors $a_i \succ_{Al} a_j$ existe pour a_i . Dans ce cas, a_i n'aura pas l'intention de quitter C tant que l'agent a_j est dans C par soucis que ce dernier ne réduise sa fiabilité. De plus, a_i ne voudrait pas perdre la possibilité d'avoir davantage d'alliances avec les autres agents de la coalition C en utilisant la transitivité des alliances (cf. proposition 3) et la récompense associée à sa participation à la coalition.

Nous déduisons que si un agent primaire a_i a une alliance non-dominée avec au moins un agent de sa coalition, il n'aura pas l'intention de la quitter. Au contraire, il cherchera à préserver et à améliorer ses relations avec les autres agents de sa coalition. Ainsi, la coalition sera dans un équilibre de Nash. Ce qui prouve notre proposition. $\square$

Corollaire 2.2 *La recommandation permet au processus de formation de coalitions de converger et de former des coalitions qui sont dans un équilibre de Nash.*

Proposition 2.7 *Une coalition stable qui est dans un équilibre de Nash, reste toujours dans cet état de stabilité si à chaque fois qu'une ou des ressources sont requises pour la coalition, il y a un agent primaire de cette coalition qui a une alliance non-dominée avec au moins un agent capable de fournir le ou les ressources requises.*

Démonstration. Soit a_i un agent primaire d'une coalition C_1 qui est dans un équilibre de Nash. Alors, a_i a au moins une alliance non-dominée avec un agent a_j de C_1 . Si a_i a un allié a_k dans sa vue qui est disponible et avec qui il a une alliance non-dominée et qui peut stabiliser la coalition, grâce à la transitivité des alliances (proposition 3), alors l'allié a_k va intégrer la coalition pour garder le niveau de préférence de cette alliance avec a_i . Ce qui montre que la coalition sera encore dans un équilibre de Nash. Ce qui prouve notre proposition. $\square$

La proposition qui suit, analyse nos deux protocoles d'exploration afin de dégager l'apport qu'ils délivrent à notre mécanisme de formation de coalitions.

Proposition 2.8 *La persistance des coalitions est rendue possible grâce aux protocoles DEP et WEP qui permettent à une coalition de revenir à chaque instabilité vers un état d'équilibre de Nash en utilisant les alliances, la transitivité des alliances et le mécanisme de recommandation des agents.*

Démonstration. Soit $A = \{a_1, a_2, a_3, a_4, a_5\}$ un ensemble d'agents où a_1 est l'agent initiateur d'une demande de coalition. Au début de la négociation pour la formation de la coalition demandée par a_1 , il n'y a aucune alliance entre les agents, c'est-à-dire, $\nexists Al_{a_i, a_j} \ i \neq j \ \forall \ i, j \in [1, 5]$. Si l'ensemble des agents $a_j \in A, j \in [2, 5]$ négocient avec a_1 pour atteindre leur but, établir des alliances Al_{a_1, a_j} et obtenir la récompense *reward* associée à l'exécution de la tâche de la coalition, alors, si un agent a_j accepte de participer à la coalition, il doit accepter d'établir aussi l'alliance Al_{a_1, a_j} et chacun des agents a_1 et a_j va mettre à jour son *historique* en mettant la valeur de la fiabilité de l'autre agent comment étant maximale vu que les agents sont encore à leur première interaction. Ainsi, chaque alliance Al_{a_1, a_j} sera non-dominée et si une coalition $C_1 = \{a_1, a_2, a_3, a_4, a_5\}$ est formée, cette coalition sera stable c'est-à-dire dans un équilibre de Nash comme nous l'avons prouvé par la proposition 6. Considérons cette même coalition $C_1 = \{a_1, a_2, a_3, a_4, a_5\}$ où seul a_1 possède une alliance non-dominée avec tous les autres agents $a_j / j \in [2, 5]$ de C_1 , alors, grâce à a_1 , la coalition est dans un équilibre de Nash. Si a_1 quitte la coalition C_1 , alors C_1 sera instable pour un instant mais grâce à l'utilisation des protocoles DEP et WEP par le reste des agents $a_j / j \in [2, 5]$ en réaction à cette instabilité de C_1 , ces derniers vont essayer de remplacer a_1 en cherchant dans leur voisinage, l'agent qui est le plus fiable et le plus stable (probabilité de stabilité plus grande). Cette recherche est effectuée sur la base des informations contenues dans leur *historique*. Lorsque l'historique ne le permet pas, la recherche s'étend par recommandation.

L'ensemble des agents qui ont tendance à vouloir quitter une coalition sont ceux qui n'ont pas confiance en un ensemble d'agents membres de leur coalition et qui n'ont pas d'*alliances* à préserver. Cependant, l'utilisation des protocoles *DEP* et *WEP* permet de remplacer des agents qui ont quitté leur coalition par d'autres agents de fiabilité et probabilité de stabilité maximale ayant accepté l'établissement d'une alliance, ce qui mène encore la coalition vers un autre état d'équilibre de Nash.

Considérons une nouvelle coalition $C'_1 = C_1 \setminus a_1 \cup a'_1 = \{a'_1, a_2, a_3, a_4, a_5\}$ où a'_1 est l'agent qui a rejoint la coalition C_1 après le départ de l'agent a_1 et où il existe une *alliance* $Al_{a'_1, a_2}$ entre a'_1 et a_2 mais pas d'*alliance* avec a_3, a_4 et a_5 .

Si a_3, a_4 et a_5 quittent la coalition C'_1 , les agents a'_1 et a_2 utiliseraient les protocoles *DEP* et *WEP* pour trouver des agents fiables de probabilité de stabilité maximale qui se trouvent dans leur *historique* ou bien établiraient de nouvelles *alliances* avec des agents de leur voisinage qui peuvent rejoindre C'_1 sans forcément la restructurer.

De plus, en cas de difficultés pour la coalition C'_1 à réaliser ses actions en raison d'une amplification de la tâche de la coalition ou bien à cause de la disparition de certains agents de la coalition, cela aurait mené à l'exécution des protocoles de recherche de persistance *DEP* et *WEP* pour trouver un autre état de stabilité de cette coalition.

Ainsi, à chaque événement qui mène à l'instabilité de leur coalition, les agents qui restent dans la coalition chercheraient toujours à intégrer des agents fiables et stables avec qui une *alliance non-dominée* pourrait être établie. Ce qui mènerait donc, dans tous les cas, la coalition à converger vers un équilibre de Nash. $\square$

Corollaire 2.3 *Chaque fois qu'un ensemble d'agents primaires qui ont des subordonnés disponibles quittent leur coalition, cette coalition converge toujours vers un état d'équilibre de Nash.*

La proposition ci-dessous, revient sur la nécessité et l'intérêt de connaître le niveau seuil du risque d'échec des coalitions.

Proposition 2.9 *Pour assurer la persistance d'une coalition et éviter l'échec de celle-ci, les agents doivent maintenir le coût de la ou des tâches en termes de ressources utilisées par unité de temps en dessous d'un seuil critique p connu par tous les agents de la coalition.*

Démonstration. Considérons la variable $(\gamma)_{k \geq 0} = (\hbar_k - \hbar_{k-1})_{k \geq 0}$ qui modélise l'intervalle de temps entre deux participations dans une tâche. Cette variable est celle d'une distribution *Exponentielle* $E(\mu)$. Soit α le coût moyen des tâches en termes de ressources utilisées par unité de temps. Si l'échec dans l'exécution d'une tâche qui peut entraîner l'échec de la coalition ne peut pas arriver avant l'instant $\hbar_n$, alors nous avons seulement besoin de considérer la séquence des variables aléatoires

notées $\bar{S}_n$ qui est le cumul des coûts des participations où $\bar{S}_0=0$ et $\forall n \geq 1, \bar{S}_n=\sum_{k=1}^n [X_k - (p * h_n)]$

(a) Si $p < \alpha$, alors $\lim_{n \rightarrow +\infty} \bar{S}_n = +\infty \forall R^0 \geq 0, \xi_{C^t}=1$.

(b) Si $p=\alpha$, alors $\lim_{n \rightarrow +\infty} \inf \bar{S}_n = -\infty$ et $\lim_{n \rightarrow +\infty} \sup \bar{S}_n = +\infty \forall R^0 \geq 0, \xi_{C^t}=1$.

(c) Si $p > \alpha$, alors $\lim_{n \rightarrow +\infty} \bar{S}_n = -\infty \forall R^0 \geq 0, \xi_{C^t} \leq 1$.

En effet, si $p < \alpha$, ce résultat découle de la *loi des grands nombres*. Si $p=\alpha$, nous remarquons que $\bar{S}_n$ est la somme de n variables aléatoires $[X_k - (p * h_n)]$ identiques et centrées où $1 \leq k \leq n$. Si $p > \alpha$, nous avons $\lim_{n \rightarrow +\infty} \bar{S}_n = -\infty$. $\square$

Du fait des contraintes de participation dans une coalition en tant qu'agent primaire ou subordonné, s'il existe une coalition C instable dans une structure de coalition, cette instabilité n'affecte pas nécessairement les autres coalitions. Cette remarque s'explique par le fait que :

Si C et C' existent, alors $\forall a_i \in A_C$ et $\forall a_j \in A_{C'}$, a_i et a_j ne sont pas motivés à quitter leur coalition qui est dans un équilibre de Nash.

Si C devient instable du fait que l'agent a_i à quitter le backbone de C , ou bien, à cause d'une amplification de la tâche T_C , alors cette instabilité de C est gérée par les agents du backbone de C grâce aux protocoles *DEP* et *WEP*, sans nécessiter le désengagement des agents n'appartenant pas à C .

De plus, $\forall a_j \in A_{C'}$, la seule chose que l'instabilité de C peut entraîner au pire des cas, c'est la recherche d'un autre agent subordonné. Cette recherche de subordonné est locale à l'agent qui l'effectue. D'où le corollaire ci-dessous.

Corollaire 2.4 *Dans une structure de coalitions, l'instabilité d'une des coalitions n'affecte pas la stabilité des autres coalitions même si celles-ci partagent des agents subordonnés avec la coalition instable.*

Si deux coalitions C et C' deviennent instables, alors d'après le protocole *WEP* :

- les agents de C savent ce qui est requis pour stabiliser C' - les agents de C' savent aussi comment stabiliser C . De là, $\forall a_i \in A_C$ et $\forall a_j \in A_{C'}$, il a la possibilité de migrer vers l'une des coalitions si cela lui permet de stabiliser la coalition de destination et de retrouver une récompense que sa coalition d'origine ne peut plus lui fournir.

Si C et C' partagent des agents (par exemple, un agent primaire de C est un agent subordonné de C' , etc.), alors s'ils sont tous les deux instables, ils peuvent s'échanger des agents de manière transparente si cela permet de stabiliser l'une des coalitions.

Ainsi, s'il y a des coalitions instables, les agents de ces coalitions savent les coalitions capables d'être stabilisées et ils peuvent faire le choix de les stabiliser même si c'est au détriment de leur coalition d'origine qui ne peut plus leur fournir les alliances et les récompenses nécessaires au maintien d'un état d'équilibre de Nash. D'où le corollaire ci-dessous.

Corollaire 2.5 *Si une ou plusieurs coalitions ont atteint leur seuil du risque d'échec, certains échecs de coalitions peuvent être évités.*

2.8 ÉVALUATIONS DE PERFORMANCES

Nous avons effectué un ensemble d'évaluations analytiques et expérimentales de notre mécanisme de formation de coalitions. Ceux-ci, afin d'approfondir l'étude de notre mécanisme en plus de l'analyse théorique proposée dans la section précédente et pour introduire une comparaison avec d'autres mécanismes de formation de coalitions. Dans l'évaluation analytique, nous montrons que notre mécanisme permet la formation de coalitions stables et qui continuent à l'être malgré les changements d'état des tâches (amplification ou régression en intensité) ou la dynamique des agents (aléatoirement disponible ou indisponible). Pour mener cette étude expérimentale, nous avons développé un nouveau simulateur conçu pour tenir compte du contexte dynamique des agents et des tâches. Les simulations nous permettent de mettre en exergue les propriétés de nos coalitions persistantes dans un contexte plus réaliste que l'évaluation analytique.

2.8.1 Évaluation analytique de notre mécanisme

Les agents sont déployés dans des composants électroniques. D'après la loi de Weibull (Guerin et al. 2001)(Hribar et Duka 2010), une méthode qui reconfigure une structure électronique après chaque indisponibilité d'un de ses composants électroniques qui sont dépendants ou non, est dans un contexte de composants non-remplaçables (contexte série), de fiabilité $R1(t) = \exp - [(\frac{\lambda_1 t}{\beta}) * (t - \gamma)]$. Donc, nous pouvons déduire que les méthodes de formation de coalitions qui ne gèrent pas l'indisponibilité des agents et qui prévoient une restructuration des coalitions à chaque indisponibilité d'un agent comme dans (Shehory et Kraus 1996)(Zick et al. 2012)(Chalkiadakis et al. 2010) sont aussi dans un contexte série.

Cependant, d'après la loi de Weibull, une méthode comme la nôtre qui tolère et qui propose des stratégies de substitution des composants indisponibles sans nécessité de reconfigurer leur structure électronique est dans un contexte de composants remplaçables (contexte parallèle), de fiabilité $R2(t) = \exp - [(\frac{\lambda_2 t}{\beta}) * (t - \gamma)]$.

Ici, $\lambda_1 = \frac{C(\omega t)}{N_s * \omega t}$ et $\lambda_2 = \frac{C(\omega t)}{N_0 * \omega t}$ où $C(\omega t)$ est le nombre de composants indisponibles durant l'intervalle de temps ωt , N_0 est le nombre initial de composants électroniques (agents) et N_s est le nombre de composants électroniques (agents) en état de marche après l'intervalle de temps ωt . Pour faire la comparaison entre la fiabilité $R1$ et $R2$, nous avons mené un ensemble de simulations analytiques pour $R1$ et $R2$. Ainsi, les paramètres β , γ et η sont les mêmes pour les simulations avec $R1$ et $R2$.

Ici, $0 \leq \beta \leq 4$ caractérise les causes d'indisponibilité des composants (par exemple, manque de ressources), $-\infty \leq \gamma \leq +\infty$ est le paramètre de position qui indique si les composants utilisés sont neufs ou pas, et η est le paramètre d'échelle, c'est-à-dire la période durant laquelle on étudie la structure électronique (par exemple, 2 heures).

Les résultats des simulations sont discutés dans les figures ci-dessous. Pour ce faire, nous considérons à l'état initial que $N_0=100$. De plus, que les causes d'indisponibilité des composants (agents) sont aléatoires et inconnues ($\beta=1$), que les composants sont neufs ($\gamma=0$) et que cette étude est menée sur une période d'une heure ($\eta=1$). Les résultats de la Figure 2.20 montrent que dans un contexte où les composants électroniques hébergeant les agents peuvent aléatoirement devenir indisponibles, notre méthode fournit une fiabilité plus grande que les méthodes actuelles ($R2 > R1$).

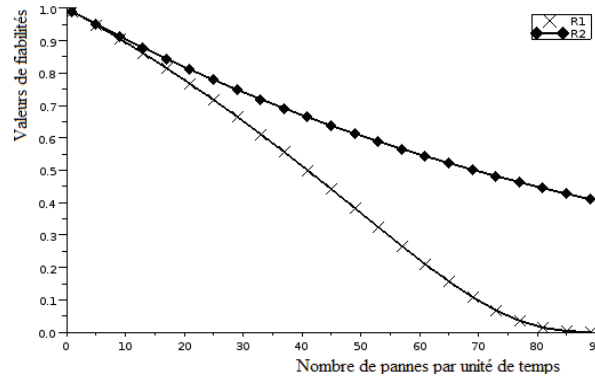


FIGURE 2.20 – Fiabilité d'exécution des tâches en fonction du taux de défaillance des composants hébergeant les agents.

Dans un autre ensemble de simulations analytiques où nous maintenons $N_0=100$, $\gamma=0$, $\eta=1$, nous cherchons à étudier la fiabilité des agents avec notre mécanisme lorsque la cause de l'indisponibilité des agents (β) est aléatoire et suit une des distributions de probabilité très souvent utilisées à savoir : la distribution de probabilité Normale ($\text{panne} \sim N(\mu, \sigma^2)$ où $\mu=0.5$ et $\sigma=0.1$), la distribution de probabilité Uniforme ($\text{panne} \sim U(a, b)$ où $a=0$ et $b=10$), la distribution de probabilité de Poisson ($\text{panne} \sim P(\lambda)$ où $\lambda=0.5$) ou la distribution de probabilité Brownienne $((B_t)_{t>0})$ où t est le paramètre temps).

En évaluant les pourcentages des différents paramètres ($R1$, $R2$ et Pannes) en fonction des états de simulations, les Figures 2.21, 2.23, 2.22 et 2.24, montrent que même si l'indisponibilité des agents suit des distributions de probabilité bien connues, notre mécanisme de formation de coalitions reste toujours plus fiable à produire des coalitions stables.

Ainsi, ces résultats issus des simulations analytiques prouvent que dans le cas où des composants électroniques hébergeant des agents ont des indisponibilités aléatoires, notre mécanisme de formation de coalitions stables ($R2$) donne une meilleure fiabilité que tous les mécanismes qui ne tiennent pas compte de ces indisponibilités et qui restructurent les coalitions à chacune de leur occurrence.

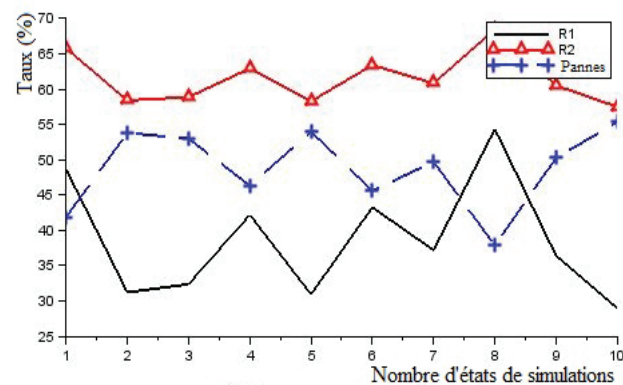


FIGURE 2.21 – Comparaison entre R1 et R2 lorsque au fil du temps l'indisponibilité des agents (panne) suit une distribution de probabilité Normale.

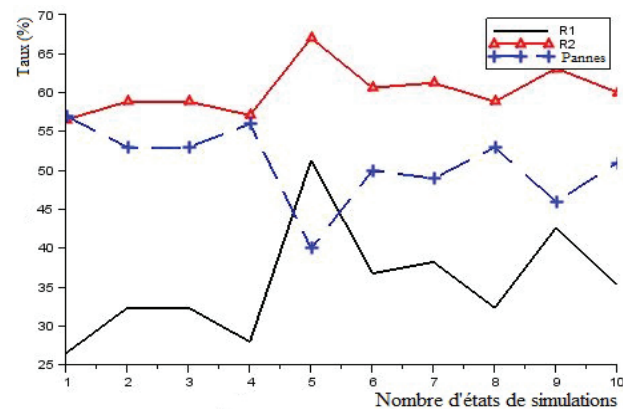


FIGURE 2.22 – Comparaison entre R1 et R2 lorsque au fil du temps l'indisponibilité des agents (panne) suit une distribution de probabilité de Poisson.

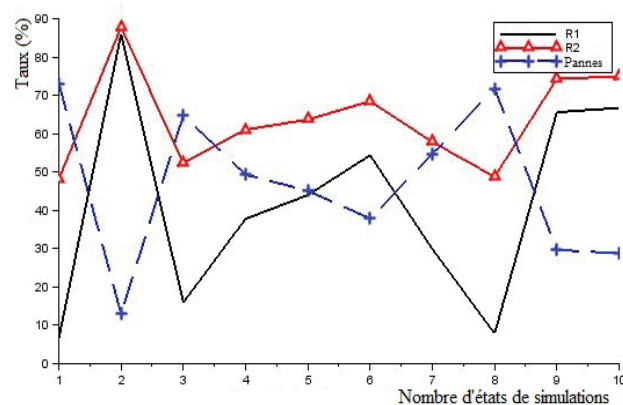


FIGURE 2.23 – Comparaison entre R1 et R2 lorsque au fil du temps l'indisponibilité des agents (panne) suit une distribution de probabilité Uniforme.

Dans la section suivante, nous effectuons un ensemble de simulations expérimentales pour mettre

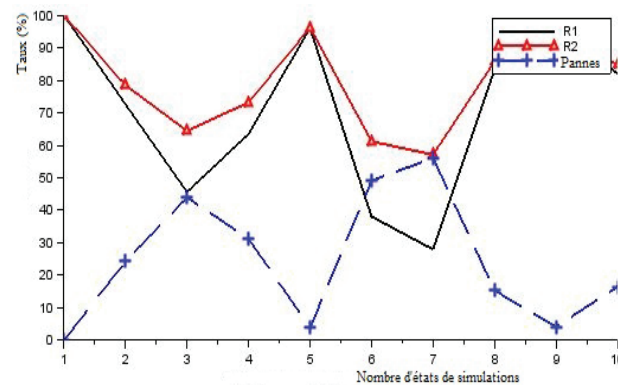


FIGURE 2.24 – Comparaison entre R1 et R2 lorsque au fil du temps l'indisponibilité des agents (panne) suit une distribution de probabilité Brownienne.

en exergue la capacité de notre mécanisme à produire des coalitions persistantes dans un état d'équilibre de Nash.

2.8.2 Évaluation expérimentation de notre mécanisme

Nous supposons que les agents sont intégrés dans des composants électroniques (Smartphones, robots, capteur, ...) autonomes qui sont déployés dans des zones sinistrées. Nous considérons une ville où une coordination est nécessaire pour les évacuations de personnes et une extinction des incendies. Un agent est déployé dans chacun des composants électroniques mis à la disposition des intervenants. L'état de chaque tâche (amplification, régression, stationnaire) a une évolution non-déterministe même si celle-ci est prise en charge par les secours. Chaque tâche est assignée à un agent qui doit utiliser ses stratégies afin d'arriver à former une coalition pour permettre son exécution. À titre d'exemple, une tâche de prise en charge d'un immeuble en feux peut avoir comme sous-tâches : - la recherche de survivants, - l'évacuation des survivants, - l'extinction des différents foyers du feu, - la mise en place d'un cordon de sécurité, etc.

Une stratégie d'un agent peut être : (1) l'utilisation d'*alliances* précédemment établies, (2) le calcul et l'utilisation de l'information sur le risque que la tâche devienne incontrôlable, (3) la négociation du délai de participation avant l'établissement d'une *alliance* sachant le temps d'exécution stochastique de la tâche (cf. équation 2.4), (4) l'utilisation d'une demande de *recommandation*.

Durant les négociations pour la formation d'une coalition, chaque agent qui devient agent *primaire* de la coalition doit avoir au moins une ressource dont la coalition à besoin, être responsable de la tâche ou sous-tâche dans laquelle il participe, trouver un agent subordonné et des ressources en cas de nécessité de stabilisation de leur coalition. Cela est identique à ce qui se passe dans un contexte réel d'assistance en situation de crise, où il y a un ensemble d'intervenants (pompiers, médecins, etc.) qui doivent se déployer dans une zone sinistrée, coordonner leurs actions et leurs informa-

tions, gérer les tâches qui leur sont assignées, etc. tout cela de manière décentralisée sans contrôle centralisé entre les intervenants. De plus, suivant le contexte dans lequel se trouve l'intervenant à un instant donné, il doit décider dans quel groupe (coalition) participer afin d'être le plus efficace, avec qui se synchroniser et échanger des informations, etc. en considérant bien sûr ses contraintes de ressources et les contraintes de temps des tâches.

2.8.2.1 Méthodologies

Nous avons développé notre propre simulateur sous Java vu que les travaux actuels du domaine ne tiennent pas compte du contexte dynamique des agents et des tâches que nous considérons. Dans nos simulations, les agents sont autonomes et égoïstes. Aléatoirement, un agent peut initier une demande de coalitions, il peut aussi participer ou non à une tâche. La localisation et les ressources requises pour une tâche sont aussi générées de la même manière. Le nombre d'agents dans l'environnement est choisi entre 10 et 100 et le nombre de tâches simultanées dans l'environnement est compris entre 3 et 8 tâches. Les propriétés de chaque tâche sont générées aléatoirement. Les simulations sont effectuées sur une machine Intel i7 (4 GHZ) avec 4GB de RAM.

Un agent qui a une tâche pour laquelle il est nécessaire de former une coalition va initier une négociation décentralisée en utilisant ses stratégies, en tenant compte de ses contraintes et de celles de la tâche, tout en cherchant à maximiser sa récompense. Il en est de même pour les agents qui participent dans les négociations et la coalition.

Les négociations et les coalitions sont gérées de manière décentralisée pour que la stabilité des coalitions soit moins difficile à atteindre et à maintenir. Ainsi, un agent *primaire*, pour des besoins de stabilité, peut négocier séparément avec sa vue. Cependant, les agents *primaires* qui sont dans la même coalition doivent synchroniser leurs connaissances à propos de la stabilité de leur coalition et de l'exécution de la ou des tâches de cette coalition.

Par ailleurs, les négociations sont soumises à des contraintes temporelles. Les récompenses et les *alliances* sont modélisées par des fonctions qui dépendent du temps. Au cours du processus de formation des coalitions ou bien une fois que les coalitions sont formées, les ressources disponibles évoluent dynamiquement en fonction de la disponibilité des agents et de l'exécution des tâches. Aléatoirement, nous désactivons un ensemble d'agents de ces coalitions (agents du *backbone* ou des agents *subordonnés*) pour simuler l'indisponibilité des agents, des ressources et de la topologie réseau des composants électroniques.

Les messages sont librement échangés entre les agents en respectant le mode *non-return broadcast* et le *TTL* (Time To Live) des messages. Notons que, même s'il y a suffisamment de ressources dans l'environnement de déploiement des agents, leurs localisations et leurs disponibilités ne sont pas connues à l'avance. À cela s'ajoute, l'impossibilité de savoir à l'avance si les agents seront favorables à une coalition ou pas, car les agents sont autonomes et égoïstes avec des disponibilités aléatoires.

Les scénarios expérimentaux sont répétés avant chaque simulation ou ensembles de simulations en fonction des propriétés du mécanisme de formation de coalitions que nous voulons examiner.

2.8.2.2 Résultats

La figure 2.25 montre le pourcentage de coalitions formées et de coalitions stables malgré la dynamique des tâches et la disponibilité aléatoire des agents (pannes) qui suit la distribution de *Poisson* ($panne \sim P(0.5)$). Nous observons que tant qu'il y a suffisamment de ressources (40 agents), le taux de stabilisation des coalitions est toujours supérieur à 70 %. Cependant, s'il n'y a pas suffisamment de ressources pour l'ensemble des coalitions (< 20 agents), 50 % des coalitions seront néanmoins stabilisées.

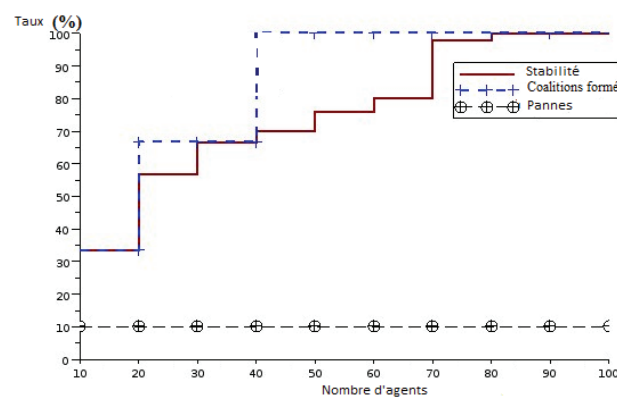


FIGURE 2.25 – Pourcentage de stabilisation des coalitions sachant que l'indisponibilité aléatoire des agents suit une distribution de Poisson $P(0.5)$.

Dans la suite, nous présentons le taux de stabilisation d'un ensemble de coalitions (Figure 2.26), c'est-à-dire la probabilité d'avoir, $\Phi_i/i \in [1, 3]$, coalitions stables en fonction des *Ratios* lorsque l'indisponibilité aléatoire (panne) des agents suit une distribution Normale, $panne \sim N(0.5, (0.1)^2)$. Le *Ratio* est égal au nombre de ressources disponibles sur le nombre total de ressources nécessaires pour former la coalition.

Par exemple, s'il y a trois coalitions à former dans un environnement comportant 25 agents qui peuvent rejoindre chacune des trois coalitions, si la première coalition nécessite 10 agents, la seconde coalition 8 agents et la troisième coalition 12 agents, alors le *ratio* dans un tel environnement est de : $\frac{25}{10+8+12} = \frac{25}{30} = 0,833$.

Le temps moyen nécessaire pour former ces coalitions stables est présenté dans la Figure 2.27. Ces résultats sont une moyenne de 10 simulations pour chaque *Ratio* de l'environnement.

Dans la simulation suivante (Figure 2.28), nous considérons initialement 20 agents de disponibilité aléatoire $panne \sim N(0.5, (0.1)^2)$. Nous lançons 6 tâches de manière séquentielle suivant les configurations décrites dans le tableau 2.3. Nous comparons le temps nécessaire pour former une coalition

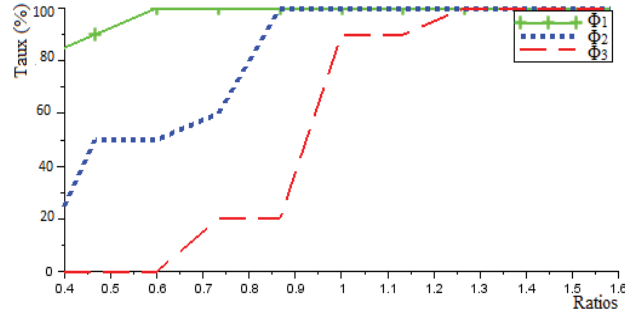


FIGURE 2.26 – Niveau de stabilité garanti suivant les ratios lorsque les indisponibilités des agents suivent une distribution Normale panne $\sim N(0.5, (0.1)^2)$.

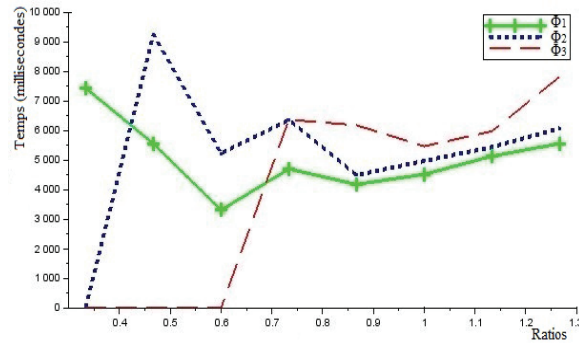


FIGURE 2.27 – Le temps moyen nécessaire pour former $\Phi_i/i \in [1, 3]$ coalitions stables suivant les ratios et quand les indisponibilités des agents suivent une distribution Normale panne $\sim N(0.5, (0.1)^2)$.

avec notre mécanisme avec le temps nécessaire pour les autres mécanismes qui procèdent en calculant l'ensemble des coalitions stables avant de commencer leurs négociations et leurs formations (noté CS dans la figure 2.28). Le mécanisme de (Chalkiadakis et al. 2012) en est un exemple. Les configurations avant chaque simulation :

Nombre de simulations	1	2	3	4	5	6
Nombre d'agents	10	20	40	60	80	100
Distribution de probabilité	Normale $N(0.5, (0.1)^2)$					

TABLE 2.3 – Les différentes configurations des six simulations séquentielles lorsque les indisponibilités des agents suivent une distribution de Poisson $P(0.5)$.

La figure 2.28 montre qu'après la deuxième tâche, notre méthode donne des temps de formation de coalitions plus petits. Au début, les deux mécanismes ont presque les mêmes temps, cependant, notre mécanisme de formation de coalitions n'utilise aucune connaissance a priori ce qui n'est pas le cas des autres mécanismes de formation de coalitions.

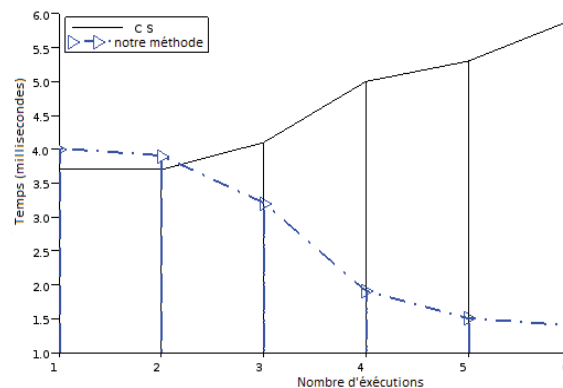


FIGURE 2.28 – Comparaisons des temps nécessaires pour former les coalitions.

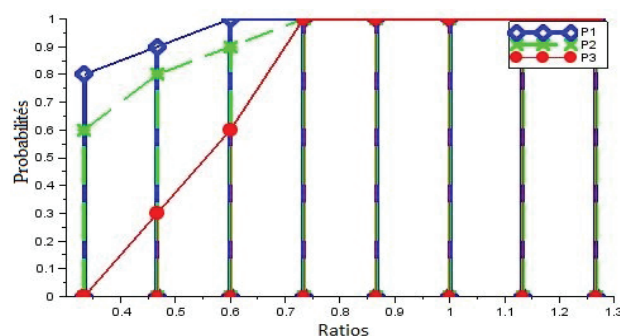
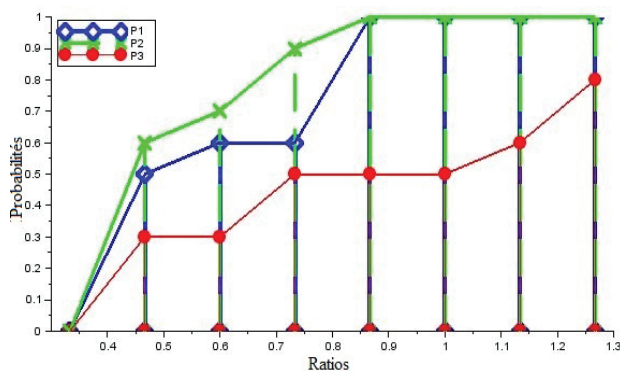
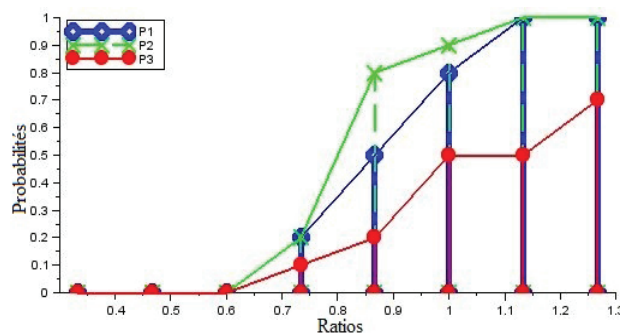
Dans un tel contexte d'environnement dynamique avec plusieurs facteurs imprévisibles sur les tâches et sur les agents, ces résultats expérimentaux s'expliquent sans nul doute par le fait que :

- Notre mécanisme est complètement décentralisé lors des négociations. Pour rappel, chaque agent qui accepte d'être dans le *backbone* de la coalition fait des négociations en parallèle pour que la coalition soit validée et quelle reste stable durant toute l'exécution des tâches de la coalition.
- La construction dynamique des *historiques* permet à chaque agent de connaître dynamiquement la fiabilité, la probabilité de stabilité et la ou les *alliances* qu'il a avec chaque agent voisin ou non, de connaître les agents avec qui il est préférable de coopérer afin d'obtenir une coalition persistante. À cela s'ajoute le fait que le mécanisme de *recommandation* permet aussi à un agent d'utiliser son *historique* pour faciliter ses recherches pour la formation d'une coalition persistante.

Nous étudions dans les simulations ci-dessous, l'influence du calcul du vecteur invariant des tâches pour l'obtention de la stabilité des coalitions fournies par notre mécanisme lorsque les indisponibilités (pannes) des agents suivent une distribution Normale ($panne \sim N(0.5, (0.1)^2)$). Dans la légende des figures ci-dessous, $P1$ signifie que, seule une tâche sur les trois a une probabilité de devenir incontrôlable, Π_2 supérieure ou égale à 0.5. $P2$ signifie que, les deux tâches sur les trois ont une probabilité de devenir incontrôlables, Π_2 supérieure ou égale à 0.5. $P3$ signifie que, toutes les trois tâches ont la même probabilité de devenir incontrôlables, c'est-à-dire, Π_2 identique pour les trois tâches.

Ces figures montrent que, l'utilisation des chaînes de Markov (MDP) pour la détermination des différents vecteurs invariants des tâches dynamiques est adaptée à notre contexte d'étude, car :

- elle permet aux agents d'avoir les mêmes connaissances sur les risques et urgence des tâches et la nécessité de trouver plus de ressources pour assurer la stabilité des coalitions sans que les agents aient à communiquer.
- Les agents cherchent toujours à former des coalitions stables pour toutes les tâches urgentes au regard de leur risque de devenir incontrôlable. La valeur du risque est obtenue après calcul du vecteur invariant et des ratios.

FIGURE 2.29 – Probabilité d'obtenir une coalition stable sachant les probabilités $P1$, $P2$ et $P3$.FIGURE 2.30 – Probabilité d'avoir deux coalitions stables sachant les probabilités $P1$, $P2$ et $P3$.FIGURE 2.31 – Probabilité d'avoir trois coalitions stables sachant les probabilités $P1$, $P2$ et $P3$.

La Figure 2.29 montre que, lorsque c'est uniquement l'une des tâches qui a un risque de devenir incontrôlable supérieur à 0.5, alors les agents décident avec une très grande probabilité de former une coalition stable pour cette tâche plus prioritaire avant de former les coalitions des autres tâches dont les risques de devenir incontrôlable sont plus faibles.

Si deux des tâches ont des risques de devenir incontrôlables supérieurs à 0.5, la Figure 2.30 montre que, ces deux tâches vont recevoir la même attention. Par conséquent, les agents vont former les deux coalitions pour ces deux tâches avant de gérer la troisième tâche qui est moins prioritaire. Ce qui fait que sur cette figure, on observe la diminution des probabilités de former des coalitions stables qui dépendent aussi des ratios de l'environnement de déploiement. Cette diminution des

probabilités de former des coalitions stables et plus visible dans la Figure 2.31 car en fonction des ratios, les agents cherchent à former toutes les coalitions nécessaires aux trois tâches qui ont des probabilités de devenir incontrôlables supérieures à 0.5. Cependant, le fait que dans cette figure, P_2 est au-dessus de P_1 qui est à son tour au-dessus de P_3 , cela signifie que quand les ratios ne permettent pas de former toutes les coalitions stables nécessaires aux trois tâches, les agents vont chercher le maximum de coalitions stables possibles en fonction de la difficulté à former ces coalitions stables et en fonction des ratios de l'environnement de déploiement.

Une coalition est considérée comme simple à former et à stabiliser si les agents qui cherchent à former la coalition sont fiables, avec une grande probabilité de stabilité et avec des *alliances* qui peuvent leur permettre de trouver plus de ressources en utilisant leurs *alliances* ou des demandes de *recommandations*.

Dans la suite, nous montrons l'intérêt d'utiliser notre protocole d'exploration localisée (DEP) et notre protocole d'exploration étendue (WEP) dans un contexte dynamique comme le nôtre où il n'y a aucune connaissance a priori sur l'évolution des tâches et sur les indisponibilités des agents égoïstes. Pour ce faire, nous étudions d'abord les performances de notre mécanisme de formation de coalitions lorsqu'il n'est pas nécessaire d'assurer la persistance des coalitions stables, avant de le comparer au cas où il est nécessaire d'assurer la persistance des coalitions en utilisant notre mécanisme de formation de coalitions. Cette étude est menée dans un contexte où les indisponibilités (pannes) des agents suivent une distribution de probabilités Normale ($panne \sim N(0.5, (0.1)^2)$). Un exemple avec cinq tâches dans l'environnement de déploiement des agents est présenté dans la Figure 2.32, où nous étudions les temps nécessaires pour former et stabiliser les cinq coalitions correspondant aux cinq tâches. Φ_i mesure la probabilité d'avoir $i \in [1, 5]$ coalition(s) stable(s).

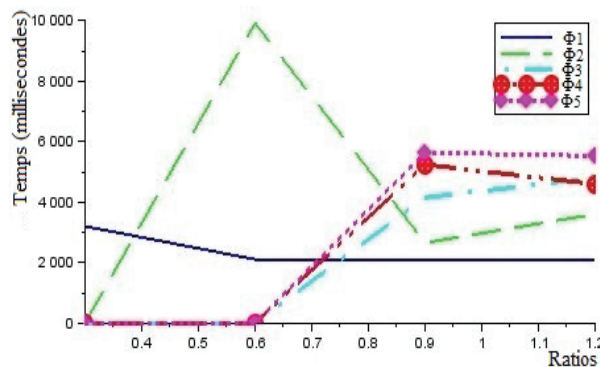


FIGURE 2.32 – Temps nécessaires pour former cinq coalitions stables lorsque les protocoles d'exploration DEP (Depth Exploration Protocol) et WEP (Width Exploration Protocol) sont utilisés.

Une étude similaire est menée avec huit tâches comme le montre la Figure 2.33. Φ_i mesure la probabilité d'avoir $i \in [1, 8]$ coalition(s) stable(s). Ces résultats illustrent l'importance de nos protocoles d'exploration pour raccourcir les temps nécessaires à la formation ou à la stabilisation dynamique des coalitions dans un contexte dynamique et incertain comme le nôtre.

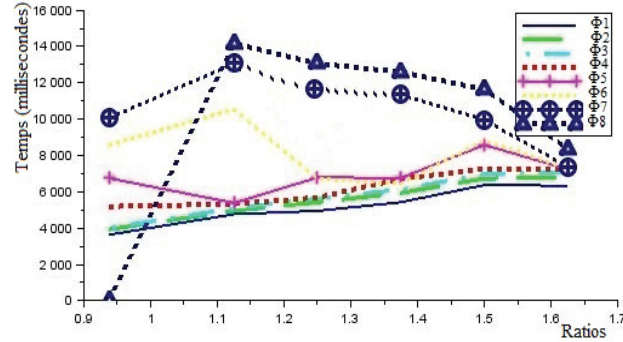


FIGURE 2.33 – Temps nécessaires pour former huit coalitions stables lorsque les protocoles d'exploration DEP (Depth Exploration Protocol) et WEP (Width Exploration Protocol) sont utilisés.

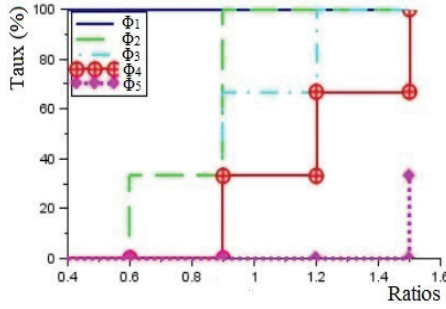


FIGURE 2.34 - Stabilisation dynamique des coalitions sans utilisation des protocoles d'explorations DEP et WEP.

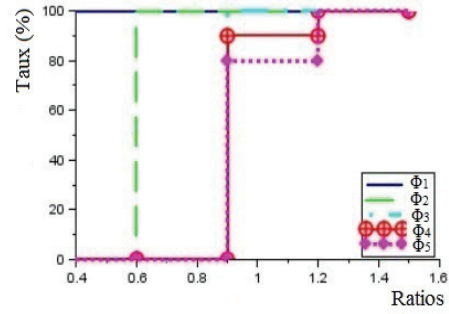


FIGURE 2.35 - Stabilisation dynamique des coalitions quand les protocoles d'explorations WEP et DEP sont utilisés.

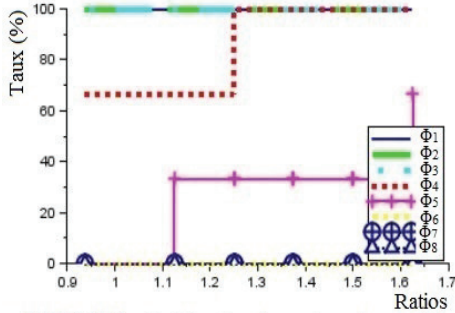


FIGURE 2.36 - Stabilisation dynamique des coalitions sans utilisation des protocoles d'explorations DEP et WEP.

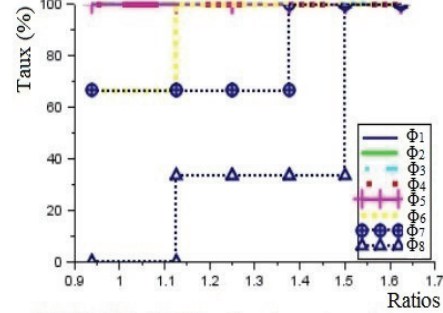


FIGURE 2.37 - Stabilisation dynamique des coalitions quand les protocoles d'explorations DEP et WEP sont utilisés.

La comparaison entre la Figure 2.34 et la Figure 2.35 ou entre la Figure 2.36 et la Figure 2.37 montre que nos protocoles d'exploration localisée et d'exploration étendue rendent possible la stabilisation dynamique des coalitions et une convergence rapide vers des coalitions Nash-stables (dans un équilibre de Nash) dès que les ratios sont égaux à 1.6. Par ailleurs, nous avons un temps raisonnable t : $t \in [2s, 6s]$ (cf. Figure 2.32) pour cinq coalitions et $t \in [6s, 8s]$ (cf. Figure 2.33) pour huit coalitions (avec une coalition formée pour chaque tâche).

2.8.3 Évaluation du coût des messages

Dans cette étude, nous ne considérons que les messages envoyés par chaque agent, car les coûts sont plus élevés du point de vue des ressources réseaux utilisées et des ressources énergétiques des composants électroniques ad-hoc où les agents sont déployés (Smartphone, capteur, robot, etc.). Pendant les négociations, pour un agent voisin d'un agent qui a initié la négociation, le nombre de messages envoyés est égal au minimum à 1 (acceptation d'une offre ou une *recommandation*) et au maximum 3 (établissement d'une *alliance*, acceptation d'une offre et une *recommandation*). Pour un agent initiateur d'une demande de formation d'une coalition, le nombre de messages envoyés est égal au minimum à $1+n$ (déclaration de la demande coalition et acquittement des réponses ayant permis l'obtention des n ressources requises pour sa coalition) et au maximum à $3+n$ (envoi des trois déclarations possibles pour une demande de coalition (cf. algorithme 2 ligne 16 et 17) avant de pouvoir acquitter les n ressources requises pour sa coalition).

Soit, Msg_{init}^{emit} le nombre de messages envoyés par un agent initiateur lors d'une demande de coalition et Msg^{emit} le nombre de messages envoyés par un agent autre que l'initiateur de la demande de formation d'une coalition. Soit, $card(\vartheta_{a_i})=n$ où a_i est l'agent initiateur de la demande de formation d'une coalition. Ainsi, a_i doit envoyer au maximum $3 + n$ messages pour recevoir n messages de réponses à acquitter. D'où, $Msg_{init}^{emit}=3 + n + n$. Chaque agent $a_j \in \vartheta_{a_i}$ qui veut participer dans la coalition de l'agent a_i doit envoyer au maximum 3 messages avant de chercher un agent *subordonné* et de recevoir au plus m messages à acquitter, en supposant qu'il a $card(\vartheta_{a_j})=m$ agents voisins. Ce qui fait que, $Msg^{emit}=3m$.

Soit, $\bar{Msg}$ le coût maximum en termes de messages envoyés durant tout un processus de formation de coalitions. Alors, $\bar{Msg}=3 + n + n + 3m$ si $m \approx n$, $Max(\bar{Msg})=5n + 3$ c'est-à-dire $\bar{Msg}$ est aussi borné.

Le coût d'une synchronisation du *backbone* est aussi borné et dépend de deux configurations réseaux extrêmes.

1- Topologie maillée : Chaque agent $a_i \in A'$ qui est dans le *backbone* qui initie une demande de synchronisation de connaissances (par exemple, stabilité du backbone, disparition d'agents, ...), n'envoie qu'un message pour joindre tous les agents du *backbone*. Ainsi, $\bar{Msg}=1$.

2-Topologie linéaire (topologie en bus) : Chaque agent $a_i \in A'$ du *backbone* qui initie une demande de synchronisation de connaissances va générer des messages en cascades entre les autres agents du *backbone*. Ce qui va engendrer un coût total de synchronisation égal à $\bar{Msg}=n - 1$ si $card(A')=n$ (c'est-à-dire, n agents dans le *backbone*).

Ainsi, pour toutes les topologies réseaux intermédiaires entre les deux topologies ci-dessus, le coût des messages $\bar{Msg}$ est borné tel que $1 \leq \bar{Msg} \leq n - 1$.

Cette étude des interactions entre agents montre que, notre mécanisme de formation de coalitions

prend aussi en compte les contraintes de ressources des équipements ad-hoc qui hébergent les agents, les contraintes de ressources réseaux et les coûts de communication des agents.

CONCLUSION DU CHAPITRE

La formation de coalitions est une importante approche qui vise à faciliter la collaboration et la coordination des agents d'un système multi-agent où chaque agent a un but à atteindre.

Toutefois, dans la plupart des scénarios réalistes, comme celui de cette étude (déploiement des agents dans un environnement sinistré), seul un certain nombre d'agents sont capables de coopérer et d'initier des actions collectives. Pour entreprendre ces actions collectives, les agents doivent trouver un état de consensus qui est bénéfique pour chacun d'eux, d'établir des accords sur ces consensus et coordonner les actions de chaque membre de manière collective. Cela implique que la formation de coalitions d'agents ne peut pas se faire de manière arbitraire par des agents qui ne sont pas connectés ou liés les uns aux autres. La plupart des mécanismes de formation de coalitions actuelles prennent en considération ces contraintes, ouvrant ainsi d'autres axes de recherche pour l'aide à la prise de décision des humains dans des contextes où des humains ayant des équipements communicants doivent établir des équipes efficaces et aptes à réaliser leurs tâches et à atteindre leurs buts.

Cependant, plusieurs contextes d'évolution des applications concrètes sont caractérisés par une dynamique des tâches à réaliser et par des changements parfois imprévisibles de leurs actions pour s'adapter à leur environnement dû au fait que les humains sont le plus souvent autonomes et libres de participer à l'exécution des tâches ou non. Ainsi, utiliser des techniques de coordination multi-agent fondées sur la formation de coalitions pour les contextes réalistes, requière un mécanisme de formation de coalitions dynamique susceptible d'engendrer des coalitions capables de se stabiliser dynamiquement.

Pour cette raison, ce travail a abordé un problème original et difficile pour la formation de coalitions stables où la robustesse face aux événements dynamiques (indisponibilité des agents, amplification des tâches, etc.) et la stabilité des coalitions sont difficiles à prévoir avant le début de l'exécution des tâches du fait des incertitudes sur les indisponibilités des agents et sur les tâches stochastiques. Ce qui nous a amené à proposer un mécanisme de formation de coalitions dynamique fondé sur : 1- notre mécanisme de *recommandation*, 2- un ensemble de lois de probabilité, 3- les chaînes de Markov pour le calcul du risque de dissolution des coalitions afin de simplifier la recherche des consensus entre les agents et la formation dynamique de coalitions qui sont dans un état d'équilibre de Nash. Nous avons aussi proposé deux protocoles d'exploration pour permettre une stabilisation dynamique des coalitions au fil des événements impliquant les agents ou les tâches, c'est-à-dire la formation de coalitions persistantes. Le premier est un protocole d'exploration localisée nommé

DEP (*Depth Exploration Protocol*) et le second est un protocole d'exploration étendue nommé *WEP* (*Width Exploration Protocol*). Nos résultats analytiques et expérimentaux montrent l'efficacité de notre mécanisme décentralisé lorsqu'il est nécessaire de former des coalitions stables du point de vue de Nash et lorsqu'il est nécessaire de maintenir leur persistance dans un contexte dynamique et incertain.

Ce travail soulève aussi quelques directions de recherche non moins importantes. L'une des plus importantes est, la recherche de coalitions capables de s'auto-stabiliser dynamiquement où le bien-être social des agents est maximisé et où il y a des incertitudes sur les dépendances des agents, sur les tâches et sur les externalités des coalitions qui dépendent à la fois des agents et des tâches stochastiques.

3

FORMATION DE COALITIONS A-CORE : S-NRB

Dans ce chapitre, nous nous concentrons sur la mise en œuvre d'un mécanisme décentralisé de formation de coalitions stables dans un contexte de tâches stochastiques qui tient compte des incertitudes - sur les dépendances des agents, - sur les conflits dus aux préférences des agents et - sur la disponibilité imprévisible des agents. Nous ne supposons aucune connaissance a priori sur les coalitions stables à former. Par ailleurs, nous considérons qu'il n'est pas possible de calculer de façon centralisée ces coalitions avant le début de l'exécution des tâches en raison des incertitudes des agents et la contrainte de temps des tâches stochastiques. Les différences principales par rapport au chapitre précédent sont qu'ici, nous recherchons des coalitions stables au sens du Core et nous tenons compte des dépendances entre les agents. Le mécanisme de formation de coalitions stables que nous proposons est dit S-NRB (Sequential Non-Return Broadcast) et permet de former des coalitions dites A-core, c'est-à-dire, Core-stable et Auto-stables. Notre mécanisme 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.

3.1 OBJECTIFS ET APPROCHE GÉNÉRALE

Les réseaux de Smart-phones, de robots et de capteurs qui améliorent la sécurité et les prises de décision des humains, soulèvent certains questionnements dans plusieurs domaines de recherche dont les systèmes multi-agents. Pour cette raison, plusieurs applications multi-agents actuelles axent leurs solutions dans un contexte où les agents sont déployés dans des robots autonomes et dans des dispositifs personnels connectés.

Les objectifs des agents peuvent être divers, par exemple, dans les situations de sinistres, il s'agira de fournir une meilleure coordination des secours par l'établissement de réseaux ad-hoc et de favoriser l'émergence de décisions pertinentes qui tiennent compte des préférences et des capacités des intervenants et des robots. Dans ces contextes, lorsque chaque agent est déployé dans un seul composant ad-hoc (e.g. PDA, capteur, Smartphone, robot), son but est de maximiser l'utilité et l'efficacité de son intervenant (e.g. robot, secouriste humain) favorisant les travaux en équipe d'agents (coalitions) en cas de nécessité.

Plusieurs mécanismes de formation de coalitions ont été étudiés dans le domaine des agents (e.g., (Sandholm et al. 1999)(Shehory et Kraus 1998)), cependant, un ensemble d'aspects intéressants qui sont sous-jacents à l'utilisation de tels réseaux ad-hoc pour le déploiement des agents, restent faiblement explorés.

La principale contribution de ce chapitre est la proposition d'un mécanisme de formation de coalitions *core stables* qui est adaptatif, décentralisé et asynchrone. La propriété adaptative de notre mécanisme signifie que les coalitions *core stables* doivent pouvoir se stabiliser dynamiquement en fonction du contexte des agents. Par mécanisme décentralisé, nous sous-entendons qu'en aucun cas les interactions entre les agents ou les calculs ne doivent être centralisés.

Plus en détail, nous proposons un mécanisme de formation de coalitions décentralisé et asynchrone appelé *S-NRB* (*Sequential Non-Return Broadcast*) qui se fonde sur des négociations multilatérales et 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, notre mécanisme *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 qui vont s'opérer 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 qui nous sert d'exemple illustratif 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 ont des composants électroniques qui hébergent des agents dans leur kit de secours, doivent prendre des décisions en fonction : - de leur groupe d'appartenance avant le déploiement sur le terrain et - des informations contextuelles auxquelles se réfèrent les agents. 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.). En toute rigueur, 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.

3.2 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 $A = \{a_1, \dots, a_n\}$ 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 $A_c \subset A = \{a_1, a_2, \dots, a_k\}$ 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 un ensemble de tâches stochastiques à exécuter par la coalition C et V_c est la récompense associée à la coalition C après l'exécution de T_c . La définition d'une tâche est telle que donnée dans le chapitre précédent.

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 : $\{R_{a_i}, Aut_{a_i}^t, Hs_{a_i}, \vartheta_{a_i}^t, U_{a_i}, L_{a_i}^{Net}\}$.

Pour l'agent a_i , l'information R_{a_i} décrit sa ressource et $Aut_{a_i}^t$ son autonomie énergétique à l'instant t . Hs_{a_i} représente son *historique*. Ces paramètres sont déjà définis dans le précédent chapitre. La définition de la notion de vue $\vartheta_{a_i}^t$ d'un agent a_i à l'instant t reste aussi la même, cependant dans la

formalisation, un agent n'a plus besoin de vérifier les *alliances* qu'il partage avec un agent dès que ce dernier est dans son voisinage (cf. figure 3.1).

U_{a_i} est la fonction d'utilité de a_i et $L_{a_i}^{Net}$ définit, à un instant t , le niveau de dépendance de l'agent a_i par rapport aux autres agents de son voisinage $\vartheta_{a_i}^t$ dans une structure réseau notée Net .

Définition 3.1 La fonction d'utilité U_{a_i} d'un agent a_i s'exprime sous la forme $U_{a_i} = \sum_{c=1}^{\infty} u_c^{a_i}$ où $u_c^{a_i}$ est son utilité suite à sa participation à la coalition C . $U_{a_i}^t$ est la valeur de la fonction d'utilité de l'agent a_i à l'instant t .

Les agents utilisent les processus de négociation pour former des coalitions stables qui maximisent leur récompense dans chacune de leurs coalitions malgré les incertitudes et la dynamique du contexte de déploiement. 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 3.2 La fonction d'utilité qu'un agent a_i cherche à maximiser en participant dans la coalition C s'écrit :

$$u_c^{a_i} = v_{a_i} - \text{Cost}_c^{a_i} \quad (3.1)$$

Cette fonction d'utilité est une information privée vue que la fonction de coût $\text{Cost}_c^{a_i}$ est aussi privée. La seule connaissance partagée par les agents est la récompense attendue par la coalition V_C une fois que la tâche T_C est réalisée.

Définition 3.3 Soit $L_{a_i}^{Net}$ les dépendances de a_i avec les agents de sa vue $\vartheta_{a_i}^t$ qui forment une quelconque structure réseau Net telle que $\vartheta_{a_i}^t \subseteq Net$.

$\forall a_i \in Net, \exists L_{a_i}^{Net}$ telle que $L_{a_i}^{Net} = \{\gamma_{a_i}^{Net}, H_{a_i}^{Net}, S_{a_i}^{Net}\}$.

$\gamma_{a_i}^{Net}$ est l'ensemble des agents dont dépend a_i . $S_{a_i}^{Net}$ est l'ensemble des agents qui dépendent de a_i . $H_{a_i}^{Net} = (\vartheta_{a_i}^t \setminus \gamma_{a_i}^{Net}) \setminus S_{a_i}^{Net}$ 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 simplement avoir des accords antérieurs auxquels ils doivent tenir compte au moment du déploiement. De plus, certains intervenants peuvent n'être que des volontaires indépendants.

Exemple 3.1 Considérons le déploiement des agents ci-dessous :

Si nous considérons que, a_2 et a_3 dépendent seulement de a_6 et que le reste des agents sont uniquement des voisins, alors :

$$\begin{aligned} L_{a_1}^{Net} &= \{\{\emptyset\}, \{a_5, a_6\}, \{\emptyset\}\} & L_{a_2}^{Net} &= \{\{a_6\}, \{a_3\}, \{\emptyset\}\} \\ L_{a_3}^{Net} &= \{\{a_6\}, \{a_2\}, \{\emptyset\}\} & L_{a_4}^{Net} &= \{\{\emptyset\}, \{a_5\}, \{\emptyset\}\} \\ L_{a_5}^{Net} &= \{\{\emptyset\}, \{a_1, a_4\}, \{\emptyset\}\} & L_{a_6}^{Net} &= \{\{\emptyset\}, \{a_1\}, \{a_2, a_3\}\} \end{aligned}$$

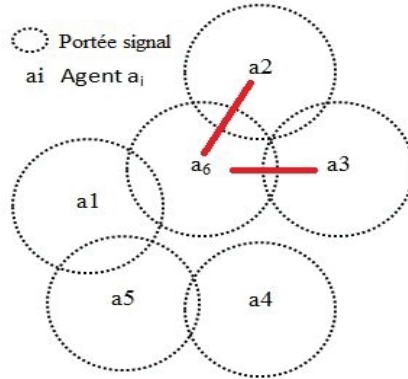


FIGURE 3.1 – Répartition (Net) des agents dans l'environnement de déploiement. La figure suggère que, $\vartheta_{a_1}^t = \{a_5, a_6\}$, $\vartheta_{a_2}^t = \{a_6, a_3\}$, $\vartheta_{a_3}^t = \{a_2, a_6\}$, $\vartheta_{a_4}^t = \{a_5\}$, $\vartheta_{a_5}^t = \{a_1, a_4\}$ et $\vartheta_{a_6}^t = \{a_1, a_2, a_3\}$. Le trait entre les agents a_6 et a_2 et celui entre les agents a_6 et a_3 matérialisent le fait que c'est a_6 qui gère la participation des agents a_2 et a_3 dans l'environnement de déploiement.

Ainsi, les agents doivent aussi tenir compte de ces dépendances pour déterminer leurs préférences et leurs interactions prioritaires ou interdites lorsqu'ils cherchent à maximiser leur utilité.

Nous réutilisons aussi les notions de probabilité de stabilité, de fiabilité et d'alliances déjà introduites dans le chapitre précédent.

Avec ces exigences et ces incertitudes, nous présentons dans la section suivante, notre nouveau mécanisme de formation de coalitions stables appelé *S-NRB* qui se fonde sur des négociations multilatérales. Mécanisme qui cherche une convergence rapide vers des coalitions *core stables*, lors des phases d'exécution des tâches stochastiques.

3.3 MÉCANISME DE FORMATION DE COALITIONS

Nous présentons un ensemble de stratégies de négociation que les agents utilisent afin de surmonter leurs incertitudes relatives à leurs disponibilités, leurs préférences et aux tâches à réaliser. L'objectif est de trouver le ou les meilleures coalitions stables qui maximisent le bien-être social des agents (coalitions *core stables*) et qui peuvent se stabiliser dynamiquement (coalitions A-cores).

Définition 3.4 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 $\forall C, C' \in E_C, u_C^{a_i} \geq u_{C'}^{a_i}$.
- Tout agent de C a une alliance non-dominée avec au moins un agent de C , c'est-à-dire,

$\forall a_i \in C, \exists Al_{a_i, a_j} : a_j \in C.$

- Les agents de C forment un ensemble connexe, c'est-à-dire, $\forall a_i \in C, \exists a_j \in \vartheta_{a_i}^t : a_j \in C.$

Notre mécanisme de formation de coalitions $S\text{-NRB}$ est fondé sur : - un principe de sondage parallèle et asynchrone et - un ensemble de règles afin d'éviter des offres erronées ou en boucles dues aux incertitudes et aux dynamiques du contexte considéré.

Pour ce faire, les agents utilisent des offres composées.

Définition 3.5 L'offre composée pour la formation d'une coalition C d'un agent a_i est notée $O_c^{a_i}$. Formellement, $O_c^{a_i} = \{\{B_i, B_j, B_k, \dots\}, \{pl_{a_i}\}\} : B_i = (R_i, \Delta_i, U_i)$. B_i est un but à atteindre par l'offre qui spécifie, la ressource R_i requise pour T_c , les contraintes Δ_i de B_i et la récompense U_i associée à B_i . $pl_{a_i} = \{a_i, a_j, a_k, \dots\} : a_i \in A$ désigne une liste de sondage qui permet de savoir dynamiquement à un instant donné, les agents qui acceptent de s'associer pour un but de l'offre de a_i .

Définition 3.6 Deux agents a_j et a_k sont en conflit pour une offre composée $O_c^{a_i}$, lorsque a_j et a_k ont la même contribution marginale ($X_{a_j} = X_{a_k}$), le même but $B_x \in O_c^{a_i}$ et veulent tous les deux participer à la même coalition C .

Notations

$\equiv$ L'équivalence entre deux paramètres.

$\neq$ La non équivalence entre deux paramètres.

$\succ_P$ Dominance au sens de Pareto.

$\succ_L$ Dominance au sens de Lorenz.

3.3.1 Notre mécanisme $S\text{-NRB}$

Le mécanisme de formation de coalitions $S\text{-NRB}$ (cf. algorithme 5), peut être décrit en ces quatre étapes ci-dessous :

Étape 1 : Stratégies de diffusion des offres de formation de coalition.

Tout agent a_j qui reçoit une offre $O_c^{a_i}$ pour la formation d'une coalition C doit envoyer au maximum un message avec un $TTL=1$ (cf. équation 3.7) dans son voisinage représenté par sa vue $\vartheta_{a_j}^t$. 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, pour simplifier les négociations et ainsi raccourcir les délais d'attentes des réponses des agents.

- Un agent a_j ne diffuse le message de sondage contenant l'offre $O_c^{a_i}$ dans son voisinage direct $\vartheta_{a_j}^t$ que s'il accepte de participer à la coalition C (cf. ligne 1 - 4 algorithme 5). Cet envoi du message est effectué après avoir vérifié que son but $G_{a_j} \equiv B_i : B_i \in O_c^{a_i}$, que son utilité $U_{a_j}^t$ courante (c'est-à-dire, à l'instant t) est supérieure à son utilité $U_{a_j}^{t-1}$ à l'instant $t - 1$ et après avoir inscrit son identifiant dans la liste de sondage $pl_{a_i} \subset O_c^{a_i}$.

La figure 3.2 montre que, si l'agent a_5 diffuse l'offre il n'y a que les agents de sa vue (a_4 et a_1) qui la reçoivent.

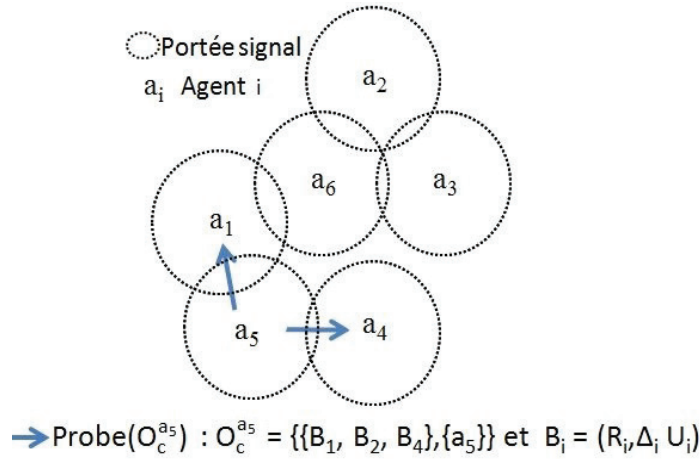


FIGURE 3.2 – Diffusions séquentielles du message de sondage pour l'offre $O_c^{a_5} = \{\{B_1, B_2, B_4, B_5\}, \{\emptyset\}\}$, c'est-à-dire, une offre de formation d'une coalition C assignée à a_5 . Seuls les agents de la vue de a_5 reçoivent l'offre de a_5 , car $TTL = 1$ dans le message de sondage. De plus, a_4 et a_1 doivent prendre une décision locale avant de contacter à leur tour, les agents de leur vue.

À noter que, si un agent accepte une offre, il supprime de cette offre l'information sur son but $G_{a_j} \equiv B_i : B_i \in O_c^{a_i}$ 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 formation de coalition.

- Un agent peut répondre à l'offre par un rejet grâce à un message de type $ResetProbe(O_c^{a_i})$ (cf. ligne 26 - 28 algorithme 5) ou par une demande de modification de l'offre (contre-proposition) en utilisant un message de type $counter(O_c^{a_i})$ (cf. ligne 23 - 25 algorithme 5). 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 $O_c^{a_i}$ proposée.

Étape 2 : Stratégies d'optimisation des messages envoyés.

$\forall a_j$ tel que, $\forall a_k \in \vartheta_{a_j}^t, a_k \in pl_{a_i}$ (c'est-à-dire, un agent avec des voisins déjà dans la liste de sondage) ou $\vartheta_{a_j}^t = \emptyset$ (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, ce qui peut entraîner l'apparition de plusieurs chemins de messages entre deux agents.

Par exemple, sur la figure 3.2, s'il y a des problèmes de transmission entre a_6 et a_3 , un message de a_1 peut passer par a_6, a_2 et puis a_3 au lieu de passer par a_6 et a_3 . De plus, dans cette configuration, aucun agent ne peut contacter en même temps tous les autres agents.

En résultat, plusieurs listes de sondage $pl_{a_i}, pl'_{a_i}, pl''_{a_i}, \dots$, peuvent coexister pour une même offre si des mesures ne sont pas prises pour l'éviter.

Pour cette raison, S-NRB cherche aussi à détecter dynamiquement les listes de sondage redondantes et à privilégier les agents avec qui la stabilité et le bien-être social des agents sont préservés. Cela, afin de favoriser la formation de coalitions (*core stables*).

La figure 3.3 montre que, l'agent a_1 accepte et propage l'offre dans son voisinage sans donner sa réponse à a_5 , car chaque agent ne doit envoyer qu'un message comme réponse. Nous remarquons que l'agent a_4 accepte et donne sa réponse à a_5 , car sa vue est vide.

Dans la figure 3.4, a_6 refuse l'offre, car il donne sa réponse sans poursuivre le sondage pour l'offre.

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 la coalition C doit effectuer une fusion des différentes listes de sondage afin de limiter les conflits entre les agents et de produire une seule liste de sondage (cf. ligne 11 - 14 algorithme 5). Suite à cela, la liste est

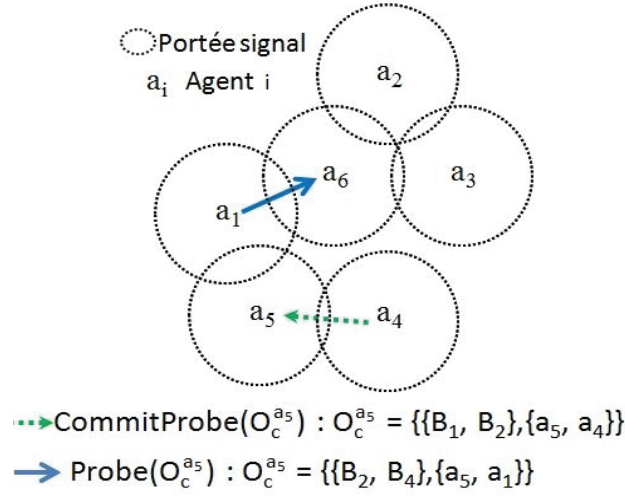


FIGURE 3.3 – Prises de décision individuelles avant de propager le message de sondage pour l'offre. Le voisinage de a_4 est vide donc il donne sa réponse, mais a_1 poursuit le sondage pour l'offre proposée par a_5 . Nous voyons une différence des listes de sondage, car $pl_{a_5} = \{\{a_5, a_4\}\}$ dans la réponse de a_4 alors que $pl_{a_5} = \{\{a_5, a_1\}\}$ dans le message de a_1 . Cela est dû au fait que, a_4 et a_1 ne sont pas dans le même voisinage et chacun d'eux ignore la réponse de l'autre.

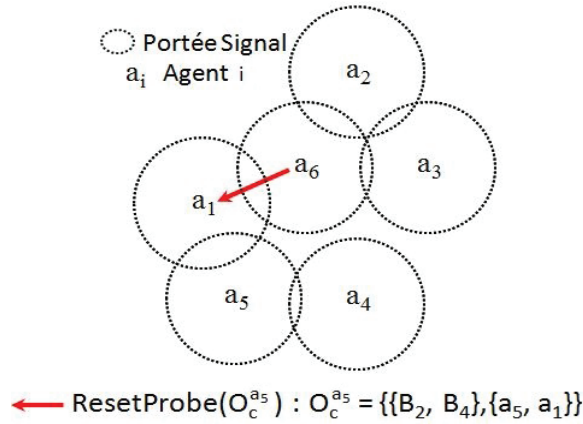


FIGURE 3.4 – Prises de décisions individuelles avant de répondre au message de sondage pour l'offre. Dans le cas du refus de a_6 , nous voyons qu'il n'y a pas de changement dans la liste de sondage par rapport à la figure 3.3 et puis a_6 ne contacte pas les autres agents de sa vue (a_2 et a_3).

diffusée de proche en proche avec un $TTL=1$. L'algorithme 6 décrit le mécanisme de fusion des listes de sondages.

Définition 3.7 Soit $pl_{a_i}^1, pl_{a_i}^2, pl_{a_i}^3$ un ensemble de 3 listes de sondage pour une même offre $O_c^{a_i}$ de formation de coalition. La fusion de ces listes de sondage est le fait de les associer de telle sorte qu'on ait une seule liste de sondage pl_{a_i} résultante qui ne contient pas de conflits entre les agents. Formellement,

$$pl_{a_i} = (pl_{a_i}^1 \cup pl_{a_i}^2 \cup pl_{a_i}^3) : (pl_{a_i}^1 \cap pl_{a_i}^2 \cap pl_{a_i}^3) = \emptyset \text{ et } \forall a_x, a_y \in pl_{a_i}, a_x \neq a_y, \text{ alors } G_{a_x} \not\equiv G_{a_y}.$$

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

Si $(\cup B_i \subset O_c^{a_i}) \neq (\cup_{a_j \in pl_{a_i}} G_{a_j})$, 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 5).

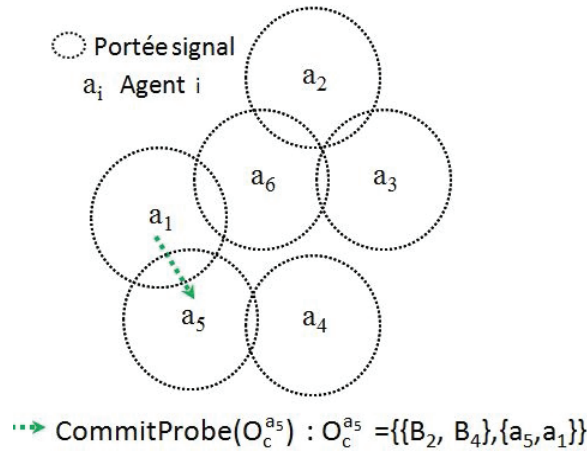


FIGURE 3.5 – Réponse d'un agent après la fin du sondage de son voisinage. Suite au refus de a_6 , a_1 donne sa réponse à a_5 pour indiquer sa décision et celle de son voisinage.

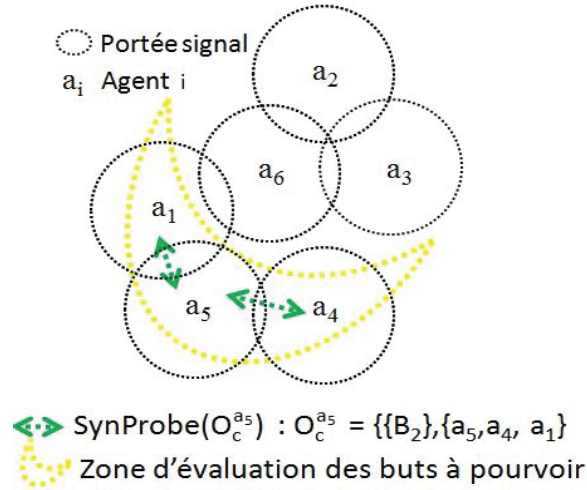


FIGURE 3.6 – Prise de décision collective entre les agents a_1 , a_4 et a_5 qui ont accepté l'offre pour décider s'il est nécessaire de poursuivre le sondage pour l'offre. Cette décision dépend de l'état d'acceptation des buts de l'offre et de la disponibilité des agents de la liste de sondage.

Pour ce faire, ils utilisent la loi de probabilité hypergéométrique (Yates et Goodman 2005), (Baynat 2000). Cette loi de probabilité est la loi a priori pour la prédiction de l'issue probable d'un sondage.

Ainsi,

$\forall a_j \in pl_{a_i}, \text{échantillon}=n=\text{Card}(pl_{a_i}), \text{population}=N=\text{Card}(\cup B_i) : B_i \in O_c^{a_i}.$

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.

$\forall a_j \in pl_{a_i} : a_x \in \vartheta_{a_j}^t, a_x \notin pl_{a_i}$, 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 afin de former la coalition C pour l'offre $O_c^{a_i}$. La probabilité de $X=k$ est donnée par l'équation :

$$P[X = k] = \frac{C_{NQ_{counter}}^k C_{NP_{counter}}^{n-k}}{C_N^n} \quad (3.2)$$

L'espérance de cette probabilité est :

$$E(X) = nQ_{counter} \quad (3.3)$$

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 Marshall et Olkin (1979).

Définition 3.8 Pour tout $x \in \mathbb{R}_+^q$, le vecteur de Lorenz généralisé associé à x est le vecteur :

$$L(x) = (x_1, x_1 + x_2, x_1 + x_2 + x_3, \dots, x_1 + x_2 + \dots + x_q)$$

où, $x_1 \geq x_2 \geq x_3 \geq \dots \geq x_q$ représentent les composant de x rangées par ordre décroissant. L'expression d'une $n^{ième}$ composante de $L(x)$ peut s'écrire sous la forme :

$$L_n(x) = \sum_{i=1}^n x_i.$$

Définition 3.9 On dit que x domine y au sens de Lorenz et on le note par $x \succsim_L y$, si

$\forall x, y \in \mathbb{R}_+^q, L(x) \succsim_P L(y)$, c'est-à-dire, $L(x)$ domine $L(y)$ au sens de Pareto.

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

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, sinon c'est a_k qui est le responsable.

Nous préférons utiliser la dominance de Lorenz pour éviter les cas où une prise de décision est impossible avec la dominance de Pareto.

Exemple 3.2 Soit deux vecteurs $X = (12, 6)$ et $Y = (10, 7)$. Dans ce cas, utiliser la dominance au sens de Pareto ne nous permet pas de conclure, car pour le premier paramètre, X est meilleur, mais pour le second paramètre, Y est meilleur. Cependant, avec la dominance au sens de Lorenz, nous voyons que X domine Y , car $\sum_{i=1}^2 L_i(x) = 18$ et $\sum_{i=1}^2 L_i(y) = 17$.

Pour chaque agent, sa décision pour participer à C dépend de sa fonction d'utilité, de ses dépendances, de ses alliances et de sa fiabilité.

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 $O_c^{a_i}$ pour la formation d'une coalition C . Cela, permet de différencier les offres, car de manière simultanée, nous pouvons avoir plusieurs tâches à exécuter et plusieurs offres provenant d'agents distincts.

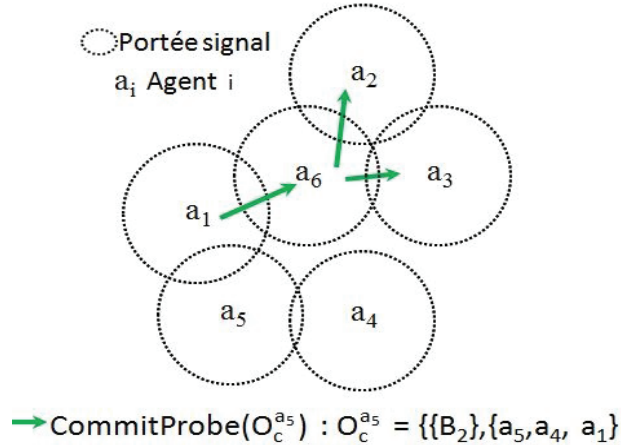


FIGURE 3.7 – Message de confirmation de la continuité du sondage pour indiquer la nécessité d'avoir plus de réponses favorables à la formation de la coalition pour l'offre. Ici, seulement a_1 envoie ce message de confirmation, car tous les voisins de a_5 sont déjà dans la liste de sondage pl_{a_5} et a_4 n'a que a_5 comme voisin. Nous n'avons pas séparé les étapes de réception du message par a_6 et l'étape où a_6 envoie le message, car a_6 ne fait que relayer le message. Cela, du fait que a_6 a déjà rejeté l'offre de formation de C .

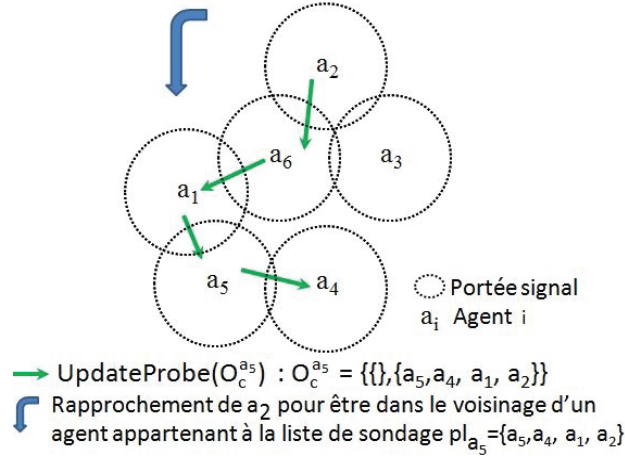


FIGURE 3.8 – Réponses supplémentaires d'agents après la confirmation de la continuité du sondage pour l'offre. Les agents peuvent avoir à se déplacer en cas de nécessité, pour former un ensemble connexe. Dans cette figure, nous supposons que c'est a_2 qui se déplace. Ici aussi, la réponse de a_2 est simplement relayée par les agents qui la reçoivent, car c'est un message de mise à jour d'informations.

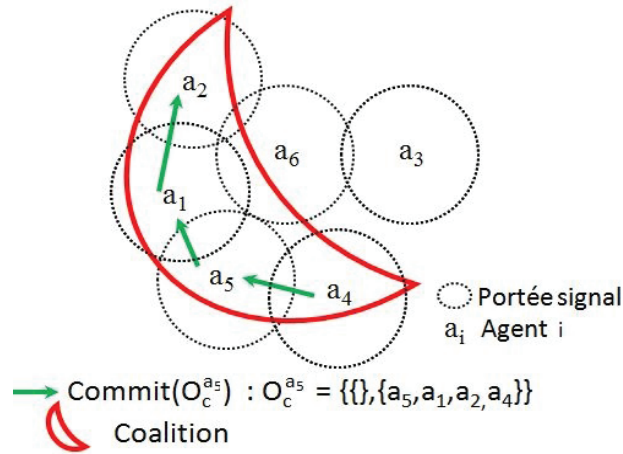


FIGURE 3.9 – Validation de la liste de sondage pour l'offre. La coalition pour l'offre par les agents de la liste de sondage $pl_{a_5} = \{a_1, a_2, a_4, a_5\}$ comme l'ensemble des buts de l'offre sont acceptés. Ainsi, $A_c = \{a_1, a_2, a_4, a_5\}$.

Dans la section suivante, nous détaillons les stratégies que les agents utilisent pour résoudre leurs conflits sans compromettre la convergence vers des coalitions *core stables*.

Algorithme 5 : : 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

```

1: 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
2:   if  $\vartheta_{aj}^t \neq \emptyset$  et  $a_j \in pl_{a_i}$  then
3:     Envoyer  $Probe(O_c^{a_i}) : pl_{a_i} = \{a_i, a_j\} \forall a_k \in \vartheta_{aj}^t$ 
4:   end if
5:   if  $\vartheta_{aj}^t = \emptyset$  et  $a_j \in pl_{a_i}$  then
6:     Envoyer  $CommitProbe(O_c^{a_i}) : pl_{a_i} = \{a_i, a_j\} \forall a_k \in \vartheta_{aj}^t$ 
7:   end if
8:   if  $\vartheta_{aj}^t \neq \emptyset$  et  $a_j \in pl_{a_i}$  et  $\exists CommitProbe(O_c^{a_i})$  then
9:     Envoyer  $Commit(pl_{a_i})$ 
10:  end if
11:  if  $(\exists a_j \in pl_{a_i})$  ou  $(\exists a_k \in pl'_{a_i})$  et  $(a_k \notin pl_{a_i})$  then
12:     $UpdateProbe(O_c^{a_i})$  //pour mettre à jour  $pl_{a_i}$  après fusion de  $pl_{a_i}$  et  $pl'_{a_i}$ 
13:  end if
14:  if  $\exists$  conflit entre  $a_j$  et  $a_k$  then
15:     $Matching(pl_{a_i}, pl'_{a_i})$  //pour initier l'exécution de S-NRB-CRP() (algorithme 6)
16:    Envoyer  $UpdateProbe(O_c^{a_i})$  //pour mettre à jour  $pl_{a_i}$ 
17:  end if
18:  if  $\cup B_i \subset O_c^{a_i} = \cup_{a_j \in pl_{a_i}} G_{a_j}$  then
19:    Valider  $C = \{\forall a_j \in pl_{a_i}\}$  et  $G_c = \cup_{a_j \in pl_{a_i}} G_{a_j}$ 
20:  else
21:     $\forall a_j \in pl_{a_i}$  trouver plus de ressources en utilisant  $Al_{a_j, a_k} : a_k \in A, L_{a_j}^{Net}$  et  $Counsel(a_i)$  //Le
     $Counsel(a_i)$  est pour guider les recherches de  $a_i$  (cf. règle 2 section règles générales).
22:  end if
23: else
24:  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
25:    Envoyer  $Counter(O_c^{a_i})$ 
26:  else
27:    Envoyer  $ResetProbe(O_c^{a_i})$ 
28:  end if
29: end if

```

3.3.1.1 Protocole de résolution des conflits du mécanisme S-NRB

Comme nous l'avons présenté à la définition 3.6, ces conflits peuvent bloquer la procédure de validation de la coalition C , si une méthode efficace de résolution des conflits n'est pas définie.

Le protocole de résolution des conflits S-NRB-CRP (Sequential Non-Return Broadcast Conflict Resolution Protocol) est fondé sur :

- une phase de comparaison (matching) des listes de sondage et l'utilisation de l'index de Jaccard (Jahanbakhsh et al. 2012) développé de la ligne 3 à la ligne 11 de l'algorithme 6), pour mesurer les dissemblances,
- la domination au sens de Lorenz pour les prises de décision en fonction de certains critères de performance. Nous l'utilisons de la ligne 12 à la ligne 19 de l'algorithme 6),
- un mécanisme de demande de garants (*Grant*) comme indiqué de la ligne 20 à la ligne 22 de l'algorithme 6). Ce mécanisme permet d'assurer une décision distribuée et cohérente s'il n'y a pas de dominance entre les agents en conflits.

Étape 1 : Matching et mesure de dissemblances.

L'indice et la distance de Jaccard sont deux métriques utilisées pour évaluer la similarité et la diversité entre des ensembles. L'indice de Jaccard (ou coefficient de Jaccard) est le rapport entre le cardinal (la taille) de l'intersection des ensembles considérés et le cardinal de l'union des ensembles.

Définition 3.10 Soit deux ensembles A et B , l'indice de Jaccard est :

$$J(A, B) = \frac{\text{Card}(A \cap B)}{\text{Card}(A \cup B)} \quad (3.4)$$

Cet indice devient pour n ensembles $S_1, S_2, \dots, S_n$:

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

La distance de Jaccard mesure la dissemblance entre les ensembles. Elle consiste à soustraire l'indice de Jaccard à 1.

$$J_\delta(A, B) = 1 - J(A, B) = \frac{\text{Card}(A \cup B) - \text{Card}(A \cap B)}{\text{Card}(A \cup B)} \quad (3.5)$$

De la même manière que pour l'indice de Jaccard, la généralisation pour les n ensembles $S_1, S_2, \dots, S_n$ est :

$$J_\delta(S_1, S_2, \dots, S_n) = 1 - J(S_1, S_2, \dots, S_n) = \frac{\text{Card}(S_1 \cup S_2 \cup \dots \cup S_n) - \text{Card}(S_1 \cap S_2 \cap \dots \cap S_n)}{\text{Card}(S_1 \cup S_2 \cup \dots \cup S_n)}.$$

À la ligne 4 de l'algorithme 6, $Jacc(a_j)$ et $Jacc(a_k)$ signifie que a_j et a_k doivent donner respectivement leur liste $L_{a_j}^{Jacc}$ et $L_{a_k}^{Jacc}$.
 $L_{a_j}^{Jacc}$ et $L_{a_k}^{Jacc}$ 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 la même coalition C.

Si nous notons par $AllianceSet^{a_x}$ l'ensemble des agents alliés d'un agent a_x , alors :

$$L_{a_j}^{Jacc} = (\gamma_{a_j}^{Net} \cup S_{a_j}^{Net} \cup AllianceSet^{a_j}) \cap pl_{a_i}.$$

$$L_{a_k}^{Jacc} = (\gamma_{a_k}^{Net} \cup S_{a_k}^{Net} \cup AllianceSet^{a_k}) \cap pl_{a_i}.$$

$Jaccard(a_j, a_k)$ correspond à l'indice de Jaccard entre les deux ensembles $L_{a_j}^{Jacc}$ et $L_{a_k}^{Jacc}$ obtenue respectivement par les agents a_j et a_k .

Étape 2 : Prise de décision en se fondant sur la dominance au sens de Lorenz.

À cette étape, la dominance de Lorenz permet de faire un choix entre deux agents a_j et a_k . Il est fondé sur l'utilisation des 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 $P_s^{a_k}$.

En posant :

$$L(x) = (x_1, x_1 + x_2, x_1 + x_2 + x_3) = (Card(L_{a_j}^{Jacc}), Card(L_{a_j}^{Jacc}) + \rho_{a_j}, Card(L_{a_j}^{Jacc}) + \rho_{a_j} + P_s^{a_j}) \text{ et}$$

$$L(y) = (y_1, y_1 + y_2, y_1 + y_2 + y_3) = (Card(L_{a_k}^{Jacc}), Card(L_{a_k}^{Jacc}) + \rho_{a_k}, Card(L_{a_k}^{Jacc}) + \rho_{a_k} + P_s^{a_k}).$$

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. Dans ce cas, la participation de l'agent a_j est préférée à celle de l'agent a_k .

Dans la mesure où le calcul de la dominance de Lorenz ne permet pas de sélectionner un agent parmi ceux qui sont en conflits, un mécanisme de demande de garants est alors initié.

La ligne 21 de l'algorithme 6 ($Grant(a_j, a_k)$) est une requête de demande de garants des agents a_j et a_k qui sont en conflits. Cette demande est envoyée aux agents $a_x \in pl_{a_i} \setminus \{a_j, a_k\}$ et $a_y \in pl'_{a_i} \setminus \{a_j, a_k\}$. Si aucune des étapes précédentes ne permet de départager les agents en conflits, c'est-à-dire, a_j et a_k , alors l'agent qui est choisi est tel que : $Max(\rho_{a_j}, \rho_{a_k})$, $Max(Aut_{a_j}^t, Aut_{a_k}^t)$ et $Max(P_s^{a_j}, P_s^{a_k})$.

Cependant, pour éviter des impasses dans le mécanisme de demande de garants, les agents doivent observer certaines contraintes. Nous détaillons ces contraintes dans la section suivante.

Algorithme 6 : 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** $a_j \in S_{a_k}^{Net}$ **then**
- 2: Notifier à a_j que sa demande de participation est annulée
- 3: **else**
- 4: $L_{a_j}^{Jacc} = Jacc(a_j)$ et $L_{a_k}^{Jacc} = Jacc(a_k)$
- 5: $Jaccard(a_j, a_k) = \left| \frac{L_{a_j}^{Jacc} \cap L_{a_k}^{Jacc}}{L_{a_j}^{Jacc} \cup L_{a_k}^{Jacc}} \right|$
- 6: **if** $1 - Jaccard(a_j, a_k) \neq 0$ **then**
- 7: **if** $Card(L_{a_j}^{Jacc}) < Card(L_{a_k}^{Jacc})$ **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: $L(x) = (x_1, x_1 + x_2, x_1 + x_2 + x_3) = (Card(L_{a_j}^{Jacc}), Card(L_{a_j}^{Jacc}) + \rho_{a_j}, Card(L_{a_j}^{Jacc}) + \rho_{a_j} + P_s^{a_j})$
- 14: $L(y) = (y_1, y_1 + y_2, y_1 + y_2 + y_3) = (Card(L_{a_k}^{Jacc}), Card(L_{a_k}^{Jacc}) + \rho_{a_k}, Card(L_{a_k}^{Jacc}) + \rho_{a_k} + P_s^{a_k})$
- 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: **else**
- 21: $Grant(a_j, a_k)$
- 22: **end if**
- 23: **end if**
- 24: **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**
- 25: Sélectionner un agent tel que, $Max(\rho_{a_j}, \rho_{a_k}), Max(Aut_{a_j}^t, Aut_{a_k}^t)$ et $Max(P_s^{a_j}, P_s^{a_k})$.
- 26: **end if**
- 27: **end if**

3.3.1.2 Mécanisme de demande de garants (*Grant*)

Nous donnons de manière succincte l'ensemble des étapes de ce mécanisme :

Étape 1 : Demande de garants pour des agents en conflits.

Définition 3.11 Un agent a_i est garant pour un agent a_j , si a_i se fonde sur ses alliances, sur ses dépendances avec a_j ou sur la fiabilité de a_j pour indiquer sa préférence pour a_j par rapport aux agents qui sont en conflit avec a_j .

Un agent ne peut être garant d'un autre agent que s'il est dans au moins l'une des listes de sondage pour l'offre de formation d'une coalition C où les agents sont en conflits. Cette contrainte permet d'éviter les cas où, un agent qui n'a pas accepté la demande de formation de C ne déstabilise volontairement le processus de résolution de conflits et par là, entraîner l'échec de la formation de C .

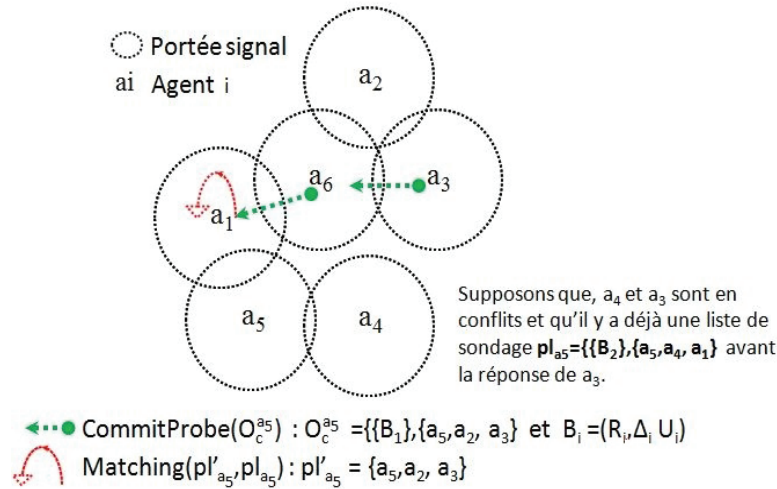


FIGURE 3.10 – Détection de conflits entre deux agents par l'agent a_1 suite à la réception de la réponse de a_3 . En supposons que a_3 et a_4 sont en conflits.

Étape 2 : Risque pour un agent qui est garant ou qui a un garant.

Si un agent qui a un garant quitte la coalition ou l'environnement avant la fin de l'exécution de la tâche, sa fiabilité ainsi que celle de ses garants sont réduites par les agents qui sont restés dans la coalition. De plus, chaque agent garant doit supporter le coût supplémentaire dû à la disparition d'un agent dont il est le garant, si la coalition est devenue instable. Cette condition peut mener

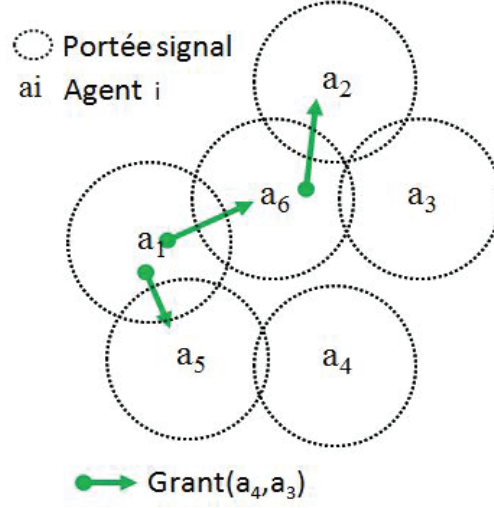


FIGURE 3.11 – a_1 envoie une demande pour la recherche de garants des agents a_3 et a_4 qui sont des agents en conflits. À noter que les agents en conflits ne sont pas contactés.

à une réduction de l'utilité d'un agent garant, car l'utilité d'un agent dépend des coûts qu'il doit supporter (cf. équation 3.1).

Étape 3 : Calcul du poids des garants.

Ce calcul se fonde sur le nombre d'alliés qu'un agent a parmi ses agents garants et sur la moyenne des fiabilités des agents garants.

Soit $AllianceSet^{a_i}$ l'ensemble des agents alliés d'un agent a_i . Soit AG l'ensemble des agents garants de l'agent a_i . Le poids des garants de l'agent a_i est égal à :

$$\frac{(AllianceSet^{a_i} \cap AG) + \sum_{a_x \in AG} \rho_{a_x}}{Card(AG)} \quad (3.6)$$

Plus cette valeur est grande, plus l'agent est préféré.

Étape 4 : Proposition d'une nouvelle liste de sondage sans conflits.

Soit $pl_{a_i}, pl'_{a_i}, pl''_{a_i}, \dots$ un ensemble de listes de sondage pour une même offre $O_c^{a_i}$ de formation de coalition. Proposer une liste de sondage unique dans ce cas signifie, formellement :

$pl_{a_i} \cup pl'_{a_i} \cup pl''_{a_i} \cup \dots : pl_{a_i} \cap pl'_{a_i} \cap pl''_{a_i} \cap \dots = \emptyset$ (fusion de toutes les listes de sondage telle qu'il n'y ait pas de répétition des agents dans la liste finale) et

$\forall a_x \in pl_{a_i}, a_y \in pl'_{a_i}, a_z \in pl''_{a_i} \dots \forall B_i \in O_c^{a_i}, G_{a_x} \neq G_{a_y} \neq G_{a_z}$ (avoir une liste finale sans agents en conflits).

Étape 5 : Attente d'amendements justifiés.

Il peut arriver qu'un agent a_j demande à ce qu'un autre agent soit retenu dans la liste de sondage pl_{a_i} à cause de ses dépendances, de ses *alliances*, ... avec ce dernier. Pour ce faire, l'agent a_j qui a émis l'amendement sur la pl_{a_i} , doit indiquer sa probabilité de stabilité $P_s^{a_j}$ et accepter d'être garant pour l'ensemble des agents qu'il demande d'ajouter dans pl_{a_i} . Si aucun amendement n'est fait, pl_{a_i} est validée comme une coalition. Dans le cas où, des amendements ont été apportés sur pl_{a_i} , les agents choisissent une nouvelle liste de sondage pl'_{a_i} qui est la plus connexe et qui a le plus d'agents qui ont des *alliances non-dominées* dans pl'_{a_i} .

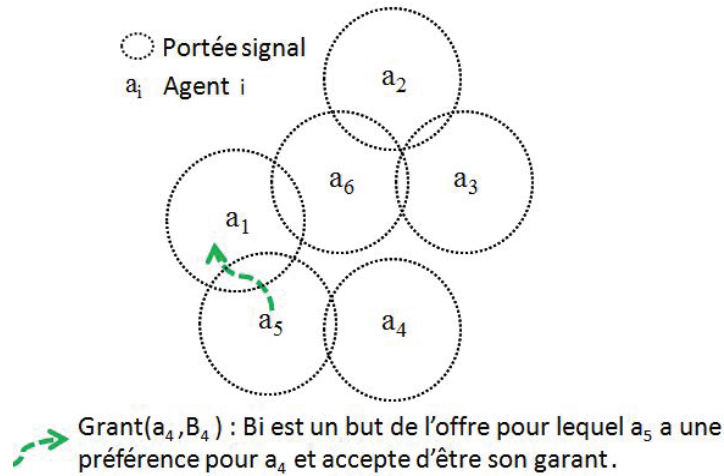


FIGURE 3.12 – Dans cette figure, nous supposons que a_5 se porte garant de l'agent a_4 , mais que a_3 n'a pas de garant.

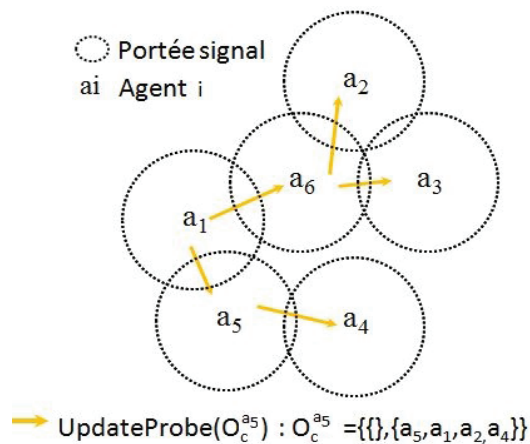


FIGURE 3.13 – a_1 propose une nouvelle liste suite à la réponse de a_5 . C'est finalement a_4 qui est inclus dans la liste de sondage pl_{a_5} en attendant des amendements sur cette nouvelle liste pl_{a_5} .

Dans la section ci-dessous, nous faisons une analyse de notre mécanisme de formation de coalitions *S-NRB* afin de prouver son efficacité, sa convergence et sa complexité.

3.3.2 Analyse de S-NRB

Nous donnons un ensemble de propriétés de *S-NRB* qui mène à l'existence de coalitions *A-core*. Nous montrons que les négociations multilatérales de *S-NRB* convergent toujours malgré les incertitudes et la dynamique du contexte.

Lemme 1 *S-NRB est un mécanisme décentralisé et asynchrone qui permet des prises de décisions décentralisées.*

Démonstration. Soit A un ensemble d'agents $a_i : i \in [1, |A|]$. $\forall a_i \in A : \exists O_c^{a_i}$, a_i peut utiliser *S-NRB* sans aucun ordonnanceur si $\vartheta_{a_i}^t \neq \emptyset$, c'est-à-dire, a_i peut envoyer un message $Probe(O_c^{a_i})$ à tout $a_j \in \vartheta_{a_i}^t$ dès qu'il a une offre $O_c^{a_i}$. $\forall a_i \in A, a_i \in pl_{a_i} : \vartheta_{a_i}^t \neq \emptyset$, alors $\exists a_j \in A$ qui peut mettre à jour la liste de sondage pl_{a_i} grâce à un message $UpdateProbe(O_c^{a_i})$ ou qui peut négocier sa participation à la liste pl_{a_i} à valider comme une coalition sans l'existence d'un ordonnanceur ou l'attente d'un tour de rôle. Ce qui prouve notre lemme. $\square$

Lemme 2 *Qu'une possibilité de former une coalition core stable existe ou non, S-NRB termine toujours sans blocage.*

Démonstration. Soit $U_{a_j}^t$ l'utilité de a_j à l'instant t . $\forall Probe(O_c^{a_i})$ d'un agent a_i , un agent $a_j \neq a_i$ ne diffuse ce message de sondage que si $U_{a_j}^{t-1} \leq U_{a_j}^t$ et $\vartheta_{a_j}^t \neq \emptyset$, 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 . De plus, le choix entre a_i et a_j se fait en cherchant celui qui est le plus fiable des deux pour la coalition, suite à une demande de garants. Cette méthode évite le cas où les agents 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. $\square$

Lemme 3 *S'il est possible de former une coalition core stable, S-NRB permet aux agents de converger vers celle-ci.*

Démonstration. Soit $U_{a_i}^t$ l'utilité de a_i à l'instant t et $U_{a_j}^t$ l'utilité de a_j à l'instant t .

$\forall B_i \in G_c$ et $B_i \in O_c^{a_i}$,

si $\exists a_i, a_j \in A : R_{a_i} \in B_i, Aut_{a_i}^t \neq 0, U_{a_i}^{t-1} \leq U_{a_i}^t$ et $R_{a_j} \in B_i, Aut_{a_j}^t \neq 0, U_{a_j}^{t-1} \leq U_{a_j}^t$,

alors, a_i (respectivement a_j) va atteindre un accord en utilisant ses dépendances $L_{a_i}^{Net}$ (respectivement $L_{a_j}^{Net}$). Cela se vérifie à la ligne 1 de l'algorithme 6.

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.

Tout accord entre a_i et a_j peut être contrôlé par chaque agent de leur liste de sondage grâce au mécanisme de demande de garants comme à la ligne 21 de l'algorithme 6.

Si la liste de sondage pl_{a_i} est validée comme une coalition C , cela signifie que :

- (1) $\forall a_i \in pl_{a_i}, \exists a_j \in pl_{a_i} : a_j \in \gamma_{a_i}^{Net}$ ou $a_j \in H_{a_i}^{Net}$ ou $a_j \in S_{a_i}^{Net}$;
- (2) $\forall a_i \in pl_{a_i}, U_{a_i}^{t-1} \leq U_{a_i}^t$;
- (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), réduire son utilité et être considéré comme non fiable par les agents avec lesquels il a des dépendances ou qui sont ses garants. Ce qui prouve notre lemme. $\square$

Théorème 2 *S-NRB permet toujours la convergence vers une coalition core stable si celle-ci existe.*

Démonstration. Le lemme 1, prouve que S-NRB fonctionne même si des agents deviennent aléatoirement indisponibles et que S-NRB a un contrôle distribué. Le lemme 2 montre que S-NRB termine son exécution et le lemme 3 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. $\square$

Lemme 4 *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 accepté de partager la coalition C .*

Démonstration. $\forall a_j \in A$, s'il reçoit une offre $O_c^{a_i}$, cela signifie que, $a_j \in \vartheta_{a_i}^t$ ou $\exists \{a_x, a_y, \dots\} \in A : a_j \in L_{a_i}^{Net}$ et $a_i \in L_{a_j}^{Net}$.

Ainsi, si a_i et $a_j \in pl_{a_i}$ alors, $a_j \in \vartheta_{a_i}^t$ ou $\exists \{a_x, a_y, \dots\} \in pl_{a_i}$ car dans notre approche a_x ne diffuse pl_{a_i} que s'il appartient à pl_{a_i} (cf. lignes 3 - 5 algorithme 5) ou s'il existe un message de mise à jour de pl_{a_i} ($UpdateProbe(O_c^{a_i})$). Dans ce dernier cas, si un agent a_j souhaite atteindre un but $B_i \in O_c^{a_i}$ il doit appartenir à $\vartheta_{a_i}^t$ ou à $\vartheta_{a_x}^t : a_x \in pl_{a_i}$. Ce qui prouve notre lemme. $\square$

En supposant que les agents sont déployés dans des composants électroniques (e.g. capteurs, Smart-phones, ..) qui ont des ressources énergétiques limitées, alors, le coût des communications est plus important lors des transmissions de messages. C'est pourquoi, S-NRB tient compte de ces contraintes afin de borner dynamiquement les messages transmis durant les négociations.

Lemme 5 *Le coût des messages est borné et polynomial pour tout agent jusqu'à la convergence de S-NRB.*

Démonstration. Soit $\bar{Msg}$ le coût des messages de S-NRB. $\forall a_j \in A$, si a_j rejette l'offre $O_c^{a_i}$, il ne répond qu'avec 1 message à l'agent par qui il a reçu $O_c^{a_i}$ ou au plus il répond avec $Card(\theta_{a_j}^t) : \theta_{a_j}^t \neq \emptyset$ messages pour informer tous les agents voisins qui lui ont envoyé l'offre $O_c^{a_i}$. Ainsi, $1 \leq \bar{Msg} \leq Card(\theta_{a_j}^t)$.

$\forall a_j \in A$, si a_j accepte l'offre $O_c^{a_i}$, il n'envoie qu'un message de broadcast dans son voisinage $\theta_{a_j}^t \neq \emptyset$ sauf à l'agent par qui il a reçu l'offre $O_c^{a_i}$. Ce message de broadcast a aussi pour objectif de diffuser l'offre $O_c^{a_i}$ au reste des agents. De plus l'agent a_j doit envoyer une réponse après réception des réponses de ses agents voisins de $\theta_{a_j}^t$. Notons que, a_j doit fusionner sa réponse avec celles de ses voisins avant d'envoyer ce message de broadcast, cela, afin de résoudre en local les conflits entre les agents (cf. algorithme 6) et de construire une liste de sondage pl_{a_i} unique. Ainsi, $\bar{Msg} = 2$. Cependant, en cas de nécessité de résolution de conflits, le coût des messages devient $\bar{Msg} = 2 + 3 = 5$. Cela est dû au fait que, durant la résolution des conflits, il y a un premier message de notification des conflits à résoudre et si nécessaire un message de demande de garants des agents en conflits (cf. *Grant* ligne 13 algorithme 6) et un autre message pour diffuser le résultat de la résolution des conflits. Donc, pour N agents en conflits, nous avons $2 \leq \bar{Msg} \leq 1 + 3N$.

Cependant, $\forall a_j \in A$, si a_j décide de faire une contre-proposition (counter) pour une offre $O_c^{a_i}$, il doit envoyer un message qui spécifie l'ensemble des buts de l'offre $B_i \in O_c^{a_i}$ concernés par la contre-proposition. Une contre-proposition d'un agent a_j peut être acceptée ou rejetée. Pour ces deux cas, le coût des messages reste identique à ce qui est décrit précédemment. Ainsi, si a_j fait une contre-proposition avant de rejeter l'offre, alors on a : $1 \leq \bar{Msg} \leq Card(\theta_{a_j}^t) + 2$ sinon si c'est une contre-proposition avant d'accepter l'offre, alors on a : $1 \leq \bar{Msg} \leq 2 + 3N$.

Sachant que le nombre d'agents de l'environnement de déploiement n'est pas infini, $\forall a_j \in A$, $Card(\theta_{a_j}^t)$ et N sont bornés. De là nous déduisons que, pour tous les cas énoncés précédemment, le coût des messages est borné et $\bar{Msg}$ est polynomial. Ce qui prouve notre lemme. $\square$

Lemme 6 *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.*

Démonstration. Considérons les agents a_i , a_j et a_k tels que $a_j \in \theta_{a_i}^t$ et $a_k \in \theta_{a_j}^t$.

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_j et a_k ne seront pas en conflits car si a_j reçoit l'offre $O_c^{a_i}$, il ajoute son identifiant dans la liste de sondage pl_{a_i} et supprime l'information concernant $G_{a_j} \equiv B_i \in O_c^{a_i}$ avant d'envoyer l'offre $O_c^{a_i}$ à $a_k \in \vartheta_{a_i}^t$.

Ainsi, si a_k reçoit $O_c^{a_i}$ alors l'information $G_{a_k} \equiv B_i \notin O_c^{a_i}$ 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, $a_j, a_k \in \vartheta_{a_i}^t$ alors, les deux agents vont recevoir l'offre $O_c^{a_i}$ 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 6. 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. $\square$

Théorème 3 *S-NRB permet une auto-stabilisation des coalitions core stables.*

Démonstration. Le lemme 3 prouve la convergence de S-NRB vers des coalitions *core stables* où aucun agent n'est motivé à quitter sa coalition. Le lemme 4 montre que, chaque événement 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 2 et 5 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 (cf. lemme 1) 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. Par ailleurs, les agents doivent posséder un ensemble de garants (cf. lemme 6). 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. $\square$

Nous donnons de manière succincte, dans la section suivante, les règles supplémentaires de fonctionnement que les agents observent durant les phases de coordination dans leur environnement .

3.3.3 Règles générales

Nous avons déjà eu à définir dans les sections précédentes, les règles propres à chaque étape de notre mécanisme de formation de coalitions. C'est pourquoi, dans ce qui suit, nous ne donnons que les règles communes à toutes les étapes.

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 $\forall a_i \in C, u_c^{a_i} \geq u_c^{a_j} : C' = C \cup a_j$. ■

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, $\forall a_i \in C, u_c^{a_i} \geq u_c^{a_j} : C' = C \cup a_j$,
- (2) a_j peut conseiller a_i s'il existe une *alliance* Al_{a_i, a_j} ou s'il existe une *alliance* Al_{a_i, a_k} telle que, $a_k \in C$ puisse se porter garant (*grant*) de 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 $\forall a_i \in C, u_c^{a_i} \geq u_c^{a_j} : C' = C \cup a_j$.

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. ■

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.

$$1 \leq TTL \leq \left\lfloor \frac{S_z}{2 * Y} \right\rfloor \quad (3.7)$$

où S_z est la taille de l'environnement couvert par la tâche, Y la portée du signal du composant qui héberge l'agent qui a initié le message (e.g. Bluetooth 100 mWatt, $Y \leq 100$ 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. ■

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. ■

Dans la section suivante, nous donnons des propriétés supplémentaires de notre mécanisme de formation de coalitions S-NRB.

3.4 ÉVALUATIONS DE PERFORMANCES DE NOTRE MÉCANISME S-NRB

La section évaluations analytiques a pour objectif d'étudier la complexité de notre mécanisme en matière de coût de messages pour un ensemble de distributions de probabilités fréquemment utilisées pour estimer l'efficacité des mécanismes de formation de coalitions. La partie expériences cherche à prouver - la convergence des processus de négociation vers des coalitions *core stables*, et - l'existence des propriétés d'auto-stabilisations des coalitions formées (coalitions *A-core*). De même, les expériences ont été effectuées lorsque les conflits des agents et les tâches suivent un ensemble de distributions de probabilités.

3.4.1 Évaluations analytiques

Nous donnons par simulations analytiques le coût des messages $\bar{M}sg$ jusqu'à la convergence de notre mécanisme de formation de coalitions S-NRB. Nous effectuons cette étude en considérant les conflits des agents dus à leurs préférences et à leurs dépendances suivent une distribution de probabilités - Uniforme ($conflits \sim U(a, b)$ où $a=0$ et $b=10$), - Normale ($conflits \sim N(\mu, \sigma^2)$ où $\mu=0.5$ et $\sigma=0.1$) ou - de Poisson ($conflits \sim P(\lambda)$ où $\lambda=0.5$). Ici, $\bar{M}sg$ est le coût des messages.

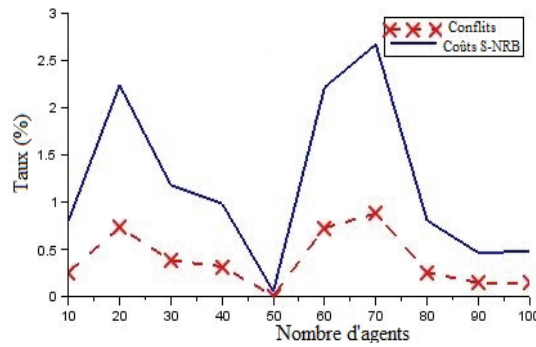


FIGURE 3.14 – Coût des messages jusqu'à la convergence de S-NRB si $conflits \sim U(a, b)$ où $a=0$ et $b=10$.

Les conflits suivent une distribution uniforme $U(a, b)$ de paramètres $a=0$ et $b=10$. La Figure 3.14 montre qu'à un instant donné, $Max(\bar{M}sg) \approx \bar{M}sg * 2.5$ si le pourcentage de conflits est de 100% et $Max(\bar{M}sg) < \bar{M}sg * 1.5$ si le pourcentage de conflits est de 50%. Si le pourcentage de conflits est nul, $\bar{M}sg$ est très faible.

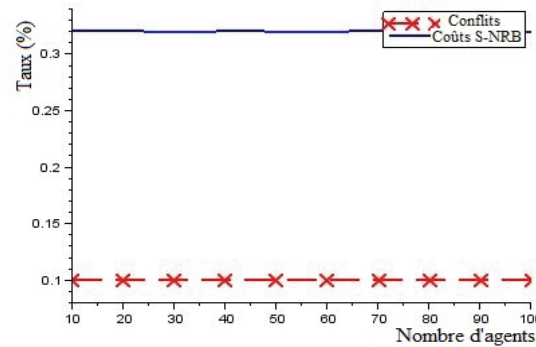


FIGURE 3.15 – Coût des messages jusqu'à la convergence de S-NRB si $conflits \sim N(\mu, \sigma^2)$ où $\mu=0.5$ et $\sigma=0.1$.

La Figure 3.15 montre que, le pourcentage des conflits pour un instant donné ne dépasse jamais 10% et $Max(\bar{M}sg) \approx \bar{M}sg * 0.32$ si les conflits suivent une distribution de probabilités Normale $N(\mu, \sigma^2)$ de paramètres $\mu=0.5$ et $\sigma=0.1$.

La Figure 3.16 montre que, le pourcentage des conflits ne dépasse jamais 6% et $Max(\bar{M}sg) \approx \bar{M}sg * 0.16$ si les conflits suivent une distribution de probabilités de Poisson $P(\lambda)$ de paramètre

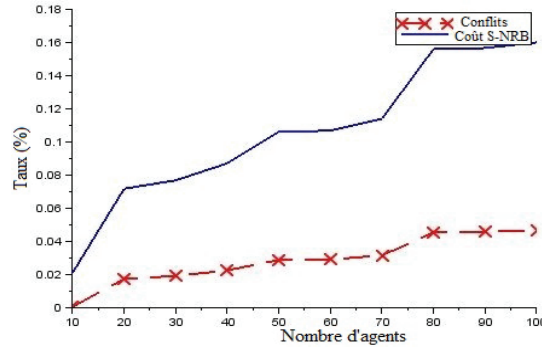


FIGURE 3.16 – Coût des messages jusqu'à la convergence de S-NRB si $\text{conflits} \sim P(\lambda)$ où $\lambda=0.5$.

$\lambda=0.5$.

Ces résultats de simulations analytiques montrent que, S-NRB converge toujours sans un coût exponentiel et le coût des messages ne dépend pas du nombre d'agents mais seulement du nombre de conflits avant la convergence de S-NRB. De plus, le pourcentage de conflits pour chaque instant t donné est borné pour les différentes distributions de probabilités ci-dessus.

3.4.2 Expériences

3.4.2.1 Configurations et méthodologie

Pour l'ensemble des expériences que nous avons menées, une tâche est assignée à un agent unique qui doit utiliser ses stratégies de négociation afin d'établir une coalition stable pour l'exécution de cette tâche. La stratégie d'un agent durant une phase de négociation peut porter sur la manière d'utiliser ou d'établir ses *alliances*, la manière d'utiliser ses dépendances, et la manière de raisonner sur les contraintes et les récompenses associées à une tâche.

De même que précédemment, nous effectuons ces simulations quand les indisponibilités des agents, les conflits dus aux préférences et dépendances des agents et les états des tâches suivent des distributions de probabilités de loi Uniforme, Normale ou de Poisson.

Dans notre simulateur développé sous Java, chaque agent est autonome et égoïste avec des dépendances natives (un agent a des obligations envers un ensemble d'agents dès son déploiement) ou non. Aléatoirement, un agent peut initier un processus de formation de coalitions, être libre de tout engagement envers les autres agents ou devenir indisponible. La dynamique de la localisation et des états des tâches est générée 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 dans l'environnement est compris entre 3 et 8 tâches où les propriétés de chaque tâche sont générées aléatoirement. Les simulations sont effec-

tuées dans une machine Intel i7 (4 GHZ) avec 4GB de RAM.

Pour maximiser son utilité et atteindre son but, chaque agent a_i tient compte de ses caractéristiques (ressources, autonomie d'énergie, historique, voisinage, utilité, dépendances) et des contraintes liées aux tâches à exécuter.

Chaque négociation est décentralisée et a une durée au-delà de laquelle elle est abandonnée s'il n'y a pas d'accord définitif entre les agents. Cette durée n'est pas fixée à l'avance, car elle dépend de la dynamique des agents et des tâches sans oublier que les *alliances* proposées et les *récompenses* associées aux tâches ont une durée de validité limitée dans le temps.

Les ressources disponibles dans l'environnement de déploiement des agents évoluent dynamiquement car elles dépendent de la disponibilité des agents et des tâches en cours d'exécution. À cause de la dynamique des ressources disponibles ou requises pour une tâche, comme pour le chapitre précédent, nous faisons nos simulations en considérant les *ratios*. Pour rappel, un *ratio* est égal au nombre total de ressources disponibles dans l'environnement de déploiement sur le nombre total de ressources nécessaires pour former les coalitions requises pour exécuter les tâches.

Par exemple, dans un environnement *Net*, supposons qu'il y a 12 agents de même caractéristiques. Si dans *Net*, l'objectif est de former deux coalitions où la première coalition requiert 9 agents et où la seconde coalition requiert 6 agents, alors nous avons, $ratio = \frac{12}{9+6} = \frac{12}{15} = 0.8$.

Pour simuler l'indisponibilité des agents, nous désactivons aléatoirement un ensemble d'agents. De plus, cette action a pour objectif de simuler les changements de topologies réseaux des agents et de tester les propriétés d'auto-stabilisation des coalitions formées. Les messages sont librement échangés entre les agents sans les notions de tour de rôle et sans l'intervention d'un agent organisateur. Ces communications respectent le principe du *non-return broadcast* et du *TTL*. Pour obtenir des valeurs moyennes, nous répétons 10 fois chaque simulation.

3.4.2.2 Résultats

Nous étudions ici, la capacité des coalitions *core stables* formées à se stabiliser dynamiquement (coalitions *A-core*).

Les résultats des simulations illustrés par ces figures montrent que, si une coalition est *core stable*, elle reste *core stable* même s'il y a des changements dynamiques des *ratios* dans le temps qui impacte la quantité de ressources requises pour continuer l'exécution d'une tâche.

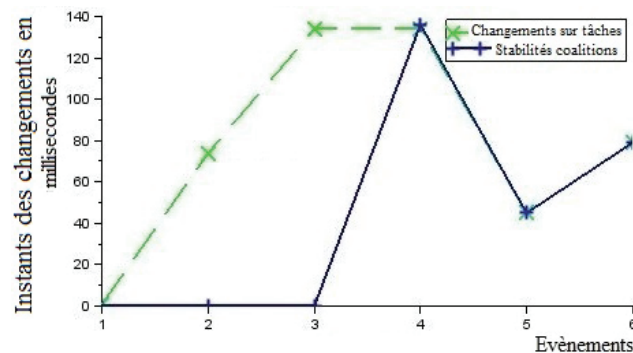


FIGURE 3.17 – Auto-stabilisation des coalitions core stables formées avec notre mécanisme S-NRB si les changements d'états des tâches et la disponibilité des agents suivent une distribution de probabilités Uniforme $U(a,b)$ où $a=0$ et $b=10$. Pour ces six changements visibles sur la figure, les ratios sont de : 1.42, 1.66, 2, 3.33, 1.66 et 2.

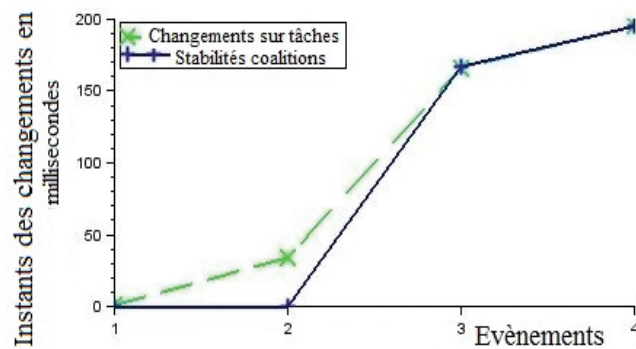


FIGURE 3.18 – Auto-stabilisation des coalitions core stables formées avec notre mécanisme S-NRB si les changements d'états des tâches et la disponibilité des agents suivent une distribution de probabilités Normale $N(\mu, \sigma^2)$ où $\mu=0.5$ et $\sigma=0.1$. Pour ces quatre changements visibles sur la figure, les ratios sont de : 0.90, 3.33, 3.33, 2.50.

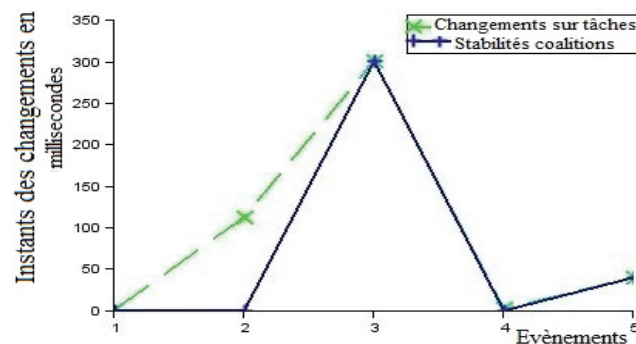


FIGURE 3.19 – Auto-stabilisation des coalitions core stables formées avec notre mécanisme S-NRB si les changements d'états des tâches et la disponibilité des agents suivent une distribution de probabilités de Poisson $P(\lambda)$ où $\lambda=0.5$. Pour ces cinq changements visibles sur la figure, les ratios sont de : 3.33, 1, 2.5, 2, 1.66.

CONCLUSION DU CHAPITRE

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. Ce mécanisme s'applique même en présence d'incertitudes sur : - les dépendances et les conflits qui peuvent apparaître entre les agents pour une formation de coalition, - les préférences et les disponibilités des agents, et - les coalitions à former pour les tâches stochastiques à exécuter dans un contexte ouvert (les agents peuvent être indisponibles à tout moment). À cause de ces incertitudes et des contraintes sur les tâches et les agents, nous ne supposons aucune connaissance a priori sur les coalitions stables à former. Nous considérons 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.

Nous avons proposé un mécanisme de formation de coalitions *core stables* que nous avons appelé S-NRB. Nous avons, principalement, prouvé sa convergence vers des coalitions *core stables* qui se stabilisent dynamiquement dans le temps (auto-stabilité) en cas de disparition d'un ensemble d'agents ou de changements sur les contraintes des tâches. Notre mécanisme S-NRB utilise : 1- un ensemble de lois de probabilité pour prédire les événements qui peuvent changer les états des tâches ou pour prédire les indisponibilités des agents, 2- les principes de la théorie des jeux pour identifier les coalitions *core stables*. L'analyse théorique et les simulations analytiques et expérimentales montrent le fonctionnement de notre mécanisme S-NRB. En effet, en distribuant le contrôle et les prises de décisions lors des processus de formation de coalitions, ce mécanisme permet de surmonter les incertitudes et les contraintes du contexte considéré afin de trouver les coalitions *A-cores*.

4

EXTENSION DE LA FORMATION DE COALITIONS A-CORE : P-NRB

Ce chapitre se positionne dans un contexte multi-agents hétérogène et distribué qui est caractérisé par la dynamique et les incertitudes de l'environnement de déploiement. Des incertitudes liées d'une part aux dépendances, aux préférences, aux disponibilités et aux conflits qui peuvent arriver entre les agents et, d'autre part, au caractère stochastique des tâches à exécuter. Dans ce contexte incertain, nous ne considérons aucune connaissance qui peut nous permettre de déterminer a priori les coalitions stables à former pour l'exécution de ces tâches. Spécifiquement, nous cherchons à étendre nos travaux présentés dans le chapitre précédent qui ont conduit à la proposition du mécanisme de formation de coalitions S-NRB (Sequential Non-Return Broadcast). Notre nouveau mécanisme de formation de coalitions est parallèle, asynchrone, décentralisé et anytime (trouvent des coalitions stables en cas d'interruption prématurée du processus de formation de coalitions). Nous appelons P-NRB (Parallel Non-Return Broadcast) ce nouveau mécanisme de formation de coalitions A-core. P-NRB combine les principes de la théorie des jeux, avec le formalisme des chaînes de Markov (MDP) et les lois de probabilités. Le mécanisme converge graduellement vers les coalitions stables qui maximisent le bien-être social des agents. Les analyses et les simulations que nous avons menées, montrent l'efficacité du mécanisme P-NRB. De plus, il améliore ou étend dans la plupart des cas, notre mécanisme S-NRB. Par étend, nous comprenons que, P-NRB fournit des propriétés qu'on ne peut pas obtenir avec S-NRB.

4.1 OBJECTIFS ET APPROCHE GÉNÉRALE

Nous gardons les mêmes défis de formation de coalitions du chapitre précédent. Cela signifie que, l'environnement considéré dans ce chapitre, est une situation de gestion de sinistres à l'aide d'agents intelligents. Les agents sont déployés sur des plateformes diverses comme des robots autonomes, des dispositifs personnels et des capteurs. Les agents ont pour but de faciliter l'échange d'informations cohérentes, d'aider à une répartition efficace des intervenants et d'éviter une amplification des risques en pertes humaines et matérielles. Les tâches à exécuter sont considérées comme nécessitant une collaboration entre plusieurs agents (intervenants), par la formation de coalitions. Plus en détail, nous proposons un mécanisme de formation de coalitions A-core qui est adaptatif, décentralisé et asynchrone et qui améliore ou étend dans la plupart des cas notre mécanisme S-NRB présenté dans le chapitre précédent. Par P-NRB étend S-NRB, nous voulons dire que, P-NRB fournit des propriétés qu'on ne peut pas obtenir avec S-NRB. Nous appelons notre nouveau mécanisme, P-NRB (Parallel Non-Return Broadcast). P-NRB est fondé sur la négociation multilatérale dans laquelle le contrôle et les prises de décisions des agents sont décentralisées et se font en parallèle. Les différences fondamentales par rapport à S-NRB est que, P-NRB est capable de fournir comme résultat, des coalitions stables en cas d'interruption prématurée du processus de formation des coalitions (propriété *anytime*) et qu'en cas de convergence, P-NRB retourne toujours les meilleures coalitions A-core comme résultat.

Pour trouver les coalitions A-core, le mécanisme P-NRB présente une nouvelle combinaison de techniques diverses. Particulièrement, il exploite certaines propriétés des lois de probabilité pour modéliser la dynamique des événements des tâches et la disponibilité des agents ; il utilise des méthodes de la théorie des jeux pour trouver dynamiquement les coalitions requises en profitant des dépendances des agents et de la structure du réseau où les agents sont déployés ; il se fonde aussi sur le formalisme des *Processus de Décision Markov* (MDP) pour examiner et adapter dynamiquement les coalitions A-core à former. Cette combinaison livre une nouvelle solution de formation de coalitions et aborde le problème de la dynamique. Il permet d'apporter des réponses aux problèmes qui n'ont pas été résolus par notre mécanisme S-NRB proposé dans le chapitre précédent.

4.2 CONCEPTS DE NOTRE MODÈLE

Nous reprenons les mêmes définitions du chapitre précédent en ce qui concerne les notions de tâches, de coalitions, de coalitions A-core, de but pour un agent ou pour une coalition et d'offre composée. Nous notons encore par $A = \{a_1, \dots, a_n\}$, l'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. De plus, par C nous définissons une coalition telle que $C=\{A_c, G_c, T_c, V_c\}$, où $A_c \subset A=\{a_1, a_2, \dots, a_k\}$ est un ensemble d'agents qui partage la coalition C et qui a comme ensemble de buts $G_c \subseteq \{G_{a_i} : i \in N, a_i \in A_c\}$. T_c définit un ensemble de tâches stochastiques à exécuter par la coalition C et V_c est la récompense associée à la coalition C après l'exécution de l'ensemble T_c .

Pour un agent, nous maintenons les mêmes propriétés que celles du chapitre précédent. Nous rappelons que les paramètres essentiels pour un agent a_i s'exprime à un instant t sous la forme :

$$\{R_{a_i}, Aut_{a_i}^t, Hs_{a_i}, \vartheta_{a_i}^t, U_{a_i}, L_{a_i}^{Net}\}.$$

Pour un agent a_i , l'information R_{a_i} décrit sa ressource et $Aut_{a_i}^t$ son autonomie énergétique à l'instant t . Hs_{a_i} représente l'historique de l'agent a_i et s'exprime toujours comme décrit dans le chapitre précédent, c'est-à-dire, en fonction des alliances, des fiabilités et des probabilités de stabilité des agents. La fonction d'utilité U_{a_i} d'un agent a_i à l'instant t se note toujours sous la forme $U_{a_i}^t$. $L_{a_i}^{Net}$ définit le niveau de dépendance de l'agent a_i par rapport aux autres agents de son voisinage $\vartheta_{a_i}^t$ dans une structure réseau notée Net .

Avec ces exigences, nous présentons dans la section suivante, notre nouveau mécanisme de formation de coalitions stables appelé *P-NRB (Parallel Non-Return Broadcast)*. Ce mécanisme cherche une convergence rapide vers des coalitions *core stables* pour permettre l'exécution des tâches stochastiques. Ce mécanisme vise aussi à trouver des coalitions *core stables* capables de se stabiliser dynamiquement dans le temps (propriété d'auto-stabilité) et à fournir des coalitions stables en cas d'interruption prématurée des négociations du processus de formation de coalitions (propriété *anytime*).

4.3 MÉCANISME DE FORMATION DE COALITIONS

De la même manière que dans le chapitre précédent, nous considérons que nous avons des offres composées. Formellement, une offre composée d'un agent a_i qui veut former une coalition C est : $O_c^{a_i} = \{\{B_i, B_j, B_k, \dots\}, \{pl_{a_i}\}\} : B_i = (R_i, \Delta_i, U_i)$.

B_i est un but à atteindre par une offre qui spécifie, une ressource R_i requise pour une tâche ou un ensemble de tâches T_c qui a induit la demande de formation de coalitions. Un but B_i contient aussi des contraintes Δ_i est une récompense U_i pour la formation de la coalition. Notre mécanisme utilise aussi une liste de sondage $pl_{a_i} = \{a_i, a_j, a_k, \dots\} : \forall a_x \in pl_{a_i} \in A$ qui permet de savoir à un instant t , l'ensemble des agents qui sont d'accord avec une offre donnée.

4.3.1 Notre mécanisme P-NRB

Notre mécanisme P-NRB est décrit par l'algorithme 7 ci-dessous. Pour plus de clartés, sa description est donnée sous la forme d'étapes comme suit :

Étape 1 : Génération d'offres et prises de décisions.

- Une offre d'un agent est diffusée avec un *TTL* en mode non-return broadcast où chaque agent incrémente de 1, la valeur du nombre de sauts (*Hop*) effectués par le message avant de le diffuser dans son voisinage (*vue*). Le *Hop* est différent du *TTL*, car le *Hop* permet de savoir le nombre de nœuds (agents) par où le message est passé alors que le *TTL* permet de limiter le nombre maximal de sauts qu'un message peut effectuer. Nous rappelons que dans nos considérations, un nœud est un composant qui héberge un agent (Personal Digital Assistant (PDA), Smart-phones, capteurs, ...).

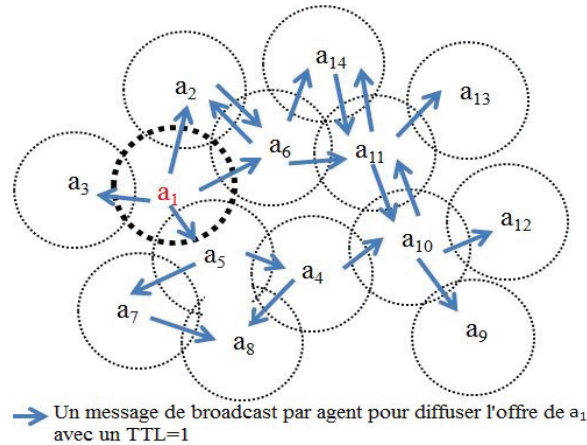


FIGURE 4.1 – Diffusion de l'offre. Dans ce cas, tout agent a_i qui reçoit d'un agent a_j un message de sondage pour une offre composée, il le diffuse dans son voisinage $\vartheta_{a_i}^t \setminus \{a_j\}$, c'est-à-dire, en mode non-return broadcast. Dans cette figure, l'agent a_1 est l'initiateur de l'offre composée et nous avons supposé des problèmes de transmission de messages entre les agents a_5 et a_8 au moment de la diffusion de cette offre.

- Chaque agent détermine son *rang* en évaluant l'expression :

$$rang = Hop - \lfloor \frac{Hop}{2} \rfloor * 2 \quad (4.1)$$

donc, $rang=0$ ou 1. La valeur initiale du *Hop* égale à 1 et est donnée par l'agent qui a initié l'offre.

Étape 2 : Sondage du voisinage du saut suivant.

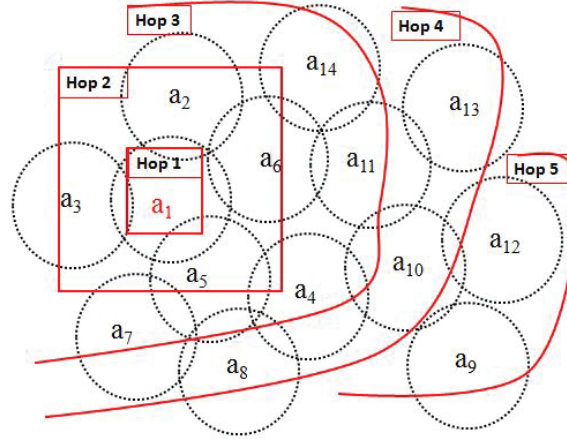


FIGURE 4.2 – Définition des rangs des agents, c'est-à-dire, chaque agent calcul son rang en utilisant l'équation 4.1.

Si $\text{rang}=1$ pour un agent, alors, il considère qu'il est responsable du sondage ($\text{probe}(O_c^{a_i})$) de ses voisins du saut suivant, sauf si ce voisinage est vide.

Définition 4.1 Nous appelons *niveau-discussions*, les niveaux de sondages qui vont impliquer pour une offre $O_c^{a_i}$, des négociations par voisinage direct entre les agents, une résolution de conflits locaux et des prises de décisions partielles.

La figure 4.3 présente un exemple de la formation des *niveau-discussions*. L'agent initiateur de l'offre crée un *niveau-discussion* avec les agents du $\text{Hop}=2$, que chaque agent du $\text{Hop}=3$ crée un *niveau-discussion* avec ses voisins de $\text{Hop}=4$, et ainsi de suite.

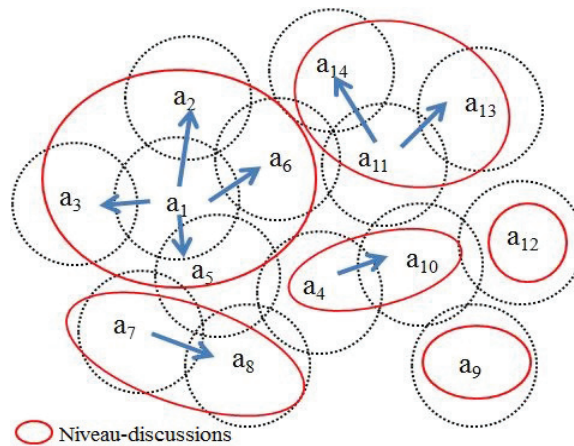


FIGURE 4.3 – Formation des *niveau-discussions* en se fondant sur les rangs des agents.

Pour éviter les chevauchements ou la duplication des *niveau-discussions*, un agent initiateur d'un *niveau-discussion* ne peut pas accepter d'être sondé par un autre agent pour la même offre. De plus,

un agent ne peut participer que dans un seul *niveau-discussion* et si un agent est déjà sondé, il ne peut pas initier un *niveau-discussion* pour la même offre. Il faut noter que, dans le cas où un agent doit choisir dans quel *niveau-discussion* participer, il se base sur ses préférences et ses dépendances.

Étape 3 : Gestion des conflits locaux à chaque *niveau-discussion*.

À cette étape, chaque *niveau-discussion* doit résoudre en local les conflits en matière de buts et de dépendances de ses agents, identifier le poids (nombre d'alliances, stabilité et connectivité) de chaque agent avant de diffuser une réponse résultant de la négociation et de la résolution de(des) conflit(s). La réponse du *niveau-discussion* peut être un *rejet* ou un *weak-accept*. Nous faisons la distinction entre le poids d'un agent dans un *niveau-discussion* et le poids du *niveau-discussion*.

Le poids d'un agent se fonde sur sa vue, sur ses alliances et sur ses dépendances (cf. équation 4.2), tandis que le poids du *niveau-discussion* se fonde sur des préférences multidimensionnelles des agents du *niveau-discussion*. Cette dernière utilise l'*intégrale de Choquet* qui est un opérateur d'agrégation de préférences (Grabisch 1995) (cf. équation 4.9).

- Une réponse *rejet* d'un *niveau-discussion* signifie qu'aucun de ses agents n'a accepté l'offre.
- Une réponse *weak-accept* d'un *niveau-discussion* signifie qu'il y a un ensemble d'agents noté *W-Set* qui sont d'accord avec l'offre et qui se proposent de participer à la coalition *C* de l'offre si leur utilité n'est pas réduite et si leurs préférences ne sont pas en conflits avec celles d'autres agents qui veulent rejoindre la coalition.

De manière formelle, un *weak-accept* d'un ensemble d'agents $W\text{-Set}=\{a_j, a_k, \dots\}$ d'un *niveau-discussion* signifie $\exists \{(G_{a_i}, a_i), (G_{a_j}, a_j), (G_{a_k}, a_k), \dots\} : \forall a_x \in W\text{-Set}, G_{a_x} \in O_c^{a_i}$. Pour différencier les réponses *weak-accept* simultanés qui peuvent émaner d'un *niveau-discussion* à cause de la possibilité d'offres simultanées, le premier champ $((G_{a_i}, a_i))$ du *weak-accept* est celui de l'agent a_i qui a initié l'offre $O_c^{a_i}$.

Le poids W_{a_i} d'un agent a_i est donné par :

$$W_{a_i} = \begin{cases} \frac{\text{Card}(H_{s_{a_i}} \cap \theta_{a_i}^t)}{\text{Card}(\theta_{a_i}^t)} & \text{if } \gamma_{a_i}^{\text{Net}} = \emptyset : \gamma_{a_i}^{\text{Net}} \subset L_{a_i}^{\text{Net}} \\ W_{\gamma_{a_i}}^{\text{Net}} & \text{if } \gamma_{a_i}^{\text{Net}} \neq \emptyset : \gamma_{a_i}^{\text{Net}} \subset L_{a_i}^{\text{Net}} \end{cases} \quad (4.2)$$

En présence de conflits dans un *niveau-discussion*, les agents utilisent leur poids et leur fiabilité (cf. équation 2.3) afin de trouver un accord. Dans ce cas, les agents qui ont les poids les plus élevés et qui sont plus fiables sont sélectionnés. Si ces deux paramètres ne permettent pas de décider, la décision est fondée sur les probabilités de stabilité les plus élevées. Notons qu'un agent ne peut pas mentir sur sa fiabilité, car elle peut être vérifiée par les agents de son *niveau-discussion*. De même,

un agent ne peut pas mentir sur son poids, car cela dépend de ses alliances et de ses dépendances avec les agents de son *niveau-discussion*.

Le poids du *niveau-discussion* n'est important que si une réponse *weak-accept* de l'offre existe. Ce poids tient compte seulement des préférences des agents de *W-Set* c'est-à-dire les agents qui ont répondu avec un *weak-accept* à l'offre. Pour évaluer le poids du *niveau-discussion*, les étapes sont les suivantes :

- Chaque agent évalue son modèle de préférences multidimensionnelles pour les agents *W-Set* du *niveau-discussion* en fonction de son utilité et des critères (fiabilité, stabilité, poids) de ces agents. Ce modèle de préférences multidimensionnelles pour un agent a_i sur un ensemble d'agents *W-Set* en fonction d'un ensemble de critères p , est une matrice $\prod_{a_i}^p$ qui s'exprime comme suit :

$$\prod_{a_i}^p = \begin{pmatrix} Y_{a_i,a_j}^1 & Y_{a_i,a_k}^1 & \dots & Y_{a_i,a_n}^1 \\ Y_{a_i,a_j}^2 & Y_{a_i,a_k}^2 & \dots & Y_{a_i,a_n}^2 \\ \vdots & \vdots & \ddots & \vdots \\ Y_{a_i,a_j}^p & Y_{a_i,a_k}^p & \dots & Y_{a_i,a_n}^p \end{pmatrix} \quad (4.3)$$

Chaque ligne x concerne les préférences suivant un critère $x \in p$. Y_{a_i,a_j}^1 est la préférence de l'agent a_i à l'égard de l'agent a_j en fonction du critère 1. Notons que, $Y_{a_i,a_i}^x = 0 \forall x \in p$.

Dans notre contexte, du fait que les agents sont égoïstes et ont des disponibilités aléatoires, les préférences de a_i pour a_j sont des informations privées et dépendent de la fiabilité ρ_{a_j} , de la probabilité de stabilité $P_s^{a_j}$ et de la différence entre le poids W_{a_j} de a_j et du poids W_{a_i} de a_i .

À cause des incertitudes et de la dynamique du contexte que nous considérons dans ce travail, nous ne supposons aucune connaissance a priori sur ces préférences des agents. Raison pour laquelle notre mécanisme P-NRB fournit les moyens à chaque agent de pouvoir calculer dynamiquement ses préférences.

Le premier critère est la fiabilité, le second est la probabilité de stabilité et le troisième est le poids de l'agent. Ainsi,

$$Y_{a_i,a_j}^1 = \begin{cases} \rho_{a_j} - \frac{\sum_{a_k \in A'} (\rho_{a_k})}{\text{Card}(A')} & \text{if } Y_{a_i,a_j}^1 > 0 \\ 0 & \text{if } Y_{a_i,a_j}^1 \leq 0 \end{cases} \quad (4.4)$$

$$Y_{a_i,a_j}^2 = \begin{cases} P_s^{a_j} - \frac{\sum_{a_k \in A'} (P_s^{a_k})}{\text{Card}(A')} & \text{if } Y_{a_i,a_j}^2 > 0 \\ 0 & \text{if } Y_{a_i,a_j}^2 \leq 0 \end{cases} \quad (4.5)$$

$$Y_{a_i, a_j}^3 = \begin{cases} W_{a_j} - W_{a_i} & \text{if } Y_{a_i, a_j}^3 > 0 \\ 0 & \text{if } Y_{a_i, a_j}^3 \leq 0 \end{cases} \quad (4.6)$$

- Chaque agent a_i applique l'intégrale de Choquet pour obtenir son vecteur de préférences Π_{a_i} pour une coalition avec chaque agent de son *niveau-discussion* appartenant à l'ensemble $W\text{-Set}$. Pour ce faire, a_i trie dans un ordre décroissant les valeurs de chaque colonne de la matrice de préférences (cf. équation 4.3). Après cela, a_i calcule sa préférence pour chaque agent $a_j \in W\text{-Set}$ comme suit :

$$X_{a_i, a_j} = \sum_{k \in [1,3]} (Y_{a_i, a_j}^k - Y_{a_i, a_j}^{k+1}) * \mu(p) \quad (4.7)$$

$\mu(p)$ est le poids d'un ensemble de critères. Ainsi,

$$\Pi_{a_i} = (X_{a_i, a_j}, X_{a_i, a_k}, \dots) : \{a_i, a_j, a_k, \dots\} \subseteq W\text{-Set}.$$

$\mu(p)$ est calculé sachant que $p = 1$ représente la fiabilité, $p = 2$ représente la probabilité de stabilité et $p = 3$ représente le poids. Donc, $\mu(1) = \rho_{a_i}$, $\mu(2) = P_s^{a_i}$, $\mu(3) = W_{a_i}$.

Du fait des incertitudes sur les agents et sur les tâches stochastiques, nous considérons que la fiabilité est plus importante que la probabilité de stabilité et le poids des agents lors des prises de décision. Ainsi, $\mu(1, 2) = \mu(1, 3) = \frac{1 - \rho_{a_i}}{2}$, $\mu(2, 3) = 1 - \rho_{a_i}$ and $\mu(1, 2, 3) = 1$.

Exemple 4.1 Soit le modèle de préférences multidimensionnelles $\prod_{a_4}^p$ d'un agent a_4 défini comme suit pour les critères $p1, p2$ et $p3$ (fiabilité, stabilité et poids). Donc, $p = (p1, p2, p3)$.

$\prod_{a_4}^p$	Agent a_i	a_1	a_2	a_3
$p1$		0.19	0.10	0.14
$p2$		0.15	0.12	0.17
$p3$		0.11	0.17	0.17

(4.8)

Considérons les poids individuels et collectifs des trois critères ($p1, p2$ et $p3$) comme suit : $\mu(p1) = 0.45$, $\mu(p2) = 0.45$, $\mu(p3) = 0.3$, $\mu(p1, p2) = 0.5$, $\mu(p1, p3) = 0.9$, $\mu(p2, p3) = 0.9$, $\mu(p1, p2, p3) = 1$. De là, a_4 construit une séquence décroissante des valeurs de ses préférences pour chaque critère $p1, p2$ et $p3$. a_4 obtient les valeurs de séquence suivantes (0.19, 0.15, 0.11) pour a_1 , (0.17, 0.12, 0.10) pour a_2 et (0.17, 0.17, 0.14) pour a_3 . Ainsi, en utilisant l'intégrale de Choquet, $\Pi_{a_4} = (0.148, 0.133, 0.149)$, car :

$$X_{a_4, a_1} = [(0.19 - 0.15) * 0.45] + [(0.15 - 0.11) * 0.5] + [(0.11 - 0) * 1] = 0.148.$$

$$X_{a_4, a_2} = [(0.17 - 0.12) * 0.3] + [(0.12 - 0.10) * 0.9] + [(0.10 - 0) * 1] = 0.133.$$

$$X_{a_4, a_3} = [(0.17 - 0.17) * 0.45] + [(0.17 - 0.14) * 0.9] + [(0.14 - 0) * 1] = 0.149.$$

- Le poids du *niveau-discussion* devient :

$$\overline{\Pi} = \frac{\sum_{a_i \in A' \subset A} \Pi_{a_i}}{\text{Card}(A')} \quad (4.9)$$

où $A' = W\text{-Set}$ est l'ensemble des agents qui ont répondu avec un *weak-accept*. Les agents qui maximisent $\overline{\Pi}$ sont toujours préférés (cf. lignes 5 et 6 algorithme 8).

Donc, s'il y a un conflit dans un *niveau-discussion*, les agents vont préférer ceux qui maximisent le poids $\overline{\Pi}$ de leur *niveau-discussion* pour permettre aux agents d'un même *niveau-discussion* d'avoir une probabilité plus grande de partager la même coalition. En effet, plus le *niveau-discussion* a un poids élevé, plus il est considéré comme étant formé d'agents qui se trouvent dans un certain état d'équilibre. En cas de conflits avec d'autres agents externes à leur *niveau-discussion*, ces agents sont préférés.

Étape 4 : Fusion des décisions des *niveau-discussions*.

Nous rappelons que nous avons déjà définis dans le chapitre précédent, la notion de fusion de deux listes de sondage.

Si $\text{rang}=0$ pour un agent qui est dans $W\text{-set}$, alors celui-ci considère qu'il doit participer à la gestion des négociations entre les *niveau-discussions* s'il existe un *niveau-discussion* dans son voisinage de saut suivant (cf. figure 4.4).

Cette manière de faire a pour objectif de fusionner les décisions provenant de plusieurs *niveau-*

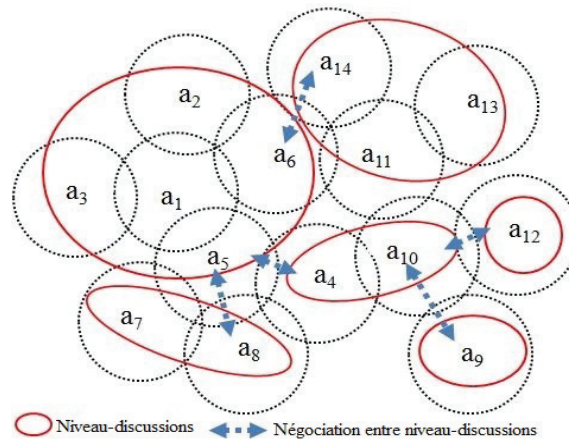


FIGURE 4.4 – Prise de décision entre les *niveau-discussions*.

discussions. Les agents de $\text{Hop}=2$ gèrent les décisions des *niveau-discussions* de $\text{Hop}=3$, les agents de $\text{Hop}=4$ gèrent les décisions des *niveau-discussions* de $\text{Hop}=5$, ainsi de suite.

Algorithme 7 : : P-NRB

Require: $O_c^{a_i} = \{\{B_i : i \in [1, |T_c|]\}, \{pl_{a_i} : i \in [1, |A|]\}\}$

RESULT : Coalition C

```

1: if  $O_c^{a_i} \forall a_j \in A : pl_{a_i} = \{a_i\}$  then
2:    $rang = Hop - \lfloor \frac{Hop}{2} \rfloor * 2$  et  $Hop = Hop + 1$ 
3:    $send(O_c^{a_i}, Hop) : TTL = 1$ 
4: end if
5: if  $a_j : rang == 1$  et  $\vartheta_{a_j}^t \neq \emptyset$  then
6:    $probe(O_c^{a_i}) : TTL = 1$  //pour faire le sondage du voisinage de saut suivant.
7: end if
8:  $\forall a_k : rang = 1$  et  $a_j \in \vartheta_{a_j}^t \cup a_j$ 
9: if  $\exists G_{a_j} \in O_c^{a_i}$  et  $U_{a_j}^{t-1} \leq U_{a_j}^t$  et  $Aut_{a_j}^t \neq 0$  then
10:   $W_{a_j} = \frac{Card(Hs_{a_j} \cap \vartheta_{a_j}^t)}{Card(\vartheta_{a_j}^t)}$  ou  $W_{\gamma_{a_j}}^{Net}$  si  $\exists L_{a_i}^{Net} : \gamma_{a_i}^{Net} \neq \emptyset$ 
11:   $W-Set = a_j$ 
12:  while  $\exists (G_{a_k}, a_k) \forall a_k \in \vartheta_{a_j}^t$  do
13:     $W-Set = W-Set \cup a_k$ 
14:    if  $\exists a_j, a_k : G_{a_j} \equiv G_{a_k}$  et  $(a_j, a_k) \subseteq W-Set$  then
15:       $Max(W_{a_j}, W_{a_k}), Max(\rho_{a_j}, \rho_{a_k})$  et  $Max(P_s^{a_j}, P_s^{a_k})$  //Nous rappelons que l'opérateur  $\equiv$  permet de
      vérifier l'équivalence entre deux paramètres.
16:    end if
17:  end while
18: end if
19: if  $a_j : rang == 1, \vartheta_{a_j}^t = \emptyset, \exists G_{a_j} \in O_c^{a_i}, U_{a_j}^{t-1} \leq U_{a_j}^t$  et  $Aut_{a_j}^t \neq 0$  then
20:   $W-Set = a_j$ 
21: end if
22: if  $W-Set \neq \emptyset$  then
23:   $\forall a_j \in W-Set$  calcule  $\prod_{a_j}^p$ 
24:  if  $\exists \prod_{a_i}$  then
25:    Calculer  $\overline{\prod}$  et envoyer un message  $commit(W-Set) : TTL = 1$ 
26:  end if
27:  if  $a_j : rang == 0$  et  $\vartheta_{a_j}^t = \emptyset$  et  $a_j \in W-Set$  then
28:    Diffuser  $commit(W-Set) : TTL = 1$ 
29:  end if
30:  if  $a_j : rang == 0$  et  $W_{a_j}$  est maximal et réception du  $commit(W-Set)$  then
31:     $MCL()$  //pour faire appel à l'algorithme 8
32:  end if
33: end if

```

Ce choix est fondé sur les principes suivants : (1) un message ne peut pas quitter le niveau $Hop=1$ à destination de $Hop=3$ sans passer par le niveau $Hop=2$, (2) l'objectif de décentralisation du mécanisme P-NRB et de partage des coûts en matière de ressources utilisées par chaque agent dans les processus de négociations, (3) la possibilité de permettre à chaque agent de participer à toutes les prises des décisions s'il a un but à atteindre avec la formation d'une coalition pour l'offre.

Remarquons que, c'est par souci de clarté que ces étapes illustrées par la figure 4.1, la figure 4.2, la figure 4.3 et la figure 4.4 sont présentées séparément mais elles se font de manière parallèle et asynchrone.

Étape 5 : Négociations et prises de décisions entre les *niveau-discussions*.

Dans cette étape, les négociations se font entre les *niveau-discussions* qui ont émis un *weak-accept* (cf. algorithme 8). Ces négociations sont gérées par les agents de *rang=0* ayant le plus grand poids dans leur *niveau-discussions*. Cependant, si un agent est seul dans son *niveau-discussion* il gère lui-même ses négociations. En cas de poids identiques, l'agent qui a la fiabilité la plus grande est choisi pour la gestion de la négociation avec les autres *niveau-discussions*.

- Pour la fusion des décisions provenant de chaque *niveau-discussion*, en cas de conflit entre deux *niveau-discussions* (des agents ayant le même but, ayant des préférences incompatibles, ...), les agents utilisent les poids des agents en conflit et les poids des *niveau-discussions* concernés pour résoudre les conflits. Ainsi, les agents de poids supérieur et appartenant à des *niveau-discussions* de poids plus élevés sont privilégiés, car ils ont une plus grande probabilité de fournir une coalition stable si tous les agents sont dans une même coalition.

- Après fusion, un nouveau *weak-accept* pour une coalition est généré s'il existe encore des buts de l'offre à couvrir. Cela signifie que la coalition n'est validée que si elle est *core stable* et si les agents forment un réseau connexe (chaque agent de la coalition a au moins un agent voisin qui appartient aussi à la coalition). Cette décision de valider une coalition se fonde aussi sur une prédiction de stabilité utilisant les MDP.

Étape 6 : Validation ou rejet d'une coalition.

Cette étape ne requiert pas de synchronisation supplémentaire entre les *niveau-discussions* concernés, car les agents vont utiliser une chaîne de Markov ergodique pour déterminer si la coalition sera *core stable* avec les agents impliqués et pour évaluer l'état de la tâche stochastique T_c que la

Algorithme 8 : : MCL() (Merging Chatting-Levels)**Require:** $W\text{-Set}=X1$ and $W\text{-Set}=X2 : X1, X2 \subset A$ **RESULT :** $commit(X1 \cup X2) : X1 \cap X2 = \emptyset$

```

1: if  $a_j \in X1 : rang=0$  et  $(X1 \setminus a_j) = \{a_k, a_l, \dots\} : W_{a_j} = Max(W_{a_k}, W_{a_l}, \dots)$  then
2:   if  $\nexists a_j \in X1$  et  $a_x \in X2 : G_{a_j} \equiv G_{a_x} \in O_c^{a_i}$  then
3:      $commit(X1 \cup X2) : X1 \cap X2 = \emptyset$ 
4:   end if
5:   if  $\exists a_j \in X1$  et  $a_x \in X2 : G_{a_j} \equiv G_{a_x} \in O_c^{a_i}$  then
6:     if  $W_{a_j} > W_{a_x}$  et  $\overline{II}$  de  $a_j$  est maximal then
7:       Sélectionner  $a_j$  et envoyer un  $commit(X1 \cup X2) : X1 \cap X2 = \emptyset$ 
8:     else
9:       if  $W_{a_x} > W_{a_j}$  et  $\overline{II}$  de  $a_x$  est maximal then
10:        Sélectionner  $a_x$  et envoyer un  $commit(X1 \cup X2) : X1 \cap X2 = \emptyset$ 
11:      else
12:        Calcul du vecteur invariant du MDP de leurs tâches //pour prédire la stabilité de leur
        coalition si  $X1 \cup X2 : X1 \cap X2 = \emptyset$  est validé en coalition avec la participation de  $a_j$  ou de  $a_x$ .
13:      end if
14:    end if
15:  end if
16: end if

```

future coalition doit prendre en charge. Le théorème 8 montre que si le MDP est ergodique, les agents ont, après calcul du vecteur invariant du MDP, la même connaissance sur la stabilité de la coalition à valider.

La manière dont nous utilisons le MDP et la méthode itérative d'obtention de son vecteur invariant sont largement discutées dans le chapitre 2. D'autres informations utiles pour la compréhension des propriétés des MDP sont expliquées en annexe.

Dans la section suivante, mettons en exergue les propriétés les plus importantes de notre mécanisme de formation de coalitions qui prouvent que nous avons atteint les objectifs affichés au début de ce travail dans ce contexte dynamique et incertain.

4.3.2 Analyse de notre mécanisme P-NRB

Cette analyse de notre mécanisme de formation de coalitions, montre qu'il est *anytime* et il permet de trouver les coalitions *A-core*. Le lemme 7 ci-dessous, prouve que notre mécanisme de formation de coalitions est décentralisé, asynchrone et que le contrôle des prises de décisions multi-agents est distribué.

Lemme 7 *P-NRB est un mécanisme distribué et asynchrone avec des prises de décisions multi-agents décentralisées.*

Démonstration. Considérons un ensemble d'agents A . $\forall a_i \in A : O_c^{a_i} \neq \emptyset$ peut utiliser le mécanisme P-NRB si $\vartheta_{a_i}^t \neq \emptyset$ ($Probe(O_c^{a_i}) \forall a_j \in \vartheta_{a_i}^t : pl_{a_i} = \{a_i\}$) sans attendre un quelconque tour de rôle et sans l'intervention d'un ordonnanceur. $\forall a_i \in A$ doit fournir sa décision (rejet, *weak-accept* ou validation) durant un processus de formation d'une coalition en gérant un *niveau-discussion* si son *rang*=1 (cf. lignes 6 à 22 de l'algorithme 7), ou bien, en gérant la négociation entre des *niveau-discussions* si son *rang*=0 (cf. lignes 28 à 33 de l'algorithme 7). La manière dont un agent devient responsable du maintien de la cohérence des décisions de son *niveau-discussion* ou devient gestionnaire de la négociation et de la fusion des décisions provenant de plusieurs *niveau-discussions* se fonde seulement sur un calcul local de chaque agent sans l'intervention d'un ordonnanceur. C'est-à-dire de manière distribuée et asynchrone (cf. lignes 1 à 5 de l'algorithme 7). Ce qui prouve notre lemme. $\square$

Le lemme ci-dessous prouve qu'il n'y a pas d'états bloquants durant un processus de formation de coalitions et que le mécanisme P-NRB finit toujours par se terminer sur un état final qui peut produire des coalitions stables ou non.

Lemme 8 *Le mécanisme P-NRB se termine toujours sans blocage même si des coalitions core stables ne sont pas trouvées.*

Démonstration. $\forall \text{Probe}(O_c^{a_i})$, un agent doit propager ce message de sondage dans son voisinage si $\theta_{a_i}^t \neq \emptyset$ en respectant le principe du non-return broadcast et le *TTL* du message afin d'éviter les messages envoyés en boucles. Chaque conflit entre deux agents a_i et a_j est géré par les agents de leur *niveau-discussion* en sélectionnant l'agent qui maximise le poids de son *niveau-discussion*. De plus, le choix porte sur l'agent dont : le poids est le plus grand ($\text{Max}(W_{a_i}, W_{a_j})$), la fiabilité la plus grande ($\text{Max}(\rho_{a_i}, \rho_{a_j})$) et la probabilité de stabilité la plus grande ($\text{Max}(P_s^{a_i}, P_s^{a_j})$). Cette manière de gérer les conflits permet d'éviter les cas où deux agents a_i et a_j sont dans une impasse pour un accord sans possibilités de prises de décisions décentralisées des agents qui attendent une validation de la coalition. À noter que, la résolution des conflits s'effectue de manière décentralisée et localisée dans le ou les *niveau-discussions* concernés par les conflits des deux agents (cf. lignes 13 à 18, algorithme 7). Ainsi, tous les blocages (deadlocks) sont évités, dans les communications et les négociations durant les processus de formation de coalitions avec le mécanisme P-NRB. Ce qui donne la preuve de notre lemme. $\square$

Théorème 4 *Une réponse weak-accept signifie qu'il existe un ensemble d'agents noté W-Set appartenant à un niveau-discussion qui sont d'accord pour la formation d'une coalition C en garantissant au moins une partition Nash-stable dans C.*

Démonstration. Soit W-Set représentant un ensemble d'agents d'un *niveau-discussion* qui ont répondu à une offre $O_c^{a_i}$ par un *weak-accept*. Notons par $U_{a_x}^t$ l'utilité de l'agent a_x à l'instant t .

- (1) $\forall a_j, W\text{-Set} = W\text{-Set} \cup a_j$ si et seulement si $G_{a_j} \in O_c^{a_i}$ et $U_{a_j}^{t-1} \leq U_{a_j}^t$ et $\text{Aut}_{a_j}^t \neq 0$. Cela signifie que, $\forall a_j \in W\text{-Set}$, il veut participer à la coalition C de l'offre $O_c^{a_i}$.
- (2) $\forall a_i, a_j \in W\text{-Set}, G_{a_i} \neq G_{a_j}$ et $U_{a_i}^{t-1} \leq U_{a_i}^t, U_{a_j}^{t-1} \leq U_{a_j}^t$. Cela signifie qu'il n'existe aucun conflit entre les agents appartenant à l'ensemble W-Set et que l'utilité de chacun de ces agents est maximale.
- (3) $\forall W_{a_i}$ de $a_i \in W\text{-Set}$, W_{a_i} dépend de ses alliances avec les agents de sa vue $\theta_{a_i}^t$. De plus, à chaque coopération avec d'autres agents, a_i cherche à maximiser son poids et sa fiabilité, car s'il se désengage de l'ensemble W-Set, son poids et sa fiabilité vont être réduits.

Les points (1), (2) et (3) décrits ci-dessus signifient que, $\forall a_i \in W\text{-Set}$ il n'est pas motivé à quitter l'ensemble W-Set pour améliorer son utilité en s'engageant dans une autre négociation pour une autre offre et il accepte la formation de la coalition C avec les agents de W-Set. Ainsi, $W\text{-Set} \subseteq C$ est une partie Nash-stable de C. Ce qui prouve notre théorème. $\square$

Théorème 5 *La fusion de deux ensembles d'agents qui ont répondu par un weak-accept à une offre $O_c^{a_i}$ fournit toujours un ensemble d'agents qui sont en accord pour la formation de la coalition C dont au moins une partition de C est Nash-stable.*

Démonstration. Considérons deux ensembles d'agents $X1$ et $X2$ comme représentant deux ensembles d'agents de deux *niveau-discussions* différents qui ont répondu par un *weak-accept* à une offre de formation d'une coalition C . Le théorème 4, prouve que les deux ensembles $X1$ et $X2$ sont dans un équilibre de Nash. La ligne 3 de l'algorithme 8 montre que $X1 \cup X2$ est telle que $X1 \cap X2 = \emptyset$. De la ligne 5 à la ligne 7 de l'algorithme 8 nous remarquons que l'expression $X1 \cup X2$ est telle que $\forall a_i \in X1 \cup X2, W_{a_i}$ et $\prod$ (cf. équation 4.9) sont maximisés. Ainsi, si l'on effectue la fusion des deux ensembles $X1$ et $X2$, chaque agent reste d'accord pour la formation de la coalition C avec chaque agent de $X1 \cup X2$. Ce qui prouve notre théorème. $\square$

Théorème 6 *P-NRB converge toujours vers la formation de coalitions core stables si la formation de coalitions core stables est possible.*

Démonstration. Le lemme 7 et le lemme 8 prouvent que, si une réponse *weak-accept* d'un *niveau-discussion* existe, alors le mécanisme P-NRB va toujours en tenir compte et sans blocage. De plus, ces lemmes montrent que - le mécanisme P-NRB fonctionne correctement même si des agents sont indisponibles, - la terminaison du mécanisme P-NRB est toujours garantie, - en fonction de ses propriétés, chaque agent a un contrôle sur le résultat qui est retourné par le mécanisme P-NRB.

Le théorème 4 prouve que, le résultat issu de la convergence du mécanisme P-NRB est toujours une coalition stable. Le théorème 5 montre que, si une réponse *weak-accept* est émise par un *niveau-discussion* ou après fusion d'un ensemble de *niveau-discussions*, l'utilité, la fiabilité, la probabilité de stabilité et le bien-être social de l'ensemble des agents concernés par cette réponse *weak-accept* sont maximisés (cf. ligne 16 de l'algorithme 7 et ligne 1 à la ligne 7 de l'algorithme 8), car le *niveau-discussion* de poids plus grand est toujours préféré. Ainsi, le résultat retourné par P-NRB mène à la formation de coalitions où aucun agent n'est motivé à quitter sa coalition et où le bien-être social des agents est maximisé. Ce qui prouve que la convergence de P-NRB mène à la formation de coalitions *core stables* et c'est ce qui prouve notre théorème. $\square$

Lemme 9 *Une coalition C issue de la convergence du mécanisme P-NRB est core stable et $\forall a_i \in C \exists a_j \in C : a_j \in \theta_{a_i}^t$.*

Démonstration. Le théorème 4 montre que, si une réponse *weak-accept* pour une offre de formation d'une coalition C est émise, alors il existe un ensemble *W-Set* d'agents qui forment une partition Nash-stable de la coalition C . De même, si nous considérons $X1$ et $X2$ comme deux ensembles d'agents *W-Set* provenant de deux *niveau-discussions* différents, le théorème 5, montre que la fusion de ces deux ensembles d'agents qui sont dans un équilibre de Nash, donne encore un ensemble qui est dans un équilibre de Nash. De plus, chaque agent de l'ensemble $X1 \cup X2 : X1 \cap X2 = \emptyset$ a au moins un agent voisin appartenant à cet ensemble $X1 \cup X2 : X1 \cap X2 = \emptyset$ avec lequel il veut partager la coalition C à former. Ainsi, si C est validée, alors chaque agent dans C va avoir au

moins un agent voisin qui est aussi dans C . D'où la connectivité des membres de la coalition. Ce qui complète notre preuve. $\square$

Vu que nous considérons que les agents sont déployés dans des composants communicants comme, par exemple, des capteurs, des Smart-phones, etc. alors un critère important à vérifier, est l'optimisation des ressources et de la bande passante de ces composants lors du processus de formation de coalitions. D'une manière générale, pour ce genre de composants considérés, le coût des messages est plus important lors de l'envoi de messages. Ainsi, nous cherchons à prouver par le lemme ci-dessous que notre mécanisme P-NRB tient compte de ces contraintes de ressources pour borner les messages envoyés par chaque composant jusqu'à la convergence du mécanisme.

Lemme 10 *Le coût des messages est borné et polynomial pour chaque agent jusqu'à la convergence du mécanisme P-NRB.*

Démonstration. Soit $\bar{Msg}$ le coût des messages de P-NRB. Jusqu'à la formation des *niveau-discussions*, $\forall a_j \in A, \forall rang : G_{a_j} \notin O_c^{a_i}$ ou $U_{a_j}^{t-1} > U_{a_j}^t$ ou $Aut_{a_j}^t = 0$, si $\vartheta_{a_j}^t = \emptyset$, alors $\bar{Msg} = 0$, si $\vartheta_{a_j}^t \neq \emptyset$ et $Aut_{a_j}^t \neq 0$, alors $\bar{Msg} = 1$, car il doit diffuser l'offre $O_c^{a_i}$ de formation de la coalition C . $\forall a_j \in A : rang = 1, G_{a_j} \in O_c^{a_i}, U_{a_j}^{t-1} \leq U_{a_j}^t$, et $Aut_{a_j}^t \neq 0$, alors $1 \leq \bar{Msg} \leq 2$ jusqu'à la diffusion de la réponse *weak-accept* de son *niveau-discussion*, car a_j doit envoyer au moins un message de broadcast pour effectuer le sondage de son *niveau-discussion* (cf. ligne 7 de l'algorithme 7) et au plus, un message supplémentaire pour informer les membres de son *niveau-discussion* à propos de la décision finale (cf. ligne 9 à la ligne 14 de l'algorithme 7). Néanmoins, lorsque N agents ont des buts ou des préférences en conflits ($\exists a_x$ et $a_y : G_{a_x} \equiv G_{a_y}$), alors $3 \leq \bar{Msg} \leq [\frac{N*(N-1)}{2} + 2]$, car pour N agents, nous avons à résoudre au pire des cas, $\frac{N*(N-1)}{2}$ conflits. De plus, pour gérer les conflits, un agent a_j doit envoyer un message aux agents en conflits pour prendre connaissance de leurs différends puis un autre message afin d'en informer les membres du *niveau-discussion*. Ce dernier message permet d'obtenir les poids et les probabilités de stabilité des agents en conflits sachant qu'il peut calculer en local les fiabilités de ces agents (cf. lignes 15 à 17 de l'algorithme 7).

$\forall a_j \in A : rang = 0, a_j \in W\text{-Set}$ et $\vartheta_{a_j}^t \neq \emptyset$, alors a_j est responsable de la diffusion de la réponse *weak-accept* de son *niveau-discussion* dans sa *vue* $\vartheta_{a_j}^t$ (cf. ligne 28 à la ligne 30 de l'algorithme 7). Jusqu'à la réponse avec un *weak-accept*, a_j doit envoyer 1 message pour diffuser l'offre de formation de la coalition et 1 message pour spécifier son but $G_{a_j} \in O_c^{a_i}$ à l'agent voisin qui gère le sondage de son *niveau-discussion*. Un message supplémentaire qui contient son poids W_{a_j} et sa probabilité de stabilité $P_s^{a_j}$ peut être requis, si son but est en conflit avec un but d'un autre agent de son *niveau-discussion*. Ainsi, $2 \leq \bar{Msg} \leq 3$. Cependant, si $a_j : rang = 0$ et qu'il a le poids W_{a_j} le plus élevé dans son *niveau-discussion*, alors a_j est responsable de la négociation et de la fusion entre son *niveau-discussion* et les autres *niveau-discussions* de son voisinage qui ont aussi émis une réponse *weak-accept*, c'est-à-dire,

qui ont un ensemble $W\text{-Set} \neq \emptyset$ (cf. ligne 31 et ligne 32 de l'algorithme 7). Pour effectuer la fusion de deux ensembles $W\text{-Set}$ de deux *niveau-discussions*, a_j doit envoyer au moins 2 messages ($\bar{M}sg=2$), si aucun conflit n'est détecté, c'est-à-dire, 1 message pour diffuser la réponse *weak-accept* de son *niveau-discussion* et 1 message après la fusion des deux ensembles d'agents $W\text{-Set}$ concernés par les deux réponses *weak-accept*. Néanmoins, si N agents ont des buts en conflits, alors $\bar{M}sg=1+N+1$. Si a_j doit gérer la fusion de tous les *niveau-discussions* de sa *vue*, alors $\bar{M}sg=(2+N) * card(\vartheta_{a_j}^t)$. Ainsi, le coût des messages $\forall a_j \in A : rang = 0, a_j \in W\text{-Set}$ est de $2 \leq \bar{M}sg \leq (2+N) * card(\vartheta_{a_j}^t)$. Comme le nombre d'agents n'est pas infini, alors $\forall a_j \in A, Card(\vartheta_{a_j}^t)$ et N sont bornés et par conséquent le coût des messages $\bar{M}sg$ est aussi borné et est polynomial. Ce qui prouve notre lemme. $\square$

Lemme 11 *P-NRB permet aux agents de prendre des décisions collectives de manière parallèle et décentralisée sans inconsistance et sans privilégier un agent durant les processus de formation des coalitions.*

Démonstration. $\forall a_j \in A$, il peut participer dans un *niveau-discussion* et prendre une décision sur une offre sans attendre un ordonnanceur ou un tour de rôle. $\forall a_j \in A$, sa prise de décision dépend de ses paramètres $G_{a_j}, U_{a_j}, Aut_{a_j}^t, L_{a_j}^{Net}, W_{a_j}, \rho_{a_j}, P_s^{a_j}$ mais pas sur sa proximité avec l'agent qui a initié l'offre de formation de la coalition. De plus, la gestion des décisions d'un *niveau-discussion* est distribuée (cf. lemme 7). Aucune inconsistance dans les prises des décisions ne peut arriver durant un processus de formation d'une coalition, car les conflits sont résolus au fur et à mesure qu'ils sont détectés. Un ensemble $W\text{-Set}$ d'agents responsables de la réponse *weak-accept* d'un *niveau-discussion* est Nash-stable et le résultat de la fusion de deux ensembles $W\text{-set}$ reste Nash-stable (théorème 4 et théorème 5). Ce qui prouve notre lemme. $\square$

Théorème 7 *Si la convergence de P-NRB forme une coalition core stable, celle-ci est nécessairement A-core.*

Démonstration. Le théorème 6 prouve une convergence vers des coalitions *core stables*. Le lemme 9 signifie que, dans une coalition, chaque changement dynamique sur l'état d'une ou de plusieurs tâches et sur la disponibilité d'un ou de plusieurs agents, est détecté automatiquement par au moins un agent de la coalition. Le lemme 8 et le 10 montrent que, après une instabilité dans une coalition, cette dernière redevient stable après un nombre fini d'étapes sans qu'il y ait blocage du processus de stabilisation. De plus, la décision d'ajouter un ensemble d'agents dans une coalition respecte les préférences de chaque agent de la coalition (cf. lemme 11) et les conflits sont résolus de manière décentralisée au fur et à mesure de leurs apparitions (cf. Lemme 7). Sachant que, l'ajout d'un ensemble d'agents dans une coalition peut être considéré comme une fusion de deux ensembles $W\text{-Set}$ de deux *niveau-discussion*, alors le théorème 5 montre que P-NRB permet une stabilisation des coalitions formées. Par conséquent, nous pouvons déduire que P-NRB, permet la formation de coalitions A-core. Ce qui donne la preuve de notre théorème. $\square$

Théorème 8 *Comme la chaîne de Markov liée à l'état des agents et à la tâche est ergodique, alors sans nécessité de synchronisation, tous les agents impliqués dans le même processus de formation de coalitions auront la même connaissance à propos de la stabilité de la coalition qu'ils peuvent espérer en ajoutant l'un des agents en conflit.*

Démonstration. Une chaîne de Markov possède toujours au moins un vecteur invariant Π . Comme nous l'avons déjà montré dans le chapitre 3, si en plus la chaîne de Markov est ergodique alors la chaîne possède un vecteur invariant Π unique quelle que soit la distribution de probabilité initiale des états de la chaîne de Markov donnée par sa matrice P et quelles que soient les préférences et les dépendances des agents. Ce qui prouve notre théorème. $\square$

Théorème 9 *Le mécanisme P-NRB pour la recherche de coalitions A-core dans un contexte dynamique, possède la propriété anytime.*

Démonstration. Soit l'offre $O_c^{a_i}$ de formation d'une coalition C d'un agent a_i . Si l'ensemble des coalitions core n'est pas vide, cela signifie qu'il a au moins un *niveau-discussion* qui va émettre une réponse *weak-accept* pour l'offre $O_c^{a_i}$. Un tel ensemble d'agents sera Nash stable (voir, theorem 4). De plus, ces agents ayant produit le *weak-accept* se sont assurés que leur utilité est maximale, qu'il n'y a pas de conflits entre eux et que chacun a le meilleur poids du *niveau-discussion* pour le but le concernant. Ces points assurent l'existence d'un bien être social. Ainsi, même si la négociation est interrompue avant la fusion de toutes les décisions des agents des différents *niveau-discussion* de l'environnement de déploiement, une partition stable de la coalition sera déjà disponible. Ce qui prouve notre théorème. $\square$

Nous donnons de manière succincte dans la section suivante, les règles de fonctionnement supplémentaires que les agents observent durant leur phase de découverte et de coordination dans leur environnement.

4.3.3 Règles générales

Cette section détaille d'autres contraintes et règles de notre mécanisme de formation de coalitions qui assure une cohérence de notre mécanisme P-NRB en matière de processus de négociation, de communication, de calcul, etc. Cependant, les règles et les contraintes spécifiques sont fournies dans les étapes concernées. C'est pourquoi, dans ce qui suit, nous ne donnons que les règles communes à toutes les étapes.

Règle 1 : Offre valide. Un agent a_i ne peut proposer une offre $O_c^{a_i}$ pour la formation d'une coalition C que si $\vartheta_{a_i}^t \neq \emptyset$, $\exists G_{a_i} \equiv B_x \in O_c^{a_i}$ et $\forall B_x \in O_c^{a_i}$, le but B_x est requis pour la formation et/ou la stabilisation de la coalition C . Un agent a_j ne peut accepter $B_x \equiv G_{a_j} \in O_c^{a_i}$ que si est seulement si $Rx \in B_x : Rx \in R_{a_j}$. De plus, a_j ne peut participer à la coalition C que si $\forall a_k \in C, u_c^{a_k} \geq u_c^{a_j} : C' = C \cup a_j$. ■

Règle 2 : Validation d'un accord sur une offre. Une fois qu'un agent $a_j \in A \setminus C$ accepte l'offre d'un agent a_i tel qu'un but $B_x \equiv G_{a_j} \in O_c^{a_i} : a_i \in C$, alors les deux agents mettent à jour, leurs connaissances à propos des contraintes de la coalition C (état(s) et localisation(s) des tâches à exécuter) et de la stabilité de la coalition C . Si un agent a_j effectue une contre-proposition sur un but $B_x \equiv G_{a_j} \in O_c^{a_i} : a_i \in C$, alors a_j est libéré de ses précédents accords, mais si cette contre-proposition remet en cause une réponse *weak-accept* ou une coalition déjà validée, alors sa fiabilité sera réduite par les agents informés de cette contre-proposition. À noter que, cette contre-proposition entraîne de nouvelles négociations si la fiabilité de a_j lui permet de renégocier avec les agents. Un agent a_j est libéré d'un accord sans réduction de sa fiabilité si cet accord réduit l'utilité d'un agent a_k qui a déjà un accord pour une participation dans la coalition C avec une fiabilité et un poids plus élevés que ceux de a_j . ■

Règle 3 : Communication. Chaque agent peut à tout moment, formuler une offre, accepter une offre ou effectuer une contre-proposition, car dans ce travail, il n'y a pas d'ordonnanceur. Cependant, pour assurer une convergence des négociations, pour gérer leur bande passante, leur 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 qu'une information de mise à jour (disparition d'agents de C , évolution sur une tâche, nouvelle offre, ...) qui arrive d'un agent a_k , ne peut pas lui être retournée par ses agents voisins $\vartheta_{a_j}^t$.

Le *TTL* permet de définir le nombre de saut autorisé pour un message. Nous le définissons comme dans le précédent chapitre, à savoir : $1 \leq TTL \leq \lfloor \frac{Sz}{2*Y} \rfloor$

où Sz est la taille de l'environnement couverte par la tâche, Y la portée du signal du composant qui héberge l'agent qui a initié le message.

Cependant, même si un agent reçoit un message, il ne répond ou ne le diffuse en mode *non-return broadcast* que suivant ses propres stratégies. Stratégies qui visent à augmenter son efficacité et à ménager ses ressources. ■

Règle 4 : Terminaison. Une fois qu'un agent a_j accepte ou reçoit une acceptation sur une offre en négociation, alors a_j doit envoyer un message à tous les agents concernés par cette négociation afin de les informer de sa décision finale et pour mettre fin à la négociation.

Lorsqu'une réponse *weak-accept* est émise par un *niveau-discussion*, les agents de ce *niveau-discussion* doivent mettre fin aux négociations internes. Durant les négociations pour la formation de la coalition C , dans un *niveau-discussion*, si un des agents de ce *niveau-discussion* trouve une plus grande utilité en participant aux négociations d'une nouvelle offre, il peut mettre fin à ses négociations sur C .

Tout agent a_i qui reçoit un refus de son offre par un agent a_j doit interrompre sa négociation avec a_j et chercher d'autres agents susceptibles d'accepter son offre. Si des agents rivalisent pour la même offre, ceux retenus pour la coalition reçoivent une confirmation par un message de validation de participation, alors que ceux rejetés reçoivent un message de rejet, ce qui met fin aux négociations. Ainsi, tout agent dont la participation est rejetée, doit interrompre la négociation afin d'avoir la possibilité de négocier une autre offre. De plus, toute négociation a une durée au-delà de laquelle tout agent qui n'a pas un accord de participation doit modifier ou utiliser ses dépendances pour améliorer l'offre ou bien la supprimer. ■

La section suivante présente les résultats des simulations analytiques et expérimentales de notre mécanisme de formation de coalitions P-NRB. Ces résultats cherchent à montrer comment notre mécanisme surmonte dynamiquement les incertitudes sur les tâches et sur les agents pour trouver les coalitions *A-core*.

4.4 ÉVALUATIONS DE NOTRE MÉCANISME P-NRB

La section des évaluations analytiques a pour objectif d'étudier la complexité de notre mécanisme en matière de coût de messages. Ces évaluations sont effectuées pour un ensemble de distributions de probabilités. La partie expériences cherche à prouver - la convergence des processus de négociations vers des coalitions *core stables*, - l'existence des propriétés d'auto-stabilisations des coalitions formées (coalitions *A-core*), et - la propriété *anytime* de notre mécanisme de formation de coalitions. Les expériences sont aussi réalisées, lorsque les conflits des agents et les tâches suivent un ensemble de distributions de probabilités.

4.4.1 Évaluations analytiques

Nous donnons par simulations analytiques le coût des messages $\bar{M}sg$ jusqu'à la convergence de notre mécanisme de formation de coalitions P-NRB. Nous effectuons cette étude lorsque les conflits des agents dus à leurs préférences et à leurs dépendances suivent une distribution de probabilité - Uniforme ($conflits \sim U(a, b)$ où $a=0$ et $b=10$), - Normale ($conflits \sim N(\mu, \sigma^2)$ où $\mu=0.5$ et $\sigma=0.1$)

ou - de Poisson ($conflits \sim P(\lambda)$ où $\lambda=0.5$). L'expression $\bar{M}sg$ est donnée par le lemme 10. Lorsque

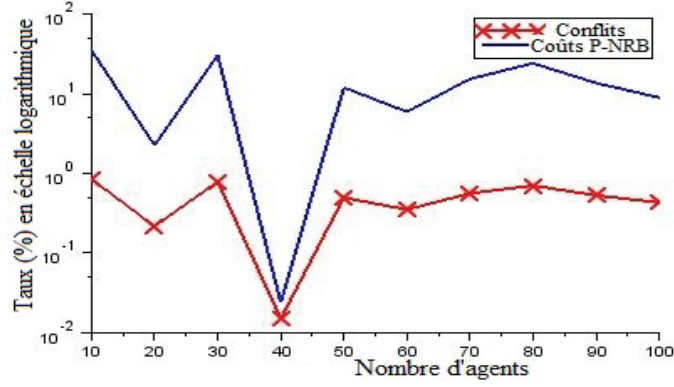


FIGURE 4.5 – Coûts des messages de P-NRB lorsque $conflits \sim U(a, b)$ où $a=0$ et $b=10$.

les conflits des agents suivent une distribution de probabilité Uniforme, la Figure 4.5 montre que, $Max(\bar{M}sg) \approx \bar{M}sg * 2$ si chaque agent est en conflit avec au moins un agent qui participe au processus de formation de coalitions ($conflits \sim 100\%$). Cependant, $Max(\bar{M}sg) \approx \bar{M}sg$ si ces conflits sont de l'ordre de 50%. La Figure 4.6 montre que, le taux de conflits des agents ne dépasse pas en

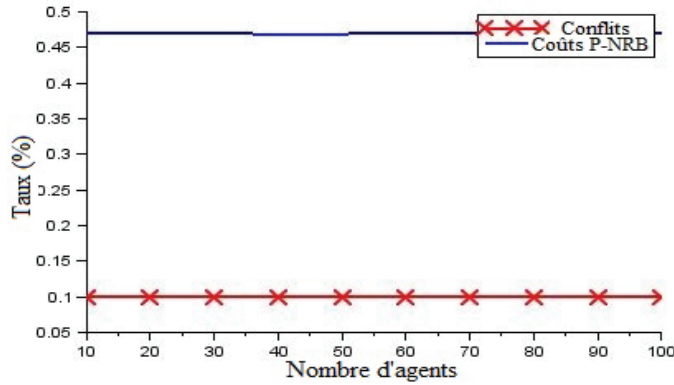


FIGURE 4.6 – Coûts des messages de P-NRB lorsque $conflict \sim N(\mu, \sigma^2)$ où $\mu=0.5$ et $\sigma=0.1$.

moyenne 10% et $\bar{M}sg * 0.45 < Max(\bar{M}sg) < \bar{M}sg * 0.5$ lorsque les conflits des agents suivent une distribution de probabilité Normale. La Figure 4.7 montre que, le taux de conflits des agents ne dépasse pas en moyenne 6% et $Max(\bar{M}sg) \approx \bar{M}sg * 0.12$ si les conflits des agents suivent une distribution de probabilité de Poisson.

Ces résultats prouvent que, notre mécanisme P-NRB converge avec un coût polynomial en matière de messages et que ce coût des messages ne dépend pas du nombre d'agents mais seulement du nombre de conflits potentiels avant la convergence de P-NRB. De plus, les taux des conflits sont bornés si la distribution de probabilité des conflits suit une loi uniforme, normale ou de Poisson.

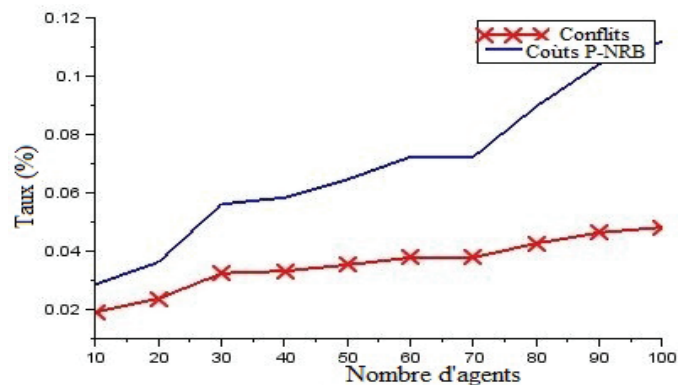


FIGURE 4.7 – Coûts des messages de P-NRB lorsque $\text{conflict} \sim P(\lambda)$ où $\lambda=0.5$.

4.4.2 Expériences

4.4.2.1 Paramétrages et méthodologies

Une tâche est assignée à un agent unique qui doit utiliser ses stratégies de négociations afin d'établir une coalition stable pour l'exécution de sa tâche. La stratégie d'un agent durant une phase de négociation peut concerner la manière d'utiliser ou d'établir ses *alliances*, la manière d'utiliser ses dépendances, et/ou la manière de raisonner sur les contraintes et les récompenses associées à une tâche.

Du fait des propriétés que nous voulons mettre en exergue dans nos simulations, nous effectuons ces simulations lorsque les indisponibilités des agents, les conflits dus aux préférences et aux dépendances des agents et les états des tâches suivent des distributions de probabilité Uniforme, Normale ou de Poisson. Dans notre simulateur développé sous Java, chaque agent est autonome et égoïste avec des dépendances natives (un agent a des obligations envers un ensemble d'agents dès son déploiement) ou non. Aléatoirement, un agent peut initier un processus de formation de coalitions, être libre de tout engagement envers les autres agents ou devenir indisponible. La dynamique de la localisation et des états des tâches est générée aléatoirement de même que les ressources nécessaires pour l'exécution d'une tâche à un instant donné t . Le nombre d'agents dans l'environnement est choisi entre 10 et 100 et le nombre de tâches simultanées dans l'environnement est compris entre 3 et 8 tâches où les propriétés de chaque tâche sont générées aléatoirement. Les simulations sont effectuées dans une machine Intel i7 (4 GHZ) avec 4GB de RAM.

Pour maximiser son utilité et pour atteindre son but, chaque agent a_i tient compte de ses caractéristiques (ressources, autonomie d'énergie, historique, voisinage, utilité, dépendances) et des contraintes liées aux tâches à exécuter. Chaque négociation est décentralisée et a une durée au-delà de laquelle elle est abandonnée s'il n'y a pas d'accord définitif. Cette durée des négociations n'est pas fixée à l'avance, car elle dépend de la dynamique des agents et des tâches et du fait que les

alliances proposées et les *récompenses* associées aux tâches ont une durée de validité limitée dans le temps. Les ressources disponibles dans l'environnement de déploiement des agents évoluent dynamiquement, car elles dépendent de la disponibilité des agents et des tâches en cours d'exécution. À cause de la dynamique des ressources disponibles ou requises pour une tâche, comme pour le chapitre précédent, nous effectuons nos simulations en considérant les *ratios*.

Pour rappel, un *ratio* est égal au nombre total de ressources disponibles dans l'environnement de déploiement sur le nombre total de ressources nécessaires pour former les coalitions requises pour exécuter les tâches.

Pour simuler l'indisponibilité des agents, nous désactivons aléatoirement un ensemble d'agents. De plus, cette action a pour objectif de simuler les changements de topologies réseaux des agents et de pouvoir tester les propriétés auto-stabilisations des coalitions formées. Les messages sont librement échangés entre les agents sans notions de tour de rôle ou l'intervention d'organisateur. Cependant, ces échanges respectent le principe du *non-return broadcast* et du *TTL*. Pour obtenir des valeurs moyennes, nous répétons 10 fois chaque contexte de simulation.

4.4.2.2 Résultats

Suivant les paramétrages et les méthodologies énoncés précédemment, nous étudions les résultats expérimentaux de notre mécanisme de formation de coalitions P-NRB afin de montrer sa capacité à former des coalitions *core stables* capables de se stabiliser dynamiquement (coalitions *A-core*). La

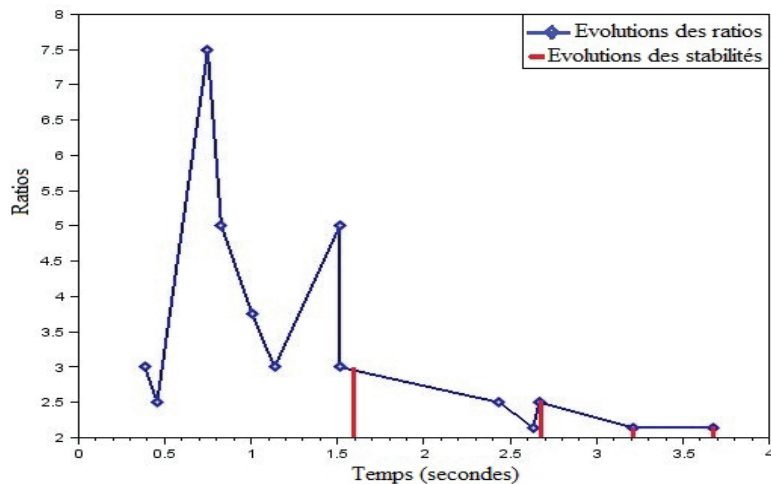


FIGURE 4.8 – Stabilité des coalitions dans le temps lorsque les Ratios suivent une distribution de probabilité Uniforme $U(a, b)$: $a=0$ et $b=10$.

Figure 4.8 montre que si une coalition *core stable* est formée, elle reste stable même en cas de décroissance ou d'accroissement des *Ratios* suivant une distribution de probabilité Uniforme (après

2.7s). Cependant, nous pouvons noter une instabilité temporaire entre 1.7s et 2.7s avant sa nouvelle configuration stable. La Figure 4.9 montre aussi que, si une coalition est *core stable*, elle reste stable

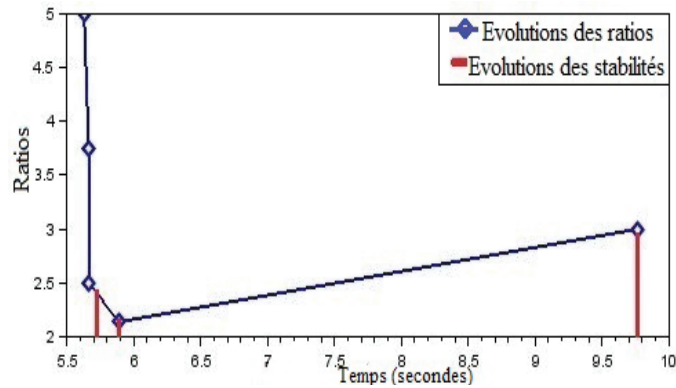


FIGURE 4.9 – Stabilité des coalitions dans le temps lorsque les Ratios suivent une distribution de probabilité Normale $N(\mu, \sigma^2)$: $\mu=0.5$ et $\sigma=0.1$.

même en cas de décroissance ou d'accroissement des *Ratios* suivant une distribution de probabilité Normale (après 5.7s). De manière générale, la durée d'instabilité d'une coalition dans ce cas est négligeable, c'est pourquoi elle n'est pas visible sur la figure. La Figure 4.10 montre aussi que, si

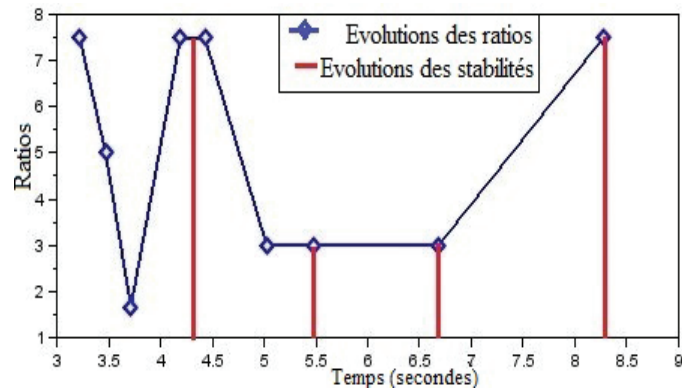


FIGURE 4.10 – Stabilité des coalitions dans le temps lorsque les Ratios suivent une distribution de probabilité de Poisson $P(\lambda)$: $\lambda=0.5$.

une coalition est *core stable*, elle reste stable même en cas de décroissance ou d'accroissement des *Ratios* suivant une distribution de probabilité de Poisson (après 4.4s). Une durée d'instabilité est moins négligeable que pour une distribution de probabilité Normale (entre 5s et 5.5s).

Ces résultats issus de ces simulations prouvent que, si une coalition est *core stable*, son instabilité due à une augmentation ou une diminution des exigences des tâches ou des agents n'implique pas sa dissolution. De plus, après une instabilité, une coalition *core stable* converge toujours vers une nouvelle configuration *core stable* sans renégocier les engagements des agents. Ces remarques prouvent la formation de coalitions *A-core*.

4.5 ÉTUDE COMPARATIVE ENTRE S-NRB ET P-NRB

4.5.1 Comparaisons analytiques entre S-NRB et P-NRB

Nous cherchons à mettre en évidence les paramètres multi-agents qui influencent les performances de chacun de ces mécanismes de formation de coalitions *core stables*.

Nous utilisons les mêmes paramètres que précédemment, c'est-à-dire, lorsque les conflits dus aux préférences et aux dépendances des agents ont une distribution :

Uniforme ($conflits \sim U(a, b)$ où $a=0$ et $b=10$), Normale ($conflits \sim N(\mu, \sigma^2)$ où $\mu=0.5$ et $\sigma=0.1$) ou de Poisson ($conflits \sim P(\lambda)$ où $\lambda=0.5$).

Dans un premier temps, nous étudions le coût de S-NRB et de P-NRB en matière de messages émis par chaque agent jusqu'à la formation des premières coalitions stables.

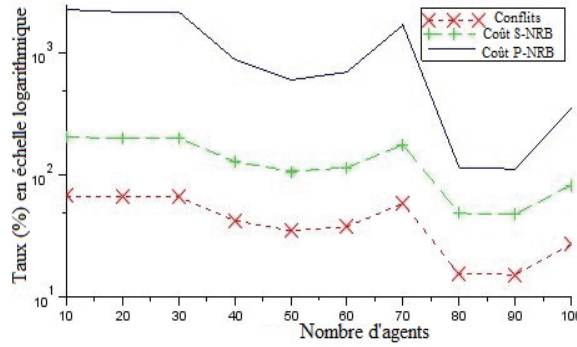


FIGURE 4.11 – Coûts en matière de messages jusqu'à la convergence de S-NRB et P-NRB, lorsque $conflits \sim U(a, b)$ où $a=0$ et $b=10$.

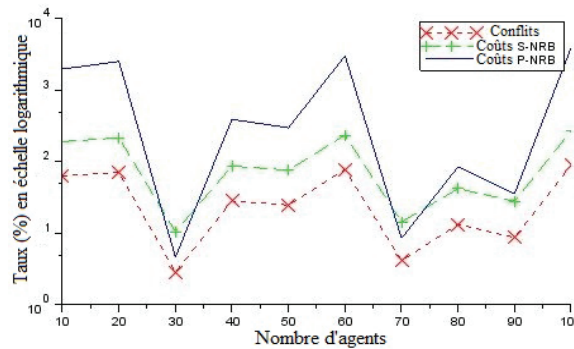


FIGURE 4.12 – Coûts en matière de messages jusqu'à la convergence de S-NRB et P-NRB, lorsque $conflits \sim N(\mu, \sigma^2)$ où $\mu=0.5$ et $\sigma=0.1$.

Les résultats des figures 4.11, 4.12 et 4.13 montrent que, si nous avons un faible pourcentage de conflits entre les agents ($Taux \leq 10$), le mécanisme P-NRB est meilleur que le mécanisme S-NRB

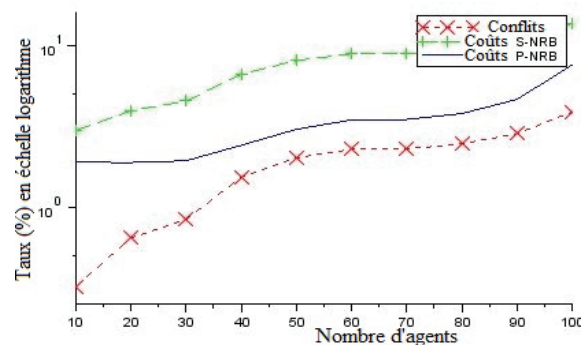


FIGURE 4.13 – Coûts en matière de messages jusqu'à la convergence de S-NRB et P-NRB, lorsque conflits $\sim P(\lambda)$ où $\lambda=0.5$.

en matière de coûts de messages. Cela, lorsque les conflits ont des distributions de probabilité Uniforme, Normale ou de Poisson. Cependant, si ce taux de conflits est supérieur à 10, le mécanisme S-NRB fournit de meilleurs coûts de messages que le mécanisme P-NRB.

Ainsi, nous déduisons que, l'augmentation du taux de conflits des agents affecte moins le mécanisme S-NRB. Cela peut s'expliquer par le fait que, S-NRB effectue une résolution séquentielle des conflits entre les agents et il privilégie les agents voisins de l'agent initiateur de l'offre composée.

Dans un deuxième temps, nous étudions le temps moyen nécessaire jusqu'à la formation des premières coalitions stables. Nous soumettons S-NRB et P-NRB aux mêmes contextes de simulations et lorsque les conflits dus aux préférences et dépendances des agents ont des distributions de probabilité Uniforme, Normale ou de Poisson.

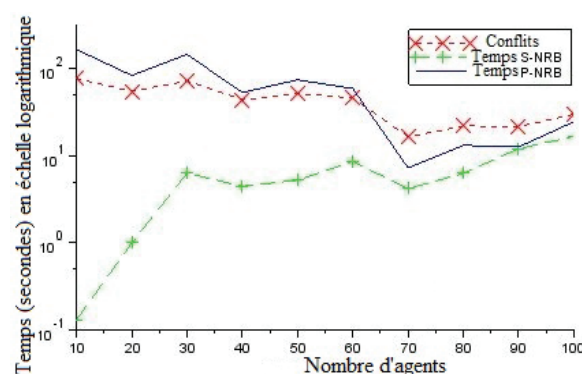


FIGURE 4.14 – Temps moyen requis pour la convergence de S-NRB et de P-NRB, si conflits $\sim U(a, b)$ où $a=0$ et $b=10$).

Sans conflits, les figures 4.14, 4.15 et 4.16 montrent que, le mécanisme S-NRB converge avant le mécanisme P-NRB. Cela s'explique par le fait que, la résolution des conflits et les prises de décisions se font de manière séquentielle dans le cas de S-NRB. La convergence de S-NRB dépend du pourcentage de conflits, du nombre d'agents requis pour former les coalitions nécessaires à

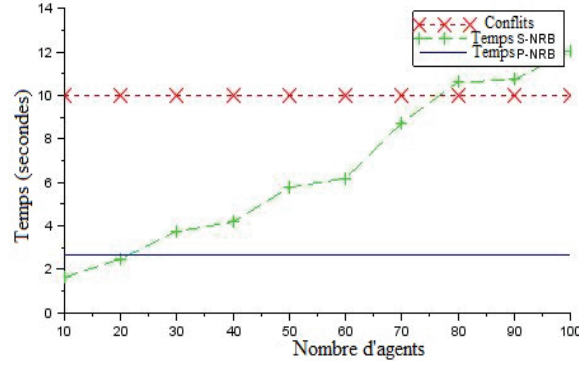


FIGURE 4.15 – Temps moyen requis pour la convergence de S-NRB et de P-NRB, si conflits $\sim N(\mu, \sigma^2)$ où $\mu=0.5$ et $\sigma=0.1$.

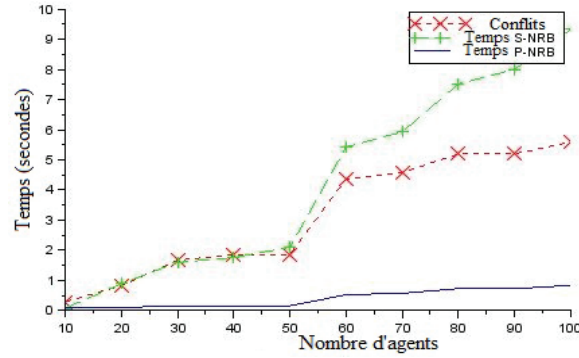


FIGURE 4.16 – Temps moyen requis pour la convergence de S-NRB et de P-NRB, si conflits $\sim P(\lambda)$ où $\lambda=0.5$.

l'exécution des tâches et de la position des agents dans l'environnement multi-agents. Cependant, P-NRB ne dépend que du pourcentage des conflits des agents.

La figure 4.14 montre que, jusqu'à 90 agents, si *conflits* $\sim U(a, b)$ où $a=0$ et $b=10$, alors le temps moyen nécessaire à la convergence de S-NRB est plus faible que le temps nécessaire à la convergence de P-NRB. Cependant, avec plus de 90 agents, les deux mécanismes ont des temps de convergence presque similaires.

Les figures 4.15 et 4.16 montrent que, le temps moyen requis pour la convergence de P-NRB reste toujours plus faible que celui de S-NRB pour tout nombre d'agents supérieur à 20 si *conflits* $\sim N(\mu, \sigma^2)$ où $\mu=0.5$ et $\sigma=0.1$ ou si *conflits* $\sim P(\lambda)$ où $\lambda=0.5$.

En conclusion, nous pouvons dire que, même si le coût en matière de messages est plus élevé avec P-NRB s'il y a un taux de conflits supérieur à 10, le mécanisme P-NRB converge toujours avant le mécanisme S-NRB.

Notons que, même si le mécanisme S-NRB a parfois des performances meilleures que le mécanisme P-NRB, S-NRB n'est pas *anytime* et de plus, il ne garantit pas que c'est les meilleures

coalitions *A-cores* qui sont formées. Nous expliquons ce dernier point par le fait qu'avec S-NRB, les agents qui ont une plus grande probabilité d'être retenus pour former une coalition, sont les agents proches de celui qui a initié l'offre de formation de coalitions. Cela, sans garantie que cette coalition est la meilleure solution.

4.5.2 Comparaison fonctionnelle entre S-NRB et P-NRB

Les résultats expérimentaux nous ont permis de conclure du point de vue fonctionnelle sur les mécanismes S-NRB et P-NRB.

Lorsque les agents qui sont proches de l'agent qui a initié l'offre composée ne permettent pas de former une coalition *A-core*, P-NRB converge plus rapidement que S-NRB. Cela est dû à la façon dont les sondages, les résolutions des conflits et les prises de décisions sont effectués par le mécanisme P-NRB.

Avec S-NRB, la résolution des conflits entre les agents ne peut s'effectuer que de manière séquentielle, car les décisions sont décentralisées mais séquentielles. Cela peut réduire les performances en matière de rapidité de convergence du processus de formation de coalitions.

Comme le montre les figures 4.14, 4.15 et 4.16, S-NRB dépend du nombre d'agents et de la position des ressources requises dans l'environnement de déploiement. Ainsi, nous pouvons noter une difficulté de passage à l'échelle de S-NRB. De plus, il peut y avoir une impossibilité de trouver une liste de sondage définitive. Cela est dû au fait que, l'association des décisions locales s'effectue de manière séquentielle et du point de vue de l'agent qui doit effectuer cette fusion des décisions, c'est-à-dire, suivant son voisinage.

Avec S-NRB, comme les agents plus proches de l'agent qui a émis l'offre de formation de coalitions initiale sont toujours privilégiés, cela peut limiter la formation des meilleures coalitions *A-core*. Cependant, cela peut aussi accélérer considérablement la convergence du processus de formation de coalitions. Raison pour laquelle, comme le montrent nos résultats de simulations, si le nombre d'agents n'est pas très élevé et si le taux de conflit est faible ou nul, S-NRB converge toujours avant P-NRB.

Une illustration de cette différence majeure entre S-NRB et P-NRB est exemplifiée par la figure 4.17 ci-dessous où nous effectuons une comparaison entre deux étapes des deux mécanismes.

Dans cette figure 4.17, nous considérons la même offre dans le cas de S-NRB et de P-NRB. De plus, nous supposons les mêmes délais de communication, les mêmes taux de conflits et la même répartition des ressources. Ainsi :

- Lorsque S-NRB est à l'étape (a) (c'est-à-dire, diffusion de l'offre de a_1 avec un $TTL = 1$, sondage et

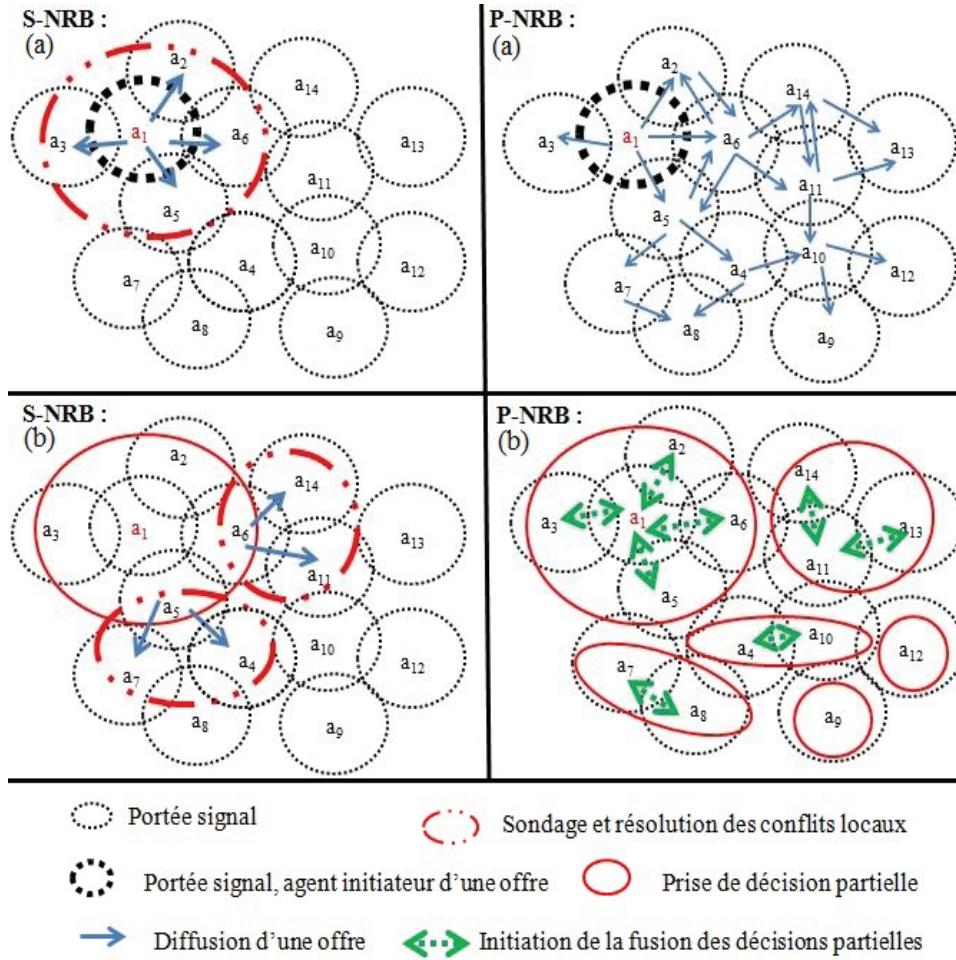


FIGURE 4.17 – Comparaisons fonctionnelles entre S-NRB et P-NRB.

résolution des conflits des agents de $\theta_{a_1}^t$), avec P-NRB les agents sont dans une étape de diffusion de l'offre de a_1 dans tout l'environnement de déploiement en mode non-return broadcast où chacun calcule aussi son rang.

- À l'étape (b) de S-NRB, on observe le début de la formation de la liste de sondage par les agents $a_1 \cup \theta_{a_1}^t$, une diffusion de l'offre de a_1 et une résolution des conflits avec les agents de $\theta_{a_5}^t$ et $\theta_{a_6}^t$. Par contre avec P-NRB, à l'étape (b), il y a la formation des *niveau-discussions*, la prise de décisions de chaque *niveau-discussion* et la détermination des agents responsables de la fusion des décisions entre les *niveau-discussions*.

On remarque qu'à ce stade du déroulement des deux mécanismes, tous les agents de l'environnement de déploiement ont déjà reçu l'information concernant l'offre de a_1 dans le cas du P-NRB ce qui n'est pas le cas pour S-NRB. Cela montre pourquoi P-NRB trouve toujours les meilleures coalitions A-cores contrairement à S-NRB. En effet, dans le cas du P-NRB tous les agents

sont informés en même temps et les agents qui sont retenus sont ceux qui fournissent une meilleure stabilité aux coalitions.

Dans le tableau 5.1 ci-dessous, nous effectuons un récapitulatif des propriétés importantes afin de mettre en évidence les forces et les faiblesses de chacun des deux mécanismes.

	Décentralisé	A-Core	Coût messages	Décision en parallèle	Anytime
S-NRB	Oui	Oui	$1 \leq \bar{M}_{sg} \leq 2 + 3N$	Non	Non
P-NRB	Oui	Oui	$0 \leq \bar{M}_{sg} \leq [\frac{N*(N-1)}{2} + 2]$	Oui	Oui

TABLE 4.1 – *Autres comparaisons entre S-NRB et P-NRB pour N agents. Ici, nous voyons qu'il y a un compromis entre coûts de messages et meilleures performances. S-NRB est meilleur $\forall N > 7$.*

Nous considérons que S-NRB n'est pas *anytime*, car s'il y a une interruption prématurée du processus de formation de coalitions, S-NRB ne fournit que dans certains cas des solutions. Par exemple, lorsque les agents du voisinage direct de l'agent qui a initié l'offre composée suffisent pour former la coalition.

CONCLUSION DU CHAPITRE

Les nouvelles applications multi-agents soulèvent de nouveaux défis. Il est nécessaire de permettre aux agents de collaborer et de former des coalitions stables tout en tenant compte du contexte stochastique des tâches et des incertitudes sur les préférences, les dépendances et les disponibilités des agents. Pour aborder ces difficultés, nous avons proposé un mécanisme de formation de coalitions, nommé P-NRB qui est adaptatif, décentralisé et asynchrone. P-NRB permet, la formation de coalitions *core stables* capables de se stabiliser dynamiquement. Nous appelons ici ces dernières, des coalitions *A-core*. De plus, P-NRB a la propriété *anytime*.

Notre mécanisme P-NRB ne suppose aucune connaissance a priori sur les coalitions stables à former et considère qu'il n'est pas possible de calculer de manière centralisée les coalitions à former à cause des incertitudes sur les agents et sur les tâches.

P-NRB combine des méthodes de la théorie des jeux avec le formalisme des *MDP* (Markov Decision Process) et les lois de probabilité. Les analyses théoriques et expérimentales montrent, que P-NRB surmonte de manière significative les difficultés posées par les incertitudes du contexte, pour former les coalitions *A-core*. De plus, nos résultats montrent que, pour différentes distributions de probabilité et de configurations multi-agents, notre mécanisme P-NRB est judicieusement utilisable pour la collaboration et la formation de coalitions dans des environnements incertains.

CONCLUSION GÉNÉRALE

Synthèses de nos travaux de thèse

L'une des plus importantes approches qui vise à faciliter la collaboration et la coordination multi-agents où chaque agent a un but à atteindre est la formation de coalitions. Nous avons abordé, dans cette thèse, le problème de la coordination dans les systèmes multi-agents. Plus précisément, la proposition de modèles de coordination par formation de coalitions stables dans des contextes incertains et ad-hoc. Nous nous sommes intéressés, en particulier, aux systèmes composés d'agents égoïstes qui doivent exécuter des tâches stochastiques.

Pour y parvenir nous avons, tout d'abord, étudié l'état de l'art des différents travaux qui se font sur la formation de coalitions d'agents, tout en passant en revue les notions fondamentales à la compréhension des problématiques relatives aux systèmes multi-agents en général et aux systèmes dotés d'agents égoïstes en particulier. Cette description nous a permis de définir les notions fondamentales à considérer, les problématiques existantes et de positionner le contexte de nos travaux de thèse.

Parmi les mécanismes de formation de coalitions, certains ne cherchent qu'à former les coalitions requises pour que les agents puissent atteindre leurs buts ou exécuter leurs tâches tandis que d'autres mécanismes ajoutent à cela le fait que les coalitions doivent être stables en respectant certaines propriétés de stabilité. Nous avons détaillé les types de stabilité recherchée dans le chapitre 1.

De ce constat, nous avons classé ces mécanismes en deux catégories. Dans la première, nous présentons les mécanismes de formation de coalitions ne faisant pas d'inférence dynamique pour pouvoir adapter les contraintes et les stabilités des coalitions. Dans la seconde, nous mettons les contributions qui recherchent la stabilité des coalitions en tenant compte des phénomènes dynamiques qui peuvent interférer avec le processus normal de la formation de coalitions. Cette dernière catégorie nous a permis de tracer les frontières de nos contributions et de les différencier des travaux qui sont proposés jusque-là sur la formation de coalitions.

En effet, dans la plupart des scénarios réalistes (par exemple, déploiement des agents dans un environnement sinistré), seuls certains agents sont capables de coopérer et d'initier des actions collectives. Cependant, pour entreprendre des actions collectives, les agents doivent trouver un état de consensus qui est bénéfique pour chacun d'eux, d'établir des accords sur ces consensus et coordonner les actions de chaque membre de manière collective. Cela implique que la formation de coalitions d'agents ne peut pas se faire de manière arbitraire par des agents qui ne sont pas connectés ou liés d'une certaine manière les uns aux autres. La plupart des mécanismes de formation de coalitions actuelles prennent en considération ces contraintes, ouvrant ainsi d'autres axes de recherche pour l'aide à la prise de décision des humains dans des contextes où des humains ayant des équipements communicants doivent établir des équipes efficaces et aptes à exécuter leurs tâches et à atteindre leurs buts.

Cependant, plusieurs contextes d'évolution des applications concrètes sont caractérisés par une dynamique des tâches à exécuter et par des changements parfois imprévisibles de leurs actions pour s'adapter à leur environnement dû au fait que les humains sont le plus souvent autonomes et libres de participer à l'exécution des tâches ou non. Ainsi, utiliser des techniques de coordination multi-agent s'appuyant sur la formation de coalitions pour les contextes réalistes, requiert un mécanisme de formation de coalitions dynamique susceptible d'engendrer des coalitions capables de se stabiliser dynamiquement.

Pour cette raison, ce travail de thèse a abordé un problème original et difficile pour la formation de coalitions stables où la robustesse face aux événements dynamiques (indisponibilité des agents, amplification des tâches, etc.) et la stabilité des coalitions sont difficiles à prévoir avant le début de l'exécution des tâches du fait des incertitudes des agents et sur les tâches stochastiques à exécuter.

Dans un premier temps, nous avons proposé un mécanisme de formation de coalitions dynamique fondé sur : 1- notre mécanisme de *recommandation*, 2- un ensemble de lois de probabilité, 3- les chaînes de Markov pour le calcul du risque de dissolution des coalitions afin de simplifier la recherche des consensus entre les agents et la formation dynamique de coalitions qui sont dans un état d'équilibre de Nash. Nous avons aussi proposé deux protocoles d'exploration pour permettre une stabilisation dynamique des coalitions au fil des événements impliquant les agents ou les tâches, c'est-à-dire la formation de coalitions persistantes. Le premier est un protocole d'exploration localisée nommé *DEP* (*Depth Exploration Protocol*) et le second est un protocole d'exploration étendue nommé *WEP* (*Width Exploration Protocol*). Nos résultats analytiques et expérimentaux montrent l'efficacité de notre mécanisme décentralisé lorsqu'il est nécessaire de former des coalitions stables du point de vue de Nash et lorsqu'il est nécessaire de maintenir leur persistance dans

un contexte dynamique et incertain.

Cette proposition soulève aussi, le problème de l'optimalité des coalitions formées, c'est-à-dire, la recherche de coalitions capables de s'auto-stabiliser dynamiquement où le bien-être social des agents est maximisé et où il y a des incertitudes sur les dépendances des agents et sur les tâches.

Suite à cela, nous avons proposé dans un deuxième temps, 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. Ce mécanisme s'applique même en présence d'incertitudes sur : - les dépendances et les conflits qui peuvent apparaître entre les agents pour une formation de coalitions, - les préférences et les disponibilités des agents, et - les coalitions à former pour les tâches stochastiques à exécuter dans un contexte ouvert (les agents peuvent être indisponibles à tout moment). À cause de ces incertitudes et des contraintes sur les tâches et les agents, nous ne supposons aucune connaissance a priori sur les coalitions stables à former. Nous considérons 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.

Nous avons proposé un mécanisme de formation de coalitions *core stables* que nous avons appelé S-NRB. Nous avons, principalement, prouvé sa convergence vers des coalitions *core stables* qui se stabilisent dynamiquement dans le temps (auto-stabilité) en cas de disparition d'un ensemble d'agents ou de changements sur les contraintes des tâches. Notre mécanisme S-NRB utilise : 1- un ensemble de lois de probabilité pour prédire les événements qui peuvent changer les états des tâches ou pour prédire les indisponibilités des agents, 2- les principes de la théorie des jeux pour identifier les coalitions *core stables*. L'analyse théorique et les simulations analytiques et expérimentales montrent le fonctionnement de notre mécanisme S-NRB. En effet, en distribuant le contrôle et les prises de décision lors des processus de formation de coalitions, ce mécanisme permet de surmonter les incertitudes et les contraintes du contexte considéré afin de trouver les coalitions *A-cores*.

Pour aborder les difficultés non résolues par S-NRB, nous avons proposé dans un troisième temps, un mécanisme de formation de coalitions, nommé P-NRB qui est adaptatif, décentralisé et asynchrone. P-NRB permet la formation de coalitions *core stables* capables de se stabiliser dynamiquement. Nous appelons ici ces dernières, des coalitions *A-core*. De plus, P-NRB a la propriété *anytime*.

Notre mécanisme P-NRB ne suppose aucune connaissance a priori sur les coalitions stables à former et considère qu'il n'est pas possible de calculer de manière centralisée les coalitions à former à cause des incertitudes sur les agents et sur les tâches.

P-NRB combine des méthodes de la théorie des jeux avec le formalisme des MDP (Markov Decision Process) et les lois de probabilité. Les analyses théoriques et expérimentales montrent que P-NRB

surmonte de manière significative les difficultés posées par les incertitudes du contexte, pour former les coalitions *A-core*. De plus, nos résultats montrent que, pour différentes distributions de probabilité et de configurations multi-agents, notre mécanisme P-NRB est judicieusement utilisable pour la collaboration et la formation de coalitions dans des environnements incertains.

Perspectives issues de la thèse

Toutes ces propositions soulèvent aussi quelques directions de recherche qui méritent d'être explorées dans le futur.

La première est comment adapter dynamiquement le comportement des agents lorsque leurs ressources sont insuffisantes pour exécuter l'ensemble des tâches et comment tenir compte des interdépendances prioritaires des agents et de la priorité des tâches stochastiques en présence d'incertitudes sur ces agents et sur ces tâches.

La seconde est comment permettre une coopération dynamique des coalitions pour mieux assurer leurs persistance et leurs stabilités, pour augmenter leur efficacité et leurs aptitudes à exécuter les tâches. Cette coopération entre les coalitions ne doit pas être limitée à une décision propre aux agents qui souhaitent partager leurs ressources entre plusieurs coalitions sans que ce soit une prise de décision globale de leur coalition.

La troisième et peut-être la principale est la recherche de coalitions capables de s'auto-stabiliser dynamiquement où le bien-être social des agents est maximisé et où il y a des incertitudes sur les externalités des coalitions qui dépendent à la fois des agents et des tâches stochastiques.

A | ANNEXES

Les notions sur les lois de probabilité et les chaînes de Markov ci-dessous sont tirées de (Baynat 2000), (Yates et Goodman 2005) et (Ortner 2007).

A.1 RAPPELS SUR LOIS DE PROBABILITÉ USUELLES

Dans cette section, nous donnons les détails des différentes lois de probabilités sur lesquelles nous nous fondons le plus souvent pour nos prédictions d'événements stochastiques qui peuvent affecter le processus de formation de coalitions.

A.1.1 Reconnaissance d'une loi de Bernoulli

Pour traiter ce type de question il convient de commencer par considérer que l'événement dont on connaît la probabilité (notée en général p) est un succès. En conséquence de quoi l'événement contraire constitue un échec. Le succès sera noté $[X_i=1]$ et l'échec sera noté $[X_i=0]$.

On a ainsi défini une variable aléatoire X_i de Bernoulli qui peut prendre 2 valeurs : 0 et 1. On note alors $X_i(\Omega) = \{0; 1\}$; $P[X_i=1] = p$ et $P[X_i=0]=1-p=q$.

A.1.2 Reconnaissance d'une loi Binômiale

On considère la succession de n essais identiques et indépendants les uns des autres d'une même épreuve ayant deux issues possibles : le succès avec la probabilité p et l'échec avec la probabilité $q=1-p$. On définit une variable X égale au nombre de succès obtenus à l'issue de ces n épreuves. X correspond à la succession (ou à la somme) de n épreuves de Bernoulli (puisque chaque essai a deux issues possibles : le succès et l'échec) identiques et indépendantes. On dit alors que X suit une loi Binômiale définie par 2 paramètres n et p .

n est le nombre d'épreuves de Bernoulli identiques et indépendantes. p est la probabilité du succès pour chaque épreuve.

Cette situation se rencontre lorsque l'on considère une urne comportant des boules de deux couleurs différentes et que l'on cherche la loi du nombre de boules d'une des deux couleurs à l'issue de n tirages (d'une boule à chaque tirage) avec remise. En effet, les tirages sont identiques si l'on suppose que l'on prélève toujours une boule à chaque tirage. En outre, si la boule tirée est systématiquement remise dans l'urne, la composition de l'urne est constante. Dès lors, la probabilité d'obtenir une boule d'une couleur donnée est la même à chaque tirage. Les tirages sont donc indépendants les uns des autres.

Exemple A.1 On considère une entreprise de service après-vente (SAV) qui intervient avec retard avec une probabilité égale à 0,25. Un client a appelé à 8 dates différentes.

- 1) Préciser la loi de X , son espérance et sa variance.
- 2) Calculer la probabilité que ce client soit victime d'au moins un retard.
- 3) Calculer la probabilité que ce client soit victime d'au moins 4 retards sachant qu'il en a subi au moins un.

1) Nous appelons succès l'événement dont on connaît la probabilité (à savoir une intervention avec un retard) que nous notons $[X_i=1]$ et échec l'événement contraire (à savoir une intervention ponctuellement) que nous notons $[X_i=0]$. Ainsi,

$$P[X_i=1]=0.25 \text{ et } P[X_i=0]=1-0.25=0.75, \text{ avec } X_i(\Omega) = \{0, 1\}.$$

A l'issue de chaque appel, la probabilité d'intervention avec retard est la même. Les appels peuvent donc être supposés indépendants les uns des autres. Par conséquent, X est la succession de 8 épreuves de Bernoulli identiques et indépendantes. X suit donc une loi Binômiale, que nous pouvons noter, $X \rightarrow B(8, 0.25)$. Ainsi, $P[X = k] = C_8^k 0.25^k 0.75^{8-k}$.

$$E(X) = 8 * 0.25 = 2.$$

$$V(X) = 8 * 0.25 * 0.75 = 1.5.$$

$$2) P[X \geq 1] = P[(X = 1) \text{ ou } (X = 2) \text{ ou } \dots \text{ ou } (X = 8)] = P[(X = 1) \cup (X = 2) \cup \dots \cup (X = 8)].$$

Il est évident que $[X \geq 1]$ est l'événement contraire de $[X < 1]$. De ce fait nous pouvons écrire aussi $P[X \geq 1] = 1 - P[X \leq 1]$. Dès lors,

$$P[X \geq 1] = 1 - P[X = 0] = 1 - C_8^0 0.25^0 0.75^{8-0} = 0.899.$$

$$3) P[X \geq 4 / X \geq 1] = \frac{P[X \geq 4 \cap X \geq 1]}{P[X \geq 1]} = \frac{P[1 \leq X \leq 4]}{P[X \geq 1]}.$$

$$P[1 \leq X \leq 4] = P[X = 1] + P[X = 2] + P[X = 3] + P[X = 4].$$

$$P[1 \leq X \leq 4] = C_8^1 0.25^1 0.75^{8-1} + C_8^2 0.25^2 0.75^{8-2} + C_8^3 0.25^3 0.75^{8-3} + C_8^4 0.25^4 0.75^{8-4}.$$

A.1.3 Reconnaissance d'une loi Hypergéométrique

On effectue n tirages sans remise dans un ensemble (exemple, populations, boules, ...) de taille N . On recherche alors la loi du nombre de réalisations d'un événement dont la probabilité de réalisation, avant que ne commencent les tirages, est égale à p .

Ici, la composition de l'ensemble considéré change à l'issue de chaque tirage puisque l'élément de l'ensemble (personne, boule, ...) qui vient d'être tiré(e) n'est pas remis(e) dans son ensemble d'origine. Si X est le nombre de succès, ou de réalisation de l'événement dont on connaît la probabilité avant que ne commencent les tirages, alors X définit une loi hypergéométrique de paramètres N , n et p .

Cette loi de tirages sans remise est également la loi des sondages. En effet, si l'on interroge n per-

sonnes issues d'une population globale de taille N , chaque individu est interrogé une seule fois. En d'autres termes, il n'est pas remis dans la population initiale après avoir été sondé.

Exemple A.2 Pour le même exemple que précédemment, on considère 8 clients différents. On en contacte 4. On admet qu'un client est mécontent s'il a fait l'objet d'une intervention avec retard. On note M le nombre de mécontents. Donner la loi de M , son espérance et sa variance.

M correspond, dans le cadre d'un sondage, au tirage de 4 clients dans une population de 8 clients sans remise. Nous pouvons en effet penser que la personne chargée de l'étude de satisfaction n'interroge chaque client qu'une seule fois. En outre, avant de commencer les tirages des individus, on connaît la probabilité d'intervention avec retard, donc de mécontentement est de 0.25. Donc nous pouvons estimer que X suit une loi Hypergéométrique qu'on note $X \rightarrow H(8, 4, 0.25)$. $P[X = k] = \frac{C_8^{0.25^k} C_8^{0.75^{4-k}}}{C_8^4}$

$$E(X) = 4 * 0.25 = 1.$$

$$V(X) = 4 * 0.25 * 0.75 * \frac{8-4}{8-1} = 0.75 * \frac{4}{7}$$

A.1.4 Reconnaissance d'une loi géométrique

On effectue n épreuves de Bernoulli identiques et indépendantes dont la probabilité de succès est p pour chacune d'elles. La probabilité de l'échec pour chacune d'elles est notée $q=1-p$.

Si nous considérons que X est le rang du 1^{er} succès, alors $P[X=k]$ correspond à la probabilité que le 1^{er} succès soit obtenu à l'issue de la k -ième épreuve. En d'autres termes, les $k-1$ premières épreuves se sont soldées par des échecs et la k -ième par un succès.

Cette loi s'observe assez fréquemment en étude de performances systèmes afin d'évaluer de manière probabiliste, l'instant d'une première panne de transmission réseau, de saturation réseau, d'une panne d'un composant informatique, etc.

Exemple A.3 En utilisant l'exemple précédent, nous étudions les lois des appels reçu par le service SAV. Soit Y la loi du rang du 1^{er} retard. Préciser la loi de Y , son espérance et sa variance.

Les appels qui définissent une loi de Bernoulli étant identiques et indépendants, le temps d'attente du 1^{er} retard définit une loi géométrique de même paramètre que la loi de Bernoulli précédemment évoquée. Ainsi, $Y \rightarrow G(0.25)$.

$$P[X = k] = 0.25 * 0.75^{k-1}.$$

$$E(X) = \frac{1}{0.25} = 4.$$

$$V(x) = \frac{0.75}{0.25^2}.$$

A.1.5 Reconnaissance d'une loi de Poisson

On observe un ensemble d'événement durant un intervalle de temps. Si nous cherchons la répétition d'un événement (succès, échec, ...) durant cet intervalle de temps, alors nous nous retrouvons dans une position où nous pouvons appliquer la loi de Poisson.

De même c'est une loi usuelle en étude de performance systèmes pour la prédiction de performance durant un temps déterminé, grâce à l'utilisation d'un prototype du système informatique à évaluer. C'est une loi qui est applicable, par exemple, pour estimer le nombre de fois qu'un composant informatique peut être en panne durant un intervalle de temps où il est utilisé, le nombre de fois qu'un serveur aura à exécuter des requêtes spécifiques durant un intervalle de temps, etc.

La loi de Poisson résulte de la convergence de la loi Binômiale. En d'autres termes, lorsque n est très grand ($n \geq 30$) et sous réserve que $p \leq 0,1$ et $np < 15$. Dans ce cas, on peut remplacer la loi Binômiale $B(n;p)$ par une loi de poisson de paramètre $\lambda=np$. On a alors, $X \rightarrow P(\lambda)$.

Exemple A.4 Reprenons le même exemple, et considérons que X suit une loi de Poisson de paramètre 4. Calculons $P[X=5]$, $E(X)$ et $V(X)$.

$$P[X = 5] = e^{-4} * \frac{4^5}{5!}.$$

$E(X)=V(X)=4$, car la moyenne et la variance sont égales au paramètre de la loi de Poisson.

A.2 RAPPELS SUR LES PROCESSUS DE MARKOV

Les processus stochastiques sont des modèles mathématiques des phénomènes aléatoires se déroulant et évoluant dans le temps. Un processus stochastique est une famille de variables aléatoires $X(t)$ définies sur une même espace de probabilité. Le comportement de tels processus ne peut être décrit à l'avance d'une manière déterministe.

Si on se donne un ensemble des états E d'un processus stochastique et un ensemble T des instants où l'on considère le processus, alors T est soit continu ou discret. Cela implique que nous pouvons avoir soit des processus stochastiques à temps continu ou à temps discret.

Les chaînes de Markov sont des processus stochastiques ayant la propriété sans mémoire. La propriété sans mémoire signifie que, si $(X_t)_{t \in T}$ est un processus de Markov à valeurs dans E et T est soit $\mathbb{R}^+$ soit $\mathbb{N}$, alors

$$P(X_t = E_t | X_{t_n} = E_{t_n}, X_{t_{n-1}} = E_{t_{n-1}}, \dots, X_{t_0} = E_{t_0}) = P(X_t = E_t | X_{t_n} = E_{t_n}),$$

où $t \geq t_n \geq t_{n-1} \geq \dots \geq t_0 : T=(t_0, \dots, t_{n-1}, t_n)$ est l'ensemble des instants d'observation des

valeurs de $X(t)$ et où $E=(E_{t_0}, \dots, E_{t_{n-1}}, E_{t_n})$ est l'ensemble des états de $X(t)$. Donc, l'état courant résume, à lui seul, tout l'historique susceptible d'influencer son évolution future.

Définition A.1 Une chaîne de Markov à temps discret est homogène (dans le temps) si, pour tout paire d'états (E_{t_i}, E_{t_j}) et pour tout instant t , $P(X_{t_n} = E_{t_j} | X_{t_{n-1}} = E_{t_i}) = P(X_{t_{n+k}} = E_{t_j} | X_{t_{n+k-1}} = E_{t_i})$, quelque soit $k \geq 0$.

A.2.1 Probabilité de transition et matrice de transition

Pour une chaîne de Markov homogène $\{X_{t_n}, n = 0, 1, \dots\}$, on a aussi $P(X_{t_n} = E_{t_j} | X_{t_{n-1}} = E_{t_i}) = P(X_{t_1} = E_{t_j} | X_{t_0} = E_{t_i})$, $\forall n \geq 1$. On peut donc définir la probabilité de transition en une étape, de l'état E_{t_i} à E_{t_j} comme $p_{ij} = P(X_{t_1} = E_{t_j} | X_{t_0} = E_{t_i}) \forall (E_{t_i}, E_{t_j}) \in E^2$.

Cela signifie que, la probabilité p_{ij} est égale à la probabilité conditionnelle que la chaîne de Markov se trouve dans l'état E_{t_j} à l'étape suivante sachant qu'il se trouve actuellement à l'état E_{t_i} .

Si la chaîne possède $s = |E|$ états, les probabilités précédentes peuvent être rangées dans une matrice de transition $P = (p_{ij})$ de taille $s \times s$ dont les lignes et les colonnes sont indexées par les éléments de E . Cette matrice P est donc carrée.

Définition A.2 Une matrice de transition $P = (p_{ij})$ est stochastique si, ses éléments sont non négatifs ($\forall E_{t_i}$ et E_{t_j} , $p_{ij} \geq 0$) et la somme des éléments de chacune de ses lignes est égale à 1 ($\forall E_{t_i}$, $\sum_j p_{ij} = 1$)

Propriété A.1 Si P est une matrice stochastique de taille s finie, alors toute puissance P^m , $m \geq 0$, de P est aussi une matrice stochastique.

Démonstration. Le résultat est vrai pour $m=0$, car $P^0=I$ pour toute matrice P carrée. Si $m \geq 1$, alors $(P^m)_{ij} \geq 0$, $\forall i, j$ et, $\forall i$, $\sum_j (P^m)_{ij} = \sum_j \sum_k (P^{m-1})_{ik} P_{kj} = \sum_k [(P^{m-1})_{ik} \sum_j P_{kj}] = 1$. $\square$

Propriété A.2 Une matrice de transition d'une chaîne de Markov est toujours une matrice stochastique.

A.2.2 Probabilités de transition en m étapes

La probabilité conditionnelle d'aller de i à j en m étapes exactement est donnée par :

$$p_{ij}^{(m)} = P[X_m = j | X_0 = i] = P[X_{n+m} = j | X_n = i] \forall n \geq 1.$$

Cette probabilité est indépendante de n , car le processus est homogène et est appelée la probabilité de transition en m étapes de i à j .

La matrice $P^{(m)}$ dont l'élément (i, j) est égal à $p_{ij}^{(m)}$ est appelée la matrice de transition en m étapes. Il est évident que $p_{ij}^{(1)} = p_{ij}$ et $P^{(1)} = P$.

Théorème 10 Pour tout $m \geq 0$, la probabilité $p_{ij}^{(m)}$ de transition de i à j en m étapes est donnée par l'élément (i, j) de la matrice P^m .

Démonstration. Le résultat est vrai pour $m=0$ et $m=1$. Supposons-le vrai pour $m-1$. En utilisant la loi des probabilités totales, nous obtenons, pour tout $i, j \in E$,

$$p_{ij}^{(m)} = P[X_m = j | X_0 = i]$$

$$p_{ij}^{(m)} = \sum_{k \in E} P[X_m = j | X_{m-1} = k] P[X_{m-1} = k | X_0 = i]$$

$$p_{ij}^{(m)} = \sum_{k \in E} p_{kj}^{(1)} p_{ik}^{(m-1)} = \sum_{k \in E} p_{kj} p_{ik}^{(m-1)}$$

$$p_{ij}^{(m)} = \sum_{k \in E} (P)_{kj} (P^{(m-1)})_{ik}$$

$$p_{ij}^{(m)} = (P^{(m)})_{ij}.$$

□

Corollaire A.1 Si $P^n, n \geq 0$ est la matrice de transition en n étapes d'une chaîne de Markov, alors pour tout entier non négatif l et m , $P^{(l+m)} = P^{(l)} P^{(m)} \Leftrightarrow p_{ij}^{(l+m)} = \sum_{k \in E} p_{ik}^{(l)} p_{kj}^{(m)} \forall i, j \in E$.

Cette équation est l'équation de Chapman-Kolmogorov.

A.2.3 Graphes représentatifs et classification des états

La matrice de transition P d'une chaîne de Markov peut être représentée par un graphe orienté G . Les sommets sont les états E de la chaîne. Les arcs représentent les probabilités de transition positive, c'est-à-dire, si un arc relie les sommets associés aux états i et j , alors $p_{ij} > 0$.

Ainsi défini, G est appelé le graphe représentatif, ou graphe de transition, de la chaîne de Markov.

Exemple A.5 La matrice de transition d'une chaîne de Markov est :

$$\begin{bmatrix} \frac{1}{4} & \frac{3}{4} & 0 & 0 \\ 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \end{bmatrix}$$

alors les états de la chaîne de Markov $E=(E_{t_0}, E_{t_1}, E_{t_2}, E_{t_3})$. Le graphe représentatif est sous la forme :

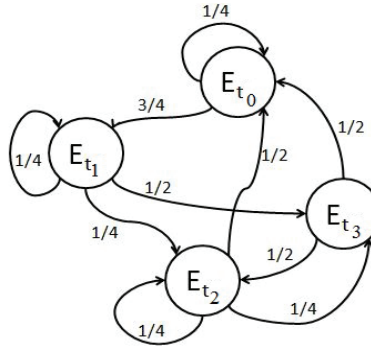


FIGURE A.1 – Graphe de transition de la chaîne de Markov.

Soit P la matrice de transition d'une chaîne de Markov et G le graphe représentatif de P . L'état j est accessible depuis l'état i s'il existe, dans G , au moins un chemin de i à j . À noter que tout état j est accessible depuis lui-même.

Propriété A.3 L'état j est accessible depuis l'état i si et seulement si il existe $n \geq 0$ tel que $p_{ij}^{(n)} > 0$.

Propriété A.4 Les états i et j communiquent s'ils sont accessibles l'un à partir de l'autre, c'est-à-dire, il existe $n \geq 0$ et $m \geq 0$ tels que $p_{ij}^{(n)} > 0$ et $p_{ji}^{(m)} > 0$.

La relation i et j communiquent est une relation d'équivalence dont les classes correspondent aux composantes fortement connexes de G . Ainsi, les états i et j communiquent si et seulement s'ils appartiennent à la même classe.

Les classes de la chaîne de Markov correspondent aux composantes fortement connexes de G .

Une classe est persistante si elle correspond à un sommet sans successeur dans G . Si tel n'est pas le cas, la classe est transitoire. Les états d'une classe persistante sont persistants ou récurrents et ceux d'une classe transitoire sont transitoires. Une classe persistante composée d'un seul état est absorbante et un état est absorbant s'il forme, à lui seul, une classe persistante.

Propriété A.5 L'état i est absorbant si et seulement si $p_{ii}=1$ et $p_{ij}=0 \forall i \neq j$.

Une chaîne de Markov est irréductible si elle ne compte qu'une seule classe. Dans le cas contraire, elle est réductible.

Propriété A.6 Une chaîne de Markov est irréductible si et seulement si son graphe représentatif est fortement connexe.

Propriété A.7 Une chaîne de Markov est irréductible si et seulement si toutes ses paires d'états communiquent.

Propriété A.8 Une chaîne de Markov est absorbante si tous ses états persistants sont absorbants.

A.2.4 Périodicité des chaîne de Markov

La période d de l'état i d'une chaîne de Markov est égale au plus grand diviseur commun de tous les n pour lesquels $p_{ii}^{(n)} > 0$. L'état i est périodique lorsque $d > 1$ et apériodique lorsque $d = 1$.

Propriété A.9 L'état i a une période d si et seulement si d est le plus grand diviseur commun des longueurs des circuits du graphe représentatif G passant par i .

Propriété A.10 Si $p_{ii} > 0$, l'état i est apériodique.

Propriété A.11 Les états d'une classe ont tous la même période.

La période étant une propriété de classe, on parlera de classes périodiques ou apériodiques et de chaînes de Markov irréductibles périodiques ou apériodiques selon les propriétés de leurs états.

A.2.5 Distribution initiale et comportement transitoire

La distribution des états d'une chaîne de Markov après n transitions est notée $\pi^{(n)}$. Cette distribution est un vecteur de probabilités contenant la loi de la variable aléatoire X_n .

$$\pi_i^{(n)} = P[X_n = i], \forall i \in E.$$

La distribution initiale est $\pi^{(0)}$. Dans le cas où l'état initial est connu avec certitude et est égal à i , on a $\pi_i^{(0)} = 1$ et $\forall j \neq i, \pi_j^{(0)} = 0$.

Théorème 11 Soit P la matrice de transition d'une chaîne de Markov et $\pi^{(0)}$ la distribution de son état initial. $\forall n \geq 1, \pi^{(n)} = \pi^{(n-1)}P$ et $\pi^{(n)} = \pi^{(0)}P^n$.

$$\text{Démonstration. } \forall j \in E, \pi_j^{(1)} = P[X_1 = j] = \sum_{i \in E} P[X_1 = j | X_0 = i] P[X_0 = i]$$

$$\pi_j^{(1)} = \sum_{i \in E} p_{ij} \pi_i^{(0)} = \sum_{i \in E} \pi_i^{(0)} p_{ij}.$$

Ainsi, $\pi^{(1)} = \pi^{(0)}P$. Si la chaîne de Markov est homogène, nous avons immédiatement $\pi^{(n)} = \pi^{(n-1)}P$ $\forall n \geq 1$.

Pour démontrer le second il suffit de résoudre l'équation de récurrence précédente par substitution. $\square$

A.2.6 comportement asymptotique des chaînes irréductibles

L'étude du comportement à long terme d'une chaîne de Markov cherche à vérifier est-ce que la distribution $\pi^{(n)}$ converge lorsque $n \rightarrow \infty$. Si tel est le cas, alors cette étude donne la limite π^* et la dépendante de π^* par rapport à la distribution initiale π^0 . De là, si l'état i est persistant, elle évalue la proportion du temps passée dans cet état et le nombre moyen de transitions entre deux visites successives de cet état. Si l'état i est transitoire, elle calcule le nombre moyen de visites de cet état. Une distribution π est invariante ou stationnaire si $\pi = \pi P$.

Propriété A.12 Si $\lim_{n \rightarrow \infty} \pi^n$ existe, alors la limite est une distribution invariante (asymptotique ou stationnaire).

Propriété A.13 Une chaîne de Markov possède toujours au moins une distribution invariante.

Théorème 12 La distribution π^n des états d'une chaîne de Markov converge vers une distribution invariante π^* indépendante de la distribution initiale π^0 si et seulement si la suite des puissances de la matrice de transition P de la chaîne converge vers une matrice stochastique P^* dont toutes les lignes sont égales entre elles. De plus, si tel est le cas, chaque ligne de P^* est égale à P^* .

Démonstration. La condition est nécessaire car si, indépendamment de $\pi^{(0)}$, $\lim_{n \rightarrow \infty} \pi^n = \pi^*$, il suffit de considérer successivement les distributions initiales

$$\pi_1^0 = (1 \ 0 \ 0 \ \dots \ 0), \pi_2^0 = (0 \ 1 \ 0 \ \dots \ 0) \ \dots \ \pi_s^0 = (1 \ 0 \ 0 \ \dots \ 0)$$

pour obtenir

$$\pi^* = \lim_{n \rightarrow \infty} \pi^n = \lim_{n \rightarrow \infty} \pi_i P^n = \lim_{n \rightarrow \infty} (P^n)_i = (P^*)_i.$$

Ainsi, P^* existe et toutes les lignes sont égales à π^* .

La condition est suffisante. Si, P^* existe et si $p_{ij}^* = p_j^* \ \forall i \in E$, on a :

$$\lim_{n \rightarrow \infty} \pi^n = \lim_{n \rightarrow \infty} \pi^{(0)} P^n = \pi^{(0)} \lim_{n \rightarrow \infty} P^n = \pi^{(0)} P^* \text{ et la limite } \pi^* \text{ existe. De plus,}$$

$$\pi_j^* = \sum_{i \in E} \pi_i^{(0)} p_{ij}^* = \sum_{i \in E} \pi_i^{(0)} p_j^* = p_j^* \sum_{i \in E} \pi_i^{(0)} = p_j^*$$

et π^* est indépendante de $\pi^{(0)}$ et identique à n'importe quelle ligne de P^* . $\square$

Ce qui suit résume le comportement asymptotique des chaînes irréductibles et apériodiques.

Propriété A.14 Soit P la matrice de transition d'une chaîne irréductible et apériodique, alors :

- La matrice P^n tend vers une matrice stochastique P^* lorsque n tend vers l'infini.
- Les lignes de P^* sont toutes égales entre elles.

- p_{ij}^* pour tout $i, j \in E$. - Pour toute distribution initiale $\pi^{(0)}$, $\lim_{n \rightarrow \infty} \pi^n = \lim_{n \rightarrow \infty} \pi_{(0)} P^n = \pi^*$, qui est la solution unique de l'équation $\pi P = \pi$: $\pi 1 = 1$.
- π^* est égal à n'importe quelle ligne de la matrice P^* .
- $\forall i \in E$, $\pi_i^* = \frac{1}{\mu_i}$, où μ_i est l'espérance du nombre de transitions entre deux visites successives de l'état i .

Donc, pour n suffisamment grand, $\pi^{(n)} \simeq \pi^*$. π_i^* est la probabilité que la chaîne se trouve dans l'état i à un instant quelconque. Cette valeur représente aussi la proportion du temps passé dans l'état i .

Dans le cas où une chaîne de Markov admet une distribution asymptotique unique et indépendante de la distribution initiale, on dit que la chaîne de Markov est ergodique.

Propriété A.15 Les chaînes irréductibles et apériodiques sont ergodiques.

BIBLIOGRAPHIE

- S. Abdallah et V. Lesser. Organization-based cooperative coalition formation. *IAT*, pages 162–168, September 2004. (Cité page 31.)
- S. Aknine et P. Caillou. Agreements without disagreements. *ECAI*, pages 3–7, August 2004. (Cité page 33.)
- S. Aknine, P. Caillou, et A. Slodzian. Méthode consensuelle de formation de coalitions multi-agents. *RFIA*, pages 979–988, January 2004a. (Cité page 33.)
- S. Aknine, S. Pinson, et M. F. Shakun. A multi-agent coalition formation method based on preference models. *GDN*, pages 513–538, November 2004b. (Cité page 32.)
- S. Arib et S. Aknine. A plan based coalition formation model for multi-agent systems. *IAT*, pages 365–368, 2011. (Cité page 35.)
- S. Arib et S. Aknine. Preferences and constraints for agent coalition formation. *IAT*, pages 130–137, 2013. (Cité page 35.)
- S. Arib, S. Aknine, et T. Génin. Handling agent’s incomplete information in a coalition formation model. *ACAN*, 2014. (Cité page 35.)
- J. P. Aubin. Cooperative fuzzy games. *Journal Mathematics of Operations Research*, 6 :1–13, February 1981. (Cité page 34.)
- K. Avrachenkov, L. Cottatellucci, et L. Maggi. Cooperative markov decision processes : time consistency, greedy players satisfaction, and cooperation maintenance. *International Game Theory Review*, pages 239–262, February 2013. (Cité page 39.)
- R. Axelrod. *The Evolution of Cooperation*. Basic Books, New York, 1984. (Cité page 33.)
- J. F. Banzhaf. Weighted voting doesn’t work : A mathematical analysis. *Rutgers Law Rev.*, pages 317–343, 1965. (Cité page 22.)

- S. Barberà et A. Gerber. On coalition formation : durable coalition structures. *Mathematical Social Sciences*, 45 :185–203, April 2003. (Cité page 35.)
- B. Baynat. *Théorie Des Files D'attente - Des Chaînes De Markov Aux Réseaux À Forme Produit*. Hermes Science Publications - Lavoisier, June 2000. (Cité pages 46, 47, 56, 74, 109 et 169.)
- E. T. Bell. Exponential numbers. *Amer. Math. Monthly*, 41 :411–419, 1934. (Cité pages 10 et 29.)
- B. Blankenburg et M. Klusch. On safe kernel stable coalition forming among agents. *The Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 580–587, July 2004. (Cité page 34.)
- A. Bogomolnaia et M. O. Jackson. The stability of hedonic coalition structures. *In Games and Economic Behavior*, pages 201–230, 2002. (Cité page 28.)
- G. Bonnet et C. Tessier. Multi-agent collaboration : A satellite constellation case. *STAIRS*, pages 24–35, July 2008. (Cité page 38.)
- S. Brânzei et K. Larson. Coalitional affinity games. *AAMAS*, pages 1319–1320, May 2009. (Cité page 38.)
- P. Caillou, S. Aknine, et S. Pinson. A multi-agent method for forming and dynamic restructuring of pareto optimal coalitions. *AAMAS*, pages 1074–1081, July 2002. (Cité page 37.)
- C. Castelfranchi, M. Miceli, et A. Cesta. Dependence relations among autonomous agents. *The 3rd European Workshop on Modeling Autonomous Agents in a Multi-Agent World*, pages 215–227, December 1992. (Cité page 36.)
- G. Chalkiadakis et C. Boutilier. Bayesian reinforcement learning for coalition formation under uncertainty. *AAMAS*, pages 1090–1097, July 2004. (Cité page 37.)
- G. Chalkiadakis, E. Elkind, E. Markakis, M. Polukarov, et N. Jennings. Cooperative games with overlapping coalitions. *Journal of AI Research*, pages 179–216, September 2010. (Cité pages 36 et 84.)
- G. Chalkiadakis, E. Elkind, et M. Wooldridge. *Computational Aspects of Cooperative Game Theory*, volume 5, No 6. Morgan & Claypool, October 2011. (Cité pages 1, 2, 9 et 34.)
- G. Chalkiadakis, E. Markakis, et N. R. Jennings. Coalitional stability in structured environments. *The 11th International Conference on Autonomous Agents and Multiagent Systems*, pages 779–786, June 2012. (Cité pages 33 et 90.)
- V. Conitzer et T. Sandholm. Complexity of determining nonemptiness of the core. *IJCAI*, pages 613–618, 2003. (Cité page 23.)

- V. Conitzer et T. Sandholm. Computing shapley values, manipulating value division schemes, and checking core membership in multi-issue domains. *AAAI*, pages 219–225, 2004. (Cité page 23.)
- R. Conte et J. S. Sichman. Depnet : How to benefit from social dependence. *Journal of Mathematical Sociology*, 20 :161–177, 1995. (Cité page 29.)
- J. Contreras, F. Wu, M. Klusch, et O. Shehory. Coalition formation in a power transmission planning environment. *International Conference on Practical Applications of autonomous Agents and Multi-Agent Systems*, pages 21–23, April 1997. (Cité page 34.)
- V. D. Dang, R. K. Dash, A. Rogers, et N. R. Jennings. Overlapping coalition formation for efficient data fusion in multi-sensor networks. *AAAI*, pages 635–640, 2006. (Cité page 38.)
- V. D. Dang et N. R. Jennings. Generating coalition structures with finite bound from the optimal guarantees. *AAMAS*, pages 564–571, 2004. (Cité pages 12 et 13.)
- E. Elkind, T. Rahwan, et N. R. Jennings. *Computational coalition formation*. G. Weiss, Ed. MIT Press, 2013. (Cité pages 1, 2 et 9.)
- P.F. Faye, S. Aknine, O. Shehory, et M. Sène. Adapting agent’s interactions in dynamic contexts. *Joint International Conference of the INFORMS GDN*, June 2014a. (Cité page 43.)
- P.F. Faye, S. Aknine, O. Shehory, et M. Sène. Formation de coalition stable dans un contexte non-déterministe et instable. *Reconnaissance de Formes et l’Intelligence Artificielle (RFIA)*, July 2014b. (Cité page 43.)
- P.F. Faye, S. Aknine, O. Shehory, et M. Sène. Stabilizing agent’s interactions in dynamic contexts. *The 28th IEEE International Conference on Advanced Information Networking and Applications (AINA 2014)*, May 2014c. (Cité page 43.)
- P.F. Faye, S. Aknine, O. Shehory, et M. Sène. Stable collaboration in unstable environments. *International workshop on cooperative games in multiagent systems (CoopMAS)*, May 2014d. (Cité page 43.)
- J. Ferber. *LES SYSTEMES MULTI-AGENTS. Vers une intelligence collective*. Dunod, December 1997. (Cité pages 1 et 2.)
- J. A. Filar et L. A. Petrosjan. Dynamic cooperative games. *International Game Theory Review*, pages 47–65, March 2000. (Cité page 39.)
- T. Génin et S. Aknine. Coalition formation strategies for self-interested agents. *ECAI*, pages 418–422, July 2008. (Cité page 32.)
- T. Génin et S. Aknine. Coalition formation strategies for self-interested agents in hedonic games. *ECAI*, pages 1015–1016, August 2010a. (Cité page 32.)

- T. Génin et S. Aknine. Coalition formation strategies for self-interested agents in task oriented domains. *WI-IAT*, pages 205–212, August 2010b. (Cité page 31.)
- M. Grabisch. Fuzzy integral in multicriteria decision making. *Fuzzy Sets and Systems*, 69 :279–298, 1995. (Cité page 137.)
- F. Guerin, B. Dumon, et R. Hambli. Determining the shape parameter of a weibull distribution from mechanical damage models. *IEEE International Conference on Reliability and Maintainability*, pages 156–160, January 2001. (Cité page 84.)
- B. Horling, R. Vincent, R. Mailler, J. Shen, R. Becker, K. Rawlins, et V. Lesser. Distributed sensor network for real time tracking. *AGENTS'01 Autonomous Agents*, pages 417–424, 2001. (Cité page 38.)
- L. Hribar et D. Duka. Weibull distribution in modeling component faults. *52nd International Symposium ELMAR, IEEE Computer Society*, pages 183–186, September 2010. (Cité page 84.)
- K. Jahanbakhsh, V. King, et G. C. Shojai. Predicting missing contacts in mobile social networks. *Pervasive and Mobile Computing, Elsevier*, 8, issue 5 :698–716, October 2012. (Cité page 114.)
- T. Jiang, G. Theodorakopoulos, et J. S. Baras. Coalition formation in manets. *Army Science Conference, Orlando, FL, USA*, pages 1–8, November 2006. (Cité page 37.)
- J. P. Kahan et A. Rapoport. *Theories of coalition formation*. L. Erlbaum Associates, 1984. ISBN 0898592984. (Cité pages 9 et 21.)
- H. Keinänen et M. Keinänen. Simulated annealing for coalition formation. *ECAI*, pages 857–858, 2008. (Cité page 15.)
- S. Ketchpel. Forming coalitions in the face of uncertain rewards. *AAAI*, pages 414–419, July 1994. (Cité page 34.)
- M. A. Khan, D. Turgut, et L. Bölöni. Optimizing coalition formation for tasks with dynamically evolving rewards and nondeterministic action effects. *Autonomous Agents and Multi-Agent Systems*, pages 415–438, May 2011. (Cité page 37.)
- M. Klusch et A. Gerber. Issues of dynamic coalition formation among rational agents. *Intelligent Systems, IEEE Computer Society*, pages 42–47, June 2002. (Cité pages 36 et 37.)
- M. Klusch et O. Shehory. A polynomial kernel-oriented coalition formation algorithm for rational information agents. *International Conference on Multi-Agent Systems (ICMAS)*, pages 157–164, December 1996. (Cité page 34.)

- L. Kranich, A. Perea, et H. Peters. Core concepts for dynamic tu games. *International Game Theory Review*, pages 43–61, 2005. (Cité page 39.)
- J. Levine et R. E. Dalton. Minimum periods, modulo p , of first-order bell exponential integers. *Math. Comput.*, 16 :416–423, 1962. (Cité pages 10 et 29.)
- C. Li et K. Sycara. Algorithms for combinatorial coalition formation and payoff division in an electronic marketplace. *AAMAS*, pages 120–127, November 2002. (Cité page 34.)
- R. D. Luce et H. Raiffa. *Games and Decisions : Introduction and Critical Survey*. John Wiley and Sons, Ltd., Rutgers, The State University of New Jersey, 1957. (Cité page 33.)
- D. W. Manchala. E-commerce trust metrics and models. *IEEE Internet Computing*, 4 :36–44, April 2000. (Cité page 36.)
- M. Mares. *Fuzzy cooperative games - Cooperation with vague expectations*. Springer-Verlag Berlin Heidelberg, June 2001. ISBN 978-3-7908-1820-8. (Cité page 34.)
- A. W. Marshall et I Olkin. *Inequalities : Theory of Majorization and Its Applications*, volume 143. Academic Press, Inc., New York, 1979. (Cité page 110.)
- M. F. Melvin. Modeling and supporting task-oriented group processes : Purposeful complex adaptive systems and evolutionary systems design. *GDN*, pages 305–317, December 1996. (Cité page 32.)
- I. Milchtaich et E. Winter. Stability and segregation in group formation. *Games and Economic Behavior*, pages 318–346, February 2002. (Cité page 36.)
- Pragnesh Jay Modi. Distributed constraint optimization for multiagent systems. *PhD, University of Southern California, Los Angeles, CA, USA*, 2003. (Cité page 14.)
- J. F. Nash. Equilibrium points in n -person games. *Proceedings of the National Academy of Sciences*, 36 : 48–49, January 1950. (Cité pages 44, 49 et 55.)
- R. Ortner. Linear dependence of stationary distributions in ergodic markov decision processes. *Operations Research Letters*, pages 619–626, January 2007. (Cité pages 74 et 169.)
- M.J. Osborne et A. Rubinstein. A course in game theory. *MIT Press*, 1994. (Cité page 19.)
- T. Rahwan et N. R. Jennings. An algorithm for distributing coalitional value calculations among cooperating agents. *AIJ*, pages 535–567, June 2007. (Cité page 11.)
- T. Rahwan et N. R. Jennings. Coalition structure generation : Dynamic programming meets anytime optimization. *AAAI*, pages 156–161, 2008a. (Cité page 13.)

- T. Rahwan et N. R. Jennings. An improved dynamic programming algorithm for coalition structure generation. *AAMAS*, 3 :1417–1420, May 2008b. (Cité pages 13 et 30.)
- T. Rahwan, T. Michalak, et N. R. Jennings. A hybrid algorithm for coalition structure generation. *AAAI*, pages 1443–1449, July 2012a. (Cité pages 13 et 30.)
- T. Rahwan, T. P. Michalak, E. Elkind, P. Faliszewski, J. Sroka, M. Wooldridge, et N. R. Jennings. Constrained coalition formation. *AAAI*, 2011. (Cité page 14.)
- T. Rahwan, T. P. Michalak, N. R. Jennings, M. Wooldridge, et P. McBurney. Coalition structure generation in multi-agent systems with positive and negative externalities. *IJCAI*, pages 257–263, 2009a. (Cité page 15.)
- T. Rahwan, T. P. Michalak, N. R. Jennings, M. Wooldridge, et P. McBurney. Anytime coalition structure generation in multi-agent systems with positive or negative externalities. *AIJ*, 2012b. (Cité page 15.)
- T. Rahwan, S. D. Ramchurn, V. D. Dang, et N. R. Jennings. Near-optimal anytime coalition structure generation. *IJCAI*, pages 2365–2371, 2007. (Cité page 13.)
- T. Rahwan, S. D. Ramchurn, A. Giovannucci, et N. R. Jennings. An anytime algorithm for optimal coalition structure generation. *JAIR*, 34 :521–567, January 2009b. (Cité pages 12, 13 et 30.)
- M. H. Rothkopf, A. Pekeč, et R. M. Harstad. Computationally manageable combinatorial auctions. *Management Science*, 44 :1131–1147, August 1998. (Cité page 12.)
- J. C. Rubiera, J. M. M. Lopez, et J. D. Muro. A fuzzy model of reputation in multi-agent systems. *5th International Conference on Autonomous Agents*, pages 25–26, June 2001. (Cité page 36.)
- T. Sandholm. Distributed rational decision making. In *the textbook Multiagent Systems : A Modern Introduction to Distributed Artificial Intelligence*, Weiss, G., ed., MIT Press, pages 201–258, May 1999. (Cité page 33.)
- T. Sandholm, K. Larson, M. Andersson, O. Shehory, et F. Tohmé. Anytime coalition structure generation with worst case guarantees. *arXiv preprint cs/9810005*, pages 46–53, July 1998. (Cité page 33.)
- T. W. Sandholm, K. Larson, M. Andersson, O. Shehory, et F. Tohmé. Coalitions structure generation with worst case guarantees. *AI Journal*, pages 209–238, July 1999. (Cité pages 9, 12, 13 et 101.)
- T. Scully, M. G. Madden, et G. Lyons. Coalition calculation in a dynamic agent environment. *ICML*, page 93, 2004a. (Cité page 15.)

- T. Scully, M. G. Madden, et G. Lyons. Coalition calculation in a dynamic agent environment. *ICML*, page 93, July 2004b. (Cité page 31.)
- S. Sen, I. Sen, et P. S. Dutta. Searching for optimal coalition structures. *Fourth International Conference on MultiAgent Systems*, pages 286–292, July 2000. (Cité page 15.)
- L. S. Shapley. *A Value for N-Person Games*. Princeton University Press, USA, 1944. (Cité page 19.)
- O. Shehory. Coalition formation : Towards feasible solutions. *Fundamenta Informaticae*, pages 107–124, Novembre 2004. (Cité page 33.)
- O. Shehory et S. Kraus. Formation of overlapping coalitions for precedence-ordered task-execution among autonomous agents. *International Conference on Multiagent Systems (ICMAS)*, pages 330–337, December 1996. (Cité pages 36 et 84.)
- O. Shehory et S. Kraus. Methods for task allocation via agent coalitions formation. *AI Journal*, pages 165–200, May 1998. (Cité pages 11 et 101.)
- J. S. Sichman. *DEPINT : Dependence-Based Coalition Formation in an Open Multi-Agent Scenario*, volume 1. JASSS, March 1998. (Cité page 36.)
- L. Sless, N. Hazon, S. Kraus, et M. Wooldridge. Forming coalitions and facilitating relationships for completing tasks in social networks. *AAMAS*, pages 261–268, May 2014. (Cité page 38.)
- J. Suijs, P. Borm, A. De Waegenaere, et S. Tijs. Cooperative games with stochastic payoffs. *European Journal of Operational Research*, pages 193–205, February 1999. (Cité page 35.)
- R. S. Sutton et A. G. Barto. *Reinforcement Learning : An Introduction*. MIT Press, February 1998. (Cité page 31.)
- S. Ueda, A. Iwasaki, M. Yokoo, M. Silaghi, K. Hirayama, et T. Matsui. Coalition structure generation based on distributed constraint optimization. *AAAI*, 2010. (Cité page 14.)
- D. Šišlák, M. Pěchouček, M. Reháč, J. Tožička, et P. Benda. Solving inaccessibility in multiagent systems by mobile middle agent. *Multiagent and Grid Systems*, pages 73–87, April 2005. (Cité page 37.)
- J. Yamamoto et K. Sycara. A stable and efficient buyer coalition formation scheme for e-marketplaces. *AGENTS'01 Autonomous Agents*, pages 576–583, June 2001. (Cité page 34.)
- R. D. Yates et D. J. Goodman. *Probability and Stochastic Processes : A Friendly Introduction for Electrical and Computer Engineers*. John Wiley and Sons, INC, Rutgers, The State University of New Jersey, 2005. (Cité pages 46, 47, 56, 109 et 169.)

- D. Ye, M. Zhang, et D. Sutanto. Integrating self-organisation into dynamic coalition formation. *The 11th International Conference on Autonomous Agents and Multiagent Systems*, pages 1253–1254, June 2012. (Cité page 36.)
- D. Yun Yeh. A dynamic programming approach to the complete set partitioning problem. *BIT*, 26 : 467–474, August 1986. (Cité page 12.)
- B. Yu et M. P. Singh. A social mechanism of reputation management in electronic communities. *International Workshop on Cooperative Information Agents*, pages 154–165, July 2000. (Cité page 36.)
- Y. Zick, G. Chalkiadakis, et E. Elkind. Overlapping coalition formation games : Charting the tractability frontier. *The 11th International Conference on Autonomous Agents and Multiagent Systems*, pages 787–794, June 2012. (Cité pages 36 et 84.)
- G. Zlotkin et J. S. Rosenschein. Coalition, cryptography, and stability : Mechanisms for coalition formation in task oriented domains. *AAAI*, pages 432–437, 1994. (Cité page 32.)

Titre Modèles de formation de coalitions stables dans un contexte ad-hoc et stochastique.

Résumé Les approches multi-agents abordent la problématique de la coordination distribuée, généralement, par des mécanismes de négociation fondés sur les modèles de préférences et la recherche du bien-être social des agents. Cependant, ces approches ne tiennent pas réellement compte de la dynamique des tâches ou de la disponibilité aléatoire des agents. En décentralisant le processus de coordination, on se confronte naturellement à la difficulté de la corrélation des comportements locaux par rapport aux propriétés globales souhaitées du système. De plus, pour le contexte stochastique considéré dans cette thèse, le problème de la coordination exige une stabilisation dynamique des interactions et une convergence rapide vers les meilleures solutions. Nous proposons, dans ce travail, des méthodes de coordination permettant une adaptation dynamique des agents aux activités de leurs utilisateurs et à leur environnement. Nos méthodes intègrent les spécificités intrinsèques des actions entreprises par les agents et celles de leurs tâches qui évoluent de manière non déterministe. Travailler dans un contexte ad-hoc et dynamique, pour les agents, empêche : 1- l'existence d'une vue globale du système qui reflète une image complète de l'environnement de déploiement ; 2- l'existence de connaissances a priori sur la manière de se coordonner en raison de l'absence d'une structure centralisée et de la disponibilité aléatoire des entités considérées. Nous avons proposé différentes stratégies comportementales pour faciliter la stabilisation dynamique des interactions entre les agents et la convergence vers les meilleurs états de coordination. Pour ce faire, nous avons intégré dans les modèles de raisonnement des agents, des mécanismes permettant l'établissement d'alliances entre les agents et un support pour les recommandations dynamiques. Notre conception des alliances et des recommandations permet à un agent d'évoluer de manière autonome, d'identifier dynamiquement les agents voisins fiables avec qui coopérer et de former avec son voisinage des partitions Nash-stables selon les exigences de l'environnement de déploiement (disponibilité, urgence des tâches stochastiques,...). Nous avons développé des stratégies de proposition de coalitions fondées sur les négociations et les alliances pour guider la recherche et la génération de nouvelles propositions. Pour répondre à la difficulté de corrélation entre les comportements locaux des agents et les propriétés de l'environnement de déploiement des agents, nous utilisons de manière originale les modèles markoviens. Nous avons développé des stratégies d'acceptation de propositions fondées sur le compromis, la fiabilité et la recherche d'agents suppléants dans le cas d'une indisponibilité ou pour la stabilisation dynamique des interactions ou pour une convergence rapide vers les meilleures solutions. Dans la continuité de ces travaux, nous nous sommes focalisés sur : (1) la prise en compte des interdépendances entre les agents, (2) le développement de modèles de coordination inter groupe d'agents pour augmenter l'efficacité des agents dans un souci d'optimisation des coûts (énergies et ressources consommées) imputés aux composants ad-hoc communicants (des PDA, Capteurs, Robots, ...) où les agents sont déployés. Pour répondre au premier point, nous avons proposé le modèle S-NRB (Sequentiel Non-return Broadcast) et le modèle

P-NRB (Parallel Non-return Broadcast) pour la coordination distribuée qui cherchent à maximiser le bien-être social des agents. Le second point, nous a amené à proposer le modèle DMS (Dynamic Merging and Splitting) de formation de coalitions intégrant de manière dynamique ces interdépendances. Toutes ces propositions ont été étudiées de manière théorique et expérimentale pour mettre en exergue les propriétés intrinsèques de nos méthodes. Nous avons ainsi développé un simulateur spécialement pour le type de contexte considéré.

Mots-clés Ad-hoc, Stabilité, Coalition, Coordination, Stochastique.

Title Stable coalition formation mechanisms in ad-hoc and stochastic context.

Abstract An ad-hoc and stochastic context prevents : 1- the existence of a global view of the system that reflects a complete image of the deployment environment ; 2- the existence of a priori knowledge because of the lack of a centralized structure, the dynamic of the tasks and the random availability of the entities. We proposed different strategies to facilitate the dynamic stabilization of the interactions between the agents and the convergence towards better coordination states. Our conception of alliances and recommendations allows an agent to evolve independently, to dynamically identify reliable neighboring agents with whom to cooperate and to form Nash-stable or Core stable coalitions according to the requirements of the deployment environment. To face with the challenges of the correlation between local behavior of the agents and the properties of their environment, we use in an original way the Markovian models. We also focused on taking into account the interdependencies between the agents to increase their efficiency in order to optimize the imputed costs of the ad-hoc components where the agents are deployed. This led us to propose both mechanisms, the S-NRB (Sequential Non-return Broadcast) and the P-NRB (Parallel Non-return Broadcast) for distributed coordination seeking to maximize the social welfare of the agents. To highlight the intrinsic properties of our methods, our whole proposals have been studied theoretically and experimentally through our simulator.

Keywords Ad-hoc, Stability, Coalition, Coordination, Stochastic.